

# Contribuição para o Controlo Adaptativo Não Linear: Redes Neurais Recorrentes no Contexto da Regulação da Saída

*Jorge Manuel Oliveira Henriques*

*Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra*



Tese submetida à  
UNIVERSIDADE DE COIMBRA  
Para obtenção do grau de Doutor em  
Engenharia Informática

*Dezembro de 2000*



Contribuição para o Controlo  
Adaptativo Não Linear:  
Redes Neurais Recorrentes no  
Contexto da Regulação da Saída

*Jorge Manuel Oliveira Henriques*

*Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra*



Tese submetida à  
UNIVERSIDADE DE COIMBRA  
Para obtenção do grau de Doutor em  
Engenharia Informática

*Dezembro de 2000*



*Tese realizada sob a orientação de:*

*Professor Doutor António Dourado Pereira Correia*

Professor Associado com Agregação do  
Departamento de Engenharia Informática da  
Faculdade de Ciências e Tecnologia da  
Universidade de Coimbra



# Agradecimentos

**E**ste trabalho foi realizado sob a orientação científica do Prof. Dr. António Dourado. Quero pois, em primeiro lugar, expressar o meu reconhecimento pelo seu apoio, pela sua amizade e pela motivação sempre demonstradas ao longo dos mais de dez anos que com ele tenho tido a oportunidade de trabalhar.

Ao CISUC, Centro de Informática e Sistemas da Universidade de Coimbra, na pessoa do seu presidente Prof. Dr. António Dourado, gostaria de agradecer as condições proporcionadas para a realização do trabalho.

Ao Prof. André Titli, LAAS-CNRS, *Laboratoire d'Analyse et d'Architecture des Systèmes-Centre National de la Recherche Scientifique*, Toulouse-França, agradeço a possibilidade que me foi oferecida de ter permanecido no LAAS. Pela suas sugestões e, sobretudo, pela disponibilidade sempre demonstradas, o meu obrigado.

Ao Prof. Bernardino Castillo-Toledo os meus sinceros agradecimentos. Foi além de um excelente conselheiro um bom amigo. Algumas das ideias aqui apresentadas foram desenvolvidas com ele durante a minha estada no LAAS. Pelo que beneficieei da interacção que com ele mantive gostaria de lhe expressar o meu reconhecimento.

Ainda aos colegas do INSA-*Institut National des Sciences Appliquées*, em especial à Sabina, Maria, Paulo, Armando, Mário, Chema e Artur, o meu obrigado pelos momentos agradáveis que com eles partilhei.

Aos colegas do grupo de Computação Adaptativa do DEI - *Departamento de Engenharia Informática*, Dr.<sup>a</sup> Bernardete Ribeiro, Eng.s Alberto Cardoso, Paulo Carvalho, Carlos Vaz, Francisco Duarte, Carlos Pereira, Paulo Gil, Amâncio Santos e José Victor e Eng. Rui Paiva do grupo de Sistemas e Multimédia o meu agradecimento pelo bom ambiente proporcionado e pelas discussões estimulantes, que em grande parte contribuiriam para o desenvolvimento desta tese.

De entre estes, gostaria de agradecer ao Eng. Paulo Gil de uma forma especial. Pelas suas observações, quase sempre pertinentes, e que tantas vezes me levaram a inflectir noutras direcções, pelas conversas de tantos momentos partilhados e, principalmente, pelas alegrias e tristezas que nestes três anos passamos juntos.

Para todos os funcionários do *DEI*, e em especial para a D.<sup>a</sup> Amália, ficam também os meus agradecimentos.

À minha família, e de uma forma muito especial aos meus pais, o meu obrigado.

Às minhas garotas: filha, mulher e sogra o obrigado tanto pelo apoio, indispensável para que este trabalho fosse realizado, como pela paciência que tiveram, têm, e certamente hão-de continuar a ter.

Finalmente, a todos os que não mencionei e, que de uma maneira ou de outra me ajudaram na prossecução deste trabalho, o meu muito obrigado.

Coimbra, 28 de Dezembro de 2000

A handwritten signature in black ink, reading "Jorge Manuel Pereira Almeida". The signature is written in a cursive style with a large initial 'J'.

O trabalho de investigação apresentado nesta dissertação foi parcialmente financiado pelo Ministério da Ciência e Tecnologia, ao abrigo do programa PRAXIS XXI.

*À Laura, à Teresa*

*Aos meus pais*



# Resumo

A evolução do controlo automático tem-se caracterizado nos últimos anos por algum antagonismo entre duas escolas: uma baseada em métodos analítico-algébricos e outra baseada em técnicas de *soft computing* (computação adaptativa). Qualquer uma delas tem desenvolvido esforços para encontrar soluções para problemas de controlo não linear, complexos e incertos. Por um lado a primeira, a escola dita convencional, usando métodos rigorosos, baseados em técnicas bem estabelecidas de sistemas lineares e não lineares, oferece uma teoria coerente mas tem apresentado algumas dificuldades, principalmente quando não é possível obter modelos precisos para os processos a controlar. Por outro lado, baseada principalmente em metodologias que usam as capacidades das redes neuronais e dos sistemas difusos, têm sido desenvolvido um número elevado de métodos e de arquiteturas, que na prática têm proporcionado resultados aceitáveis para problemas complexos, mas que em termos de coerência, sistematização e generalidade requerem ainda muitos desenvolvimentos.

Torna-se cada vez mais claro que apenas a colaboração das duas escolas pode conduzir a uma nova etapa no desenvolvimento do controlo automático, pretendendo o trabalho apresentado nesta dissertação caminhar nessa direcção.

Neste contexto, procura-se contribuir para o desenvolvimento e compreensão das potencialidades de estruturas híbridas, concebendo sistemas de controlo que simultaneamente tirem partido das potencialidades das metodologias de *soft computing* e das técnicas convencionais.

Tendo em vista estes objectivos foram realizados quer estudos teóricos, quer estudos experimentais relativos a estruturas que se baseiam neste princípio. Aquela que se apresenta em pormenor nesta dissertação caracteriza-se por combinar numa estrutura adaptativa redes neuronais recorrentes e a teoria de regulação da saída (*output regulation*), uma técnica de controlo não linear. Pretende-se com esta abordagem utilizar tanto as capacidades de identificação das redes neuronais como as propriedades de estabilidade da teoria de regulação.

Principalmente pelas suas propriedades de aproximação universal, e pela sua interpretabilidade como modelo de estados não lineares, propõe-se a aplicação de uma rede neuronal recorrente na modelização do sistema a controlar. Assim sendo, este pode ser substituído pelo modelo neuronal, transformando o problema de controlo num problema de controlo não linear, adequado ao projecto de técnicas não lineares, de que a teoria de regulação é um caso particular.

O objectivo principal da teoria de regulação consiste em projectar um controlador de forma a que o sistema em malha fechada seja estável e simultaneamente o erro de controlo convirja para zero. Aplicada a sistemas não lineares, a solução do problema de controlo é reduzido à resolução de um sistema não linear de equações de diferenças para as quais é difícil, ou mesmo impossível, encontrar uma solução analítica. Nesta dissertação, de forma a solucionar o problema de regulação propõe-se, baseado na arquitectura particular da rede recorrente, um procedimento iterativo análogo a um algoritmo de colocação de pólos, capaz de assegurar a convergência da solução.

Contudo, se a determinação dos parâmetros que caracterizam a rede for efectuada *à priori* não se pode esperar que o modelo neuronal seja preciso. De facto, existem sempre erros de modelização, situações imprevisíveis e variações do próprio sistema. Assim, de forma a ultrapassar o problema, considera-se o modelo neuronal inserido numa estrutura adaptativa, na qual os seus parâmetros são adaptados a partir de dados de entrada-saída.

Para o efeito, interpreta-se a rede recorrente como um observador não linear e propõe-se, por analogia com um filtro de Kalman dual, um método capaz de simultaneamente estimar os estados e os parâmetros do modelo neuronal. Recorrendo à teoria de Lyapunov e a técnicas de observação não linear assegura-se a estabilidade e convergência do método proposto.

Finalmente, são efectuados estudos experimentais e de simulação no sentido de validar as características de estabilidade da metodologia de controlo proposta, quando aplicada a sistemas com características distintas. Compara-se ainda o seu desempenho face a técnicas convencionais, nomeadamente controladores *PI* e de realimentação de variáveis de estado.

*Palavras Chave:* Controlo adaptativo não linear, redes neuronais recorrentes, observadores não lineares, regulação da saída, estabilidade.

# Abstract

The evolution of automatic control in the last few years has been characterized by some antagonism between two schools: one based on the analytical-algebraic approach and the other based on processing tools stemming from soft computing theory. Both have been working on to develop control methodologies for complex systems. The analytical-algebraic school, using rigorous methods for linear and non-linear systems, has built a coherent body of knowledge, however still fails to solve problems where it is not possible to obtain rigorous models. The other school, using mainly neural networks and fuzzy systems, has developed a number of methods and control architectures that may solve in practice difficult problems. Nevertheless, the resulting body of knowledge lacks coherence, systematization and generality.

It is now becoming clear that only the collaboration of the two schools may lead to a new stage in the history of automatic control science and technology. The present work intends to be a contribution in this direction.

In this context, some analytical and experimental results have been carried out in order to developed hybrid control structures, by integrating the attributes of soft computing methodologies, like neural and fuzzy systems, with the well-established system control techniques. Particularly, in this work it is presented a structure that fuses a recurrent neural network with the output regulation control theory to obtain a control methodology for general non-linear discrete time systems. It is intended with this approach to profit from the identification capabilities of neural networks with the stability properties of the output regulation theory.

Given the universal approximation properties, as well as its intrinsic analogy to the non-linear state space form, a recurrent neural network is derived and applied for modeling non-linear plants. This network architecture has the well-known properties of multilayer perceptrons and has the ability to incorporate temporal behavior. This is the starting-point to neural networks in identification and control of non-linear systems. The recurrent neural model can replace the unknown system, transforming the original problem of control into a non-linear control problem, suitable to be designed by a non-linear control technique, like the output regulation theory.

The main goal of the output regulation theory is to derive a control law such that simultaneously the closed loop system is stable and the tracking error converges to zero. If the system is linear time invariant and their parameters are completely known, a linear regulator can be designed. However, if the system is non-linear the problem becomes tougher. In this case, the solution of the output regulator problem leads to a set of non-linear difference equations, to which is very difficult, or even impossible, to derive a closed solution. To solve the regulator equations an iterative procedure based on a pole placement algorithm is proposed ensuring the convergence of the regulator equations.

Tough, assuming an exact knowledge of the systems parameters in many cases is not viable. In fact, if the training is performed off-line, the non-linear neural model is not exact in the presence of disturbances, parameter variations and uncertainties. In these conditions, the output regulator cannot assure convergence of the tracking error to zero. To overcome this an adaptive strategy is applied, providing on-line estimation of the network parameters. Estimation is implemented on-line, based on input-output data, ensuring the stability and convergence of the estimation error. The estimation strategy is developed following a dual Kalman filter strategy using both Lyapunov theory and non-linear observation techniques.

Simulation and experimental results collected from several laboratorial processes are used to confirm the viability and effectiveness of the proposed methodology. The control performance of the proposed technique is compared to well-established techniques, like *PI* controllers and state space feedback controllers.

*Key words:* Non-linear adaptive control, recurrent neural networks, non-linear observers, output regulation theory, stability.





# Índice

## 1.

### Introdução

1.1	Motivações .....	3
1.2	Objectivos .....	9
1.3	Contribuições .....	12
1.4	Estrutura.....	15
1.5	Convenções e Meios.....	18

## 2.

### Redes Neurais Aplicadas à Modelização

2.1		
	Modelização de Sistemas .....	21
2.1.1	<i>Generalidades</i> .....	21
2.1.2	<i>Técnicas de Soft Computing Aplicadas à Identificação</i> .....	29
2.1.3	<i>Redes Neurais</i> .....	31
2.1.4	<i>Conclusão</i> .....	38
2.2		
	Estruturas de Redes Neurais .....	39
2.2.1	<i>Redes Neurais Estáticas</i> .....	39
2.2.2	<i>Redes Dinâmicas a partir de Redes Estáticas</i> .....	44
2.2.3	<i>Estruturas Recorrentes</i> .....	47
2.2.4	<i>Estruturas Neurais Affine</i> .....	53

2.3	
Estimação de Parâmetros em Redes Recorrentes	55
2.3.1	Generalidades..... 55
2.3.2	Extensão do Método de BP a Redes Recorrentes..... 58
2.3.3	Métodos de Optimização Não Linear ..... 64
2.3.4	Estimação de Estados ..... 66
2.3.5	Estimação de Parâmetros e de Estados com Técnicas de Observação ..... 71
2.3.6	Técnicas de Observação Aplicadas às Redes Neurais ..... 74
2.3.7	Metodologia Proposta..... 76
2.4	
Conclusão	79

## 3.

### Redes Neurais Aplicadas ao Controlo

---

3.1	
Controlo de Sistemas	83
3.1.1	Introdução ..... 83
3.1.2	Controlo Convencional Adaptativo ..... 84
3.1.3	Controlo Baseado em Técnicas de Soft Computing..... 89
3.1.4	Redes Neurais no Controlo ..... 95
3.2	
Estruturas Neurais de Controlo	99
3.2.1	Introdução ..... 99
3.2.2	Classificação de Estruturas..... 100
3.2.3	Metodologia Proposta: Controlo Não Linear Adaptativo Baseado no Modelo Neuronal ..... 112
3.3	
Controlo Não Linear	115
3.3.1	Introdução ..... 115
3.3.2	Linearização Retroactiva ..... 116
3.3.3	Aplicação de Técnicas de Linearização Retroactiva a Redes Neurais ..... 122
3.3.4	Estrutura de Controlo Proposta..... 128
3.4	
Conclusão	129

## 4.

### Regulação da Saída com Redes Neurais Recorrentes

---

4.1		
Estrutura Neuronal Proposta		133
4.1.1	<i>Estruturas Affine</i> .....	133
4.1.2	<i>Rede Neuronal Recorrente Proposta</i> .....	134
4.2		
Estimação de Parâmetros na Metodologia RSRN		138
4.2.1	<i>Estimação Simultânea de Parâmetros e Estados</i> .....	138
4.2.2	<i>Estimação de Estados</i> .....	141
4.2.3	<i>Estimação de Parâmetros</i> .....	148
4.2.4	<i>Algumas Considerações</i> .....	154
4.3		
Regulação da Saída na Metodologia RSRN		155
4.3.1	<i>Introdução</i> .....	155
4.3.2	<i>Teoria de Regulação da Saída</i> .....	156
4.3.3	<i>Algoritmo Iterativo para a Resolução das Equações de Regulação</i> .....	163
4.3.4	<i>Aplicação do Algoritmo Iterativo na Estimação de Parâmetros</i> .....	170
4.4		
Conclusão: Metodologia RSRN		173

## 5.

### Casos de Estudo

---

5.1		
Processo Térmico - PT326		179
5.1.1	<i>Introdução</i> .....	179
5.1.2	<i>Identificação Experimental</i> .....	184
5.1.3	<i>Inicialização de Parâmetros</i> .....	186
5.1.4	<i>Variação da Dinâmica</i> .....	197
5.1.5	<i>Análise de Resultados</i> .....	208

5.2		
Reactor Químico - CSTR		210
5.2.1	<i>Introdução</i> .....	210
5.2.2	<i>Identificação Experimental</i> .....	214
5.2.3	<i>Controlador PI</i> .....	218
5.2.4	<i>Controlador RSRN</i> .....	221
5.2.5	<i>Análise de Resultados</i> .....	229
5.3		
Tanques Acoplados - DTS200		230
5.3.1	<i>Introdução</i> .....	230
5.3.2	<i>Identificação Experimental</i> .....	234
5.3.3	<i>Controlo por Realimentação de Variáveis de Estado</i> .....	236
5.3.4	<i>Perturbações (Introdução de Fugas)</i> .....	242
5.3.5	<i>Análise de Resultados</i> .....	246
5.4		
Conclusão		247

## 6.

### Conclusão e Perspectivas

---

6.1		
Conclusão		251
6.2		
Perspectivas		253

### Apêndices

A.1	.....	258
A.2	.....	262
B.1	.....	265
B.2	.....	271
B.3	.....	291
B.4	.....	297
B.5	.....	317

# Notação

## Matricial

$rank(A)$	Característica da matriz $A$ .
$\mathfrak{R}^{n,m}$	Conjunto das matrizes reais de dimensão $(n,m)$ .
$\mathcal{C}^p$	Conjunto de funções $p$ vezes diferenciáveis.
$\mathcal{C}^\infty$	Conjunto de funções infinitamente diferenciáveis.
$\mathbb{N}$	Conjunto dos números naturais.
$\mathfrak{R}$	Conjunto dos números reais.
$\mathbb{N}^n$	Conjunto dos vectores naturais de dimensão $n$ .
$\mathfrak{R}^n$	Conjunto dos vectores reais de dimensão $n$ .
$L_g f$	Derivada de Lie da função $f$ em relação ao vector $g$ .
$diag(A)$	Diagonal da matriz $A$ .
$\delta(\cdot)$	Função de Dirac.
$\Delta x$	Incremento de $x$ .
$A^{-1}$	Inversa da matriz $A$ .
$I_{n,m}$	Matriz identidade de dimensão $(n,m)$ .
$0_{n,m}$	Matriz nula de dimensão $(n,m)$ .
$\angle \cdot \angle, \angle \cdot \angle_2$	Norma Euclidiana ou quadrática.
$G, \nabla_f$	Primeira derivada ou gradiente de uma função $f$ .
$H, \nabla_f^2$	Segunda derivada ou <i>Hessiana</i> de uma função $f$ .
$tr(A)$	Traço da matriz $A$ .
$A^T$	Transposta da matriz $A$ .
$ \cdot $	Valor absoluto.

## Identificação

$u(k)$	Acção de controlo no instante $k$ .
$a_i(k)$	Activação do neurónio $i$ no instante $k$ .
$Z^N$	Conjunto de dados de dimensão $N$ .
$J$	Crítério ou função de custo.
$e_u(k)$	Erro de identificação (de controlo) no instante $k$ . $e_u(k) = u(k) - \hat{u}(k).$
$e_y(k)$	Erro de identificação (de saída) no instante $k$ . $e_y(k) = y(k) - \hat{y}(k) = y(k) - y_n(k)$
$\varepsilon(k)$	Erro de identificação (estados) no instante $k$ . $\varepsilon(k) = x(k) - \hat{x}(k) = x(k) - x_n(k)$
$x(k), x_p(k)$	Estado do sistema real no instante $k$ .
$\hat{x}(k), x_n(k)$	Estado estimado ou activação de um neurónio interno no instante $k$ .
$\sigma(\cdot)$	Função de activação de um neurónio.
$k$	Instante de amostragem no domínio discreto.
$T$	Intervalo de amostragem.
$B$	Matriz de entrada (espaço de estados)
$C$	Matriz de saída (espaço de estados)
$n_u$	Número de entradas.
$n_n, n$	Número de estados do modelo neuronal.
$n_p$	Número de estados do processo real.
$n_b$	Número de regressões da variável de entrada.
$n_a$	Número de regressões da variável de saída.
$n_y$	Número de saídas.
$q^{-1}$	Operador atraso unitário.
$A$	Parte linear (espaço de estados)
$D$	Parte não linear (espaço de estados)
$w_{ij}$	Peso da ligação do neurónio $i$ para o neurónio $j$ .

$W, \theta$	Pesos da rede, Parâmetros do modelo.
$\mathcal{N}$	Rede neuronal
$\zeta(k)$	Ruído Branco Gaussiano.
$y_d(k)$	Saída desejada ou referência no instante $k$ .
$\hat{y}(k), y_n(k)$	Saída do modelo de predição ou saída da rede neuronal no instante $k$ .
$y(k)$	Saída do sistema real no instante $k$ .
$t$	Variável tempo no domínio contínuo
$U(k)$	Vector de regressão definido por $U(k) = [ u(k), \dots, u(k - n_b) ]$ .
$Y(k)$	Vector de regressão, definido por $Y(k) = [ y(k), \dots, y(k - n_a) ]$ .
$\varphi(\cdot)$	Vector de regressão.

## Controlo

$e_c(k)$	<p>Erro de controlo, definido para o processo real:</p> $e_c(k) = y_d(k) - y(k)$ $e_c(k) = ( y_d(k) - \hat{y}(k) ) + ( \hat{y}(k) - y(k) )$ $e_c(k) = e_n(k) - e_y(k).$
$e_n(k)$	<p>Erro de regulação, definido para o modelo do processo (neuronal):</p> $e_n(k) = y_d(k) - \hat{y}(k) = y_d(k) - y_n(k).$
$r$	Grau relativo.
$\underline{B}$	Linearização do modelo não linear, matriz de entrada.
$\underline{A}$	Linearização do modelo não linear, matriz de estado.
$\underline{C}$	Linearização do modelo não linear, matriz de saída.
$q_0, q_1$	Parâmetros de um controlador <i>PI</i> .
$x^o$	Ponto de equilíbrio ou regime de operação.
$\rho$	Vector constituído por perturbações e saída desejada

# Acrónimos

ARMAX	<i>Auto Regressive Moving Average with eXternal input</i> Auto Regressivo de média móvel com entrada externa.
ARX	<i>Auto Regressive with eXternal Input</i> Auto Regressivo com entrada externa.
BIBO	<i>Bounded Input Bounded Output</i> Entrada limitada saída limitada.
BP	<i>BackPropagation</i> Retropropagação.
BPTT	<i>BackPropagation Trough Time</i> Retropropagação temporal.
CM	<i>Center Manifold</i>
CMT	<i>Center Manifold Theory</i>
DEKF	<i>Dual Extended Kalman Filter</i> Filtro de Kalman dual estendido.
DKF	<i>Dual Kalman Filter</i> Filtro de Kalman dual.
EKF	<i>Extended Kalman Filter</i> Filtro de Kalman estendido.
FL	<i>Feedback Linearization</i> Linearização Retroactiva.
FNN	<i>Feedforward Neural Network</i> Rede com ligações para a frente.
IMC	<i>Internal Model Control</i> Controlador com modelo interno.
I-O	<i>Input - Output</i> Entrada - saída.
KF	<i>Kalman Filter</i> Filtro de Kalman.
LMS	<i>Least Mean Squares</i> Média dos mínimos quadráticos.

LS	<i>Least Squares</i> Mínimos quadráticos.
MC	<i>Matching Conditions</i> Condições de verificação
MIMO	<i>Multiple Input Multiple Output</i> Múltiplas entradas várias saídas.
MISO	<i>Multiple Input Single Output</i> Múltiplas entradas uma saída.
MLNN	<i>Multi Layer Neural Network</i> Rede neuronal de várias camadas.
MRAC	<i>MRAC-Model Reference Adaptive Control</i> Controlador Adaptativo por Modelo de Referência.
NARMAX	<i>Nonlinear AutoRegressive Moving Average with eXternal input</i> Não linear auto Regressivo de média móvel com entrada externa.
OR	<i>Output Regulation</i> Regulação da Saída.
PI	<i>Proportional + Integral</i> Proporcional + Integral.
PID	<i>Proportional + Integral + Derivative</i> Proporcional + Integral + Derivativo.
RBFNN	<i>Radial Basis Function Neural Network</i> Rede neuronal de função de base radial.
RLS	<i>Recursive Least Squares</i> Mínimos quadráticos recursivos.
RNN	<i>Recurrent Neural Network</i> Rede neuronal recorrente.
RSRN	<i>Output Regulation with Recurrent Neural Networks</i> Regulação da saída com Redes Neurais Recorrentes
RTRL	<i>Real Time Recurrent Learning</i> Aprendizagem recorrente em tempo real.
SC	<i>Soft Computing</i> Computação Adaptativa.
SISO	<i>Single Input Single Output</i> Uma entrada uma saída.
SMO	<i>Sliding Mode Observer</i> Observador de modo deslizante.
SSCC	<i>Sum of Squared Change of Control</i> Soma do valor quadrático da variação da acção de controlo.

<i>SSCE</i>	<i>Sum of Squared Control Error</i> Soma do valor quadrático do erro de controlo (diferença entre a saída desejada e a saída real).
<i>SSE, SSIE</i>	<i>Sum of Squared (Identification) Error</i> Soma do valor quadrático do erro (de identificação).
<i>STAC</i>	<i>Self Tuning Adaptive Control</i> Controlador Adaptativo Auto Ajustável.
<i>TBPTT</i>	<i>Truncated BackPropagation Trough Time</i> Retropropagação temporal truncada.
<i>TDLNN</i>	<i>Time Delay Neural Network</i> Rede neuronal com entradas de instantes passados.
<i>ZD</i>	<i>Zero Dynamics</i>

# 1.

# Introdução

*“It is difficult to find more fashionable and attractive terms in modern theory of automatic control than the terms of adaptive and learning. At the same time, it is not simple to find any other concepts, which are less complex and more vague.”*

Y. Tsylin, 1971.

O projecto de sistemas de controlo tem sido, é, e certamente continuará a ser uma área fascinante. Desde as suas origens até à actualidade, impulsionada quer por novos conceitos teóricos, quer pela necessidade de satisfação das exigências cada vez maiores da indústria, quer pelo desenvolvimento de novas tecnologias computacionais, o projecto de sistemas de controlo tem evoluído de uma forma impressionante.

Ainda há pouco mais de três décadas surgia o termo controlo moderno, para designar técnicas de controlo que usam modelos de estados, em contraste com as técnicas clássicas, introduzidas nos anos 30 e 40, baseadas em funções de transferência. Qualquer delas, referidas hoje em dia como convencionais ou clássicas, proporciona uma teoria sólida, bem estabelecida, para a análise e projecto de controladores no caso de sistemas lineares e invariantes no tempo. E, se estas técnicas têm alcançado sucesso num número elevado de aplicações práticas, a crescente complexidade dos processos trouxe consigo novos desafios e levou à necessidade de desenvolvimento de outras metodologias.

Por outro lado, a aplicação de técnicas e metodologias de *Soft Computing* (computação adaptativa) de que são exemplos as redes neuronais, a lógica difusa e os algoritmos genéticos, têm mostrado potencialidades para resolver situações práticas de controlo, para as quais as metodologias convencionais não se têm mostrado tão adequadas. Em particular as redes neuronais, pelas suas capacidades em captar a

dinâmica de sistemas, e pela possibilidade de se adaptarem continuamente a variações no ambiente, têm-se mostrado eficientes na identificação e controlo de sistemas complexos.

Esta dissertação deve pois, antes de mais, ser entendida como uma contribuição para o desenvolvimento de sistemas de controlo híbridos, recorrendo às sinergias entre técnicas de controlo convencional e metodologias de *soft computing*, inseridas em estruturas de controlo adaptativo. Reconhece-se que estas novas metodologias não devem ser vistas apenas como uma alternativa, mas sim como um complemento, às técnicas convencionais existentes.

Pretende-se, em última análise, desenvolver um sistema de controlo genérico, com potencialidades de aplicação prática, que motive a indústria na implementação de estratégias semelhantes à apresentada.

# 1.1

## Motivações

### 1.1.1 Técnicas Convencionais de Controlo

A teoria de controlo convencional é, regra geral, bem sucedida no caso em que o processo a controlar pode ser razoavelmente descrito por um modelo linear. Quando assim é, partindo de uma descrição do processo quer recorrendo a funções de transferência, quer recorrendo a modelos de estados, estas técnicas proporcionam uma teoria bem estabelecida para o projecto de controladores, assim como métodos de análise de propriedades fundamentais tais como a estabilidade, a observabilidade ou a controlabilidade para o sistema em questão.

No entanto, na maioria das aplicações práticas a realidade é outra. De facto, grande parte dos processos industriais são difíceis de caracterizar, apresentam uma dinâmica não linear, estão sujeitos a variações, a incertezas e perturbações, o que dificulta na prática a aplicação de tais técnicas. Assim, quando se pensa em desenvolver um modelo que descreva o comportamento de um processo, este terá necessariamente de possuir um elevado grau de complexidade, não se mostrando adequado quer a representação por função de transferência, quer por modelos de estados. Por outro lado, quando se pensa em efectuar o projecto de um controlador requer-se que o modelo a utilizar seja simples, de preferência linear e de parâmetros fixos.

Uma solução de compromisso consiste na implementação de uma estrutura adaptativa. Neste caso, assume-se um modelo de estrutura fixa (geralmente linear) sendo, no entanto, os seus parâmetros variáveis. Seguindo esta abordagem os parâmetros do sistema são actualizados em cada instante, por um processo de identificação, e os parâmetros do controlador são por sua vez adaptados em função destes últimos permitindo, pelo menos teoricamente, assegurar um bom desempenho do sistema de controlo para várias condições de operação do processo.

Embora os sistemas de controlo adaptativo convencionais tenham sido introduzidos com o fim de lidar com sistemas complexos, a sua implementação prática tem-se mostrado pouco robusta, principalmente em ambientes cujo grau de incerteza é elevado. De facto, as metodologias adaptativas, baseadas em modelos lineares, têm à partida algumas limitações. Uma vez que se considera uma estrutura linear e fixa, as incertezas resumem-se à determinação dos parâmetros do modelo e, sendo assim, mesmo

recorrendo à adaptação de parâmetros, os modelos lineares não são suficientemente flexíveis para representar sistemas não lineares genéricos.

A par das técnicas adaptativas têm sido desenvolvidas, especificamente para processos não lineares, técnicas de controlo que incorporam directamente no seu projecto o conhecimento das não linearidades, resultando os controladores também eles não lineares. Em particular, é objectivo da técnica de linearização retroactiva (*feedback linearization*) transformar, partindo do conhecimento rigoroso do sistema e através de uma realimentação de variáveis de estado, um sistema não linear num correspondente linear, possível de controlar usando uma qualquer técnica linear.

No entanto, se do ponto de vista teórico a teoria de controlo não linear é notável, permitindo efectuar um cancelamento perfeito das não linearidades, por outro lado, sob o ponto de vista prático, pela sua forte dependência quer dos parâmetros do modelo, quer da descrição do sistema, quer ainda da necessidade da acessibilidade dos estados do processo, estas técnicas sofrem de sérios problemas de implementação prática e de robustez.

Do exposto, para sistemas não lineares e complexos, o projecto de um controlador recorrendo a técnicas convencionais transforma-se numa tarefa que não pode ser considerada de forma alguma trivial. Por um lado, se se considera uma estrutura linear, mesmo recorrendo à adaptação de parâmetros, não existem formalismos genéricos, de forma a que a teoria de controlo adaptativo linear possa ser facilmente estendida a sistemas não lineares. Por outro, se forem usadas técnicas não lineares, então cada caso é um caso, dependendo da não linearidade particular do problema em questão, não existindo além disso uma metodologia de controlo que permita lidar com sistemas não lineares de uma forma genérica, tal como existe para sistemas lineares.

### 1.1.2 Técnicas de *Soft Computing*

O conceito *soft computing*, introduzido por Zadeh, consiste na associação de metodologias de computação baseadas na lógica difusa, redes neuronais e algoritmos genéticos. Um dos seus objectivos é proporcionar uma base para concepção, projecto e aplicação de sistemas capazes de lidar com incertezas e imprecisões, utilizando cada uma destas metodologias isoladamente ou em conjunto. Quando combinadas originam sistemas híbridos, de que são exemplos os sistemas neuro-difusos.

Uma vez que estas técnicas se baseiam em métodos numéricos, Bezdek propôs o termo *Computational Intelligence* (inteligência computacional), para distinguir estas técnicas, quando aplicadas à modelização de *sistemas inteligentes*, das técnicas de inteligência artificial, baseadas em métodos simbólicos.

As metodologias de *soft computing* aplicadas a problemas de controlo desde cedo mostraram elevado potencial e o seu sucesso tem sido inquestionável na solução de problemas práticos que os controladores convencionais, mesmo adaptativos, têm tido dificuldades, especialmente na presença de complexidades, não linearidades e incertezas. Assim, durante o passado recente, tem havido um interesse especial em compreender as capacidades destas técnicas e, em particular, como podem ser incorporadas em esquemas

de controlo, originando uma nova área de investigação, designada por *Controlo Inteligente*, por oposição às técnicas convencionais.

Os sistemas difusos facultam um meio de converter o conhecimento linguístico, baseado, por exemplo, no conhecimento de um operador, numa estratégia de controlo, implementada sob a forma de regras difusas. Por esta razão são particularmente adequados a sistemas complexos, cuja análise e projecto são difíceis seguindo abordagens convencionais quantitativas, mas que na prática são convenientemente controlados se for usada uma abordagem qualitativa.

As redes neuronais são constituídas por elementos de processamento elementar (os neurónios) associados entre si por meio de ligações ponderadas (pesos ou parâmetros). Cada neurónio caracteriza-se por possuir uma estrutura relativamente simples, fazendo uso de não linearidades bem definidas e tratáveis matematicamente.

Aplicadas ao controlo, as redes neuronais apresentam-se como uma ferramenta computacional que permite, pelas suas características, ultrapassar algumas das dificuldades dos métodos convencionais. As suas propriedades de aproximação universal, de generalização, e a sua aptidão em ajustar permanentemente os seus parâmetros a partir de dados de operação, têm-lhes permitido responder a dois dos maiores desafios para os quais as técnicas convencionais não têm sido capazes de dar resposta satisfatória: rigor e generalidade na modelização e capacidade de adaptação, de forma a lidar com variações na dinâmica. Estas propriedades mostram-se particularmente significativas em sistemas complexos, não lineares e variantes no tempo.

Apesar do sucesso indiscutível em solucionar problemas práticos, as técnicas de *soft computing* caracterizaram-se, numa fase inicial, pela carência de resultados teóricos sendo a sua especificação efectuada na maioria das vezes, através de técnicas empíricas. Por esta razão não foram apresentados resultados bem consolidados de estabilidade e convergência, tal como existem para as técnicas convencionais.

Actualmente, e apesar dos avanços entretanto efectuados nos últimos anos, a investigação relacionada com o projecto de controladores usando técnicas de *soft computing*, encontra-se ainda numa fase de maturação, de compreensão das suas capacidades e na forma de as inserir eficazmente em estruturas de controlo.

### 1.1.3 Incorporação de Técnicas de *Soft Computing* em Estruturas de Controlo

Do que foi exposto, tornam-se claros os motivos que conduziram à elaboração desta dissertação. De um lado, as técnicas ditas convencionais, baseadas em teorias bem estabelecidas, principalmente úteis no projecto de controladores e análise de propriedades fundamentais tais como estabilidade e convergência. De outro, as técnicas de *soft computing* e as capacidades práticas demonstradas na resolução de problemas de controlo não lineares e complexos. Da intersecção das duas técnicas, entendidas não como incompatíveis mas, antes pelo contrário, complementares, surgiu o ponto de partida para esta dissertação: o desenvolvimento de estruturas híbridas, que combinem numa mesma

estrutura metodologias de *soft computing* e convencionais, de forma a tirar partido das potencialidades de cada uma delas.

Pretende-se, em última análise, investigar e desenvolver estruturas de controlo que incorporem metodologias de *soft computing* mas que sejam acima de tudo analisadas segundo técnicas bem estabelecidas da teoria convencional.

Se a motivação era clara no que respeita ao seu aspecto geral, a opção a tomar quer pela estrutura global de controlo, quer por cada uma das metodologias específicas a cada uma das técnicas (*soft computing* e convencional) não foi naturalmente possível de efectuar à partida. Por este motivo foram conduzidos, ao longo da elaboração desta dissertação, vários trabalhos de investigação quer teóricos quer experimentais que justificassem as opções entretanto tomadas (alguns dos quais se apresentam em apêndice).

Nesta dissertação descreve-se em pormenor a metodologia considerada como mais relevante, uma metodologia de controlo adaptativa indirecta, que combina as potencialidades de modelização de uma rede neuronal recorrente com técnicas bem estabelecidas de controlo não linear. Apresentam-se seguidamente as motivações que estiveram na base da opção tomada.

#### 1.1.4 Redes Neurais Recorrentes em Estruturas de Controlo

Sabe-se que a introdução de dinâmica em modelos neuronais pode ser conseguida de várias formas. De entre estas, sem dúvida que uma das mais conhecidas é a estrutura com recorrência externa (*time delay neural network*), que consiste em relacionar a saída actual em função de valores passados de entradas e de saídas do processo, considerados como entradas na rede. Neste tipo de redes, a recorrência é externa e a variável tempo é introduzida como uma transformação do tempo em espaço, feita portanto de uma forma explícita ao nível da arquitectura.

As redes recorrentes, propriamente ditas, são conceptualmente diferentes uma vez que a variável tempo é incluída de uma forma implícita. Assim, reconhecem-se vantagens adicionais, principalmente pela sua capacidade em incorporar de uma forma inerente dinâmica, o que lhes permite serem utilizadas com naturalidade na modelização de sistemas.

Por outro lado reconhecem-se as vantagens das técnicas de controlo convencional que fazem uso de modelos de estados, de que é exemplo a realimentação de variáveis de estado (linear). No entanto, na prática a totalidade dos estados é quase sempre inacessível, o que dificulta a aplicação prática deste tipo de técnicas. Este problema pode, em parte, ser ultrapassado recorrendo ao projecto de observadores, que permitem determinar uma estimativa para os estados do processo, usando apenas dados de entrada e de saída.

Pensa-se que é neste ponto que as técnicas neuronais e convencionais se tocam, e as vantagens de cada uma delas podem ser combinadas. Na verdade, pelas suas características, as redes neuronais recorrentes possuem uma estrutura que pode ser

interpretada como um modelo de estados não linear, constituindo um formalismo base para ultrapassar o problema da estimação de estados e de parâmetros. Ou seja, entendidas neste contexto, as redes recorrentes não são senão modelos de estados não lineares com capacidades de aproximação universal e de generalização e caracterizadas por não linearidades bem definidas e tratáveis, ideais, portanto, para aplicação de técnicas de controlo não linear.

Na Figura 1.1-1 mostra-se a estrutura que motivou o trabalho realizado nesta dissertação.

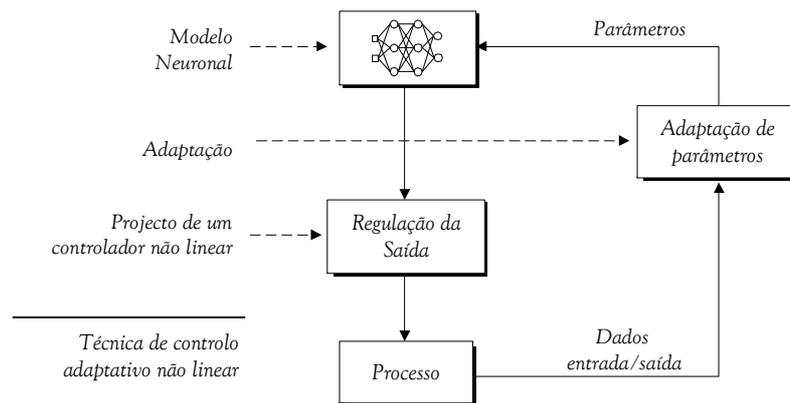


Figura 1.1-1: Estrutura considerada.

Seguindo esta concepção refere-se, em primeiro lugar, o papel desempenhado pelos modelos neuronais como metodologias genéricas de modelização de sistemas dinâmicos não lineares, cujos parâmetros podem ser adaptados recorrendo a dados de entrada-saída do processo.

Em segundo lugar, interpretando a rede como um modelo de estados não linear, é possível usar técnicas bem estabelecidas que incluem a análise de estabilidade e convergência para a adaptação dos parâmetros da rede, nomeadamente a partir da teoria de observação de sistemas não lineares e da teoria de estabilidade de Lyapunov.

Em terceiro lugar, a motivação no projecto de controladores baseados na teoria de controlo não linear, concretamente na teoria de regulação da saída (*output regulation*), pelas possibilidades que proporcionam em lidar directamente com as não linearidades do modelo neuronal.

Apesar do sucesso e dos avanços efectuados nos últimos anos existem ainda algumas limitações que não permitem uma maior exploração por parte da indústria de todo o potencial que as redes neuronais podem proporcionar, quando aplicadas em sistemas de controlo. Assim, motiva-se a elaboração deste trabalho, como uma contribuição para que estes problemas possam vir a ser ultrapassados, desenvolvendo para o efeito sistemas de controlo que permitam responder às reais necessidades da indústria, em particular para processos não lineares, variantes e sujeitos a incertezas.

É importante referir que a motivação para a elaboração de uma dissertação nesta área específica surgiu naturalmente da inspiração de vários trabalhos, dos quais não é possível deixar de referir:

- i) o trabalho de Zbikowski (1994), pelas ideias que introduziu acerca da possibilidade das redes neuronais serem utilizadas na formulação de um problema de controlo genérico, facto que sem dúvida, constitui uma forte motivação para a investigação nesta área;
- ii) o trabalho de Elman (1990), pelo impulso que deu às redes neuronais recorrentes e pela interpretação destas como modelos de estado não lineares;
- iii) os trabalhos de Rovithakis e Christodoulou (1994), (1995), (1996), (1997) e de Poznyak e colaboradores, Poznyak *et al.* (1999), Yu e Poznyak (1999a), Yu *et al.* (1999b), relativos a algoritmos de adaptação aplicados a modelos de estado neuronais, baseados fundamentalmente na teoria de estabilidade de Lyapunov;
- iv) o trabalho de Thau (1973) relativo a técnicas de observação de sistemas não lineares e estudos de estabilidade;
- v) os trabalhos de Jin, Nikiforuk e Gupta, Jin *et al.* (1993), (1994a), (1995a), (1995b), (1999a), (1999b), pela diversidade dos pontos que focam, arquitecturas, algoritmos de treino, análise de estabilidade, entre outros, que proporcionaram uma visão enriquecedora das capacidades das redes aplicadas, tanto à modelização, como ao controlo;
- vi) os trabalhos de Isidori e Byrnes (1990), (Isidori, 1996), relativo à teoria de controlo não linear, e em particular à teoria de regulação, que permite tratar de uma forma eficiente o problema de controlo de sistemas não lineares, tendo em conta aspectos de estabilidade.

Em conclusão, as redes neuronais permitem ultrapassar algumas das limitações existentes nas técnicas convencionais, possuindo propriedades que motivam a sua utilização, principalmente em tarefas de modelização. No entanto, seria inconcebível ignorar os resultados alcançados na teoria de controlo convencional e pensar nas redes neuronais apenas como uma alternativa a estas técnicas e não como um complemento.

Neste sentido, as redes neuronais apresentam-se capazes de estabelecer uma ponte adequada entre as teorias convencionais e de *soft computing*, contribuindo para o desenvolvimento de teorias unificadoras.

# 1.2

## Objectivos

Qualquer trabalho realizado na área de controlo automático, antes de tudo, deve ter como objectivo final o desenvolvimento de estratégias de controlo que possam ser capazes de responder às necessidades e aos desafios da indústria. O trabalho realizado nesta dissertação deve ser visto como um meio para alcançar esse fim. Ou seja, estabelece-se como objectivo principal o desenvolvimento de uma metodologia genérica, que use as potencialidades das técnicas de *soft computing* e ao mesmo tempo seja baseada em métodos convencionais, capaz de oferecer uma solução passível de ser aplicada em ambientes industriais.

Porque na maioria dos processos reais a informação disponível consiste em dados de entrada e de saída, considera-se que na metodologia a desenvolver a determinação da acção de controlo deverá ser obtida apenas a partir deste tipo de informação. Assume-se ainda como válida a capacidade de aproximação dos modelos neuronais, ou seja, aceita-se que se uma rede neuronal descrever convenientemente a dinâmica entrada-saída do processo, então será possível alcançar um bom desempenho com um controlador que seja projectado com base no modelo neuronal obtido. Assim, o sistema a controlar pode ser substituído pelo modelo neuronal, de estrutura fixa mas de parâmetros variáveis, transformando o problema original de controlo em dois problemas parciais: um problema de identificação, no qual deverão além da especificação da estrutura serem estimados os parâmetros da rede e o de projecto de um controlador não linear, a efectuar com base no modelo neuronal.

Contudo, porque num sistema real existem sempre efeitos imprevisíveis, perturbações, incertezas, assumir-se que o modelo neuronal desenvolvido representa com rigor o processo a ser controlado, não é viável em termos práticos. Assim sendo, o projecto de um controlador baseado no modelo neuronal, conduzirá inevitavelmente a resultados indesejados, nomeadamente à existência de erros de controlo em regime final. A incorporação de uma estratégia adaptativa surge como uma forma de ultrapassar esta limitação.

Como resultado das ideias expostas surge uma estrutura de controlo adaptativo, mais precisamente uma estrutura de controlo adaptativo indirecto ou auto ajustável (*self tuning*), em que os parâmetros do controlador são obtidos indirectamente a partir dos parâmetros do modelo neuronal, como se mostra na Figura 1.2-1.

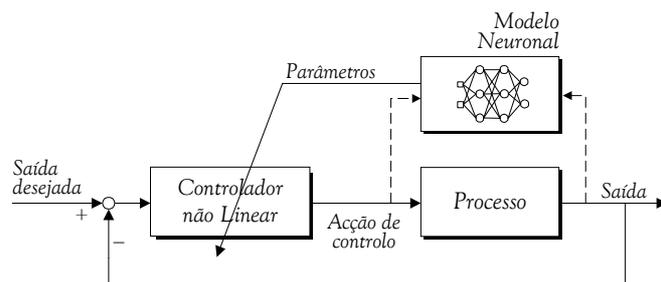


Figura 1.2-1: Estrutura adaptativa indirecta.

Assim estabelecido, o projecto de um sistema de controlo pode ser subdividido nas seguintes etapas parciais: modelização neuronal, adaptação do modelo neuronal e projecto de um controlador não linear.

### 1.2.1 Modelização

Como primeiro objectivo estabelece-se a utilização de redes neuronais, e em particular as redes recorrentes pela suas características dinâmicas, na modelização do processo a controlar. Resultará daqui um modelo de estados do tipo *black-box*, assumindo-se portanto o desconhecimento do modelo físico do processo. Por outras palavras, pretende-se nesta dissertação abordar o problema da modelização tal como um problema de aproximação usando para o efeito um aproximador universal, neste caso particular uma rede neuronal recorrente.

### 1.2.2 Adaptação

Como segundo objectivo, pela sua interpretabilidade como modelos de estados, pretendem-se desenvolver algoritmos de adaptação, baseados em técnicas rigorosas (nomeadamente de Lyapunov), que assegurem as propriedades fundamentais de estabilidade e convergência. Em concreto, pretende-se levar a cabo o projecto de um observador dinâmico, de forma a adaptar simultaneamente os estados e os parâmetros do modelo neuronal, seguindo uma estratégia análoga à de um filtro de Kalman dual.

### 1.2.3 Controlo

Em terceiro lugar, e relativamente à estratégia de controlo, é objectivo fazer uso de uma técnica de controlo não linear, a ser projectada com base no modelo paramétrico neuronal desenvolvido. Em particular, pelas propriedades de estabilidade que proporciona, tenciona-se investigar a possibilidade do uso da teoria de regulação da saída.

Pretende-se ainda tirar partido da especificidade da não linearidade presente nas redes neuronais (tangentes hiperbólicas, portanto funções não lineares do tipo *Lipschitz*) e a partir dessa especificidade desenvolver algoritmos de controlo particulares para a estrutura considerada.

### 1.2.4 Aplicação Prática

Conforme referido anteriormente, o objectivo de qualquer trabalho deste tipo deverá, de preferência, passar pela sua validação prática num processo industrial.

Na impossibilidade de tal ter acontecido, estabelece-se como objectivo a validação das metodologias desenvolvidas quer em processos laboratoriais, quer em processos de simulação. Pretende-se ainda efectuar estudos comparativos com metodologias usualmente utilizadas na indústria, de que são exemplo os controladores *PI* (*proporcional + integral*).

Refira-se que foram efectuados estudos de investigação experimentais, para uma das metodologias desenvolvidas, num processo à escala industrial (Plataforma Solar de Almería). Concretamente, implementou-se um sistema de supervisão difuso, baseado em controladores sintonizados por redes neuronais, a um sistema de colectores solares inserido numa plataforma solar. Espera-se, assim que seja possível, vir a aplicar a esta plataforma solar a metodologia neuronal desenvolvida nesta dissertação.

# 1.3

## Contribuições

Como primeira contribuição desta dissertação refere-se a arquitectura *RSRN-Regulação da Saída com Redes Neurais Recorrentes*, aplicada ao projecto de um controlador adaptativo que reúne as vantagens das técnicas de controlo não linear e as potencialidades de modelização das redes neuronais. A interpretação destas últimas como observadores permite desempenhar um papel unificador das teorias de controlo não linear com as técnicas neuronais.

Como segunda contribuição refere-se, de uma forma genérica, o estudo e desenvolvimento de várias estruturas de controlo híbridas, que combinam metodologias de *soft computing* e metodologias convencionais.

Neste contexto, foram desenvolvidos vários trabalhos de interesse quer pela sua contribuição genérica na investigação de estruturas híbridas, quer pela sua contribuição particular que levou ao desenvolvimento da estratégia de controlo proposta nesta dissertação.

Em concreto, foram feitos estudos experimentais de comparação entre metodologias que incorporam modelos difusos, modelos neuronais e mesmo a sua combinação. Dois destes trabalhos são apresentados nos apêndices B.1 e B.2. Destaca-se o segundo pela aplicação prática a um campo de painéis solares de escala industrial, situado na Plataforma Solar de Almería, sul de Espanha, um centro de investigação aplicada de técnicas para aproveitamento de energia solar.

Foram ainda efectuados estudos teóricos e estudos experimentais no que se refere às capacidades das redes neuronais recorrentes, em particular à possibilidade da sua aplicação em-linha (*on-line*) na identificação de processos de dinâmica variável (apêndice B.3), contribuindo desta forma para a compreensão das suas potencialidades.

Para o caso particular de aplicação de modelos neuronais recorrentes em estruturas de controlo, referem-se contribuições realizadas na investigação da sua combinação com estratégias de controlo linear (apêndice B.4) e com técnicas de controlo preditivo (apêndice B.5).

Esta dissertação introduz algumas contribuições, nas seguintes áreas: modelização, projecto de controladores e estabilidade do sistema de controlo.

### 1.3.1 Modelização

Relativamente às estruturas neuronais, inseridas em esquemas de controlo, salientam-se os seguintes aspectos:

- i) Interpretação da rede recorrente como um modelo constituído por uma parte linear e outra não linear. Daqui resulta a possibilidade de introdução de conhecimento inicial, sob a forma de um modelo de estado linear, no modelo neuronal;
- ii) Interpretação das redes como modelos de estados não lineares, o que possibilita a determinação dos seus parâmetros com base numa estrutura análoga à de um filtro de Kalman dual.

Relativamente à adaptação de parâmetros e de estados:

- i) Propõe-se um algoritmo de adaptação dos parâmetros do modelo neuronal, baseado na teoria de estabilidade de Lyapunov, capaz de assegurar propriedades de estabilidade e convergência;
- ii) Propõe-se um algoritmo de estimação dos estados da rede, baseado na teoria de observação para sistemas não lineares, para o qual são estabelecidas condições de estabilidade.

### 1.3.2 Controlo

Em relação ao controlo de sistemas não lineares referem-se as seguintes contribuições:

- i) É proposta uma classificação de sistemas de controlo que utilizem na sua estrutura redes neuronais;
- ii) A teoria de regulação da saída, uma das técnicas de controlo não linear, permite projectar um controlador de tal forma que o sistema em malha fechada seja estável e simultaneamente o erro de controlo convirja para zero. No entanto, a sua solução implica a necessidade de resolução de um sistema não linear de equações de diferenças, conhecido por equações de regulação. Uma das contribuições principais desta dissertação, consiste no desenvolvimento de um algoritmo capaz de fornecer uma solução para o sistema de equações. Em concreto, propõe-se um algoritmo iterativo, específico para a arquitectura neuronal proposta e para o tipo de não linearidade em questão (funções *Lipschitz*), capaz de assegurar a convergência das equações de regulação.

### 1.3.3 Estabilidade do Sistema de Controlo

Outra contribuição prende-se com o esforço realizado no sentido de estabelecer condições de estabilidade para sistemas de controlo que utilizem redes neuronais recorrentes, no domínio discreto. Nesse sentido são propostos três teoremas, dois deles

relativos à adaptação de parâmetros e estados e outro relativo ao projecto do controlador. Garantem-se assim condições que asseguram a estabilidade do sistema de controlo, considerando como independentes cada um dos procedimentos de identificação e de controlo (princípio de separação).

- i) *Teorema 1: Estimação de estados para a estrutura RSRN*, neste teorema são estendidos conceitos de observação aplicados a sistemas contínuos não lineares ao caso particular do modelo não linear neuronal proposto (no domínio discreto);
- ii) *Teorema 2: Adaptação de parâmetros na estrutura RSRN*, a partir da estabilidade de Lyapunov e do princípio de invariância de La Salle é proposto um método de adaptação dos parâmetros do modelo neuronal, capaz de garantir a estabilidade e convergência do erro de estimação;
- iii) *Teorema 3: Convergência do algoritmo RSRN*, usando a especificidade das não linearidades presentes no modelo utilizado (tangente hiperbólica) é proposto um método iterativo capaz de solucionar o problema de regulação e garantir a sua convergência.

### 1.3.4 Aplicação Prática

Finalmente apresenta-se, para diferentes tipos de processos, um conjunto de resultados, pois os processos em causa apresentam características, em certa medida, análogas às encontradas em processos industriais. Justifica-se assim o seu interesse, tendo em vista a implementação futura da estratégia de controlo em ambientes industriais.

# 1.4

## Estrutura

Incluindo o presente, esta dissertação é composta por seis capítulos: 2- Redes Neurais Aplicadas à Modelização, 3- Redes Neurais Aplicadas ao Controlo, 4- Regulação da Saída com Redes Neurais Recorrentes, 5- Casos de Estudo e 6- Conclusão e Perspectivas.

De forma a não prejudicar a leitura da dissertação, mas uma vez que a sua realização foi decisiva, optou-se por apresentar em apêndice alguns trabalhos efectuados. Assim, apresentam-se dois apêndices: Apêndice A, relativo à teoria de estabilidade de Lyapunov e normas de vectores e matrizes; Apêndice B, onde são apresentados os trabalhos acima referidos.

### *Capítulo 2*

#### Redes Neurais Aplicadas à Modelização

Neste capítulo formula-se o problema de modelização de uma forma genérica e justifica-se a opção pelo uso de modelos neuronais. Discutem-se aspectos relativos à selecção da sua estrutura, à estimação dos seus parâmetros e introduzem-se as arquitecturas recorrentes adequadas à modelização de sistemas dinâmicos.

Pela sua interpretabilidade como modelos de estados, aborda-se o problema de estimação dos seus parâmetros, tal como um problema de observação, à semelhança de um filtro de Kalman dual.

### *Capítulo 3*

#### Redes Neurais Aplicadas ao Controlo

Este capítulo centra-se no problema de controlo neuronal, isto é, em técnicas de controlo que utilizam de alguma forma redes neuronais na sua estrutura, referindo-se como um dos aspectos principais a proposta de uma classificação de sistemas de controlo neuronal.

Começa-se por proporcionar uma visão genérica de metodologias de controlo adaptativo e de que forma as redes neuronais podem ser incorporadas nestas estruturas. Revêem-se e classificam-se os vários tipos de metodologias de controlo neuronal e

enquadram-se alguns dos trabalhos realizados ao longo da elaboração desta dissertação. Introduce-se a estrutura final proposta, uma estrutura híbrida que combina redes neuronais com técnicas de controlo não linear. Termina-se revendo trabalhos análogos, que combinam técnicas de controlo não linear com modelos neuronais em estruturas indirectas.

## Capítulo 4

### Regulação da Saída com Redes Neuronais Recorrentes

Neste capítulo justifica-se a estrutura para o modelo neuronal proposto nesta dissertação e considera-se o problema de estimação dos seus parâmetros. Recorrendo a técnicas de observação não lineares e à teoria de Lyapunov estabelecem-se condições para a estabilidade e convergência do algoritmo de adaptação, uma das contribuições principais desta dissertação.

Usando o conceito de controlador baseado num modelo, e de entre as metodologias de controlo não linear, utiliza-se a teoria da regulação da saída e propõe-se a sua aplicação ao modelo neuronal. Adequa-se a teoria de regulação à rede neuronal recorrente proposta, desenvolvendo, para o efeito, um algoritmo baseado na especificidade das não linearidades nela presentes, para o qual são estabelecidas condições de convergência.

## Capítulo 5

### Casos de estudo

Neste capítulo apresentam-se três casos de estudo de aplicação da metodologia de controlo proposta, de forma a avaliar a sua eficácia em situações concretas. Em qualquer deles compara-se a estratégia proposta com técnicas convencionais bem conhecidas, concretamente com controladores *PI* nos primeiros e segundos casos, e realimentação de variáveis de estado, para o terceiro caso.

O primeiro caso de estudo é uma aplicação experimental a um processo térmico. O objectivo principal é investigar aspectos de inicialização de parâmetros relativos aos procedimentos de identificação e de controlo e simultaneamente avaliar o desempenho do controlador em presença de variações na dinâmica do processo.

O segundo é um estudo de simulação em que é considerado o problema do controlo da concentração de um componente num reactor químico. O objectivo principal é avaliar o desempenho do sistema de controlo quando o processo a controlar é fortemente não linear.

No terceiro caso, novamente uma aplicação experimental, está em causa o controlo dos níveis de um processo constituído por três tanques acoplados. O objectivo é investigar a capacidade da metodologia proposta quando aplicada a processos multi-variáveis e comparar o seu desempenho com metodologias de controlo baseadas em modelos de primeiros princípios que assumem a acessibilidade dos estados.

## Capítulo 6

### Conclusão e Perspectivas

Tecem-se finalmente no capítulo 6 algumas conclusões gerais e apontam-se linhas de orientação para a continuidade futura deste trabalho.

## Apêndice A

### Teoria de Lyapunov e Normas de Matrizes

Apresentam-se neste apêndice, para além de teorias relativas à estabilidade de Lyapunov e princípio de invariância de La Salle, resultados relativos a normas de vectores e matrizes e suas propriedades fundamentais.

## Apêndice B

### Alguns Trabalhos Realizados

Neste apêndice são apresentados cinco dos trabalhos realizados durante a elaboração desta dissertação:

#### B.1

Henriques, J., J. Vitor, C. Pereira e A. Dourado  
*Experimental on-line Learning for a Benchmark Process*,  
CONTROLO98, 3<sup>rd</sup> Portuguese Conference on Automatic Control, 193-198,  
Setembro 1998, Coimbra, Portugal.

#### B.2

Henriques, J., A. Cardoso e A. Dourado  
*Supervision and C-Means Clustering of PID Controllers for a Solar Power Plant*,  
International Journal of Approximate Reasoning, 22, 73-91, 1999.

#### B.3

Henriques, J., P. Gil, A. Dourado e H. Duarte-Ramos  
*Application of a Recurrent Neural Network in On-Line Modelling of Real-Time Systems*,  
ESIT99, European Symposium on Intelligent Techniques,  
Junho 1999, Crete, Grecia.

#### B.4

Henriques, J., A. Dourado  
*Multivariable Adaptive Control Using an Observer Based on a Recurrent Neural Network*,  
International Journal of Adaptive Control and Signal Processing, 13, 241-259, 1999.

#### B.5

Henriques, J., P. Gil, A. Dourado e H. Duarte-Ramos  
*Non-linear Multivariable Predictive Control: Neural Versus First Principle Modelling Approach*,  
Submetido para publicação na revista Journal of Systems and Control Engineering, 1999.

# 1.5

## Convenções e Meios

No que se refere a terminologias estrangeiras optou-se por efectuar a sua tradução nos casos em que a terminologia é correntemente aceite. Nos casos em que surgiram dúvidas, optou-se por manter a denominação original (em itálico).

Os meios computacionais para a realização deste trabalho foram um *Pentium* a 100 MHz com o sistema operativo *Windows 98*<sup>1</sup>.

Os programas de suporte utilizados foram os seguintes:

- i) o texto foi escrito usando o processador *Microsoft Word 2000*<sup>2</sup>.
- ii) os diagramas foram produzidos no programa *Visio Technical*<sup>3</sup>.
- iii) as simulações foram efectuadas usando o *MatLab*<sup>4</sup>.

Nas experiências realizadas, foram utilizados os processos laboratoriais *PT326* e *DTS200*, assim com as respectivas placas de aquisição de dados, existentes no *LACI – Laboratório de Automação e Controlo Industrial* do Departamento de Engenharia Informática da Universidade de Coimbra.

Para a implementação prática usou-se a linguagem C, e em concreto o compilador *Watcom C/C++*<sup>5</sup>.

---

<sup>1</sup> Copyright ©1998- Microsoft Corporation

<sup>2</sup> Copyright ©2000- Microsoft Corporation

<sup>3</sup> Copyright ©1997- Visio Professional, Version 5.0

<sup>4</sup> Copyright ©1998- The MathWorks Inc., Version 5.2

<sup>5</sup> Copyright ©1995- Watcom International Corp., Version 10.5

# 2.

## Redes Neuronais

### Aplicadas à Modelização

*“Massively parallel arrays of simple processing elements, the so-called neural network models, can be used to greatly enhance and extend techniques for identification and control of complex, non-linear dynamic systems.”*

J. Slotine, 1996.

O paradigma básico da teoria de controlo é definido por um processo a controlar,  $P$ , e um controlador,  $C$ , interligados entre si, Figura 2.0-1. O controlador usa medidas do processo e, com base nelas, calcula valores de actuação sobre o processo, de forma a alcançar determinados objectivos,  $J$ .

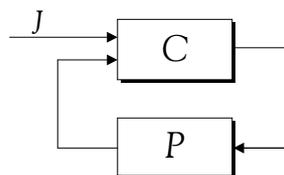


Figura 2.0-1: Sistema de Controlo.

O processo a controlar deverá ser descrito por um modelo matemático que seja representativo do processo, e ao mesmo tempo útil para análise e projecto do controlador. Pensado em termos de rigor, existe por vezes a possibilidade de desenvolver, com base no conhecimento físico do sistema, modelos precisos. Recorre-se aos princípios

fundamentais da mecânica, da química, da termodinâmica e obtêm-se os chamados modelos baseados em primeiros princípios, definidos por um conjunto de equações diferenciais. Contudo, na maioria dos casos, existem apenas dados de entrada e de saída e é com base nestes que se deve inferir um modelo matemático, procedimento vulgarmente designado por identificação experimental. Introduzem-se então modelos paramétricos, definidos por uma determinada estrutura (geralmente fixa) e caracterizados por um conjunto de parâmetros, a determinar através de um procedimento de estimação.

Se por um lado existe uma teoria sólida acerca de metodologias de identificação de sistemas lineares, invariantes no tempo, por outro a sua aplicação a processos do mundo real tende a fracassar. De facto, na prática os processos reais são complexos, variantes no tempo e não lineares. A aptidão para lidar com este tipo de factores, concretamente as não linearidades, a complexidade e a variância temporal é o ponto de partida para o uso das redes neuronais na identificação e no controlo de processos.

Mas se estas capacidades das redes proporcionam uma ferramenta poderosa a sua aplicação ao controlo continua a ser um assunto de investigação em aberto. Há ainda questões por responder: Qual a arquitectura mais apropriada para uma tarefa específica? Como deve a rede ser treinada, isto é, como devem ser determinados os seus parâmetros? Essencialmente são estas as questões abordadas neste capítulo, que se encontra dividido em quatro secções.

Na secção 2.1 apresentam-se conceitos genéricos de modelização, noções de redes neuronais e justifica-se a opção por estas últimas como modelos paramétricos a utilizar na modelização de processos.

Na secção 2.2 revêem-se as principais estruturas neuronais dando-se atenção especial às recorrentes, pela sua capacidade incorporar dinâmica de uma forma inerente, logo pela sua aptidão em modelizar sistemas dinâmicos. Destaca-se a sua interpretabilidade como modelos de estados, descritos portanto de uma forma adequada ao futuro projecto de controladores.

Na secção 2.3 aborda-se o problema do treino de uma rede recorrente, o mesmo é dizer, da estimação dos parâmetros que a caracterizam. Revêem-se algoritmos específicos e aborda-se o problema segundo outras duas perspectivas: uma, em que o treino é visto como um caso particular de um problema genérico de optimização não linear, de que são exemplos o método de Newton e o de Levenberg-Marquardt; outra, em que a estimação de parâmetros é efectuada através de técnicas de identificação estabelecidas para sistemas dinâmicos, de que é exemplo o filtro de Kalman.

Finalmente, na secção 2.4, referem-se algumas conclusões salientando-se o interesse das metodologias neuronais quando inseridas em estruturas de controlo.

# 2.1

## Modelização de Sistemas

*Nesta secção abordam-se os problemas de modelização e de identificação de uma forma genérica.*

*Introduzem-se as redes neuronais como modelos paramétricos particulares e enquadra-se o problema do seu treino como um procedimento genérico de estimação de parâmetros.*

### 2.1.1 Generalidades

#### Especificação do Problema

O procedimento de modelização tem por objectivo construir um modelo matemático capaz de descrever o comportamento estático e dinâmico de um determinado sistema. E, se por um lado, um modelo estático descreve apenas a relação entre as variáveis do processo em regime permanente ou estacionário, um modelo dinâmico implica necessariamente uma descrição, quer em regime estacionário, quer em regime transitório, devendo portanto ser capaz de descrever relações temporais entre as variáveis do processo. Na maioria dos sistemas reais as variáveis de entrada e de saída são analógicas, isto é, definidas no domínio contínuo. Assim, de forma a viabilizar o uso de computadores digitais, torna-se fundamental proceder à discretização dos sinais analógicos. Ao longo desta dissertação é assumido que os dados são discretos, a serem amostrados com um determinado intervalo de amostragem constante<sup>1</sup>. Na **Error! Reference source not found.**, usando um diagrama de blocos, descreve-se esquematicamente um sistema.

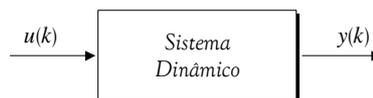


Figura 2.1-1: Representação esquemática de um sistema.

---

<sup>1</sup> A forma de como seleccionar correctamente o intervalo de amostragem não é objectivo deste estudo.

A variável  $u(k) \in \mathfrak{R}^{n_u}$  define a entrada do sistema no instante de tempo discreto  $k$  e a saída do sistema é representada por  $y(k) \in \mathfrak{R}^{n_y}$ . As variáveis de entrada e de saída podem ser escalares ou vectores. No primeiro caso, em que  $n_u = n_y = 1$ , o sistema é denominado SISO<sup>2</sup>. No segundo, quando existem várias entradas e várias saídas, o sistema é denominado MIMO<sup>3</sup>.

## Objectivos da Modelização

Quando se efectua a modelização de um sistema pensa-se, obviamente, em aprofundar o conhecimento sobre o sistema em questão. Como objectivos da modelização referem-se, entre outros:

- o *projecto de controladores* (Åström e Wittenmark, 1984), em que o modelo construído serve de base ao desenvolvimento e determinação dos parâmetros de um controlador;
- o *problema de predição*, quando se tem por objectivo prever valores futuros para algumas variáveis, a serem usados, por exemplo, em esquemas de controlo preditivo;
- a *estimação de variáveis não mensuráveis* (Kalman, 1960), quando não é possível medir directamente algumas das variáveis do sistema;
- na *área de diagnóstico de falhas*, constituindo um meio de detecção de funcionamentos anómalos do sistema (Chen e Patton, 1999).

Nesta dissertação a tarefa de modelização tem fundamentalmente em vista o primeiro objectivo, ou seja, o projecto de sistemas de controlo.

## Modelização Analítica e Modelização Experimental

Em relação aos procedimentos a seguir, com a finalidade de obter um modelo para um dado sistema, existem basicamente dois caminhos: modelização analítica e modelização experimental.

O primeiro, quando é possível obter o modelo apenas com base em conhecimento prévio, isto é, com base em leis físico-químicas, mecânicas ou termodinâmicas que o descrevam (também conhecido por modelo de primeiros princípios), ou ainda se é possível determinar uma estrutura para o modelo com base neste conhecimento, sendo os parâmetros desconhecidos estimados a partir de dados de funcionamento. Regra geral, o modelo analítico assim obtido resulta num sistema de equações diferenciais ou de diferenças, respectivamente para os casos contínuo e discreto. Contudo, o conhecimento necessário ao seu desenvolvimento é na maioria das vezes insuficiente, ou mesmo inexistente; na prática dispõe-se apenas de dados de operação, a partir dos quais é necessário inferir um modelo matemático para o sistema em causa.

---

<sup>2</sup> SISO- Single Input Single Output

<sup>3</sup> MIMO- Multiple Input Multiple Output

Assim, o segundo caminho apontado para a modelização é a construção de modelos baseados em dados, conhecido por identificação experimental. Dá origem a modelos vulgarmente conhecidos por modelos *black-box* (Ljung, 1999), em oposição aos modelos analíticos também conhecidos por modelos *white-box*. Esta denominação (*black-box*) deve-se ao facto de exprimir apenas relações empíricas entre as entradas e saídas, sem ter em conta o que se passa no interior do sistema.

Nesta dissertação o problema da modelização é abordado seguindo o último ponto de vista, ou seja, uma abordagem experimental.

## Formulação Matemática do Problema de Identificação

Em termos formais o problema pode ser estabelecido à custa de cinco factores (Ljung, 1999), (Soderström e Stoica, 1989): um sistema a identificar  $P$ , um conjunto de dados experimentais  $Z$ , uma estrutura particular  $M$ , um método de estimação e um critério de validação  $J$ . Em termos genéricos pode-se descrever o problema usando a seguinte equação:

$$y(k+1) = \mathfrak{S}(Z, M(\theta)) \quad (2.1-1)$$

Assume-se assim que existe uma função  $\mathfrak{S}$ , capaz de descrever a dinâmica do sistema, de tal forma que é possível estabelecer uma relação entre saídas futuras<sup>4</sup>,  $y(k+1)$ , e um conjunto de dados observados  $Z$  com base numa estrutura particular  $M(\theta)$ , em que  $\theta$  representa o conjunto de parâmetros que caracterizam a estrutura  $M$ .

A justificação teórica para a formulação do problema segundo esta descrição foi sugerida por Takens (1981). Takens mostrou que caso se considere um sistema sem ruído em que as observações dos dados possam ser feitas com infinita precisão, então é possível reconstruir com exactidão a dinâmica interna do sistema a partir de dados passados da saída. Além disso, a quantidade de dados necessária ( $N$ ) deverá verificar a relação  $N > 2n + 1$ , em que  $n$  define a dimensão do vector de estados do sistema. Na sua formulação original o teorema de Takens refere-se apenas a sistemas sem entradas externas. Casdagli reformulou o teorema de Takens, de forma a poder ser aplicado a sistemas entrada-saída, potenciando o seu uso na modelização de sistemas (Casdagli, 1991).

Na prática, porque se recorre a um conjunto finito de dados, e porque existem sempre perturbações, o processo de identificação tem de ser entendido como uma aproximação da função  $\mathfrak{S}$ . Assim, a equação (2.1-1) é na prática dada por (2.1-2),

---

<sup>4</sup> Nesta dissertação referem-se, por vezes, apenas sistemas SISO. No entanto, este facto justifica-se por simplicidade e clareza de escrita e, a menos que nada seja dito em contrário, pressupõe-se tratar de sistemas genéricos MIMO.

$$\hat{y}(k+1) = \hat{\mathfrak{S}}(Z, M(\hat{\theta})) = f(Z, M(\hat{\theta})) \quad (2.1-2)$$

em que variável  $\hat{y}(k+1)$  representa o valor estimado da saída real  $y(k+1)$ ,  $f = \hat{\mathfrak{S}}$  é uma aproximação de  $\mathfrak{S}$  e  $\hat{\theta}$  define uma estimativa, determinada com base nos dados observados, para os parâmetros  $\theta$  que caracterizam o sistema a identificar.

#### *Determinação do Modelo*

De forma a obter um modelo matemático experimental considera-se uma formulação recursiva, em quatro etapas distintas (Pintelon e Schoukens, 1991):

- i) *recolha de dados*: que consiste na obtenção de um conjunto de dados  $Z$ , o mais representativo possível do sistema a identificar;
- ii) *determinação da estrutura*: que tem por objectivo seleccionar, a partir de um conjunto de estruturas possíveis, uma estrutura particular  $M$ ;
- iii) *estimação de parâmetros*: na qual, para a estrutura seleccionada devem ser determinados, a partir de um determinado critério  $J$ , o conjunto ideal de parâmetros  $\hat{\theta}$ ;
- iv) *validação*: onde o modelo matemático determinado nos passos anteriores é testado, nomeadamente recorrendo a um conjunto de dados não utilizado na fase de estimação.

#### **i) Recolha de um conjunto de dados, $Z$**

O primeiro aspecto a ter em conta é o tipo de sinais que se deve considerar. Se um conjunto de dados possuir informação suficiente, de tal forma que os parâmetros do sistema possam ser determinados correctamente, diz-se que o sinal é persistentemente excitante (Ljung, 1999). Se para sistemas lineares esta propriedade garante, sob certas condições bem estabelecidas, a convergência e consistência da estimação dos parâmetros, não existem resultados teóricos análogos para sistemas não lineares. Neste caso apenas se poderá reivindicar que os dados serão mais ricos, possibilitando assim, em princípio, melhores resultados de identificação.

#### **ii) Selecção de uma estrutura particular, $M$**

De todos os passos este é, sem dúvida, o mais importante das etapas de identificação. A caracterização de uma determinada estrutura, considerada representativa do processo que se pretende identificar, pode ser efectuada segundo várias perspectivas, das quais se referem: a partir da descrição do sistema, usando uma formulação entrada-saída ou no espaço de estados; a partir do tipo e do número de dados utilizados; a partir das relações a estabelecer entre entradas e saídas, nomeadamente lineares ou não lineares, estáticas ou dinâmicas; a partir de conceitos determinísticos ou estocásticos.

### Modelos Lineares

A estrutura linear é certamente a mais simples que se pode considerar para representação da função  $\mathfrak{S}$ . A equação (2.1-3) descreve um modelo SISO linear, conhecido vulgarmente por modelo ARX<sup>5</sup>, em que a estimação de saídas futuras é feita a partir de valores passados de entradas e saídas (Ljung, 1999).

$$\hat{y}(k) = \hat{b}_1 u(k-1) + \dots + \hat{b}_{n_b} u(k-n_b) - \hat{a}_1 y(k-1) - \dots - \hat{a}_{n_a} y(k-n_a) + \zeta(k) \quad (2.1-3)$$

A variável  $\hat{y}(k) \in \mathfrak{R}$  representa o valor estimado da saída actual  $y(k)$ ,  $\hat{b}_i$  e  $\hat{a}_i$  são estimativas dos valores reais  $b_i$  e  $a_i$  que definem o modelo linear e  $\zeta(k)$  representa um processo estocástico (ruído branco), definido como uma sequência de variáveis aleatórias, independentes e igualmente distribuídas. As constantes  $n_b$  e  $n_a$  representam o número de valores passados considerados, respectivamente para a entrada e saída, cujos valores são normalmente deduzidos a partir de conhecimento que exista sobre o comportamento transitório do sistema a identificar.

Os modelos lineares têm sido aplicados a inúmeras situações em diversos campos de engenharia. A relativa facilidade na estimação dos seus parâmetros ( $b_i$  e  $a_i$ ) e a teoria bem estabelecida relativamente a sistemas lineares, baseados em parte no uso de funções de transferência para a descrição da dinâmica do sistema, justificam o seu sucesso, Mendel (1986) e Landau *et al.* (1997).

### Modelos Não Lineares

Embora tenham sido obtidos resultados aceitáveis aplicados a um grande número de aplicações práticas, as capacidades de modelização das representações lineares são por vezes insuficientes para descrever com rigor o sistema que se pretende modelizar. De facto, porque a maioria das aplicações reais são não lineares um modelo linear mostra-se inadequado. Surgem desta forma os modelos não lineares, como uma generalização dos modelos lineares, de que é exemplo o modelo NARX<sup>6</sup>, descrito por (2.1-4), (Leontaritis e Billings, 1985) e (Chen e Billings, 1989).

$$\hat{y}(k) = f(u(k-1), \dots, u(k-n_b), y(k-1), \dots, y(k-n_a) + \zeta(k)) \quad (2.1-4)$$

A função  $f: \mathfrak{R}^d \rightarrow \mathfrak{R}^{n_y}$  ( $d = n_a + n_b$ ) é uma função não linear, a partir da qual se pretende aproximar a função  $\mathfrak{S}$ , definida em (2.1-1).

Outro aspecto que permite caracterizar uma determinada estrutura relaciona-se com o tipo de sinais considerados, ou seja, o comportamento dinâmico de um sistema pode ser descrito matematicamente de uma forma determinística ou, em alternativa, recorrendo a conceitos estocásticos. A introdução de ruído branco  $\zeta(k)$ , equações (2.1-3) e (2.1-4), origina que os valores da saída não sejam determinísticos, isto é, valores bem

<sup>5</sup> ARX- AutoRegressive with eXogenous input

<sup>6</sup> NARX- Nonlinear AutoRegressive with eXogenous input model

definidos a partir de informação passada, mas entendidos como variáveis aleatórias. Esta dissertação tem em conta apenas aspectos determinísticos, considerando-se desprezável a importância relativa do ruído em relação aos sinais de entrada e de saída.

#### Formulação Genérica

Do exposto, e de uma forma genérica, o processo de modelização pode ser definido através da equação (2.1-5).

$$\hat{y}(k) = f(\varphi(k), \hat{\theta}) \quad (2.1-5)$$

Decompõe-se assim o problema de modelização em dois problemas parciais: a escolha do vector de regressão,  $\varphi(k) \in \mathfrak{R}^d$ , da forma,

$$\varphi(k) = [ u(k-1), \dots, u(k-n_b), y(k-1), \dots, y(k-n_a) ] \quad (2.1-6)$$

e a especificação da estrutura para a função não linear  $f(\cdot)$ , que efectua a transformação do espaço de regressão no espaço de saída,  $f: \mathfrak{R}^d \rightarrow \mathfrak{R}^{n_y}$ .  $\hat{\theta} \in \mathfrak{R}^{n_\theta}$  define o conjunto dos  $n_\theta$  parâmetros a determinar.

Relativamente à especificação da estrutura que caracteriza a função não linear  $f(\cdot)$ , refere-se a abordagem considerada por Sjöberg *et al.* (1994) e (1995), na qual é proposta uma generalização para os modelos não lineares do tipo *black-box*. No seu trabalho, e de forma a inferir uma aproximação para  $\mathfrak{S}$ , sugerem a sua expansão segundo uma base de funções, da seguinte forma.

$$\mathfrak{S}(\varphi(k), \theta) = \sum_i \hat{\theta}(i) g_i(\varphi(k)) \quad (2.1-7)$$

A função não linear  $g_i(\varphi(k))$ , definida como função base, estabelece um mapeamento  $\mathfrak{R}^d \rightarrow \mathfrak{R}$ , em que  $d$  define a dimensão do vector de regressão. A variável  $\hat{\theta}(i)$  representa os componentes do vector  $\hat{\theta}$ .

Por exemplo, o modelo linear (2.1-3), é um caso particular de (2.1-7), em que a função base coincide com o vector de regressão, ou seja,  $g_i(\varphi(k)) = \varphi(k)$ ,  $i = 1, \dots, d$ . O vector de parâmetros é definido por  $\hat{\theta} = [ \hat{b}_1 \dots \hat{b}_{n_b} \hat{a}_1 \dots \hat{a}_{n_a} ]$ .

Uma vez considerada a descrição de um modelo genérico segundo (2.1-7) coloca-se o problema de especificar quais as funções  $g_i(\varphi(k))$  que permitem efectuar essa expansão.

Uma possibilidade consiste em assumir apenas um tipo de funções  $\vartheta(\varphi(k))$ , ou seja, qualquer função  $g_i(\varphi(k))$  pode ser definida por (2.1-8),

$$g_i(\varphi(k)) = \vartheta(\varphi(k), \beta_i, \gamma_i) \quad (2.1-8)$$

em que  $\beta_i$  e  $\gamma_i$  definem parâmetros de diferentes características: ao parâmetro  $\beta_i$  está associado um ganho (operação multiplicação) e a  $\gamma_i$  um desvio (operação adição). Usando esta descrição, Sjöberg *et al.* (1994) e (1995) interpretam, como casos particulares, vários modelos normalmente utilizados: séries de Fourier, *Waveletes*, funções *Spline*, redes neuronais com funções de activação do tipo sigmoidal, redes neuronais de base radial e sistemas difusos, entre outros.

As redes neuronais com funções de activação do tipo sigmoidal, assunto de interesse nesta dissertação, constituem de facto um caso particular de (2.1-8). Na verdade, como se verá futuramente, uma rede neuronal com funções de activação do tipo sigmoidal,  $\sigma(\cdot)$ , pode ser descrita na forma (2.1-9), como um caso particular de (2.1-8).

$$g_i(\varphi(k)) = \sigma(\beta_i \varphi(k) + \gamma_i) \quad (2.1-9)$$

#### Modelos de estados

Outra estrutura alternativa aos modelos entrada-saída é a formulação no espaço de estados. A equação (2.1-10) descreve um sistema discreto, usando uma representação no espaço de estados.

$$x(k+1) = f(x(k), u(k)) \quad (2.1-10a)$$

$$y(k) = h(x(k)) \quad (2.1-10b)$$

A variável  $x(k) \in \mathfrak{R}^n$  define o vector de estado, no instante  $k$ . As funções  $f: \mathfrak{R}^n + n_u \rightarrow \mathfrak{R}^n$  e  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n_y}$  representam mapeamentos não lineares. Note-se que, tal como referido para os modelos entrada-saída, os modelos de estado podem ser obtidos tanto a partir da observação de dados do sistema (modelos experimentais) como a partir do conhecimento físico do sistema (modelos analíticos).

Têm sido apontadas vantagens no uso de modelos de estados (Rivals e Personnaz, 1995a) e (Rivals, 1995b). Primeiro, porque os modelos de estados podem descrever uma gama mais vasta de sistemas dinâmicos do que os modelos de entrada-saída, pois nestes últimos apenas a saída  $y(k)$  tem valores desejados enquanto as variáveis de estado  $x(k)$  não são impostas. Desta forma os modelos de estado são mais flexíveis, isto é, possuem uma maior capacidade de descrever sistemas complexos. Segundo, os modelos de estado requerem um menor número de observações passadas. Enquanto neste tipo de modelos a saída actual é apenas função do valor do estado e da entrada no instante anterior, nos modelos de entrada-saída são necessários  $n_a$  e  $n_b$  valores passados de saídas e entradas, respectivamente.

Outra das vantagens da representação no espaço de estados reside na abordagem de sistemas MIMO de forma idêntica à de sistemas SISO, o que não é verdade no caso de sistemas representados por funções de transferência. Além disso, é sempre possível descrever um modelo entrada-saída, linear ou não linear, no correspondente espaço de estados. No entanto, o contrário não é necessariamente verdadeiro, isto é, pode não ser possível encontrar uma representação entrada-saída para um modelo de estados (Leontaritis e Billings, 1985).

### iii) Estimação de parâmetros $\hat{\theta}$ : minimização de um critério $J$

Depois de seleccionada a estrutura, que se assume ser capaz de representar o comportamento de um determinado sistema, deve-se determinar o conjunto de parâmetros  $\hat{\theta}$  que a caracterizam. De forma a estimar os parâmetros  $\hat{\theta}$  é utilizado como ponto de partida um conjunto de dados de treino  $Z$  (2.1-11), consistindo em  $N$  exemplos de observações passadas e correspondentes valores para a saída  $y(k)$ .

$$Z = [ \Phi(k) Y(k) ] \quad (2.1-11)$$

$\Phi$  é constituído pelos  $N$  vectores de regressão  $\Phi = [ \varphi(k-1) \dots \varphi(k-N) ]^T$  e  $Y$  pelas correspondentes  $N$  saídas  $Y = [ y(k-1) \dots y(k-N) ]^T$ .

Uma abordagem típica, usual em problemas de estimação de parâmetros, consiste em minimizar uma função de custo, definida pelo somatório dos quadrados dos erros, ( $SSE^7$ ), (2.1-12).

$$J(\hat{\theta}) = \frac{1}{2} \sum_{i=1}^N [e_y^T(k) e_y(k)] \quad (2.1-12)$$

O termo  $e_y(k) \in \mathfrak{R}^{n_y}$  define o erro de identificação em cada instante  $k$ , dado por (2.1-13),

$$e_y(k) = y(k) - \hat{y}(k, \hat{\theta}) \quad (2.1-13)$$

em que a variável  $\hat{y}(k, \hat{\theta})$  representa a saída do modelo, dados os parâmetros  $\hat{\theta}$ . A minimização do critério (2.1-12) é um processo complexo uma vez que na prática, e de uma forma genérica, não é possível obter uma solução analítica para o problema. Assim, a solução do critério é quase sempre realizada à custa de um processo iterativo, assunto a ser analisado nas secções 2.3 e 4.2.

### iv) Validação do modelo obtido

Apesar de se pretender minimizar um determinado critério, o maior ou menor valor deste pode ser insuficiente para concluir da qualidade do modelo obtido. Uma medida

---

<sup>7</sup> SSE- Sum of Squared Error

mais relevante da qualidade do modelo é a sua capacidade de generalização, isto é, a sua capacidade de predição em exemplos não usados como dados de treino (Ljung, 1999).

Outra das formas de aferir a qualidade de um modelo é através da sua complexidade. Se um modelo é demasiado simplista não será suficientemente flexível para descrever a dinâmica do sistema a modelizar, conduzindo a uma situação de sub-ajustamento. Por outro lado, se o modelo é demasiado complexo, então o número elevado de graus de liberdade permitirá aproximar não só os dados de treino mas também eventualmente o ruído adicional. Esta última situação é conhecida por sobre-ajustamento, e para ambas as situações se reconhecem dificuldades de generalização (Haykin, 1994) e (Bishop, 1995a).

Tal como sugerido por Akaike deve-se respeitar o princípio de que o modelo ideal deve ser conseguido à custa de um compromisso entre a melhor aproximação e o menor número de parâmetros possível (Ljung, 1999). Outra solução apontada para o problema de sobre-ajustamento consiste em otimizar o número de parâmetros por uma operação de poda (*pruning*). Exemplos deste tipo de técnica são as metodologias *Optimal Brain Damage* (Le Cun *et al.*, 1990) e *Optimal Brain Surgeon* (Hassibi *et al.*, 1993).

## 2.1.2 Técnicas de *Soft Computing* Aplicadas à Identificação

Apesar da teoria clássica de identificação de sistemas ter por base técnicas bem estabelecidas, o que é particularmente verdadeiro para o caso linear, esta tem tido dificuldades em sistemas complexos, não lineares e variantes no tempo. Recentemente, têm surgido formas alternativas/complementares na tentativa de ultrapassar estas dificuldades das quais se destacam as técnicas de *soft computing*.

Como referido, as redes neuronais são constituídas essencialmente por elementos de processamento simples (neurónios) interligados entre si por conexões pesadas (Haykin, 1994). Estas estruturas, inicialmente pensadas para emularem o funcionamento de aprendizagem do cérebro humano, apresentam nas suas capacidades de aproximação e de adaptação o seu principal atractivo, razão principal pelo qual têm sido usadas com sucesso na modelização de sistemas de dinâmica complexa. Se, por um lado, estas propriedades constituem um argumento positivo a seu favor, por outro, o tipo de informação que nelas é possível incorporar constitui um argumento contrário. De facto, a informação por elas armazenada não é possível de interpretar de uma forma qualitativa, manifestando um problema de falta de transparência. Além disso, a existência de conhecimento prévio, que possa existir sobre o sistema a modelizar, não é facilmente inserida no modelo neuronal.

Como grande vantagem dos sistemas difusos, refere-se a sua capacidade de representar informação de uma forma transparente, através de informação linguística, estabelecida por um conjunto de regras. Mas, se apresenta vantagens em termos de transparência, apresenta também desvantagens em termos de projecto: a definição e a sintonização das funções de pertença que definem quantitativamente os valores

linguísticos, é na prática moroso e quase sempre efectuada por um processo total ou parcialmente empírico. É especialmente interessante para sistemas complexos, cuja análise e projecto são difíceis usando abordagens tradicionais quantitativas, mas que na prática pode ser descrito por um operador humano de uma forma qualitativa, já que estes modelos facultam um meio ideal de incorporar o conhecimento humano, heurístico e pouco preciso. De facto, o conhecimento fornecido pela experiência de um operador, definido por meio de um conjunto de regras difusas, expressa de uma forma qualitativa as dependências entre as variáveis do sistema a modelizar.

Os algoritmos genéticos consistem basicamente num algoritmo de optimização que emula os princípios da evolução natural, tendo como grande vantagem a sua capacidade de obter soluções globais para problemas de optimização. Têm, no entanto, nos elevados tempos de computação a sua maior desvantagem.

Como também já foi referido, apesar destas metodologias poderem ser implementadas isoladamente, elas não são incompatíveis, antes pelo contrário complementam-se, dando origem às chamadas metodologias híbridas. As redes neuronais podem ser combinadas com os sistemas difusos e exploradas as suas capacidades de tolerância a ambientes imprecisos e incertos, aprendendo a partir da experiência (dados) e adaptando-se às condições de operação. Concilia-se assim a capacidade dos sistemas difusos de lidar com incertezas e a capacidade de adaptação das redes neuronais. Recorrendo a técnicas de aprendizagem, pode-se automaticamente sintonizar as funções de pertinência, reduzindo assim o tempo de projecto e aumentando, em princípio, o desempenho do sistema difuso a projectar.

Um exemplo de um sistema neuro-difuso foi proposto por Paiva (1999), no qual as regras difusas podem ser interpretadas directamente a partir da estrutura de uma rede neuronal, sendo a rede treinada através de um procedimento normal de retropropagação. Adicionalmente, é possível efectuar a inicialização prévia dos parâmetros da rede, através da incorporação de um conjunto de regras que traduzam a experiência e conhecimento existentes *à priori*.

## Modelização por Redes Neurais

Foi estabelecido, como um dos objectivos desta dissertação, a utilização de redes neuronais na identificação e controlo de sistemas. Os procedimentos a seguir no que respeita ao problema de identificação, conhecido no contexto das redes neuronais por aprendizagem ou treino supervisionado, são em tudo idênticos aos referidos de uma forma genérica: assume-se existir um conjunto de dados,  $Z$ , uma estrutura paramétrica (uma rede neuronal)  $M$ , para a qual são determinados os parâmetros (pesos das ligações)  $\hat{\theta}$ , através de um determinado método de estimação e segundo um determinado critério (conhecido por erro de treino),  $J$ .

O treino diz-se supervisionado uma vez que é necessária a existência de um supervisor que forneça à rede os valores desejados para a sua saída. Exemplo de um algoritmo deste tipo é o algoritmo de retropropagação, (*backpropagation*) (Rumelhart *et*

al., 1986). De forma distinta, no treino não supervisionado, não é fornecida à rede o valor desejado para a sua saída, sendo apenas fornecidos os valores das entradas, devendo a rede ser capaz de classificar as entradas em grupos com características similares. Exemplos de algoritmos de treino não supervisionado são os algoritmos competitivos de Kohonen (1987) e o algoritmo ART<sup>8</sup> (Carpenter e Grossberg, 1987).

Relativamente à notação utilizada no contexto das redes neuronais, refere-se a seguinte.

$$y_n(k) = \mathcal{N}(\varphi(k), W) \quad (2.1-14)$$

Por analogia com (2.1-5) à função  $f(\cdot)$  equivale o modelo neuronal  $\mathcal{N}(\cdot)$ , à saída do modelo genérico  $\hat{y}(k)$  corresponde a saída do modelo neuronal  $y_n(k)$ , e aos parâmetros  $\hat{\theta}$  correspondem os pesos da rede,  $W$ .

## 2.1.3 Redes Neuronais

### Introdução

As origens das redes neuronais remontam a 1943 com a introdução do trabalho de McCulloch e Pitts (1943). McCulloch e Pitts propuseram um modelo matemático capaz de representar um evento no sistema nervoso, definindo para tal um neurónio artificial. Mostraram que através desta estrutura seria possível, em princípio, realizar qualquer operação aritmética ou função lógica. No entanto, não apresentaram qualquer método para a determinação dos parâmetros do seu modelo.

O neurónio definido por McCulloch e Pitts, ilustrado na Figura 2.1-2, é constituído por  $n_u$  entradas  $u_i$ ,  $i = 1, \dots, n_u$  e uma só saída,  $y$ , estando associado a cada uma das entradas um peso,  $w_i$ . A soma ponderada das entradas,  $a$ , designada por activação do neurónio, depois de transformada por uma função não linear  $\sigma(\cdot)$ , permite obter a sua saída  $y$ , definida pela equação (2.1-15).

$$a = \sum_{i=1}^{n_u} w_i u_i + b \quad (2.1-15a)$$

$$y = \sigma(a) \quad (2.1-15b)$$

---

<sup>8</sup> ART- Adaptive Resonance Theory

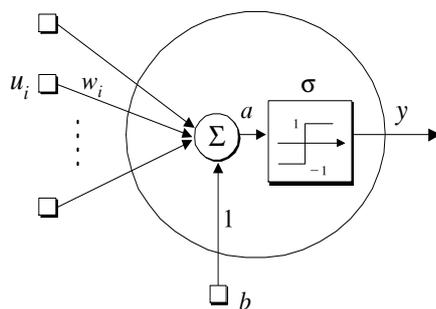


Figura 2.1-2: Neurónio de McCulloch e Pitts.

O parâmetro  $b$ , conhecido por termo de polarização (*bias*) permite obter uma saída não nula quando todas as entradas  $u_i$  são nulas. A função de activação por eles utilizada foi a função binária (*hard-limiter*), definida por (2.1-16).

$$\sigma(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (2.1-16)$$

A escolha da função de activação depende da aplicação em causa, apresentando-se algumas das funções mais utilizadas na Tabela 2.1-1, visualizadas na Figura 2.1-3.

Função de activação	Equação
Binária	$\sigma(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases}$
Linear	$\sigma(x) = x$
Sigmoidal Unipolar: Função logística	$\sigma(x) = \frac{1}{1 + e^{-\alpha x}}, \alpha \in \mathfrak{R}$
Sigmoidal Bipolar: Tangente Hiperbólica	$\sigma(x) = \text{tgh}(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}}, \alpha \in \mathfrak{R}$
Gaussiana	$\sigma(x) = e^{-\frac{\ x - c\ ^2}{2v^2}}, x, c, v \in \mathfrak{R}^n$

Tabela 2.1-1: Exemplos de funções de activação.

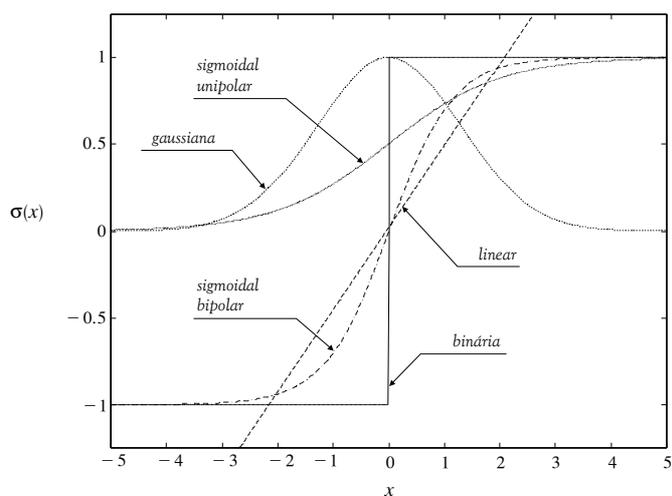


Figura 2.1-3: Exemplos de funções de activação.

A contribuição que se seguiu ocorreu em 1949 com o trabalho de Hebb (1949), propondo, pela primeira vez, uma regra de aprendizagem para os pesos da rede. Posteriormente, seguindo as ideias de Hebb, Rosenblatt (1958) introduziu o perceptrão, um tipo de modelo semelhante ao neurónio de McCulloch e Pitts. Para o perceptrão Rosenblatt propôs uma regra de aprendizagem, para a qual provou as suas propriedades de convergência, através do célebre teorema de convergência do perceptrão.

Em 1960 Widrow e Hoff (1960) propuseram a *ADALINE*<sup>9</sup>, cuja topologia era equivalente a um perceptrão de apenas uma camada e a *MADALINE*, uma rede *ADALINE* multi-camada. A sua estrutura era similar, tanto na topologia como nas capacidades de aproximação, ao perceptrão de Rosenblatt, contudo, em vez da função de activação considerada ser a binária, usaram como função de activação a linear. Introduziram também um método de aprendizagem com base no método dos mínimos quadráticos ( $LS^{10}$ ), mais eficaz do que a regra de aprendizagem proposta por Rosenblatt para o perceptrão. Note-se ainda que a aplicação do perceptrão a problemas de classificação é referida como a primeira aplicação prática de redes neuronais (Widrow e Smith, 1963).

Em 1969 Minsky e Papert (1969) apontaram as limitações das topologias existentes na altura. Num exemplo clássico, demonstraram que o perceptrão era incapaz de modelizar a simples operação OU-exclusivo (*X-OR*). De facto, a regra de aprendizagem do perceptrão e o método dos mínimos quadráticos foram desenvolvidos apenas para redes de uma camada e, por isso, ambas as topologias sofriam da mesma limitação: apenas permitiam classificar problemas linearmente separáveis. Ambos os autores, Rosenblatt e Widrow e Hoff, perceberam que esta dificuldade seria possível de ultrapassar caso se considerassem redes multi-camada, contudo não foram capazes de generalizar os seus algoritmos a este tipo de estruturas.

<sup>9</sup> *ADALINE*- *AD*aptive *LI*near *E*lement

<sup>10</sup> *LS*- *L*east *S*quares

Em parte devido às críticas apresentadas por Minsky e Papert, e também pelas limitações inerentes aos computadores da altura, a investigação relacionada com as redes neuronais foi praticamente inexistente nos anos 70. Uma das excepções foi a pesquisa realizada por Grossberg sobre *Self Organization Maps* usando regras de aprendizagem competitivas (Grossberg, 1976). As suas ideias viriam a servir de base à metodologia ART, (Carpenter e Grossberg, 1987). Ainda nesta altura outros investigadores contribuíram pontualmente para o desenvolvimento da área. São exemplos dessa investigação os trabalhos efectuados no campo das memórias associativas por Anderson (1972) e Kohonen (1972).

Nos anos 80 duas contribuições fundamentais viriam a tornar-se decisivas para o ressurgimento da investigação na área das redes neuronais.

A primeira, quando Hopfield (1982) estabeleceu uma nova forma de entender a computação efectuada por uma rede neuronal recorrente (ou dinâmica). Considerando uma rede dinâmica estável estabeleceu um novo princípio de armazenamento de informação, tal como uma memória associativa. Usando o conceito de energia e a teoria de estabilidade de Lyapunov enunciou condições de forma a assegurar a estabilidade da sua estrutura recorrente. Mais tarde, Cohen e Grossberg (1983), propuseram um princípio geral para o projecto de redes recorrentes, que incluía a rede de Hopfield como um caso particular. No seu trabalho, foram estabelecidos resultados fundamentais na estabilidade de redes neuronais recorrentes usando o método directo da teoria de estabilidade de Lyapunov.

A segunda grande contribuição foi o bem conhecido algoritmo de retropropagação ( $BP^{11}$ ), para o treino de redes neuronais multi-camada, proposto por Rumelhart *et al.* (1986). Na verdade algoritmos capazes de treinar redes multi-camada tinham já sido propostos independentemente por Werbos (1974) e Le Cun (1985). Contudo, foi apenas com a publicação do trabalho de Rumelhart, para o qual também contribuíram as capacidades computacionais em franco progresso, que recuperou o interesse da comunidade científica para o estudo das redes neuronais.

## Algumas Características das Redes Neuronais

### Redes Multi-Camada

Sabia-se que o problema de classificação inerente ao perceptrão poderia ser resolvido se os perceptrões fossem combinados, de forma a obter uma rede multi-camada. Na Figura 2.1-4 mostra-se uma rede multi-camada  $MLNN^{12}$ , com quatro camadas. Neste caso particular a função de activação usada na segunda e terceira camada, também conhecidas por camadas escondidas ou internas, é a tangente hiperbólica (Tabela 2.1-1), sendo usada na camada da saída uma função de activação linear.

---

<sup>11</sup> BP- Backpropagation

<sup>12</sup> MLNN- MultiLayer Neural Network

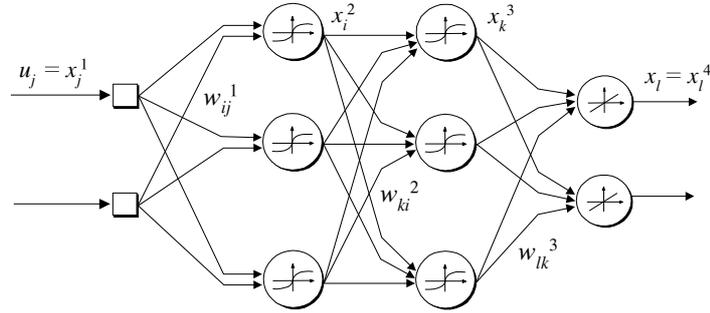


Figura 2.1-4: Rede neuronal com quatro camadas.

Para uma rede com  $L$  camadas considera-se a entrada como a primeira camada ( $l = 1$ ), a saída como a última ( $l = L$ ), sendo as restantes  $L - 2$  internas. Define-se  $N_l$  como o número de neurónios em cada camada, e  $x_i^l$  como a saída do neurónio  $i$  da camada  $l$ . Das definições é possível escrever, para as camadas  $l = 2, \dots, L$ , (2.1-17).

$$a_i^l = \sum_{j=1}^{N_{l-1}} w_{ij}^{l-1} x_j^{l-1} + b_i^{l-1} \quad (2.1-17a)$$

$$x_i^l = \sigma(a_i^l) \quad (2.1-17b)$$

Como casos particulares do caso geral (2.1-17) referem-se os casos em que  $l = 2$ , onde é considerada como entrada  $x_j^1 = u_j$ , e o caso em que  $l = L$  em que se considera a função de activação  $\sigma(\cdot)$  linear. Para o exemplo particular considerado na Figura 2.1-4 tem-se  $L = 4$ ,  $N_1 = n_u = 2$ ,  $N_2 = N_3 = 3$ ,  $N_4 = n_y = 2$ .

### Interpretação como uma função base

Tal como foi referido anteriormente, a partir dos resultados de Sjöberg *et al.* (1994) e (1995), o problema de modelização consiste na escolha de uma função base segundo a qual é efectuada uma expansão da função a aproximar. Seguindo esta perspectiva as redes neuronais constituem uma das funções possíveis de utilizar como base nessa expansão.

Por um lado, uma rede neuronal do tipo sigmoideal (2.1-17), com uma unidade interna e apenas uma saída,  $y_n$ , pode ser definida por (2.1-18),

$$y_n = \sum_{i=1}^N w_i^2 \sigma \left( \sum_{j=1}^{n_u} w_{ij}^1 u_j + b_i^1 \right) \quad (2.1-18)$$

ou ainda

$$y_n = \sum_{i=1}^N w_i^2 \sigma(\vec{w}_i^1 \mathbf{v} + b_i^1) \quad (2.1-19)$$

em que  $\vec{w}_i^1 \in \mathfrak{R}^{n_u}$  representa a linha  $i$  da matriz  $W^1$ .  $N$  define o número de neurónios na unidade interna,  $\mathbf{v} \in \mathfrak{R}^{n_u}$  representa a entrada e  $w_i^2 \in \mathfrak{R}$  os pesos da unidade interna para a unidade de saída. O parâmetro  $b_i^1 \in \mathfrak{R}$  corresponde a um termo de polarização, relativo à unidade interna.

Por outro, se for escolhida como um caso da equação geral (2.1-7) a função base já definida em (2.1-9)

$$g_i(\varphi(k)) = \sigma(\beta_i \varphi(k) + \gamma_i)$$

em que  $\beta_i \in \mathfrak{R}^d$  define um vector de parâmetros, da mesma dimensão do vector de regressão  $\varphi(k)$ , sendo  $\gamma_i$  um escalar, pode-se escrever (2.1-20).

$$y_n = \sum_{i=1}^N \hat{\theta}_i \sigma(\beta_i \varphi(k) + \gamma_i) \quad (2.1-20)$$

Esta última equação representa na verdade uma rede com uma unidade interna, (2.1-19), como representado na Figura 2.1-5. À entrada corresponde o vector de regressão,  $\varphi(k) = \mathbf{v}$ , os pesos da entrada para a unidade interna são definidos por  $\beta_i = \vec{w}_i^1$ , e os desta última para a de unidade de saída por  $\hat{\theta}_i = w_i^2$ . O parâmetro  $\gamma_i = b_i$  representa o termo de polarização.

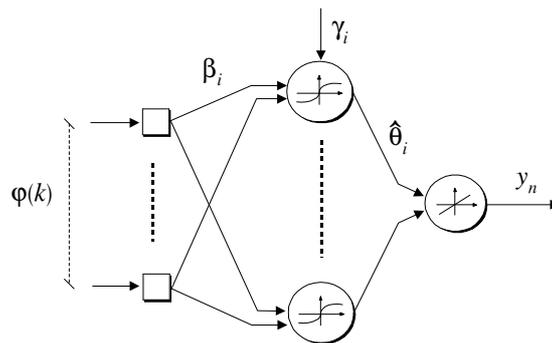


Figura 2.1-5: As redes neuronais como uma função base.

Entendida desta forma, uma rede neuronal permite efectuar a expansão de uma determinada função a partir de funções base  $g_i(\varphi(k))$  definidas por (2.1-9), ou seja, pelos neurónios que constituem a unidade interna.

### Algumas vantagens

A partir de uma rede com uma única camada interna, Cybenko (1989), Funahashi (1989), Hornik *et al.* (1990), e Hornik (1991) mostraram que se fossem usados na camada interna um número suficiente de neurónios (com funções de activação do tipo sigmoidal) e usando um mecanismo de treino adequado seria possível aproximar, com um grau de precisão arbitrário, uma qualquer função contínua não linear. No entanto outras estruturas, tais como séries trigonométricas, expansões polinomiais, *Splines* e funções ortogonais gozam da mesma propriedade. Surge naturalmente a questão do porquê do sucesso das redes neuronais.

Uma das razões principais prende-se com o problema da maldição da dimensionalidade (*curse of dimensionality*). Barron (1993) mostrou, ainda que parcialmente, que a parametrização usando redes neuronais é vantajosa em relação a expansões usando outras abordagens, especialmente em espaços de grande dimensão. Concretamente, para uma classe particular de funções  $f: \mathfrak{R}^d \rightarrow \mathfrak{R}$ , sendo  $d$  a dimensão do espaço de entrada, Barron mostrou que a aproximação destas com uma rede MLNN, definida por uma única camada interna, seria possível de caracterizar pela seguinte norma quadrática de erro (2.2-21).

$$\|f(x) - \mathcal{N}(x)\|_2 < \mathfrak{O}\left(\frac{C_f}{M^{1/2}}\right) \quad (2.2-21)$$

$\mathcal{N}(x)$  representa a saída do modelo neuronal para uma entrada  $x$ .  $\|\cdot\|_2$  define a norma quadrática,  $C_f$  é um valor finito (primeiro momento da transformada de Fourier da função  $f$ ) e  $M$  é o número de neurónios da camada interna.

Por outro lado, mostrou que não existe nenhuma combinação linear de  $M$  funções incluindo polinómios, séries trigonométricas ou funções *Splines*, tais que a saída por elas obtida,  $\hat{f}(\cdot)$ , possam alcançar um erro inferior a (2.2-22).

$$\|f(x) - \hat{f}(x)\|_2 < \mathfrak{O}\left(\frac{C_f}{M^{1/d}}\right) \quad (2.2-22)$$

Assim, neste último caso, à medida que a dimensão do espaço de entrada ( $d$ ) aumenta tornam-se claras as vantagens dos modelos neuronais.

## 2.1.4 Conclusão

Além dos progressos teóricos realizados, de que é exemplo o anterior (Barron), os resultados práticos têm mostrado que as redes possuem propriedades notáveis quando aplicadas a problemas de identificação, principalmente no caso de sistemas não lineares. Se as capacidades de aproximação de funções não lineares constituem o ponto de partida para a identificação de sistemas não lineares, outras características fundamentais justificam a sua utilização neste contexto:

- a sua capacidade de aprendizagem a partir de exemplos, isto é, a partir de informação de entrada-saída;
- a sua habilidade de aprender permanentemente novas situações, ou seja, de se adaptar em-linha a novas situações;
- o seu potencial para elevadas velocidades de computação, facto que se justifica pela possibilidade de paralelismo e pela facilidade de implementação em hardware;
- o facto de a informação ser distribuída por toda a estrutura favorece a sua robustez e tolerância a falhas;
- a generalidade do seu projecto, isto é, uma vez determinada a sua estrutura, existem formas bem definidas, genéricas, de determinar os seus parâmetros;
- a forma de tratar sistemas *MIMO*, analogamente a sistemas *SISO*.

Por estas razões, as redes neuronais possuem características notáveis que lhes tem permitido ser utilizadas, com sucesso, como ferramentas computacionais na modelização de sistemas.

# 2.2

## Estruturas de Redes Neurais

*Depois de na secção anterior se terem abordado os problemas de modelização e terem sido introduzidos os conceitos básicos de redes neuronais, aprofunda-se nesta secção o estudo acerca de estruturas neuronais.*

*Consideram-se as redes neuronais estáticas, capazes de representar apenas mapeamentos estáticos e introduzem-se as redes recorrentes propriamente ditas, capazes de incorporar características dinâmicas.*

### 2.2.1 Redes Neurais Estáticas

Na secção anterior definiu-se matematicamente uma rede neuronal, introduzindo-se o perceptrão e a rede multi-camada *MLNN*. Uma vez que o fluxo de informação se efectua apenas num sentido, da entrada para a saída, e não existe recorrência, as redes multi-camada são também usualmente designadas por redes com ligações para a frente (redes *feedforward*).

Nesta secção revêem-se os resultados principais sobre as capacidades universais de aproximação das redes neuronais, uma das características fundamentais que tem incentivado o interesse por estas. Serão revistos nomeadamente os resultados relativos a redes *MLNN* e a redes *RBFNN*<sup>1</sup>. Introduce-se o teorema de Kolmogorov e mostra-se como pode ser utilizado na demonstração das capacidades de aproximação, no caso particular de redes multi-camada com funções de activação do tipo sigmoideal. Depois, ainda com o mesmo objectivo, faz-se referência ao teorema de Stone-Weierstrass e a sua relação com as redes neuronais, nomeadamente com as *RBFNN*.

### Os teoremas de Kolmogorov e de Kurkova

Em 1900 Hilbert propôs uma série de 23 problemas, conjecturando no décimo terceiro a possibilidade de existirem funções contínuas de três variáveis que não poderiam ser representadas como uma sobreposição de funções contínuas de apenas duas

---

<sup>1</sup> *RBFNN*- Radial Basis Function Neural Network

variáveis. Os resultados de Kolmogorov (Kolmogorov, 1957) vieram contrariar a hipótese de Hilbert. Com esse propósito Kolmogorov enunciou o seguinte teorema:

**Teorema (Kolmogorov, 1957)**

Qualquer função contínua  $f(x_1, \dots, x_n)$  definida no espaço de dimensão  $n$ ,  $E^n = [0,1]^n$  com  $n \geq 2$ , pode ser escrita na forma (2.2-1),

$$f(x_1, \dots, x_n) = \sum_{i=1}^{2n+1} \chi_i \left[ \sum_{j=1}^n \phi_{ji}(x_j) \right] \quad (2.2-1)$$

em que  $\chi_i$  e  $\phi_{ji}$  representam funções contínuas de apenas uma variável. As funções  $\phi_{ji}$  são independentes de  $f(\cdot)$ , sendo no entanto as funções  $\chi_i$  específicas para  $f(\cdot)$ .

O teorema de Kolmogorov enuncia que qualquer função contínua de  $n$  variáveis,  $f(x_1, \dots, x_n)$ , pode ser descrita exactamente pela sobreposição de funções contínuas de apenas uma variável e uma simples operação, neste caso a adição. Posteriormente, Sprecher (1965) substituiu as funções  $\phi_{ji}$  por  $v_j \phi_i$ , em que  $v_j$  e  $\phi_i$  representam, respectivamente, termos constantes e funções monótonas crescentes. Mais tarde Lorentz (1976), reformulou o teorema de Kolmogorov substituindo as funções  $\chi_i$  por uma única função  $\chi$ , e a equação (2.2-1) é reformulada por (2.2-2).

$$f(x_1, \dots, x_n) = \sum_{i=1}^{2n+1} \chi \left[ \sum_{j=1}^n v_j \phi_i(x_j) \right] \quad (2.2-2)$$

Foi Hecht-Nielsen (1987) quem primeiro estabeleceu uma relação entre o teorema de Kolmogorov, usando a modificação proposta por Lorentz, e as redes neuronais. Hecht-Nielsen constatou que a equação (2.2-2) descreve na verdade uma rede neuronal, constituída por duas camadas internas, tal como se mostra na Figura 2.2-1, em que os pesos das ligações entre a primeira e segunda camada são unitários, o mesmo acontecendo para os pesos da terceira para a quarta camada.

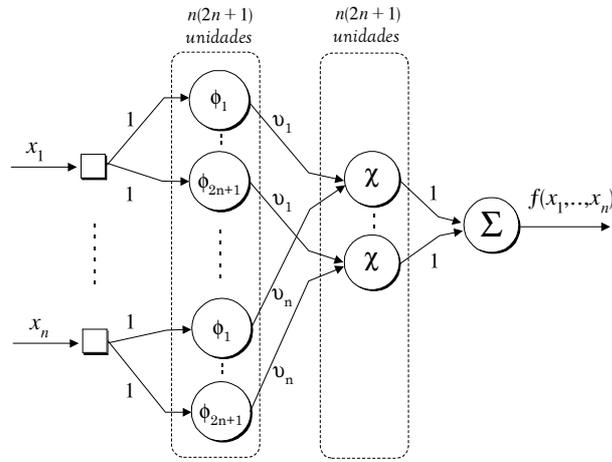


Figura 2.2-1: Relação entre o teorema de Kolmogorov (reformulado por Lorentz) e as redes neurais.

Contudo, o facto de as funções  $v_j$ ,  $\phi_i$  e  $\chi$  serem fortemente não suaves, e os termos  $\chi$  serem dependentes da função a aproximar, conduziu ao criticismo destes resultados. Girosi e Poggio (1989), defensores deste criticismo, sustentaram a ideia de que o teorema de Kolmogorov era irrelevante, no contexto das redes neurais. Tais pressupostos foram contrariados por Kurkova (1991), ao provar através do seguinte teorema que os resultados de Kolmogorov eram na verdade relevantes no âmbito das redes neurais.

**Teorema (Kurkova, 1991)**

Seja  $n \in \mathbb{N}$  com  $n \geq 2$  e  $\sigma$  uma função do tipo sigmoidal. Seja  $f$  a função a aproximar, verificando  $f \in C^0(E^n)$ , isto é, diferenciável em  $E^n$ . Nestas condições existe um inteiro  $M \in \mathbb{N}$ , uma constante  $\varepsilon \in \mathfrak{R}^+$  e funções  $\chi_i, \phi_{ji} \in S(\sigma)$  tais que para  $(x_1, \dots, x_n) \in E^n$  se verifica **Error! Reference source not found.**

$$\left\| f(x_1, \dots, x_n) - \sum_{i=1}^M \chi_i \left( \sum_{j=1}^n \phi_{ji}(x_j) \right) \right\| < \varepsilon \tag{2.2-3}$$

$S(\sigma)$  representa o conjunto de todas as funções da forma  $\phi(x) = \sum_{i=1}^M a_i \sigma(b_i x + c_i)$ , em que

$a_i, b_i$  e  $c_i$  são parâmetros reais.

Este resultado implica que qualquer função contínua pode ser aproximada, com precisão arbitrária, por uma rede MLNN do tipo sigmoidal de quatro camadas. No entanto, comparando (2.2-2) com **Error! Reference source not found.** constata-se que a representação exacta original no resultado de Kolmogorov é substituída, neste último teorema, por uma representação aproximada.

## As redes MLNN e o Teorema de Stone-Weierstrass

O teorema original de Weierstrass mostra que uma qualquer função contínua pode ser aproximada por uma sequência de polinómios com uma precisão arbitrária. O teorema de Weierstrass foi reformulado por Stone (1948) estendendo os resultados a funções genéricas, não necessariamente polinomiais, estabelecendo o conhecido teorema de Stone-Weierstrass.

Sendo assim, o teorema de Stone-Weierstrass apresenta-se como uma alternativa ao teorema de Kolmogorov, como ferramenta de base na demonstração das propriedades de aproximação das redes neuronais MLNN.

Cybenko (1989) foi o primeiro a demonstrar rigorosamente, servindo-se do teorema de Stone-Weierstrass, que uma rede com apenas uma camada interna, com funções de activação do tipo sigmoideal constitui um aproximador universal. Além deste, outros dois trabalhos foram independentemente publicados em 1989: o trabalho de Funahashi (1989) e o trabalho de Hornik *et al.* (1989), generalizados no seguinte teorema.

### Teorema (Funahashi, 1989)

Seja  $\sigma(\cdot)$  uma função contínua, limitada e monótona crescente. Seja  $D$  um subconjunto compacto de  $\mathfrak{R}^n$  e  $f(x_1, \dots, x_n)$  uma função real e contínua em  $D$ . Então, para qualquer  $\varepsilon > 0$ , existe um inteiro  $M$  e um conjunto de constantes reais  $\alpha_i$ ,  $b_i$  e  $w_{ij}$  tais que:

$$\left\| f(x_1, \dots, x_n) - \mathcal{N}(x_1, \dots, x_n) \right\| < \varepsilon \quad (2.2-4)$$

$\mathcal{N}(x_1, \dots, x_n)$  representa uma rede neuronal definida por (2.2-5).

$$\mathcal{N}(x_1, \dots, x_n) = \sum_{i=1}^M \alpha_i \sigma \left( \sum_{j=1}^n w_{ij} x_j + b_i \right) \quad (2.2-5)$$

A rede neuronal definida em (2.2-5) possui  $n$  entradas e apenas uma camada interna, constituída por  $M$  neurónios. De facto assim definida, a saída de cada um dos neurónios  $i$  resulta da afectação por um determinado peso  $w_{ij}$  de cada uma das entradas  $x_j$  e de um termo de polarização,  $b_i$ . O valor da saída da rede resulta por sua vez da combinação linear da saída dos neurónios de cada camada interna, ponderadas por um peso  $\alpha_i$ . Note-se que se estes resultados fornecem uma base sólida para a modelização de sistemas não lineares não permitem obter uma estimativa para o número de neurónios necessários a utilizar nem como devem ser obtidos os pesos que a caracterizam.

Refira-se ainda que as propriedades de aproximação universal de redes de três camadas, contendo na sua camada interna funções de activação do tipo sigmoideal, não pode ser obtida directamente usando o teorema de Stone-Weierstrass. Para tal é

necessário recorrer a conceitos de análise funcional. Uma classe de redes neuronais que satisfaz directamente o teorema de Stone-Weierstrass é a rede de base radial, da forma (2.2-6).

$$y_n = \mathcal{N}(x) = \sum_{i=1}^M w_i \Psi(\|x - c_i\|) \quad (2.2-6)$$

As variáveis  $x \in \mathfrak{R}^n$  e  $y_n \in \mathfrak{R}$  representam respectivamente a entrada e a saída da rede, admitindo neste caso que se tem apenas uma saída. A rede é constituída por uma camada interna, de  $M$  neurónios, sendo cada neurónio caracterizado por  $\Psi(\cdot)$ , uma função não linear do tipo radial. Os parâmetros  $c_i \in \mathfrak{R}^n$  representam os centros de cada uma das funções de base radial e  $\|\cdot\|$  define uma norma, usualmente Euclidiana. As constantes  $w_i$  definem os pesos da ligação entre a saída de cada um dos neurónios da camada interna e a saída da rede. Um exemplo de uma função que obedece a estas características é a função Gaussiana, Tabela 2.1-1. Basicamente uma rede *RBFFNN* é uma rede *MLNN* de três camadas, com funções de activação Gaussianas na camada interna, como se mostra na Figura 2.2-2.

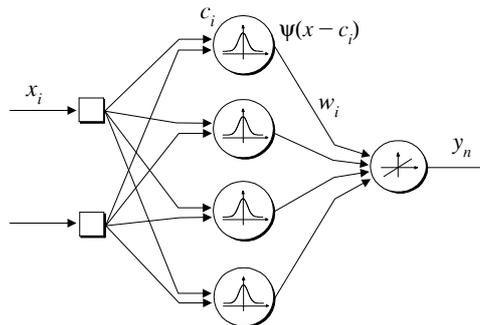


Figura 2.2-2: Rede neuronal RBFNN.

## Redes MLNN versus RBFNN

Quando se pensa seleccionar uma de entre as topologias referidas, redes *MLNN* ou *RBFNN*, há que ter em conta alguns aspectos. Em primeiro lugar, ambas permitem a aproximação de uma qualquer função não linear. A grande diferença entre ambas reside no tipo da função de activação utilizada, o que lhes confere características distintas. Se, por um lado, pela sua forma, as *RBFNN* permitem obter saídas não nulas apenas para valores de entrada relativamente próximos dos centros das funções Gaussianas, as funções do tipo sigmoideal permitem obter valores de saída não nulos em regiões bastante mais vastas das variáveis de entrada. Por esta razão, as redes do tipo sigmoideal permitem construir mapeamentos globais sendo por isso capazes, em princípio, de efectuar

generalizações em regiões do espaço de entrada em que pouco ou nenhum conhecimento existe. Em contraste, as redes *RBNN* constroem mapeamentos locais, pelo que a sua capacidade de generalização é mais limitada.

Por outro lado, devido às suas propriedades de localidade, as redes *RBNN* permitem, em princípio, um treino mais rápido uma vez que este é local (o treino numa dada região do espaço de entrada não afecta o treino já efectuado noutra região). A equivalência entre este tipo de redes e alguns modelos difusos (Takagi-Sugeno) é outra das vantagens, possibilitando assim a transparência do modelo neuronal.

Além das características já apontadas, as *RBNN* sofrem do problema da maldição da dimensionalidade (*curse of dimensionality*) o que significa que o número de neurónios existentes na unidade interna aumenta exponencialmente com a dimensão do espaço de entrada, ou seja, para os mesmos resultados, e em termos de aproximação, as *RBNN* necessitam de possuir mais neurónios comparativamente com as redes *MLNN*.

Tendo em atenção as considerações referidas optou-se nesta dissertação por utilizar redes com funções de activação do tipo sigmoidal. E, de entre estas, optou-se pela tangente hiperbólica pela razão principal de que, segundo alguns autores (Haykin, 1994), as funções assimétricas (ímpares) permitem que o treino seja mais rápido, Tabela 2.1-1 e Figura 2.1-3).

## 2.2.2 Redes Dinâmicas a partir de Redes Estáticas

As redes neuronais referidas até ao momento têm em comum o facto de serem estáticas. No entanto, um dos objectivos desta dissertação visa o uso de redes neuronais na identificação de processos, surgindo portanto a necessidade de estender a aplicação de redes neuronais a sistemas dinâmicos. Por esta razão, é agora abordado o problema da incorporação de dinâmica em modelos neuronais, quer considerando estruturas específicas para o efeito, quer a partir das estruturas anteriores (estáticas). Assim sendo consideram-se:

- i) redes recorrentes obtidas à custa da modificação das redes *MLNN*, distinguindo-se duas formas de o conseguir: recorrência externa e ligações dinâmicas;
- ii) redes neuronais recorrentes propriamente ditas *RNN*<sup>2</sup>, a serem estudadas mais em pormenor na secção 2.2.3. Referem-se aquelas em que a dinâmica é introduzida ao nível do neurónio (neurónios dinâmicos) ou em que é considerada realimentação nas unidades internas.

---

<sup>2</sup> *RNN*- Recurrent Neural Networks

## Redes MLNN com Recorrência Externa

Uma das primeiras arquitecturas capaz de lidar com informação proveniente de sistemas dinâmicos foi a WIT<sup>3</sup>, proposta independentemente por Lapedes e Farber (1987) e por Lang *et al.* (1990), onde é usado o conceito de janela temporal. De forma a incorporar o factor temporal em redes neuronais estáticas é usado um atraso unitário,  $q^{-1}$ , ou uma série de atrasos unitários, permitindo considerar as saídas em instantes anteriores tal como uma entrada multi-variável estática.

Uma variação desta arquitectura consiste em usar duas janelas temporais: uma para os dados de saída (tal como em estruturas WIT) e outra para os dados da entrada, como ilustrado na **Error! Reference source not found.**

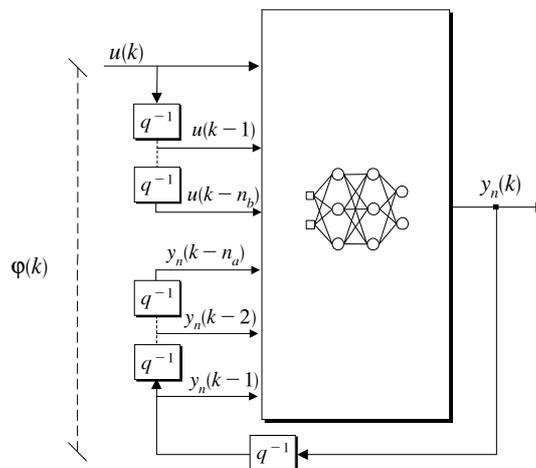


Figura 2.2-3: Rede MLNN com recorrência externa.

Esta estrutura foi proposta por Narendra e Parthasarathy (1990) e ficou conhecida por TDLNN<sup>4</sup>. A saída da rede,  $y_n(k)$ , é determinada por (2.2-7),

$$\begin{aligned} y_n(k) &= f(y_n(k-1), \dots, y_n(k-n_a), u(k-1), \dots, u(k-n_b)) \\ &= f(Y_n(k-1), U(k-1)) \end{aligned} \quad (2.2-7)$$

em que as variáveis de regressão  $U(k-1) = [u(k-1), \dots, u(k-n_b)]$  e  $Y_n(k-1) = [y_n(k-1), \dots, y_n(k-n_a)]$  são constituídas por valores passados das entradas e saídas. As constantes  $n_b$  e  $n_a$  definem, respectivamente, a ordem da memória a usar quer na entrada, quer na saída e  $f: \mathfrak{R}^{n_a + n_b} \rightarrow \mathfrak{R}^{n_y}$  é uma função não linear.

<sup>3</sup> WIT- Window in Time Network

<sup>4</sup> TDLNN- Time DeLay Neural Network

Pela analogia deste tipo de modelo com o modelo de resposta impulsional infinita, *IIR*<sup>5</sup>, este tipo de topologia é também conhecido por rede neuronal *IIR*. Da mesma forma, a arquitectura *WIT* é conhecida por rede neuronal de resposta impulsional finita, *FIR*<sup>6</sup>.

Note-se que esta rede não é propriamente recorrente uma vez que não há de verdade recorrência nos elementos que constituem a rede mas apenas uma recorrência externa, razão pela qual se poderá dizer que esta é uma rede de recorrência externa. No entanto, a arquitectura *TDLNN* fornece uma forma simples de incorporar dinâmica, permitindo, deste modo, a uma rede estática reproduzir sistemas dinâmicos.

No modelo acima descrito, (2.2-7), os valores da saída da rede são realimentados para a entrada da própria rede. Diz-se, neste caso, que o modo de operação é em paralelo (Narendra e Parthasarathy, 1990). Em oposição, no caso série-paralelo, considera-se que são utilizados valores passados do próprio processo como entradas na rede. Neste último caso a saída da rede seria descrita por (2.2-8).

$$y_n(k) = f(y(k-1), \dots, y(k-n_a), u(k-1), \dots, u(k-n_b)) \quad (2.2-8)$$

Se o modo série-paralelo tem o inconveniente de gerar modelos com a capacidade de prever apenas o próximo valor da variável de saída, o modelo paralelo permite fazer previsões para valores futuros, que não apenas para o próximo valor. No entanto, o processo de estimação é, neste último caso, geralmente mais lento e, regra geral, conduz a modelos menos perfeitos, tendo em conta os dados a aproximar.

## Ligações Dinâmicas

Uma alternativa às redes *TDLNN*, tendo em vista a incorporação de dinâmica, consiste na modificação das conexões entre os neurónios. Uma representação possível será considerar as ligações não como sendo estáticas, tal como nas *TDLNN*, mas como constituindo sistemas dinâmicos. Filtros de resposta impulsional finita ou de resposta impulsional infinita são exemplos de sistemas dinâmicos lineares sugeridos por Tsoi e Back (1992). Por sua vez, DeVries e Príncipe (1992) sugeriram o uso de filtros *Gamma*. Um filtro *Gamma* de ordem  $n$  é uma estrutura com uma entrada,  $u(k)$ , e  $n$  saídas,  $x_i(k)$   $i = 1, \dots, n$ , como se mostra na Figura 2.2-4.

---

<sup>5</sup> *IIR- Infinite Impulse Response*

<sup>6</sup> *FIR- Finite Impulse Response*

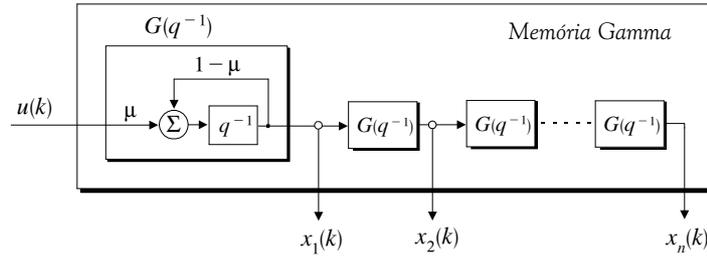


Figura 2.2-4: Memória Gamma de ordem n.

Formalmente consiste numa série de sistemas de primeira ordem cuja resposta a um impulso para a saída  $x_i(k)$  é definida de uma forma recursiva, por (2.2-9)

$$x_i(k) = (1 - \mu) x_i(k-1) + \mu x_{i-1}(k-1) \quad (2.2-9)$$

com

$$x_0(k) = u(k)$$

A variável  $\mu \in ]0,1[$  define uma constante de tempo, que controla a velocidade a que a informação é esquecida. Se  $\mu \approx 0$ , o sistema reduz-se a um atraso unitário,  $q^{-1}$ , enquanto que o valor de  $\mu \approx 1$  permite obter uma memória de longa duração.

## 2.2.3 Estruturas Recorrentes

### Neurónios Dinâmicos

Gupta e Rao (1993), propuseram o uso de neurónios dinâmicos, *DNU*<sup>7</sup>, isto é, a introdução explícita de memória em cada um dos neurónios. Cada neurónio possui elementos de atraso, sendo a saída actual função de valores passados das entradas e saídas. Para o exemplo concreto de um sistema linear, de segunda ordem, define-se a equação que descreve a dinâmica de cada neurónio por (2.2-10), e mostra-se esquematicamente na Figura 2.2-5 essa representação.

$$a(k) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} u(k) \quad (2.2-10a)$$

$$y(k) = \sigma(a(k)) \quad (2.2-10b)$$

<sup>7</sup> DNU- Dynamic Neural Unit

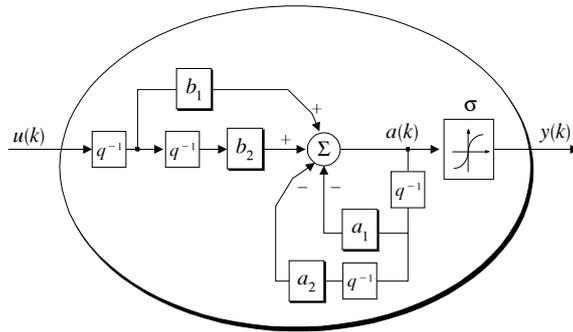


Figura 2.2-5: Neurónio dinâmico.

## Redes Neurais Recorrentes: Recorrência Interna

As redes neuronais com dinâmica interna são conceptualmente diferentes das que têm recorrência externa, uma vez que a sua estrutura incorpora explicitamente dinâmica, introduzida pela realimentação existente ao nível dos neurónios internos. A presença de dinâmica permite conseguir modelos mais complexos, proporcionando, em princípio, vantagens adicionais em relação a uma rede estática com recorrência externa.

Esta propriedade sugere que as redes recorrentes devem ser preferencialmente utilizadas na modelização de sistemas dinâmicos. Analisando o problema sob outra perspectiva, nas redes com recorrência externa a variável tempo é processada como uma transformação do tempo em espaço. A introdução do tempo é assim feita de uma forma explícita ao nível da estrutura. Nas redes recorrentes propriamente ditas a variável tempo é introduzida ao nível do neurónio e de uma forma implícita.

Em diante, sempre que se designar uma rede como sendo recorrente subentende-se uma com recorrência interna.

A ideia da introdução de recorrência interna em redes neuronais deve-se a Hopfield (1984). A estrutura por ele proposta consiste numa rede com apenas uma camada, que inclui realimentação dos valores da saída para a entrada, com um atraso unitário, ponderados por um determinado valor, como descrito na equação (2.2-11) e mostrada na Figura 2.2-6, para o caso particular de três neurónios.

$$x_i(k+1) = \sigma \left( \sum_{j=1}^n w_{ij} x_j(k) \right) \quad (2.2-11a)$$

$$x(0) = x_0 \quad (2.2-11b)$$

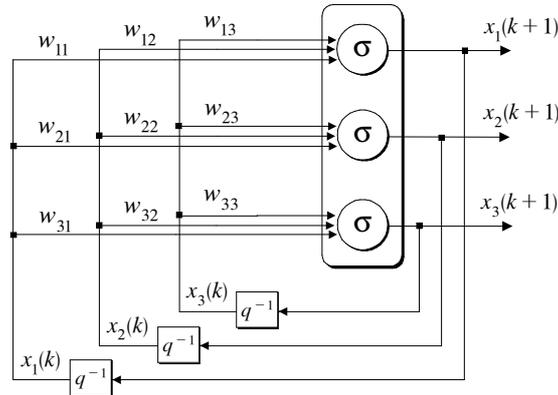


Figura 2.2-6: Rede de Hopfield.

A equação anterior pode ser interpretada como um sistema dinâmico discreto, descrito no espaço de estados. Cada uma das saídas  $x_i(k)$ ,  $i = 1, \dots, n$ , representa o estado da rede neuronal no instante  $k$  e  $\sigma(\cdot)$  é uma função de activação, binária no caso particular da rede de Hopfield. A equação pode ser descrita em termos matriciais por (2.2-12).

$$x(k+1) = \sigma(Wx(k)) \quad (2.2-12)$$

A variável  $x \in \mathfrak{R}^n$  define o vector de estados e  $W \in \mathfrak{R}^{n \times n}$  a matriz de pesos. A partir de um dado valor inicial  $x_0 = x(0)$ , o sistema evolui e, assumindo que é estável, atinge um ponto de equilíbrio. Hopfield demonstrou em que condições se poderia assegurar a estabilidade da sua rede, no teorema de estabilidade de Hopfield (1982), posteriormente revisto por Cohen e Grossberg (1983).

Note-se que não são consideradas entradas externas. Por esta razão a rede de Hopfield pode ser utilizada para devolver um determinado padrão previamente armazenado em memória, em resposta à representação de um padrão incompleto ou corrompido por ruído, razão pela qual este tipo de estrutura é interpretado como uma memória associativa.

Na sequência do trabalho de Hopfield surgiram então outras soluções, visando a descrição de sistema dinâmicos. Uma delas, à custa do uso de unidades adicionais, designadas de contexto, foi investigada por Jordan (1986), Elman (1990), Pollack (1990), Robinson e Fallside (1991) e Williams e Ziepsner (1995).

Elman, em particular, propôs uma rede parcialmente recorrente, denominada *Simple Recurrent Network*, em que as ligações para a frente eram possíveis de treinar sendo, no entanto, fixas as ligações de realimentação. Adicionalmente às unidades de entrada  $u \in \mathfrak{R}^{n_u}$ , de saída  $y \in \mathfrak{R}^{n_y}$  e internas  $x^h \in \mathfrak{R}^n$ , uma rede de Elman possui uma unidade de contexto  $x^c \in \mathfrak{R}^n$ , unidade esta que memoriza o valor da unidade interna no instante anterior. Apesar de teoricamente uma rede de Elman ter a capacidade de representar um

sistema não linear de ordem  $n$ , têm sido sugeridas algumas modificações de forma a melhorar as suas capacidades de aprendizagem. Pham e Xing (1995a) e (1995b) propuseram o uso de realimentação adicional na unidade de contexto de cada neurónio para si próprio, à qual se encontra associado um peso fixo  $\alpha \in \mathfrak{R}^+$ . A equação (2.2-13) descreve a rede de Elman modificada, mostrada na Figura 2.2-7, considerando dois neurónios.

$$x^h(k+1) = \sigma( W^x x^c(k+1) + W^u u(k) ) \quad (2.2-13a)$$

$$x^c(k+1) = x^h(k) + \alpha x^c(k) \quad (2.2-13b)$$

$$y_n(k) = W^y x^h(k) \quad (2.2-13c)$$

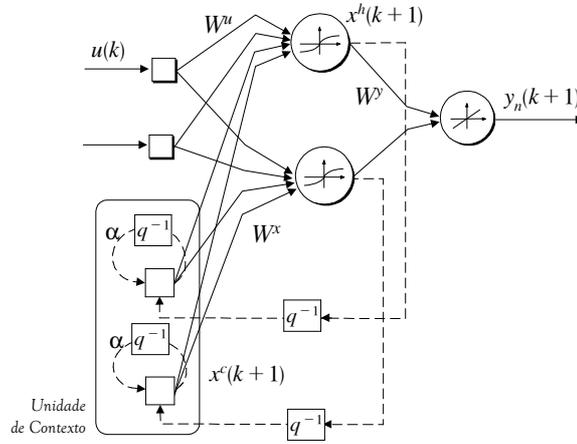


Figura 2.2-7: Rede de Elman modificada.

As matrizes de ligação  $W^x \in \mathfrak{R}^{n,n}$ ,  $W^u \in \mathfrak{R}^{n,n_u}$  e  $W^y \in \mathfrak{R}^{n_y,n}$  definem os pesos da rede. Este tipo de rede, tal como a de Hopfield, considera uma recorrência que se designa por global, dado que a saída de um neurónio da camada interna pode ser realimentada para outro qualquer neurónio na mesma camada. Por outro lado, nas redes ditas de recorrência local, propostas por Back e Tsoi (1991) e Frasconi *et al.* (1992), a realimentação é efectuada, como o próprio nome indica, localmente, constituindo portanto um caso particular das redes de recorrência global. Existe apenas uma realimentação entre cada neurónio da camada interna e ele próprio, como se descreve na equação (2.2-14) e se mostra na Figura 2.2-8, para o caso de uma rede com três neurónios.

$$x_j(k+1) = \sigma( w_j^x x_j(k) + \sum_{i=1}^{n_u} w_{ji}^u u_i(k) ) \quad j = 1, \dots, n \quad (2.2-14a)$$

$$y_n(k) = \sum_{i=1}^n w_j^y x_j(k) \quad (2.2-14b)$$

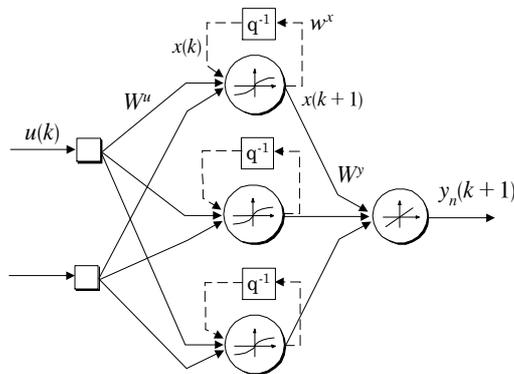


Figura 2.2-8: Rede com recorrência local.

Se as redes com recorrência global, casos de Hopfield e Elman, são aproximadores universais (Kremer, 1995), (Jin *et al.*, 1995a), (Lixin, 1999), tal facto não se verifica para as redes locais (Frasconi e Gori, 1996). Contudo, relativamente à complexidade dos métodos de treino, estes são comparativamente mais fáceis de implementar as arquitecturas locais, uma das razões que motiva a sua aplicação prática (Campolucci *et al.*, 1997).

## Classificação de Estruturas Recorrentes

Durante a última década diversas estruturas recorrentes têm vindo a ser propostas, pelo que não seria apropriado referi-las exaustivamente. Têm no entanto surgido algumas tentativas de as sistematizar e classificar, de que são exemplos os trabalhos de Narendra e Parthasarathy (1990), Nerrand *et al.* (1993), Horne e Giles (1995), Tsoi e Back (1997), Moreau (1998), e Kremer, (1999a) e (1999b).

O trabalho pioneiro neste campo foi efectuado por Narendra e Parthasarathy (1990), em que consideraram quatro modelos I, II, III e IV de redes recorrentes (note-se que na verdade estes são modelos de recorrência externa). Nerrand e seus colaboradores, (Nerrand *et al.*, 1993), propuseram uma forma canónica para a representação de redes recorrentes, que consistia em dois subsistemas: um constituído somente por elementos de atraso e outro constituído por elementos não lineares (os neurónios) e os pesos das ligações (estáticos). Apesar de defenderem a ideia de que qualquer rede recorrente poderia ser transformada neste tipo de representação canónica, não apresentaram no entanto um método explícito que o permitisse fazer.

Horne e Giles (1995), propuseram uma classificação baseada na observabilidade dos estados. Dependendo da sua acessibilidade consideraram duas subdivisões: uma em que os estados da rede são constituídos por entradas e saídas, portanto possíveis de observar,

o mesmo é dizer, possível de medir; outra em que os estados da rede, definidos pelas saídas da unidade interna, não são acessíveis. A primeira classe de redes inclui, por exemplo, as redes *TDLNN*, enquanto a segunda inclui as redes de Elman. Por sua vez a segunda classe de redes era ainda subdividida em redes com recorrência local e redes com recorrência global.

Um trabalho de referência neste domínio é o de Tsoi e Back (1997). Propõem e classificam de uma forma sistemática a grande maioria das redes recorrentes existentes, apresentando mesmo algumas estruturas unificadoras. Concretamente, desenvolveram uma classificação tendo em conta vários aspectos: sentido das ligações, podendo ser para a frente (*feedforward*) ou de realimentação (*feedback*); tipo de ligações, distinguindo ligações estáticas e dinâmicas; tipo de recorrência, considerando recorrência local, não local e global.

Moreau (1998) interpretou as redes recorrentes como sistemas dinâmicos, descritos no espaço de estados, onde a saída de cada neurónio da camada interna,  $x \in \mathfrak{R}^n$ , pode ser entendido com um estado do sistema. Assim, em função dos elementos que participam na função de activação de cada neurónio,  $\sigma(\cdot)$ , Moreau considerou os seguintes tipos de redes recorrentes:

i) *Rede de Hopfield genérica:*

$$x(k+1) = A x(k) + W^x \sigma(x(k)) \quad (2.2-15)$$

ii) *Perceptrão dinâmico*

$$x(k+1) = \sigma(W^x x(k) + b_0) \quad (2.2-16)$$

iii) *Perceptrão dinâmico com recorrência local*

$$x(k+1) = A x(k) + \sigma(W^x x(k) + b_0) \quad (2.2-17)$$

iv) *Perceptrão dinâmico com uma camada interna*

$$x(k+1) = H \sigma(W^x x(k) + b_0) \quad (2.2-18)$$

Nas equações (2.2-15)-(2.2-18)  $A \in \mathfrak{R}^{n,n}$  define uma matriz diagonal (de valores próprios localizados no interior do círculo de raio unitário),  $W^x \in \mathfrak{R}^{n,n}$  e  $H \in \mathfrak{R}^{n,n}$  são matrizes de ligação e  $b_0 \in \mathfrak{R}^n$  é um vector de polarização (*bias*).

Mais recentemente Kremer (1999a) e (1999b) sugeriu uma teoria comum para descrever modelos neuronais que incorporem a variável temporal. A sua classificação é baseada em quatro aspectos: a forma como o estado é calculado, a forma como a saída é calculada, a forma como é possível introduzir informação na rede e a forma como os seus parâmetros podem ser treinados. Note-se no entanto que Kremer estende a classificação de estruturas e entende o problema já do ponto de vista de inicialização e treino de parâmetros.

## 2.2.4 Estruturas Neurais *Affine*

Referiram-se as estruturas de Hopfield, Elman e as de recorrência local como exemplos de estruturas recorrentes. Se fosse objectivo desta dissertação tratar apenas o problema da modelização, a descrição de possíveis estruturas estaria certamente incompleta e seria, como tal, necessário citar outros trabalhos nesta área. No entanto, uma vez que o objectivo não é a modelização em si mas a modelização como um instrumento base para efectuar o projecto de um controlador, não se justifica uma revisão exhaustiva das estruturas neuronais, apenas como ferramentas de modelização. Ou seja, o problema da definição da estrutura não se limita a desenvolver apenas uma arquitectura que tenha determinadas capacidades de aproximação, a ser utilizada na modelização de um sistema dinâmico. O problema vai mais além, tem de se pensar desde logo em como efectuar de uma forma eficaz o seu treino e na sua utilidade para o projecto de um controlador.

Assim, interessa principalmente analisar as redes sob o ponto de vista da sua incorporação em estruturas de controlo, no caso presente num esquema de controlo adaptativo indirecto. Na verdade, da definição da sua estrutura irão depender vários aspectos:

- i) *projecto do controlador*: uma vez que se segue uma metodologia de controlo adaptativo indirecto, em que os parâmetros do controlador são determinados a partir dos do modelo neuronal, resulta daqui que a facilidade do cálculo dos parâmetros do controlador dependerá de uma forma decisiva do tipo de estrutura a considerar para o modelo;
- ii) *estimação de parâmetros*: a complexidade da estrutura a propor será determinante no desenvolvimento de algoritmos eficientes, que permitam efectuar a determinação de parâmetros dessa mesma estrutura, especialmente quando implementados em-linha;
- iii) *análise de propriedades básicas*: o estudo de propriedades tais como estabilidade e convergência, estará fortemente relacionado com a complexidade da estrutura considerada.

De entre as estruturas que seguem esta filosofia, isto é, a relativa facilidade de aplicação para o projecto de controladores e de desenvolvimento de metodologias de estimação de parâmetros, as estruturas neuronais *affines* constituem, sem dúvida, uma classe a destacar.

### Sistemas *Affine*

De uma forma genérica, o termo *affine*, aplicado a uma função dependente de várias variáveis, expressa uma relação de linearidade para uma delas.

No caso do sistema descrito em **Error! Reference source not found.**, o termo *affine* expressa uma relação de linearidade entre a acção de controlo  $u(k)$  e o estado  $x(k)$ .

$$\begin{aligned}x(k+1) &= f(x(k)) + g(x(k)) u(k) \\ y(k) &= h(x(k))\end{aligned}\tag{2.2-19}$$

Como casos particulares de **Error! Reference source not found.** referem-se a representação entrada-saída e a representação no espaço de estados (segundo uma forma normal).

No primeiro caso o sistema é representado por uma equação do tipo,

$$y(k+1) = f(x(k)) + g(x(k)) u(k)\tag{2.2-20}$$

no segundo por

$$\begin{aligned}x_1(k+1) &= x_2(k) \\ &\vdots \\ x_{n-1}(k+1) &= x_n(k) \\ x_n(k+1) &= f(x(k)) + g(x(k)) u(k)\end{aligned}\tag{2.2-21}$$

### Estruturas Neurais

Em relação a estruturas neuronais que verificam a relação *affine* referem-se, no caso da descrição entrada-saída, os trabalhos de Narendra e Parthasarathy (1990), Chen e Khalil (1992) e (1995), TeBraake *et al.* (1999), Botto *et al.* (1999). Seguindo a descrição entrada-saída o sistema (2.2-20) é substituído por

$$y_n(k+1) = \mathcal{N}_1(x(k)) + \mathcal{N}_2(x(k)) u(k)\tag{2.2-22}$$

em que  $\mathcal{N}_1(\cdot)$  e  $\mathcal{N}_2(\cdot)$  definem redes neuronais, usualmente do tipo *MLNN* ou *RBFNN* e  $x(k)$  é um vector constituído por informação passada de entrada-saída.

No segundo caso, usando uma descrição no espaço de estados, referem-se os trabalhos de Jin *et al.* (1993), Jagannathan e Lewis (1996a) e Jagannathan *et al.* (1996b).

$$\begin{aligned}x_1(k+1) &= x_2(k) \\ &\vdots \\ x_{n-1}(k+1) &= x_n(k) \\ x_n(k+1) &= \mathcal{N}_1(x(k)) + \mathcal{N}_2(x(k)) u(k)\end{aligned}\tag{2.2-23}$$

Da mesma forma que na representação entrada-saída as funções  $f(x(k))$  e  $g(x(k))$  são substituídas por redes neuronais  $\mathcal{N}_1(\cdot)$  e  $\mathcal{N}_2(\cdot)$ .

## 2.3

# Estimação de Parâmetros em Redes Recorrentes

*Uma vez seleccionada uma determinada estrutura para uma rede neuronal há que determinar os seus parâmetros, isto é, os pesos entre as suas ligações. Nesta secção formula-se o problema e apresentam-se métodos específicos desenvolvidos para o efeito.*

*Numa segunda abordagem entende-se uma rede como uma qualquer transformação não linear e a estimação como um problema genérico de optimização não linear. Por fim, entendida como um sistema não linear, descrito no espaço de estados, considera-se a possibilidade de utilização de técnicas de observação para o problema de estimação dos seus parâmetros.*

### 2.3.1 Generalidades

#### Métodos Em-Diferido e Métodos Em-Linha

A estimação de parâmetros é, no contexto de redes neuronais, sinónimo de treino ou aprendizagem dos pesos da rede. Os métodos de treino (ou de estimação) capazes de determinar os parâmetros, isto é, os pesos de uma rede, podem ser subdivididos em duas categorias: métodos em-linha (*on-line*) e métodos em-diferido (*off-line*). Nestes últimos, também conhecidos por métodos em *batch*, é inicialmente gerado um conjunto de dados de treino e a determinação dos pesos da rede é efectuada de uma só vez, com base em todo esse conjunto de dados. Nos primeiros é usada, em cada instante, informação para actualizar os pesos da rede.

Nos métodos em-diferido, além da necessidade de se considerar *a priori* todo o conjunto de dados de treino, estes apresentam outras desvantagens. Por um lado, de forma a permitirem a representação adequada de uma grande quantidade de dados, pode tornar-se necessário considerar uma rede com um número elevado de neurónios. Deve ter-se em conta que uma rede é um aproximador universal, mas quantos neurónios são

necessários para que o erro seja o desejado? Por outro lado, será pouco provável que o conjunto de dados de treino disponível reflecta todas as situações possíveis e todas as regiões de operação. Daqui surge outro problema: como seleccionar convenientemente o conjunto de dados de treino? E, mesmo assumindo que para o conjunto de dados existente se fez uma estimação aceitável, caso o processo sofra variações na sua estrutura não se poderá garantir que o modelo represente adequadamente o processo.

Os métodos em-linha eliminam por um lado as hipóteses restritivas referidas para os métodos em-diferido mas originam, por outro, novos problemas. Uma vez que tendem a efectuar uma minimização instantânea, podem conduzir a uma degradação da qualidade do modelo. Como se pode garantir a qualidade dos modelos? Mais ainda, como assegurar a estabilidade e convergência do processo adaptativo de estimação?

Nesta dissertação opta-se por se considerar um treino prévio dos parâmetros da rede, a partir da informação (dados) inicialmente disponível. Depois, quando em operação, procede-se à actualização em-linha dos parâmetros da rede. A opção por cada um dos métodos particulares a utilizar em cada uma das fases é justificada no final desta secção, após a descrição dos vários métodos.

## Métodos de Gradiente

O objectivo do treino, ou aprendizagem de uma rede, consiste, à semelhança de um qualquer problema de estimação, secção 2.1.1, em determinar os pesos  $W$  da rede, a partir de um conjunto  $Z^N$  de dados de treino, de forma a minimizar um determinado critério de custo  $J$ , do tipo SSE, definido em (2.1-12).

A variável  $e_y \in \mathfrak{R}^{n_y}$  dada por (2.3-1),

$$e_y(k) = y(k) - y_n(k, W) \quad (2.3-1)$$

é designada por erro de identificação, e é definida como a diferença entre a saída real do processo a identificar,  $y(k)$ , e a saída do modelo neuronal,  $y_n(k, W)$ . Devido à não linearidade do problema não é possível obter, na maioria das vezes, uma solução analítica. Na prática usa-se um método iterativo, com os pesos a serem obtidos por (2.3-2), admitindo que se conhece uma estimativa inicial para os parâmetros,  $W(0) = W_0$ .

$$W(k+1) = W(k) + \eta \Xi(W) \quad (2.3-2)$$

É usual lidar com o incremento ou variação dos parâmetros, definido por (2.3-3).

$$\Delta W(k+1) = W(k+1) - W(k) = \eta \Xi(W) \quad (2.3-3)$$

$\Xi(W)$  define a direcção de procura e  $\eta$  define uma constante positiva ou, em alternativa, uma matriz positiva definida, designada por ganho ou, no contexto de redes neuronais, por coeficiente de aprendizagem. Os métodos de estimação distinguem-se

fundamentalmente pela forma como é especificado o valor de  $\Xi(W)$ , determinado a partir da informação adquirida até ao instante em questão.

O método de gradiente, em que os pesos da rede são adaptados segundo a direcção oposta à do gradiente, (2.3-4), é um dos mais conhecidos.

$$\Xi(W) = G = - \frac{\partial J(W)}{\partial W} = - \nabla J(W) \quad (2.3-4)$$

Neste caso a direcção de procura  $\Xi(W)$  é definida pela matriz  $G$ , o *Jacobiano*.

### Extensão dos Métodos de Gradiente a Redes MLNN

Seguindo a formulação de Rumelhart *et al.* (1986), considere-se uma rede com  $L$  camadas, ( $L - 2$  camadas internas, uma de entrada e uma de saída) tal como descrito em (2.1-17). Dado um conjunto de  $N$  padrões de treino, em que  $p = 1, \dots, N$  define o índice de cada um deles e sejam os respectivos valores desejados para a saída,  $y_p \in \mathfrak{R}^{n_y}$ , correspondentes ao padrão  $p$ . Pretende-se minimizar o critério (2.3-5).

$$\min_W J = \sum_{i=1}^N J_p(w_{ij}) \quad (2.3-5a)$$

$$J_p(w_{ij}) = \frac{1}{2} \sum_{i=1}^{n_y} [y_{ip} - x_{ip}^L]^T [y_{ip} - x_{ip}^L] \quad (2.3-5b)$$

A variável  $x_{ip}^L$  define a saída do neurónio  $i$ , para a camada  $L$  (saída), correspondente ao padrão  $p$  e  $y_{ip}$  representa a saída desejada  $i$  para o padrão  $p$ . O algoritmo de retropropagação, ou regra delta generalizada<sup>1</sup> como também é conhecido, definido por,

$$\Delta w_{ij}^l = \eta p_{ip}^l x_{jp}^{l-1} \quad (2.3-6)$$

em que  $x_{ip}^l$  define a saída do neurónio  $i$ , para a camada  $l$  e os valores de  $p_{ip}^l$ , são determinados para a camada de saída ( $l=L$ ) a partir de (2.3-7a) e para as restantes camadas por (2.3-7b),

$$p_{ip}^L = (y_{ip} - x_{ip}^L) \sigma'(a_{ip}^L) \quad (2.3-7a)$$

---

<sup>1</sup> Generalized Delta Rule

$$p_{ip}^l = \sum_{r=1}^{N_{l+1}} (p_{rp}^{l+1} w_{ri}^{l+1}) \sigma'(a_{ip}^l) \quad (2.3-7b)$$

onde  $\sigma'(\cdot)$  representa a derivada da função de activação.

A maior vantagem da aplicação deste tipo de método é a sua simplicidade, permitindo uma implementação prática relativamente fácil. Como maior desvantagem aponta-se as suas propriedades de convergência, bastante lentas, especialmente quando o resultado se aproxima de um mínimo. Para ultrapassar a sua ineficácia têm sido propostas algumas alterações ao método original. Uma das mais conhecidas passa pela introdução de um termo adicional, denominado momento (Jacobs, 1988), que permite suavizar e simultaneamente aumentar a velocidade de aprendizagem. Outras alternativas passam pela introdução de coeficientes de aprendizagem variáveis ou mesmo pela adaptação das funções de activação (Jervis e Fitzgerald, 1993).

Ainda para o treino de redes MLNN foi proposta a utilização do método dos mínimos quadráticos recursivos,  $RLS^2$ , de forma a aumentar a velocidade de convergência, de que são exemplos os trabalhos de Mahmood *et al.* (1992) e Bilski e Rutkowski (1998). Verma (1997), usando uma transformação inversa da função de activação, converte o problema de treino na resolução de um sistema de equações lineares conseguindo, pelo menos para os exemplos por ele apresentados, tempos de treino bastante inferiores aos de retropropagação.

### 2.3.2 Extensão do Método de BP a Redes Recorrentes

O método de retropropagação (BP) não pode ser directamente aplicado a redes recorrentes uma vez que nestas existe o problema de dependência temporal. Numa rede recorrente a saída de cada neurónio interno,  $x(k)$ , depende em cada instante da sua saída no instante anterior  $x(k-1)$ , que por sua vez também depende da saída daquele que lhe precede,  $x(k-2)$ . Continuando o raciocínio, pode-se dizer que a saída de um neurónio depende de toda a história passada, desde o instante inicial.

---

<sup>2</sup> RLS- Recursive Least Squares

## Formulação do Problema

Considere-se uma rede recorrente particular com  $n_u = 2$  entradas externas e  $n = 3$  neurónios, **Error! Reference source not found.** Seja  $x_i(k)$  a saída do neurónio  $i$  no instante  $k$  e seja  $u_j(k)$  a entrada externa  $j$  nesse instante.

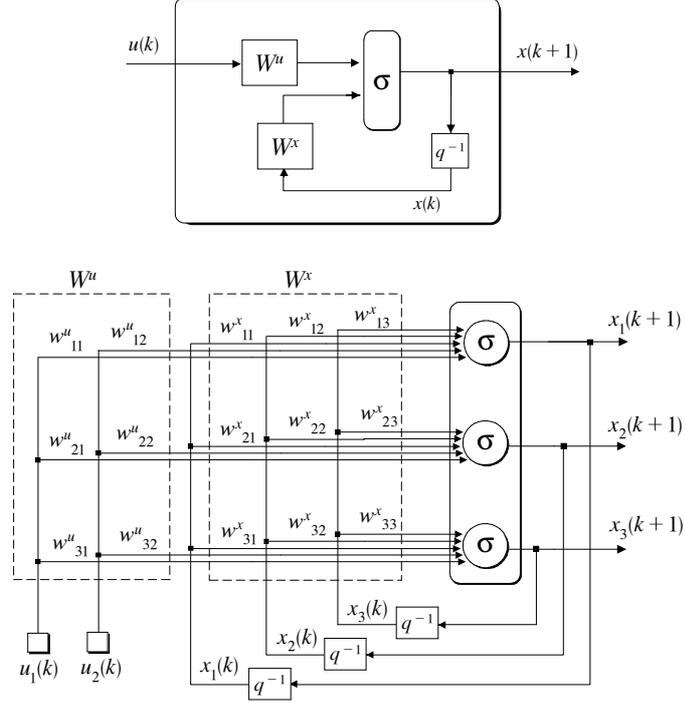


Figura 2.3-1: Exemplo de uma rede recorrente.

Defina-se ainda um vector  $v(k) \in \mathfrak{R}^{n+n_u}$ , constituído por

$$v_i(k) = \begin{cases} u_i(k) & \text{se } i \in U \\ x_i(k) & \text{se } i \in X \end{cases} \quad (2.3-8)$$

em que  $U$  define o conjunto dos índices que representam as entradas  $u_i(k)$  e  $X$  o conjunto dos índices que representam os neurónios  $x_i(k)$ . A dinâmica de cada neurónio é descrita por:

$$a_m(k+1) = \sum_{i \in X} w_{mi}^x x_i(k) + \sum_{i \in U} w_{mi}^u u_i(k) \quad (2.3-9a)$$

$$x_m(k+1) = \sigma(a_m(k+1)) \quad (2.3-9b)$$

A variável  $w_{mi}^x$  define o peso da ligação do neurónio  $i$  para o neurónio  $m$ ,  $w_{mi}^u$  define o peso da ligação da entrada  $i$  para o neurónio  $m$ . A variável  $a_m(k)$  representa um valor intermédio (activação) e  $\sigma(\cdot)$  define a função de activação. Usando (2.3-8), a equação (2.3-9) pode ser escrita como (2.3-10).

$$x_m(k+1) = \sigma \left( \sum_{i \in U \cup X} w_{mi} v_i(k) \right) \quad (2.3-10)$$

onde  $w_{mi}$  representa o componente  $(m,i)$  da matriz de pesos global,  $W = [W^x \ ; \ W^u] \in \mathfrak{R}^{n, n+n_u}$ .

Assumindo que se conhece  $y_m(k)$ , o valor desejado para a saída  $x_m(k)$ , o objectivo da estimação consiste em determinar os valores dos parâmetros  $W$ , de forma a que seja minimizado o seguinte critério:

$$J(k) = \frac{1}{2} \sum_{m \in X} e_m(k)^T e_m(k) \quad (2.3-11)$$

em que o erro  $e_m(k)$  é dado por (2.3-12).

$$e_m(k) = y_m(k) - x_m(k) \quad (2.3-12)$$

Existem dois conceitos distintos no treino de uma rede recorrente: um em que se considera a evolução dinâmica da rede e se pretende aproximar essa dinâmica ao longo do tempo, outro, em que se considera apenas o estado de equilíbrio que ela atinge. Este último conceito é equivalente à noção de memória associativa, já utilizada por Hopfield. Para este caso especial, em que a trajectória é entendida como constante, Pineda (1987) foi o primeiro a constatar que o algoritmo de retropropagação poderia ser facilmente generalizado, obtendo-se o algoritmo de retropropagação recorrente (RBP<sup>3</sup>). Por sua vez, Almeida (1987) obteve resultados análogos, em tempo discreto, aos que Pineda tinha obtido em tempo contínuo.

No caso genérico, em que a saída da rede evolui segundo uma determinada trajectória, são citados como referências fundamentais os seguintes algoritmos de estimação de parâmetros:

- i) *retropropagação temporal* (BPTT<sup>4</sup>), investigado, entre outros, por Werbos (1974) e (1990) e Rumelhart *et al.* (1986);
- ii) *recorrente em tempo real* (RTRL<sup>5</sup>), investigado por Williams e Zipser, (1989a), (1989b) e (1995) e Pearlmutter (1990);

---

<sup>3</sup> RBP- Recurrent BackPropagation

<sup>4</sup> BPTT- BackPropagation Through Time

- iii) retropropagação dinâmica (DBP<sup>6</sup>), introduzido por Narendra e Parthasarathy (1991).

### Algoritmo de Retropropagação Temporal (BPTT)

O algoritmo *BPTT* é basicamente um algoritmo de gradiente, baseado na extensão do algoritmo de retropropagação (aplicável às redes *MLNN*) a redes recorrentes. A compreensão do algoritmo é intuitiva: uma rede recorrente é expandida formando uma rede multi-camada, na qual é inserida uma camada por cada instante de operação, sendo o cálculo das derivadas depois efectuado como se tratasse de uma rede *MLNN*. Na Figura 2.3-2 mostra-se a rede recorrente considerada na **Error! Reference source not found.**, consistindo em  $N + 1$  camadas, correspondentes aos instantes  $k - N$  até  $k$ .

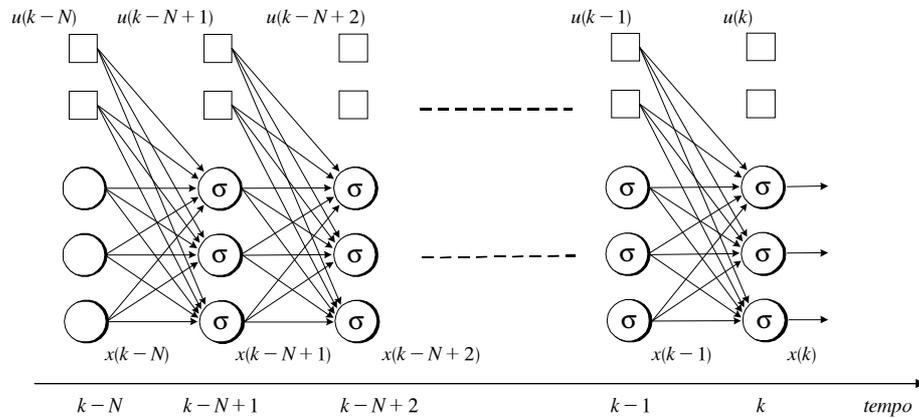


Figura 2.3-2: Expansão da rede recorrente da **Error! Reference source not found.**

O cálculo das derivadas, usando o algoritmo de retropropagação, é dado por (2.3-13), (Williams e Zipsler, 1995).

$$\frac{\partial J}{\partial w_{ij}} = \eta \sum_{k-N}^k p_i(k) v_j(k-1) \quad (2.3-13)$$

O valor de  $p_i(k)$  define uma sensibilidade que é calculada recursivamente para  $i \in X$  e para  $k \in [k - N, k]$  por:

$$p_i(k) = \varepsilon_i(k) \sigma'(a_i(k)) \quad (2.3-14a)$$

<sup>5</sup> RTRL- Real Time Recurrent Learning

<sup>6</sup> DBP- Dynamic BackPropagation

$$\varepsilon_i(k-1) = \sum_{j \in X} w_{ji} p_j(k) \quad (2.3-14b)$$

iniciando-se o processo com (2.3-15).

$$\varepsilon_i(k) = e_i(k) = y_i(k) - x_i(k) \quad (2.3-15)$$

Se além do valor da saída em cada instante  $k$  forem especificados os valores para os instantes intermédios  $[k-N, \dots, k-1]$  a equação (2.3-14b) é reescrita como (2.3-16).

$$\varepsilon_i(k-1) = e_i(k-1) + \sum_{j \in X} w_{ji} p_j(k) \quad (2.3-16)$$

A maior desvantagem do *BPTT* é a de que não pode ser aplicado directamente na prática, tal como é definido. De facto, se assim fosse, seria necessária uma capacidade de memória infinita para armazenar toda a informação passada, razão pela qual por vezes se denota este algoritmo por *BPTT*( $\infty$ ).

#### Algoritmo Recorrente em Tempo Real (RTRL)

O algoritmo *RTRL*, ao contrário do *BPTT*, não se baseia na duplicação das unidades para o cálculo das derivadas parciais. No *RTRL* a determinação das derivadas parciais é efectuada tendo directamente em conta as dependências temporais, (Werbos, 1974). Permite a actualização dos pesos enquanto os dados vão sendo processados e é, por esta razão, conseqüentemente um método possível de aplicar em-linha, sendo este um dos seus maiores atractivos. Usando as regras de derivação, é possível definir a actualização dos parâmetros como (2.3-17).

$$\Delta w_{ij}(k) = -\eta \frac{\partial J(k)}{\partial w_{ij}} = \eta \sum_{m \in X} e_m(k) \frac{\partial y_m(k)}{\partial w_{ij}} \quad (2.3-17a)$$

$$\Delta w_{ij}(k) = \sum_{m \in X} e_m(k) p_{ij}^m(k) \quad (2.3-17b)$$

Nesta equação a variável  $\eta \in \mathfrak{R}^+ \setminus 0$  define um ganho e  $p_{ij}^m(k)$ , com  $i \in U, j \in U \cup X$ , define a sensibilidade do neurónio  $m$  em relação à variação no peso  $w_{ij}$ . Após algumas manipulações tem-se que

$$p_{ij}^m(k+1) = \sigma'_m(a_m(k)) \left[ x_j(k) \delta_{mi} + \sum_{n \in U} w_{mn} p_{ij}^n(k) \right] \quad (2.3-18)$$

em que  $\delta_{mi}$  define o operador de *Kronecker*, inicializando-se o processo iterativo estabelecendo  $p_{ij}^n(0) = 0$ , (Williams e Zipser, 1995).

### *BPTT versus RTRL*

Apesar de numa primeira análise serem distintos é possível obter os algoritmos *BPTT* e *RTRL* a partir da mesma formulação (Srinivasan *et al.*, 1994) e (Beaufays e Wan, 1994). A diferença entre o *BPTT* e o *RTRL* reside na forma como a expansão das derivadas e a regra da cadeia são aplicadas. O *BPTT* caracteriza-se por ser local no espaço, mas não no tempo e por ser computacionalmente simples, mas impossível de aplicar na prática. Para a sua aplicabilidade em-linha, é necessário proceder-se a algumas modificações, obtendo-se necessariamente métodos aproximados, de que é exemplo o algoritmo de retropropagação temporal truncado, *TBPTT*<sup>7</sup>.

Por sua vez o *RTRL* é local no tempo mas não no espaço, é computacionalmente mais complexo mas é um método inerentemente em-linha, permitindo a sua aplicação prática sem necessidade de alterações. Diga-se que, em rigor, o algoritmo *RTRL* também não é um método exacto, uma vez que em cada instante é tido em conta na actualização dos parâmetros apenas o valor anterior destes. Para que o método fosse exacto teria de se ter presente toda a evolução dos parâmetros, desde o instante inicial (Williams e Zipser, 1990c).

Em termos de comparação Williams e Zipser (1989b) referem uma melhor eficácia e convergência para o *TBPTT* do que para o *RTRL* e justificam este resultado uma vez que, segundo eles, a aproximação causada por se considerarem apenas os últimos instantes é melhor do que a aproximação efectuada no *RTRL*.

## Modificação dos Algoritmos *BPTT* e *RTRL*

Devido às limitações que qualquer um dos algoritmos originais *BPTT* e *RTRL* apresenta, têm sido efectuadas várias tentativas no sentido de melhorar a suas propriedades.

Pela impossibilidade de aplicação prática do algoritmo *BPTT*, já que implicava a necessidade de uma memória de capacidade infinita que armazenasse toda a história passada, surgiu o já referido algoritmo *TBPTT* (Williams, 1990a). Em cada instante consideram-se para o processo de estimação apenas os últimos  $N$  valores passados, denotando-se o algoritmo por *BPTT(N)*. Uma vez que a informação passada é esquecida, o método é particularmente conveniente em situações em que é aceitável e/ou conveniente esquecer a informação passada (Henriques e Dourado, 1999a). Em Pearlmutter (1995) referem-se exemplos de aplicações práticas que usam com sucesso este método de estimação.

Elman (1990) considera ainda uma versão mais simplificada do *BPTT* em que a informação passada é ignorada, e são tidos em conta apenas um ou dois instantes anteriores, permitindo aplicar em-linha o *BPTT*. Obviamente a precisão do método é limitada, mas a ideia que presidiu à sua metodologia foi estabelecer um compromisso entre, por um lado, a precisão e, por outro, o tempo de cálculo e a capacidade de memória computacional requerida.

---

<sup>7</sup> *TBPTT*- Truncated BackPropagation Through Time

Uma possibilidade de reduzir a complexidade do algoritmo *RTRL* consiste em simplesmente não ter em conta os elementos cuja sensibilidade é próxima de zero, isto é,  $p_{ij}^n \approx 0$ , ideia explorada por Zipser (1989). Basicamente, enquanto no *BPTT* se consegue uma simplificação ignorando dependências temporais, a técnica proposta por Zipser ignora determinadas dependências estruturais. A estratégia por ele desenvolvida, *Sub Grouping Strategy*, considera a rede recorrente original subdividida em várias redes de menor dimensão, com a propagação do gradiente a ser efectuada independentemente em cada subrede, ou seja, o valor de  $p_{ij}^n$  é considerado nulo sempre que  $i$  e  $n$  digam respeito a subredes distintas. Lu (1996), para ultrapassar problemas de representação existente na metodologia de Zipser, propôs o algoritmo *Mode Exchange RTRL*, capaz de operar segundo duas modalidades distintas: modalidade estática e modalidade dinâmica. Na primeira (estática), que ocorre enquanto o critério de erro quadrático diminui, considera-se  $p_{ij}^n = 0$ . Na segunda (dinâmica), que ocorre quando o critério aumenta, todos os valores  $p_{ij}^n$  são calculados, tal como no algoritmo original *RTRL*.

Outra possibilidade, que não é na verdade um método aproximado, baseia-se em considerar apenas  $N$  instantes com base no *BPTT* mas usar também o *RTRL* para incorporar toda a história passada. Esta técnica foi desenvolvida por Williams e Peng (1990b) e Schmidhuber (1992) e, tal como referido por Williams e Zipser (1995), constitui certamente a forma mais eficiente de calcular com rigor o valor do gradiente. Também neste contexto, Campolucci *et al.* (1997), propuseram um método que permite combinar a relativa simplicidade do *BPTT* com a possibilidade de aplicação em em-linha do *RTRL*. Contudo, os seus resultados eram apenas aplicáveis a redes com recorrência local.

### 2.3.3 Métodos de Optimização Não Linear

Até ao momento tem-se abordado o problema da estimação recorrendo a conceitos e técnicas especificamente desenvolvidos para as redes neuronais. No entanto, se se entender um modelo neuronal como uma qualquer função não linear, o problema de estimação pode ser formulado como um problema genérico de optimização não linear. É extensa a bibliografia sobre este assunto referindo-se métodos de análise numérica, investigação operacional e computação estatística, Gill *et al.* (1981), Fletcher (1987), Bertsekas e Tsitsiklis (1996), só para citar alguns.

Visto segundo esta perspectiva o método de gradiente é um método de primeira ordem, uma vez que utiliza apenas informação calculada com base nas primeiras derivadas. Os métodos de segunda ordem, que tiram partido do conhecimento das derivadas de segunda ordem permitem, pelo menos em princípio, obter melhores resultados. São exemplos os métodos de Newton, Gauss-Newton e Levenberg-Marquardt, que se passam a descrever sucintamente.

De forma a melhorar a eficiência do método de gradiente, pode-se efectuar uma modificação na direcção de procura,  $\Xi(W)$ , (2.3-2), introduzindo para o efeito uma matriz definida positiva,  $M(W)$ , como se descreve na equação (2.3-19).

$$\Xi(W) = M(W) (-\nabla J(W)) \quad (2.3-19)$$

Se no método de gradiente  $M(W) = \eta$ , no método de Newton a matriz  $M(W)$  é definida como sendo a inversa da *Hessiana*,

$$M(W) = H^{-1} \quad (2.3-20)$$

que por sua vez é definida como a matriz das segundas derivadas.

$$H = \frac{\partial^2 J}{\partial W \partial W^T} \quad (2.3-21)$$

Uma desvantagem do método de Newton é a necessidade de usar informação de segunda ordem e a possibilidade de existência de singularidades, que dificulta, ou mesmo impossibilita o cálculo da inversa.

Os métodos de quasi-Newton tentam, em certa medida, ultrapassar o problema estimando o valor da *Hessiana* com base apenas em informação de primeira ordem. No método Gauss-Newton é calculada uma aproximação, da *Hessiana*  $H_a$ , tendo em conta apenas o cálculo do gradiente (*Jacobiano*) (2.3-4), desprezando-se o termo de segunda ordem.

$$H_a \approx \frac{\partial J^T}{\partial W} \frac{\partial J}{\partial W} = G^T G \quad (2.3-22)$$

Contudo, mantém-se a necessidade de calcular a inversa de uma matriz. Uma forma possível de ultrapassar o problema consiste em adicionar uma restrição adicional, proposta por Levenberg-Marquardt (Marquardt, 1963).

$$H_a \approx \frac{\partial J^T}{\partial W} \frac{\partial J}{\partial W} + \gamma I = G^T G + \gamma I \quad (2.3-23)$$

A variável  $\gamma$  é uma constante real positiva e  $I$  é uma matriz identidade de dimensões apropriadas. Para  $\gamma = 0$  o método reduz-se ao método de Newton, para  $\gamma$  elevado obtém-se o método de gradiente. Este método é particularmente adequado a problemas mal condicionados, de que é exemplo o problema em causa, ou seja, a estimação de parâmetros em redes neuronais.

Como exemplos de aplicação deste tipo de técnicas à estimação de parâmetros de modelos neuronais referem-se os métodos de gradiente conjugado (Moller, 1993), de

Gauss-Newton (Stager e Agarwal, 1997) ou o método de Levenberg-Marquardt (Hagan e Menhaj, 1994).

## 2.3.4 Estimação de Estados

Se por um lado são reconhecidas vantagens no uso de modelos de estados, por outro, porque na prática os estados são quase sempre inacessíveis, a sua aplicação torna-se, grande parte das vezes, inviável. Uma forma de ultrapassar o problema é a construção de observadores, inferindo uma estimativa para o valor do estado do sistema, apenas a partir de medidas das suas entradas e saídas.

### Observadores Para Sistemas Lineares

#### Observador de Luenberger

Para sistemas lineares e invariantes a teoria de projecto de observadores encontra-se bem estabelecida (Ljung e Hunnarson, 1990). Para sistemas determinísticos referem-se os observadores de Luenberger (Luenberger, 1966), (Luenberger, 1971) e suas extensões (Bhattacharyya, 1976), (Fairman e Gupta, 1980), (Reilley, 1983). Basicamente um observador de Luenberger possui um termo de correcção, em que a diferença entre a saída do sistema real e a do modelo de estados é usada para actualizar o valor do estado estimado.

Considerando-se um sistema linear e invariante, descrito por (2.3-24).

$$x(k+1) = A x(k) + B u(k) \quad (2.3-24a)$$

$$y(k) = C x(k) \quad (2.3-24b)$$

O observador de Luenberger é definido por (2.3-25),

$$\hat{x}(k+1) = A \hat{x}(k) + B u(k) + L (y(k) - C \hat{x}(k)) \quad (2.3-25)$$

em que  $\hat{x} \in \mathfrak{R}^n$  é uma estimativa para o valor do estado a determinar,  $x \in \mathfrak{R}^n$ . O erro de estimação de estado, definido pela diferença entre o estado e a sua estimativa, dado por (2.3-26).

$$\varepsilon(k) = x(k) - \hat{x}(k) \quad (2.3-26)$$

Assumindo o sistema observável é possível determinar uma matriz  $L \in \mathfrak{R}^{n,y}$ , designada por ganho de observação, de tal forma que a dinâmica do erro seja definida por (2.3-27).

$$\varepsilon(k+1) = (A - LC) \varepsilon(k) = A_o \varepsilon(k) \quad (2.3-27)$$

Pode-se desta forma assegurar a convergência (para zero) do erro de observação desde que para tal se garanta que a matriz  $A_o$  seja Hurwitz (valores próprios dentro do círculo de raio unitário), garantindo que  $\hat{x}(k)$  convirja assintoticamente para  $x(k)$ .

### Filtro de Kalman

O correspondente ao observador de Luenberger (determinístico) é, no caso estocástico, o filtro de Kalman (Kalman, 1960). Tal como o observador de Luenberger, o filtro de Kalman inclui um factor corrector, sendo, no entanto, o ganho de observação otimizado de forma a solucionar um problema de estimação de mínimos quadráticos.

O filtro calcula dinamicamente uma estimação da covariância do erro e, em cada iteração, recalcula o ganho óptimo para o observador. Para sistemas lineares, considerando ruído branco Gaussiano, o filtro de Kalman permite uma solução óptima, no sentido da máxima verosimilhança, para o problema de estimação de estados.

## Observadores Para Sistema Não Lineares

Se para sistemas lineares se referem os observadores de Luenberger e o filtro de Kalman, não existe uma teoria genérica para sistemas não lineares. Tendo como ponto de partida a utilização de modelos neuronais recorrentes não lineares na identificação, e dentro dos métodos de observação adequados a este tipo de modelos, referem-se:

- i) *métodos de linearização*, cujo objectivo é permitir a aplicação de técnicas para sistemas lineares, a partir de uma linearização do modelo não linear;
- ii) *métodos de verificação*, em que se utiliza um observador linear, mas são estabelecidas condições necessárias e suficientes de convergência;
- iii) *observadores de modo deslizante*, que são basicamente observadores lineares com uma parte adicional não linear, baseados na teoria de sistemas de estrutura variável e caracterizados pelas suas propriedades de robustez, quer em relação a incertezas de modelização quer em relação a incertezas de parâmetros;
- iv) *observadores integrais*, que incluem, tal como os controladores integrais, um termo integral capaz de garantir a convergência do erro de identificação.

### Métodos de Linearização

Uma solução possível para sistemas não lineares consiste em aproximar o sistema não linear por um linear (através de uma linearização) e implementar-se uma técnica linear de observação, tal como se de um sistema linear se tratasse. Portanto, são técnicas que requerem o conhecimento exacto do sistema e das suas não linearidades.

Um exemplo deste tipo de técnicas é o filtro de Kalman estendido, *EKF*<sup>8</sup>. Proposta por Kopp e Orford (1963), é nesta técnica efectuada uma linearização do modelo não

---

<sup>8</sup> *EKF- Extended Kalman Filter*

linear e é depois utilizado um filtro de Kalman. Consiste numa metodologia recursiva, que combina de uma forma óptima observações (com ruído) com predição inferida pelo modelo, e permite obter uma solução aproximada, sob o ponto de vista da máxima verosimilhança, Ljung e Hunnarson (1990).

Baumman e Rugh (1986) propuseram uma técnica que consistia também numa linearização, só que efectuada de forma a assegurar que a dinâmica do erro de estimação do sistema linearizado fosse estável para qualquer regime de operação, à custa da especificação dos valores próprios da matriz da dinâmica do erro. Contudo, para a sua aplicação é necessária a resolução de um sistema de equações diferenciais, fortemente acoplado, aspecto que não é encorajador para o seu uso.

### Métodos de Verificação

Se as técnicas anteriores são genéricas, possíveis à partida de implementar em qualquer sistema não linear, as técnicas de verificação são apenas aplicáveis a sistemas particulares, constituídos por uma parte linear e outra não linear que apresente determinadas características.

A ideia foi introduzida por Thau (1973) e consiste em projectar um observador para a parte linear e verificar depois em que condições o observador assegura a estabilidade e convergência incluindo também a parte não linear. Os resultados de Thau consideram não linearidades específicas do tipo *Lipschitz*, e dizem respeito a sistemas contínuos. Saliente-se que o método de Thau não é na realidade um método sistemático para o projecto de observadores, mas apenas uma metodologia que fornece condições necessárias e suficientes para a convergência de um observador não linear (de Luenberger) projectado para o sistema linear.

Existem, além dos de Thau, outros resultados considerando este tipo de observador aplicado a sistemas com não linearidades limitadas. Exemplos são os trabalhos de Raghavan e Hendrick (1994), Rajamani (1998) e Schreier *et al.* (1999). De comum, o facto de todos eles serem em tempo contínuo e de usarem a teoria de Lyapunov para garantir as propriedades de estabilidade e de convergência.

### Observadores de Modo Deslizante

Ao contrário dos observadores já referidos que assumem, e por isso dependem, do conhecimento da descrição da dinâmica do sistema, os observadores de modo deslizante (SMO<sup>9</sup>), caracterizam-se por serem, de certa forma, insensíveis em relação à forma das não linearidades. Um observador de modo deslizante tem uma estrutura variável (VSO<sup>10</sup>), sendo projectado de tal forma que a trajectória do erro se situe num determinado hiper-plano atractivo.

Basicamente, consistem num observador de Luenberger e num termo adicional (descontínuo) que lhes confere a capacidade de lidar com incertezas e não linearidades (Slotine *et al.*, 1987), tendo, no entanto, na maior parte das vezes, mais a ver com incertezas do que com não linearidades (Utkin, 1992). Além disso, existe uma condição

---

<sup>9</sup> SMO- *Sliding Mode Observer*

<sup>10</sup> VSO- *Variable Structure Observer*

fundamental para a sua utilização: terem de garantir as condições de verificação (*matching conditions*).

Um dos trabalhos de referência neste contexto é o de Walcott e Zak (1987a) e (1987b). Para uma classe particular de sistemas não lineares, cuja não linearidade é limitada, Walcott e Zak propuseram um observador de modo deslizante, determinado a partir de metodologias específicas de controlo robusto. Recorrendo à teoria de Lyapunov demonstraram as propriedades de convergência e estabilidade para a sua metodologia assumindo que se cumpram as referidas condições de verificação.

Outros exemplos são os trabalhos de Yaz e Azemi, (1993) que estenderam os resultados de Walcott e Zak a sistemas não lineares estocásticos e os de Choi *et al.* (1999) e Koshkouei e Zinober (1999) a sistemas SISO, em que as não linearidades não necessitavam obrigatoriamente de garantir as condições de verificação. Em (Haskara *et al.* 1998) é utilizada esta técnica de observação, combinada com o princípio de controlo equivalente, garantindo tempos finitos de convergência, à semelhança de um controlador de tempo de estabelecimento finito (*deadbeat*).

### Observadores Integrais

Na sua formulação o observador de Luenberger possui um termo que corrige o valor do estado actual por uma quantidade proporcional ao erro de predição, isto é, proporcional à diferença entre o valor da saída real e o valor da saída do modelo. Um observador integral caracteriza-se por, além do termo proporcional, conter também um termo corrector integral. Esta técnica foi originalmente proposta por Kim *et al.* (1989) no contexto do diagnóstico e detecção de falhas, de forma a permitir lidar com imprecisões e incertezas no modelo. Além desta propriedade refere-se a capacidade de rejeição de perturbações, analisada por Beale e Shafai (1989). Linder (1997) estendeu esta ideia a observadores estocásticos, concretamente a filtros de Kalman, obtendo um observador denominado *Proportional Integral Kalman Filter*. No caso de sistemas determinísticos, um observador integral define-se por (2.3-28).

$$\hat{x}(k+1) = A\hat{x}(k) + B u(k) + K_p (y(k) - C\hat{x}(k)) + B v(k) \quad (2.3-28a)$$

$$v(k+1) = v(k) + K_i (y(k) - C\hat{x}(k)) \quad (2.3-28b)$$

A inclusão do termo integral,  $Bv(k)$ , permite à partida ultrapassar dificuldades que o filtro de Kalman possui, como a impossibilidade de operação caso existam variações nos parâmetros que caracterizam o sistema. De facto, a inclusão deste termo permite, pelo menos teoricamente, o desempenho correcto do par estimador e controlador (por realimentação de variáveis de estado) facto que não é verdade para o proporcional, caso existam incertezas nos parâmetros do modelo.

#### Determinação de $K_p$ e $K_i$

De forma a determinar as matrizes  $K_p \in \mathfrak{R}^{n,n_y}$  e  $K_i \in \mathfrak{R}^{n_y,n_y}$  a equação (2.3-28) pode ser descrita, tal como um observador de Luenberger (2.3-25).

$$\hat{z}(k+1) = \underline{A}\hat{z}(k) + \underline{B}u(k) + \underline{K}(y(k) - C\hat{z}(k)) \quad (2.3-29a)$$

$$y(k+1) = \underline{C}\hat{z}(k) \quad (2.3-29b)$$

em que

$$z(k) = \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (2.3-30a)$$

$$\underline{K} = \begin{bmatrix} K_p \\ K_i \end{bmatrix} \quad (2.3-30b)$$

$$\underline{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \quad (2.3-30c)$$

$$\underline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (2.3-30d)$$

$$\underline{C} = [C \quad 0] \quad (2.3-30e)$$

Para que o observador seja estável as matrizes  $K_p$  e  $K_i$  devem ser seleccionadas de forma a que a matriz (2.3-31) tenha valores próprios estáveis.

$$\begin{bmatrix} A - K_p C & B \\ K_i C & I \end{bmatrix} \quad (2.3-31)$$

Para a determinação dos ganhos de observação,  $K_p$  e  $K_i$ , Shafai *et al.* (1996) propõem a utilização de técnicas oriundas da teoria de controlo óptimo. Em concreto, é por eles estabelecido um critério do tipo (2.3-32), em que as matrizes  $Q$  e  $R$  são positivas definidas,

$$J = \sum_k (x^T Q x + u^T R u) \quad (2.3-32)$$

sendo a solução obtida resolvendo a seguinte equação de Riccati.

$$P = \underline{A} P \underline{A}^T - \underline{A} P \underline{C}^T (\underline{C} P \underline{C}^T + R)^{-1} \underline{C} P \underline{A}^T + Q \quad (2.3-33)$$

Os ganhos de observação são depois obtidos a partir de (2.3-34).

$$\underline{K} = \begin{bmatrix} K_p \\ K_i \end{bmatrix} = -\underline{A} P \underline{C}^T (\underline{C} P \underline{C}^T + R)^{-1} \quad (2.3-34)$$

### 2.3.5 Estimação de Parâmetros e de Estados com Técnicas de Observação

Apesar das técnicas de observação serem originalmente aplicadas ao problema da estimação de estados elas podem, de forma idêntica, ser aplicadas à estimação de parâmetros. Para o efeito, os parâmetros são tratados como um vector de “estados”, e o seu valor é determinado como se de estados, propriamente ditos, se tratassem.

Assim, qualquer uma das técnicas anteriores pode, depois de feitas as devidas adaptações, ser usada na estimação de parâmetros de modelos. De entre elas destaca-se o denominado filtro de Kalman dual estendido *DEKF*<sup>11</sup>.

#### Filtro de Kalman Dual Estendido

O filtro de Kalman dual estendido, combina a estimação de estados com a estimação de parâmetros em sistemas não lineares (considerando a sua linearização), ideia inicialmente proposta para sistemas lineares por Nelson e Stear (1994).

Considere-se um sistema não linear, descrito de uma forma genérica por:

$$x(k+1) = f(x(k), u(k), \theta) \quad (2.3-35a)$$

$$y(k+1) = C x(k+1) = C f(x(k), u(k), \theta) \quad (2.3-35b)$$

Pretende-se determinar, a partir da observação das entradas e das saídas, uma estimativa  $\hat{x}(k)$  para os estados do sistema  $x(k)$ , e simultaneamente uma estimativa  $\hat{\theta}$  para os parâmetros  $\theta$ . Assume-se ainda a existência de ruído branco aditivo sobre a saída,  $\zeta_x(k)$ , e sobre os estados  $v_x(k)$ , respectivamente, de variância  $Q^x$  e  $R$ .

Para determinar os parâmetros  $\theta$ , estes consideram-se como constituindo um vector, a tratar de forma idêntica aos estados  $x$ . De forma a estabelecer uma equação que permite efectuar a actualização dos parâmetros, considera-se que a melhor estimativa para o instante seguinte é o valor que se conhece neste instante, (2.3-36), isto é, os parâmetros são considerados constantes.

$$\hat{\theta}(k+1) = \hat{\theta}(k) \quad (2.3-36)$$

---

<sup>11</sup> *DEKF- Dual Extended Kalman Filter*

Tal como para a actualização dos estados, na actualização dos parâmetros assume-se a existência de ruído,  $\zeta_{\theta}(k)$ , e  $v_{\theta}(k)$ , respectivamente de variância  $Q^{\theta}$  e  $R$ .

Para a estimação de estados e de parâmetros consideram-se dois estimadores, do tipo EKF, a operar em simultâneo. Enquanto um deles estima o valor dos estados  $\hat{x}$ , usando para tal o valor estimado dos parâmetros  $\hat{\theta}$ , o outro determina uma estimação para os parâmetros  $\hat{\theta}$ , com base nos valores estimados para os estados  $\hat{x}$ .

O processo iterativo é inicializado considerando uma estimativa para o valor dos estados e parâmetros iniciais, respectivamente  $\hat{x}(0) = \hat{x}_0$  e  $\hat{\theta}(0) = \hat{\theta}_0$ , assim como das respectivas matrizes de covariância iniciais  $P_{x_0}$  e  $P_{\theta_0}$ , que reflectem de alguma forma a qualidade da estimação efectuada. Para  $k = 1, \dots, \infty$ , o processo iterativo é efectuado em duas etapas distintas: predição e actualização com base nas medidas efectuadas. O símbolo ( $\bar{\quad}$ ) no termo  $\hat{\theta}_k^-$  é utilizado para diferenciar a estimação, no instante  $k$ , antes da fase de actualização de informação e após a actualização de informação (Wan e Nelson, 2000).

1. *Predição:*

i) *parâmetros*

$$\hat{\theta}_k^- = \hat{\theta}_{k-1} \quad (2.3-37a)$$

$$P_{\theta_k}^- = P_{\theta_{k-1}} + Q^{\theta} \quad (2.3-37b)$$

ii) *estados*

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, \hat{\theta}_k^-) \quad (2.3-38a)$$

$$P_{x_k}^- = A_{k-1} P_{x_{k-1}} A_{k-1}^T + Q^x \quad (2.3-38b)$$

2. *Actualização:*

i) *estados*

$$K_k^x = P_{x_k}^- (C_k^x)^T (C_k^x P_{x_k}^- (C_k^x)^T + R)^{-1} \quad (2.3-39a)$$

$$\hat{x}_k = \hat{x}_k^- + K_k^x e_k \quad (2.3-39b)$$

$$P_{x_k} = (I - K_k^x C_k^x) P_{x_k}^- \quad (2.3-39c)$$

ii) *parâmetros*

$$K_k^{\theta} = P_{\theta_k}^- (C_k^{\theta})^T (C_k^{\theta} P_{\theta_k}^- (C_k^{\theta})^T + R)^{-1} \quad (2.3-40^a)$$

$$\hat{\theta}_k = \hat{\theta}_k^- + K_k^\theta e_k \quad (2.3-40b)$$

$$P_{\theta_k} = (I - K_k^\theta C_k^\theta) P_{\theta_k}^- \quad (2.3-40c)$$

em que

$$e_k = y_k - C \hat{x}_k^- \quad (2.3-41)$$

$$A_{k-1} = \left. \frac{\partial f(x, \hat{\theta}_k^-)}{\partial x} \right|_{\hat{x}_{k-1}} \quad (2.3-42)$$

$$C_k^x = \left. \frac{\partial f(\hat{x}_k, \theta)}{\partial x} \right|_{\hat{x}_k} = C \quad (2.3-43)$$

$$C_k^\theta = C \left. \frac{\partial f(\hat{x}_{k-1}, u_k, \hat{\theta}_k^-)}{\partial \theta} \right|_{\hat{\theta}_{k-1}} = C \left. \frac{\partial \hat{x}_k^-}{\partial \theta} \right|_{\hat{\theta}_{k-1}} \quad (2.3-44)$$

Este procedimento representa-se esquematicamente na Figura 2.3-3.

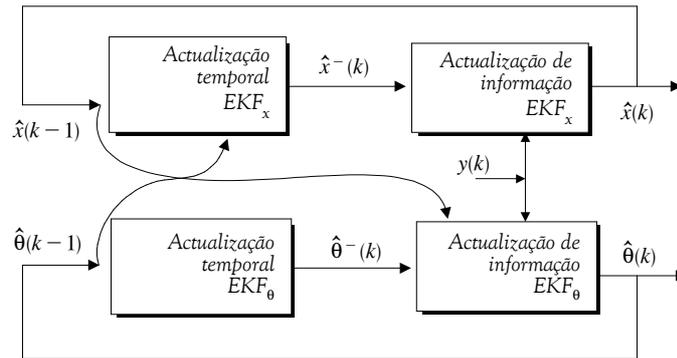


Figura 2.3-3: Filtro de Kalman Dual Estendido.

Se à primeira vista o DEKF parece uma simples concatenação de dois filtros EKF na verdade não é bem assim. De facto, quando se efectua o cálculo das derivadas em (2.3-44), ignora-se a dependência entre  $\hat{x}(k)$  e  $\hat{x}(k-1)$ . Assim, para que a linearização fosse correctamente efectuada ter-se-ia de ter em conta esta dependência temporal, implicando o cálculo de derivadas dinâmicas. Mais ainda, nas etapas de actualização, tanto o valor de  $\hat{x}(k-1) \equiv \hat{x}_{k-1}$  como o valor de  $\hat{x}(k) \equiv \hat{x}_k$  dependem de  $\hat{\theta}$  e do ganho  $K_k^\theta$ . Daí que a sua implementação precisa é, além de complexa, pesada do ponto de vista computacional (Wan e Nelson, 2000). Por esta razão na prática estas dependências são muitas vezes ignoradas, eliminando-se a necessidade de derivadas dinâmicas.

## 2.3.6 Técnicas de Observação Aplicadas às Redes Neurais

Se uma rede neuronal recorrente for entendida como um modelo de estados não linear, o que de facto pode ser assumido, é possível aplicar técnicas de observação na estimação dos seus parâmetros e dos seus estados. Ou seja, transforma-se o problema original de treino de uma rede recorrente num problema de estimação simultâneo de parâmetros (os pesos da rede  $W = \hat{\theta}$ ) e de estados (saída da unidade interna  $x_n = \hat{x}$ ).

Note-se que é interessante constatar que o problema de determinação dos pesos da rede pode ser analisado segundo duas perspectivas distintas. A primeira, em que se pretendem determinar os parâmetros de um modelo, que é em particular um modelo neuronal, tendo neste caso sentido falar de estimação de parâmetros de modelos neuronais usando técnicas de observação. A segunda, em que se assume à partida que a rede deve desempenhar o papel de um observador, o mesmo é dizer, que se pretende estimar os estados de um sistema. Fala-se então de observadores neuronais.

### Estimação de Parâmetros de Redes Recorrentes Usando Técnicas de Observação

Uma das técnicas de observação (para sistemas não lineares) que tem sido utilizada com sucesso na estimação dos parâmetros de redes neuronais é o *EKF*.

A ideia original partiu de Singhal e Wu (1989), implementando o *EKF* para o treino de redes *MLNN* (sem recorrência). Porque se considerava a correlação entre todos os parâmetros da rede este algoritmo foi denominado por *Global Extended Kalman Filter*. Ruck *et al.* (1992) e Chadran (1994) investigaram a relação deste último método com o de retropropagação. Concluíram, que se por um lado tinha a vantagem de proporcionar melhores resultados de aproximação, por outro tinha no peso computacional a sua principal desvantagem.

Para a estimação de parâmetros de redes recorrentes o algoritmo *EKF* foi inicialmente proposto por Matthews (1990) em aplicações de processamento de sinal.

De todos os trabalhos de aplicação de algoritmos de *EKF* a redes neuronais recorrentes, provavelmente o mais conhecido é o de Puskorious e Feldkamp (1991) e (1994). De forma a diminuir a complexidade do *EKF* sugeriram uma técnica baseada em subconjuntos de parâmetros que, para as finalidades do algoritmo, consideraram independentes. No algoritmo por eles proposto, denominado *Decoupled Extended Kalman Filter*, considera-se nula a covariância entre parâmetros de subconjuntos distintos, de alguma forma semelhante ao trabalho de Zipser (1989), proposto para a simplificação do *RTRL*.

Ainda relativamente à aplicação do algoritmo *EKF* no contexto de redes neuronais referem-se os trabalhos de Jin *et al.* (1995a), onde é proposto o algoritmo *Decomposed Extended Kalman Filter*, e os trabalhos de Wan e Nelson (1997) e (2000).

## Observadores Baseados em Redes Neurais

Os casos de estudo em que as redes neuronais são directamente interpretadas como observadores são escassos. Sørensen (1991) e (1996) propôs, para um sistema dinâmico não linear, uma rede neural MLNN a executar as funções de um observador. Concretamente, na sua metodologia, inspirada em técnicas de observação lineares, a rede é definida por uma equação do tipo (2.3-45), treinada de forma a fornecer um termo corrector, tal como um observador de Luenberger.

$$x_n(k+1) = \hat{x}(k+1) = \mathcal{N}(\hat{x}(k), e_y(k), u(k), W) \quad (2.3-45)$$

O treino da rede é efectuado em-diferido, com base na minimização do erro de predição da saída,  $e_y(k) = y(k) - y_n(k)$ . Para o treino da rede Sørensen sugere os algoritmos de retropropagação (Sørensen, 1991) e o de Gauss-Newton (Sørensen, 1996).

O projecto de observadores seguindo este conceito, que se podem designar por estáticos, uma vez que são baseados em redes MLNN e o seu treino é efectuado em-diferido, foi estudado com detalhe em Levin e Narendra (1996), inserido num contexto global de métodos de identificação e controlo para sistemas não lineares.

Seguindo a mesma abordagem Ahmed e Riyaz (2000) usaram como observador uma rede MLNN, que se assemelha a um filtro de Kalman estendido e sugerem dois esquemas de treino, ambos em-diferido. No primeiro esquema a rede é treinada de forma a fornecer o termo corrector, com base na minimização do erro de predição da saída,  $e_y(k) = y(k) - y_n(k)$ , tal como proposto por Sørensen. No segundo esquema, o treino é baseado na minimização directa do erro de estado,  $\epsilon(k) = x(k) - x_n(k)$ , o que pressupõe a acessibilidade dos estados do processo, hipótese que na prática será apenas viável em situações particulares. Para os exemplos por eles apresentados são obtidos melhores resultados que os proporcionados pelo EKF.

Suykens *et al.* (1995) e (1996) propuseram de forma semelhante modelos neuronais recorrentes entendidos como modelos não lineares. Usam o algoritmo de retropropagação dinâmica, proposto por Narendra e Parthasarathy, (1991), sendo a rede treinada em-diferido de forma a parametrizar directamente um filtro de Kalman.

Em oposição aos observadores estáticos, em que o treino da rede é efectuada em-diferido, referem-se os observadores dinâmicos, em que os parâmetros da rede são adaptados em-linha. Puskorious e Feldkamp (1994) consideraram uma rede recorrente a desempenhar as funções de um observador em que os seus parâmetros eram adaptados em-linha. A estrutura era semelhante a um filtro de Kalman e defenderam a ideia de que uma rede recorrente, treinada por métodos baseados em teorias bem estabelecidas, tal como o filtro de Kalman, poderia implementar eficazmente as funções de um observador.

Ainda neste contexto Kim *et al.* (1997), para uma classe particular de sistemas não lineares SISO, introduziram uma rede recorrente que funciona como um observador. Como principal inovação a salientar o facto de combinarem técnicas robustas de

observação, concretamente observadores de modo deslizante, o que lhes permite provar rigorosamente, usando a teoria de Lyapunov, a estabilidade da metodologia proposta.

Tal como Kim *et al.*, Yu *et al.* (2000) combinam técnicas de modo deslizante e redes neuronais recorrentes para servirem de observadores para uma classe *affine* de processos MIMO não lineares. Para esta classe particular estabelecem condições de estabilidade para a adaptação dos parâmetros (em-linha). Kulawski e Bryds (2000), propõem uma classe particular de modelos neuronais, com uma estrutura diagonal, análoga a uma de recorrência local. Utilizando o algoritmo de retropropagação dinâmica demonstram teoricamente a estabilidade da sua metodologia, tanto nos aspectos de identificação como de controlo.

Henriques e Dourado (1998a) propõem o uso de uma rede modificada de Elman para desempenhar as funções de um observador e utilizam um algoritmo do tipo *TBPTT* de forma a permitir a actualização dos seus parâmetros (em-linha). Inserido numa estrutura de controlo adaptativo o observador neuronal permite a implementação de técnicas de controlo baseadas em realimentação de variáveis de estado. A eficácia da rede neuronal recorrente, a actuar como um observador para o projecto de um controlador, é validada na prática pela sua aplicação a um processo laboratorial multi-variável (*DTS200*).

### 2.3.7 Metodologia Proposta

Nesta secção foram revistos alguns métodos capazes de ser aplicados na determinação dos pesos de uma rede recorrente, nomeadamente os algoritmos *BPTT* e *RTRL*, especificamente desenvolvidos para redes recorrentes, os métodos de optimização não linear, tais como o método de Gauss-Newton e o de Levenberg-Marquardt e técnicas de observação, tal como o filtro de Kalman dual estendido.

Tendo em atenção as suas propriedades e pela sua adequação a serem aplicadas em-linha ou em-diferido optou-se por:

- i) se for conhecido, para o processo a identificar, um modelo de estados aproximado ou uma função de transferência, este conhecimento prévio pode ser incorporado, uma vez que o modelo neuronal pode ser interpretado como um modelo linear com uma parte não linear, como se verá futuramente;
- ii) admitindo a existência de um conjunto de dados relativos ao processo, é efectuada uma estimação prévia dos pesos da rede. Para tal usa-se o método de Levenberg-Marquardt;
- iii) após esta estimação inicial procede-se à actualização em-linha dos parâmetros da rede utilizando técnicas de observação, a apresentar na secção 4.2.

#### Estimação Em-Diferido

O caso particular das redes neuronais, deve ser considerado com atenção especial uma vez que o problema de estimação é muitas vezes mal condicionado especialmente

quando existe um número elevado de neurónios (Saarinen *et al.*, 1993). Por esta razão, algoritmos que usam apenas informação de primeira ordem, tais como os de gradiente, são particularmente lentos. Recomenda-se, por outro lado, a implementação de técnicas de regularização (Pedersen, 1997).

Em termos práticos, se o número de parâmetros a determinar é reduzido (da ordem das dezenas), o método de Gauss-Newton ou de Levenberg-Marquardt são eficientes. A memória necessária para este tipo de algoritmos é contudo elevada, proporcional ao quadrado do número de parâmetros. Para um número elevado de parâmetros (da ordem das centenas) aconselha-se o método de gradiente conjugado (Moller, 1993). Este caracteriza-se por ser um método de gradiente onde em cada iteração a direcção de busca é perpendicular à seguida no instante imediatamente anterior (Johansson *et al.*, 1992), (Kalman e Kwasny, 1997). A memória necessária é apenas proporcional ao número de parâmetros a determinar e os resultados são, regra geral, razoáveis. Se o número de parâmetros a determinar é intermédio referem-se os métodos de quasi-Newton.

Estas considerações justificam a opção pelo método de Levenberg-Marquardt como meio de determinar uma estimativa inicial para os parâmetros da rede.

Seja um conjunto de dados  $Z$ , constituído por  $N$  pares de valores de entradas e de saídas.

$$Z = [ U(k) Y(k) ] \quad (2.3-46)$$

Sejam estes vectores definidos por

$$U(k) = [ u(k), u(k-1), \dots, u(k-N+1) ]^T \quad (2.3-47)$$

$$Y(k) = [ y(k), y(k-1), \dots, y(k-N+1) ]^T \quad (2.3-48)$$

Considere-se ainda o critério, que inclui um termo de regularização, dado por:

$$J(W) = \frac{1}{N} \sum_{i=1}^N e_{y(k)}^T e_{y(k)} + \frac{\alpha}{2} (W - W^i)^2 \quad (2.3-49)$$

Usando o algoritmo de Levenberg-Marquardt, os parâmetros são determinados através do seguinte processo iterativo:

$$W^{i+1} = W^i + (H_a^i + \alpha^i I)^{-1} G^i \quad (2.3-50)$$

em que  $W^i$ ,  $G^i$  e  $H_a^i$  representam, respectivamente, os parâmetros a estimar, o Jacobiano e a aproximação da Hessiana  $H^i$ , em cada iteração  $i$ . A constante  $\alpha^i > 0$ , é determinada de forma a garantir que a matriz  $(H_a^i + \alpha^i I)$  seja positiva definida, (Press *et al.*, 1992).

## Estimação Em-Linha

Apesar de partirem de uma formulação distinta os métodos *EKF* e *RTRL* são, em alguns pontos, análogos. De facto, Williams (1992) mostrou que o algoritmo *EKF*, se simplificado, é idêntico ao *RTRL*: o *RTRL* é basicamente um método de gradiente com um ganho de adaptação constante, enquanto o *EKF* é um método de gradiente de ganho variável.

E, embora os algoritmos baseados no *EKF* tenham uma complexidade semelhante à do *RTRL*, oferecem melhores resultados de convergência do que este último (Williams, 1992). Uma justificação para o facto é que o *EKF* modifica de forma equivalente os pesos e os estados da rede, ou por outras palavras, os valores da unidade interna (estados) podem ser considerados directamente no processo de estimação. Outra justificação, é que o *EKF* implementa directamente a técnica de *Teacher Forcing*, proposta por Williams e Ziepsner (1989a), que consiste em usar os valores de saída do próprio processo para efectuar a retropropagação do erro em detrimento dos da rede, como é usualmente efectuado. Analisando a ideia de outra perspectiva, pode-se dizer simplesmente que usa o conceito de modelo série-paralelo.

No entanto, apesar de se reconhecerem vantagens aos métodos do tipo *EKF*, reconhece-se também que uma das seus principais desvantagens consiste no seu peso computacional, que representa um obstáculo à sua aplicação em redes de elevada dimensão. Contudo, certas simplificações, de que são exemplos a metodologia de Puskorius e Feldkamp, permitem, até certo ponto, ultrapassar estas dificuldades. Adicionalmente, o método *EKF* requer o conhecimento das matrizes de covariância do ruído, o que não é realista em modelos do tipo *black-box*. E, principalmente, porque o processo de estimação é efectuado com base numa linearização, o processo iterativo pode conduzir a resultados indesejados, podendo mesmo divergir, em casos extremos.

Assim, apesar do algoritmo *DEKF* não ser ele próprio o algoritmo escolhido nesta dissertação para a estimação de parâmetros de redes recorrentes, ele inspira a ideia que está na base para levar a cabo esta tarefa.

Desta forma, a ideia de conciliar as redes recorrentes, entendê-las como um modelo de estados e a partir daí recorrer à teoria de observação para estimar os seus parâmetros é a que se segue nesta dissertação. A estrutura a utilizar será análoga à da Figura 2.3-3, através da qual se determinam os parâmetros do modelo (pesos da rede) e simultaneamente se efectua a estimação dos seus estados (saída dos neurónios da camada interna). É, contudo, preocupação fundamental desenvolver algoritmos capazes de assegurarem a estabilidade e convergência do processo de estimação, a serem apresentados na secção 4.2, um dos inconvenientes do algoritmo *DEKF*.

## 2.4

# Conclusão

**N**a última década tem-se verificado uma actividade intensa no uso de redes neuronais como ferramentas computacionais utilizadas na modelização de sistemas e em particular de sistemas não lineares. De facto, o uso de redes neuronais permite reduzir, de forma considerável, o tempo gasto na caracterização, entenda-se modelização, de um sistema. Sendo uma rede neuronal um aproximador universal, é possível explorar as suas capacidades em aproximar funções arbitrárias ou, por outras palavras, em captar o comportamento de um determinado sistema.

De entre as várias estruturas neuronais, uma das que se mostra mais adequada ao problema em estudo nesta dissertação é a recorrente. Estas arquitecturas têm a capacidade de incorporar na sua estrutura memória, podendo, assim, descrever com naturalidade o comportamento de sistemas dinâmicos.

Os modelos neuronais dinâmicos, interpretados como modelos de estados não lineares, surgem, assim, como uma representação adequada para o futuro projecto de sistemas de controlo de sistemas não lineares. E, como observado por Zbikowski (1994), a grande vantagem dos modelos neuronais, quando aplicados no contexto do controlo, reside na sua generalidade, isto é, a possibilidade de desenvolver um metodologia sistemática e genérica para o controlo de sistemas não lineares, aspecto que tinha apenas sentido falar para sistemas lineares, usando conceitos convencionais.

É pois segundo esta direcção, as redes recorrentes entendidas como modelos de estados, que esta dissertação avançou. Outra das razões, para além dos aspectos de controlo que motivam esta interpretação, diz respeito à estimação dos parâmetros da rede. De facto, o sucesso que as técnicas de observação têm tido na estimação de estados e parâmetros de sistemas dinâmicos não lineares, nomeadamente o filtro de Kalman estendido, fundamentam o seu enquadramento em estruturas neuronais.



# 3.

## Redes Neurais

# Aplicadas ao Controlo

*“There are opportunities to make control systems with significantly increased capability by combining methodologies from feedback control and artificial intelligence. Ideas, methods and hardware are available, and it appears that control engineers have many possibilities to make useful and interesting systems. It is not clear if such systems are best labeled as intelligent.”*

K. Åström, 1996.

**P**ara sistemas lineares invariantes no tempo, as técnicas clássicas de controlo proporcionam bases sólidas para o estudo das propriedades fundamentais de controlabilidade, observabilidade e estabilidade, assim como de metodologias para o projecto de controladores. Durante as últimas décadas têm sido efectuados avanços significativos no contexto de controlo adaptativo, em particular para o caso de sistemas lineares com estrutura fixa e de parâmetros desconhecidos, tendo sido neste caso estabelecidos resultados de estabilidade global. Contudo, as técnicas baseadas em modelos lineares, mesmo aplicadas em esquemas adaptativos, não são facilmente estendidas a sistemas não lineares. Além disso, não existem para sistemas não lineares técnicas genéricas, quer para estudar as suas propriedades, ou para efectuar o projecto de controladores, quer em relação ao estudo da sua estabilidade.

Por outro lado, os resultados obtidos nesta última década têm demonstrado que as técnicas de *soft computing* são particularmente eficientes na identificação e controlo de sistemas não lineares e complexos. De entre deste tipo de técnicas destacam-se as redes

neurais principalmente pelas suas capacidades de aproximação e pelo tipo particular de estrutura (modelos paramétricos com não linearidades bem definidas) que permitem encarar o problema do controlo de uma forma genérica. E, se a sua aplicação ao controlo se centrou, numa fase inicial, mais em aspectos práticos do que teóricos tem-se assistido a vários esforços no sentido de investigar propriedades relacionadas com a estabilidade e a convergência. Esta dissertação caminha nessa direcção, e um dos objectivos é dotar estruturas neurais com mecanismos de adaptação e controlo baseados em teorias de estabilidade.

Assim, e uma vez apresentados no capítulo anterior algumas estruturas neurais e métodos capazes de estimar o valor dos seus parâmetros, aborda-se agora o problema do projecto de um controlador. De entre as possíveis soluções a teoria de controlo não linear disponibiliza um método prático e eficaz de resolver o problema de controlo de um sistema quando este se encontra descrito por um modelo não linear, como é o caso de uma rede neuronal recorrente.

No entanto, assumir que o modelo neuronal é fixo, condiciona à partida a flexibilidade da solução. Na verdade, o projecto de um regulador baseado no modelo não assegura necessariamente a convergência do erro do sistema real caso existam variações da dinâmica, perturbações ou outros factores que o modelo não tenha em conta. Daí a razão para o uso de uma estratégia adaptativa, capaz de ajustar permanentemente os parâmetros da rede neuronal e consequentemente os do controlador.

Tendo em vista enquadrar a arquitectura final de controlo utilizada, inserida no contexto mais vasto do controlo neuronal e simultaneamente apresentar os trabalhos conduzidos ao longo desta dissertação, divide-se este capítulo em quatro secções distintas.

Na secção 3.1 são apresentados aspectos genéricos de controlo, adaptativo ou não e como as redes neurais podem ser inseridas em estruturas de controlo.

Na secção 3.2 propõe-se uma classificação de arquitecturas de controlo que utilizam redes neurais, servindo ao mesmo tempo dois propósitos: introduzir os vários trabalhos realizados ao longo da elaboração da dissertação; permitir enquadrar e justificar a estrutura final desenvolvida, caracterizada por uma metodologia de controlo não linear híbrida, classificada em termos de controlo convencional como adaptativa indirecta.

Na secção 3.3 introduzem-se algumas técnicas de controlo não linear, em particular a linearização retroactiva, com potencialidades de aplicação a redes recorrentes. Revêem-se nesta secção trabalhos de controlo adaptativo não linear que incorporam na sua estrutura modelos neurais e introduz-se a técnica de regulação da saída, como uma alternativa a implementar à linearização retroactiva.

Termina-se na secção 3.4 referindo algumas conclusões.

# 3.1

## Controlo de Sistemas

*Introduzem-se nesta secção técnicas de controlo adaptativo e mostra-se como as redes neuronais podem ser incorporadas em estruturas de controlo convencional.*

*Se por um lado podem ser entendidas como uma simples extensão de modelos lineares a modelos não lineares, por outro, são capazes de dotar os sistemas de controlo com as capacidades que lhes são inerentes (complexidade, não linearidade e incertezas).*

### 3.1.1 Introdução

Por controlo de um sistema entende-se as acções efectuadas de forma a forçar a sua dinâmica a seguir uma evolução desejada. Nesse sentido, as variáveis de entrada,  $u(k)$ , são entendidas como as sujeitas a manipulação no sentido de conduzir as saídas  $y(k)$ , a um determinado valor desejado,  $y_d(k)$ . Na Figura 3.1-1, ilustra-se o problema de controlo, usando o conceito de realimentação. O controlador recebe as medidas da saída  $y(k)$  e/ou dos estados  $x(k)$  e, em função dos objectivos a atingir, determina a acção de controlo  $u(k)$  a aplicar ao sistema.

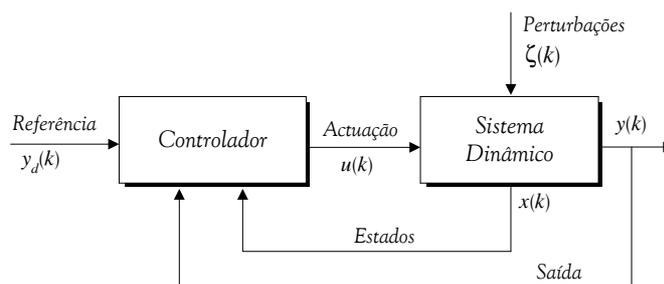


Figura 3.1-1: Sistema de controlo por realimentação.

Dependendo da forma particular do controlador, isto é, do modo como é usada a informação para determinar a acção de controlo, assim se distinguem os vários tipos de controladores. Os controladores ditos convencionais, de que são exemplos os

controladores por colocação de pólos ou os controladores de realimentação de variáveis de estado, entre outros, para os quais existe uma teoria bem estabelecida, têm por base uma série de pressupostos (Wellstead e Zarrop, 1991). Assumem a linearidade do sistema a controlar; assumem o seu conhecimento exacto, expresso segundo a forma de um modelo matemático, regra geral uma função de transferência ou um modelo de estados; baseiam-se ainda no pressuposto de que os seus parâmetros são constantes, sendo portanto invariantes ao longo do tempo.

Só que assumir que qualquer sistema pode ser descrito por um modelo linear de parâmetros fixos pode, em muito casos, não ser realista e, mesmo que isso fosse verdade, nem sempre é fácil obter esses parâmetros com precisão. Além do mais, e uma vez determinados, muito dificilmente os parâmetros de um sistema permanecem inalteráveis, seja devido ao desgaste sofrido pelos componentes do sistema, seja por variações ambientais. Nestas condições, cumprir o objectivo de controlo, utilizando um controlador convencional, pode tornar-se numa tarefa bastante complicada e frustrante.

Duas das técnicas de controlo, capazes de ter em conta incertezas de parâmetros, são as robustas e as adaptativas, distinguindo-se principalmente por na primeira os parâmetros do controlador serem fixos e na segunda serem variantes.

A teoria de controlo robusto trata *a priori* o problema das incertezas, assumindo previamente que estas pertencem a uma determinada classe. O projecto do controlador (invariante), cujos parâmetros são determinados através da minimização de um critério geralmente do tipo  $H_2$  ou  $H_\infty$ , é depois efectuado de modo a que o sistema de controlo resulte, em certa medida, insensível à variação das incertezas (Morari e Zafiriou, 1989). Na prática as técnicas de controlo robusto são de certo modo restritivas, uma vez que assumem o conhecimento prévio das incertezas, entre as quais se encontram as variações de parâmetros sendo, além disso, estes métodos principalmente aplicáveis a sistemas lineares

Ao contrário das técnicas robustas, as técnicas adaptativas tratam o problema das incertezas a partir de uma identificação permanente das características do sistema a controlar.

### 3.1.2 Controlo Convencional Adaptativo

Um controlador adaptativo convencional caracteriza-se, fundamentalmente, por se basear num modelo linear de estrutura fixa cujos parâmetros são automaticamente ajustados, sempre que existam variações nas propriedades ou características do sistema que está a ser controlado. Assim, existe sempre a possibilidade de o sistema de controlo compensar mudanças imprevisíveis que ocorram no processo. Se as características do sistema forem permanentemente identificadas e existir paralelamente um cálculo dos parâmetros do controlador, pode então pensar-se que o controlador também se adaptará a variações ocorridas no sistema e desta forma obter-se-á um desempenho satisfatório, quaisquer que sejam as condições de operação do sistema. Na Figura 3.1-2 representa-se um sistema de controlo adaptativo.

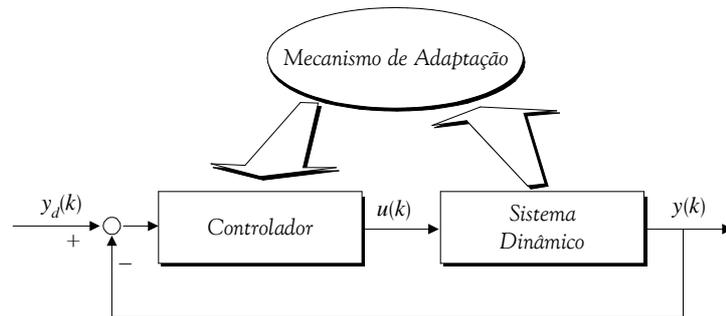


Figura 3.1-2: Sistema de controlo adaptativo.

Conforme se pode constatar na figura anterior, o sistema é composto fundamentalmente por duas malhas de realimentação, uma com o objectivo de estimar os parâmetros do sistema (mecanismo de adaptação) e outra que faz o controlo propriamente dito. A malha de adaptação confere ao sistema de controlo potencialidades para lidar com variações de parâmetros e variações na dinâmica do processo, enquanto que a malha de realimentação permite manter desempenhos aceitáveis do sistema de controlo.

### Origens do Controlo Adaptativo

As origens do controlo adaptativo remontam aos finais dos anos 50. Gregory (1958), motivado pelo projecto de pilotos automáticos de aeronaves que necessitavam de operar em condições bastante diversas de velocidade e altitude, estabeleceu uma nova filosofia de controlo. No seu trabalho, *Adaptive Flight Control*, Gregory concluiu da incapacidade de operação de um controlador de ganhos constantes para todas as gamas de operação. Introduziu o método de escalonamento de ganhos (*gain scheduling*), que consistia basicamente na definição de parâmetros que permitissem caracterizar a dinâmica do sistema e, com base nestes, proceder a um escalonamento (selecção) do controlador adequado. Contudo, a aplicação das suas ideias foi limitada devido à inexistência de uma base teórica, capaz de entender as propriedades e as características do seu algoritmo e porque a implementação de um controlador era, naquele tempo, necessariamente analógica. Além disso, os resultados obtidos foram também, de certo modo, desanimadores.

Na tentativa de solucionar um problema semelhante Whitaker introduziu em 1958 uma outra ideia, o controlo adaptativo por modelo de referência MRAC<sup>1</sup>. Basicamente era definido um modelo de referência (considerado ideal) e com base na diferença entre a saída real e a saída deste último modelo (erro) eram adaptados os parâmetros do controlador. Whitaker desenvolveu ainda uma regra segundo a qual deveriam ser adaptados os parâmetros do controlador, conhecida por regra MIT (por ter sido desenvolvida no *Massachusetts Institute of Technology*).

---

<sup>1</sup> MRAC- Model Reference Adaptive Control

A existência de ruídos impôs a necessidade de uma solução estocástica para os problemas de estimação e controlo, o que levou Kalman (1960) a apresentar de uma forma analítica o conceito de controlo adaptativo auto ajustável *STAC*<sup>2</sup>. Usando o princípio de separação, era efectuada uma estimação explícita dos parâmetros do sistema e usada como lei de controlo uma lei de controlo óptima, *LQR*<sup>3</sup>.

Os anos 60 caracterizaram-se por grandes desenvolvimentos na área do controlo e em particular do controlo adaptativo: a definição do conceito de estado e a aplicação da teoria de estabilidade, antes introduzida por Lyapunov, possibilitou o estudo das propriedades de convergência das estruturas adaptativas. A teoria do controlo estocástico deu passos importantes e a introdução da programação dinâmica, (Bellman, 1957), permitiu consolidar o conhecimento dos processos adaptativos. Tsypkin (1971), mostrou que um problema de controlo adaptativo podia ser expresso à custa de uma equação recursiva. Åström e Wittenmark (1973) contribuíram para que a identificação em-linha, inserida numa estrutura de controlo adaptativo, fosse finalmente percebida e correctamente formulada.

A evolução tecnológica dos anos 70, aliada ao desenvolvimento teórico dos anos 60, permitiu a implementação prática de controladores adaptativos digitais, fazendo renascer o interesse pelo controlo adaptativo, em particular pelas metodologias de controlo indirecto e directo, de que são exemplos respectivamente a metodologia auto ajustável e a modelo de referência.

### **Controlo Adaptativo Indirecto**

Um controlador adaptativo auto ajustável caracteriza-se por ser efectuada em cada instante uma estimação dos parâmetros do sistema a controlar. Depois, usando o princípio da equivalência certa (os parâmetros são assumidos como se fossem correctos) e tal como se se tratasse de um controlador convencional, é projectado o controlador segundo uma certa lei ou critério (*PID*, colocação de pólos, variância mínima, etc.). Generalizando, todos os controladores cujos parâmetros dependem dos da função de transferência do processo são possíveis de utilizar. Porque se assume que a dinâmica e, consequentemente, os parâmetros que caracterizam o sistema podem variar, os do controlador também variarão, tendo por isso sentido falar de controladores com parâmetros variáveis ou ajustáveis. O diagrama de blocos da Figura 3.1-3 ilustra esquematicamente esta estrutura de controlo.

---

<sup>2</sup> *STAC- Self Tuning Adaptive Control*

<sup>3</sup> *LQR- Linear Quadratic Regulation*

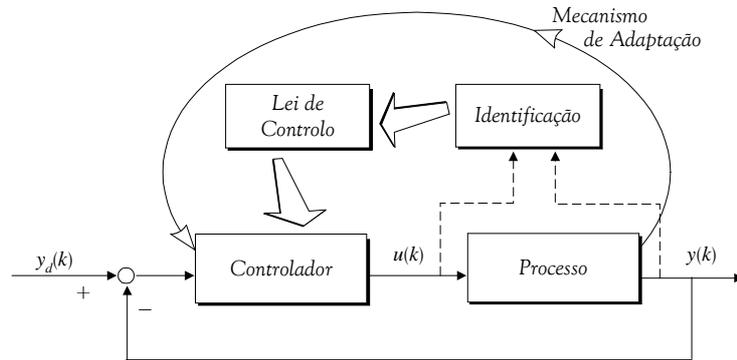


Figura 3.1-3: Controlador adaptativo indirecto ou auto ajustável.

Como se referiu, a ideia básica foi proposta por Kalman e depois desenvolvida por Chang e Rissaen que propuseram o uso do método dos mínimos quadráticos para a estimação dos parâmetros e a lei de variância mínima como lei de controlo. Åström e Wittenmark (1973) propuseram um algoritmo com o objectivo de minimizar a variância da saída de um sistema, descrito por um modelo ARMAX<sup>4</sup>, que mais tarde é redefinido por Clarke e Gawthrop (1975), introduzindo o conceito de variância mínima generalizada. Edmunds introduziu em 1976 a ideia de controlo por colocação de pólos, capaz de permitir uma especificação dos pólos do sistema em malha fechada, definindo à partida a estabilidade do sistema de controlo. Wellstead *et al.* (1979), estenderam os resultados de Edmunds a sistemas de fase não mínima, sem efectuar o cancelamento de zeros, e Åström (1983) generaliza os controladores por colocação de pólos. Os resultados apresentados por Åström permitem efectuar um cancelamento apenas dos zeros estáveis (no interior do circulo de raio unitário) e podem ser aplicados não só a problemas de regulação mas também aos problemas de perseguição.

#### Controlo Adaptativo Directo

A ideia básica deste tipo de metodologia, ilustrada na Figura 3.1-4, consiste em considerar um modelo de referência, isto é, um modelo representativo da dinâmica considerada ideal para o sistema a controlar  $y_d(k)$ , comparar o valor da saída deste modelo (saída desejada) com o valor da saída real do sistema  $y(k)$  e, com base nesta comparação  $e(k)$  (erro), definir uma regra de adaptação para os parâmetros do controlador. O objectivo final é forçar o erro a ser nulo, portanto a saída real do sistema a seguir a do modelo de referência.

<sup>4</sup> ARMAX- AutoRegressive Moving Average with external input

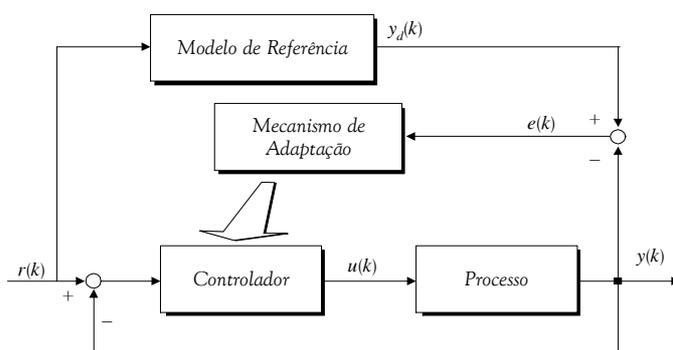


Figura 3.1-4: Controlador adaptativo directo ou por modelo de referência.

O controlador adaptativo por modelo de referência teve a sua origem na formulação de um problema em tempo contínuo. Depois de Whitaker, Parks (1966) usando a representação no espaço de estados e a teoria de estabilidade de Lyapunov, reescreveu as leis de adaptação MIT, resolvendo o problema de estabilidade não solucionada por Whitaker. Narendra e Valavani (1980a) e (1980b), solucionaram o problema da estabilidade global de um sistema de controlo adaptativo aplicado a um sistema contínuo linear, definido no espaço de estados ou por uma função de transferência. Por outro lado, Landau (1979) propôs o uso da teoria da hiper-estabilidade (introduzida por Popov) para provar a estabilidade de uma estrutura adaptativa, também aplicada a sistemas contínuos. Para sistemas discretos destacam-se os trabalhos de Edgart (1980), onde são unificados vários métodos, e o trabalho de Goodwin *et al.* (1980) onde são estabelecidos, para uma determinada lei de adaptação, a convergência global de um sistema MIMO linear em tempo discreto.

A estrutura de controlo por modelo de referência diz-se directa uma vez que a regra de adaptação determina directamente a forma como os parâmetros do controlador devem ser ajustados. No caso do esquema auto ajustável os parâmetros do controlador são ajustados indirectamente, isto é, primeiro obtêm-se os parâmetros do sistema e só depois com base nestes são calculados os parâmetros do controlador.

Embora o princípio de um controlador adaptativo seja simples a sua análise é no entanto complicada uma vez que ao considerar-se uma variação dos parâmetros em cada instante, tal resulta necessariamente num sistema não linear. Existem no entanto, para este tipo de metodologia, resultados bem estabelecidos, destacando-se os trabalhos de Åström e Wittenmark (1990) para o caso indirecto e o de Goodwin *et al.* (1980) para o caso directo, ambos no domínio discreto.

### Novos Desafios

Para sistemas complexos, em que existam interacções entre as suas variáveis, apresentando comportamentos não lineares, o projecto de um controlador transforma-se numa tarefa complexa por uma série de razões. Primeiro, porque não é fácil determinar uma estrutura adequada para descrever o sistema, ao contrário dos sistemas lineares, para os quais existe uma teoria bem consolidada (funções de transferência ou modelos de estados). Segundo, porque não existe uma metodologia de controlo que permita lidar

com sistemas não lineares de uma forma genérica, tal como existe para sistemas lineares. Terceiro, porque mesmo que seja estabelecida uma estrutura correcta para o sistema, há ainda que determinar os seus parâmetros a partir de dados, havendo sempre ambiguidades na sua determinação e situações não previstas que serão impossíveis de ter em conta. Em conclusão, e segundo Narendra (1996), as dificuldades de projecto de um controlador acabadas de referir correspondem a três tipos de factores: complexidade, não linearidades e incertezas.

Para lidar com este tipo de problemas têm sido efectuados progressos no que diz respeito às teorias de controlo adaptativo, as quais se têm mostrado como sendo técnicas viáveis na resolução de alguns problemas práticos. Contudo, as técnicas de controlo adaptativo convencionais têm as suas limitações. De facto, porque é utilizada uma estrutura linear fixa de parâmetros desconhecidos, o problema da incerteza é reduzido apenas à determinação destes parâmetros. Assim, as aplicações que melhor se adequam a esquemas de controlo adaptativo são aquelas em que se tem um conhecimento prévio do sistema, capaz de caracterizar correctamente a sua estrutura (linear). Além disso, mesmo recorrendo à adaptação de parâmetros, os modelos lineares não são suficientemente flexíveis para representar sistemas não lineares genéricos. Cada caso é um caso, dependendo das não linearidades particulares de cada problema, e face à ausência de formalismos genéricos, a teoria de controlo adaptativo linear não pode facilmente ser estendida a sistemas não lineares.

As técnicas adaptativas de controlo, para sistemas não lineares, necessitam de novas soluções e as metodologias baseadas em técnicas de *soft computing* tem permitido efectuar avanços nesse sentido.

### 3.1.3 Controlo Baseado em Técnicas de *Soft Computing*

A introdução dos computadores digitais, nos anos 70, abriu novas perspectivas em todas as áreas da Engenharia. Em particular, possibilitou à comunidade do Controlo desenvolver teorias e aplicações que muito dificilmente poderiam ser implementados com componentes analógicos, de que é exemplo a identificação em-linha. No entanto, apesar das potencialidades proporcionadas pelos computadores digitais, o tipo de informação utilizado permaneceu essencialmente do mesmo tipo (dados de sensores, isto é, quantitativa).

Motivados pelo desenvolvimento extraordinário dos computadores e pelas capacidades por eles oferecidas, começou a dar-se importância ao desenvolvimento de sistemas capazes de efectuar uma gestão efectiva da informação existente: adequados a processar não só grandes quantidades de informação mas também tratar diferentes tipos de informação e, a partir desta, criar conhecimento de forma a permitir tomar decisões.

Neste contexto, surge o conceito de *soft computing*, caracterizado pela associação de metodologias de computação que proporcionem bases para a concepção e projecto de sistemas capazes de lidar com imprecisões, com incertezas, ou aprender a partir de dados

e de se adaptar às condições de operação (Zadeh, 1994). A aplicação destas ferramentas a problemas de controlo dá origem a uma nova área de investigação, surgindo o conceito de *Controlo Inteligente*.

Como técnicas de *soft computing* destacam-se as redes neuronais, originalmente desenvolvidas para reproduzir o processo biológico de aprendizagem humano (usando informação quantitativa) e os sistemas difusos, com o objectivo de construir uma estrutura formal que permitisse captar a imprecisão do conhecimento humano, tal como ele é expresso através de linguagens naturais (informação qualitativa).

E, se durante os últimos 20 anos têm sido desenvolvidos esforços no sentido de compreender as potencialidades destas metodologias e a forma de proceder eficazmente à sua incorporação em estruturas de controlo, a investigação de teorias e métodos convencionais baseados em modelos matemáticos bem definidos (equações diferenciais) não estagnou. Como consequência o estado actual do controlo automático enfrenta três paradigmas (Dourado, 1997): o paradigma integro-diferencial (modelos matemáticos), o paradigma de dados (informação quantitativa) e o paradigma linguístico (informação qualitativa).

Assiste-se actualmente a uma série de esforços no sentido de estabelecer pontes entre os vários paradigmas e de desenvolver abordagens unificadoras. Neste contexto, o termo controlo inteligente é usado frequentemente, contudo, a sua definição está longe de se encontrar bem estabelecida. Tem havido no entanto várias tentativas de caracterizar o seu significado (Baker, 1992), (Passino, 1993) e (Antsaklis, 1994), mas não existe ainda consenso para a sua definição. Dourado (2000) define um controlador inteligente da seguinte forma:

*“ Um controlador é inteligente se consegue captar alterações ocorridas no ambiente ou no processo controlado e adaptar-se a essas alterações de forma a manter o desempenho do sistema total de controlo inalterável. “*

Note-se que, contudo, o termo adaptação é em termos convencionais associado essencialmente à estimação de parâmetros. Assim, de forma a dar-lhe um significado mais abrangente usa-se por vezes o termo aprendizagem como uma extensão da estimação de parâmetros (espaço de modelos) para o espaço genérico de informação. E da mesma forma que o conceito de controlo inteligente é ambíguo também o é o de aprendizagem. Na verdade as definições de aprendizagem e adaptação têm sido usadas na literatura de uma forma bastante confusa. Åström (1989) refere-se ao termo adaptação como a variação do comportamento em função de novas circunstâncias e ao termo aprendizagem como implicando necessariamente um aumento e transferência de conhecimento. Isto é, o conceito de aprendizagem é mais vasto e diz respeito à capacidade de um sistema em melhorar o seu desempenho tanto para novas como para situações passadas. O conceito de aprendizagem relaciona-se portanto com o conceito de generalização que, de alguma forma, consiste em construir uma representação do conhecimento global e não local, como no caso da adaptação. Evidentemente que o conceito de aprendizagem é mais atraente, mas inevitavelmente mais complexo.

Relativamente ao controlo, pode-se dizer que, em certa medida, um controlador adaptativo usado para ajustar os seus parâmetros em função de uma nova situação, adquire conhecimento que se reflecte no valor dos seus parâmetros. Contudo, mais tarde, se uma situação semelhante ocorrer novamente, o mecanismo de adaptação recalculará novamente os parâmetros uma vez que nenhuma informação passada foi armazenada. Neste sentido pode-se dizer que na verdade nada foi aprendido, ou por outras palavras, os controladores adaptativos não têm memória de longa duração e por isso são incapazes de “recordar” os parâmetros já determinados em situações passadas.

Seguindo esta definição, terá sentido falar de aprendizagem (entendida como a capacidade de armazenar informação e de a utilizar depois), utilizando técnicas de *soft computing* em problemas de classificação ou aproximação de funções estáticas, se a estimação de parâmetros for efectuada em-diferido. No entanto, para o caso particular em estudo, a identificação em-linha de modelos neuronais, os termos aprendizagem e adaptação dos pesos de uma rede são termos perfeitamente equivalentes. Portanto, segundo este ponto de vista, o que está envolvido nesta dissertação é o conceito de adaptação e não aprendizagem de parâmetros. Abra-se aqui um parêntesis para referir que, apesar da aparência em contrário, não se conhece nenhum trabalho que efectue aprendizagem em-linha dos pesos de uma rede, todos fazem na verdade adaptação. Explicando melhor, perante uma qualquer situação, seja nova ou antiga (para a qual já tinha sido efectuada uma determinação de parâmetros) o mecanismo de adaptação é “cego” e torna a adaptar os parâmetros como se de uma nova situação se tratasse.

### **Técnicas de *Soft Computing***

#### *Redes Neuronais*

Como definido no capítulo 2, as redes neuronais são sistemas compostos por várias unidades de processamento simples (neurónios) operando em paralelo e definidas pela sua arquitectura, pesos de ligação e tipo de processamento efectuado em cada unidade. Como característica principal é a sua capacidade de aprender a partir de exemplos que as torna adequadas a problemas de aproximação, como a modelização de sistemas (capítulo 2) e o uso em esquemas de controlo, a abordar na secção 3.2.

#### *Sistemas Difusos*

Outra das técnicas que tem alcançado um manifesto sucesso no contexto do controlo inteligente são as técnicas que utilizam lógica difusa, um conceito matemático capaz de lidar com incertezas, conceito esse inicialmente introduzido por Zadeh (1965).

Além de poderem ser aplicados na modelização de sistemas, refere-se a primeira aplicação ao controlo de sistemas por Mamdani e Assilian (1975). A lógica difusa faculta um meio de converter o conhecimento linguístico numa estratégia de controlo. De facto, as acções de controlo fornecidas por um controlador difuso são definidas por meio de uma série de regras difusas que expressam de uma forma qualitativa as dependências entre as variáveis do sistema a controlar. Assim, particularmente em sistemas complexos, cuja análise e projecto são difíceis seguindo abordagens tradicionais quantitativas, mas que na prática podem ser controlados directamente por um operador, o problema pode

ser formulado com relativa facilidade seguindo uma abordagem qualitativa, através do recurso à lógica difusa.

Na Figura 3.1-5 mostra-se um esquema genérico de um controlador difuso, composto por quatro unidades fundamentais: a unidade de fuzificação, a base de conhecimento, o motor de inferência e a unidade de desfuzificação.

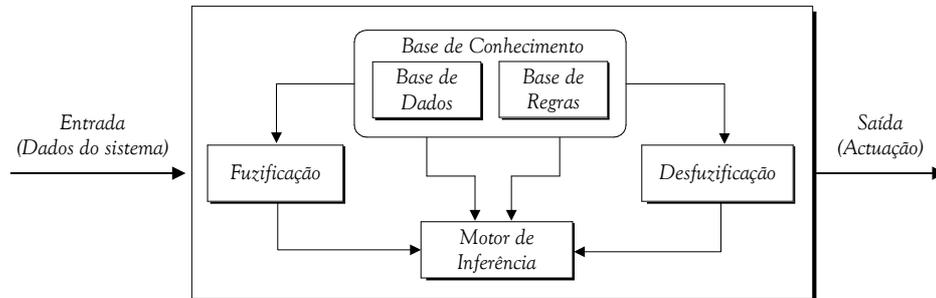


Figura 3.1-5: Sistema de controlo difuso.

A unidade de fuzificação é responsável por converter um valor numérico de entrada num valor difuso. Por outro lado, a base de conhecimento é composta por um conjunto de regras (base de regras) e por uma base de dados, que contém informação que os outros módulos necessitam para funcionar, tal como a informação sobre as funções de pertinência, factores de escala, entre outros. As regras, que definem o funcionamento do controlador, são do tipo (3.1-1).

$$\text{Se (estado do sistema) então (actuação sobre o sistema)} \quad (3.1-1)$$

A primeira parte das regras (*estado do sistema*) é definida como premissa ou antecedente; a segunda parte (*actuação sobre o sistema*) como conclusão ou conseqüente. Refira-se que pode ser necessário, e usualmente assim acontece, proceder à conjunção e disjunção de regras. Distinguem-se dois tipos de regras difusas: linguísticas ou de Takagi-Sugeno. Nas primeiras, tanto o antecedente como o conseqüente são conjuntos difusos, nas segundas, os antecedentes são conjuntos difusos e os conseqüentes são valores numéricos, cujo resultado é obtido em função dos antecedentes. Note-se que é possível classificar os controladores difusos em função do tipo de regras que utilizam. Assim fala-se de controladores de Mamdani, que utilizam regras linguísticas e de controladores de Sugeno que utilizam regras de Takagi-Sugeno.

É função do motor de inferência determinar o valor difuso de saída, com base nos princípios estabelecidos na fuzificação e na base de conhecimento, podendo ser baseado em regras individuais ou na sua composição.

Por fim, a unidade de desfuzificação usa o resultado obtido pelo motor de inferência e converte-o num valor numérico, a ser aplicado como sinal de actuação ao sistema.

Tal como nos sistemas de controlo adaptativo convencional é possível dividir os controladores adaptativos difusos em duas classes principais: directos e indirectos, consoante os parâmetros do controlador difuso sejam directamente adaptados em função do erro ou caso os parâmetros do processo sejam previamente determinados, sendo

depois os parâmetros do controlador determinados com base nos do processo. É comum contudo classificar os controladores adaptativos em função dos parâmetros que se pretendem adaptar. Assim, referem-se os controladores auto-sintonizados, quando se adaptam as funções de pertinência ou factores de escala e os controladores auto-organizados, quando é a base de regras que é modificada, ou quando se parte de um conjunto inicial vazio de regras e se constrói a estratégia de controlo à medida que se controla o processo, (Ramos, 1998).

#### *Algoritmos Genéticos*

Outra metodologia utilizada no contexto do controlo inteligente, são os algoritmos genéticos. Um algoritmo genético consiste essencialmente num método de pesquisa baseado num mecanismo de selecção natural, tendo sido inicialmente proposto por Holland (1975). Pelas suas características, este tipo de algoritmos é aplicado fundamentalmente a problemas de optimização, apresentando como grande vantagem a sua capacidade (teórica) de obter soluções óptimas globais para problemas complexos, tendo no entanto o grande inconveniente de serem demasiado morosos a alcançar uma solução aceitável. Assim, os algoritmos genéticos têm sido explorados no contexto de controlo especialmente em situações que envolvam uma optimização de parâmetros em-diferido, quer se trate de redes neuronais ou de sistemas difusos.

Aplicados a problemas matematicamente mal definidos, os algoritmos genéticos têm sido utilizados com sucesso na determinação automática de parâmetros de sistemas difusos, seja de funções de pertinência (Meredith *et al.*, 1993), ou do conjunto de regras (Karr *et al.*, 1991), ou mesmo de ambas (Linkens e Nyongesa, 1996).

Da mesma forma, e para redes neuronais, uma aplicação imediata é o seu uso como uma alternativa aos métodos de gradiente para determinar os pesos da rede (Machado e Rocha, 1992), (De *et al.*, 1995). Podem ainda ser aplicados na determinação da arquitectura, incluindo o número de camadas, o número de unidades de processamento assim como os pesos da rede.

#### **Opção pelas Redes Neuronais**

Relativamente aos algoritmos genéticos, e tendo em atenção os objectivos inicialmente propostos, estes foram à partida eliminados. De facto, os algoritmos genéticos têm tido sucesso principalmente em aplicações em-diferido, não sendo adequados para aplicações em-linha, uma vez que o tempo de convergência a eles associado é elevado.

Entre ter de optar pelas redes neuronais ou pelos modelos difusos pensa-se que a escolha deve ser conduzida principalmente pelo tipo de conhecimento que existe ou que pode ser adquirido do processo. Assim, se existir informação qualitativa, por exemplo fornecida por um operador, certamente que a escolha adequado recairia sobre um sistema difuso. Caso o conhecimento consista em dados numéricos, como é o caso do trabalho em investigação, então a escolha não será óbvia.

Em Henriques *et al.* (1998b) são desenvolvidas e aplicadas a um processo laboratorial três metodologias de controlo, incluídas nas metodologias de controlo que usam técnicas de *soft computing*. O objectivo principal é avaliar as suas potencialidades para efeitos de

controlo, em particular de adaptação no caso de ocorrerem variações, nomeadamente, na dinâmica do processo. A primeira metodologia consiste num controlador neural do tipo *RBFNN*, sendo adaptados em-linha os centros das funções Gaussianas e os pesos da camada da saída. A segunda, consiste numa metodologia híbrida onde uma rede do tipo sigmoidal é utilizada como modelo para o processo, sendo usado um algoritmo convencional de colocação de pólos para produzir a acção de controlo. Finalmente, na terceira metodologia, implementa-se um controlador adaptativo difuso do tipo auto-organizado, sendo o mecanismo adaptativo em cada iteração realizado pela alteração dos consequentes de cada regra. Da análise dos resultados experimentais é possível concluir que, para o processo laboratorial estudado (*PT326*), qualquer das metodologias permite desempenhos aceitáveis, não sendo possível considerar, inequivocamente, uma delas superior às restantes.

No entanto, a motivação pelo desenvolvimento de estruturas híbridas, que combinem metodologias de *soft computing* com metodologias convencionais, levou a optar pelas redes neuronais. Justifica-se esta opção uma vez que a análise das suas propriedades é mais intuitiva, isto é, mais compatível com técnicas de controlo convencional que os sistemas difusos.

### **Redes Neurais como Modelos Não Lineares**

De facto, do ponto de vista do controlo convencional uma rede neuronal não é senão uma descrição matemática que permite efectuar um mapeamento não linear entrada-saída. Entendido desta forma, um modelo neuronal é um subconjunto de ambas as teorias, do lado convencional entendido como um modelo não linear, do lado de *soft computing* entendida como um modelo computacional capaz de “*aprender*” a partir de dados de operação e que, ainda que de uma forma bastante simplificada, pretende reproduzir o processo de aprendizagem humano.

Além disso, a sua aptidão para capturar não linearidades e a sua capacidade de permitir o ajuste dos seus parâmetros a partir de dados permite que sejam incorporadas em estruturas de controlo adaptativo. É neste ponto que as metodologias de *soft computing* e convencionais se tocam. Assim entendidas, as redes neuronais oferecem uma *ponte* perfeita entre as técnicas de *soft computing* e as técnicas convencionais de controlo, tornando-se natural a utilização de técnicas de controlo não lineares aplicadas a modelos neuronais.

O objectivo de um sistema de controlo adaptativo convencional (modelos lineares de estrutura fixa) é na medida do possível, ser suficientemente flexível para lidar com sistemas de parâmetros desconhecidos, apenas através da adaptação dos seus parâmetros. Adicionalmente uma rede tem em conta não apenas incertezas ao nível dos parâmetros, mas também não linearidades. Isto é, apesar da sua estrutura ser fixa, lidam directamente com não linearidades, pois possuem a capacidade (pelo menos teórica) de reproduzir qualquer tipo de não linearidade. E esta é, sem dúvida, uma das grandes vantagens das redes neuronais, ou seja, a sua generalidade.

Um argumento contrário ao uso de redes neuronais em esquemas adaptativos é justificada em (Hrycej, 1997) através do facto de que um controlador adaptativo só faz sentido caso o modelo a adaptar seja linear, não existindo vantagens adicionais caso seja

não linear. De forma a investigar este problema é efectuado em Pereira e Henriques (Pereira *et al.*, 2000) um estudo experimental entre duas estratégias de controlo adaptativo: uma baseada num modelo linear e outra num modelo neuronal (não linear). Na primeira estratégia são usadas técnicas convencionais de controlo linear e na segunda é usada uma rede *RBFNN*, ambas inseridas em estruturas de controlo adaptativo. Para o caso linear é usado um modelo do tipo *ARX*, sendo os parâmetros estimados em-linha através de um método de esquecimento selectivo (Parkum *et al.*, 1992), e para lei de controlo implementa-se um algoritmo de colocação de pólos. No caso neuronal, a aprendizagem é efectuada seguindo uma estratégia mista, sendo os parâmetros que caracterizam as funções Gaussianas obtidas por um processo de *clustering* (Moody, 1989) e os pesos da camada de saída pelo mesmo processo de estimação usado no modelo linear. As metodologias são aplicadas a um processo laboratorial (*PT326*) cujos resultados mostram o superior desempenho da estratégia neuronal, principalmente quando são introduzidas variações na dinâmica do processo, validando-se, assim, para este caso particular, a opção por modelos neuronais não lineares em estratégias de controlo adaptativo.

Para concluir, se para sistemas lineares adaptativos existem resultados genéricos de estabilidade, tal não acontece para sistemas de controlo neuronal adaptativo. Assim pensa-se que além de outros aspectos para os quais ainda não existem respostas definitivas, tais como: o tipo de funções de activação a utilizar, como efectuar a determinação dos pesos da rede ou como seleccionar o conjunto de dados de treino, a investigação deve, preferencialmente avançar no sentido do desenvolvimento de estruturas de controlo que incorporem redes neuronais, de forma a tirar partido das suas potencialidades, mas que ao mesmo tempo sejam baseados em técnicas bem estabelecidas de controlo, capazes de assegurar, entre outros, resultados de estabilidade e convergência,

## 3.1.4 Redes Neurais no Controlo

### Passado

Foi apenas nos meados da década de 60 que se refere pela primeira vez a aplicação das redes neuronais no controlo. Concretamente, Widrow e Smith (1963) utilizaram uma rede *Adaline*, anteriormente proposta por Widrow, e mostraram que esta era capaz de ser usada no controlo de um pêndulo invertido. Basicamente, numa fase inicial era utilizado um controlador humano, que servia de supervisor ao treino da rede, rede essa que numa fase posterior o substituíam. Mais tarde, na década de 70, Albus (1975a) e (1975b), desenvolveu um controlador baseado numa rede neuronal específica, a que chamou *CMAC*<sup>5</sup>. A rede neuronal consistia numa memória do tipo associativa (tabela) e os seus resultados teóricos foram validados, na prática, no controlo de um robot. Nos anos 80, Barto *et al.* (1983) propuseram o método de reforço crítico, que se caracteriza,

---

<sup>5</sup> *CMAC- Cerebral Model Articulation Controller*

fundamentalmente, por utilizar um espécie de reforço na aprendizagem, em vez de uma regra típica de gradiente. Uma analogia interessante é dada por Hertz *et al.* (1991): “*um professor ensina um estudante como aprender, um crítico apenas diz se está a proceder bem ou mal*”.

Foi contudo a redescoberta do algoritmo de retropropagação (Rumelhart *et al.*, 1986) e a consequente disponibilização de uma ferramenta genérica de estimação de parâmetros de redes MLNN que impulsionou, de uma forma genérica, o desenvolvimento das redes neuronais e, em particular, que motivou a sua utilização no contexto do controlo. Um dos primeiros a verificar o interesse de incorporar redes neuronais em sistemas de controlo foi Psaltis *et al.* (1988). Propõem os autores duas arquitecturas de aprendizagem, uma generalizada e a outra especializada, com o objectivo de obter um modelo inverso do processo, a ser usado posteriormente como controlador.

Em alternativa, de forma a obter um modelo inverso, foi proposto por Kawato *et al.* (1987), o método *Feedback Error Learning*. Enquanto Psaltis obtinha o modelo inverso directamente, Kawato propôs um método indirecto. A rede era colocada previamente em paralelo com um controlador convencional e aprendia a desempenhar as funções deste último, eliminando assim a futura necessidade do uso do controlador convencional.

Outro marco importante na história do controlo neuronal foi levado a cabo por Jordan (1989) usando o algoritmo de retropropagação para treinar simultaneamente o modelo e um controlador neuronal (modelo inverso). Tal abordagem, é referida como tendo sido independentemente descoberta por Narendra e Parthasarathy (1990), Nguyen and Widrow (1990) e Werbos (Werbos, 1990). Embora Jordan e Rumelhart (1990) reclamem a autoria de uma ideia inovadora para o controlo de sistemas, esta, do ponto de vista de controlo convencional, não possui nada de novo, mostrando-se análoga a uma arquitectura de controlo adaptativo, em particular uma estrutura de controlo adaptativo indirecto. De facto, nesta arquitectura é utilizada uma rede como modelo do processo e uma outra como controlador, sendo os parâmetros da rede que desempenha as funções de controlador calculados com base no modelo. As duas estruturas são colocadas em série, servindo apenas a primeira (modelo) para permitir a retropropagação do erro até à segunda (controlador). Seguindo o mesmo princípio, Nguyen e Widrow aplicaram um esquema de treino do tipo *BPTT*, ao controlo do *truck backer-upper*, treinando uma das redes para desempenhar as funções de um controlador não linear. Werbos utilizou, pela primeira vez a palavra “*neuro-controlo*” para descrever a aplicação das redes neuronais no contexto do controlo de sistemas.

Narendra e Parthasarathy (1990) demonstraram que as redes neuronais poderiam ser usadas com eficácia no controlo de sistemas multi-variáveis não lineares. Distinguem quatro classes principais de modelos que utilizam redes neuronais do tipo MLNN, das quais a segunda e a terceira classes (classes II e III) são particularmente adequadas para o problema de controlo. De facto, estas classes podem ser entendidas como pertencentes à classe de sistemas *affine*, para as quais é possível solucionar de uma forma directa o problema de controlo, através de técnicas de linearização retroactiva (*feedback linearization*), introduzidas por Isidori (1989).

Chen (1990) propõe, independentemente de Narendra, um esquema de controlo também baseado na ideia de linearização retroactiva, mas em que o treino das redes é efectuado em-linha, ao contrário de Narendra e Parthasarathy em que o treino era em-diferido. Chen implementa na verdade uma metodologia de controlo adaptativo indirecto, combinando pela primeira vez redes neuronais *MLNN*, treinadas pelo algoritmo de retropropagação, numa estrutura de controlo adaptativa convencional.

Destacam-se ainda nesta fase inicial os trabalhos de Saint-Donat *et al.* (1991) e de Hunt e Sbarbaro (1991). Em Saint-Donat é incorporada uma rede neuronal num esquema de controlo preditivo, aplicada ao controlo do *PH* de um reactor químico. Hunt e Sbarbaro sugerem que as redes neuronais podem ser incorporadas em esquemas convencionais de controlo por modelo interno (*internal model control*), permitindo assim tirar partido das características de estabilidade e robustez deste tipo de estruturas. Por fim, destaca-se o trabalho de Chu *et al.* (1990), uma vez que foram os primeiros a utilizar redes recorrentes (de Hopfield) aplicadas ao controlo.

Após esta fase inicial (fins da década de oitenta, início da década de noventa), têm surgido uma quantidade extraordinária de trabalhos relacionados com a aplicação de redes neuronais no controlo. Têm sido propostas uma diversidade enorme de estruturas, todas elas reclamando para si algo de novo, não sendo feitas, por vezes, referências a outros trabalhos semelhantes. Nesta dissertação não se pretende rever de uma forma abrangente as aplicações de redes neuronais no contexto do controlo limitando-se aos objectivos inicialmente traçados, ou seja, a metodologias de controlo híbridas baseadas em técnicas de controlo não linear.

## Presente e Futuro

O projecto de controladores usando técnicas de *soft computing*, se por um lado cria novos desafios, por outro, proporciona metodologias alternativas/complementares, para além dos controladores convencionais, principalmente pela sua capacidade em lidar com incertezas, complexidades e não linearidades.

Se por um lado existem provas dadas das suas capacidades por outro lado, existem poucas aplicações na indústria deste tipo de metodologias. E de entre as razões que se podem referir para este facto, salientam-se as seguintes: não existe uma metodologia sistemática, sólida e bem definida para o projecto deste tipo de controladores; não existem aplicações sonantes que possam ser considerados exemplos de referência a seguir e, desta forma, transmitir a necessária confiança aos responsáveis da indústria.

Neste contexto, Narendra (1996), refere os seguintes dados: durante 1990-1995 foram publicados 9955 artigos em revistas de engenharia contendo as palavras “*redes neuronais*”. De entre estes, 8000 tratam de problemas de aproximação de funções e reconhecimento de padrões, isto é, obtenção de mapeamentos estáticos. Ainda em relação aos 9955, 1960 artigos são dedicados a problemas de controlo, que de alguma forma incluem redes neuronais. Neste caso dos 1960, 353 são na área de aplicações, dos quais 43% estão relacionados apenas com aspectos teóricos. Dos restantes, 28% referem-se a simulações computacionais e 23% estão relacionados com experiências laboratoriais sendo apenas 4% (14 artigos) relativos a aplicações industriais reais.

Para terminar, e da análise da literatura existente, pensa-se que estão dadas provas quanto à possibilidade de utilização prática das metodologias neuronais no controlo. Contudo, está-se ainda numa fase de maturação, de compreensão das capacidades destas metodologias e de reflexão da forma de como as inserir eficazmente em estruturas de controlo.

Esta tese pretende ser uma contribuição para esse fim. Propõe-se deste modo uma técnica de controlo genérica baseada em redes neuronais recorrentes que é acima de tudo abordada segundo técnicas bem estabelecidas de controlo não linear.

# 3.2

## Estruturas Neurais de Controlo

*Nesta secção revêem-se estruturas de controlo que utilizam redes neuronais. Apresenta-se uma classificação para este tipo de sistemas de controlo e introduzem-se alguns dos trabalhos desenvolvidos ao longo da elaboração desta dissertação, integrando-os neste contexto.*

### 3.2.1 Introdução

O treino de uma rede neuronal, a ser aplicada num esquema de controlo, pode ser efectuado em-diferido ou em-linha, resultando daí respectivamente esquemas de controlo fixos ou adaptativos. No primeiro caso o conjunto de dados de treino é conhecido *à priori* e a determinação dos pesos da rede uma vez efectuada não é alterada. No segundo caso, é usada, em cada instante, informação para actualizar os parâmetros da rede. As vantagens e desvantagens de cada um destes tipos de treino foi já abordado aquando da modelização (secção 2.3.1), aplicando-se os princípios aí referidos também ao caso do controlo.

Saliente-se neste caso o facto de que os controladores fixos requerem necessariamente um elevado grau de optimização inicial, tendo em vista a obtenção de um controlador preciso, além dos inevitáveis problemas de generalização. Os métodos adaptativos eliminam a necessidade de uma fase inicial precisa mas, por outro lado levantam o problema da estabilidade da malha de controlo, aspecto de grande importância. Os modelos neuronais, pelas suas características, são candidatos ideais a incorporar em esquemas de controlo adaptativo. Possuem uma estrutura paramétrica, de não linearidades fixas, tratáveis e ao mesmo tempo genéricas, cujos parâmetros podem ser ajustados permanentemente a partir de dados de operação.

### 3.2.2 Classificação de Estruturas

O sucesso das redes neuronais no controlo tem contribuído de uma forma extraordinária para o seu próprio desenvolvimento. Daí que tenham surgido nos últimos anos um número elevado de trabalhos, tanto em conferências internacionais como em revistas da especialidade. Dada a grande multiplicidade de assuntos e campos de aplicação é difícil distinguir aquilo que é realmente novidade e o que não é, tornando-se pois difícil sistematizar e dar uma visão global de todos os trabalhos efectuados neste contexto. Tem havido no entanto diversas tentativas de classificação de estruturas de controlo que incorporam redes neuronais, de que são exemplos os trabalhos de Werbos (1991), Hunt *et al.* (1992), Omatu *et al.* (1996) e Agarwal (1997).

Werbos sugere uma divisão em cinco categorias, baseada na função que a rede desempenha no sistema de controlo. Hunt *et al.* distinguem nove classes diferentes, em função das leis de aprendizagem e da estrutura da rede, abordando quase todos os paradigmas neuronais de controlo existentes à altura. Omatu *et al.* classificam os controladores neuronais em cinco categorias, de uma forma semelhante a Werbos. Agarwal apresenta uma classificação hierárquica, começando por distinguir duas classes principais: redes a desempenhar as funções propriamente ditas de um controlador e redes a ser usadas no projecto de controladores. Descreve depois, apontando vantagens e desvantagens, cada uma das subclasses, não conseguindo evitar sobreposições na sua classificação.

Nesta dissertação propõe-se contribuir para a classificação de estruturas neuronais de controlo, apresentando-se uma divisão inspirada na classificação anteriormente citada para métodos adaptativos convencionais, directos e indirectos, procurando-se uniformizar as estruturas existentes de controlo neuronal.

Antes de tudo, e de uma forma genérica, uma rede neuronal caracterizada por uma determinada estrutura e por um conjunto de parâmetros  $W$ , é inserida no contexto do controlo sempre com o objectivo de minimizar um critério  $J(W)$ , (3.2-1), tirando-se pois partido das suas capacidades de aproximação.

$$\mathcal{N}(W): \min_W J(w) \quad (3.2-1)$$

Posto isto, a incorporação das redes tem-se pautado por duas ideias distintas:

- i) *Controladores neuronais propriamente ditos* - neste caso a rede desempenha as funções de um controlador, fazendo sentido falar-se de um controlador neuronal. Regra geral, a rede é treinada de forma a representar o modelo inverso do processo, distinguindo-se duas formas de o conseguir: directamente, sem recorrer a informação do processo ou, indirectamente, à custa de informação do processo. Entendidos como esquemas de controlo convencional, secção 3.1.2, poderemos classificar os primeiros como esquemas de controlo directo, e os segundos como esquemas de controlo indirecto.

- ii) *Controladores híbridos* - neste caso a rede neuronal de alguma forma serve de base ao projecto de um controlador, não sendo contudo ela própria um controlador. É portanto uma estruturas que combina várias técnicas, sendo uma delas neuronal. Entendidos como esquemas de controlo convencional, secção 3.1.2, poderemos vê-los como esquemas de controlo indirecto, sendo portanto os parâmetros do controlador obtidos indirectamente.

Na Figura 3.2-1 mostra-se a classificação proposta.

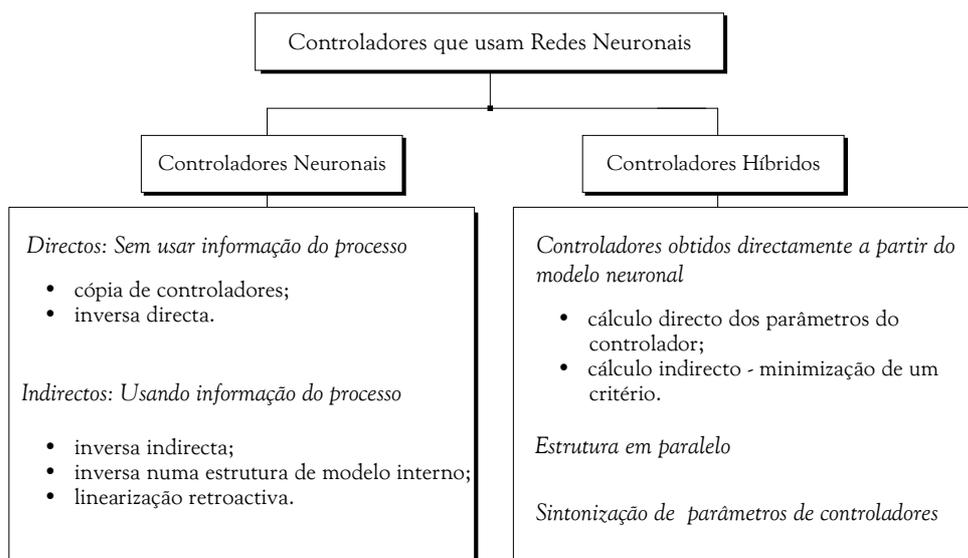


Figura 3.2-1: Classificação de controladores que usam redes neuronais.

## Controladores Neurais

As metodologias de controlo que utilizam directamente as redes neuronais como controladores podem ser englobados em duas categorias distintas, dependendo do tipo de informação que utilizam:

- i) *Directos* - Controladores neuronais que não usam informação do processo:
- cópia de controladores;
  - inversa directa.
- ii) *Indirectos* - Controladores neuronais que usam informação do processo:
- inversa indirecta;
  - inversa numa estrutura de modelo interno (IMC<sup>1</sup>);
  - linearização retroactiva.

<sup>1</sup> IMC- Internal model control

Note-se que, se as duas subclasses de controladores neuronais se distinguem com base na informação que utilizam elas podem, de forma idêntica, ser distinguidas pelo tipo de critério que utilizam. Assim, para a primeira subclasse (i) o critério a minimizar é determinado directamente a partir da acção de controlo, (3.2-2), não havendo por isso necessidade da existência de informação do processo.

$$\mathcal{N}(W): \min_W J[ e_u(k) ] = \min_W J[ u(k) - \hat{u}(k) ] \quad (3.2-2)$$

Na segunda subclasse (ii) o critério é determinado a partir da saída do processo (3.2-3) e daí a necessidade de se usar informação sobre o processo.

$$\mathcal{N}(W): \min_W J[ e_n(k) ] = \min_W J[ y_d(k) - y_n(k) ] \quad (3.2-3)$$

A seguir descreve-se cada uma das estruturas.

i) *Directos - Controladores neuronais que não usam informação do processo*

#### Cópia de Controladores

Este é talvez o método mais intuitivo e também mais simples para o projecto de um controlador neuronal e talvez por essa razão tenha dado origem à primeira aplicação de controlo usando redes neuronais (Widrow e Smith, 1963). Basicamente, neste tipo de metodologia tira-se partido da propriedade fundamental das redes, a sua capacidade de aproximação de funções não lineares e complexas.

Na fase de aprendizagem o treino é efectuado tendo em operação o controlador existente, como se mostra na Figura 3.2-2. A diferença entre a saída do controlador e a da rede neuronal,  $e_u(k) = u(k) - \hat{u}(k)$ , é usada para o seu treino. A sua implementação é portanto efectuada com o objectivo de copiar o comportamento de um controlador já existente (por exemplo um operador), de forma a substituí-lo futuramente. Regra geral, este tipo de metodologia não é adaptativa, sendo a aprendizagem efectuada em-diferido.

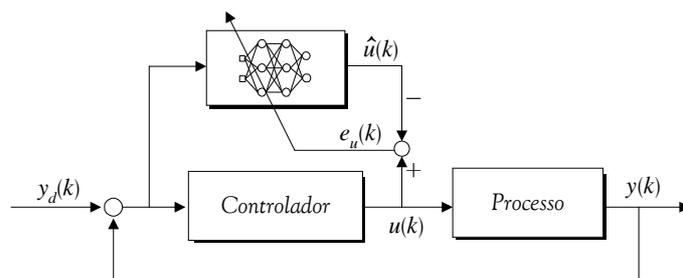


Figura 3.2-2: Estrutura cópia de controladores.

Torna-se evidente que esta estratégia não consiste propriamente no projecto de um controlador mas apenas na tentativa de copiar um controlador já existente, não sendo óbvia a utilidade de tal procedimento. Porém, a existência de um modelo de um controlador pode ser vantajosa pois além da possível substituição de um operador humano, uma rede neuronal pode ser bastante mais simples sob o ponto de vista

computacional do que um controlador que envolva cálculos complexos. Este tipo de metodologia é entendida por Kawato *et al.* (1987) como consistindo na aprendizagem de um modelo inverso, o que se justifica porque de facto, uma vez treinado o controlador neuronal, ele desempenha as funções de uma inversa do processo.

Uma das aplicações de maior sucesso usando este esquema de controlo neuronal é o *Intelligent Arc Furnace* usado na indústria de aço, desenvolvido pela *Neural Applications Corporation*, (Staib, 1993). O processo de fundição de aço, utilizando energia eléctrica, é altamente complexo e não linear, existem interacções entre as suas variáveis, pelo que efectuar a sua modelização baseada em primeiros princípios é praticamente impossível. Contudo, as redes neurais mostraram-se adequadas ao problema. Em concreto, uma rede neuronal constituída por cerca de 200 neurónios (um número elevado em comparação com os exemplos habituais) é utilizada sendo os seus pesos adaptados de 15 em 15 segundos em blocos de 150 dados. Outros exemplos práticos de sucesso desta metodologia são referidos por Bishop (1995b).

### Inversa Directa

A ideia da metodologia inversa é, tal como a anterior, utilizar as capacidades de aproximação das redes, só que neste caso não para captar a dinâmica de um sistema (controlador), mas sim, para captar a dinâmica da inversa do processo. A ideia foi originalmente proposta por Psaltis *et al.* (1988) e por eles designada de aprendizagem generalizada (em oposição à especializada a referir futuramente). Na Figura 3.2-3 mostra-se esquematicamente esta estrutura. A saída desejada  $y_d(k)$  é aplicada à rede que gera um sinal de controlo  $\hat{u}(k)$ , que uma vez aplicado ao processo produz a saída  $y(k)$ . A saída por sua vez é aplicada à rede neuronal resultando num novo valor  $\hat{u}^*(k)$ . O treino da rede tem por base a minimização do erro  $e_u(k) = \hat{u}(k) - \hat{u}^*(k)$ .

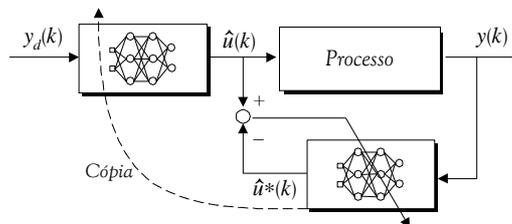


Figura 3.2-3: Estrutura inversa directa.

Desta forma, a rede procura obter um modelo inverso de um processo dinâmico. Contudo, a obtenção de uma inversa conduz sempre a alguns problemas, nomeadamente a existência de tempos de atraso, impossíveis de inverter e a presença de sistemas de fase não mínima que conduz a inversas instáveis, Stephanopoulos e Han (1994) e Shinsky (1994). Além disso, se a relação entrada-saída do sistema não é bijectiva poder-se-á obter uma solução para a qual a acção de controlo não conduz necessariamente a saída ao seu valor desejado, ou por outras palavras, um erro nulo de controlo,  $e_u(k) = u(k) - \hat{u}(k)$ , não conduz necessariamente a um erro nulo da saída,  $e_n(k) = y_d(k) - y_n(k)$ .

Pelas razões apontadas, não são muitas as aplicações que utilizam esta metodologia, no entanto, salientam-se os trabalhos de Bhat e McAvoy (1990) e Psychogios e Ungar (1991). Note-se, no entanto, que é comum a utilização deste tipo de estruturas inseridas numa estrutura de controlo com modelo interno, como se verá futuramente.

*ii) Indirectos - Controladores neuronais que usam informação do processo*

Em qualquer das estruturas anteriores, a rede é treinada de forma a minimizar um critério baseado no erro de controlo, (3.2-2). Pelo contrário, nesta subclasse, o critério baseia-se no erro da saída (3.2-3). Contudo, a saída da rede  $y_n(k)$  depende do sinal de controlo  $u(k)$  que, por sua vez, depende dos parâmetros do controlador,  $W$ , a determinar. Usando a regra da cadeia pode-se escrever (3.2-4).

$$\frac{\partial J}{\partial W(k)} = \frac{\partial J}{\partial y_n(k)} \frac{\partial y_n(k)}{\partial u(k)} \frac{\partial u(k)}{\partial W(k)} \quad (3.2-4)$$

Para que a equação anterior possa ser aplicada torna-se necessário conhecer o termo  $\frac{\partial y_n(k)}{\partial u(k)}$  (Jacobiano), que depende das características do sistema a controlar.

**Inversa Indirecta**

Esta é uma das metodologias mais comuns no contexto do controlo neuronal. Proposta independentemente por Jordan e Rumelhart (1990), Narendra e Parthasarathy (1990) e Nguyen e Widrow (1990) a ideia principal consiste em considerar, além da rede que desempenha as funções de controlador, uma outra rede neuronal como modelo do processo (assumido como diferenciável), de forma a ser possível determinar o valor do Jacobiano, definido em (3.2-4). Basicamente, o modelo neuronal do processo é visto como uma camada adicional da rede que serve de controlador, usado para permitir a retropropagação do erro de controlo. Adicionalmente, de forma a eliminar eventuais dificuldades no cálculo da inversa do processo, considera-se uma dinâmica diferente da identidade (modelo de referência), Figura 3.2-4.

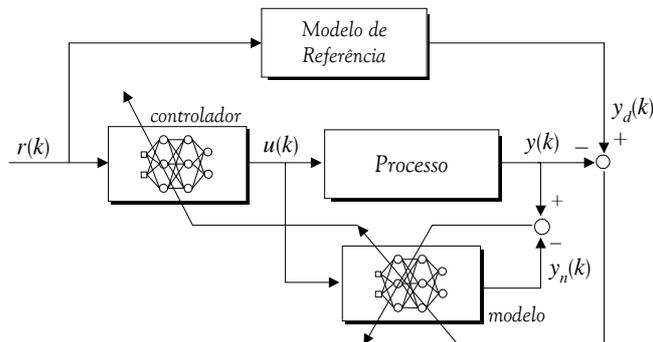


Figura 3.2-4: Estrutura inversa indirecta.

Note-se que se o modelo de referência for unitário procede-se à determinação de uma inversa. Porque os parâmetros do controlador são de alguma forma determinados com base no processo a estrutura é considerada uma estrutura de controlo indirecta, como já foi referido.

Apesar de na maioria das vezes se utilizar outra rede como modelo do processo é importante notar que na verdade apenas se exige a existência de um modelo capaz de fornecer o termo  $\frac{\partial y(k)}{\partial u(k)}$ , não necessariamente outra rede neuronal. Psaltis *et al.* (1988) propuseram o método de aprendizagem especializada em que o cálculo da derivada era efectuado usando a aproximação do *Jacobiano* por (3.2-5), eliminando a necessidade de existência de um modelo para o processo.

$$\frac{\partial y(k)}{\partial u(k)} \approx \frac{\Delta y(k)}{\Delta u(k)} = \frac{y(k) - y(k-1)}{u(k) - u(k-1)} \quad (3.2-5)$$

Saerens e Soquet (1991) propuseram ainda uma simplificação mais severa assumindo apenas dois possíveis valores.

$$\frac{\partial y(k)}{\partial u(k)} = -1 \text{ ou } +1 \quad (3.2-6)$$

#### Inversa numa Estrutura de Modelo Interno

Talvez a metodologia com maior sucesso que utiliza o modelo inverso neuronal é a estrutura de modelo interno, proposta por Hunt e Sbarbaro (1991), técnica por sua vez originalmente desenvolvida no contexto de controlo robusto por Morari e Zafriou (1989).

A estrutura é análoga à da inversa indirecta, podendo adicionalmente ser considerado um filtro de forma a reduzir a sensibilidade do esquema a perturbações e a variações bruscas de referência, Figura 3.2-5. Neste esquema uma rede é inicialmente treinada como modelo do processo e outra, a desempenhar as funções de controlador, é treinada como inversa do processo. Como grande vantagem o facto de que o sistema de controlo total é estável desde que o controlador, modelo e processo sejam estáveis. Como desvantagem o facto de ser aplicável apenas a sistemas estáveis e a sistemas de fase não mínima.

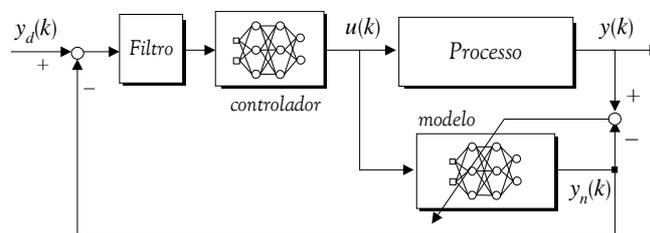


Figura 3.2-5: Estrutura por modelo interno.

Usando este tipo de estrutura Ribeiro (1995) desenvolve, com sucesso, uma metodologia de modelização, controlo e diagnóstico para um processo multi-variável industrial utilizando redes neuronais, concretamente para um forno rotativo de cal, inserido no sistema de caustificação do processo kraft da indústria de pasta de papel. Numa fase inicial é especificado e treinado um modelo neuronal *MLNN*, assim como a respectiva inversa. Estes são depois inseridos numa estrutura de controlo *IMC*, facto que se justifica pelas características de robustez e estabilidade desta estrutura. Adicionalmente é efectuada a monitorização das condições de operação e diagnóstico de situações anómalas usando, para tal, outra rede neuronal, previamente treinada de forma a reconhecer padrões de entrada indicadores de situações anómalas.

Outros exemplos de trabalhos que utilizam redes neuronais numa estrutura *IMC* são os de Lightbody e Irwin (1995), Rivals e Personnaz (1996) e Brown *et al.* (1997).

#### Linearização Retroactiva

Narendra e Parthasarathy (1990) e Chen (1990) introduzem a ideia de combinar redes *MLNN* de forma a obterem modelos na forma *affine*, permitindo deste modo efectuar o projecto do controlador usando conceitos de linearização retroactiva ( $FL^2$ ), uma técnica de controlo não linear. Concretamente, Narendra utiliza três redes neuronais *MLNN* para implementar um sistema de controlo. Duas delas,  $\mathcal{N}_1(\cdot)$  e  $\mathcal{N}_2(\cdot)$ , são usadas para obter um modelo do tipo *affine*, (3.2-7).

$$y_n(k) = \mathcal{N}_1(u(k-1), \dots, u(k-n_b), y(k-1), \dots, y(k-n_a)) \\ + \mathcal{N}_2(u(k-1), \dots, u(k-n_b), y(k-1), \dots, y(k-n_a)) u(k) \quad (3.2-7)$$

A terceira,  $\mathcal{N}_3(\cdot)$ , é depois treinada de forma a reproduzir a inversa da segunda, isto é,  $\mathcal{N}_3(\cdot) = \mathcal{N}_2(\cdot)^{-1}$ . É assim possível determinar a acção de controlo directamente a partir de (3.2-8), usando conceitos de linearização retroactiva.

$$u(k) = \mathcal{N}_3(\cdot) [y_d(k) - \mathcal{N}_1(\cdot)] \quad (3.2-8)$$

## Controladores Híbridos

Esta classe distingue-se da anterior pelo facto de o controlador não ser propriamente uma rede neuronal, mas estas serem utilizadas, de alguma forma, no projecto de um controlador. Incluem-se nesta classificação as seguintes principais classes de controladores:

- i) *Controladores obtidos directamente a partir do modelo neuronal:*
- cálculo directo dos parâmetros do controlador;
  - cálculo indirecto - minimização de um critério.

---

<sup>2</sup> *FL- Feedback Linearization*

ii) *Estrutura em paralelo*

iii) *Sintonização de parâmetros de controladores*

Na primeira subclasse (i) usa-se explicitamente um modelo neuronal para descrever o processo e os parâmetros do controlador são determinados com base no modelo neuronal, sendo, por esta razão, classificados como controladores indirectos. Podem-se ainda considerar duas subclasses. Uma, em que os parâmetros do controlador são directamente obtidos a partir dos do modelo neuronal, englobam-se aqui os métodos de controlo não linear de que são exemplos as técnicas de linearização retroactiva e a de regulação da saída. Outra, em que o objectivo de controlo é especificado sob a forma de um critério ou uma função objectivo e a acção de controlo é determinada de forma a minimizar esse critério, de que são exemplos as técnicas de controlo preditivo e de controlo óptimo.

Na segunda subclasse (ii) assume-se a existência de um controlador e a introdução de estruturas neuronais tem como objectivo compensar possíveis deficiências do controlador que se encontra em operação. Na terceira subclasse (iii) assume-se, tal como na subclasse anterior, a existência prévia de um controlador cujos parâmetros são determinados (ou sintonizados) com o auxílio de uma rede neuronal.

i) *Controladores obtidos directamente a partir do modelo neuronal*

**Cálculo Directo dos Parâmetros**

Neste caso são exploradas as capacidades de aproximação das redes neuronais como modelos do processo. Na verdade, uma rede neuronal é um modelo paramétrico, de parâmetros  $W$  e, neste sentido, é possível determinar os parâmetros de um controlador  $C = f(W)$ , como se mostra na Figura 3.2-6.

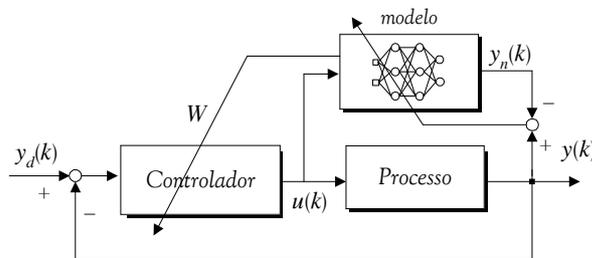


Figura 3.2-6: Determinação directa de parâmetros.

A utilização deste tipo de metodologia pode basear-se em técnicas lineares, efectuando neste caso uma linearização do modelo neuronal ou então em técnicas de controlo não linear.

*Linearização Local*

Neste tipo de técnica aborda-se o problema de controlo não linear efectuando uma linearização do modelo não linear e projectando um controlador linear válido, em princípio apenas na vizinhança desse regime de operação. Neste contexto, usando uma

rede neuronal *MLNN* previamente treinada, Ahmed e Tasaddup (1994a) e (1994b) propõem um esquema do controlo considerando uma linearização local em cada instante, possibilitando o uso de controladores convencionais. Segundo os autores, e uma vez que o projecto de controladores lineares proporciona resultados bem estabelecidos faz todo o sentido utilizar este tipo de metodologia. Sørensen (1996) de forma semelhante determina, a partir de um modelo neuronal, os parâmetros de um sistema linear e projecta um controlador linear por colocação de pólos. Suykens *et al.* (1997), a partir de uma determinada transformação, interpretam um modelo neuronal não linear como um modelo linear com incertezas, para o qual é possível aplicar as teorias de controlo robusto.

Em Henriques e Dourado (1999a) é proposto um esquema de controlo adaptativo indirecto que combina redes neuronais recorrentes e técnicas de controlo linear. Concretamente, é usada uma rede de Elman para capturar a dinâmica do processo, treinada em-linha, usando uma modificação do algoritmo *BPTT*. Pela sua interpretabilidade como um modelo de estados não linear, a rede neuronal mostra-se adequada para o desenvolvimento de controladores que usem técnicas de realimentação de estados, sendo no caso particular implementada, em simultâneo, uma técnica de desacoplamento e colocação de pólos. Para permitir o uso de técnicas de controlo linear é efectuada em cada instante uma linearização local do modelo neuronal, sendo a estratégia validada num processo laboratorial multi-variável (*DTS200*).

#### Controlo não Linear

Nesta abordagem o projecto do controlador é efectuada utilizando técnicas de controlo não linear, de que é exemplo a linearização retroactiva, Isidori (1995). A ideia básica da linearização retroactiva é cancelar as não linearidades do modelo neuronal através de realimentação, de forma a que o modelo resultante seja linear. Pela sua importância a próxima secção (3.3) é dedicada a trabalhos que seguem esta abordagem.

#### Cálculo indirecto dos Parâmetros - Minimização de um Critério

Tanto nos controladores óptimos como nos preditivos o objectivo de controlo é especificado sob a forma de um critério, sendo a acção de controlo resultante da sua minimização. Os controladores óptimos/preditivos neuronais não são senão um caso particular de controladores óptimos/preditivos não lineares, nos quais o modelo de predição é uma rede neuronal, Figura 3.2-7.

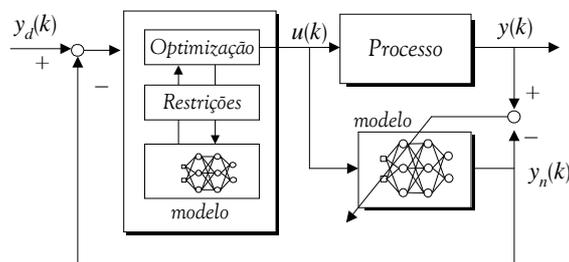


Figura 3.2-7: Estrutura de controlo óptimo/preditivo.

Regra geral, pretende-se minimizar um critério da forma (3.2-9),

$$J = \sum_{i=1}^N [y_d(k+i) - y_n(k+i)]^2 + \sum_{i=1}^N \lambda_i [u(k+i) - u(k+i-1)]^2 \quad (3.2-9)$$

em que  $N$  é uma constante que define o horizonte de optimização e  $\lambda_i$  é uma constante que permite atribuir importâncias relativas a futuras variações na acção de controlo.

São exemplos de controladores preditivos que incorporam redes neuronais os trabalhos de Babuska *et al.* (1999), Botto *et al.* (1999) e Draeger *et al.* (1995). No contexto de controlo óptimo referem-se os de Goh e Edwards (1994) e Niesler e Plessis (1995).

Em Henriques *et al.* (1999d) são feitos estudos experimentais de comparação entre duas estratégias de controlo preditivo, aplicados a um processo multi-variável laboratorial (*DTS200*). Uma delas faz uso explícito de um modelo físico obtido a partir de primeiros princípios, a outra, híbrida, combina um modelo neuronal recorrente num esquema de controlo preditivo. O problema de minimização não linear é solucionado por um método de gradiente, de uma forma análoga à fase de estimação dos parâmetros da rede. Na segunda estratégia, o modelo físico não linear (primeiros princípios) é em cada instante linearizado e o problema de controlo solucionado pela resolução de um problema de programação linear quadrática. Aplicadas ao sistema laboratorial, qualquer das metodologias, permitem obter resultados aceitáveis. No entanto, quer em presença de dinâmicas não tidas em conta na modelização, quer em presença de variações na dinâmica a capacidade de adaptação das redes mostra-se superior. Além disso, pela sua generalidade e facilidade de projecto (não havendo necessidade de desenvolvimento de um modelo de primeiros princípios) as redes neuronais mostram-se vantajosas.

Gil e Henriques (1999) desenvolvem e aplicam na prática a um processo laboratorial uma metodologia de controlo preditivo utilizando como modelo de predição uma rede neuronal. Em particular, e a partir de dados de operação, uma rede recorrente é previamente treinada usando o algoritmo de Levenberg-Marquardt. Depois, a partir de uma linearização local do modelo neuronal, soluciona-se em cada instante um problema de controlo óptimo sujeito a restrições, formulado como um problema de programação quadrática. Contudo, das experiências realizadas constata-se a existência de erros de controlo em regime final, justificados quer por defeitos de modelização, quer pela degradação do modelo devido à linearização local efectuada. O problema é ultrapassado com a introdução (a montante do controlador preditivo) de um compensador integral, que tem em consideração o sinal de erro afectado por uma componente exponencial decrescente, de forma que apenas pequenos desvios sejam considerados. Elimina-se assim, em certa medida, a necessidade de adaptação em-linha do modelo neuronal.

#### ii) Estrutura em Paralelo

Em certos casos existe à partida um conhecimento, que pode até não ser muito rigoroso, do processo a controlar, por exemplo, sob a forma de um modelo linear ou mesmo de um modelo baseado em primeiros princípios. Assim sendo, não terá interesse,

nem tão pouco será conveniente, proceder à obtenção de um modelo global tipo *black-box*, uma vez que se isso acontecer o conhecimento existente será desprezado.

Uma alternativa a considerar, como se mostra na Figura 3.2-8, consiste num esquema híbrido, onde uma rede neuronal opera em paralelo com o modelo existente, de forma a que discrepâncias entre este e o processo real sejam compensadas.

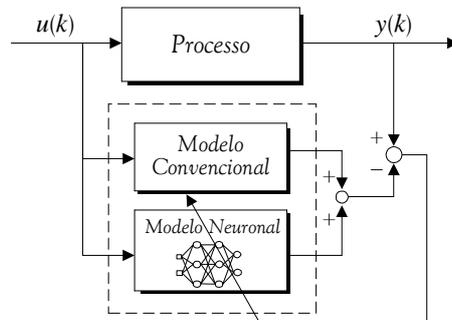


Figura 3.2-8: Estrutura de controle em paralelo.

Seguindo o mesmo raciocínio, o princípio pode, de forma análoga, ser usado para o controlador onde, além de um qualquer controlador (projectado com base no modelo conhecido), é utilizada uma rede neuronal para compensar eventuais desvios que o controlador não consiga eliminar. Note-se que a utilização de cada uma das compensações, do modelo ou controlador, pode ser implementada independentemente, isto é, só uma delas ou as duas em simultâneo.

Como grande vantagem deste tipo de metodologia resulta a possibilidade de adicionar, a um controlador já existente, estruturas neuronais, sem que para isso o controlador tenha de ser substituído. Tem assim grande potencial de aplicação prática, uma vez que podem ser incorporadas em esquemas de controlo que se encontram a operar, sem os substituir, possibilitando, em princípio, melhorar o desempenho do sistema global de controlo.

Em Omatu *et al.* (1996) é proposta uma metodologia que incorpora redes neuronais num esquema de controlo não linear por realimentação de variáveis de estado. Para compensar eventuais erros de modelização, originados por um observador linear (Luenberger), uma rede neuronal é utilizada, à semelhança da Figura 3.2-8. Para o projecto do controlador usa-se uma técnica de controlo óptimo, usando os estados do observador e os parâmetros do modelo linear. Uma segunda rede neuronal é utilizada em paralelo com o controlador, compensando eventuais deficiências da parte deste.

Uma ideia análoga é seguida por Fujii *et al.* (1999). Propõem para a parte linear uma técnica convencional de controlo por colocação de pólos, sendo a contribuição não linear constituída por duas redes neuronais, uma combinada em paralelo com o modelo linear, para compensar erros de modelização e a segunda incluída no controlador, treinada para descrever a inversa da primeira.

Zhu *et al.* (1999) propõem apenas uma rede neuronal para melhorar o desempenho de um controlador adaptativo indirecto. Consideram o processo a controlar constituído pela combinação de dois modelos, um linear e uma rede neuronal. Para a identificação

do modelo linear é utilizado um algoritmo recursivo de mínimos quadráticos, sendo por sua vez a rede neuronal treinada por um algoritmo de retropropagação, de forma a minimizar a diferença entre a saída do processo real e a do modelo linear. Quanto à lei de controlo, é implementado um algoritmo de variância mínima generalizada, permitindo ter em conta a contribuição de ambos os modelos (linear e rede neuronal).

### iii) Sintonização de Parâmetros de Controladores

Outra aplicação das redes neurais no contexto do controlo é a sua utilização na sintonização de parâmetros de controladores cuja estrutura é conhecida e fixa, mas cujos parâmetros são desconhecidos. Existem em operação controladores que permitem obter desempenhos perfeitamente aceitáveis, necessitando para tal apenas de uma sintonização correcta dos seus parâmetros, de que são exemplo os controladores *PID*. Estes têm uma estrutura simples e proporcionam resultados aceitáveis para grande parte das aplicações de controlo, nomeadamente em instalações industriais. De facto, refere-se em Åström e Wittenmark (1995), que mais de 95% dos controladores usados na indústria são do tipo *PID* e em Omatu *et al.* (1996), refere-se o valor de 85%, no caso particular da indústria Japonesa. A dificuldade principal de implementação surge na sua sintonização, especialmente quando o processo a controlar é complexo e se caracteriza por regimes de operação distintos. É neste ponto, na selecção dos ganhos do controlador, que as redes neurais podem ser úteis. Na Figura 3.2-9 mostra-se um esquema de controlo em que uma rede neuronal é usada para sintonizar os parâmetros de um controlador *PID*.

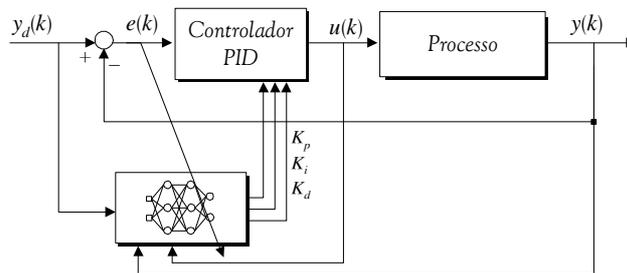


Figura 3.2-9: Sintonização de controladores *PID*.

Para tal procede-se à minimização de um critério usando informação obtida não apenas a partir de dados de operação (entradas, saídas e erro) mas também de características extraídas do seu funcionamento. Note-se que, tal como a estrutura anterior, esta metodologia tem fortes potencialidades de aplicação podendo ser facilmente incorporada num esquema de controlo já existente, sem necessariamente o substituir.

Exemplos deste tipo de metodologias aplicadas a controladores *PID* são desenvolvidas, em Li e Tso (1998) e Omatu (1999), onde é proposto um esquema de controlo adaptativo capaz de actualizar os seus parâmetros (do *PID*).

Ruano *et al.* (1992) propõem uma rede neuronal do tipo *MLNN* capaz de auto-sintonizar os parâmetros de um controlador *PID*. Usando as suas propriedades de aproximação, uma rede neuronal é utilizada para captar os mapeamentos entre determinadas medidas de identificação e os valores óptimos de um controlador *PID*,

aplicável tanto em malha aberta como em malha fechada. A abordagem por eles seguida proporciona resultados similares aos obtidos quando é efectuada uma sintonização através da optimização de um critério do tipo *ITAE-Integral Time Multiplied Absolute Error* permitindo, além disso, reduzir de uma forma considerável o número de iterações necessárias à obtenção de um mínimo local.

Henriques *et al.* (1999b) propõem uma estratégia de controlo baseada numa supervisão difusa de controladores *PID*, previamente sintonizados com o auxílio de redes neuronais. A metodologia é desenvolvida e aplicada na prática a um campo de colectores solares existente numa plataforma solar, Almeria-Espanha. A opção por tal metodologia justifica-se pelo facto de o processo se caracterizar fundamentalmente por apresentar ao longo do dia condições de operação distintas, dependendo principalmente da variação da radiação solar e da temperatura de referência. Assim, numa primeira fase, foram identificadas as regiões de operação representativas da dinâmica do processo. Depois, e a partir de dados experimentais, obtidos tanto pelos autores, como por outros utilizadores da plataforma, foram sintonizados em-diferido, usando uma metodologia baseada em redes neuronais, para cada região de operação os correspondentes controladores *PID*, que se sabia serem adequados ao processo em questão. Uma vez em operação, é função do supervisor difuso identificar cada uma das regiões de operação e comutar para o respectivo controlador *PID*. Entendida sob outra perspectiva esta metodologia pode ser interpretada como uma técnica de escalonamento de ganhos. Resultados experimentais, levados a cabo na plataforma, provaram a validade da estratégia desenvolvida.

### 3.2.3 Metodologia Proposta: Controlo Não Linear Adaptativo Baseado no Modelo Neuronal

O objectivo desta dissertação pautou-se desde o seu início pela investigação de estruturas híbridas que pudessem combinar as capacidades de técnicas de *soft computing* com resultados bem conhecidos da teoria de controlo. E nesse sentido, além de estudos teóricos, vários estudos experimentais aplicados a processos laboratoriais foram levados a cabo, de forma a dar uma orientação sobre a metodologia mais adequada.

Podem-se distinguir dois grupos fundamentais de trabalhos realizados. Trabalhos em que são feitas comparações entre várias metodologias de controlo e trabalhos em que são investigadas as características das técnicas particulares de controladores que usam redes neuronais na sua estrutura.

Assim, para o primeiro grupo referem-se:

- i) *estratégias adaptativas*: se por um lado são óbvias as vantagens dos métodos adaptativos em relação aos métodos não adaptativos, por outro, as vantagens do uso de modelos não lineares em esquemas adaptativos não são assim tão evidentes em relação ao uso de modelos lineares. De forma a investigar as vantagens e desvantagens foram feitos em (Pereira *et al.*, 2000) estudos experimentais usando duas estratégias adaptativas, numa usando um modelo linear convencional noutra um modelo neuronal (não linear).

- ii) *estratégias preditivas*: outra das vantagens reivindicadas pelas metodologias de *soft computing* é a sua capacidade em oferecer resultados, pelo menos tão relevantes, quanto as metodologias convencionais. Nesse sentido foram investigadas e comparadas experimentalmente duas estratégias de controlo (no caso particular preditivas) que utilizam modelos físicos, obtidos a partir de primeiros princípios e estratégias que utilizem modelos neuronais (Henriques *et al.*, 1999d).
- iii) *estratégias de soft computing*: incluídas nas classes de metodologias de controlo inteligente, foram feitas comparações experimentais entre técnicas adaptativas de controlo usando modelos difusos, modelos neuronais com funções de activação globais (sigmóidais) e funções de activação locais (Gaussianas), (Henriques *et al.*, 1998b).

Em relação ao segundo grupo, e dentro de metodologias de controlo híbridas (secção 3.2.2), isto é, que de alguma forma usam redes neuronais na sua estrutura, não sendo, contudo, o controlador uma rede neuronal, destacam-se os seguintes trabalhos:

- i) *técnicas de controlo linear*: foram numa fase inicial desenvolvidos e implementados na prática esquemas de controlo adaptativo indirecto usando estruturas híbridas, combinando redes neuronais com a teoria de controlo linear, (Henriques e Dourado, 1998a), (Henriques e Dourado, 1999c). Depois, ainda usando controladores lineares, considera-se o modelo neuronal recorrente como um observador, sendo investigados métodos de controlo por realimentação de variáveis de estado (Henriques e Dourado, 1998c), (Henriques e Dourado, 1999a).
- ii) *sintonização de controladores*: tirando partido das vantagens fundamentais das redes neuronais e dos sistemas difusos, respectivamente em capturar a dinâmica de um processo através de dados de operação e capacidade de lidar com incertezas, foi desenvolvido e implementado num processo real de escala industrial uma metodologia híbrida, consistindo em controladores *PID* sintonizados por redes neuronais, supervisionados por um sistema difuso (Henriques *et al.*, 1999b).
- iii) *técnicas de controlo preditivo*: foram investigadas as capacidades de predição de modelos neuronais recorrentes, em particular incorporados como modelo de predição em esquemas de controlo preditivo, (Gil *et al.*, 1999), (Gil *et al.*, 2000a) e (Gil *et al.*, 2000b).
- iv) *técnicas de controlo não linear*: mais recentemente foram efectuados estudos teóricos e experimentais sobre técnicas de controlo não linear, em particular a teoria da regulação da saída (Henriques *et al.*, 2000a), (Henriques *et al.*, 2000b) e (Henriques *et al.*, 2000c).

Analisando o trabalho realizado, e a partir do conhecimento e da experiência que daí resultou, optou-se por uma estrutura híbrida, sendo o processo descrito por um modelo neuronal recorrente e o controlador projectado com base na teoria de controlo não linear. Esta é uma ideia que se pensa ser promissora, apresenta uma estrutura versátil

em termos de identificação e controlo, sendo possível realizar independentemente cada uma das tarefas. Pode-se assim tirar partido das vantagens das técnicas de *soft computing* e convencionais, respectivamente na fase de identificação (capacidade de aproximação das redes) e na de controlo (métodos não lineares).

Finalmente, dentro das técnicas que usam modelos neuronais, optou-se por técnicas em que os parâmetros sejam calculados directamente a partir do modelo neuronal, em particular a técnica da regulação da saída (*output regulation*), principalmente pelas suas características de estabilidade e pela possibilidade de aplicação, sob determinadas condições, do princípio de separação.

Assim, omitem-se pormenores acerca das outras abordagens exploradas e a secção seguinte limita-se à descrição das técnicas de controlo não linear com potencialidade de aplicação a modelos neuronais recorrentes.

# 3.3

## Controlo Não Linear

*Nesta secção introduzem-se técnicas de controlo não linear e em particular a técnica de linearização retroactiva, tanto no domínio contínuo como no domínio discreto.*

*Analisa-se depois a sua aplicação a modelos neuronais inseridos em estruturas adaptativas.*

### 3.3.1 Introdução

Foi principalmente na década de 80 que se iniciou o desenvolvimento de metodologias para o projecto de controladores baseados em modelos não lineares, distinguindo-se a técnica de linearização retroactiva ( $FL^1$ ), baseadas em conceitos matemáticos de geometria diferencial. Neste contexto Jakubczyk e Respondek (1980) e, independentemente, Hunt *et al.* em (1983) solucionaram o problema de linearização entrada-estados, através de uma mudança de coordenadas transformando um sistema não linear num linear equivalente. Krener e Isidori (1983) estabeleceram condições análogas de linearização entrada-saída e Isidori introduz a noção de ( $ZD^2$ ), (Isidori, 1985), mais tarde consolidada por Isidori e Moog (1988). Para uma análise pormenorizada destas técnicas de controlo não linear referem-se os trabalhos fundamentais de Isidori (1989) e (1995), Nijmeijer e Van der Schaft (1990) e Khalil (1992), aplicados no domínio contínuo.

Basicamente esta é um tipo de metodologia bem estabelecida que permite, quando em presença de não linearidades bem definidas e perfeitamente conhecidas, conceber um controlador que garante a obtenção de resultados teóricos precisos.

Contudo, porque nem sempre os parâmetros podem ser conhecidos com exactidão, surge a sua implementação em esquemas adaptativos. E neste caso, como já referido, mesmo para sistemas lineares, o problema da análise de estabilidade não é facilmente tratável. Relativamente à sua aplicação a modelos neuronais têm surgido alguns resultados de estabilidade, principalmente para o caso contínuo, utilizando técnicas de linearização retroactiva. Na secção seguinte (3.3.2) apresentam-se aspectos básicos desta

---

<sup>1</sup> *FL- Feedback Linearization*

<sup>2</sup> *ZD- Zero Dynamics*

técnica, de forma a poder-se compreender os trabalhos de controlo neuronal que as utilizam, e que serão descritos na secção 3.3.3.

### 3.3.2 Linearização Retroactiva

Uma técnica geralmente utilizada para o projecto de sistemas não lineares consiste em proceder previamente a uma linearização local e projectar depois um controlador linear. Genericamente, as técnicas de linearização retroactiva têm por objectivo conseguir não uma linearização local mas uma linearização global, através do cancelamento das não linearidades por realimentação. Uma vez obtido um modelo linear é depois possível efectuar o projecto de um controlador para sistemas lineares, como por exemplo um controlador *PID*, um controlador preditivo ou um controlador por colocação de pólos.

#### Sistemas Contínuos

Considere-se um sistema SISO contínuo não linear, descrito por (3.3-1).

$$\dot{x}(t) = f(x(t), u(t)) \quad (3.3-1a)$$

$$y(t) = h(x(t)) \quad (3.3-1b)$$

As variáveis  $u(t) \in \mathfrak{R}$ ,  $y(t) \in \mathfrak{R}$  e  $x(t) \in \mathfrak{R}^n$  definem respectivamente a entrada, saída e estados do sistema no instante  $t$  e as funções  $f(\cdot)$  e  $h(\cdot)$  consideram-se diferenciáveis. A técnica de linearização retroactiva consiste na implementação de uma realimentação de variáveis de estado, do tipo (3.3-2),

$$u(t) = \psi(x(t), v(t)) \quad (3.3-2)$$

de forma a transformar um sistema cuja relação entrada-saída é não linear num em que a relação entrada-saída seja linear. Na Figura 3.3-1 representa-se esquematicamente a ideia. O processo original apresenta uma relação  $u(t) \leftrightarrow y(t)$  não linear, contudo após a introdução da realimentação  $\psi(\cdot)$ , a relação  $v(t) \leftrightarrow y(t)$  é linear.

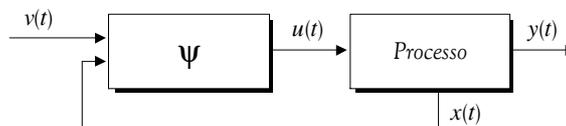


Figura 3.3-1: Linearização retroactiva.

#### Sistemas Affine

Para o caso de sistemas contínuos existe uma teoria bastante extensa sobre linearização retroactiva, especialmente aplicada a sistemas *affine*. Mesmo para sistemas genéricos do tipo (3.3-1), em que a relação *affine* não é verificada é, na maioria das vezes,

possível efectuar uma transformação de forma a que o sistema nas novas coordenadas possa ser linearizado por realimentação. Esta é uma das razões que justifica o sucesso desta classe de modelos.

Considere-se então o caso particular de um sistema *affine*, assim designado porque o modelo é linear no que refere à acção de controlo  $u(t)$ , descrito pela equação (3.3-3).

$$\dot{x}(t) = f(x(t)) + g(x(t)) u(t) \quad (3.3-3a)$$

$$y(t) = h(x(t)) \quad (3.3-3b)$$

### Grau relativo

Se for calculada a derivada da saída  $y(t)$ , aplicando a regra da cadeia, obtém-se a equação (3.3-4).

$$\frac{\partial y(t)}{\partial t} = \dot{y}(t) = \frac{\partial h(x(t))}{\partial x(t)} \frac{\partial x(t)}{\partial t} = \frac{\partial h(x(t))}{\partial x(t)} [f(x(t)) + g(x(t)) u(t)] \quad (3.3-4)$$

No contexto de controlo não linear é comum utilizar-se a seguinte notação.

$$\dot{y}(t) = L_f h(x) + L_g h(x) u(t) \quad (3.3-5)$$

Os termos  $L_f h(x)$  e  $L_g h(x)$  representam, respectivamente, as derivadas de Lie da função  $h(x(t))$  em relação a  $f(x(t))$  e  $g(x(t))$ , definidos por (3.3-6) e (3.3-7).

$$L_f h(x) = \frac{\partial h(x(t))}{\partial x(t)} f(x(t)) \quad (3.3-6)$$

$$L_g h(x) = \frac{\partial h(x(t))}{\partial x(t)} g(x(t)) \quad (3.3-7)$$

Se o termo (3.3-7) for não nulo ( $L_g h(x) \neq 0$ ) então a derivada da saída depende explicitamente de  $u(t)$ . Caso contrário, procede-se à determinação da segunda derivada da saída, obtendo-se (3.3-8).

$$y^{(2)}(t) = L_f^2 h(x) + L_g L_f h(x) u(t) \quad (3.3-8)$$

À semelhança de (3.3-7) definem-se os termos  $L_f^2 h(x) = L_f [L_f h(x)]$  e  $L_g L_f h(x) = L_g [L_f h(x)]$ . Se uma vez mais a segunda derivada da saída não depender explicitamente de  $u(t)$  o processo de derivação pode-se repetir até que isso aconteça.

**Definição**

O sistema SISO definido por (3.3-3) tem um grau relativo  $r \in \mathbb{N}$  se

$$L_g h(x) = L_g L_f h(x) = \dots = L_g L_f^{r-2} h(x) = 0 \quad (3.3-9)$$

Por outras palavras, o grau relativo de um sistema é o menor inteiro positivo tal que

$$L_g L_f^{r-1} h(x) \neq 0 \quad (3.3-10)$$

Assim, diferenciando a saída  $r$  vezes obtém-se a equação (3.3-11), neste caso dependendo explicitamente de  $u(t)$ .

$$y^{(r)}(t) = L_f^r h(x) + L_g L_f^{r-1} h(x) u(t) \quad (3.3-11)$$

**Forma Normal**

É possível provar que se um sistema genérico, (3.3-1), possuir um grau relativo definido e finito então existe uma transformação  $T: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  não linear,  $z = T(x)$ , de forma que é possível obter a seguinte forma normal.

$$\begin{aligned} \dot{z}_1^a(t) &= z_2^a(t) \\ &\vdots \\ \dot{z}_{r-1}^a(t) &= z_r^a(t) \\ \dot{z}_r^a(t) &= f_z(z^a, z^b) + g_z(z^a, z^b) u(t) \\ \dot{z}^b(t) &= \lambda(z^a, z^b) \\ y(t) &= z_1^a(t) \end{aligned} \quad (3.3-12)$$

Note-se que o modelo resultante é *affine*, em que os termos  $f_z(z^a(t), z^b(t))$  e  $g_z(z^a(t), z^b(t))$  correspondem respectivamente a  $L_f^r h(x(t))$  e  $L_g L_f^{r-1} h(x(t))$  na equação (3.3-11). De salientar que os estados  $z^b(t)$  não dependem de  $u(t)$ , logo são autónomos e definem a dinâmica interna do sistema, que não pode ser controlada. A dinâmica destes estados é conhecida por *ZD*, dizendo-se o sistema de fase mínima se a dinâmica de  $\dot{z}^b(t) = \lambda(0, z^b(t))$  for assintoticamente estável.

**Linearização exacta**

Dado um sistema *affine*, ou na forma normal, a acção de controlo pode ser facilmente calculada usando uma lei de realimentação de variáveis de estado, de forma a linearizar o sistema. Para o sistema *affine* (3.3-3), se se definir a lei de realimentação genérica (3.3-2), como (3.3-13),

$$u(t) = \frac{1}{L_g L_f^{r-1} h(x)} [v(t) - L_f^r h(x)] \quad (3.3-13)$$

resulta daqui o sistema linear (3.3-14).

$$y^{(r)}(t) = v(t) \quad (3.3-14)$$

Expostos estes conceitos facilmente se entende o teorema geral para este tipo de técnicas: um sistema não linear pode ser controlado usando a técnica de linearização retroactiva se possuir grau relativo definido e finito e a ZD for assintoticamente estável, o mesmo é dizer, se for de fase mínima.

## Sistemas Discretos

Apresentam-se agora as ideias análogas no caso de sistemas em tempo discreto. Seja então um sistema genérico MIMO não linear, descrito por (3.3-15),

$$x(k+1) = f(x(k), u(k)) \quad (3.3-15a)$$

$$y(k) = h(x(k)) \quad (3.3-15b)$$

em que  $u(k) \in \mathfrak{R}^{n_u}$ ,  $y(k) \in \mathfrak{R}^{n_y}$  e  $x(k) \in \mathfrak{R}^n$ , definem respectivamente o vector entrada, o vector saída e o vector de estados do sistema, no instante  $k$ . Assume-se que as funções  $f: \mathfrak{R}^{n+n_u} \rightarrow \mathfrak{R}^n$  e  $h: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n_y}$  são funções diferenciáveis.

### Grau relativo

Para um sistema multi-variável define-se para cada saída  $y_i$  um grau relativo  $r_i \in \mathbb{N}$  se for verificada a seguinte relação,

$$\frac{\partial}{\partial u(k)} h_i^j [f(x(k), u(k))] = 0 \quad \text{se } 0 \leq j \leq r_i - 1 \quad (3.3-16a)$$

$$\frac{\partial}{\partial u(k)} h_i^{r_i} [f(x(k), u(k))] \neq 0 \quad (3.3-16b)$$

em que

$$\frac{\partial h_i(x(k))}{\partial u(k)} = \left[ \frac{\partial h_i(x(k))}{\partial u_1(k)}, \dots, \frac{\partial h_i(x(k))}{\partial u_{n_u}(k)} \right] \quad (3.3-17a)$$

$$h_i^0 [x(k)] = h_i(x(k)) = h_i [x(k)] \quad (3.3-17b)$$

$$h_i^j [x(k)] = h_i^{j-1} [f(x(k), u(k))] \quad (3.3-17c)$$

Note-se que o grau relativo para o caso de um sistema de várias saídas é definido pelo vector constituído pelos graus relativos correspondentes a cada uma das saídas.

$$r = \begin{bmatrix} r_1 \\ \vdots \\ r_{n_y} \end{bmatrix} \quad (3.3-18)$$

Intuitivamente o grau relativo de uma saída  $y_i$  define o instante  $k = r_i$  para o qual a saída  $y_i$  é afectada por uma entrada aplicada no instante  $k = 0$ . É também importante notar que o grau relativo é invariante, não podendo ser modificado por realimentação e se o sistema possuir um grau relativo infinito significa que qualquer entrada não afecta a saída em causa, o mesmo é dizer, que é não controlável. Daí assumir-se sistemas com grau relativo finito.

#### Linearização exacta

Tal como na definição do problema para o caso contínuo, pretende-se determinar uma lei de realimentação de variáveis de estado (3.3-19), de forma a que o sistema não linear (3.3-15) seja transformado num linear.

$$u(k) = \psi(x(k), u(k)) \quad (3.3-19)$$

#### Teorema: (Nijmeijer e Van der Schaft, 1990)

Um sistema pode ser linearizado por realimentação se a matriz  $E(\cdot)$ , (3.3-20), designada matriz característica do sistema, for não singular.

$$E(x(k), u(k)) = \begin{bmatrix} \frac{\partial}{\partial u} \left( h_1^{r_1-1}[f(x,u)] \right) \\ \vdots \\ \frac{\partial}{\partial u} \left( h_{n_y}^{r_{n_y}-1}[f(x,u)] \right) \end{bmatrix} \quad (3.3-20)$$

O teorema anterior estabelece apenas condições necessárias para a existência de uma solução, não estabelecendo, contudo, resultados para a sua determinação. Na verdade, diga-se que esta é na maioria das vezes uma tarefa complexa, estando quase sempre posta de parte a obtenção de uma solução analítica. Mesmo no caso *affine*, à semelhança do problema contínuo (3.3-3), esta dificuldade mantém-se. Ao contrário do caso contínuo, em que o cálculo das derivadas de Lie preserva a linearidade da relação entrada-saída, tal não se passa no caso discreto. De facto, se se considerar um sistema discreto *affine*, (3.3-21), como um caso particular de (3.3-15) (introduzido na secção 2.2.4),

$$x(k+1) = f(x(k)) + g(x(k)) u(k) \quad (3.3-21a)$$

$$y(k) = h(x(k)) \quad (3.3-21b)$$

então a saída no instante seguinte (procedimento análogo ao cálculo de uma derivada no contínuo) é obtida por (3.3-22).

$$y(k+1) = h(x(k+1)) = h(f(x) + g(x)u(k)) \quad (3.3-22)$$

E, mesmo neste caso relativamente simples, a relação de linearidade entre a entrada e a saída não se preserva. Este facto impossibilita o cálculo da acção de controlo directamente tal como no caso contínuo, uma vez que a inversa desta função (3.3-22) (que permite determinar  $u(k)$ ) não é, regra geral, fácil de obter. Principalmente por esta razão a equação (3.3-21b) é por vezes substituída por uma relação linear, do tipo (3.3-23).

$$y(k) = Cx(k) \quad (3.3-23)$$

Um caso particular de sistemas *affine*, para o qual a técnica de linearização retroactiva pode directamente ser aplicada, é descrito em (3.3-24), estrutura essa já introduzida em (2.2-20).

$$y(k+1) = f(x(k)) + g(x(k))u(k) \quad (3.3-24)$$

O estado  $x(k)$  é usualmente definido à custa de regressões (valores passados) de entradas e saídas e, neste caso a matriz característica, (3.3-20), é dada por  $E = g(x(k))$ . Assim se  $g(x(k))$  for não singular, a acção de controlo que implementa a linearização retroactiva é facilmente obtida. No caso particular de um sistema de grau relativo  $r = 1$ , a seguinte lei de realimentação,

$$u(k) = g^{-1}(x(k)) [v(k) - f(x(k))] \quad (3.3-25)$$

permite obter directamente um sistema linear, (3.3-26), à semelhança do caso contínuo.

$$y(k+1) = v(k) \quad (3.3-26)$$

Como facilmente se percebe a linearização retroactiva pode ser interpretada como um controlador de tempo de estabelecimento finito (*dead-beat*) (Glad, 1997). E, se do ponto de vista teórico a resposta é exacta, em termos práticos, principalmente por questões de robustez, a sua implementação pode não constituir uma opção aconselhável (Isermann *et al.*, 1992).

### 3.3.3 Aplicação de Técnicas de Linearização Retroactiva a Redes Neurais

Foram introduzidas noções básicas de controlo não linear, em particular técnicas de linearização retroactiva, tanto no domínio contínuo como no discreto. Este facto justifica-se pela razão de que a grande maioria dos trabalhos que combinam técnicas neuronais e métodos de controlo não linear as utilizam. De seguida é feita uma revisão desses trabalhos.

#### Sistemas Contínuos

Um dos primeiros trabalhos a estabelecer uma relação rigorosa entre as técnicas de controlo adaptativo não linear e as redes neuronais deve-se a Hancock e Fallside (1991a). Concretamente, propõem um esquema de controlo para sistemas contínuos *affine* não lineares, (3.3-5), utilizando a técnica de linearização retroactiva, (3.3-13), estabelecendo resultados de estabilidade e convergência baseados na teoria de Lyapunov. A sua ideia principal consistia em utilizar as capacidades de aproximação das redes neuronais estáticas *RBFNN* para aproximar as derivadas de *Lie*, (3.3-6) e (3.3-7), assumindo que se conhecem os estados do sistema. Posteriormente, Hancock e Fallside, (1991b), sugerem o uso adicional de um termo de controlo de modo deslizante (*SMC*<sup>3</sup>) para compensar, por um lado incertezas de modelização e, por outro, para melhorar o desempenho do controlador.

Polycarpou e Ioannou (1992), também para a mesma classe de sistemas contínuos, desenvolveram resultados de estabilidade introduzindo arquitecturas neuronais *RBFNN* e *MLNN* para aproximar as derivadas de *Lie*. Tal como Hancock e Fallside, assumem o conhecimento dos estados do sistema e os seus resultados podem ser aplicados a sistemas de linearidades desconhecidas, limitados contudo a sistemas *SISO*.

Sanner e Slotine (1992) introduzem uma metodologia de controlo adaptativo directo para uma classe de sistemas não lineares, mais genérica que a de Polycarpou, para os quais não é possível obter uma parametrização explícita ou que contêm incertezas na sua dinâmica. Usam redes *RBFNN* e combinam as técnicas de linearização retroactiva com um termo corrector de modo deslizante assegurando a estabilidade global do sistema de controlo.

Para o caso particular de redes *MLNN*, Yesildirek e Lewis (1995), estabelecem resultados de estabilidade, aplicáveis a sistemas contínuos *SISO* descritos na forma normal, (3.3-12), considerando além da linearização retroactiva um termo adicional de modo deslizante.

Todos os casos citados têm em comum o facto de utilizarem redes *RBFNN* ou *MLNN* para aproximar as derivadas de *Lie*. Rovithakis e Christodoulou (1994), ao contrário do que tinha sido feito até então, consideram apenas uma rede recorrente, para uma classe particular de sistemas não lineares contínuos, (3.3-27), em que  $\sigma_1(\cdot)$  e  $\sigma_2(\cdot)$

---

<sup>3</sup> *SMC*- *Sliding Mode Control*

definem funções de activação do tipo sigmoidal e  $A$ ,  $B$  e  $D$  são matrizes de dimensões adequadas.

$$\dot{x}(t) = A x(t) + D \sigma_1(x(t)) + B \sigma_2(x(t)) u(t) \quad (3.3-27)$$

Estes são talvez dos primeiros a apresentarem trabalhos importantes sobre estabilidade utilizando redes neuronais recorrentes. Posteriormente Rovithakis e Christodoulou, (1995) e (1996) eliminam-se algumas das condições demasiado restritivas do seu trabalho original. Para o caso em que a rede neuronal tem uma estrutura adequada ao processo a identificar, apresentam resultados que asseguram a convergência do erro de controlo e ao mesmo tempo a limitação do valor de todos os sinais em malha fechada, mesmo no caso em que os parâmetros da rede não convirjam para os valores correctos. Posteriormente, estendem o problema ao caso em que o número de estados é diferente do número de entradas, (Rovithakis e Christodoulou, 1997).

Fabri e Kadirkamanathan (1996) propõem uma metodologia adaptativa para uma classe de sistemas contínuos *affine*, e usam redes RBFNN numa estrutura análoga à de Sanner e Slotine. A principal diferença existente no seu trabalho reside no facto de que a rede é construtiva, capaz de crescer de acordo com a localização espacial dos estados do sistema, resultando uma rede óptima, em termos de dimensão.

De todos os trabalhos existentes, que combinam redes neuronais recorrentes e técnicas de controlo não linear talvez seja o trabalho de Poznyak e Yu um dos melhor fundamentados. Para a mesma classe de redes recorrentes usadas por Rovithakis e Christodoulou, (3.3-27), propõem um controlador constituído por duas partes distintas: uma determinada a partir da lei de linearização retroactiva (baseada no modelo neuronal) e outra parte correctora, para compensar eventuais erros estruturais e de modelização.

Em Yu *et al.* (1998) é considerada um termo de compensação do tipo modo deslizante e são apresentados resultados de convergência e estabilidade no caso da identificação e apenas de limitação do erro, para o caso do controlo. Yu e Poznyak (1999a) propõem, em substituição do termo de modo deslizante, um termo adicional baseado na teoria de controlo óptimo. Em Yu *et al.* (1999b), é feita uma uniformização de metodologias propondo, além das duas formas de compensação por eles anteriormente sugeridas (modo deslizante e óptimo) outras duas, uma baseada num método exacto e outra num método aproximado. Poznyak *et al.* (1999) apresentam dois resultados fundamentais para o seu tipo de metodologia: um de convergência do erro de identificação, caso em que a rede neuronal reproduz correctamente o processo, ou apenas a sua limitação caso existam defeitos estruturais da rede; no outro estabelecem condições para a limitação do erro de controlo.

Apesar de existirem progressos significativos na incorporação de redes neuronais recorrentes em estruturas de controlo adaptativo não linear, de que são exemplo os trabalhos de Rovithakis e de Poznyak, a sua grande limitação de aplicação prática reside na necessidade de conhecimento dos estados.

Admitindo o princípio de separação, Li e Babuska (1999) propõem um método de controlo adaptativo que combina as técnicas de linearização retroactiva com redes RBFNN (usadas para aproximar as derivadas de  $Lie$ ) e um observador (*high gain observer*)

para estimar os estados do sistema. Os seus resultados são válidos para sistemas não lineares SISO, em tempo contínuo, descritos na forma normal (3.3-12).

Para o mesmo problema, Delgado *et al.* (1995), propõem o uso de redes recorrentes, do tipo (3.3-28), e estabelecem resultados de estabilidade num contexto não adaptativo para uma classe de sistemas contínuos SISO.

$$\dot{x}(t) = A x(t) + D \sigma(x(t)) + B u(t) \quad (3.3-28a)$$

$$y(t) = h(x(t)) \quad (3.3-28b)$$

A contribuição principal do trabalho de Delgado é a interpretação da rede como um observador, eliminando a necessidade do conhecimento dos estados do processo, ao contrário dos trabalhos de Rovithakis e Poznyak. A rede é previamente treinada para capturar a dinâmica (entrada-saída) do processo e é depois aplicada uma técnica de linearização retroactiva em que os estados realimentados são os fornecidos pela rede. Para ter em conta eventuais erros de modelização, sugerem um controlador *PID* adicional, na malha externa.

Finalmente referen-se os trabalhos de Brdys e Kulawski. Em Brdys *et al.* (1998) é proposta uma metodologia de controlo adaptativo indirecto para sistemas contínuos MIMO não lineares, de estados não acessíveis, combinando redes neuronais recorrentes do tipo (3.3-29), com a técnica de linearização retroactiva. Para a metodologia proposta são apresentados resultados de estabilidade, admitindo que o sistema a controlar é estável e a referência é constante.

$$\dot{x}(t) = A x(t) + D \sigma(x(t)) + B u(t) \quad (3.3-29a)$$

$$y(t) = C x(t) \quad (3.3-29b)$$

Recentemente, Kulawski e Brdys (2000), estendem os resultados anteriores para referências mais genéricas e provam a estabilidade da estrutura de controlo. Os seus resultados são válidos caso exista uma variação lenta dos parâmetros do processo, em relação ao processo de estimação, devendo este último assegurar um erro de modelização assintoticamente estável.

### Comentários finais

A título de conclusão, pode-se dizer que no caso contínuo, começam a existir resultados de estabilidade e convergência para técnicas de controlo adaptativo neuronal, aplicados a classes particulares de modelos neuronais, nomeadamente para sistemas *affine*. Estes resultados têm-se distinguido principalmente pela forma como as redes neuronais são utilizadas.

- i) No primeiro caso, as redes são do tipo *RBFNN* ou *MLNN* e são incorporadas com o objectivo de aproximar as derivadas de *Lie*, de forma a que seja depois possível efectuar um controlador por linearização retroactiva. Os estados podem ser assumidos como acessíveis, pertencendo a esta abordagem os trabalhos de Hancock e Fallside, Polycarpu e Ioannou, Sanner e Slotine,

Yesildirek e Lewis, ou estimados por um observador, como o trabalho de Li e Babuska;

- ii) No segundo caso é utilizada apenas uma rede neuronal (recorrente), para aproximar a dinâmica do processo. Distinguem-se, tal como no primeiro, os casos em que o controlador usa directamente os estados do processo (Rovithakis e Christodoulou, Poznyak e Yu) ou o caso em que são usados os estados estimados pela rede recorrente (Delgado, Bryds e Kulawski).

Em comum todas as abordagens têm o facto de utilizarem a teoria de Lyapunov para estabelecer resultados de estabilidade e convergência, e recorrendo a técnicas de linearização retroactiva para solucionar o problema do controlo de sistemas estáveis.

## Sistemas Discretos

Sem dúvida que o trabalho de Narendra e Parthasarathy (1990), incentivou de uma forma significativa o uso das redes neuronais no contexto da identificação e controlo de sistemas. Propuseram o uso de redes *MLNN* com ligações externas (*TDLNN*) de forma a incorporar dinâmica e mostraram, através de exemplos de simulação, que as redes neuronais podem efectivamente ser usadas no contexto da identificação e controlo adaptativo de sistemas. Contudo, os seus resultados não apresentaram provas teóricas, mas apenas resultados de simulação.

Mais tarde, em Levin e Narendra (1993) são investigados, de uma forma rigorosa, conceitos de controlo não linear usando redes neuronais. Concretamente, a estratégia seguida consiste na implementação de conceitos de linearização retroactiva treinando para isso três redes. A primeira com o objectivo de aproximar o sistema não linear (3.3-15) (assumindo o conhecimento dos estados), a segunda com o objectivo de determinar uma aproximação para a acção de controlo, (3.3-19), treinando ainda uma terceira rede de forma a reproduzir uma transformação de coordenadas do sistema não linear para uma forma normalizada, para a qual fosse possível implementar directamente a técnica da linearização retroactiva.

Para a demonstração dos seus resultados usam dois teoremas fundamentais (usados também pela maioria dos autores): o teorema da função inversa e o teorema da função implícita. Basicamente o primeiro assume que é possível efectuar a inversão local do mapeamento de entrada-saída, enquanto o segundo assume que a acção de controlo  $u(k)$ , capaz de conduzir a saída  $y(k)$  a um valor desejado  $y_d(k)$ , é única.

Levin e Narendra (1996) apresentam resultados semelhantes, assumindo neste caso o desconhecimento dos estados. Usando modelos de entrada-saída, são por eles estabelecidas duas contribuições fundamentais: a justificação analítica do uso de modelos de entrada-saída não lineares e a demonstração da existência de uma acção de controlo capaz de assegurar que a saída do sistema alcance o valor desejado em  $r$  instantes, sendo  $r$  o grau relativo do sistema.

Para o caso de redes recorrentes não adaptativas Suykens *et al.* (1996), (1997) estabelecem resultados teóricos de convergência e estabilidade. Entendendo a rede como

um modelo de estados não linear, (3.3-15), é efectuado o seu treino (em-diferido) de forma a captar a dinâmica do processo. Uma vez obtido o modelo neuronal, é implementada uma determinada transformação (*Linear Fractional Transformation*), sendo a rede interpretada como um modelo constituído por uma parte linear e outra não linear entendidas como incertezas. Introduzem a teoria  $NL_q$ , e estudam o problema do controlo seguindo uma abordagem robusta convencional.

Para o caso adaptativo destaca-se o trabalho de Chen (1990). Usando a técnica de linearização retroactiva incorpora, num esquemas de controlo adaptativo indirecto, uma classe de modelos entrada-saída *affine*, do tipo (3.3-24). Chen usa a capacidade das redes neuronais de três camadas *MLNN*, de forma a aproximar as funções  $f(x)$  e  $g(x)$ , (3.3-24), de uma forma análoga ao problema de aproximação das derivadas de *Lie* no caso contínuo. É usada apenas informação de entrada-saída, isto é, o estado é definido por

$$x(k) = [u(k-1), \dots, u(k-n_b), y(k-1), \dots, y(k-n_a)] \quad (3.3-30)$$

em que  $n_b$  e  $n_a$  representam o número de entradas e saídas passadas.

Posteriormente, Chen e Khalil (1992), apresentam para a mesma metodologia resultados de convergência local, limitados contudo, como os próprios autores assumem, a situações irrealistas. Em Chen e Khalil (1995), incorporando uma não linearidade no algoritmo de identificação (*dead-zone*), eliminam algumas das restrições anteriores e estendem os seus resultados de convergência a sistemas com um grau relativo genérico.

Em relação ao uso de redes recorrentes, se no caso contínuo se destacam os trabalhos de Poznyak e Brdys, no caso discreto destacam-se os trabalhos de Jin, Nikiforuk e Gupta. Estes têm apresentado excelentes contribuições não só no contexto do controlo neuronal, mas também noutras áreas, tais como na proposta de novas arquitecturas (Gupta e Rao, 1993), (Jin *et al.* 1994a), em resultados de aproximação de redes recorrentes, tanto no caso discreto (Jin *et al.*, 1995a) como no caso contínuo (Jin *et al.*, 1999a), em métodos de treino de redes recorrentes (Jin *et al.*, 1995b) e (Jin *et al.*, 1995c), na análise de estabilidade de redes recorrentes (Jin *et al.*, 1994c) e (Jin e Gupta, 1996) e em resultados de estabilidade de treino de redes recorrentes (Jin *et al.*, 1999b).

No caso particular do controlo adaptativo neuronal, Jin *et al.* (1994a) propõem uma classe particular de redes recorrentes (*Multilayered Recurrent Neural Networks*) num esquema de modelo de referência. A grande novidade do seu trabalho é não assumir qualquer restrição na forma da rede, eliminando a necessidade de modelos *affine*, mas impossibilitando por isso a aplicação directa das técnicas de linearização retroactiva para solucionar o problema de controlo. Para o efeito recorrem a métodos numéricos, usando em particular o conceito de controlo equivalente.

Os mesmos autores, Jin *et al.* (1994b), estendem os resultados anteriores a redes recorrentes genéricas, (3.3-15), sem contudo, tal como no caso anterior, apresentarem quaisquer resultados de estabilidade. Mais tarde Jin *et al.* (1995b) e Jin *et al.* (1995c) propõem o uso das capacidades das redes *MLNN* para identificação de modelos do tipo (3.3-24). Como contribuição principal destaca-se a introdução de um novo algoritmo de

treino (*Weight Decoupled Extended Kalman Filter*), como uma variante do filtro de Kalman estendido.

Para esquemas de controlo directo salienta-se o trabalho de Jagannathan *et al.* (1996b) em que uma rede *MLNN* é incorporada num esquema de controlo adaptativo *MRAC* aplicável a uma classe específica de sistemas (forma normal). De forma a assegurar a estabilidade, os parâmetros da rede são adaptados usando uma modificação do algoritmo de retropropagação, considerando uma alteração no coeficiente de aprendizagem e introduzindo um termo adicional. Como contribuição principal para a sua abordagem, reclamam a não necessidade de se verificar a condição de persistência de excitação do sinal de entrada nem do princípio da equivalência certa. Como desvantagem, a necessidade de conhecimento dos estados.

Tal como Jin *et al.* (1994b), e com o objectivo de generalização a modelos discretos genéricos, não necessariamente da classe *affine*, Te Braake *et al.* (1997) propõe o uso de modelos não lineares genéricos de entrada-saída, (3.3-31), em que o estado  $x(k)$  é definido à custa de regressões (valores passados) de entradas e saídas (3.3-30).

$$y(k+1) = f(x(k), u(k)) \quad (3.3-31)$$

Utilizam uma rede neuronal *MLNN* para aproximar a dinâmica do sistema, (3.3-31), e para o cálculo da acção de controlo introduzem um método aproximado, obtido a partir de uma linearização local, conseguida à custa de uma expansão de Taylor (com o mesmo objectivo Jin utiliza um método numérico). O sistema linearizado resultante é depois inserido num esquema de controlo linear preditivo.

Porque a linearização retroactiva transforma um problema original de restrições lineares num problema de restrições não lineares, não é possível a aplicação directa de métodos de programação quadrática (com restrições) quando combinados com técnicas de controlo preditivo. Botto (1996) e Botto *et al.* (1999) propõem um método aproximado, permitindo usar algoritmos de programação quadrática incorporados num esquema de controlo preditivo, em que o sistema não linear discreto é representado por uma rede neuronal *affine*. Te Braake *et al.* (1999) combinam as ideias anteriores (Te Braake *et al.*, 1997) e (Botto *et al.*, 1999) resultando uma metodologia de controlo preditivo, com o sistema descrito por (3.3-31), para a qual podem ser impostas restrições (nas entradas).

### Comentários finais

Tal como no caso contínuo, distinguem-se duas abordagens principais na combinação de redes neuronais com técnicas de controlo não linear:

- i) Na primeira abordagem são usados modelos *affine* entrada-saída do tipo (3.3-24), e são utilizadas redes estáticas (*RBFFNN* ou *MLNN*) para aproximar as funções  $f(x(k))$  e  $g(k)$ , num procedimento semelhante à aproximação das derivadas de Lie, no caso contínuo. Inserem-se neste contexto os trabalhos de Narendra, Levin e Parthasarathy, Chen e Khalil, Jagannathan e Lewis, Jin, Nikiforuk e Gupta, Te Braake e Botto;

- ii) Na segunda abordagem considera-se um modelo de estados neuronal (recorrente) e usam-se os estados da rede na determinação da acção de controlo. Exemplos são os trabalhos de Suykens, num contexto não adaptativo e os de Jin *et al.* (1994b) num contexto adaptativo.

Contudo, se para o caso de sistemas contínuos têm surgido resultados de estabilidade tanto para modelos entrada-saída como para modelos de estado, não têm existido progressos semelhantes no caso dos sistemas adaptativos discretos. Assim, para o caso de modelos entrada-saída tem sido estabelecidos resultados, dos quais os trabalhos de Chen e Khalil são referência. Para o caso de modelos de estados não se conhecem quaisquer resultados, referindo-se que o único trabalho conhecido neste contexto (redes recorrentes) é o de Jin *et al.* (1994b), que no entanto não apresenta quaisquer estudos de estabilidade.

Uma das razões que de alguma forma justificam esta escassez de resultados prende-se com o facto de a relação *affine* não poder ser preservada em sistema discretos, (3.3-22), (Isidori e Byrnes, 1990), o que dificulta a solução do problema de controlo e daí a razão do uso de modelos do tipo (3.3-24).

Outra razão fundamental prende-se com a aplicação da teoria de Lyapunov. Em tempo contínuo a função de Lyapunov pode ser escolhida de tal forma que a derivada do erro seja linear, propriedade que não é mantida para sistemas discretos.

### 3.3.4 Estrutura de Controlo Proposta

O objectivo inicialmente proposto considera o desenvolvimento de metodologias de controlo híbridas, baseadas em métodos rigorosos, nomeadamente o estudo das propriedades de estabilidade e convergência. Assim, tendo em vista esse fim, subdivide-se o problema em duas partes distintas, identificação e controlo.

Quanto ao primeiro problema, o de identificação, este foi introduzido no capítulo 2. Foram introduzidas técnicas capazes de adaptar os parâmetros de um modelo neuronal recorrente, justificando-se na altura a opção por técnicas de observação.

Quanto ao segundo, opta-se nesta dissertação por uma técnica de controlo não linear, concretamente a teoria da regulação da saída, (*output regulation*). O uso desta técnica justifica-se principalmente pelas suas características de estabilidade e por incorporar, quando verificadas determinadas condições, um princípio de separação. Adicionalmente, devido ao tipo de não linearidade em questão (tangente hiperbólica), a solução do problema de controlo (a ser definido por um sistema de equações de diferenças) é possível de solucionar eficazmente.

Por outro lado, a regulação da saída tem a vantagem, relativamente à técnica de linearização retroactiva, de poder ser aplicada a sistemas de fase não mínima e de a sua solução não conduzir a valores de actuação demasiado bruscos. De facto, a técnica de linearização retroactiva resulta num controlador do tipo *dead-beat*, sendo possível na técnica de regulação de saída especificar a velocidade de convergência, através de uma localização adequada de valores próprios do sistema em malha fechada.

# 3.4

## Conclusão

**S**e a aplicação das redes neuronais ao controlo se caracterizou numa fase inicial por se basear em técnicas empíricas, assiste-se actualmente a uma tentativa de trazer para o contexto das técnicas de controlo convencional redes neuronais, capazes de analisar propriedades básicas de estabilidade, convergência e robustez.

O ponto de contacto fundamental entre as redes neuronais e a teoria de controlo deve-se à capacidade de aproximação das redes neuronais, permitindo que o processo a controlar possa ser substituído por um modelo neuronal de estrutura fixa, mas de parâmetros variáveis, transformando o problema original de controlo em dois problemas parciais: um problema de identificação (dos parâmetros da rede) e outro de projecto de um controlador, a ser efectuado com base no modelo neuronal.

Pelas suas características de generalidade, (no sentido de poderem ser aplicadas a uma grande variedade de processos) e porque são constituídas por não linearidades bem definidas (por exemplo, funções do tipo sigmoideal), perfeitamente tratáveis do ponto de vista matemático, os modelos neuronais permitem estabelecer as bases para uma abordagem sistemática no projecto de um controlador.

As técnicas de controlo não linear, que têm directamente em conta o tipo de não linearidade presente nas redes neuronais, surgem naturalmente como o meio adequado para o projecto do controlador.

Nesta secção abordou-se o problema de controlo considerando metodologias que incorporam na sua estrutura redes neuronais.

Concretamente, foram revistos conceitos de controlo convencional, controlo adaptativo e analisadas formas de incorporar redes neuronais em estruturas de controlo. Apresentou-se uma classificação para essas estruturas neuronais de controlo, introduzindo-se alguns trabalhos desenvolvidos durante a elaboração desta dissertação como exemplos particulares dessas estruturas. Justifica-se ainda nesta secção o tipo de arquitectura a implementar, uma estrutura híbrida, consistindo numa rede recorrente (a servir de modelo) sendo o controlador determinando por técnicas de controlo não linear.

Esta estrutura, do ponto de vista convencional, é entendida como uma estrutura de controlo não linear adaptativa indirecta.

Foram igualmente introduzidos conceitos de controlo não linear, em particular a técnica de linearização retroactiva, utilizada na maioria dos trabalhos que combinam conceitos rigorosos de controlo adaptativo com redes neuronais do tipo *affine*.

# 4.

## Regulação da Saída com Redes Neurais Recorrentes

*“Recently, the potential of artificial neural networks in control problems has attracted much attention. Many problems such as identification, input-output linearization, stabilization and tracking have been addressed in the literature. The use of the regulation theory combined with a dynamic neural identifier, to obtain robust trajectory tracking for non linear systems, is one of these ideas.”*

B. Castillo-Toledo, 1999.

**A**presenta-se neste capítulo a metodologia de controlo proposta nesta dissertação, *RSRN- Regulação da Saída com Redes Neurais Recorrentes*. Combina-se, numa estrutura adaptativa, uma rede neuronal recorrente do tipo *affine* com a teoria de regulação da saída, uma técnica de controlo não linear.

A teoria de controlo não linear, de que é exemplo a linearização retroactiva e a teoria de regulação, proporciona um meio eficaz de solucionar o problema de controlo de sistemas não lineares tendo por base o conhecimento exacto do sistema a controlar, descrito por um modelo matemático.

Principalmente pelas suas propriedades de aproximação e pela possibilidade da sua interpretação como um modelo de estados não lineares, propõe-se a utilização de uma rede recorrente para representar o sistema a controlar. Desta forma, formula-se o problema de controlo como um problema não linear, adequado ao projecto de controladores que usam técnicas não lineares. Na Figura 4.0-1 mostra-se a estrutura utilizada, considerando a teoria de regulação como técnica de controlo não linear.

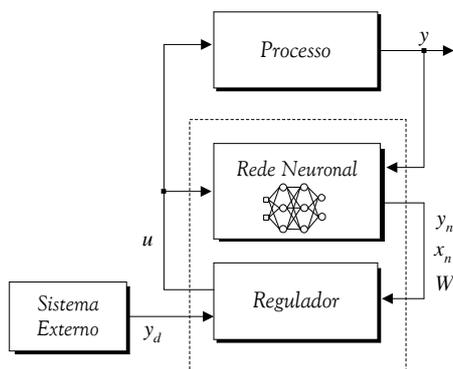


Figura 4.0-1: Estrutura de controle RSRN.

A teoria de regulação permite determinar a acção de controlo  $u$  que força a saída do modelo  $y_n$  a convergir para um valor desejado  $y_d$  (gerado por um sistema externo), garantindo simultaneamente que o sistema em malha fechada seja estável. No entanto, a sua determinação passa pela resolução de um sistema de equações não linear, conhecido por equações de regulação, para o qual é difícil, ou mesmo impossível, obter uma solução analítica.

Uma das contribuições desta dissertação consiste no desenvolvimento de um algoritmo capaz de obter uma solução, de uma forma aproximada, para o sistema de equações em questão. Em concreto, propõe-se um algoritmo iterativo, específico para a estrutura neuronal proposta e para o tipo de não linearidade em questão (funções *Lipschitz*), capaz de assegurar a convergência das equações de regulação.

Contudo, se a determinação dos parâmetros que caracterizam o modelo neuronal for efectuada *a priori* não se pode esperar que o modelo neuronal seja preciso. Existem sempre erros de modelização e variações do próprio sistema. Assim, de forma a ultrapassar o problema, considera-se o modelo neuronal inserido numa estrutura adaptativa, na qual os seus parâmetros são adaptados a partir de dados de entrada-saída, garantindo a convergência da saída do modelo neuronal  $y_n$  para a do processo  $y$ .

Para o efeito, interpreta-se a rede recorrente como um observador e propõe-se, por analogia com um filtro de Kalman dual, um método capaz de simultaneamente estimar os seus parâmetros  $W$  e os seus estados  $x_n$ . Recorrendo à teoria de Lyapunov e a técnicas de observação não linear assegura-se a estabilidade e convergência do método proposto.

Este capítulo encontra-se dividido em quatro secções. Na secção 4.1 propõe-se e justifica-se a estrutura neuronal recorrente *affine* usada como modelo genérico para o sistema a controlar propondo-se, na secção 4.2, uma metodologia para a estimação dos seus parâmetros e dos seus estados. Na secção 4.3 revê-se a teoria de regulação da saída e a forma de a aplicar ao modelo neuronal recorrente. Na secção 4.4 apresentam-se algumas conclusões relevantes.

# 4.1

## Estrutura Neuronal Proposta

*De entre os modelos neuronais destacou-se na secção 2.2 as redes recorrentes, pela possibilidade de incorporarem dinâmica.*

*Justifica-se nesta secção a estrutura particular proposta, uma estrutura recorrente do tipo affine.*

### 4.1.1 Estruturas *Affine*

De entre as várias estruturas descritas na secção 2.2 deu-se especial atenção aos modelos neuronais recorrentes e, de entre estes, referiram-se as estruturas do tipo *affine* pelas potencialidades que permitem no desenvolvimento, quer de metodologias de estimação dos seus parâmetros, quer de aspectos relacionados com o projecto de controladores.

Chegado o momento de ter de considerar uma determinada estrutura, vários aspectos têm de ser tidos em conta:

- i) preferência por modelos entrada-saída ou por modelos de estados;
- ii) capacidade de aproximação: idealmente dever-se-á ter a garantia de que a estrutura a utilizar permita aproximar o processo a modelizar;
- iii) facilidade de projecto do controlador, isto é, a complexidade relativa à determinação dos parâmetros do controlador a partir do modelo neuronal;
- iv) facilidade de estimação dos parâmetros da rede e, se possível, incorporação de conhecimento existente, regra geral, sob a forma de um modelo de estados ou sob a forma de uma função de transferência;
- v) facilidade de análise das propriedades fundamentais de estabilidade e convergência, quer do ponto de vista de identificação, quer do ponto de vista do sistema de controlo.

## 4.1.2 Rede Neuronal Recorrente Proposta

Antes de mais, e pelas razões já expostas na secção 2.2, optou-se por modelos de estados em detrimento de modelos entrada-saída. E, de entre os primeiros, optou-se por um que de alguma forma estabeleça um compromisso entre os requisitos anteriores.

Considera-se como modelo neuronal recorrente a seguinte estrutura *affine*, descrita na equação (4.1-1) e representada na Figura 4.1-1.

$$x_n(k+1) = A x_n(k) + D \sigma(x_n(k)) + B u(k) \quad (4.1-1a)$$

$$y_n(k) = C x_n(k) \quad (4.1-1b)$$

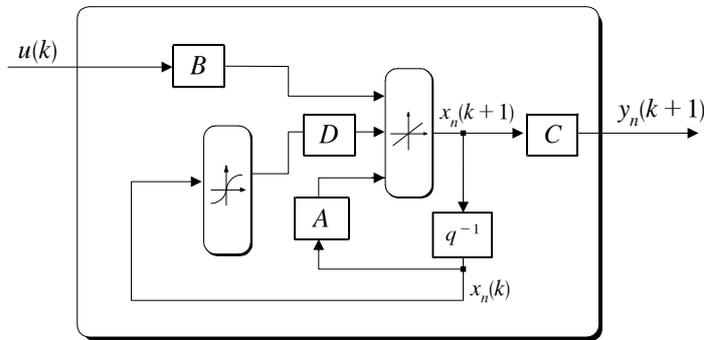


Figura 4.1-1: Estrutura proposta.

A variável  $x_n \in \mathfrak{R}^n$  representa a saída da camada interna,  $u \in \mathfrak{R}^{n_u}$  e  $y_n \in \mathfrak{R}^{n_y}$  definem, respectivamente, a entrada e a saída da rede. A função de activação  $\sigma(\cdot)$  é uma tangente hiperbólica que, aplicada a um vector  $x \in \mathfrak{R}^n$ , tem como resultado um outro vector, com se descreve em (4.1-2).

$$\sigma(x) = \sigma \left( \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = \begin{bmatrix} \sigma(x_1) \\ \vdots \\ \sigma(x_n) \end{bmatrix} \quad (4.1-2)$$

$A \in \mathfrak{R}^{n,n}$ ,  $B \in \mathfrak{R}^{n,n_u}$ ,  $C \in \mathfrak{R}^{n_y,n}$  e  $D \in \mathfrak{R}^{n,n}$  são matrizes, cujos valores se pretende determinar. Além disso, admite-se, por razões a explicar oportunamente, que os valores próprios da matriz  $A$  se localizam no interior do círculo de raio unitário. A saída da unidade interna, dada por (4.1-1a), pode ser interpretada como um vector de estados, à semelhança de um sistema descrito no espaço de estados. Zbikowski (1994) usa o termo *hiper-estado* para se referir a esta analogia.

Recordando a classificação introduzida por Moreau (2.2-15), poder-se-á designar esta estrutura por rede de Hopfield genérica, com entrada adicional exógena. Pode também ser entendida como uma estrutura mista, isto é, constituída por uma parte linear e uma

não linear. Para a sua definição contribuíram os trabalhos de Elman (1990), Delgado *et al.* (1995), Jin *et al.* (1995a), Rovithakis *et al.* (1997), Poznyak *et al.* (1999), Yu *et al.* (1999b) e Brdys *et al.* (1998) e Kulawski e Brdys (2000).

Começa-se por se referir que foram as redes de Elman e o trabalho de Jin que proporcionaram a primeira ideia para a estrutura a usar nesta dissertação. Baseados nos resultados relativos a aproximação de redes *MLNN*, Jin *et al.* (1995a), demonstraram as capacidades de aproximação para uma classe específica de redes recorrentes. Concretamente para a rede neuronal definida por (4.1-3),

$$\begin{aligned}x_n(k+1) &= \Lambda x_n(k) + \sigma(A x_n(k) + B u(k)) \\y_n(k) &= C x_n(k)\end{aligned}\tag{4.1-3}$$

mostraram a sua capacidade de aproximação de uma qualquer trajectória limitada originada por um sistema discreto, descrito no espaço de estados, seja ele realizado por um sistema dinâmico ou por uma função contínua.  $A$ ,  $B$  e  $C$  são matrizes de dimensões adequadas e  $\Lambda$  é uma matriz diagonal, cujos elementos  $\lambda_i \in ]0,1[$  são constantes.

Refira-se que o teorema correspondente para o caso contínuo foi introduzido independentemente por Sontag (1992) e Funahashi e Nakamura (1993). Note-se, contudo, que quanto ao meio de caracterizar a estrutura em questão, isto é, o número de neurónios necessário e o método de estimação para o conseguir, não é dada qualquer informação. Ou seja, os resultados provam apenas a possibilidade teórica da existência de uma solução para o problema, mas não de como a encontrar.

Note-se contudo que os resultados de Jin *et al.* dizem respeito a redes do tipo não *affine* e não a redes *affine*, como é o caso da estrutura proposta. No entanto, apesar de não existirem resultados teóricos, pelo menos que se conheça, que provem as capacidades de aproximação universal de estruturas *affine* em tempo discreto, tal facto não invalida a sua aplicação. Kulawski e Brdys (2000) sugerem em relação a este problema, o uso de neurónios adicionais na camada interna, além dos julgados necessários para um aproximador universal (casos da rede de Elman), para colmatar a eventual limitação deste tipo de estruturas.

Ainda em relação a um qualquer modelo neuronal, para o qual existam resultados teóricos de aproximação, será legítimo esperar que a aproximação gerada por esse modelo seja perfeita? Se em teoria sim, na prática dificilmente tal facto se observaria, uma vez que o número de neurónios na camada interna seria necessariamente elevado. E se assim fosse as dificuldades tanto de identificação como de projecto de controladores tornar-se-iam decerto proibitivas.

Por outro lado, porque se incorpora uma estratégia adaptativa, esta permite ultrapassar em certa medida o problema, uma vez que os parâmetros da rede são ajustados permanentemente.

Em segundo lugar, a estrutura proposta neste trabalho resulta da inspiração dos trabalhos de Poznyak e Yu, Rovithakis e Christodoulou, Brdys e Kulawski, onde se considera uma estrutura *affine* de forma a facilitar, essencialmente, a análise de

estabilidade do processo de identificação. Poznyak e Yu, e Rovithakis e Christodoulou consideram uma estrutura neuronal do tipo (3.3-27),

$$\dot{x}_n(t) = A x_n(t) + D \sigma_1(x_n(t)) + B \sigma_2(x_n(t)) u(t)$$

enquanto Brdys e Kulawski, propõem a seguinte estrutura (3.3-29),

$$\begin{aligned} \dot{x}_n(t) &= \Lambda_A x_n(t) + \Lambda_D \sigma(x_n(t)) + B u(t) \\ y_n(t) &= C x_n(t) \end{aligned}$$

em que as matrizes  $\Lambda_A$  e  $\Lambda_D$  são diagonais, definido assim uma recorrência local.

A contribuição principal deste tipo de trabalhos consiste no desenvolvimento de metodologias de estimação baseadas na teoria de estabilidade de Lyapunov. Contudo, qualquer um deles considera sistemas descritos em tempo contínuo, para os quais os resultados de estabilidade são mais fáceis de estabelecer que no caso discreto. Mais ainda, Poznyak, Yu, Rovithakis e Christodoulou, assumem o conhecimento dos estados do sistema, não sendo tal limitação imposta por Brdys e Kulawski. Em ambos os casos, é assumido que a matriz da parte linear ( $A$  ou  $\Lambda_A$ ) é Hurwitz (valores próprios no semi-plano esquerdo).

Ainda em relação à estimação de parâmetros, um dos aspectos a salientar na estrutura proposta, diz respeito à sua eficácia. Porque as não linearidades são apenas na informação, sendo o modelo linear nos parâmetros, é relativamente fácil desenvolver leis de adaptação que permitam a análise de estabilidade e convergência, tal como foi seguido nos trabalhos de Rovithakis e Poznyak. Partindo desta ideia, já alguns trabalhos foram conduzidos, nomeadamente no caso das redes RBFNN, redes neuronais de Volterra (Rayner, 1989) e *Functional Link Neural Networks* (Hussain *et al.*, 1997).

Outro aspecto que pode ser entendido como uma vantagem é a interpretação como um observador de estruturas análogas à proposta, tendo sido explorado por Raol (1995), Delgado *et al.* (1995), Pham e Xing (1995a), Hwang e Sung (1996), Sørensen (1996), Kim *et al.* (1997), Henriques e Dourado (1998c) e (1999a), Ahmed e Riyaz (2000) e Yu *et al.* (2000).

Assim, dados os valores de entrada  $u(k)$  e de saída  $y(k)$  pretende-se estimar o valor do seu estado  $x(k)$ . No caso particular das redes, o problema de observação consiste na determinação da saída do neurónio da camada interna. Contudo acresce a dificuldade pelo facto de não se conhecerem também os parâmetros da rede.

Finalmente, justifica-se a estrutura proposta pela inspiração no trabalho de Suykens *et al.* (1996). Após uma determinada transformação Suykens *et al.* interpretam uma rede recorrente como um modelo constituído por duas partes: uma linear e outra não linear, ou como eles o entendem, um modelo linear com incertezas.

Entendida desta forma, a rede não é propriamente um modelo *black-box*, sendo possível fornecer estimativas iniciais para o seu treino. Assim, técnicas bem conhecidas de identificação linear usando métodos de mínimos quadráticos (Ljung, 1999), sub-

espaços (De Moor *et al.* 1999), e CVA<sup>1</sup> (Larimore e Seborg, 1999), podem ser utilizadas para gerar um modelo linear de modo a proporcionar a sua incorporação prévia na rede. Além do mais, esta visão de modelo linear com incertezas tem uma aplicação directa no projecto de controladores que usem técnicas de controlo linear robustas, tal como é sugerido e implementado em (Suykens *et al.*, 1996).

Em conclusão, a metodologia proposta pode ser entendida como uma tentativa de estabelecer um compromisso entre dois objectivos fundamentais: o desenvolvimento de estruturas que sirvam de base à modelização de um sistema não linear genérico mas ao mesmo tempo que seja útil para as tarefas de estimação dos seus parâmetros e projecto de um controlador.

Formalmente, o tipo de estrutura proposta pode ser entendida como constituída por uma parte linear e outra não linear bem definida ( $\sigma$  - tangente hiperbólica). E apesar das linearidades serem bem definidas, todas do mesmo tipo e perfeitamente tratáveis do ponto de vista matemático, constituem modelos genéricos com capacidade de aproximação, sendo este um dos aspectos mais relevantes atractivos das redes neuronais.

---

<sup>1</sup> CVA- Canonical Variate Analysis

## 4.2

# Estimação de Parâmetros na Metodologia RSRN

*Da literatura pode concluir-se que as redes neuronais recorrentes, interpretadas como observadores, são particularmente eficientes na identificação e controlo de sistemas não lineares.*

*Propõe-se nesta secção, à semelhança de um filtro de Kalman dual, uma metodologia para estimação de estados e de parâmetros da rede. Enunciam-se e demonstram-se dois teoremas para a metodologia RSRN: um para a estimação de estados e outro para a adaptação de parâmetros.*

### 4.2.1 Estimação Simultânea de Parâmetros e Estados

Por analogia com os procedimentos de estimação e de observação refere-se daqui em diante o termo adaptação no caso da determinação dos parâmetros da rede ( $W = \hat{\theta}$ ) e o termo estimação no caso de determinação dos seus estados (activação da unidade interna,  $x_n = \hat{x}$ ). O esquema de estimação e de adaptação proposto é mostrado esquematicamente na Figura 4.2-1.

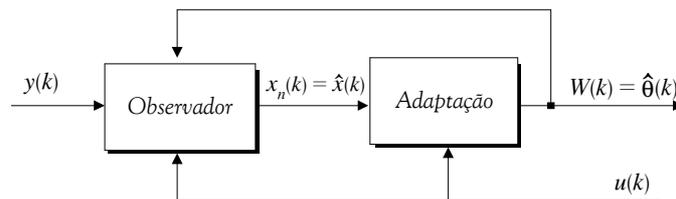


Figura 4.2-1: Estimação de estados e adaptação de parâmetros.

Para o devido efeito, consideram-se os procedimentos de estimação de estados e de adaptação de parâmetros tal como se fossem independentes.

Assim, para o primeiro problema, o da adaptação dos parâmetros da rede, é proposto um método resultante da aplicação da teoria de estabilidade de Lyapunov, baseado em resultados estabelecidos no contexto de controlo (Byrnes *et al.*, 1983). É ainda proposto um método iterativo, baseado num algoritmo de localização de valores próprios que tem em consideração o tipo de não linearidade em questão (*Lipschitz*), o qual será apresentado na secção 4.3.

Para o problema de estimação de estados foram referidas (secção 2.3) quatro possibilidades de técnicas de observação possíveis de aplicação aos modelos recorrentes em estudo: de linearização, de verificação, de modo deslizante e integrais.

Tendo em atenção o modelo proposto (4.1-1) optou-se por implementar o segundo método, uma vez que é o que mais se adequa ao problema em causa. Em concreto, é proposta a extensão para o caso discreto de resultados estabelecidos para o caso contínuo (Thau, 1973), tendo por base o conhecimento do tipo de não linearidade. Para o mesmo problema, é ainda proposta a formulação do problema de observação utilizando um observador de modo deslizante, a partir dos resultados de Mahamod e Bahnasawi, (1994). Ao contrário do método de Thau, que assume o conhecimento do tipo de não linearidade, neste método assume-se apenas a existência de um valor limite para a não linearidade.

## Formulação do Problema

Seja o processo genérico não linear que se pretende identificar, descrito no espaço de estados por (4.2-1).

$$x_p(k+1) = f(x_p(k), u(k)) \quad (4.2-1a)$$

$$y(k) = C x_p(k) \quad (4.2-1b)$$

O vector  $x_p \in \mathfrak{R}^{n_p}$  define o estado do processo real, que se assume desconhecido,  $u \in \mathfrak{R}^{n_u}$  e  $y \in \mathfrak{R}^{n_y}$  definem, respectivamente, a entrada e a saída do processo.

Tendo por base as capacidades de aproximação das redes recorrentes, considere-se uma rede recorrente, (4.1-1), como passível de representar convenientemente o sistema (4.2-1). Seja então

$$x_s(k+1) = A^* x_s(k) + B^* u(k) + D^* \sigma(x_s(k)) \quad (4.2-2a)$$

$$y_s(k) = C^* x_s(k) \quad (4.2-2b)$$

em que o vector  $x_s \in \mathfrak{R}^{n_s}$  define o estado do modelo neuronal, não necessariamente da mesma ordem de  $x_p \in \mathfrak{R}^{n_p}$ . Pela mesma razão (capacidade de aproximação), assume-se existirem matrizes  $A^*$ ,  $B^*$ ,  $C^*$  e  $D^*$ , de valores desconhecidos, que permitem aproximar a

dinâmica do processo real. Assume-se ainda que a matriz  $A^*$  é Hurwitz (valores próprios no interior do círculo de raio unitário) e que o par  $(A^*, C^*)$  é observável.

A estrutura considerada pode ser entendida como uma estrutura mista, constituída por uma parte linear e outra não linear, descrita por (4.2-3),

$$x_s(k+1) = A x_s(k) + B u(k) + \zeta(x_s(k)) \quad (4.2-3a)$$

$$y_s(k) = C x_s(k) \quad (4.2-3b)$$

em que

$$\zeta(x_s(k)) = D \sigma(x_s(k)) \quad (4.2-4)$$

Ou seja, a não linearidade é originada pela incorporação da função tangente hiperbólica  $\sigma(\cdot)$ , uma função pertencente à classe de funções *Lipschitz*.

#### Definição

Uma função  $f(\cdot)$  diz-se *Lipschitz* se existe uma constante finita  $\gamma$ , denominada constante de *Lipschitz*, de tal forma que para todo o  $x_1, x_2 \in \mathfrak{R}$  se verifica a seguinte equação

$$\exists \gamma \in \mathfrak{R}, \forall x_1, x_2 \in \mathfrak{R} : \angle f(x_1) - f(x_2) \angle \leq \gamma \angle x_1 - x_2 \angle \quad (4.2-5)$$

Partindo do princípio que existe uma rede neuronal capaz de descrever a dinâmica do processo (4.2-2) e considerando que após a fase inicial de treino se consideram fixas e conhecidas as matrizes  $A = A^*$  e  $C = C^*$  (determinadas em-diferido pelo algoritmo de Levenberg-Marquardt) pretende-se actualizar em-linha os valores de  $B(k)$  e  $D(k)$  correspondentes a  $B^*$  e  $D^*$ . Adicionalmente, assume-se ainda o desconhecimento do estado da rede  $x_s(k)$  projectando-se um observador para o estimar, do tipo (4.2-6).

$$x_n(k+1) = \vartheta(A, B(k), D(k), x_n(k), y_n(k), y_s(k), u(k)) \quad (4.2-6a)$$

$$y_n(k) = C x_n(k) \quad (4.2-6b)$$

Formulado assim o problema, pretende-se determinar os estados do modelo neuronal, isto é, a saída da unidade interna,  $x_n(k) \in \mathfrak{R}^n$  e simultaneamente os parâmetros que o caracterizam, as matrizes  $B(k) \in \mathfrak{R}^{n,u}$  e  $D(k) \in \mathfrak{R}^{n,n}$  ( $A$  e  $C$  assumem-se conhecidas).

É importante notar que, na verdade, a rede não se comporta como um observador, no sentido mais estrito. De facto, não se espera em situação alguma determinar com rigor

o valor do estado do processo real  $x_p$ , que pode até nem ter significado físico. O que se pretende de facto é:

- i) tendo em atenção que pode existir uma infinidade de representações no espaço de estados para um processo, pretende-se determinar uma delas e o consequente vector de estado ( $x_s$ ) de forma a que a saída do processo real ( $y$ ) e do modelo neuronal ( $y_s$ ) sejam idênticas;
- ii) assumir que é possível, através de um processo de identificação, determinar os valores das matrizes da rede neuronal, de forma que os estados  $x_n$  e as saídas  $y_n$  do modelo neuronal a determinar convirjam para os valores do modelo neuronal ideal, respectivamente  $x_s$  e  $y_s$ .

## 4.2.2 Estimação de Estados

O modelo neuronal em questão, como já referido, pode ser entendido como sendo constituído por duas partes: uma linear e outra não linear. A não linearidade é devida à tangente hiperbólica e tem um papel predominante na opção da metodologia para efectuar a observação dos estados. Assim, em função do conhecimento que se assume para a não linearidade, pode-se decidir pela técnica de observação a seguir.

- i) se se assumir o conhecimento da não linearidade (função do tipo *Lipschitz*, (4.2-5)) pode-se projectar um observador baseado na teoria de observação para sistemas não lineares.
- ii) se se assumir a não linearidade apenas como uma incerteza limitada (a função tangente hiperbólica é limitada  $\sigma(x) \in [-1,1]$ ) então aborda-se o problema segundo a teoria de observação para sistemas robustos.

## Aplicação da Teoria de Observação Não Linear

Propõe-se de seguida estender ao modelo neuronal discreto adoptado nesta dissertação os resultados estabelecidos por Thau (1973), para o domínio contínuo. Para o efeito, enuncie-se o seguinte teorema:

### **Teorema (Thau, 1973):**

Seja o sistema contínuo (4.2-7).

$$\dot{x}(t) = A x(t) + B u(t) + \zeta(x, u, t) \quad (4.2-7a)$$

$$y(t) = C x(t) \quad (4.2-7b)$$

Seja ainda um observador definido por (4.2-8).

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B u(t) + \zeta(\hat{x}, u, t) + L (y(t) - C \hat{x}(t)) \quad (4.2-8)$$

Das equações que definem o sistema e observador é possível definir o erro de observação  $\varepsilon(t) = x(t) - \hat{x}(t)$  por (4.2-9).

$$\dot{\varepsilon}(t) = (A - LC) \varepsilon(t) + \zeta(x, u, t) - \zeta(\hat{x}, u, t) \quad (4.2-9)$$

Se for verificada a relação (4.2-10)

$$\gamma < \frac{\lambda_{\min}(Q)}{2 \lambda_{\max}(P)} \quad (4.2-10)$$

em que  $\gamma$  é a constante de Lipschitz e  $P$  e  $Q$  são matrizes simétricas positivas definidas, satisfazendo a equação de Lyapunov (4.2-11)

$$(A - LC)^T P + P (A - LC) = -Q \quad (4.2-11)$$

então garante-se a convergência assintótica do erro de estimação, isto é,

$$\lim_{t \rightarrow \infty} \varepsilon(t) = 0 \quad (4.2-12)$$

Uma vez que o objectivo do projecto de um observador é a determinação do ganho  $L$  que posiciona os valores próprios da matriz  $(A - LC)$  em locais desejados, e uma vez que não existe uma relação directa entre os valores próprios de  $(A - LC)$  e  $P$ , percebe-se o porquê de não se considerar este método como um método de projecto mas apenas de verificação.

A partir da formulação do teorema anterior estabelece-se nesta dissertação o seguinte resultado.

### Teorema 1: Estimação de estados na Metodologia RSRN

Considere-se um sistema discreto genérico, que se assume possível de descrever pela seguinte rede neuronal recorrente (4.2-2),

$$x_s(k+1) = A x_s(k) + B u(k) + D \sigma(x_s(k))$$

$$y_s(k) = C x_s(k)$$

em que a não linearidade é do tipo Lipschitz, caracterizada por uma constante  $\gamma$ . Seja ainda a rede neuronal (observador) descrita por (4.2-13).

$$x_n(k+1) = A x_n(k) + B u(k) + D \sigma(x_n(k)) + L (y_s(k) - C x_n(k))$$

$$y_n(k) = C x_n(k) \quad (4.2-13)$$

Admitindo a condição de observabilidade para o par  $(A, C)$  determine-se um ganho  $L \in \mathfrak{R}^{n,n}$ , projectado apenas para a parte linear, de tal forma que posicione os valores próprios da matriz  $A_o$  (4.2-14), em locais desejados.

$$A_o = A - LC \quad (4.2-14)$$

O erro de estimação do sistema total (linear + não linear) converge se se verificar a relação

$$\Delta D \gamma \Delta \leq -\Delta A_o \Delta + \sqrt{\Delta A_o \Delta^2 + \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}} \quad (4.2-15)$$

em que  $P$  e  $Q$  são matrizes simétricas positivas definidas satisfazendo a equação discreta de Lyapunov.

$$A_o^T P A_o - P = -Q \quad (4.2-16)$$

#### Demonstração

Defina-se uma função de Lyapunov  $V(k)$  como em (4.2-17).

$$V(\varepsilon(k)) = \varepsilon(k)^T P \varepsilon(k) \quad (4.2-17)$$

tal que  $P$  seja positiva definida e seja o erro de estimação, definido por (4.2-18).

$$\varepsilon(k) = x_s(k) - x_n(k) \quad (4.2-18)$$

A partir das equações que definem a dinâmica do sistema (4.2-2) e o observador proposto (4.2-13) e assumindo  $A = A^*$ , pode-se escrever (4.2-19).

$$\varepsilon(k+1) = A_o \varepsilon(k) + \phi(x_s(k), x_n(k)) \quad (4.2-19)$$

Em que  $A_o = (A - LC)$  e  $\phi(x_s(k), x_n(k))$  é definido por:

$$\phi(x_s(k), x_n(k)) = D \sigma(x_s(k)) - D \sigma(x_n(k)) = D (\sigma(x_s(k)) - \sigma(x_n(k))) \quad (4.2-20)$$

Seguindo a teoria de Lyapunov (Apêndice A.1), pretende-se determinar em que condições a variação da função de Lyapunov,  $\Delta V(\varepsilon(k+1)) = V(\varepsilon(k+1)) - V(\varepsilon(k))$ , é negativa. Por definição de (4.2-17) tem-se:

$$\Delta V(\varepsilon(k+1)) = \varepsilon(k+1)^T P \varepsilon(k+1) - \varepsilon(k)^T P \varepsilon(k) \quad (4.2-21)$$

Simplificando a notação,  $\Delta V = \Delta V(\varepsilon(k+1))$ ,  $\varepsilon = \varepsilon(k)$  e  $\phi = \phi(x_s(k), x_n(k))$  vem, usando (4.2-19),

$$\Delta V = (A_o \varepsilon + \phi)^T P (A_o \varepsilon + \phi) - \varepsilon^T P \varepsilon$$

$$\Delta V = (A_o \varepsilon)^T P (A_o \varepsilon) + (A_o \varepsilon)^T P \phi + \phi^T P (A_o \varepsilon) + \phi^T P \phi - \varepsilon^T P \varepsilon \quad (4.2-22)$$

Porque os termos presentes na equação anterior são escalares, utilizando (4.2-16) pode-se escrever (4.2-23).

$$\Delta V = \varepsilon^T (A_o^T P A_o - P) \varepsilon + 2 \varepsilon^T A_o^T P \phi + \phi^T P \phi$$

$$\Delta V = -\varepsilon^T Q \varepsilon + 2 \varepsilon^T A_o^T P \phi + \phi^T P \phi \quad (4.2-23)$$

A partir da definição do erro, (4.2-18), e usando (4.2-20) tem-se

$$\phi(x) = D (\sigma(x_n + \varepsilon) - \sigma(x_n))$$

Usando as propriedades das normas de matrizes (Apêndice A.2) e atendendo a que a função tangente hiperbólica ( $\sigma$ ) é Lipschitz:

$$\|\phi\| \leq \|D\| (\|\sigma(x_n + \varepsilon) - \sigma(x_n)\|) \leq \gamma \|D\| \|x_n + \varepsilon - x_n\|$$

$$\|\phi\| \leq \gamma \|D\| \|\varepsilon\| \quad (4.2-24)$$

De (4.2-23) resulta

$$\begin{aligned} \Delta V &\leq -\lambda_{\min}(Q) \|\varepsilon\|^2 + 2 \gamma \lambda_{\max}(P) \|A_o^T\| \|D\| \|\varepsilon\|^2 \\ &\quad + \gamma^2 \lambda_{\max}(P) \|D\|^2 \|\varepsilon\|^2 \end{aligned}$$

$$\Delta V \leq \left[ -\lambda_{\min}(Q) + 2 \gamma \lambda_{\max}(P) \|A_o^T\| \|D\| + \gamma^2 \lambda_{\max}(P) \|D\|^2 \right] \|\varepsilon\|^2 \quad (4.2-25)$$

Porque se quer  $\Delta V < 0$ , então deve-se garantir

$$-\lambda_{\min}(Q) + 2 \gamma \lambda_{\max}(P) \|A_o^T\| \|D\| + \gamma^2 \lambda_{\max}(P) \|D\|^2 < 0 \quad (4.2-26)$$

Atendendo a que os termos são escalares, pode-se escrever, considerando  $x = \gamma \|D\|$ ,  $p = \lambda_{\max}(P)$ ,  $q = \lambda_{\min}(Q)$ ,  $a = \|A_o^T\| \|D\|$ .

$$p x^2 + 2ap x - q \leq 0 \quad (4.2-27)$$

A solução da equação de segundo grau

$$p x^2 + 2ap x - q = 0 \quad (4.2-28)$$

tem duas raízes:

$$x_{1,2} = \frac{-2ap \pm \sqrt{4a^2 p^2 + 4qp}}{2p}$$

$$x_{1,2} = -a \pm \sqrt{a^2 + \frac{q}{p}}$$

Como  $a, p, q > 0$  pode escrever-se, também com  $\delta > 0$ .

$$x_{1,2} = -a \pm (a + \delta)$$

Assim existirão duas raízes, uma positiva e outra negativa.

$$x_1 = -2a - \delta$$

$$x_2 = +\delta$$

verificando-se ainda

$$x_1 < x < x_2$$

Porque  $x$  é definido por uma norma, ( $x = \angle \gamma D \angle$ ), portanto um valor positivo, e porque o valor absoluto de  $x_1$  é superior a  $x_2$  conclui-se que o valor que satisfaz (4.2-27) é:

$$0 \leq x \leq x_2 = -a + \sqrt{a^2 + \frac{q}{p}}$$

Substituindo a notação utilizada demonstra-se o teorema enunciado.

$$\angle D \gamma \angle \leq -\angle A_o \angle + \sqrt{\angle A_o \angle^2 + \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}} \quad (4.2-29)$$

■

Tal como referido para o teorema de Thau, este resultado não permite o projecto, propriamente dito, do observador para o modelo neuronal, apenas estabelece uma condição necessária e suficiente para a sua convergência.

## Observadores de Modo Deslizante

Propõe-se agora a aplicação da teoria de observação robusta ao modelo neuronal considerando para o efeito a não linearidade como uma incerteza. Utiliza-se o método proposto por Mahmoud e Bahnasawi (1994) onde é introduzida uma técnica robusta de observação para uma classe particular de sistemas discretos, cuja não linearidade é limitada (como é o caso da não linearidade da estrutura neuronal considerada neste estudo). O método resultante não é senão o correspondente discreto da técnica no domínio contínuo proposta por Walcott e Zak (1987b).

Comece-se por enunciar o teorema de Mahamod e Bahnasawi, (1994).

### Teorema (Mahamod e Bahnasawi, 1994)

Considere-se um sistema não linear da forma (4.2-30).

$$x(k+1) = A x(k) + B u(k) + \zeta(x(k), u(k), k) \quad (4.2-30a)$$

$$y(k) = C x(k) \quad (4.2-30b)$$

Seja o observador definido por (4.2-31).

$$\begin{aligned} \hat{x}(k+1) = & A \hat{x}(k) + B u(k) + \zeta(\hat{x}(k), u(k), k) \\ & + L(y(k) - C \hat{x}(k)) + \nu(\hat{x}(k), u(k)) \end{aligned} \quad (4.2-31)$$

O termo  $L(y(k) - C\hat{x}(k)) = L e(k)$ , corresponde a um observador típico de Luenberger, em que o ganho  $L$  é determinado de forma a que a matriz  $A_o = (A - LC)$  tenha valores próprios em locais desejados.

O termo  $\nu(\hat{x}(k), u(k))$  corresponde ao termo de modo deslizante e é determinado por

$$\nu(\hat{x}(k), u(k)) = \begin{cases} \frac{\mu P^{-1} C^T C e(k)}{\angle C e(k) \angle} & \text{se } \angle C e(k) \angle > \eta \\ \frac{\mu P^{-1} C^T C e(k)}{\eta} & \text{se } \angle C e(k) \angle \leq \eta \end{cases} \quad (4.2-32)$$

em que  $\mu$  e  $\eta$  são constantes reais positivas. Nestas condições o erro de observação, definido por (4.2-33),

$$\varepsilon(k+1) = A_o \varepsilon(k) + \nu(\hat{x}(k), u(k)) - P^{-1} C^T h(x(k), u(k), k) \quad (4.2-33)$$

é limitado, sendo  $P$  uma matriz simétrica positiva definida obtida de forma a satisfazer a equação discreta de Lyapunov (com  $Q$  também simétrica positiva definida).

$$A_o^T P A_o - P = -Q \quad (4.2-34)$$

**Condições de Verificação**

Considera-se ainda neste teorema, como princípio fundamental, o sistema ser observável e serem garantidas as seguintes condições, conhecidas por condições de verificação

i) existe uma função  $h(x(k), u(k), k) \in \mathfrak{R}^{n_y}$ , que verifica-se a relação (4.2-35):

$$\zeta(x(k), u(k), k) = P^{-1} C^T h(x(k), u(k), k) \quad (4.2-35)$$

ii) é possível determinar uma constante,  $\mu > 0$ , que verifica

$$\|h(x(k), u(k), k)\| \leq \mu \quad (4.2-36)$$

Aplicada ao caso particular em estudo verifica-se que a equação (4.2-35) é relativamente fácil de satisfazer, mas não de resolver. Para a rede particular considerada verifica-se

$$\zeta(x_s(k)) = D \sigma(x_s(k)) \quad (4.2-37)$$

Pretende-se determinar uma constante  $\mu$ , de tal forma que

$$D \sigma(x_s(k)) = P^{-1} C^T h(x_s(k)) \quad (4.2-38)$$

e que simultaneamente

$$\|h(x_s(k))\| \leq \mu \quad (4.2-39)$$

Na prática não é necessário determinar o valor concreto de  $h(\cdot)$ , ou seja, para que o observador seja implementado é necessário apenas obter um valor  $\mu$  que verifique (4.2-36), o que no caso particular em estudo é relativamente fácil. De facto, porque se conhecem os valores de  $D$ ,  $P$  e  $C$  e a não linearidade em questão ( $\sigma$ ) é limitada ( $|\sigma(x)| < 1$ ), é possível encontrar (empiricamente) um valor para  $\mu$ .

Note-se que a formulação matemática do problema resulta na necessidade de resolução de um sistema de equações não lineares, (4.2-40a), sujeito a restrições, (4.2-40b).

$$\left\{ \begin{array}{l} A_o = A - LC \\ D \sigma(x_s) = P^{-1} C^T h(x_s) \\ A_o^T P A_o - P = -Q \end{array} \right. \quad (4.2-40a)$$

$$(4.2-40b)$$

Na equação (4.2-40) como valores conhecidos tem-se  $A$ ,  $C$ ,  $D$  e  $Q$  e pretende-se determinar os valores da função  $h(x_s)$  e de  $P$  o que, como se constata, não é um problema trivial. Uma vez determinado o valor de  $h(x_s)$  e de  $P$  é possível determinar  $\mu$  e  $L$  que permitem por sua vez implementar o observador definido em (4.2-31).

### 4.2.3 Estimação de Parâmetros

Uma vez estabelecidas formas de estimar os estados aborda-se agora o problema de determinar os parâmetros para o modelo neuronal. Tal como num filtro de Kalman dual, assume-se que se conhecem os estados  $x_s(k)$ . Como objectivo final, estabelece-se a adaptação dos parâmetros de forma a assegurar a convergência do erro de estimação (para os estados), definido por (4.2-41).

$$\varepsilon(k) = x_s(k) - x_n(k) \quad (4.2-41)$$

De forma a obter os parâmetros  $B(k)$  e  $D(k)$  considere-se, como um caso particular de (4.2-6), o seguinte modelo neuronal.

$$\begin{aligned} x_n(k+1) &= A x_n(k) + B(k) u(k) + D(k) \sigma(x_s(k)) \\ y_n(k) &= C x_n(k) \end{aligned} \quad (4.2-42)$$

O erro de estimação definido pela diferença entre o modelo a identificar (4.2-2) e o definido em (4.2-42) pode escrever-se como (4.2-43).

$$\varepsilon(k+1) = A^* x_s(k) - A x_n(k) + (D^* - D(k)) \sigma(x_s(k)) + (B^* - B(k)) u(k) \quad (4.2-43)$$

Definindo

$$\tilde{D} = D^* - D(k) \quad (4.2-44)$$

$$\tilde{B} = B^* - B(k) \quad (4.2-45)$$

e considerando  $A = A^*$  pode-se escrever:

$$\varepsilon(k+1) = A \varepsilon(k) + \tilde{D} \sigma(x_s(k)) + \tilde{B} u(k) \quad (4.2-46)$$

Ou ainda,

$$\varepsilon(k+1) = A \varepsilon(k) + \varphi(k) \vec{W}(k) \quad (4.2-47)$$

$\varphi(k) \in \mathfrak{R}^{n, n_w}$  é uma matriz de informação, função de  $\sigma(x_s(k))$  e de  $u(k)$ .  $\vec{W}(k) \in \mathfrak{R}^{n_w}$  representa um vector,

$$\vec{W}(k) = [ \vec{D}(k) \quad \vec{B}(k) ] \quad (4.2-48)$$

constituído pela variação dos pesos das matrizes a determinar, em que a constante  $n_w$  define o número de parâmetros. É possível escrever

$$\begin{aligned} \Delta \tilde{W}(k) &= \tilde{W}(k) - \tilde{W}(k-1) \\ \Delta \tilde{W}(k) &= (W^* - W(k)) - (W^* - W(k-1)) = -\Delta W(k) \end{aligned} \quad (4.2-49)$$

Assim, entre dois instantes consecutivos, pode-se relacionar  $\Delta \vec{W}(k)$  com  $\vec{\Delta W}(k)$ , um vector constituído pelos incrementos dos parâmetros das matrizes  $B(k)$  e  $D(k)$

$$\vec{\Delta W}(k) = [ \vec{D}(k) - \vec{D}(k-1) \quad \vdots \quad \vec{B}(k) - \vec{B}(k-1) ] \quad (4.2-50)$$

De (4.2-27) resulta assim (4.2-51).

$$\varepsilon(k+1) = A \varepsilon(k) - \varphi(k) \vec{\Delta W}(k) \quad (4.2-51)$$

## Teorema 2: Adaptação de Parâmetros na Metodologia RSRN

Seja um sistema genérico, que se assume poder ser descrito por uma rede neuronal com a seguinte estrutura

$$\begin{aligned} x_s(k+1) &= A^* x_s(k) + B^* u(k) + D^* \sigma(x_s(k)) \\ y_s(k) &= C x_s(k) \end{aligned}$$

Seja o modelo neuronal

$$\begin{aligned} x_n(k+1) &= A^* x_n(k) + B(k) u(k) + D(k) \sigma(x_s(k)) \\ y_n(k) &= C^* x_n(k) \end{aligned}$$

cujas matrizes  $B(k)$  e  $D(k)$  se pretendem determinar. A regra de adaptação

$$\vec{\Delta W}(k) = M(k)^{-1} \varphi(k)^T P A \varepsilon(k) \quad (4.2-52)$$

garante a estabilidade e convergência do erro  $\varepsilon(k) = x_s(k) - x_n(k)$ .

A matriz  $M(k) \in \mathfrak{R}^{n_w, n_w}$  é definida por (4.2-53) e  $P \in \mathfrak{R}^{n, n}$  é uma matriz que resulta da solução da equação discreta de Lyapunov (4.2-54).

$$M(k) = \left[ I + \frac{1}{2} \varphi(k)^T P \varphi(k) \right] \quad (4.2-53)$$

$$A^T P A - P = -Q \quad (4.2-54)$$

$Q \in \mathfrak{R}^{n,n}$  é uma matriz definida positiva, a especificar e  $I$  é uma matriz identidade de dimensões adequadas.

#### Notas

- i) uma vez que  $A$  é uma matriz de valores próprios no interior do círculo unitário a solução ( $P$ ) para a equação (4.2-54) além de existir é única (Goodwin e Sin, 1984);
- ii)  $P$  e  $M$  são matrizes simétricas, logo  $P = P^T$  e  $M = M^T$ ;
- iii) a matriz  $M$  é uma matriz quadrada de dimensão igual ao número de parâmetros a determinar ( $n_w, n_w$ ). A sua inversão, a efectuar em cada instante, apesar de existir, tem de ser considerada uma desvantagem do algoritmo.

Tendo por base o teorema de Lyapunov (Apêndice A.1.1), pretende-se encontrar uma função de Lyapunov definida positiva,  $V(\varepsilon(k+1)) > 0$ , tal que a variação dessa função seja negativa.

$$\Delta V(\varepsilon(k+1)) = V(\varepsilon(k+1)) - V(\varepsilon(k)) < 0 \quad (4.2-55)$$

Adicionalmente, de forma a provar a estabilidade assintótica, usa-se o teorema de La Salle, apêndice (A.1.2).

#### Demonstração:

Considere-se uma função de Lyapunov positiva definida, (4.2-56).

$$V(\varepsilon(k)) = \varepsilon^T(k) P \varepsilon(k) \quad (4.2-56)$$

Usando as equações (4.2-51) e (4.2-53) resulta (4.2-57).

$$\varepsilon(k+1) = A \varepsilon(k) - \varphi(k) M(k)^{-1} \varphi(k)^T P A \varepsilon(k) \quad (4.2-57)$$

A variação da função de Lyapunov, assumindo por simplicidade a seguinte notação:  $\Delta V = \Delta V(\varepsilon(k+1))$ ,  $\varepsilon = \varepsilon(k)$ ,  $\varphi = \varphi(k)$  e  $M = M(k)$ , é dada por (4.2-58).

$$\Delta V = (A\varepsilon - \varphi M^{-1} \varphi^T P A \varepsilon)^T P (A\varepsilon - \varphi M^{-1} \varphi^T P A \varepsilon) - \varepsilon^T P \varepsilon \quad (4.2-58)$$

Após algumas manipulações obtém-se (4.2-59).

$$\begin{aligned} \Delta V &= \left( (A\varepsilon)^T - (\varphi M^{-1} \varphi^T P A \varepsilon)^T \right) P \left( (A\varepsilon) - (\varphi M^{-1} \varphi^T P A \varepsilon) \right) - \varepsilon^T P \varepsilon \\ \Delta V &= (A\varepsilon)^T P (A\varepsilon) \\ &\quad - (A\varepsilon)^T P (\varphi M^{-1} \varphi^T P A \varepsilon) \\ &\quad - (\varphi M^{-1} \varphi^T P A \varepsilon)^T P (A\varepsilon) \\ &\quad + (\varphi M^{-1} \varphi^T P A \varepsilon)^T P (\varphi M^{-1} \varphi^T P A \varepsilon) \\ &\quad - \varepsilon^T P \varepsilon \\ \Delta V &= (A\varepsilon)^T P (A\varepsilon) - \varepsilon^T P \varepsilon & \textcircled{1} \\ &\quad - (A\varepsilon)^T P (\varphi M^{-1} \varphi^T P A \varepsilon) - (\varphi M^{-1} \varphi^T P A \varepsilon)^T P (A\varepsilon) & \textcircled{2} \\ &\quad + (\varphi M^{-1} \varphi^T P A \varepsilon)^T P (\varphi M^{-1} \varphi^T P A \varepsilon) & \textcircled{3} \end{aligned} \quad (4.2-59)$$

A primeira parte da expressão (4.2-59),  $\textcircled{1}$ , pode ser escrita como (4.2-60)

$$\textcircled{1} \quad \varepsilon^T A^T P A \varepsilon - \varepsilon^T P \varepsilon = \varepsilon^T (A^T P A - P) \varepsilon \quad (4.2-60)$$

A segunda parte,  $\textcircled{2}$ , porque se trata de escalares, resulta em (4.2-61)

$$\textcircled{2} \quad -2 (A\varepsilon)^T P (\varphi M^{-1} \varphi^T P A \varepsilon) \quad (4.2-61)$$

Finalmente para a terceira parte,  $\textcircled{3}$ , tem-se (4.2-62).

$$\textcircled{3} \quad (A\varepsilon)^T P^T \varphi M^{-T} \varphi^T P (\varphi M^{-1} \varphi^T P A \varepsilon) \quad (4.2-62)$$

Porque  $P = P^T$  e  $M = M^T$  tem-se ainda

$$\textcircled{3} \quad (A\varepsilon)^T P \varphi M^{-1} \varphi^T P \varphi M^{-1} \varphi^T P A \varepsilon \quad (4.2-63)$$

Atendendo a que o termo  $\varphi^T P \varphi$  pode ser escrito como

$$\varphi^T P \varphi = 2 \left[ I + \frac{1}{2} \varphi^T P \varphi \right] - 2 I \quad (4.2-64)$$

e atendendo à definição de  $M$  em (4.2-53), tem-se (4.2-65).

$$\varphi^T P \varphi = 2 M - 2 I \quad (4.2-65)$$

De (4.2-63) tem-se,

$$\textcircled{3} \quad (A\varepsilon)^T P \varphi M^{-1} (2M - 2I) M^{-1} \varphi^T PA\varepsilon \quad (4.2-66)$$

e após algumas manipulações

$$\begin{aligned} \textcircled{3} \quad & (2(A\varepsilon)^T P \varphi - 2(A\varepsilon)^T P \varphi M^{-1}) M^{-1} \varphi^T PA\varepsilon \\ \textcircled{3} \quad & 2(A\varepsilon)^T P \varphi M^{-1} \varphi^T PA\varepsilon - 2(A\varepsilon)^T P \varphi M^{-2} \varphi^T PA\varepsilon \end{aligned} \quad (4.2-67)$$

Concluindo, de cada uma das partes  $\textcircled{1}$ :(4.2-60),  $\textcircled{2}$ :(4.2-61) e  $\textcircled{3}$ :(4.2-67) resulta (4.2-68)

$$\begin{aligned} \Delta V = & \quad \varepsilon^T (A^T PA - P) \varepsilon \\ & - 2(A\varepsilon)^T P \varphi M^{-1} \varphi^T PA\varepsilon + 2(A\varepsilon)^T P \varphi M^{-1} \varphi^T PA\varepsilon \\ & - 2(A\varepsilon)^T P \varphi M^{-2} \varphi^T PA\varepsilon \end{aligned} \quad (4.2-68)$$

A equação anterior pode ser simplificada resultando em (4.2-69).

$$\Delta V = \quad \varepsilon^T (A^T PA - P) \varepsilon - 2(A\varepsilon)^T P \varphi M^{-2} \varphi^T PA\varepsilon$$

ou

$$\Delta V = \quad -\varepsilon^T Q \varepsilon - 2(\varphi^T PA\varepsilon)^T M^{-2} (\varphi^T PA\varepsilon) \quad (4.2-69)$$

O primeiro termo da expressão anterior, usando a definição (4.2-54), é negativo. Assim, para que a variação seja negativa ( $\Delta V < 0$ ) o segundo termo terá também de o ser, o que na verdade se verifica. De facto, atendendo a que se pode escrever,

$$-2(\varphi^T PA\varepsilon)^T M^{-2} (\varphi^T PA\varepsilon) = -2X^T M^{-2} X \quad (4.2-70)$$

em que  $X = (\varphi^T PA\varepsilon)$ , e uma vez que (4.2-70) define uma forma quadrática, ( $M$  é simétrica) assegura-se que este último termo é negativo.

$$-2X^T M^{-2} X < 0 \quad (4.2-71)$$

Concluindo,

$$\Delta V = -\varepsilon^T Q \varepsilon - 2X^T M^{-2} X \quad (4.2-72)$$

ou seja,

$$\Delta V < 0 \quad (4.2-73)$$

Assegura-se assim a estabilidade no sentido de Lyapunov.

Note-se que, uma vez que  $V(\boldsymbol{\varepsilon}(k)) > 0$ , então se  $\Delta V(\boldsymbol{\varepsilon}(k)) < 0$ , garante-se uma estabilidade assintótica. No entanto, apenas se pode garantir a existência de condições de estabilidade caso  $\Delta V(\boldsymbol{\varepsilon}(k)) \leq 0$ .

Assim, de forma a mostrar que o algoritmo é também convergente, o mesmo é dizer, assintoticamente estável, recorre-se ao princípio de La Salle (Apêndice A.1.2), sendo suficiente demonstrar que o maior subconjunto no qual  $\Delta V(\boldsymbol{\varepsilon}(k)) = 0$  é invariante é o subconjunto  $\{0\}$ .

De (4.2-69) resulta que se  $\Delta V$  for nulo então cada um dos termos terá também de ser nulo, ou seja,

$$\begin{cases} \boldsymbol{\varepsilon}^T (A^T P A - P) \boldsymbol{\varepsilon} = 0 \\ \boldsymbol{\varphi}^T P A \boldsymbol{\varepsilon} = 0 \end{cases} \quad (4.2-74)$$

Segundo o princípio de Invariância de La Salle pretende-se garantir que as soluções para as quais  $\Delta V(\boldsymbol{\varepsilon}(k)) = 0$  devem satisfazer simultaneamente, para qualquer valor de  $k$ , as seguintes equações

$$\begin{cases} \boldsymbol{\varepsilon}(k)^T (A^T P A - P) \boldsymbol{\varepsilon}(k) = 0 \\ \boldsymbol{\varphi}(k)^T P A \boldsymbol{\varepsilon}(k) = 0 \\ \overrightarrow{\Delta W}(\boldsymbol{\varepsilon}(k)) = 0 \end{cases} \quad (4.2-75)$$

Tendo em atenção a o sistema de equações anterior e a partir de (4.2-47), tem-se  $\boldsymbol{\varepsilon}(k+1) = A \boldsymbol{\varepsilon}(k)$ , conclui-se portanto que para  $\Delta V(\boldsymbol{\varepsilon}(k)) = 0$  necessariamente  $\boldsymbol{\varepsilon}(k) = 0$ .

Assim, e uma vez que também é verificada a equação (4.2-76)

$$V(\boldsymbol{\varepsilon}(k)) \rightarrow \infty \text{ se } |\boldsymbol{\varepsilon}(k)| \rightarrow \infty \quad (4.2-76)$$

conclui-se finalmente usando a condição *iii*) do teorema de Lyapunov (Apêndice A.1) que a origem é globalmente assintoticamente estável.

■

## 4.2.4 Algumas Considerações

Nesta secção abordou-se o problema de estimação de parâmetros de uma rede recorrente a partir de técnicas de observação, considerando uma estrutura semelhante a um filtro de Kalman dual.

Conforme referido, não se pretende em qualquer caso determinar correctamente os estados do processo real  $e$ , por isso, talvez a metodologia proposta não deva ser considerada propriamente um observador, mas sim uma descrição do sistema no espaço de estados capaz de aproximar a dinâmica entrada-saída do processo a modelizar. De facto, o que se pretende é assegurar a estabilidade e convergência do erro de saída do modelo  $e_y(k)$ , diferença entre a saída do processo e a do modelo neuronal. Uma vez que se assegura a estabilidade para o estado  $\epsilon(k)$  (teorema 2) e porque se considera a matriz  $C$  fixa, a estabilidade do erro da saída  $e_y(k) = C\epsilon(k)$  é imediata.

O método desenvolvido é aplicável apenas no caso da matriz  $A$  ser Hurwitz, isto é, os seus valores próprios se localizem no interior do círculo de raio unitário. Por outras palavras, entendendo o modelo neuronal como constituído por uma parte linear e outra não linear, terá aplicação apenas caso a parte linear seja estável.

Porque se assume esta matriz como fixa, em termos práticos, tal imposição restringe a determinação inicial dos parâmetros da rede.

# 4.3

## Regulação da Saída na Metodologia RSRN

*Nesta secção aborda-se o problema da regulação da saída aplicado de uma forma genérica a sistemas discretos e em particular ao modelo neuronal recorrente proposto.*

*Formula-se e caracteriza-se a possibilidade de obtenção de uma solução para as equações de regulação através de um processo iterativo e introduz-se um teorema relativo à convergência do algoritmo.*

### 4.3.1 Introdução

Por regulação da saída ( $OR^1$ ), aplicada a um determinado sistema, entende-se o problema de determinar uma lei de controlo tal que o sistema global em malha fechada seja localmente estável e, simultaneamente, a saída convirja para um valor desejado, gerado a partir de um sistema externo, sendo por esta razão designado por sistema exógeno.

Regra geral, os trabalhos que incluem resultados de estabilidade usam a teoria de Lyapunov. Em alternativa, a teoria de regulação usa a teoria  $CMT^2$  para estabelecer resultados de estabilidade para o problema de controlo. Além de independente da teoria de Lyapunov, o que elimina a necessidade de definição de uma função de Lyapunov (muitas vezes uma tarefa complexa), a teoria  $CMT$  tem a vantagem adicional de, em certas condições, reduzir o problema original de controlo a um outro de dimensões reduzidas.

Tal como na linearização retroactiva o problema de regulação da saída é basicamente um problema de realimentação de variáveis de estado, podendo ser encarado de duas formas. Na primeira, assume-se que os estados são acessíveis e a lei de controlo é dada por

---

<sup>1</sup>  $OR$ - Output Regulation

<sup>2</sup>  $CMT$ - Center Manifold Theory

(4.3-1), em que  $x(k)$  e  $\rho(k)$  definem, respectivamente, os estados do sistema e os estados do sistema exógeno.

$$u(k) = \gamma(x(k), \rho(k)) \quad (4.3-1)$$

Na segunda, mais realista, uma vez que na prática raramente os estados se encontram acessíveis, assume-se que se dispõe apenas de informação da saída. Neste caso usa-se um controlador dinâmico, (4.3-2), em oposição ao anterior que é denominado estático, uma vez que se usa o estado  $\hat{x}(k)$  de um sistema dinâmico auxiliar (observador).

$$u(k) = \gamma(\hat{x}(k), \rho(k)) \quad (4.3-2a)$$

$$\hat{x}(k+1) = \vartheta(\hat{x}(k), \hat{y}(k), y(k), u(k)) \quad (4.3-2b)$$

O uso simultâneo de um controlador para determinar a acção de controlo e de um observador para estimar os estados é outra das grandes motivações para o uso da teoria da regulação da saída. De facto, a teoria da regulação da saída incorpora de uma forma inerente um princípio de separação, uma vez que a acção de controlo é determinada tendo em conta a estabilidade do sistema total de controlo. Para esta técnica foram estabelecidos resultados que garantem a aplicação do princípio da separação (controlador+observador) para o caso de sistemas não lineares, tanto no domínio contínuo como no domínio discreto, respectivamente por Isidori e Byrnes (1990) e Lin e Byrnes (1994).

Se, por um lado, a teoria de regulação tem aspectos atractivos, por outro tem também as suas limitações. A resolução do problema de regulação é conduzida à resolução de um sistema não linear de equações diferenciais no caso contínuo, ou de diferenças, no caso discreto. Na verdade, esta é a grande dificuldade do problema de regulação já que não é fácil solucionar o sistema de equações sendo mesmo, na maioria dos casos, impossível determinar uma solução analítica.

Na metodologia *RSRN* proposta, combina-se a teoria de regulação com um observador neuronal. Assim, considera-se que a lei de regulação será implementada com um controlador dinâmico, sendo a equação (4.3-2b) substituída por uma rede neuronal recorrente. O tipo de não linearidade em questão (tangente hiperbólica), permite que a resolução do sistema de equações de diferenças seja solucionada (aproximadamente) de uma forma eficaz, sendo esta a ideia principal que se explora nesta metodologia.

## 4.3.2 Teoria de Regulação da Saída

O problema da regulação da saída começou a ser investigado por Davison (1976), Francis e Wonham (1976) e Francis (1977), cujos trabalhos diziam respeito a sistemas contínuos lineares e invariantes. Concretamente, Francis (1977) mostrou que a solução do problema podia ser conduzida à resolução de um sistema linear de duas equações matriciais, conhecido como *equação de Francis*. Hepburn e Wonham (1984) mostraram

que era condição necessária para a resolução do problema que as trajectórias a seguir pudessem ser obtidas a partir de um sistema externo (exógeno) e que este fosse incorporado no controlador como um modelo interno.

Para o caso de sistemas contínuos não lineares destacam-se os trabalhos de Jie e Rugh (1988) e Huang e Rugh (1990), que solucionaram o problema de sinais exógenos constantes (referência fixa) e principalmente o trabalho de Isidori e Byrnes (1990), para o caso geral de sinais exógenos variantes. Isidori e Byrnes mostraram que o problema da regulação de um sistema não linear tem solução se e só se existir solução para um sistema não linear de equações diferenciais, denominadas equações de *Francis-Isidori-Byrnes*, assumindo adicionalmente certas propriedades para a aproximação linear do sistema e do sistema exógeno. Apresentaram ainda uma estrutura possível para o controlador, facilmente implementável, análoga à que resulta da especificação de um algoritmo de atribuição de valores próprios.

Basicamente, a ideia de Isidori e Byrnes consistia em determinar uma lei de controlo que estabilize o sistema num determinado ponto de equilíbrio, caso a referência seja nula. Depois, usando conceitos e propriedades da teoria de CMT, (Carr, 1981), assegurava que em malha fechada o erro tende assintoticamente para zero, para um sinal exógeno não nulo. Para o caso particular de sistemas descritos na forma normal, com um grau relativo bem definido, mostraram ainda que a possibilidade de resolução do problema de regulação se relacionava com a localização das raízes da equação da dinâmica *ZD-zero dynamics*, concretamente, que esta seja hiperbólica, o mesmo é dizer, que nenhuma das raízes do Jacobiano que resulta da sua linearização tenha raízes no eixo real.

O problema de existência de solução para as equações de regulação é estudado em Huang e Rugh (1992a) e Huang e Rugh (1992b), como uma extensão do trabalho de Isidori e Byrnes (1990), onde o mesmo problema é abordado para uma classe particular de sistemas *affine*. Huang (1995) estabelece condições segundo as quais o problema de regulação tem solução mesmo para sistemas cujo *ZD* seja não hiperbólica, relaxando a condição de estabilidade assintótica.

Em relação ao problema da regulação aplicado a sistemas discretos, a primeira tentativa de o solucionar foi empreendida por Castillo-Toledo e Di Gennaro (1991), sendo os seus resultados apenas aplicáveis a sistemas SISO. Para sistemas discretos não lineares MIMO, Castillo-Toledo *et al.* (1993a), usando o algoritmo *Zero Output Constrained Dynamics Algorithm*, introduzido por Monaco e Normand-Cyrot (1987), mostraram que a solução do problema pode ser reduzida à solução de um sistema de equações de diferenças não lineares, considerando adicionalmente certas propriedades para a aproximação linear do sistema e do sistema exógeno. Os seus resultados são em tudo análogos aos de Isidori e Byrnes (1990), mostrando de forma semelhante que a solução do problema está directamente relacionada com a localização das raízes da primeira aproximação da *ZD* do sistema, no caso discreto sobre a circunferência de raio unitário.

Posteriormente Castillo-Toledo *et al.* (1993b), estenderam os seus resultados e estabelecem condições para a regulação de sistemas com um número de entradas superior ao número de saídas. Sureshbabu e Rugh introduzem (1995) e (1997) uma técnica de regulação que utiliza informação adicional, derivadas ou valores passados do sistema exógeno respectivamente, para os casos contínuo e discreto, permitindo além disso considerar sistemas exógenos mais genéricos. Saberi e Stoorvogel (2000), também para os casos contínuo e discreto, propõem uma formulação genérica para as técnicas de regulação. A sua formulação permite incluir as ideias de Sureshbabu e Rugh (derivadas ou regressão do sistema exógeno, respectivamente, para os casos contínuo e discreto) e consideram também a hipótese de sistemas exógenos não autónomos, a serem influenciados por um sinal de referência não gerado pelo sistema exógeno. No entanto, os seus resultados são apenas válidos para sistemas lineares.

Relativamente à resolução do sistemas de equações Castillo-Toledo *et al.* (Castillo-Toledo, 1993a) propuseram e derivaram condições para a existência de uma solução aproximada baseada numa expansão polinomial. De forma análoga, Huang e Rugh (1992a) e (1992b) apresentaram um método para o caso contínuo baseado numa expansão em séries de Taylor. Recentemente Chu e Huang (1999) propuseram também para o caso contínuo o uso de uma rede recorrente particular (*Cellular Neural Network*) para solucionar o sistema de equações diferenciais e mostraram que através de uma escolha correcta dos seus pesos a rede recorrente pode solucionar as equações de regulação.

## Sistemas Discretos Não Lineares

### Descrição do Problema

Considere-se um sistema não linear com  $n_u$  entradas e  $n_y$  saídas, descrito no espaço de estados pela seguintes equação de diferenças,

$$x(k+1) = f(x(k), u(k)) \quad (4.3-3a)$$

$$y(k) = h(x(k)) \quad (4.3-3b)$$

sendo  $f: \mathfrak{X}^{n+n_u} \rightarrow \mathfrak{X}^n$  e  $h: \mathfrak{X}^n \rightarrow \mathfrak{X}^{n_y}$  funções não lineares;  $u(k) \in \mathfrak{X}^{n_u}$ ,  $y(k) \in \mathfrak{X}^{n_y}$  e  $x(k) \in \mathfrak{X}^n$  definem respectivamente, a entrada, a saída e o estado, no instante discreto  $k$ .

Considere-se ainda um sistema adicional (exógeno), descrito por um modelo de estados,  $\rho(k) \in \mathfrak{X}^q$ , em que  $s: \mathfrak{X}^q \rightarrow \mathfrak{X}^q$  que define as perturbações e/ou saída desejada.

$$\rho(k+1) = s(\rho(k)) \quad (4.3-4)$$

A regulação da saída tem interesse especial no caso do sinal exógeno a perseguir ser 'persistente', de que são exemplo os sinais periódicos limitados. Para assegurar que o sinal exógeno é limitado assume-se que é estável no sentido de Lyapunov. De forma a garantir a persistência, exclui-se a hipótese do sinal convergir para zero, assumindo que é estável

no sentido de Poisson (Isidori, 1996). A combinação das duas propriedades anteriores é usualmente referida por estabilidade neutra e pode ser garantida caso as raízes da primeira aproximação do sistema exógeno se localizem no eixo imaginário ou círculo de raio unitário, respectivamente para os casos contínuos e discretos.

Seja a referência ou saída desejada  $y_d(k) \in \mathfrak{R}^{n_y}$  definida a partir do sistema exógeno por (4.3-5), e  $r: \mathfrak{R}^q \rightarrow \mathfrak{R}^{n_y}$ .

$$y_d(k) = r(\rho(k)) \quad (4.3-5)$$

Defina-se um sistema estendido, resultante da combinação do sistema (4.3-3a), sistema exógeno (4.3-4) e saída desejada (4.3-5), como se segue:

$$x_p(k+1) = f_p(x_p(k), u(k)) \quad (4.3-6a)$$

$$e_p(k) = e(k) = h_p(x_p(k)) \quad (4.3-6b)$$

em que

$$x_p(k) = \begin{bmatrix} x(k) \\ \rho(k) \end{bmatrix} \quad (4.3-7a)$$

$$f_p(k) = \begin{bmatrix} f(x(k), u(k), \rho(k)) \\ s(\rho(k)) \end{bmatrix} \quad (4.3-7b)$$

$$e_p(k) = r(\rho(k)) - h(x(k)) = y_d(k) - h(x(k)) \quad (4.3-7c)$$

A variável  $e(k) \in \mathfrak{R}^{n_y}$  define o erro de controlo, diferença entre o valor desejado  $y_d(k)$  e o valor real da saída  $y(k) = h(x(k))$ . Assume-se que os mapeamentos  $f(x, u, \rho)$ ,  $s(\rho)$ ,  $h(x)$  e  $r(\rho)$  são definidos por funções contínuas e diferenciáveis, verificando-se  $f(0,0,0) = 0$ ,  $s(0) = 0$ ,  $h(0) = 0$  e  $r(0) = 0$ .

### Objectivo da Regulação

Para o sistema estendido (4.3-6) pretende-se, segundo a definição de regulação da saída, determinar uma lei de controlo do tipo (4.3-1), assumindo o conhecimento dos estados (sistema e sistema exógeno), de forma a garantir que o sistema estendido seja localmente estável e que, simultaneamente, o erro  $e(k)$  seja convergente para zero.

Nesse sentido, a lei de controlo (4.3-1),  $\gamma: \mathfrak{R}^{n+q} \rightarrow \mathfrak{R}^{n_u}$ , deve verificar  $\gamma(0,0) = 0$ , e, além disso, deve ter em conta os seguintes requisitos (S1) e (S2).

**S1**

O equilíbrio, no ponto  $x = 0$ , do sistema

$$x(k+1) = f(x(k), \gamma(x(k), 0), 0) \equiv f_e(x(k), \gamma(x(k), 0)) \quad (4.3-8)$$

é localmente estável de uma forma exponencial, isto é, o sistema linear que resulta da sua linearização, considerando o sistema exógeno desconectado ( $\rho = 0$ ), apresenta um ponto de equilíbrio assintoticamente estável para  $x = 0$ .

**S2**

Existe uma vizinhança da origem  $(x, \rho) = (0, 0)$  tal que, para cada estado inicial  $(x(0), \rho(0))$ , a solução do sistema em malha fechada:

$$x(k+1) = f(x(k), \gamma(x(k), \rho(k)), \rho(k)) \quad (4.3-9a)$$

$$\rho(k+1) = s(\rho(k)) \quad (4.3-9b)$$

satisfaz a condição de erro:

$$\lim_{k \rightarrow \infty} e(k) = [r(\rho(k)) - h(x(k))] = 0 \quad (4.3-10)$$

As exigências anteriores reflectem as propriedades de estabilidade e de regulação. Estabilidade local e exponencial, caso não existam perturbações, caracterizam a condição de estabilidade (S1). Convergência do erro, para estados iniciais próximos da origem, caracterizam a condição de regulação (S2).

**Papel do Center Manifold**

Se o conceito de *ZD* é útil, uma vez que permite estabelecer condições para a existência de uma solução, a teoria de *CMT* permite estabelecer condições que caracterizem uma regulação perfeita. Note-se que ao definir-se o sistema estendido transforma-se na verdade uma equação de diferenças (o sistema) com restrições (o sistema exógeno) numa equação de diferenças sem restrições, de forma a que a teoria *CMT* possa ser aplicada, (Carr, 1981). A teoria de regulação, aplicada a sistemas discretos não lineares é baseada na teoria de *CMT* e em algumas hipóteses para a primeira aproximação do sistema, análogas às obtidas por Isidori para sistemas em tempo contínuo (Isidori e Byrnes, 1990).

*Condições para a primeira aproximação do sistema e sistema exógeno*

**A1:** O sistema exógeno é neutralmente estável, isto é, verifica as seguintes especificações:

- i) o regime de equilíbrio ( $\rho = 0$ ) é estável no sentido de Lyapunov;
- ii) os valores próprios da primeira aproximação (*S*) do sistema exógeno localizam-se sobre a circunferência de raio unitário.

$$S \equiv \left. \frac{\partial s}{\partial \rho} \right|_{\rho=0} \quad (4.3-11)$$

**A2:** Existe uma lei de controlo  $u(k) = \gamma(x(k))$ , com  $(\gamma(0) = 0)$  de forma que em malha fechada, com  $\rho = 0$ , o sistema

$$x(k+1) = f_e(x(k), \gamma(x(k))) \quad (4.3-12)$$

é estável (localmente) de uma forma exponencial. Para que isso aconteça torna-se necessário que o par  $(\underline{A}, \underline{B})$  seja controlável, sendo  $\underline{A} \in \mathfrak{R}^{n,n}$  e  $\underline{B} \in \mathfrak{R}^{n,n_u}$  matrizes resultantes da linearização de (4.3-12).

$$\underline{A} = \left. \frac{\partial f_e}{\partial x} \right|_{(0,0)} \quad (4.3-13)$$

$$\underline{B} = \left. \frac{\partial f_e}{\partial u} \right|_{(0,0)} \quad (4.3-14)$$

#### Center Manifold Theory

Se o sistema exógeno é neutralmente estável e a linearização do sistema em malha fechada (4.3-12) é estável para o ponto de equilíbrio  $(x, \rho) = (0, 0)$  então a CMT garante a existência de um *center manifold*, isto é, um mapeamento  $x = \pi(\rho)$ , exponencialmente atractivo (localmente). Além disso, e porque o *manifold* é invariante, tem todo o sentido fazer com que o erro se anule nele, isto é,  $r(\rho) - h(\pi(\rho)) = 0$ , ou seja,

$$\lim_{k \rightarrow \infty} e_\rho(k) = e(k) = h_\rho(x(k), \rho(k)) = 0 \quad (4.3-15)$$

Assim, a resposta do sistema total desde um estado inicial  $(x(0), \rho(0))$  numa vizinhança de  $(0, 0)$  converge para um estado final bem definido, independente de  $x(0)$  e dependente apenas de  $\rho(0)$ .

#### Solução do Problema

Os resultados introduzidos por Castillo-Toledo *et al.* (1993a) estabelecem condições para a existência de uma solução, assumindo as condições anteriores para as primeiras aproximações do sistema e sistema exógeno e tendo por base a teoria CMT.

#### Teorema: (Castillo-Toledo, 1993a)

Considerando que as hipóteses A1 e A2 se verificam, então o problema de regulação da saída é solucionável (localmente) se existirem mapeamentos  $x(k) = \pi(\rho(k))$  e  $u(k) = \alpha(\rho(k))$  com  $\pi(0) = 0$  e  $\alpha(0) = 0$ , satisfazendo:

$$\pi(s(\rho(k))) = f(\pi(\rho(k)), \alpha(\rho(k)), \rho(k)) \quad (4.3-16a)$$

$$0 = r(\rho(k)) - h(\pi(\rho(k))) \quad (4.3-16b)$$

Como referido inicialmente, a solução do problema de regulação é conduzida à solução de um sistema de equações de diferenças cuja solução permite determinar os mapeamentos  $x(k) = \pi(\rho(k))$  e  $u(k) = \alpha(\rho(k))$ , denominados, respectivamente *zero error constrained manifold* e *zero error constrained input*.

Quanto à estrutura do controlador, propriamente dita, Isidori e Byrnes (1990) sugeriram o uso de (4.3-17).

$$u(k) = \gamma(x(k), \rho(k)) = \alpha(\rho(k)) + K(x(k) - \pi(\rho(k))) \quad (4.3-17)$$

Note-se que o controlador é na verdade definido por  $u(k) = Kx(k) + \beta(\rho(k))$ , isto é, uma parte,  $Kx(k)$ , é função dos estados do sistema e outra,  $\beta(\rho(k)) = \alpha(\rho(k)) - K\pi(\rho(k))$ , é função dos estados do sistema exógeno, como se representa na Figura 4.3-1.

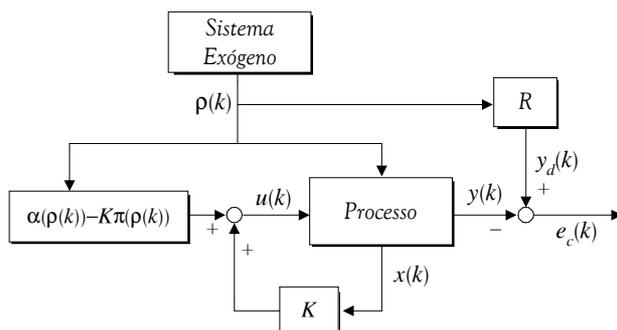


Figura 4.3-1: Estrutura de controlo.

Na Figura 4.3-1 a matriz  $R$  é definida por (4.3-18), tal que  $y_d(k) = R\rho(k)$ , com  $R: \mathfrak{R}^d \rightarrow \mathfrak{R}^n y$ .

$$R \equiv \left. \frac{\partial r}{\partial \rho} \right|_{\rho=0} \quad (4.3-18)$$

A matriz  $K \in \mathfrak{R}^{n_u \times n}$  é determinada de forma a localizar os valores próprios da primeira aproximação do sistema não linear em posições desejadas, concretamente no interior do círculo de raio unitário. Assim

$$\underline{A} + \underline{B}K = \Lambda \quad (4.3-19)$$

onde  $\underline{A}$  e  $\underline{B}$  são matrizes definidas respectivamente por (4.3-13) e (4.3-14), sendo o par  $(\underline{A}, \underline{B})$  controlável e  $\Lambda \in \mathfrak{R}^{n,n}$  é uma matriz que permite especificar os valores próprios desejados. Das várias alternativas para o cálculo de  $K$  destacam-se os algoritmos de colocação de pólos (ou melhor, de afectação de valores próprios) e as técnicas de controlo óptimo, especificados por um critério LQR<sup>3</sup>.

## Convergência

Uma vez determinada uma solução para a equação (4.3-16) garante-se a existência de um mapeamento  $x(k) = \pi(\rho(k))$ , que é um *center manifold* para o sistema em malha fechada (4.3-20).

$$x(k+1) = f_{\rho}(x(k), \gamma(x(k), \rho(k)), \rho(k)) \quad (4.3-20)$$

Uma das propriedades fundamentais do *center manifold* (Carr, 1981) é que o *manifold* é localmente atractivo, isto é, para valores na vizinhança de  $(x(0), \rho(0))$  verifica-se a condição

$$\|x(k) - \pi(\rho(k))\| \leq L \tau^k \quad (4.3-21)$$

sendo  $L$  e  $\tau$  tais que  $L > 0$  e  $0 < \tau < 1$ . Assim, garante-se a convergência do termo  $x(k) - \pi(\rho(k))$ , de forma que a acção de controlo em regime permanente é dada por  $u(k) = \gamma(x(k), \rho(k)) = \alpha(\rho(k))$ , o que, na verdade, impõe uma resposta em regime final  $y(k) = r(\rho(k)) = y_d(k)$ , (Isidori, 1996), independente do estado inicial do sistema  $x(0)$  e função apenas do valor da referência.

### 4.3.3 Algoritmo Iterativo para a Resolução das Equações de Regulação

Fundamentalmente, pelo tipo de não linearidades em questão (tangente hiperbólica), propõem-se de seguida um método iterativo capaz de solucionar eficazmente, de uma forma aproximada, as equações de regulação. O algoritmo pode ser entendido como um método de gradiente, em que o ganho de adaptação é variável em cada iteração e determinado de modo a garantir a estabilidade do procedimento iterativo. O problema é conduzido a um de atribuição de valores próprios, assegurando-se a convergência das equações de regulação, uma vez seleccionados os valores próprios adequados.

---

<sup>3</sup> LQR- Linear Quadratic Regulator

Para o efeito recorde-se o modelo neuronal em questão

$$x_n(k+1) = A x_n(k) + B u(k) + D \sigma(x_n(k)) \quad (4.3-22a)$$

$$y_n(k) = C x_n(k) \quad (4.3-22b)$$

## Algoritmo Iterativo

Defina-se o vector  $\Gamma(\rho(k)) \in \mathfrak{R}^m$ , composto pelos mapeamentos a determinar, sendo a variável  $m$  definida por  $m = n + n_u$ .

$$\Gamma(\rho(k)) = \begin{bmatrix} \pi(\rho(k)) \\ \alpha(\rho(k)) \end{bmatrix} \quad (4.3-23)$$

em que os vectores  $\pi(\rho(k)) \in \mathfrak{R}^n$  e  $\alpha(\rho(k)) \in \mathfrak{R}^{n_u}$  têm dimensões iguais ao número de estados e de entradas. Por questões de simplicidade de notação, assume-se  $\pi(\rho(k)) = \pi(\rho)$  e  $\alpha(\rho(k)) = \alpha(\rho)$ . Considere-se o caso particular do número de entradas ser igual ao número de saídas, isto é,  $n_y = n_u$ , portanto  $m = n + n_u = n + n_y$  e considere-se, ainda, o caso particular do sistema exógeno e saída desejada serem constantes, isto é,

$$\begin{aligned} \rho(k+1) &= s(\rho(k)) = \rho(k) \\ y_d(k) &= r(\rho(k)) = \rho(k) \end{aligned} \quad (4.3-24)$$

A aplicação do *Zero Output Constrained Dynamics Algorithm*, introduzido por Monaco e Normand-Cyrot (1987), conduz ao seguinte sistema de  $m$  equações.

$$i) \pi(s(\rho)) = f(\pi(\rho), \alpha(\rho), \rho)$$

$$A \pi(\rho) + B \alpha(\rho) + D \sigma(\pi(\rho)) = \pi(\rho) \quad (4.3-25)$$

$$ii) 0 = r(\rho) - h(\pi(\rho))$$

$$\begin{aligned} y_n(k) &= C x_n(k) = r(\rho) \\ C \pi(\rho) &= r(\rho) \end{aligned} \quad (4.3-26)$$

Admita-se que existe uma solução  $\pi(\rho)^*$  e  $\alpha(\rho)^*$ , que verifica as equações (4.3-26) e (4.3-25), isto é,

$$0 = A \pi(\rho)^* + B \alpha(\rho)^* + D \sigma(\pi(\rho)^*) - \pi(\rho)^* \quad (4.3-27a)$$

$$0 = C \pi(\rho)^* - r(\rho) \quad (4.3-27b)$$

Considere-se, para cada iteração  $i$ , as correspondentes soluções  $\pi(\rho)^i$  e  $\alpha(\rho)^i$ . Ainda para a iteração  $i$ , definam-se os erros  $E_1^i \in \mathfrak{X}^n$  e  $E_2^i \in \mathfrak{X}^{nu}$  dados por (4.3-28a) e (4.3-28b), respectivamente.

$$E_1^i(k) = [ A \pi(\rho)^* + B \alpha(\rho)^* + D \sigma(\pi(\rho)^*) - \pi(\rho)^* ] \\ - [ A \pi(\rho)^i + B \alpha(\rho)^i + D \sigma(\pi(\rho)^i) - \pi(\rho)^i ] \quad (4.3-28a)$$

$$E_2^i(k) = [ C \pi(\rho)^* - r(\rho) ] - [ C \pi(\rho)^i - r(\rho) ] \quad (4.3-28b)$$

Defina-se ainda o incremento entre duas iterações consecutivas como

$$\begin{bmatrix} \pi(\rho)^{i+1} \\ \alpha(\rho)^{i+1} \end{bmatrix} = \begin{bmatrix} \pi(\rho)^i \\ \alpha(\rho)^i \end{bmatrix} + \begin{bmatrix} \Delta \pi(\rho)^i \\ \Delta \alpha(\rho)^i \end{bmatrix} \quad (4.3-29)$$

Usando (4.3-28), o erro na iteração  $i+1$ , é dado por (4.3-30).

$$E_1^{i+1}(k) = [ A \pi(\rho)^* + B \alpha(\rho)^* + D \sigma(\pi(\rho)^*) - \pi(\rho)^* ] \\ - [ A \pi(\rho)^{i+1} + B \alpha(\rho)^{i+1} + D \sigma(\pi(\rho)^{i+1}) - \pi(\rho)^{i+1} ] \quad (4.3-30a)$$

$$E_2^{i+1}(k) = [ C \pi(\rho)^* - r(\rho) ] - [ C \pi(\rho)^{i+1} - r(\rho) ] \quad (4.3-30b)$$

Usando (4.3-29), entre duas iterações consecutivas,  $i$  e  $i+1$ , resultam as equações (4.3-31a) e (4.3-31b).

$$E_1^{i+1}(k) - E_1^i(k) = -A \Delta \pi(\rho)^i - B \Delta \alpha(\rho)^i \quad (4.3-31a)$$

$$-D \left( \sigma(\pi(\rho)^i + \Delta \pi(\rho)^i) - \sigma(\pi(\rho)^i) \right) + \Delta \pi(\rho)^i$$

$$E_2^{i+1}(k) - E_2^i(k) = -C \Delta \pi(\rho)^i \quad (4.3-31b)$$

Para o caso particular em estudo, a função de activação considerada (tangente hiperbólica) pertence à classe de funções *Lipschitz*. Sendo assim verifica-se (4.3-32), considerando  $x \in \mathfrak{X}$ .

$$\|\sigma(x + \Delta x) - \sigma(x)\| \leq \|\Delta x\| \quad (4.3-32)$$

ou ainda

$$\|\sigma(x + \Delta x)\| \leq \|\sigma(x)\| + \|\Delta x\| \quad (4.3-33)$$

Pode mesmo, para este tipo de funções, escrever-se:

$$\sigma(x + \Delta x) = \sigma(x) + \kappa \Delta x$$

sendo  $\kappa \in ]0, 1]$ , tal como acontece com a sua derivada, ( $\sigma'(x) \in ]0, 1]$ ), outra das propriedades interessante do tipo de não linearidade em questão. Uma vez que  $\kappa$  é um valor desconhecido considera-se como aproximação o valor da primeira derivada,  $G = \sigma'(x)$ ,

$$\sigma(x + \Delta x) \approx \sigma(x) + G \Delta x \quad (4.3-34)$$

com

$$G = \sigma'(x) = \frac{\partial \sigma(x)}{\partial x} \quad (4.3-35)$$

Para o caso de  $x = \pi \in \mathfrak{R}^n$ , como em (4.3-31), considera-se

$$\|\sigma(\pi(\rho) + \Delta\pi(\rho))\| \leq \|\sigma(\pi(\rho))\| + \|\Delta\pi(\rho)\| \quad (4.3-36)$$

de que resulta a aproximação,

$$\sigma(\pi(\rho) + \Delta\pi(\rho)) \approx \sigma(\pi(\rho)) + G \Delta\pi(\rho) \quad (4.3-37)$$

com  $G \in \mathfrak{R}^{n,n}$  uma matriz diagonal definida por (4.3-38).

$$G = \begin{bmatrix} \sigma'(\pi_1(\rho)) & & 0 \\ & \cdot & \\ 0 & & \sigma'(\pi_n(\rho)) \end{bmatrix} \quad (4.3-38)$$

Assim, substituindo a aproximação ( $\approx$ ) pela igualdade ( $=$ ), tem-se (4.3-28).

$$E_1^{i+1}(k) - E_1^i(k) = -A \Delta\pi(\rho)^i - B \Delta\alpha(\rho)^i + \Delta\pi(\rho)^i - D G \Delta\pi(\rho) \quad (4.3-39a)$$

$$E_2^{i+1}(k) - E_2^i(k) = -C \Delta\pi(\rho)^i \quad (4.3-39b)$$

A equação (4.3-39) pode ser escrita matricialmente como:

$$E^{i+1}(k) - E^i(k) = - \begin{bmatrix} A + D G - I_n & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta\pi(\rho)^i \\ \Delta\alpha(\rho)^i \end{bmatrix}$$

$$E^{i+1}(k) - E^i(k) = -M \Delta\Gamma(\rho)^i \quad (4.3-40)$$

sendo  $E^i \in \mathfrak{R}^m$  o erro relativo aos mapeamentos  $\pi(\rho)^i$  e  $\alpha(\rho)^i$  na iteração  $i$ ,  $I_n \in \mathfrak{R}^{n,n}$  uma matriz identidade e  $M \in \mathfrak{R}^{m,m}$ . Usando este procedimento o problema pode ser formulado como um problema de colocação de pólos. De facto, se o incremento  $\Delta\Gamma(\rho)^i$  for escolhido em função do erro da iteração  $i$ ,

$$\Delta\Gamma(\rho)^i = K E^i(k) \quad (4.3-41)$$

tem-se (4.3-42),

$$E^{i+1}(k) = [I_m - K M] E^i(k) \quad (4.3-42)$$

em que  $I_m \in \mathfrak{R}^{m,m}$  e  $K \in \mathfrak{R}^{m,m}$ .

Assim sendo, através de uma escolha adequada de  $K$  é possível garantir que o erro convirja se se estabelecer a relação (4.3-43),

$$E^{i+1}(k) = \Lambda E^i(k) \quad (4.3-43)$$

em que a matriz  $\Lambda \in \mathfrak{R}^{m,m}$  é uma matriz cujos valores próprios se localizam no interior do círculo de raio unitário.

### Teorema 3: Convergência do Algoritmo RSRN

Seja o sistema a controlar definido por (4.3-22).

$$x_n(k+1) = A x_n(k) + B u(k) + D \sigma(x_n(k))$$

$$y_n(k) = C x_n(k)$$

Seja ainda a matriz  $K$  determinada de forma a que os valores próprios da matriz  $\Lambda$  se localizem no interior do círculo de raio unitário.

$$\Lambda = [I_m - K M] \quad (4.3-44)$$

A seguinte especificação para o incremento  $\Delta\Gamma$

$$\Delta\Gamma(\rho)^i = K E^i(k) \quad (4.3-45)$$

garante a convergência do erro, definido em (4.3-28), se

i) o par  $(A + DG, B)$  for controlável

ii)

$$\text{rank}(M) = \text{rank} \begin{bmatrix} A + DG - I_n & B \\ C & 0 \end{bmatrix} = m, \quad m = n + n_u = n + n_y$$

### Demonstração

De forma a que a teoria CMT possa ser aplicada e garantida a existência de uma solução para o problema da regulação, devem assegurar-se determinadas condições, nomeadamente, A1 e A2 (páginas 160 e 161).

Para o caso em estudo, garante-se automaticamente a hipótese A1, uma vez que se considera a referência constante, (4.3-24). A hipótese A2 impõe a condição de controlabilidade para a primeira aproximação do sistema.

Uma vez verificadas A1 e A2 garante-se, pela teoria de CMT, a existência de uma solução para o problema de regulação.

O problema da existência de solução está directamente relacionado com a convergência do erro na equação (4.3-40). No entanto, esta convergência pode ser entendida como um problema de convergência de um problema de controlo por realimentação de variáveis de estado. De facto, considerando a notação usual neste tipo de problemas

$$x(k+1) = \underline{A} x(k) + \underline{B} u(k) \quad (4.3-46)$$

pode-se relacionar esta última equação (4.3-46) com a (4.3-40), como a seguir se apresenta.

$$\begin{aligned} x(k) &= E^i(k) \\ \underline{A} &= I_m \\ \underline{B} &= -M \\ u(k) &= \Delta\Gamma(\rho)^i \end{aligned} \quad (4.3-47)$$

Sabe-se da teoria de controlo linear que se o sistema (4.3-46) for controlável, então é possível, através de uma realimentação de variáveis de estado especificar  $u(k) = K x(k)$ , de forma a que  $\lim_{k \rightarrow \infty} x(k) = 0$ . Para que  $K$  exista, isto é, para que o sistema seja controlável

deve-se verificar para um sistema de ordem  $n$  a condição de controlabilidade (4.3-48), definida para o par  $(\underline{A}, \underline{B})$ ,

$$\text{rank} \left( \begin{bmatrix} \underline{B} & \underline{A}\underline{B} & \underline{A}^2\underline{B} & \dots & \underline{A}^{n-1}\underline{B} \end{bmatrix} \right) = n \quad (4.3-48)$$

Usando (4.3-47) então deve verificar-se, por analogia, (4.3-49).

$$\text{rank} \left( \begin{bmatrix} -M & -IM & -I^2M & \dots & -I^{m-1}M \end{bmatrix} \right) = m \quad (4.3-49)$$

o que é equivalente a

$$\text{rank}(M) = m \quad (4.3-50)$$

■

Note-se que teoricamente é possível em (4.3-42), caso a matriz  $M$  seja não singular, optar por  $K = M^{-1}$  e, desta forma, anular-se-ia o erro apenas numa iteração. Contudo, uma vez que se está a fazer uma aproximação (derivada) não é adequado este procedimento, já que não se garante na verdade a existência de uma solução exacta. Assim, é na prática preferível optar por um método iterativo, além disso mais robusto, que garanta a obtenção de resultados perfeitamente aceitáveis.

As equações de regulação (4.3-25) e (4.3-26), que permitem solucionar o problema de regulação podem ser interpretadas segundo outra perspectiva. Podem ser vistas como representando uma rede neuronal recorrente, com a mesma arquitectura que a rede neuronal original usada (4.3-22) e, desta forma, pode-se dizer que as equações de regulação e a rede recorrente se complementam.

Entendida desta forma, a solução das equações de regulação podem ser entendidas como um problema “*dual*” do problema de treino de uma rede. De facto, no caso do treino são conhecidas as entradas e saídas desejadas e pretende-se determinar os parâmetros (pesos) da rede. Neste caso conhecem-se os parâmetros da rede, as matrizes  $A$ ,  $B$ ,  $C$  e  $D$  e pretende-se, então, determinar a entrada  $u = \alpha(\rho)$  e a saída da unidade interna  $x = \pi(\rho)$ , de forma que a saída da rede convirja para o valor desejado  $r(\rho)$ .

Assim, da mesma forma que é possível aplicar este algoritmo à determinação de entradas e estados é também possível aplicá-lo directamente à adaptação de parâmetros de uma rede neuronal, procedimento proposto em Henriques *et al.* (2000b) e que a seguir se expõe. Este pertencente à classe de algoritmos de gradiente com um ganho variável, determinado de forma a garantir a convergência do erro de estimação.

### 4.3.4 Aplicação do Algoritmo Iterativo na Estimação de Parâmetros

Vários métodos têm sido propostos para efectuar a actualização dos parâmetros de uma rede neuronal recorrente. Num método iterativo típico, implementado em-linha, a actualização de parâmetros é efectuada, como referido na secção 2.3.1, por uma equação do tipo,

$$W(k+1) = W(k) + \Delta W(k) \quad (4.3-51)$$

Num método de gradiente o incremento é obtido a partir de informação de primeira ordem (primeiras derivadas) e do erro de estimação. Assim, na equação (4.3-51), o incremento  $\Delta W(k)$  é substituído por uma equação do tipo,

$$\Delta W(k) = \eta(k) \Phi(k) e_y(k) \quad (4.3-52)$$

em que  $e_y(k)$  define o erro de estimação,  $e_y(k) = y(k) - y_n(k)$ , diferença entre a saída do processo real e do modelo neuronal,  $\eta(k)$  define um ganho e  $\Phi(k)$  é função da informação disponível.

#### Algoritmo Iterativo

Seja então o sistema cujos parâmetros se pretendem determinar dado por:

$$\begin{aligned} x_n(k+1) &= A^* x_n(k) + B^* u(k) + D^* \sigma(x_n(k)) \\ y_n(k) &= C^* x_n(k) \end{aligned} \quad (4.3-53)$$

Tal como considerado na secção 4.2.3 as matrizes  $A^*$  e  $C^*$  assumem-se fixas e as matrizes  $B^*$  e  $D^*$  são em cada instante actualizadas, ou seja.

$$\begin{aligned} x_n(k+1) &= A^* x_n(k) + B(k) u(k) + D(k) \sigma(x_n(k)) \\ y_n(k) &= C^* x_n(k) \end{aligned} \quad (4.3-54)$$

Considere-se a matriz constituída pelos parâmetros a determinar,  $W(k) \in \mathfrak{R}^{n,m}$ , ( $m = n + n_u$ ) definida por (4.3-55).

$$W(k) = [ D(k) : B(k) ] \quad (4.3-55)$$

O erro de estimação  $e_y^i(k)$ , em cada iteração  $i$ , é definido por:

$$e_y^i(k) = y(k) - y_n^i(k) \quad (4.3-56)$$

A partir de (4.3-53) e de (4.3-54) é possível escrever, para o instante de tempo  $k$ ,

$$\begin{aligned} e_y^i(k) &= C^* A^* x_n(k-1) + C^* B^* u(k-1) + C^* D^* \sigma(x_n(k-1)) \\ &- C^* A^* x_n(k-1) - C^* B(k)^i u(k-1) - C^* D(k)^i \sigma(x_n(k-1)) \end{aligned} \quad (4.3-57)$$

Considerando  $C = C^*$  e  $A = A^*$

$$e_y^i(k) = C (D^* - D(k)^i) \sigma(x_n(k-1)) + C (B^* - B(k)^i) u(k-1) \quad (4.3-58)$$

ou ainda

$$e_y^i(k) = C \tilde{D}^i \sigma(x_n(k)) + C \tilde{B}^i u(k) \quad (4.3-59)$$

em que

$$\begin{aligned} \tilde{D}^i &= D^* - D(k)^i \\ \tilde{B}^i &= B^* - B(k)^i \end{aligned}$$

A equação (4.3-59) pode-se escrever de uma forma simplificada como:

$$e_y^i(k) = C \tilde{W}(k)^i \varphi(k-1) \quad (4.3-60)$$

em que

$$\tilde{W}(k)^i = [ \tilde{D}(k)^i \quad \tilde{B}(k)^i ]$$

$$\varphi(k-1) = [ \sigma(x_n(k-1)) \quad u(k-1) ]$$

Seja ainda a diferença entre duas iterações sucessivas definidas por (4.3-61).

$$e_y^{i+1}(k) - e_y^i(k) = C \tilde{W}(k)^{i+1} \varphi(k-1) - C \tilde{W}(k)^i \varphi(k-1) \quad (4.3-61)$$

Porque

$$\begin{aligned} \tilde{W}(k)^{i+1} - \tilde{W}(k)^i &= [ \tilde{D}(k)^{i+1} \quad \tilde{B}(k)^{i+1} ] - [ \tilde{D}(k)^i \quad \tilde{B}(k)^i ] \\ &= [ D^* - D(k)^{i+1} \quad B^* - B(k)^{i+1} ] - [ D^* - D(k)^i \quad B^* - B(k)^i ] \\ &= [ D(k)^i - D(k)^{i+1} \quad B(k)^i - B(k)^{i+1} ] \end{aligned}$$

$$= - \Delta W(k)^i$$

tem-se:

$$e_y^{i+1}(k) - e_y^i(k) = - C \Delta W(k)^i \varphi(k-1)$$

Para que o erro de estimação convirja, o incremento  $\Delta W(k)^i$  deve ser determinado de tal forma que entre duas iterações sucessivas se verifique:

$$e_y^{i+1}(k) = \Lambda e_y^i(k) \quad (4.3-62)$$

em que  $\Lambda$  é uma matriz de valores próprios localizados no interior do círculo de raio unitário.

Especificando do incremento  $\Delta W(k)^i$  por

$$\Delta W(k)^i = [ C^T [ C C^T ]^{-1} ] M e_y^{i-1}(k) [ [ \varphi(k-1)^T \varphi(k-1) ]^{-1} \varphi(k-1)^T ] \quad (4.3-63)$$

tem como resultado a seguinte equação para o erro em cada iteração:

$$e_y^{i+1}(k) = (I - M) e_y^i(k) \quad (4.3-64)$$

De forma a garantir que o erro converge, a matriz  $M$  deve ser especificada de forma a que os valor próprios da matriz  $(I - M) = \Lambda$  se localizem no interior do círculo de raio unitário, ou seja,  $M$  deve ser determinada a partir de (4.3-65).

$$M = I + \Lambda \quad (4.3-65)$$

Note-se que é necessário, para que o algoritmo seja aplicável, que a matriz  $[ C C^T ]$  tenha inversa e o mesmo aconteça com  $[ \varphi(k-1)^T \varphi(k-1) ]$ .

Quanto à primeira questão, e uma vez que a matriz  $C$  é fixa o problema é relativamente fácil de ultrapassar.

Quanto ao segundo, caso isso se verifique opta-se simplesmente por não efectuar qualquer adaptação dos parâmetros, o que tem todo o sentido, uma vez que um valor nulo traduz a inexistência de informação. Note-se ainda que  $[ \varphi(k-1)^T \varphi(k-1) ]$  é um escalar no caso SISO, logo a inversa existe desde que o seu valor seja não nulo. Para o caso MIMO, é necessário garantir que a matriz de informação  $[ \varphi(k-1)^T \varphi(k-1) ]$  tenha inversa, o que pode ser garantido caso a entrada seja persistentemente excitante.

# 4.4

## Conclusão: Metodologia RSRN

Nesta secção apresentou-se a metodologia RSRN- *Regulação da Saída com Redes Neurais Recorrentes*, resultante da combinação de uma rede neuronal recorrente com a teoria de regulação, de forma a obter um sistema de controlo para sistemas discretos não lineares, como se mostra na Figura 4.4-1.

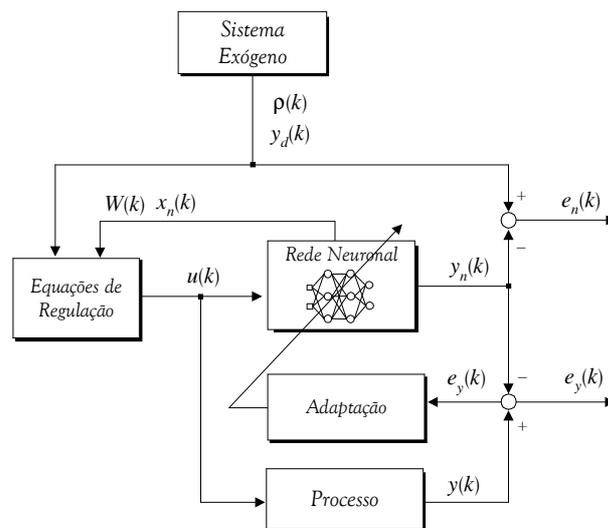


Figura 4.4-1: Estrutura de controlo RSRN proposta.

Neste sentido, foi inicialmente definida na secção 4.1 uma estrutura neuronal recorrente, pertencente à classe de sistemas *affine*. Justificou-se tal opção como uma tentativa de estabelecer um compromisso entre dois objectivos fundamentais: por um lado constituir um modelo dinâmico capaz de descrever um sistema não linear genérico a controlar e, por outro, proporcionar uma solução que viabilize o projecto de técnicas de estimação e projecto de controladores.

O objectivo final de um sistema de controlo é garantir que o erro  $e_c(k)$ , definido pela diferença entre a saída desejada e a do processo real, (4.4-1), seja nulo.

$$e_c(k) = y_d(k) - y(k) \quad (4.4-1)$$

A equação anterior pode, no entanto, ser escrita como (4.4-2),

$$e_c(k) = (y_d(k) - y_n(k)) + (y_n(k) - y(k))$$

$$e_c(k) = e_n(k) - e_y(k) \quad (4.4-2)$$

ou seja, o erro de controlo é dada pela diferença entre o erro de regulação (definido para o modelo neuronal) e o erro de estimação.

Desta forma, com a finalidade de garantir os objectivos para o sistema de controlo considerou-se a subdivisão do problema em dois problemas parciais:

- i) garantir a convergência para zero do erro de estimação  $e_y(k)$ ;
- ii) garantir a convergência para zero do erro de regulação  $e_n(k)$ .

## Estimação de Parâmetros

Pelo sucesso que os métodos oriundos da teoria de observação têm tido no treino em-linha de modelos dinâmicos não lineares interpretam-se as redes neuronais como modelos de estados e é sobre esta perspectiva que se aborda o problema de estimação dos seus parâmetros.

À semelhança de um filtro de Kalman dual, divide-se o problema em duas partes: estimação de estados e adaptação de parâmetros. Para esta classe particular de estruturas *affine* estabelecem-se resultados de estabilidade e de convergência do erro de estimação  $e_y(k)$ , a partir do segundo método de Lyapunov e do princípio do invariância de La Salle.

## Regulação da Saída

A teoria de regulação, aplicada ao modelo neuronal, garante a estabilidade do sistema de controlo e simultaneamente a convergência para zero do erro de regulação  $e_n(k)$ , ou seja, que a saída do modelo neuronal  $y_n(k)$  convirja para um determinado valor desejado  $y_d(k)$ , gerado por um sistema exógeno.

Na secção 4.3 foram revistos conceitos relativos à teoria de regulação e efectuada a sua aplicação ao modelo neuronal proposto. Neste caso, devido ao tipo de não linearidades envolvidas (tangente hiperbólica) é possível estabelecer resultados capazes de assegurarem a convergência das equações de regulação.

## Adaptação de Parâmetros e Regulação da Saída

A partir de (4.4-2) é possível escrever (4.4-3):

$$\| e_c(k) \| \leq \| e_n(k) \| + \| e_y(k) \| \quad (4.4-3)$$

Assim, e uma vez ajustados em cada instante os parâmetros da rede, e admitindo o princípio da separação, garante-se o objectivo para o qual o sistema de controlo é projectado, a convergência do erro  $e_c(k)$  para zero. De facto, uma vez que o regulador assegura

$$\lim_{k \rightarrow \infty} [ y_d(k) - y_n(k) ] = e_n(k) = 0$$

o erro total convergirá se o erro de estimação também convergir, isto é, se

$$\lim_{k \rightarrow \infty} [ y(k) - y_n(k) ] = e_y(k) = 0$$

o que de facto é garantido pelo mecanismo de adaptação dos parâmetros e dos estados.

Em conclusão, propôs-se neste capítulo uma metodologia passível de ser usada de uma forma sistemática no projecto de controladores para processos discretos.

Verifica-se que as redes recorrentes possuem características intrínsecas que fazem delas candidatas ideais na parametrização de processos. Inseridas numa estrutura adaptativa e combinadas com a teoria de regulação da saída, que incorpora de uma forma inerente um princípio de separação, apresentou-se uma solução com potencial de implementação prática, especialmente para processos não lineares.



# 5.

## Casos de Estudo

*“Needless to say, there are many control methods for nonlinear systems. In a specific application, the method presented may or may not be better than some other method. But the more methods that are available, the greater is the likelihood of finding the most appropriate method for the application”*

B. Friedland, 1997.

São considerados neste capítulo três casos de estudo da aplicação da metodologia de controlo adaptativa *RSRN*, proposta nesta dissertação, sendo um de simulação e dois processos práticos à escala laboratorial.

Num dos casos de estudo usa-se um processo térmico laboratorial e o objectivo fundamental consiste em investigar as características relativas à inicialização de parâmetros nas etapas de identificação e de controlo da metodologia *RSRN*. Avalia-se ainda a sua robustez em presença de variações na dinâmica do processo e compara-se o seu desempenho com um controlador *PI*.

Noutro considera-se o problema do controlo da concentração de um componente num reactor químico (*continuous stirred tank reactor*), um *benchmark* de controlo de sistemas. Este processo, que se caracteriza por ser fortemente não linear, é um caso típico para o qual as técnicas de controlo lineares apresentam dificuldades, adequado portanto ao estudo de metodologias de controlo não lineares. Neste sentido, desenvolve-se e implementa-se computacionalmente a metodologia *RSRN*, comparando-se a sua eficácia com um controlador convencional *PI*.

No último processo considerado está em causa o controlo de um sistema de níveis, multi-variável e com acoplamento entre as suas variáveis. Neste processo os estados são acessíveis, sendo portanto adequado à implementação de estratégias de controlo que façam uso da realimentação dos estados. Assim, é implementada uma estratégia de controlo linear (por localização dos valores próprios) em que se assume o conhecimento quer do modelo físico do processo, quer do valor dos seus estados. O desempenho deste controlador é comparado com o desempenho proporcionado pela metodologia *RSRN*, em que é usada apenas informação de entrada-saída e se assume o desconhecimento dos estados.

Este capítulo encontra-se dividido em quatro secções distintas. Nas secções 5.1, 5.2 e 5.3 descreve-se cada um dos processos referidos: o processo térmico, o reactor químico e o sistema dos três tanques. Para cada um dos casos apresentam-se os correspondentes resultados proporcionados pelas metodologias em questão. Conclui-se na secção 5.4, estabelecendo algumas considerações acerca dos resultados obtidos.

# 5.1

## Processo Térmico - PT326

*Neste primeiro caso de estudo aplica-se o controlador adaptativo neuronal RSRN ao processo térmico PT326<sup>1</sup>, um processo laboratorial que se caracteriza uma dinâmica não linear, variável no tempo, que pode facilmente ser sujeito a perturbações externas.*

*Os resultados da metodologia proposta são comparados com os proporcionados por um controlador PI.*

### 5.1.1 Introdução

Apesar de se tratar de um processo de dimensões reduzidas e que, em termos de funcionamento, é relativamente simples, o processo laboratorial PT326 apresenta dificuldades que em certa medida podem ser encontradas em muitos processos industriais, tais como em fornalhas, caldeiras e sistemas de ar condicionado. É um processo muito versátil, cujas características podem ser facilmente modificadas, permitindo deste modo investigar a capacidade de metodologias de controlo, em especial quando se deseja examinar a influência de variações da dinâmica ou perturbações sobre o processo controlado. É assim por isso adequado para implementar as ideias desenvolvidas nesta dissertação, concretamente a capacidade do controlador adaptativo neuronal em lidar com sistemas de dinâmica variável.

#### Descrição do Processo

O processo térmico PT326, mostrado na Figura 5.1-1, é fundamentalmente um processo no qual se processa uma transferência de calor.

Basicamente, um ventilador centrífugo aspira ar do meio ambiente e força-o a passar por um tubo. No interior deste existe uma resistência eléctrica (R), em forma de grelha, que aquece quando se lhe aplica uma determinada potência eléctrica. O calor gerado pela

---

<sup>1</sup> Process Trainer PT326, Feedback Instruments Limited

resistência é libertado e transmitido, por convecção, ao ar em circulação. A temperatura do ar aquecido pode ser medida por um sensor (termopar), que pode ser inserido numa de três posições,  $\Upsilon = I$ ,  $\Upsilon = II$  ou  $\Upsilon = III$  respectivamente a 2.8 cm, 14.0 cm e 28.0 cm da grelha de aquecimento. Na Figura 5.1-2 ilustra-se esquematicamente o processo.

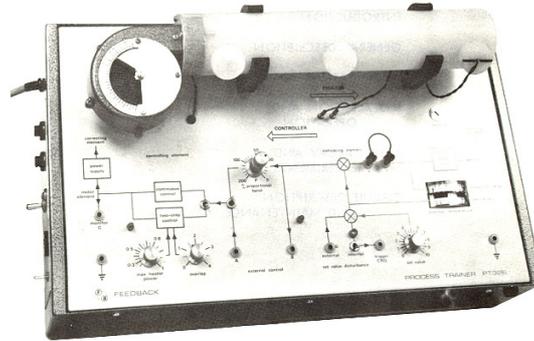


Figura 5.1-1: Processo PT326.

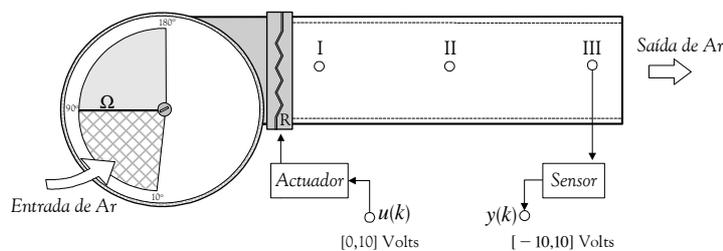


Figura 5.1-2: Diagrama esquemático do processo PT326.

A carga do processo consiste no caudal de ar que percorre o tubo, podendo ser ajustada manualmente através de uma abertura na admissão, variável entre  $\Omega = 10^\circ$  e  $\Omega = 180^\circ$  correspondendo, respectivamente, aos valores de abertura mínima e máxima.

O processo *PT326* é na verdade um sistema *MISO*<sup>2</sup>. Tem duas entradas possíveis de variar, a potência de aquecimento e a admissão de ar e uma variável a controlar, a temperatura do ar (medida numa das posições  $\Upsilon = I$ ,  $II$  ou  $III$ ). No entanto, para efeitos de controlo, assume-se um sistema *SISO* considerando como variável de actuação a potência de aquecimento. O caudal de ar é considerado como uma entrada externa, mas entendida como uma perturbação/variação a introduzir sobre a dinâmica do processo.

A acção de controlo  $u(k)$ , que determina a potência a ser fornecida ao actuador, é enviada através de um conversor D/A (Digital/Analógico) para a resistência aquecedora, manipulada através de um sinal eléctrico, numa gama de  $[0,10]$  Volts. A leitura da

<sup>2</sup> *MISO*- Multiple Input Single Output

temperatura  $y(k)$  é efectuada pelo sensor, recebido de um conversor A/D (Analógico/Digital), também como um sinal em tensão, no intervalo  $[-10,10]$  Volts.

## Caracterização do Processo

De forma a caracterizar o processo começa por se averiguar experimentalmente qual a relação entre a tensão medida pelo sensor e o valor real de temperatura, relação que se mostra na Figura 5.1-3. Como se pode observar, a variação da saída do processo no intervalo  $[0,10]$  Volts corresponde a uma variação de temperatura real na ordem dos  $40^{\circ}\text{C}$  a  $60^{\circ}\text{C}$ . Assim, em condições normais, a gama não está limitada a  $[0,10]$  Volts e facilmente se obtêm valores negativos, relativos a valores inferiores a  $40^{\circ}\text{C}$ , correspondendo o valor mínimo ao valor da temperatura ambiente.

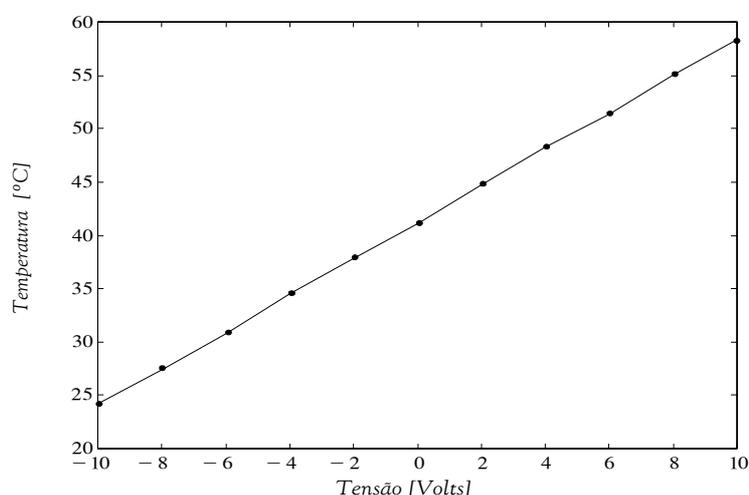


Figura 5.1-3: Relação entre temperatura real e tensão adquirida pelo sensor.

Ainda com o objectivo de caracterizar o processo, analisa-se o seu comportamento dinâmico em malha aberta, quando se lhe aplica uma variação na entrada (potência de aquecimento). Foram consideradas três situações:  $\Omega = 30^{\circ}$  e o sensor na posição  $\Upsilon = \text{II}$ ,  $\Omega = 50^{\circ}$  e o sensor na posição  $\Upsilon = \text{I}$  e  $\Omega = 10^{\circ}$  e o sensor na posição  $\Upsilon = \text{III}$ . A evolução da temperatura da saída, expressa em Volts, para cada um dos casos é mostrada respectivamente nas Figura 5.1-4, Figura 5.1-5 e Figura 5.1-6.

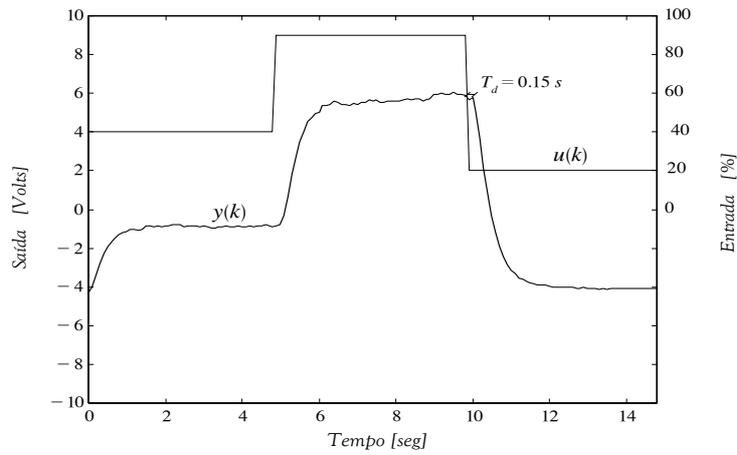


Figura 5.1-4: Evolução dinâmica da saída,  $\Omega = 30^\circ$  e  $\Upsilon = \text{II}$ .

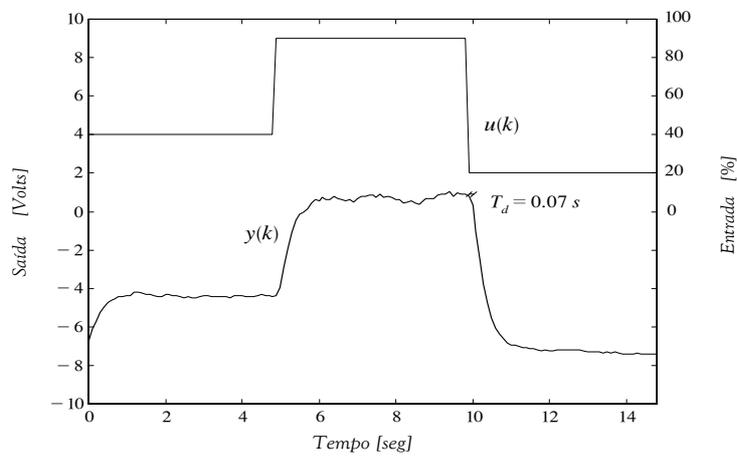


Figura 5.1-5: Evolução dinâmica da saída,  $\Omega = 50^\circ$  e  $\Upsilon = \text{I}$ .

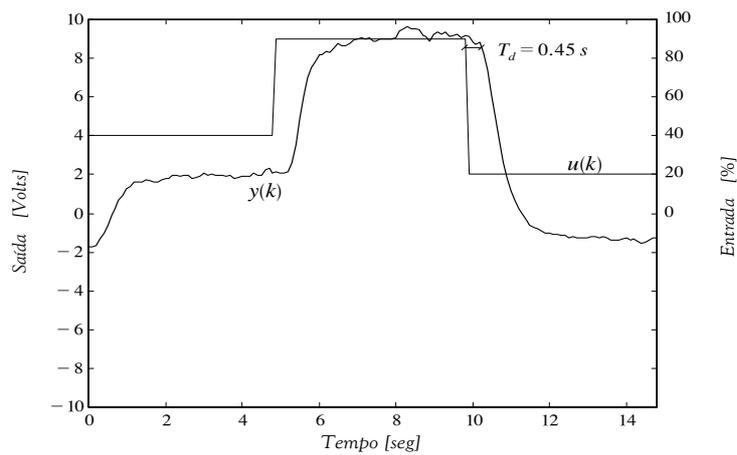


Figura 5.1-6: Evolução dinâmica da saída,  $\Omega = 10^\circ$  e  $\Upsilon = \text{III}$ .

Pela análise das figuras anteriores pode concluir-se, que em termos de funcionamento, a dinâmica do sistema depende tanto da potência de aquecimento como do caudal do ar forçado a circular. Em relação ao processo de transferência de calor propriamente dito, e de acordo com os resultados experimentais, este pode-se identificar com as características de um sistema de primeira ordem, com uma constante de tempo,  $T_1$ , na ordem dos 0.5 segundos.

Verifica-se ainda nas experiências anteriores a existência de um tempo de atraso,  $T_d$ , directamente relacionado com a posição do sensor e inversamente relacionado com a velocidade do ar, que por sua vez é directamente proporcional à abertura da admissão de ar,  $\Omega$ . Salienta-se o facto de ser possível, para as experiências realizadas, manipulando a admissão de ar e a posição do sensor, alcançar tempos de atraso compreendidos entre um valor mínimo de  $T_d = 0.07$  segundos (Figura 5.1-5) e um valor máximo de  $T_d = 0.45$  segundos (Figura 5.1-6). Assim, se a velocidade do ar for  $v$  e a distância entre a sensor e o elemento de aquecimento for  $l$ , o tempo de atraso será descrito por uma relação do tipo (5.1-1).

$$T_d = \frac{l}{v} = \frac{l}{f(\Omega)} \quad (5.1-1)$$

Tendo em atenção estes factores, o processo pode ser modelizado, de uma forma aproximada, por uma função de transferência  $P(s)$  do tipo,

$$P(s) = \frac{G}{1 + s T_1} e^{-s T_d} \quad (5.1-2)$$

conhecida por modelo do tipo Bröida. A variável  $G \in \Re$  define o ganho do processo e depende fundamentalmente da região de operação, que por sua vez é também influenciada pelo valor de admissão do ar,  $\Omega$ .

## Configuração

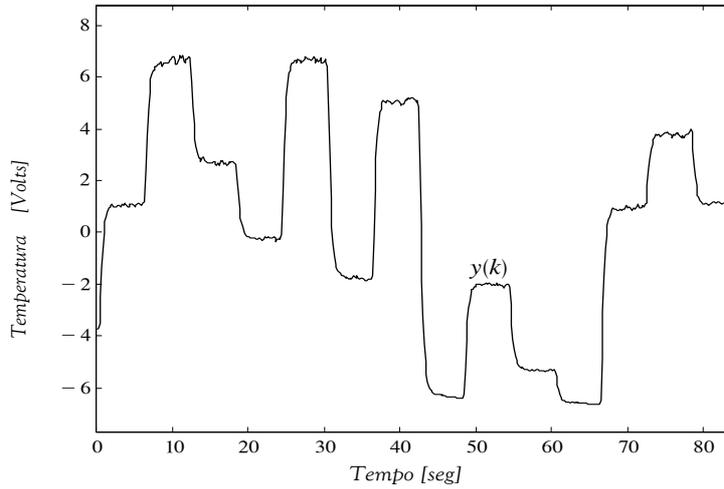
Nas experiências seguintes, e a menos que não seja referido nada em contrário, foi considerado o processo a operar nas condições apresentadas na Tabela 5.1-1.

Parâmetro	Descrição	Valor
$\Omega$	Admissão de ar	30°
$\Upsilon$	Posição do sensor	II
$T$	Intervalo de amostragem	0.15 seg

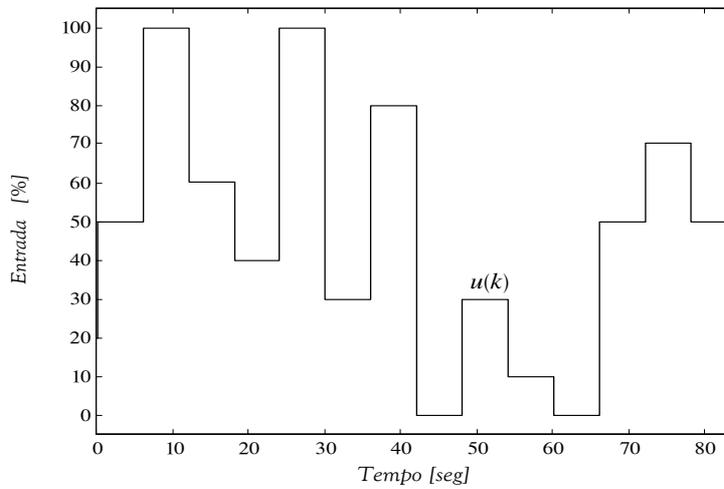
Tabela 5.1-1: Configuração utilizada nas experiências realizadas.

## 5.1.2 Identificação Experimental

De forma a determinar os parâmetros do modelo neuronal, foi adquirido do processo um conjunto de dados de saída, Figura 5.1-7(a), resultantes da aplicação da entrada mostrada na Figura 5.1-7(b).



(a)- Saída do processo PT326, ( $y$ ).



(b)- Acção de controlo, ( $u$ ).

Figura 5.1-7: Sequência de dados de identificação.

Os dados de entrada e de saída foram normalizados usando (5.1-3),

$$u_n(k) = 0.1 u(k) \quad (5.1-3a)$$

$$y_n(k) = 0.1 (y(k) - y_{min}) \quad (5.1-3b)$$

em que as variáveis  $u_n(k)$  e  $y_n(k)$  representam, respectivamente, os valores considerados como entradas e saídas utilizadas pela rede neuronal, sendo  $y_{min}$  o valor de tensão correspondente à temperatura mínima (temperatura ambiente).

De forma a identificar o processo foi, numa primeira fase, obtido um modelo linear usando para o efeito a função *ARX*, disponível no Matlab (*IdentificationToolbox*, 1998). Esta implementa o método dos mínimos quadráticos, tendo sido considerado neste caso um modelo de segunda ordem. A evolução da saída deste modelo,  $y_l(k)$ , mostra-se na Figura 5.1-8.

Após algumas experiências considerou-se como número de neurónios  $n = 2$ . Relativamente às matrizes que definem o modelo neuronal,  $A$ ,  $B$ ,  $C$  e  $D$ , estas foram inicializadas tendo em conta o conhecimento existente acerca da constante de tempo do processo (na ordem dos 0.5 segundos), valor usado na determinação dos valores próprios da matriz  $A$  (diagonal). As restantes matrizes  $B$ ,  $C$  e  $D$ , foram inicializadas como se descreve:  $B$  e  $C$  aleatoriamente, com uma distribuição Gaussiana, no intervalo  $[-0.5, 0.5]$  e  $D = 0.01 \times A$ . O algoritmo de estimação utilizado foi o de Levenberg-Marquardt (secção 2.3-7).

Da identificação resultou o modelo neuronal (5.1-4), caracterizado pelos valores dados em (5.1-5), cuja evolução,  $y_n(k)$ , se mostra na Figura 5.1-8.

$$x_n(k+1) = A x_n(k) + B u(k) + D \sigma(x_n(k)) \quad (5.1-4a)$$

$$y_n(k) = C x_n(k) \quad (5.1-4b)$$

$$A = \begin{bmatrix} 0.6698 & -0.0029 \\ -0.0002 & 0.6699 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.4637 \\ 0.8683 \end{bmatrix} \quad (5.1-5)$$

$$C = [ 0.3453 \quad -0.0889 ]$$

$$D = \begin{bmatrix} -0.0848 & 1.1318 \\ 0.0225 & -0.4097 \end{bmatrix}$$

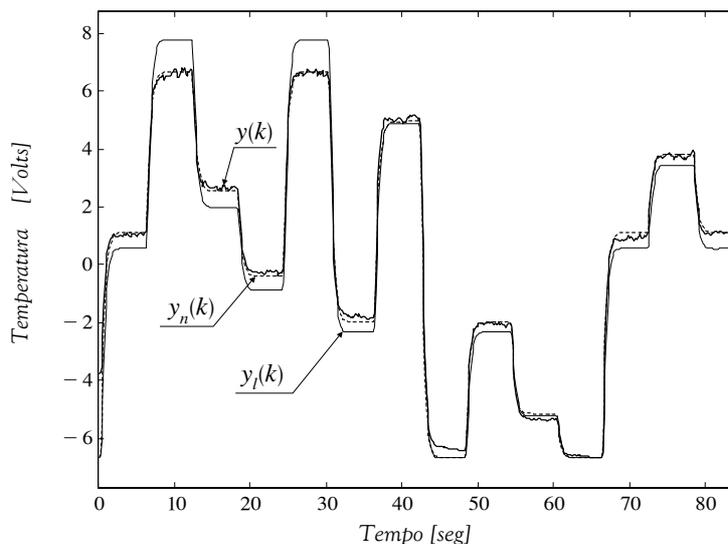


Figura 5.1-8: Identificação do PT326; dados reais ( $y$ ), do modelo neuronal ( $y_n$ ) e do modelo linear ( $y_l$ ).

Da figura anterior é visível que o modelo neuronal, não sendo perfeito, é claramente preferível ao modelo linear. De facto as não linearidades do processo, principalmente manifestadas sob a forma de um ganho variável para cada regime de operação, são capturadas pelo modelo neuronal, o que não é possível de efectuar com o modelo linear.

### 5.1.3 Inicialização de Parâmetros

O objectivo desta experiência reside na investigação da capacidade de adaptação da metodologia proposta, e de que forma os procedimentos de identificação e de controlo influenciam o sistema global de controlo. Neste sentido são realizadas três grupos de experiências: um primeiro, em que se considera um controlador não adaptativo; um segundo onde se investiga a influência da especificação dos parâmetros relativos à etapa de identificação; um terceiro, análogo ao anterior, relativo ao projecto do controlador.

Para o efeito recordam-se os procedimentos em causa para as etapas de identificação e de controlo:

#### Identificação

##### Estimação de “estados”

A estimação dos estados da rede é efectuada pelo observador,

$$x_n(k+1) = A x_n(k) + B u(k) + D (x_n(k)) + L (y(k) - C x_n(k))$$

em que o ganho  $L$  é determinado de forma a posicionar os valores próprios da matriz  $A_o$  em locais desejados.

$$A_o = A - L C$$

*Adaptação de Parâmetros*

A adaptação dos parâmetros da rede recorrente é implementada pela seguinte equação,

$$\Delta W(k) = M(k)^{-1} \varphi(k)^T P A \varepsilon(k)$$

em que  $M$  e  $\varphi$  dependem de informação passada e  $P$  é uma matriz determinada a partir da solução da equação de Lyapunov,

$$A^T P A - P = -Q_o$$

sendo  $Q_o$  uma matriz positiva definida, a ser especificada. A solução da equação anterior, ou seja, o valor de  $P$  é determinado usando a função *dlyap* (Matlab Control Toolbox, 1998).

**Controlo**

A acção de controlo resulta da implementação de,

$$u(k) = \alpha(\rho(k)) + K (x_n(k) - \pi(\rho(k)))$$

em que matriz  $K$  resulta da seguinte especificação de valores próprios,

$$\underline{A} + \underline{B} K = \underline{\Lambda}$$

com  $\underline{A}$  e  $\underline{B}$  a serem definidos pela primeira aproximação do sistema.

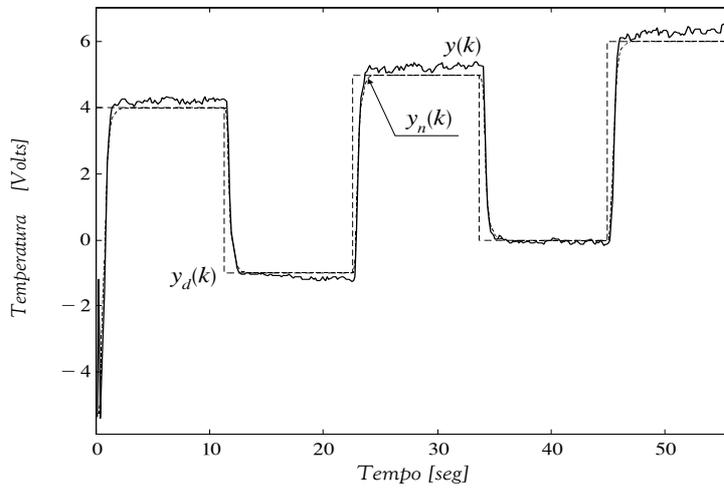
**Controlador RSRN Não Adaptativo**

Tendo por base o modelo neuronal obtido (5.1-5), projectou-se o regulador, tendo sido especificados os seguintes valores:

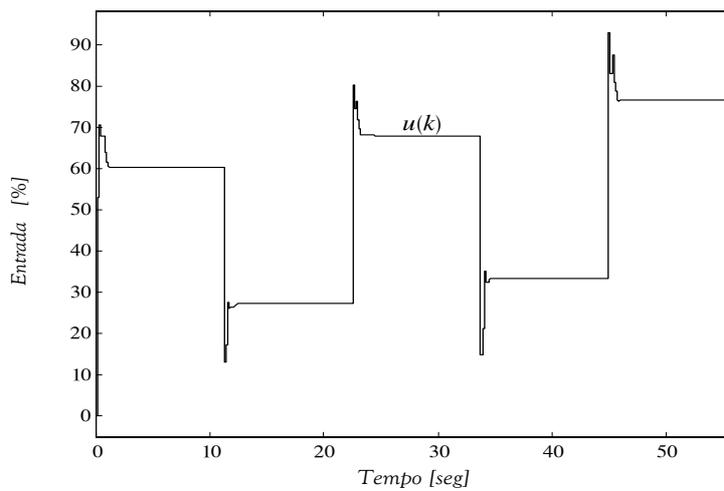
- $A_o$ : valores próprios = 0.7, de que resulta  $L = \begin{bmatrix} -0.4075 \\ -0.9046 \end{bmatrix}$
- $Q_o = 0.3 I_2$ , de que resulta  $P = \begin{bmatrix} 0.5441 & -0.0020 \\ -0.0020 & 0.5443 \end{bmatrix}$
- $\Lambda$ : valores próprios = 0.75, de que resulta  $K = [ -0.0535 \quad 0.7827 ]$

Uma vez obtido, este controlador não linear de parâmetros fixos foi implementado ao processo laboratorial PT326. Como observação principal, aos resultados apresentados na Figura 5.1-9(a), destaca-se o comportamento bastante satisfatório em termos dinâmicos do sistema de controlo ( $y_n$ ). No entanto, uma vez que se considera o modelo

neuronal para projectar o controlador e não é utilizada informação (realimentação) do processo real, não é possível anular os erros de controlo existentes em regime final, o que é compreensível.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).



(b)- Acção de controlo, ( $u$ ).

Figura 5.1-9: Controlador RSRN de parâmetros fixos.

Da experiência realizada torna-se evidente a necessidade de compensar o erro em regime final. A adaptação é uma das possíveis formas de o conseguir, aspecto que a seguir se investiga.

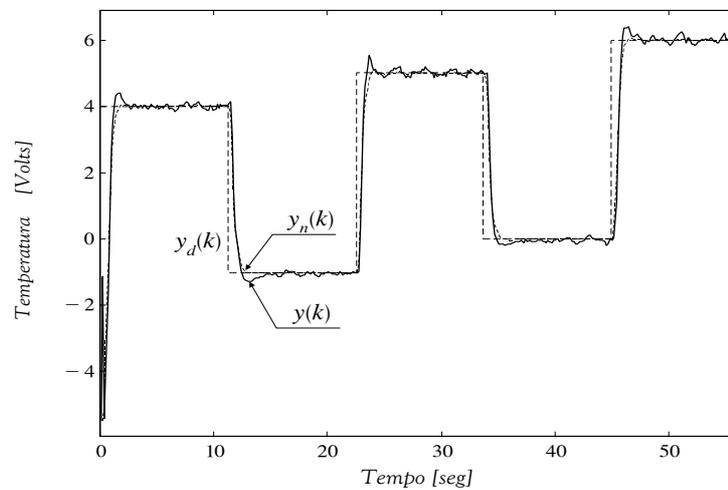
## Controlador RSRN Adaptativo

### Aspectos de Identificação: influência do ganho de Adaptação

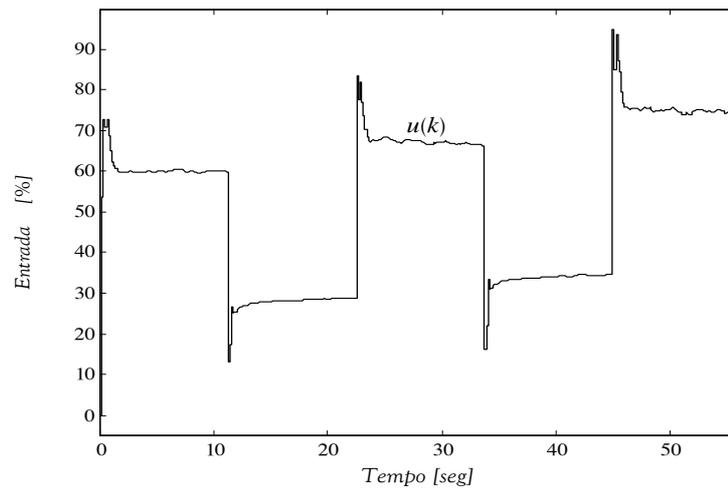
Investiga-se neste segundo grupo de experiências a influência da especificação de parâmetros na etapa de identificação, concretamente do valor da matriz  $Q_o$ , que permite especificar a velocidade de adaptação dos parâmetros do modelo neuronal. Consideram-se dois valores, o primeiro entendido como um valor pequeno e o segundo como um valor elevado. Considerando  $Q_o = 0.05 \times I_2$ , obtém-se como ganho de adaptação

$$P = \begin{bmatrix} 0.0907 & -0.003 \\ -0.0003 & 0.0907 \end{bmatrix}$$

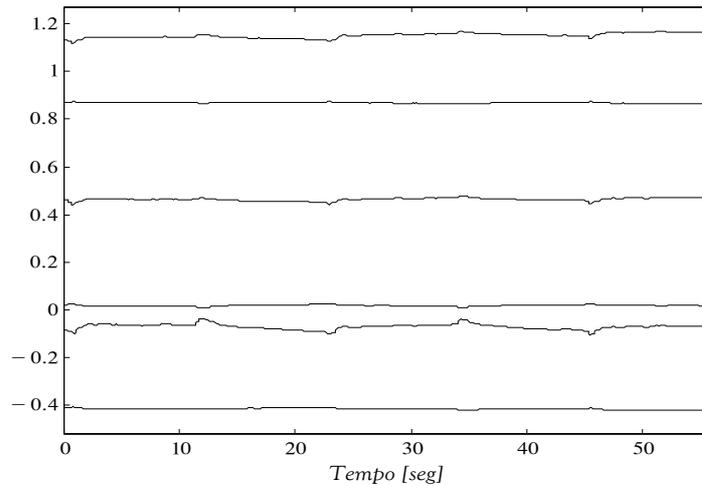
e o resultado mostra-se na Figura 5.1-10.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).



(b)- Ação de controlo, ( $u$ ).



(c)- Parâmetros do modelo neuronal.

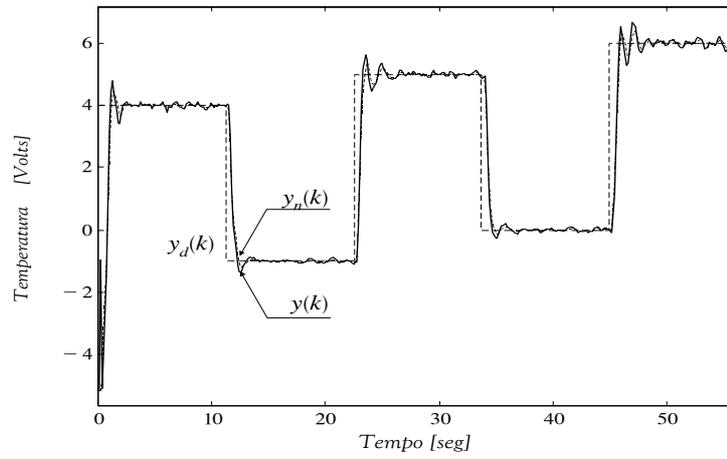
Figura 5.1-10: Controlador adaptativo RSRN;  $Q_o = 0.05 \times I_2$ ,  $\Lambda = 0.75 I_2$ .

Da evolução dos parâmetros do modelo neuronal (Figura 5.1-10c) é possível observar a reduzida capacidade de adaptação expressa pela velocidade (lenta) segundo a qual os parâmetros são ajustados. No entanto, devido à relativa precisão do modelo neuronal inicial é possível por parte do procedimento de estimação anular o erro de estimação, diferença entre a saída do processo real,  $y(k)$ , e do modelo neuronal,  $y_n(k)$ . Depois, o controlador cumpre a sua parte, anula o erro de regulação, diferença entre a saída desejada  $y_d(k)$  e a do modelo neuronal  $y_n(k)$ . A combinação dos dois permite alcançar o objectivo para o qual o sistema de controlo é projectado, ou seja, anular a diferença entre a saída desejada e a saída do processo real.

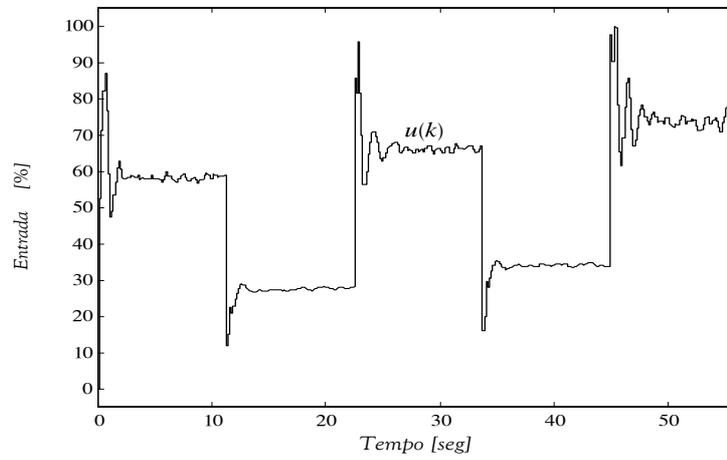
Conforme observado na secção 4.2, apesar de teoricamente o procedimento de estimação ser estável, na prática a realidade é outra. Por outras palavras, se teoricamente o aumento da capacidade de estimação, expressa pelo aumento dos valores da matriz  $Q_o$ , conduz a uma diminuição mais rápida do erro de estimação, na prática esse facto pode não se verificar. Considera-se para o efeito do valor de  $Q_o = 2 \times I_2$ , a que corresponde

$$P = \begin{bmatrix} 3.6273 & -0.0137 \\ -0.0137 & 3.6284 \end{bmatrix}$$

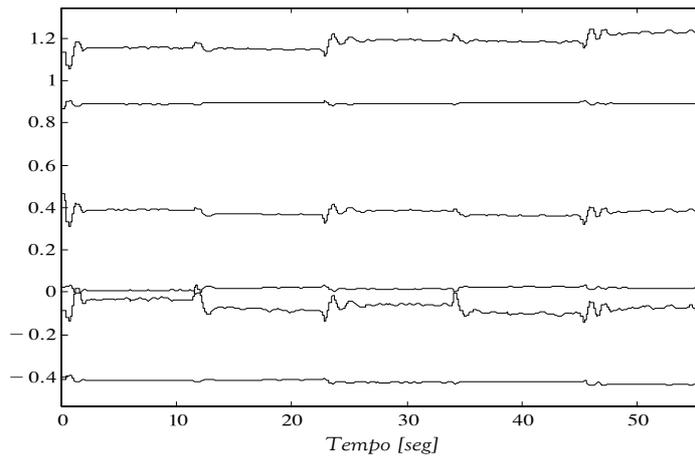
obtendo-se o resultado apresentado da Figura 5.1-11.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).



(b)- Acção de controlo, ( $u$ ).



(c)- Parâmetros do modelo neuronal.

Figura 5.1-11: Controlo RSRN adaptativo;  $Q_o = 2 \times I_2$ ,  $\Lambda = 0.75 I_2$ .

Relativamente à experiência anterior, (Figura 5.1-10), é visível a influência do aumento do valor da matriz  $Q_o$  de  $0.05 \times I_2$  para  $2 \times I_2$ . Como se pode observar na Figura 5.1-11(c) especifica-se neste último caso uma maior capacidade de adaptação, resultando daqui que os parâmetros do modelo neuronal atingem mais rapidamente o seu valor final.

Verifica-se no entanto, que na prática existe um limite máximo para o valor de  $Q_o$ , a partir do qual a saída do sistema se torna demasiado oscilatória. Por outras palavras, o aumento do valor  $Q_o$  permite conduzir teoricamente a uma melhoria do desempenho do sistema global de controlo, considerando o erro de controlo como a soma de dois erros independentes (identificação+regulação). Contudo, quando combinado com o controlador tal não se verifica, uma vez que não podem ser considerados procedimentos independentes.

Consequentemente, em termos práticos, o valor de  $Q_o$  deve ser especificado de forma a estabelecer um compromisso entre a capacidade de identificação e o comportamento transitório do sistema de controlo. No caso de estudo em causa, e após algumas experiências, optou-se pelo valor de  $Q_o = 0.2 \times I_n$ , valor a ser usado nas experiências seguintes.

Na Tabela 5.1-2 compara-se o desempenho do sistema de controlo em cada uma das situações anteriores. Como critérios são utilizados a média do somatório do valor quadrático do erro de estimação  $SSIE^3$ , a média do somatório do valor quadrático do erro de controlo (modelo neuronal)  $SSCE^4$  e a média do somatório da variação da acção de controlo  $SSCC^5$ , definidos respectivamente por (5.1-6), (5.1-7) e (5.1-8).

$$\frac{1}{N} \sum_k^N [y(k) - y_n(k)]^2 = \frac{1}{N} \sum_k^N e_y(k)^2 \quad (5.1-6)$$

$$\frac{1}{N} \sum_k^N [y_d(k) - y_n(k)]^2 = \frac{1}{N} \sum_k^N e_n(k)^2 \quad (5.1-7)$$

$$\frac{1}{N} \sum_k^{N-1} [u(k+1) - u(k)]^2 = \frac{1}{N} \sum_k^{N-1} \Delta u(k)^2 \quad (5.1-8)$$

---

<sup>3</sup> SSIE- Sum of Squared Identification Error

<sup>4</sup> SSCE- Sum of Squared Control Error

<sup>5</sup> SSCC- Sum of Squared Change of Control

Crítério	$Q_o = 0.05 \times I_2$	$Q_o = 2 \times I_2$
$\Sigma e_y^2$	0.1018	0.0733
$\Sigma e_n^2$	2.1864	2.1276
$\Sigma \Delta u^2$	53.2432	62.4771

Tabela 5.1-2: Desempenho do controlador RSRN para diferentes valores de  $Q_o$ .

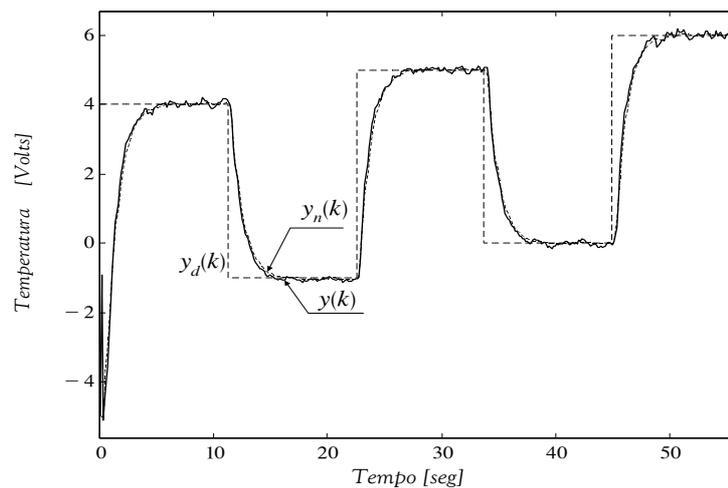
Da sua análise comprovam-se os comentários efectuados. O aumento dos valores da matriz  $Q_o$  permite, se correctamente especificados, diminuir o erro de estimação, o que conduzirá a um melhor comportamento do sistema de controlo, expresso pelo critério do erro de regulação,  $\Sigma e_n^2$ . Na Tabela 5.1-2, é ainda visível o esforço de actuação correspondente à segunda situação, facto perfeitamente visível e Figura 5.1-11(b).

#### Aspectos de Identificação: influência do ganho K

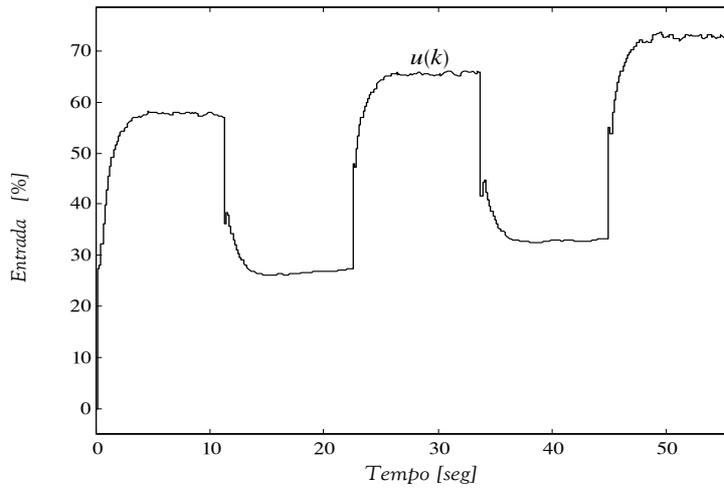
O regulador caracteriza-se por implementar um algoritmo de localização de valores próprios, cuja escolha influenciará o desempenho do sistema de controlo. Analisa-se de seguida essa influência, à semelhança do procedimento de identificação. Começa por se considerar  $\Lambda = 0.95 I_2$ , de que resulta

$$K = [ -0.0304 \quad 0.1945 ]$$

que se considera um valor pequeno. Na Figura 5.1-12 mostra-se o desempenho do sistema de controlo nesta situação.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).

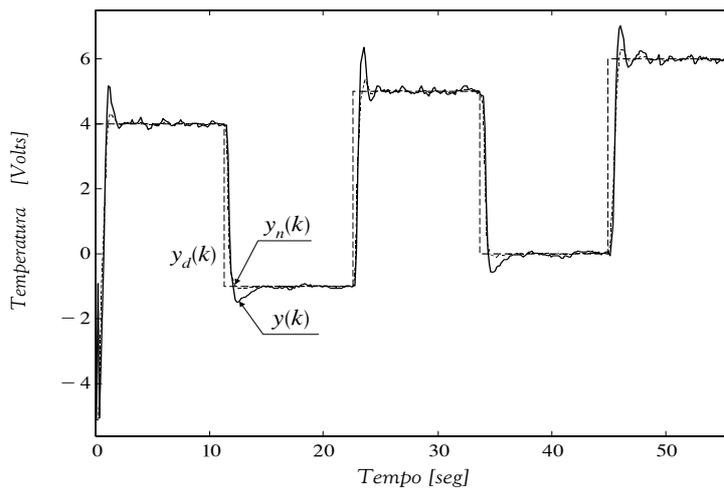
(b)- Acção de controlo, ( $u$ ).Figura 5.1-12: Controlo adaptativo RSRN;  $Q_0 = 0.20 I_2$ ,  $\Lambda = 0.75 I_2$ .

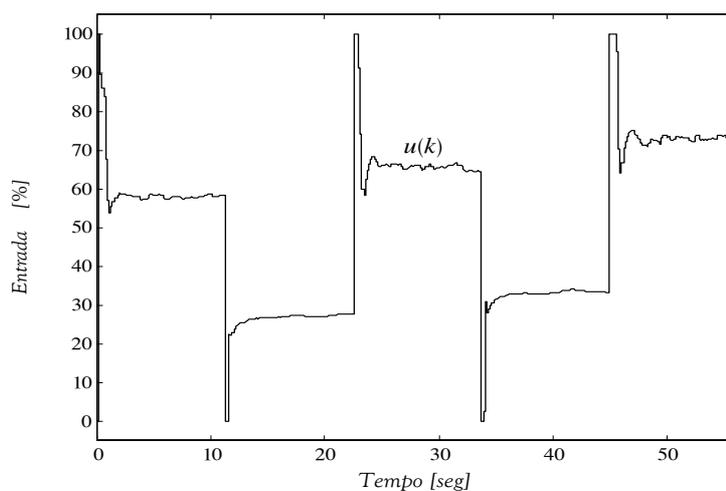
Como é possível observar, o comportamento do sistema de controlo é lento (comparado com as experiências anteriores), devido à especificação efectuada para os dos valores próprios. A saída do sistema é criticamente amortecida não apresentando qualquer sobre-elevação.

A seguir considera-se  $\Lambda = 0.5 I_2$ , de que resulta um valor considerado relativamente elevado para  $K$ , concretamente

$$K = [ -0.1519 \quad 1.2959 ]$$

Na Figura 5.1-13 mostra-se a evolução do sistema de controlo nesta situação.

(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).

(b)- Acção de controlo, ( $u$ ).Figura 5.1-13: Controlo adaptativo RSRN;  $Q_0 = 0.20$ ,  $\Lambda = 0.5 I_2$ .

Como observado na Figura 5.1-13, força-se o regulador a determinar acções de controlo mais bruscas, no sentido de conduzir mais rapidamente o valor da saída ao valor desejado, isto é, aumenta-se a capacidade do sistema de controlo em anular o erro de regulação.

Tal como nas experiências relativas ao processo de identificação, na prática este valor não poderá ser demasiado elevado, pois se assim fosse correr-se-ia o risco de conduzir o sistema global de controlo a ser demasiado oscilatório. Na Tabela 5.1-3 quantificam-se os resultados das experiências efectuadas.

Critério	$\Lambda = 0.95 I_2$	$\Lambda = 0.5 I_2$
$\Sigma e_y^2$	0.0577	0.1213
$\Sigma e_n^2$	3.2170	1.8349
$\Sigma \Delta u^2$	21.4641	94.8975

Tabela 5.1-3: Desempenho do controlador RSRN para diferentes valores de  $K$ .

Como se pode observar o aumento do valor de  $K$  (até valores aceitáveis) permite diminuir o erro de controlo, aumentando consideravelmente o esforço por parte do actuador. Como é também visível o erro de estimação sofre um aumento, facto que se justifica pelo facto dos procedimentos de identificação e controlo não serem independentes, isto é, no primeiro caso o controlador força a saída do processo a ser lenta, o que permite ao procedimento de identificação ser mais preciso. No segundo caso, a obtenção de dinâmica mais rápida, introduz dificuldades adicionais ao procedimento de identificação. Ou seja, verifica-se o velho dilema de um controlador adaptativo, condição

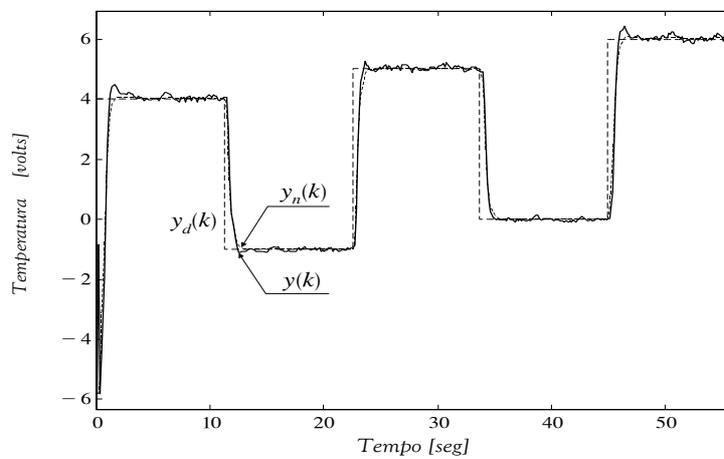
essencial para que este seja aplicável com sucesso: o procedimento de identificação deve ser mais rápido que a dinâmica do processo.

Finaliza-se este grupo de experiências considerando os valores de  $Q_o = 0.2 I_2$  e  $\Lambda = 0.70 I_2$  e os correspondentes valores para  $L$  e  $K$ .

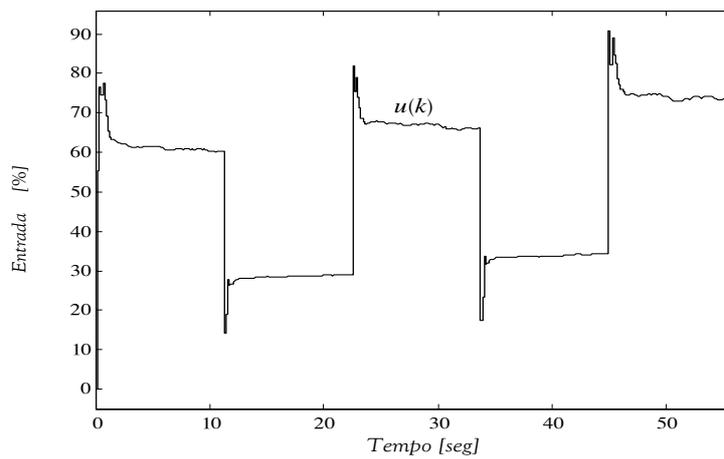
$$L = \begin{bmatrix} -0.4075 \\ -0.9046 \end{bmatrix}$$

$$K = [ -0.0400 \quad 0.6603 ]$$

O resultado mostra-se na Figura 5.1-14.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).



(b)- Ação de controle, ( $u$ ).

Figura 5.1-14: Controlo adaptativo RSRN;  $Q_o = 0.2 \times I_2$  e  $\Lambda = 0.7 \times I_2$ .

Na Tabela 5.1-4 compara-se o desempenho do sistema de controlo para todas as situações verificadas nas três experiências: controlador não adaptativo e adaptativo, considerando variações nos valores de  $Q_o$  e nos valores de  $\Lambda$ .

Critério	Fixo	$\Lambda = 0.75 I_2$		$Q_o = 0.2 I_2$		$Q_o = 0.2$
		$Q_o = 0.05 I_2$	$Q_o = 2 I_2$	$\Lambda = 0.95 I_2$	$\Lambda = 0.5 I_2$	$\Lambda = 0.75$
$\Sigma e_y^2$	0.1112	0.1018	0.0733	0.0577	0.1213	0.0866
$\Sigma e_n^2$	2.0178	2.1864	2.1276	3.2170	1.8349	2.0942
$\Sigma \Delta u^2$	55.9509	53.2432	62.4771	21.4641	94.8975	55.3725

Tabela 5.1-4: Desempenho do controlador RSRN para as experiências realizadas.

Da análise de todas as experiências constata-se que aquela que permite alcançar um menor erro de controlo é a que é conseguida para valores elevados do parâmetro  $K$ , o que era de esperar, pois é o parâmetro que permite especificar tal objectivo. Ao valor máximo de erro de controlo, relativa à situação mais desfavorável, corresponde aquela que se verifica quando é especificada uma resposta demasiado lenta (valores pequenos do parâmetro  $K$ ). Como consequência, as duas situações anteriores (ganho  $K$  elevado e pequeno) conduzem, respectivamente, ao maior e menor esforço de actuação.

Em relação à identificação referem-se como situações extremas os casos em que se considera  $Q_o$  elevado ou  $K$  pequeno, que permitem obter valores reduzidos para o erro de estimação e o caso em que não é efectuada qualquer adaptação, ou quando  $K$  é elevado, que originam erros de estimação elevados.

Concluindo, este grupo de experiências foi conduzido principalmente com objectivo de investigar a sintonização do controlador neuronal, e de que forma alguns dos seus parâmetros influenciam o sistema global de controlo. Nas experiências que se seguem analisa-se o comportamento da metodologia proposta, RSRN, quando o processo é sujeito a situações imprevistas, como é o caso de perturbações.

## 5.1.4 Variação da Dinâmica

Foram realizadas experiências de forma a avaliar o desempenho do controlador neuronal adaptativo em relação a variações no regime de operação, perturbações na carga e variações na dinâmica do processo. Além disso, de forma a concluir da eficácia da metodologia proposta, foi efectuada a sua comparação com um controlador convencional  $PI$ , justificando-se esta escolha pela razão principal de que os controladores  $PI$  são os que maior aplicação tem em ambientes industriais.

Um controlador *PI* caracteriza-se, no domínio contínuo, pela equação (5.1-9).

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int e(t) \right] \quad (5.1-9)$$

No domínio discreto um controlador *PI* é definido por (5.1-10),

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) \quad (5.1-10)$$

em que os parâmetros usados em (5.1-10), podem ser relacionados com os de (5.1-9) (Jacquot, 1995) por (5.1-11), através de uma integração trapezoidal para o termo integral e uma equação de diferenças para o termo derivativo, sendo  $T$  o intervalo de amostragem.

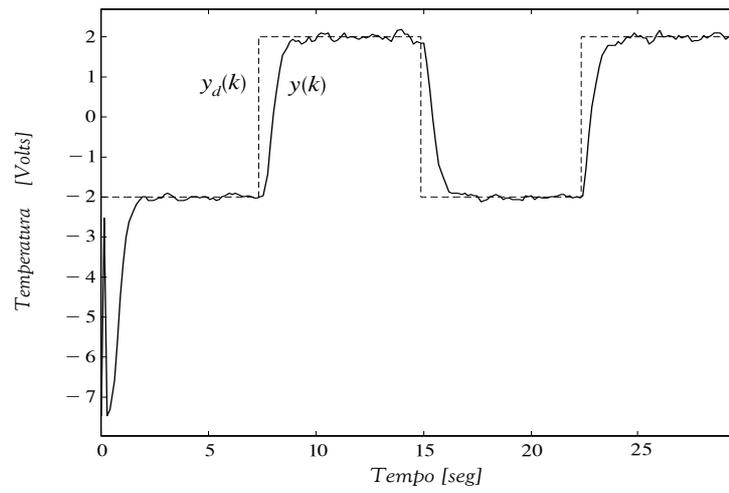
$$q_0 = K_p + \frac{K_i T}{2} \quad (5.1-11a)$$

$$q_1 = \frac{K_i T}{2} - K_p \quad (5.1-11b)$$

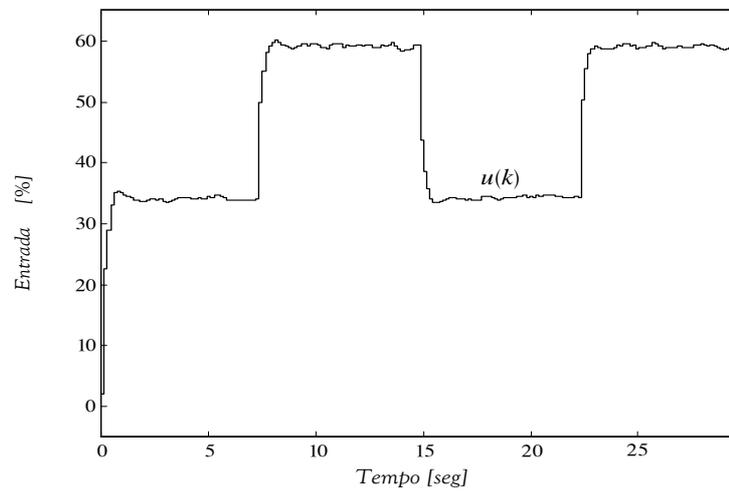
### Inicialização de Parâmetros

Uma vez que as experiências envolvem comparações tentou-se, na medida do possível, e de forma a que estas fossem justas, estabelecer condições semelhantes para a realização das experiências. Assim, os controladores *PI* e *RSRN* foram sintonizados de forma a que a sua resposta fosse semelhante nas mesmas condições. O controlador *PI* foi inicialmente sintonizado usando as regras de Ziegler-Nichols (método em malha aberta), e posteriormente ajustados manualmente os seus parâmetros. A referência consiste numa onda quadrada, cujos valores de temperatura desejada são alternados entre valores de tensão correspondentes a 2 e  $-2$  Volts. Para o controlador *PI* foram obtidos os parâmetros  $q_0 = 0.40$  e  $q_1 = -0.22$ , que permitem obter o desempenho mostrado na Figura 5.1-15.

Quanto aos parâmetros do controlador *RSRN* foram consideradas as seguintes especificações:  $A_o = 0.8 I_2$ ,  $Q_o = 0.2 I_2$  e  $\Lambda = 0.72 I_2$ . Na Figura 5.1-16 mostra-se o desempenho do sistema de controlo neuronal.

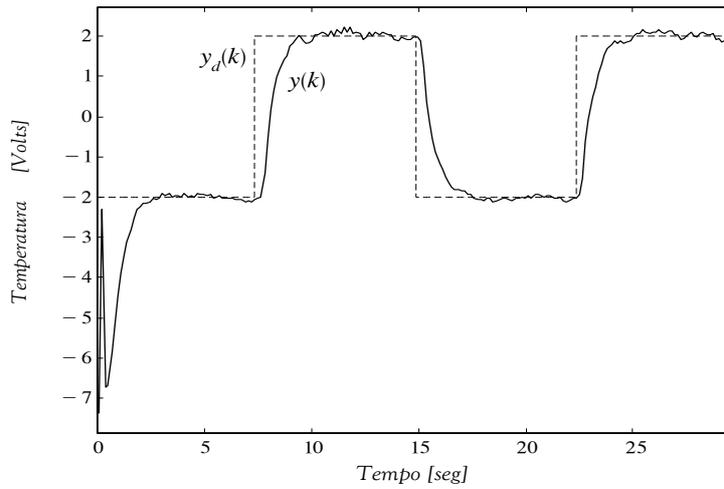


(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).

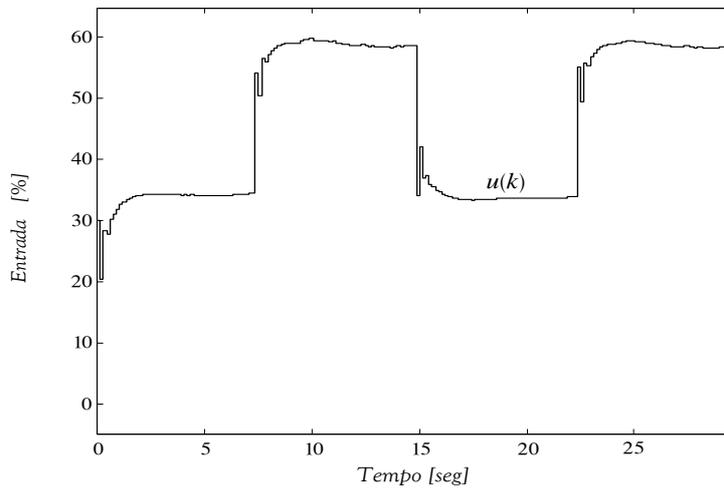


(b)- Ação de controle, ( $u$ ).

Figura 5.1-15: Controlador PI aplicado ao processo PT326.



(a)- Saída desejada ( $y_d$ ), do processo PT326 ( $y$ ) e do modelo neuronal ( $y_n$ ).



(b)- Acção de controlo, ( $u$ ).

Figura 5.1-16: Controlador RSRN aplicado ao processo PT326.

Em termos quantitativos, como pode ser observado na Tabela 5.1-5, comprova-se a semelhança do desempenho dos controladores, se for tido em conta a média do somatório do valor quadrático do erro de controlo (SSCE)  $\Sigma e_c^2$ , definido para o processo real.

$$\frac{1}{N} \sum_k^N [y_d(k) - y(k)]^2 = \frac{1}{N} \sum_k^N e_c(k)^2 \quad (5.1-12)$$

Contudo, como pode ser observado pela Figura 5.1-16 existe por parte do regulador neuronal uma acção de controlo ligeiramente mais brusca, a verificar-se principalmente na fase de transição de referência, reflectido no critério  $\Sigma \Delta u^2$ .

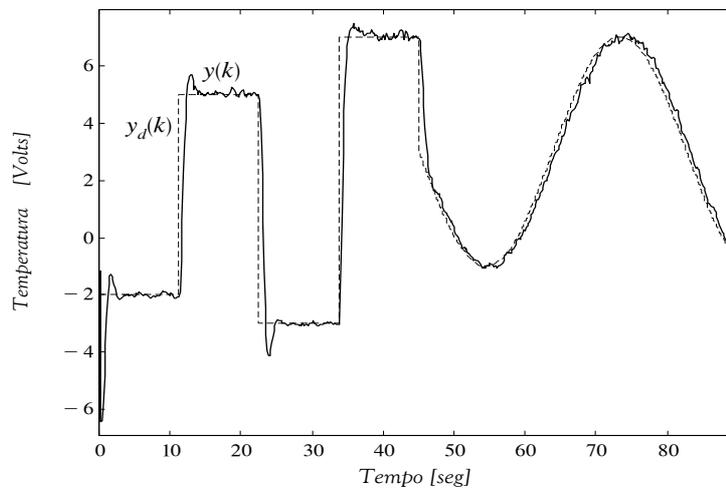
Critério	PI	RSRN
$\Sigma e_c^2$	1.8025	1.7750
$\Sigma \Delta u^2$	24.1246	27.3521

Tabela 5.1-5: Comparação do desempenho dos controladores PI e RSRN.

### Perseguição – Modificação do regime de operação

Considera-se nesta primeira experiência a saída desejada como mostrada na Figura 5.1-17(a), considerando portanto valores de referência diferentes daqueles para os quais os controladores foram inicialmente sintonizados (Figura 5.1-15 e Figura 5.1-16).

Na Figura 5.1-17 e Figura 5.1-18 apresenta-se o comportamento de cada um dos controladores, PI e RSRN.



(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).

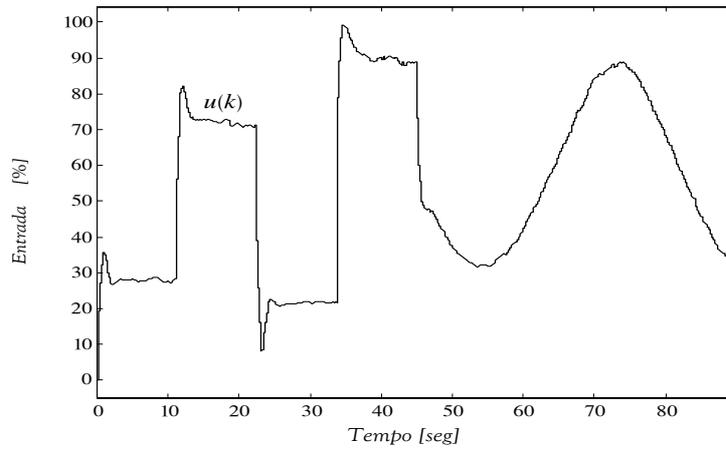
(b)- Acção de controlo, ( $u$ ).

Figura 5.1-17: Controlador PI aplicado ao processo PT326.

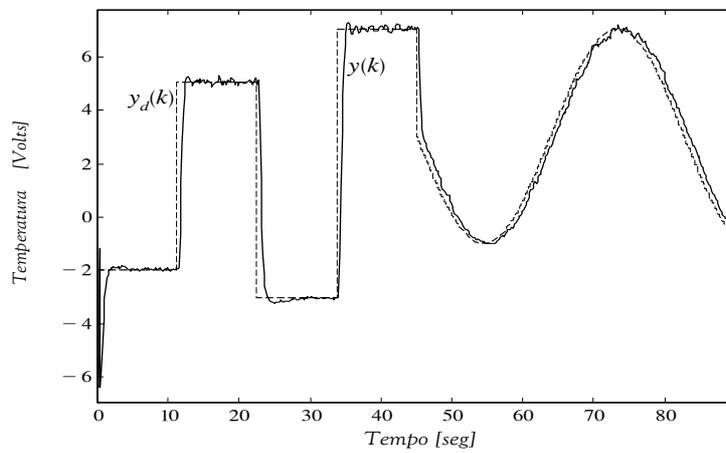
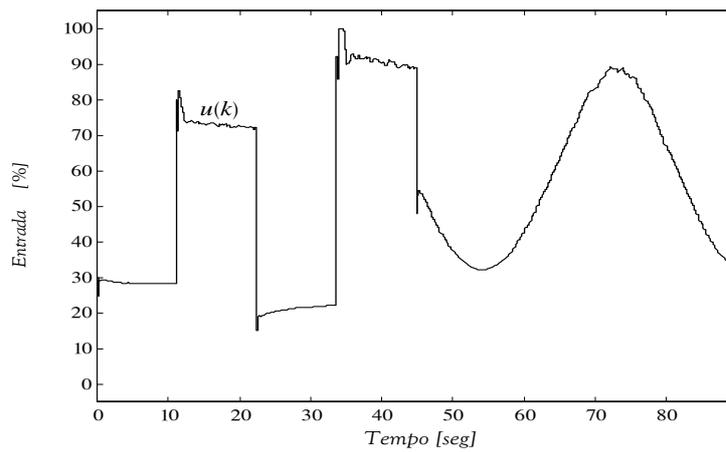
(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).(b)- Acção de controlo, ( $u$ ).

Figura 5.1-18: Controlador RSRN aplicado ao processo PT326.

Da observação das experiências realizadas, e em termos genéricos, pode-se dizer que ambos proporcionam um desempenho aceitável, permitindo que a saída atinja o valor desejado. Na Tabela 5.1-6 quantifica-se o desempenho de cada um dos controladores.

Critério	PI	RSRN
$\Sigma e_c^2$	1.5185	1.4094
$\Sigma \Delta u^2$	10.8722	21.7684

Tabela 5.1-6: Comparação dos controladores PI e RSRN; variação de referência.

Analisando mais em pormenor a Tabela 5.1-6 pode-se concluir que o regulador permite obter, neste caso particular, um desempenho ligeiramente superior. Este facto é na verdade visível na Figura 5.1-17(a) e Figura 5.1-18(a), principalmente na fase de transição do valor da referência. Por um lado o controlador PI, anteriormente sintonizado para outro regime de operação tem algumas dificuldades, originando uma resposta caracterizada pela existência de sobre-elevação.

No entanto, como pode ser observado, quer na Figura 5.1-18(b), quer na Tabela 5.1-6, existe um esforço suplementar por parte do controlador neuronal de forma a compensar as não linearidades (expressas fundamentalmente pela variação do ganho do processo  $G$ ) e originar um comportamento sem sobre-elevação na fase transitória.

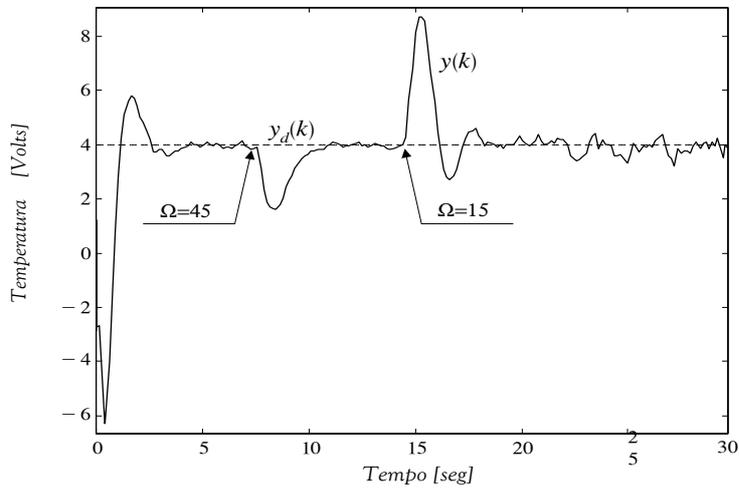
#### Perturbações – Variação do Caudal (Carga)

Analisa-se agora a capacidade de rejeição de perturbações de cada um dos sistemas de controlo PI e RSRN, no caso de se efectuar uma variação na abertura que controla o fluxo de ar,  $\Omega$ . Testa-se desta forma a robustez do controlador quando é alterado tanto o ganho do processo  $G$  como a sua dinâmica, esta última expressa pela variação do tempo de atraso  $T_d$ .

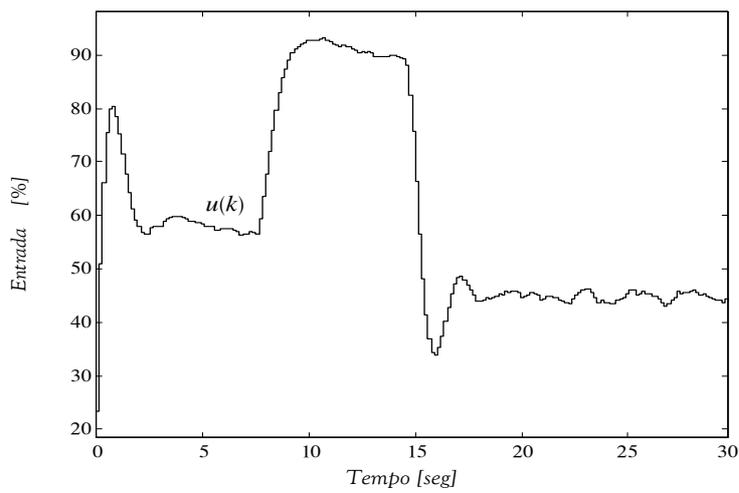
O valor desejado para a temperatura do ar ( $y_d$ ) foi considerado constante, correspondendo a um valor de tensão igual a 4 Volts. Assim, partindo da posição inicial  $\Omega = 30^\circ$ , variou-se a admissão do ar sensivelmente no instante  $t = 7.5$  segundos para  $\Omega = 45^\circ$  e no instante  $t = 15$  segundos para  $\Omega = 15^\circ$ . Introduzem-se assim simultaneamente na dinâmica do processo duas alterações, uma no ganho do processo ( $G$ ) e outra no tempo de atraso ( $T_d$ ).

De referir que se efectua desta forma uma perturbação de 50% na abertura do ar,  $\Omega = 30^\circ \pm 15^\circ$ , a que corresponde uma especificação no tempo de atraso de aproximadamente  $T_d = 0.11$  segundos para a primeira variação e de  $T_d = 0.21$  segundos para a segunda.

Na Figura 5.1-19 e Figura 5.1-20 mostra-se o comportamento do processo quando implementados cada um dos controladores, quantificando-se o seu desempenho na Tabela 5.1-7.



(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).



(b)- Acção de controlo, ( $u$ ).

Figura 5.1-19: Controlador PI; variação do caudal de ar.

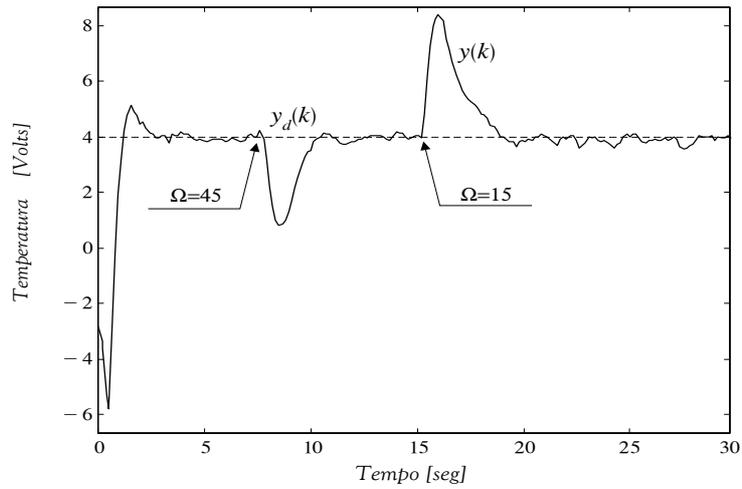
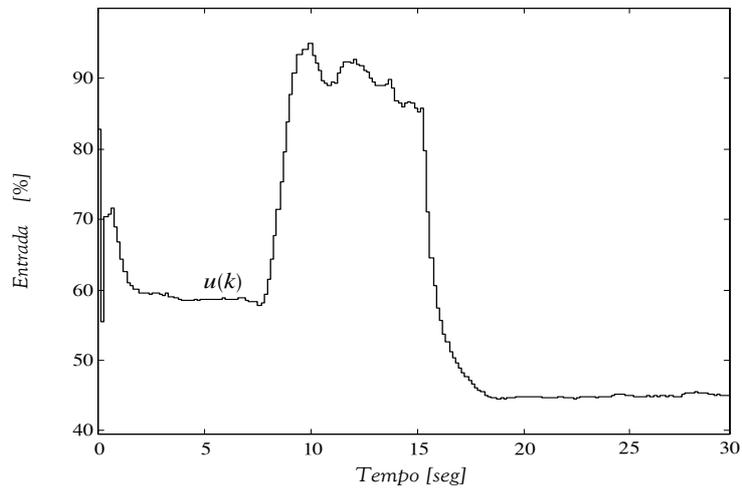
(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).(b)- Ação de controle, ( $u$ ).

Figura 5.1-20: Controlador RSRN; variação do caudal de ar.

Critério	PI	RSRN
$\Sigma e_c^2$	2.2589	2.23455
$\Sigma \Delta u^2$	8.7621	6.7125

Tabela 5.1-7: Comparação dos controladores PI e RSRN; variação do caudal de ar.

Verifica-se que ambos permitem ultrapassar satisfatoriamente a perturbação introduzida. No entanto, o controlador PI tem algumas dificuldades em controlador

adequadamente o processo a partir do instante  $t = 15$  segundos, que se reflecte, se bem que de uma forma pouco clara, no critério de erro verificado. No entanto, este facto é perfeitamente visível na Figura 5.1-19 e no critério  $\Sigma \Delta u^2$  da Tabela 5.1-7, salientando-se a dificuldade de estabilização por parte da acção de controlo e consequentemente do valor de saída.

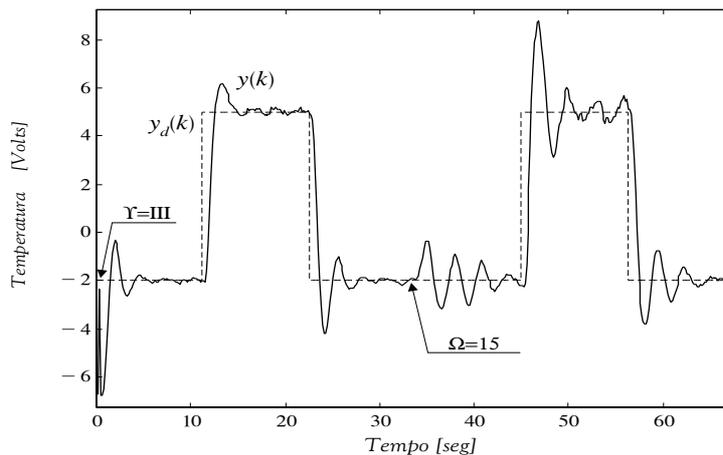
Este facto justifica-se uma vez que a variação (diminuição) na admissão de ar conduz a um aumento do tempo de atraso, para um valor diferente daquele para o qual o PI tinha sido inicialmente sintonizado. O regulador, pela sua capacidade de adaptação, mostra-se mais adequado a esta situação, permitindo captar a nova dinâmica do processo.

### Varição na Dinâmica, Alteração do Tempo de atraso

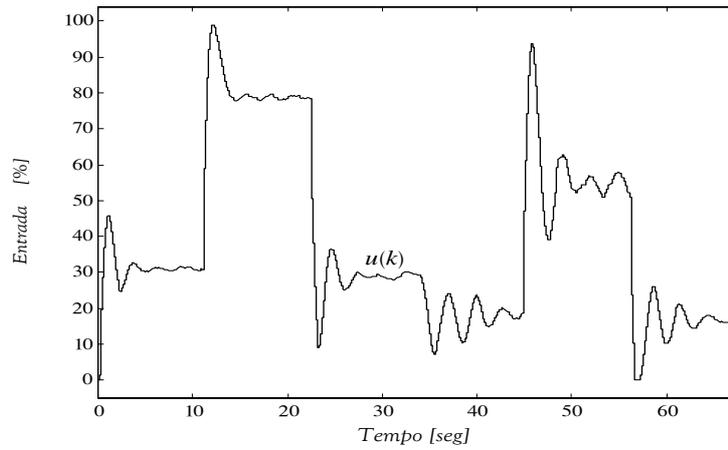
Uma das formas de testar a robustez de um sistema de controlo consiste na análise do seu comportamento quando a dinâmica do processo a controlar é alterada. No caso particular do processo PT326, o sensor originalmente colocado na posição  $\Upsilon = \text{II}$ , é alterado para a posição  $\Upsilon = \text{III}$ . Desta deslocação resulta um aumento do tempo de atraso, o que provoca uma alteração na dinâmica do processo. Além disso, de forma a aumentar ainda mais o tempo de atraso, efectua-se uma diminuição na abertura do ar (inicialmente a  $\Omega = 30^\circ$ ) para  $\Omega = 15^\circ$ , que influenciará por sua vez também o ganho do processo.

Em termos práticos, para uma abertura de  $\Omega = 30^\circ$  e o sensor na posição  $\Upsilon = \text{III}$  corresponde um tempo de atraso de cerca  $T_d = 0.25$  segundos; para uma abertura de  $\Omega = 15^\circ$  e o sensor na mesma posição corresponde um tempo de atraso de aproximadamente  $T_d = 0.40$  segundos.

Na Figura 5.1-21 e Figura 5.1-22 mostra-se a evolução da resposta para os dois sistemas de controlo, PI e RSRN.

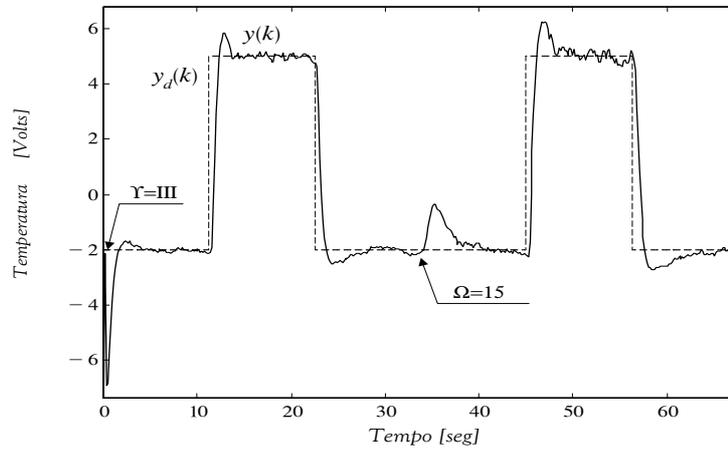


(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).

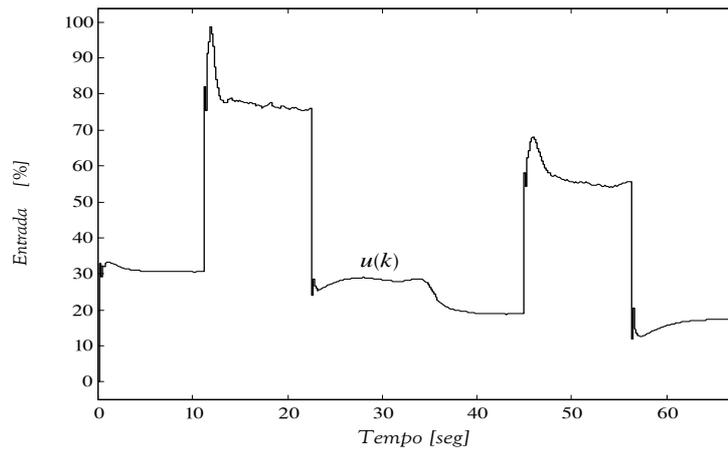


(b)- Acção de controlo, ( $u$ ).

Figura 5.1-21: Controlador PI; variação da dinâmica.



(a)- Saída desejada ( $y_d$ ) e do processo PT326 ( $y$ ).



(b)- Acção de controlo, ( $u$ ).

Figura 5.1-22: Controlador RSRN; variação da dinâmica.

Da observação da Figura 5.1-21 são evidentes as dificuldades do controlador *PI* em controlar adequadamente o processo, sendo o seu desempenho seriamente deteriorado com a introdução simultânea do tempo de atraso e variação da admissão de ar.

Para um valor pequeno do atraso o controlador mostra-se capaz de controlar satisfatoriamente o processo, à semelhança do comportamento já observado na Figura 5.1-20. Contudo, para valores superiores de tempos de atraso a resposta torna-se demasiado oscilatória. Os resultados obtidos comprovam experimentalmente as dificuldades típicas encontradas em processos com tempos de atraso: existe uma sobre-elevação inaceitável, consequência do facto do controlador utilizar informação atrasada, o que faz com que actue incorrectamente.

Observando o comportamento do controlador *RSRN*, Figura 5.1-22, conclui-se que neste caso particular o seu desempenho é superior, aspecto que pode ser quantificado na Tabela 5.1-8.

Critério	PI	RSRN
$\Sigma e_c^2$	2.6834	2.3123
$\Sigma \Delta u^2$	14.7029	24.0551

Tabela 5.1-8: Comparação dos controladores *PI* e *RSRN*; variação da dinâmica.

## 5.1.5 Análise de Resultados

As experiências efectuadas ilustraram as vantagens e desvantagens da utilização da metodologia adaptativa não linear *RSRN* em sistemas de controlo, principalmente quando se sujeita o processo a variações na sua dinâmica.

Começou por se verificar (Figura 5.1-9) que um controlador de ganhos fixos dificilmente anulará o erro em regime final, a menos que o modelo inicial seja demasiado preciso e não se verifiquem quaisquer variações no processo. Uma vez que qualquer um destes requisitos não se verifica, na prática, ultrapassa-se o problema introduzindo um mecanismo de adaptação. Assim, mesmo que o modelo neuronal inicial não seja preciso, a adaptação dos parâmetros do modelo e o consequente projecto do controlador, realizado em cada instante, permite compensar as imprecisões ou efeitos não previstos.

Fez-se posteriormente, numa primeira fase, um estudo acerca da inicialização dos parâmetros relativos aos procedimentos de identificação e de controlo. Num segundo grupo de experiências levou-se a cabo um estudo comparativo entre a metodologia *RSRN* e um controlador convencional *PI*. Consideraram-se algumas situações distintas que incluíram o funcionamento do processo em diferentes regimes de operação, e alteração da dinâmica do processo. Como conclusão, o facto de que o desempenho de qualquer dos controladores é aceitável exceptuando-se o caso de situações em que se introduzem alterações não lineares na dinâmica do processo (introdução de um tempo de atraso, portanto um efeito não linear).

Nestas últimas condições o controlador *PI*, apresenta um desempenho inadequado, que se entende, uma vez que os parâmetros foram obtidos (sintonizados) não tendo em conta a existência de tais tempo de atraso. Por outro lado, a capacidade adaptativa da metodologia de controlo *RSRN* mostrou, neste caso particular, ser mais robusta a variações de dinâmica.

Das experiências efectuadas duas observações principais podem ser efectuadas. Primeiro, que em situações de variação do ganho do processo, exemplo de uma variação linear, o desempenho de ambos os controladores é aceitável. Segundo, para situações mais complexas, como é o caso da variação da carga e de tempos de atraso o controlador neuronal permite obter um desempenho superior.

Refira-se no entanto que quando implementado o regulador, o esforço a que se sujeita o actuador é regra geral maior, que será necessariamente um factor a ter em conta e que à primeira vista poderá ser apontada como uma desvantagem. Contudo, não se pensa que assim seja, uma vez que através de uma especificação adequada dos parâmetros do sistema de controlo, será relativamente simples contornar este problema.

Em conclusão, em sistemas industriais com características semelhantes ao processo laboratorial analisado, e em face dos resultados obtidos, a metodologia de controlo proposta será certamente uma opção a ter em conta.

# 5.2

## Reactor Químico - CSTR

*Nesta secção implementa-se a metodologia neuronal de controlo RSRN na identificação e controlo de um processo químico, nomeadamente um processo CSTR.*

*É objectivo fundamental mostrar a eficácia desta estrutura de controlo aplicada a processos não lineares, de que o CSTR é uma referência.*

### 5.2.1 Introdução

Um exemplo de um processo químico, que pelas suas características tem sido considerado um *benchmark* no estudo de metodologias de controlo é o CSRT<sup>1</sup>. Apesar de descrito por um modelo relativamente simples, é um sistema interessante sob o ponto de vista de controlo. Pelas suas características, principalmente a não linearidade, o processo CSTR é um exemplo típico de um processo para o qual as metodologias de controlo convencional (lineares e fixas) não se comportam adequadamente. Assim, se por um lado é desejável alcançar condições de operação constantes, por outro a presença de não linearidades torna o projecto de um controlador fixo, para regimes de operação distintos, uma tarefa complexa.

Uma vez que a metodologia de controlo RSRN se caracteriza por ser uma metodologia não linear adaptativa, pretende-se com este exemplo evidenciar as suas características apresentando-se, numa primeira fase a sua eficácia em lidar com não linearidades e posteriormente a sua capacidade de adaptação.

São exemplos de trabalhos que aplicam redes neuronais no controlo de processos CSTR os trabalhos de (H-Chen *et al.*, 1995), (Lightbody e Irwin, 1995), (Lightbody e Irwin, 1997), (Kurtz e Henson, 1997) , (Ge *et al.*, 1998).

---

<sup>1</sup> CSRT- Continuous Stirred Tank Reactor

## Descrição do Processo

O processo CSTR consiste basicamente num tanque, onde ocorre uma reacção química irreversível, onde um componente  $A$  reage para formar um outro componente  $B$ . A reacção caracteriza-se por ser exotérmica, podendo o aumento de temperatura ser contrariado pela introdução de um líquido de arrefecimento e desta forma controlar a evolução da reacção. O líquido flui em contra corrente, através de um invólucro ( $I$ ) que reveste o reactor, como se ilustra na Figura 5.2-1.

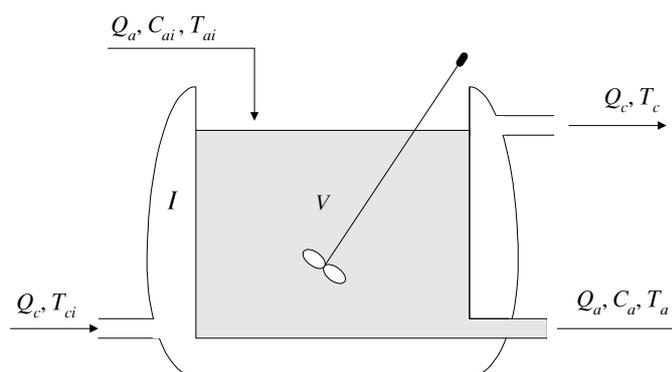


Figura 5.2-1: Tanque de reacção CSTR.

Considere-se um tanque reactor com um volume  $V$ , constante. As variáveis  $Q_a$ ,  $C_{ai}$  e  $T_{ai}$  representam, respectivamente, o caudal volúmico, a concentração e a temperatura de alimentação do componente a concentrar.  $Q_c$  e  $T_{ci}$  definem, por sua vez o caudal e a temperatura do líquido de arrefecimento. À saída do tanque, considera-se a temperatura do líquido de arrefecimento  $T_c$ , e como variáveis de interesse  $C_a$  e  $T_a$ , respectivamente, a concentração e a temperatura do componente concentrado.

O objectivo do sistema de controlo é forçar a concentração,  $C_a$ , do componente, a ter um determinado valor desejado,  $y_d$ , através da manipulação do caudal do líquido de arrefecimento,  $Q_c$ .

## Modelização do Processo

O modelo para o processo CSTR aqui considerado é proposto em (Lightbody e Irwin, 1995), (Lightbody e Irwin, 1997) e (Ge *et al.*, 1998). Basicamente as equações diferenciais, que descrevem o seu comportamento dinâmico, são obtidas a partir da aplicação dos princípios fundamentais de conservação de massa (5.2-1a) e de energia (5.2-

1b), respectivamente para a concentração  $C_a(t)$  e temperatura  $T_a(t)$ , num determinado instante  $t$  (contínuo).

$$\dot{C}_a(t) = \frac{Q_{ai}}{V} (C_{ai} - C_a(t)) - r_a(t) \quad (5.2-1a)$$

$$\dot{T}_a(t) = \frac{Q_{ai}}{V} (T_{ai} - T_a(t)) + k_1 C_a(t) e^{\frac{-E}{RT_a(t)}} + k_2 Q_c(t) \left(1 - e^{\frac{-k_3}{Q_c(t)}}\right) (T_{ci} - T_a(t)) \quad (5.2-1b)$$

Na equação (5.2-1a),  $r_a(t)$  define a velocidade de reacção por unidade de volume do componente A, dada por (5.2-2),

$$r(t) = \kappa C_a(t) \quad (5.2-2)$$

em que  $\kappa$  é o coeficiente de reacção, descrito pela relação de Arrhenius (5.2-3), que permite relacionar a velocidade de reacção  $\kappa$  com a temperatura  $T_a$ .

$$\kappa = k_o e^{\frac{-E}{RT_a(t)}} \quad (5.2-3)$$

$k_o$  é uma constante,  $E$  é a energia de activação e  $R$  a constante dos gases.

Os parâmetros para o modelo considerado apresentam-se na Tabela 5.2-1, (Lightbody e Irwin, 1997), de unidades definidas na Tabela 5.2-2.

Parâmetro	Descrição	Valor nominal
$Q_a$	Caudal de alimentação do componente A	100 l/min
$C_{ai}$	Concentração molar do componente A	1 mol/l
$T_{ai}$	Temperatura de alimentação de A	350 K
$T_{ci}$	Temperatura do líquido de arrefecimento	350 K
$E/R$	Energia de activação	$1 \times 10^4$ K
$V$	Volume do tanque	100 l
$k_o$	Factor pré-exponencial	$7.2 \times 10^{10} \text{ min}^{-1}$
$k_1$	Constante	$1.44 \times 10^{13}$
$k_2$	Constante	$1 \times 10^{-2}$
$k_3$	Constante	$7 \times 10^2$

Tabela 5.2-1: Parâmetros do modelo CSTR.

Notação	Variável	Unidades
$min$	Tempo	<i>minutos</i>
$l$	Volume	<i>litros</i>
$C_a$	Concentração	<i>moles/litro</i>
$K$	Temperatura	<i>graus Kelvin</i>

Tabela 5.2-2: Unidades utilizadas.

Considere-se o seguinte regime permanente, tal como proposto em Lightbody e Irwin, (1997).

Parâmetro	Descrição	Valor nominal
$Q_{co}$	Caudal do líquido de arrefecimento	103.41 <i>l/min</i>
$C_{ao}$	Concentração à saída	0.1 <i>mol/l</i>
$T_{ao}$	Temperatura do componente à saída	438.54 <i>K</i>

Tabela 5.2-3: Regime de operação considerado para o modelo do processo CSTR.

Da análise das equações que descrevem a dinâmica do processo, pode-se concluir que este se caracteriza por ser um processo não linear, fundamentalmente devido à existência dos termos exponenciais. Na Figura 5.2-2(a) constata-se essa relação de não linearidade, considerando o processo em malha aberta e efectuando uma variação em degrau (Figura 5.2-2b) na variável de controlo,  $Q_c$ , caudal de arrefecimento, em torno do valor nominal  $Q_{co}$ . Como é possível observar, a evolução da dinâmica da variável a controlar  $C_a$  depende fundamentalmente do valor do regime de operação em questão.

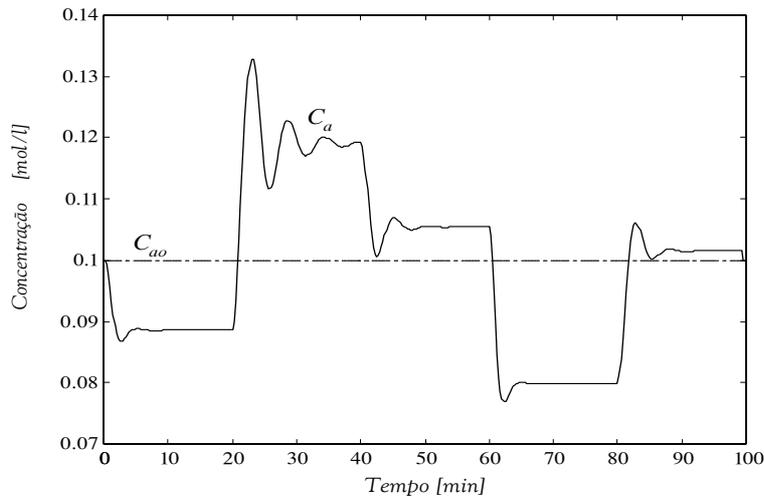
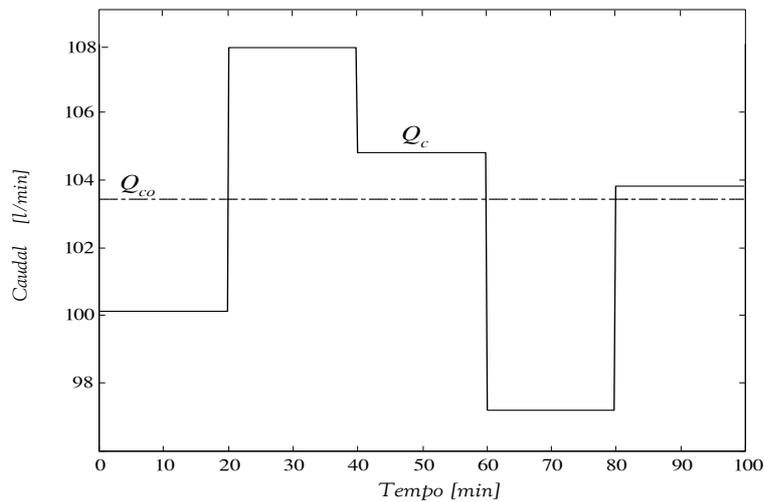
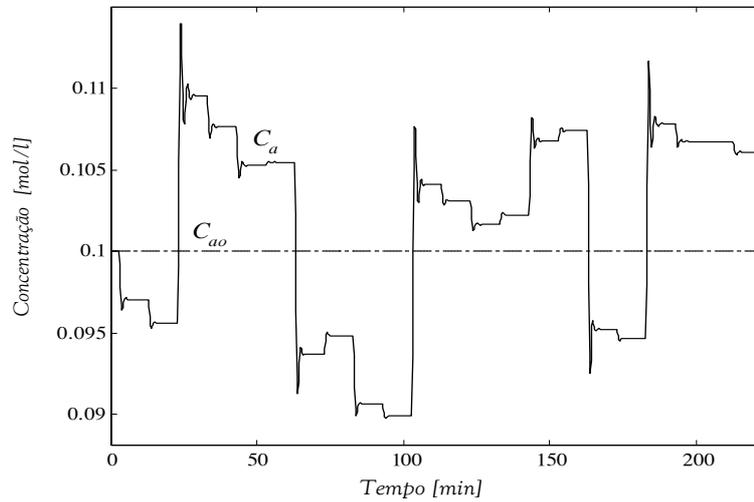
(a) - Saída do processo ( $C_a$ ).(b) - Acção de controlo ( $Q_c$ ).

Figura 5.2-2: Resposta em malha aberta do processo CSTR.

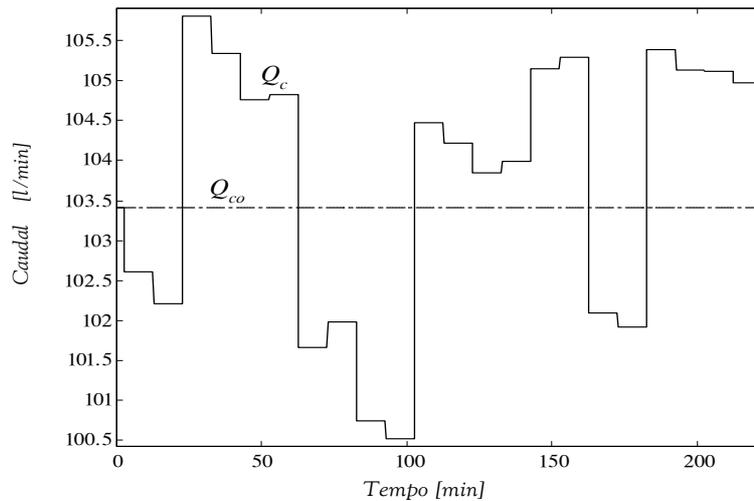
## 5.2.2 Identificação Experimental

A primeira etapa no projecto do controlador neuronal consiste na obtenção do modelo de estados neuronal, obtido por sua vez à custa de um conjunto de dados de identificação. Se para sistemas lineares, e de forma a captar a dinâmica se referem sinais persistentemente excitantes, para sistemas não lineares existe a dificuldade adicional de capturar não linearidades. Isto é, a entrada deve, além de excitar todos os modos do processo, garantir que são consideradas as várias regiões de operação de interesse. Para esse efeito, usa-se um sinal de entrada híbrido, como se mostra na Figura 5.2-3(b). Este

consiste em valores de grande amplitude, permitindo ter em conta o problema da não linearidade uma vez que conduz o processo às várias regiões de operação, e em valores de pequena amplitude (em relação aos valores de grande amplitude), o que permite lidar com várias dinâmicas dentro de cada região de operação.



(a)- Saída do processo ( $C_a$ ).



(b)- Acção de controlo ( $Q_c$ ).

Figura 5.2-3: Sequência de dados de identificação.

Após várias simulações, considerou-se como número de neurónios na camada interna  $n = 4$ , estabelecendo-se assim um compromisso entre capacidade de representação e complexidade. O algoritmo de estimação foi o de Levenberg-Marquardt, tendo sido as matrizes do modelo neuronal inicializadas com base no conhecimento existente de um modelo de estado linear.

Assim, foi numa primeira fase obtido um modelo linear para o regime de operação considerado (Tabela 5.2-1), usando para esse efeito a função *ARX* do Matlab (*Control Toolbox*, 1998). Desta identificação linear resultou um modelo, cujos valores próprios da matriz da dinâmica foram depois usados para inicializar a matriz *A* do modelo neuronal (diagonal), sendo as restantes matrizes *B*, *C* e *D*, inicializadas da seguinte forma: *B* e *C* aleatoriamente segundo uma distribuição normal, no intervalo  $[-0.5, 0.5]$  e  $D = 0.01 \times A$ .

Devido à diferença relativa entre os valores nominais das variáveis,  $Q_c = 103.41$  para a entrada e  $C_a = 0.1$  para a saída, foram utilizadas as seguintes expressões de normalização (5.2-4), respectivamente para a entrada e saída.

$$Q_{cn} = \frac{Q_c - Q_{co}}{4} \quad (5.2-4a)$$

$$C_{an} = 50 (C_a - C_{ao}) \quad (5.2-4b)$$

As variáveis  $Q_{cn}$  e  $C_{an}$  representam, respectivamente, os valores considerados como entradas e saídas utilizadas pela rede neuronal.

Na Figura 5.2-4 compara-se a saída real do processo,  $C_a$ , com os valores de saída do modelo neuronal obtido,  $y_n$ .

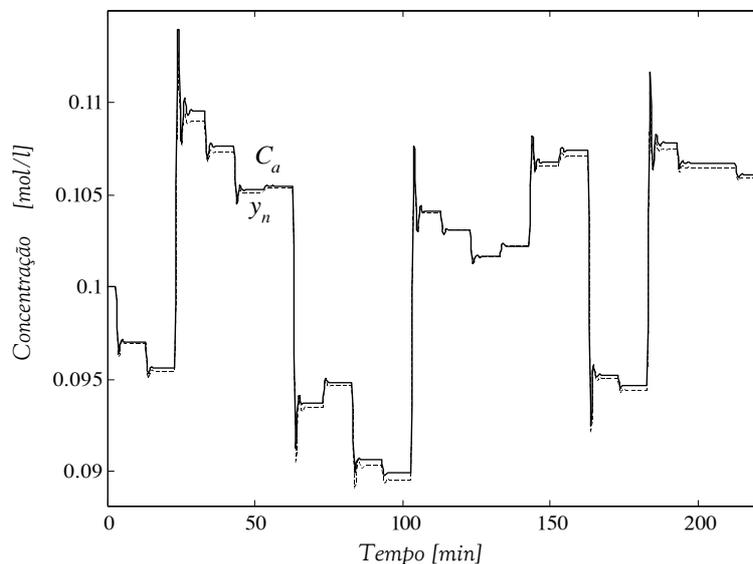


Figura 5.2-4: Identificação do processo CSTR; comparação da saída ( $C_a$ ) com a do modelo neuronal ( $y_n$ ).

Matematicamente o modelo neuronal considerado é definido por (5.2-5), e é caracterizado pelos valores dados em (5.2-6).

$$x_n(k+1) = A x_n(k) + B u(k) + D \sigma(x_n(k)) \quad (5.2-5a)$$

$$y_n(k) = C x_n(k) \quad (5.2-5b)$$

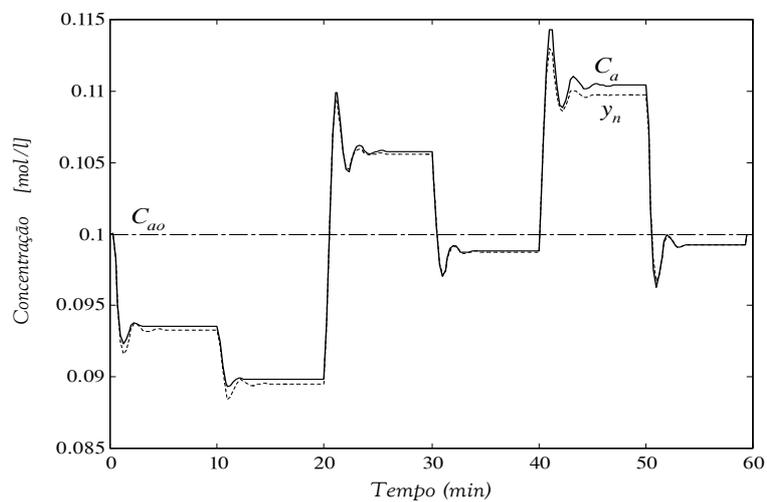
$$A = \begin{bmatrix} 0.4100 & -0.0027 & -0.0030 & 0.0012 \\ -0.0007 & 0.4100 & -0.0013 & -0.0030 \\ 0.0027 & -0.0007 & 0.4100 & -0.0016 \\ 0.0020 & -0.0003 & -0.0004 & 0.4100 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.0319 \\ 0.0829 \\ 0.3899 \\ 0.0802 \end{bmatrix} \quad (5.2-6)$$

$$C = [0.7548 \quad 1.3950 \quad -0.0584 \quad 1.0088]$$

$$D = \begin{bmatrix} 0.4080 & -0.3738 & -0.3183 & 0.4412 \\ -0.4875 & 0.2902 & 0.3424 & -0.4711 \\ 0.2675 & -0.6338 & -0.5746 & 0.9542 \\ -0.5065 & 1.0859 & 0.3430 & -0.0439 \end{bmatrix}$$

De forma a validar o modelo neuronal, foi ainda considerado um conjunto de dados de validação, cujo resultado se mostra na Figura 5.2-5.



(a)- Saída do processo ( $C_a$ ) e do modelo neuronal ( $y_n$ ).

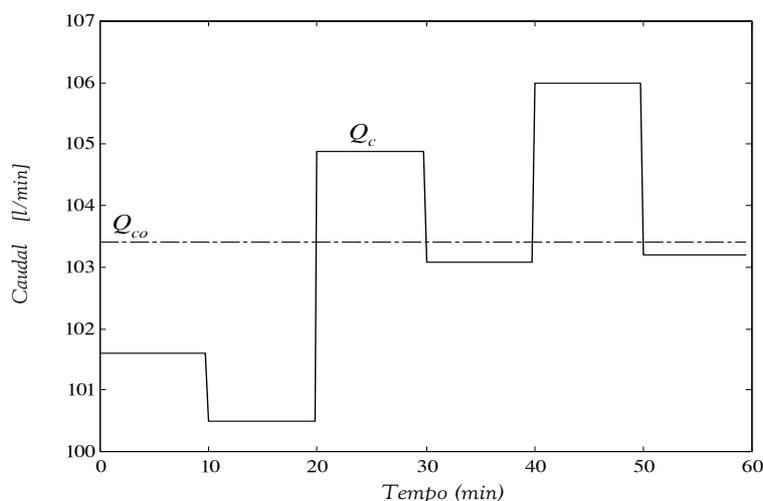
(b)- Ação de controle ( $Q_c$ ).

Figura 5.2-5: Validação do modelo neuronal.

Da identificação e validação efectuadas podem ser feitas duas observações principais.

Primeiro, a existência de erros de modelização, visível tanto na Figura 5.2-4 como na Figura 5.2-5(a). No entanto este facto pode ser justificado, pelo reduzido número de neurónios, ou ainda devido ao método de optimização, que poderá ter encontrado um mínimo local. Note-se que poderiam ter sido efectuados esforços no sentido de obviar cada um destes factores (número de neurónios, inicialização e estimação dos parâmetros), que permitissem obter um modelo neuronal mais preciso. Contudo, esse facto não se justificou, uma vez que um dos objectivos é evidenciar a capacidade de adaptação da metodologia em causa.

Em segundo lugar, apesar do modelo neuronal não ser perfeito, será preferível em termos de controlo a um modelo linear. Ou seja, se for implementado um controlador não linear, que tenha em conta as não linearidades do processo, resultará um sistema de controlo que terá mais probabilidades de sucesso do que um controlador linear.

A seguir são efectuadas simulações com o objectivo de justificar estas ideias. Considera-se para o efeito dois controladores, um *PI*, como exemplo de um controlador linear fixo, e o controlador desenvolvido *RSRN*, também fixo, como exemplo de um controlador não linear. Depois, numa fase posterior, introduz-se na estrutura de controlo neuronal a possibilidade de adaptação dos parâmetros do modelo, obtendo-se a estrutura de controlo adaptativa *RSRN*.

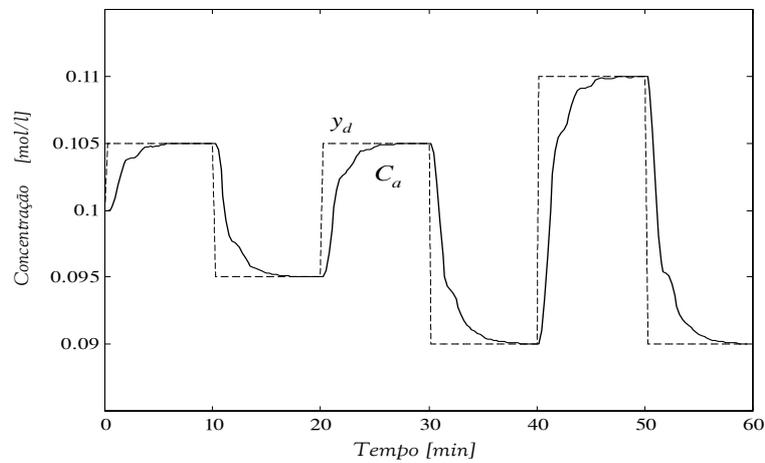
### 5.2.3 Controlador *PI*

Conforme referido na secção anterior, justifica-se a escolha do controlador *PI* como base de comparação uma vez que este é usado na maioria dos sistemas de controlo de processos químicos industriais. O controlador *PI* (5.1-10) foi inicialmente sintonizado usando as regras de Ziegler-Nichols (método em malha aberta), e os seus parâmetros

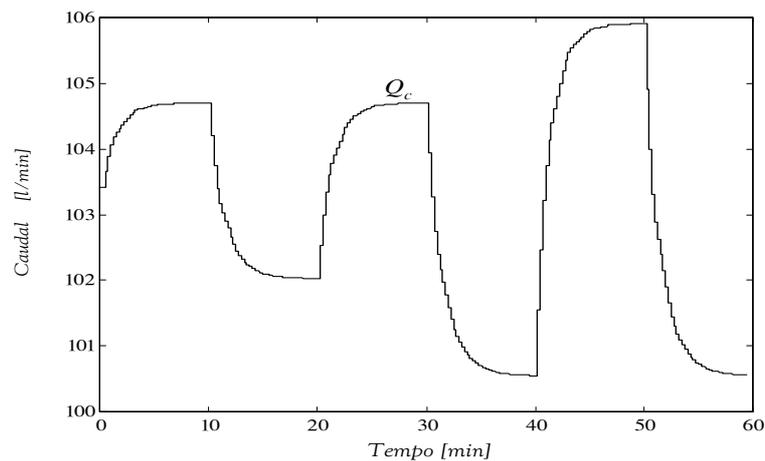
ajustados posteriormente manualmente, de forma a garantir a melhor resposta possível para uma variação de 5% ( $C_a = 0.105$ ) em torno de regime de operação. Foram efectuadas duas sintonizações, uma dando preferência ao aspecto de estabilidade, outra dando preferência à velocidade da resposta. Para cada uma das sintonizações os parâmetros obtidos foram os seguintes:  $q_0 = 0.44$  e  $q_1 = -0.19$  para a primeira sintonização e  $q_0 = 0.58$  e  $q_1 = -0.24$  para a segunda.

Nesta, e em todas as simulações que se seguem, foi considerado como intervalo de amostragem  $T = 0.25$  minutos, tendo sido utilizado para integração das equações diferenciais o método de Range-Kutta de 4ª ordem.

Na Figura 5.2-6 e Figura 5.2-7 mostra-se o desempenho do controlador PI para cada uma das sintonizações.

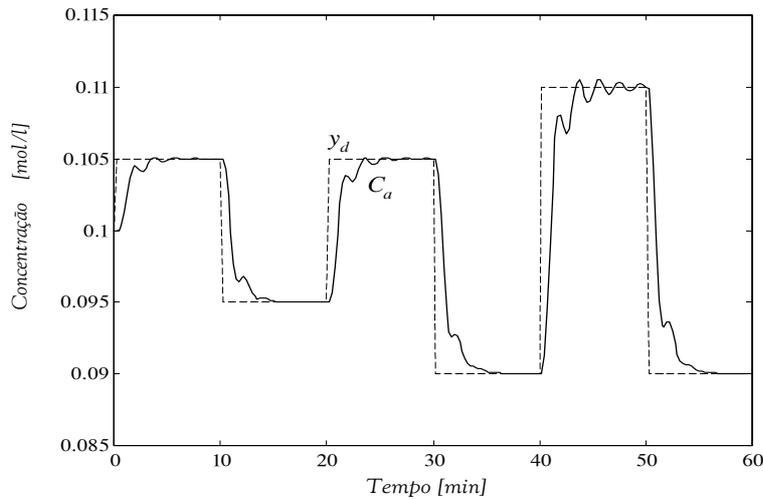


(a)- Saída desejada ( $y_d$ ) e saída do processo ( $C_a$ ).

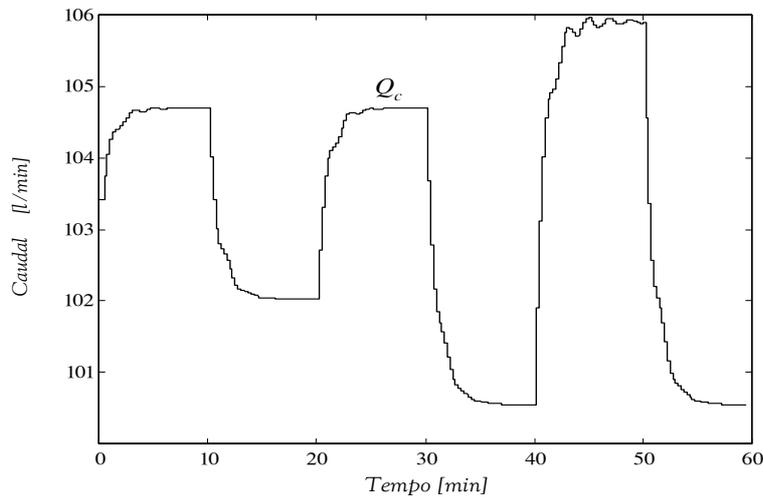


(b)- Acção de controlo ( $Q_c$ ).

Figura 5.2-6: Controlador PI;  $q_0 = 0.44$  e  $q_1 = -0.19$ .



(a)- Saída desejada ( $y_d$ ) e saída do processo ( $C_a$ ).



(b)- Acção de controlo ( $Q_c$ ).

Figura 5.2-7: Controlador PI;  $q_0 = 0.58$  e  $q_1 = -0.24$ .

Como se pode observar o controlador PI não garante em qualquer das sintonizações um bom desempenho. Se for considerada a primeira sintonização, obtém-se uma resposta algo lenta (tempo de estabelecimento de cerca de 5 minutos), sendo no entanto a saída do processo criticamente amortecida. Usando a segunda sintonização torna-se evidente, para amplitudes mais elevadas 10% ( $C_a = 0.110$ ), o facto da resposta ser quase oscilatória, sendo além disso não simétrica em torno do valor nominal, o que reflecte a natureza não linear do processo em questão. Para valores de referência de pequenas amplitudes a variação é quase simétrica indicando à partida que um modelo linear para valores próximos desta região de operação poderia ser utilizado com sucesso.

## 5.2.4 Controlador RSRN

### Controlador RSRN não Adaptativo

Aplica-se agora ao processo CSTR o controlador RSRN proposto nesta dissertação. Numa primeira fase considera-se uma versão fixa, não adaptativa, introduzindo-se depois, numa segunda fase, a adaptação dos parâmetros do modelo e consequentemente do controlador.

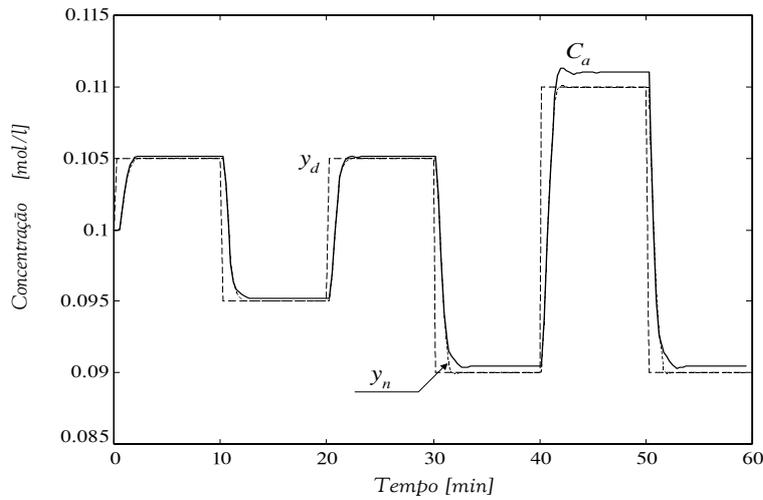
Para este caso de estudo, e após algumas simulações, foram especificados os seguintes parâmetros:  $\Lambda = 0.6 I_4$ ,  $A_0 = 0.4 I_4$  e  $Q_0 = 0.15 I_4$ , de que resultaram respectivamente:

$$K = [ -0.7242 \quad -4.9234 \quad -1.0749 \quad 0.8911 ]$$

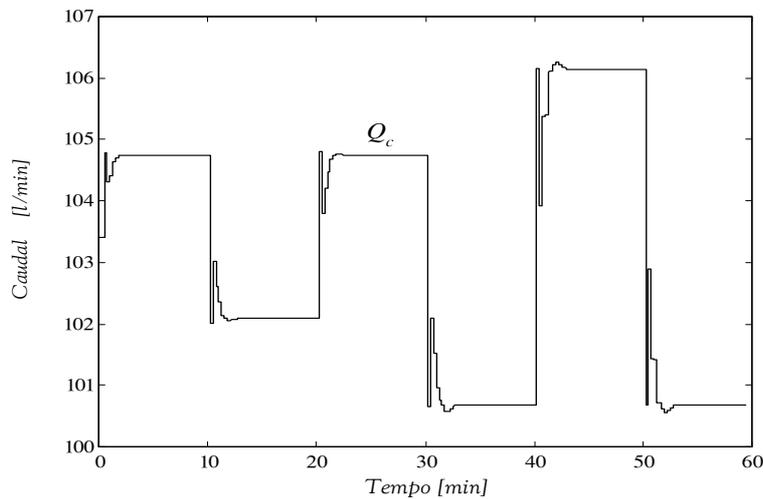
$$L = \begin{bmatrix} -0.6890 \\ 0.7821 \\ -0.2973 \\ -0.5438 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.1803 & -0.0003 & -0.0000 & 0.0003 \\ -0.0003 & 0.1803 & -0.0002 & -0.0003 \\ 0.0000 & -0.0002 & 0.1803 & -0.0002 \\ 0.0003 & -0.0003 & -0.0002 & 0.1803 \end{bmatrix}$$

Note-se que o controlador é determinado com base no modelo neuronal e aplicado simultaneamente a este e ao processo. Não é utilizada informação do processo de forma a compensar erros de controlo, definido pela diferença entre a saída desejada ( $y_d$ ) e saída do processo CSTR, ( $C_d$ ). O regulador é projectado apenas com o objectivo de anular o erro de regulação entre a saída desejada e a saída do modelo neuronal ( $y_n$ ), facto que é conseguido como pode ser verificado pela análise da Figura 5.2-8(a).



(a)- Saída desejada ( $y_d$ ), saída do modelo neuronal ( $y_n$ ) e saída do processo ( $C_a$ ).



(b)- Acção de controlo ( $Q_c$ ).

Figura 5.2-8: Controlo não adaptativo RSRN.

Como é visível na figura anterior, a resposta do sistema de controlo caracteriza-se principalmente por dois aspectos. Em primeiro lugar, e em termos de evolução da dinâmica, pelo facto de que o controlador neuronal RSRN permite compensar as não linearidades existentes, podendo considerar-se a resposta do modelo neuronal quase perfeita, Figura 5.2-8(a). Evidencia-se portanto a capacidade teórica de aplicação da teoria de regulação a modelos não lineares. Em segundo lugar, quando aplicado ao processo CSTR, como seria de esperar, o regulador não permite anular o erro em regime final, facto que se justifica uma vez que o modelo neuronal não representa com precisão o processo.

Existe, por outro lado, uma acção de controlo mais severa por parte do controlador neuronal, de forma a compensar as não linearidades do processo, visível principalmente na fase de alteração do regime de operação, Figura 5.2-8(b).

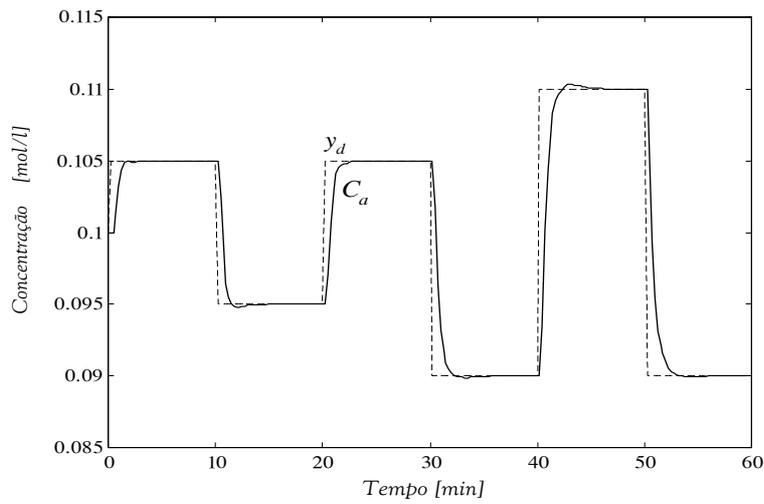
Pode-se concluir que, caso fosse possível obter um modelo inicial preciso, seriam evidente as vantagens de controladores não lineares fixos sobre controladores lineares também fixos, já que os primeiros, sendo projectados tendo em conta as não linearidades do processo permitem, em princípio, a obtenção de melhores resultados. A simulação apresentada na Figura 5.2-8 é um exemplo do que se afirma. Contudo, se for considerada uma estrutura de controlo não linear adaptativa, então as suas vantagens sobre estruturas lineares adaptativas não são tão evidentes. Refere-se o caso de controladores preditivos (Henriques *et al.*, 1999d) para os quais é francamente preferível basear a predição em modelos não lineares do que em modelos lineares.

No entanto coloca-se em causa a legitimidade de assumir como aspecto essencial um modelo inicial preciso. Por outras palavras, se se considerar um modelo fixo, por mais preciso que ele seja, para situações não previstas ou variações de dinâmicas desconhecidas, ele será incapaz de fornecer resultados adequados. Sendo assim, a componente adaptativa, ou outro qualquer mecanismo de compensação, de que a acção integral é exemplo, torna-se na prática sempre necessária.

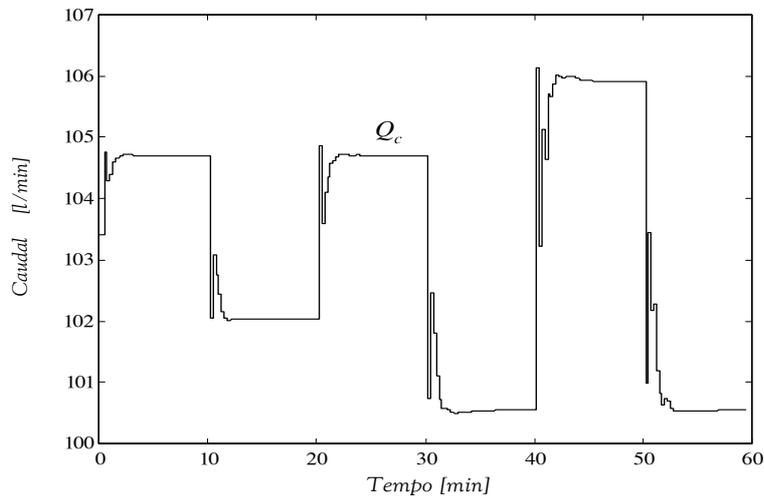
## Controlador *RSRN* Adaptativo

De forma a reduzir a diferença entre a saída do modelo neuronal e a saída do processo considera-se agora a adaptação dos parâmetros deste modelo. Assim, espera-se que o erro de controlo dado pela soma do erro de regulação e do erro de estimação (secção 4.4) convirja, uma vez que cada um dos erros considerados independentemente também converge.

Como se pode observar na Figura 5.2-9(a) o sistema de controlo neuronal apresenta, para este caso particular, um comportamento superior aos do *PI* mostrados na Figura 5.2-6 e Figura 5.2-7. Como se verifica o regulador permitir compensar as não linearidades e a adaptação permitir que o modelo descreva adequadamente o processo, o que em simultâneo conduz aos resultados observados.



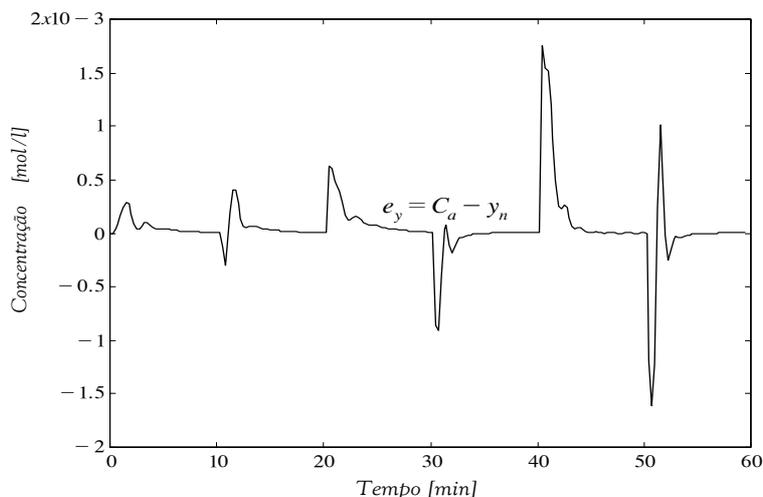
(a)- Saída desejada ( $y_d$ ) e saída do processo ( $C_a$ ).



(b)- Ação de controle ( $Q_c$ ).

Figura 5.2-9: Controlador adaptativo RSRN.

Na Figura 5.2-10 mostra-se a evolução do erro de estimação, diferença entre a saída do processo e a saída do modelo neuronal. Como se pode observar, o erro tem o seu valor máximo sempre que existem variações de regimes de operação, anulando-se passado um período transitório de adaptação.

Figura 5.2-10: Erro de estimação, ( $e_y$ ).

Na Tabela 5.2-4 comparam-se quantitativamente os resultados obtidos pelos quatro controladores implementados: *PI* para primeira sintonização (PI1), *PI* para a segunda sintonização (PI2), controlador *RSRN* não adaptativo (RSRN1) e controlador *RSRN* adaptativo (RSRN2). Os critérios utilizados são a média do valor quadrático do erro de controlo SSCE, (5.1-12), e o somatório da variação da acção de controlo SSCC, (5.1-8).

Critério	PI1	PI2	RSRN1	RSRN2
$\Sigma e_c^2$	11.3307	9.1835	5.0765	4.6702
$\Sigma \Delta u^2$	0.5853	0.8731	12.7774	15.8821

Tabela 5.2-4: Comparação do desempenho dos controladores *PI* e *RSRN*.

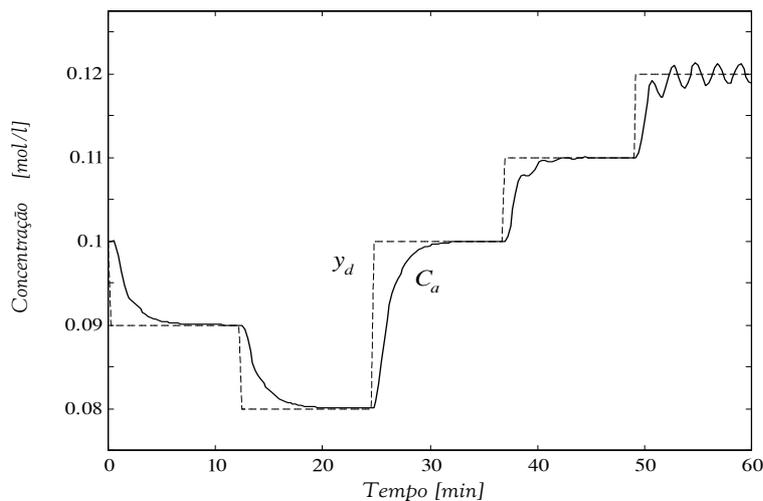
Os valores numéricos confirmam os comentários já efectuados. De facto, analisando o primeiro critério ( $\Sigma e_c^2$ ) é claramente visível a vantagem de qualquer uma das estruturas de controlo neuronal *RSRN* (adaptativa e não adaptativa) em relação aos controladores *PI*. Além disso, considerando apenas a metodologia neuronal é possível concluir que a introdução da adaptação dos parâmetros do modelo conduz a uma diminuição do erro de controlo, objectivo final do sistema de controlo.

No entanto, existem maiores variações por parte do actuador no caso do controlador *RSRN*, de forma a compensar as não linearidades do processo, o que é patente no segundo critério de erro ( $\Sigma \Delta u^2$ ), o que pode ser considerado uma desvantagem.

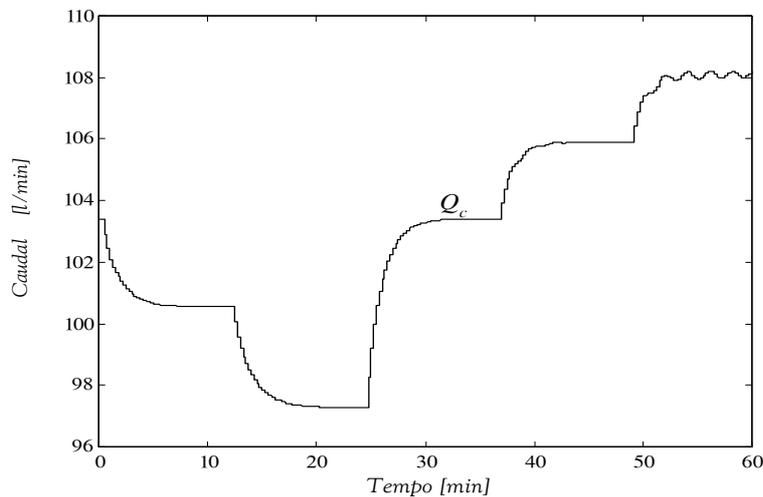
## Variação do Regime de Operação

Finalmente, compara-se o desempenho de um controlador *PI* (sintonizado anteriormente de parâmetros  $q_0 = 0.44$  e  $q_1 = -0.19$ ) com o controlador neuronal adaptativo *RSRN*, considerando que o valor desejado de concentração  $y_d$ , atinge valores elevados, concretamente variações de 20% o que corresponde  $y_d = 0.12 \text{ mol/l}$ .

Na Figura 5.2-11 e na Figura 5.2-12 mostra-se respectivamente o desempenho do controlador *PI* e o desempenho do controlador *RSRN*.



(a)- Saída desejada ( $y_d$ ) e saída do processo ( $C_a$ ).



(b)- Acção de controlo ( $Q_c$ ).

Figura 5.2-11: Controlador *PI*;  $q_0 = 0.44$  e  $q_1 = -0.19$ .

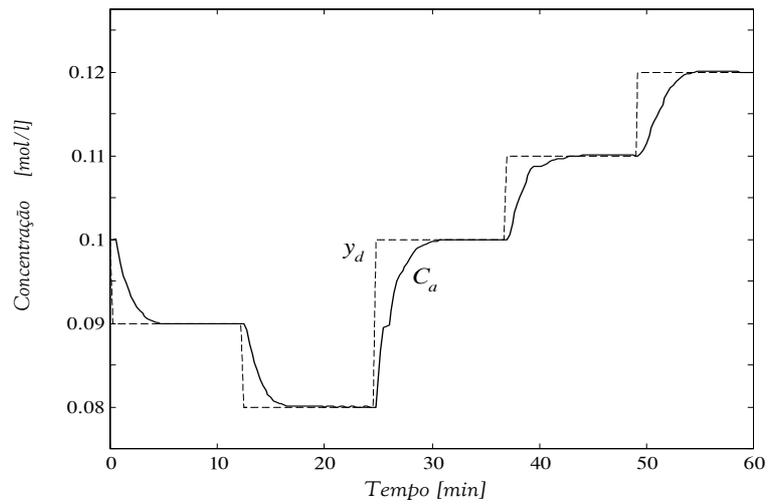
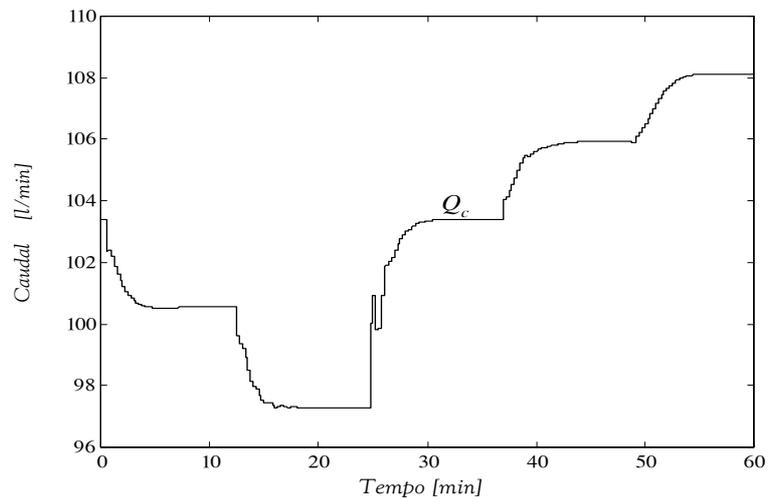
(a)- Saída desejada ( $y_d$ ) e saída do processo ( $C_a$ ).(b)- Ação de controle ( $Q_c$ ).

Figura 5.2-12: Controlador adaptativo RSRN.

Como se pode observar na Figura 5.2-12, o controlador neuronal RSRN desempenha adequadamente as suas funções, manter o valor da saída  $C_a$  no valor desejado  $y_d$ , com vantagens em relação ao controlador fixo linear (PI). Para valores extremos de variações positivas no regime de operação (+20%) é perfeitamente visível a oscilação da saída do processo no caso do controlador PI e, por outro lado, para variações negativas (-20%) a resposta do processo é demasiado lenta. Conclui-se portanto que a sintonização de um controlador PI, capaz de assegurar resultados

aceitáveis nas duas regiões de operação, resultará necessariamente de uma situação de compromisso.

Por sua vez o controlador neuronal é mais regular, funcionando satisfatoriamente para todos os regimes de operação considerados, facto que se justifica principalmente pela sua capacidade de lidar com as não linearidades do processo.

Na Figura 5.2-13 apresenta-se a evolução do erro de estimação.

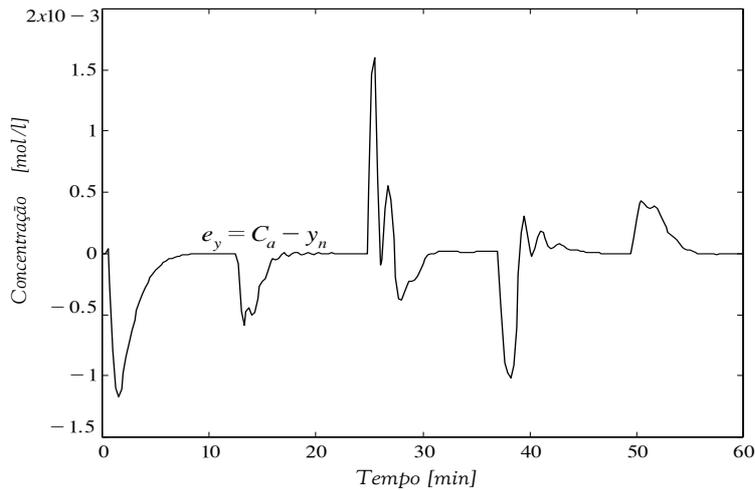


Figura 5.2-13: Erro de estimação, ( $e_y$ ).

Usando os critérios definidos em (5.1-6) e (5.1-7) comparam-se na Tabela 5.2-5 o desempenho obtido pelo controlador *PI* e pelo regulador adaptativo.

Critério	PI	RSRN
$\Sigma e_c^2$	8.9135	7.0174
$\Sigma \Delta u^2$	0.4228	0.9884

Tabela 5.2-5: Comparação do desempenho dos controladores *PI* e *RSRN*.

A análise quantitativa confirma os resultados observados na Figura 5.2-11 e Figura 5.2-12. O erro de controlo é mais elevado quando se implementa o controlador *PI*, podendo o seu valor ser reduzido implementando o controlador neuronal, que implica por sua vez um aumento do esforço de actuação.

### 5.2.5 Análise de Resultados

O objectivo deste caso de estudo consistiu fundamentalmente em validar a capacidade da metodologia de controlo neuronal *RSRN* em lidar com processos não lineares, tomando como exemplo um processo químico *CSTR*. Tendo em vista a comparação do seu desempenho, implementou-se também um controlador *PI*, sem dúvida um dos mais utilizados em processos industriais deste tipo.

Da comparação dos resultados obtidos nas várias simulações tornam-se claras, para este caso particular, as vantagens do controlador *RSRN* em relação ao *PI*.

Assim, a conclusão principal a reter é a capacidade da metodologia de controlo não linear em lidar com não linearidades (tempo de atraso variável) ou, interpretando a questão segundo outra perspectiva, quando em presença de não linearidades uma técnica de controlo não linear será mais adequada do que uma técnica linear.



# 5.3

## Tanques Acoplados - DTS200

*Nesta secção investiga-se a aplicação da metodologia de controlo RSRN a um sistema multi-variável, o processo de tanques acoplados DTS200<sup>1</sup>.*

*É objectivo fundamental investigar a capacidade de técnicas de controlo baseadas na acessibilidade de estados acessíveis, com a técnica proposta, entendida como um observador baseada apenas em informação de entrada-saída .*

### 5.3.1 Introdução

O processo dos três tanques é um processo não linear, multi-variável caracterizando-se pela existência de acoplamento entre as suas variáveis. Devido às suas características, semelhantes às encontradas em alguns processos industriais, tem sido considerado um *benchmark* na avaliação de estratégias de controlo.

Além das características apontadas, o processo dos tanques acoplados é um processo particular no qual os estados se encontram acessíveis, sendo portanto possível neste caso concreto implementar estratégias de controlo baseadas em modelos de estados, com as reconhecidas vantagens que daí advêm. No entanto, na maioria das aplicações de controlo, esta situação não se verifica, e as metodologias de controlo a ter em consideração têm que ser desenvolvidas com base apenas em informação entrada-saída.

O uso da metodologia proposta nesta dissertação, incorporando modelos neuronais desempenhando as funções de um observador, permite efectuar uma estimação simultânea de parâmetros e de estados, possibilitando a aplicação de técnicas de realimentação de estados recorrendo na verdade apenas a informação de entrada-saída.

### Descrição do Processo

O processo dos três tanques acoplados (Amira, 1996) é mostrado na Figura 5.3-1 e ilustrado esquematicamente na Figura 5.3-2. Como o próprio nome sugere é um processo constituído por três tanques ( $T_1$ ,  $T_2$  e  $T_3$ ) sendo um dos tanques ( $T_3$ ) ligado aos outros

---

<sup>1</sup> Amira GmbH

através de tubagem de secção circular, munidas de válvulas,  $v_{13}$  e  $v_{23}$ , cuja abertura pode ser ajustada manualmente. No tanque  $T_2$  encontra-se a saída principal de líquido do sistema,  $v_2$ , ligando-o a um reservatório,  $R$ . Adicionalmente, cada tanque possui na sua base outra saída de líquido ( $v_{10}$ ,  $v_{20}$  e  $v_{30}$ ), de abertura manual, com as quais se podem introduzir perturbações sob a forma de fugas e/ou obstruções, originando desta forma variações na dinâmica do processo.

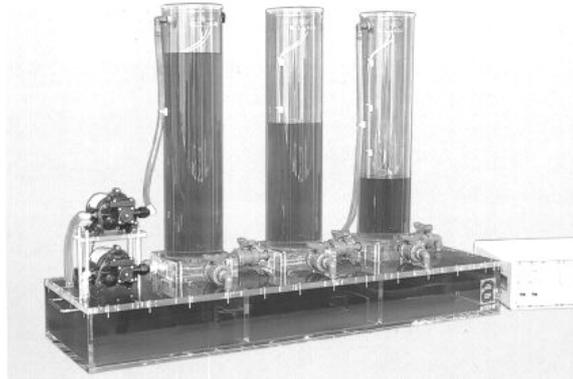


Figura 5.3-1: Processo DTS200.

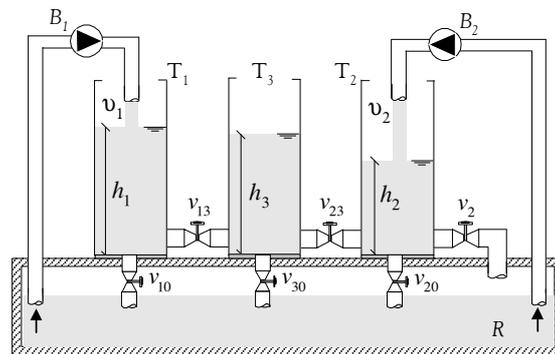


Figura 5.3-2: Diagrama esquemático do processo DTS200.

Os tanques  $T_1$  e  $T_2$  são alimentados a partir do reservatório inferior por duas bombas que lhes estão associadas, respectivamente a bomba  $B_1$  e a bomba  $B_2$ , que fornecem um caudal volúmico  $v_1$  e  $v_2$  a cada um dos tanques. Para medição do nível do líquido  $h_i$ ,  $i = 1, 2, 3$ , é utilizado em cada um dos tanques um transdutor de pressão diferencial *piezoresistivo*.

O objectivo do sistema de controlo é fazer com que o nível do líquido  $h_1$  e  $h_2$ , relativo a cada um dos tanques  $T_1$  e  $T_2$ , seja um determinado valor desejado  $y_{d1}$  e  $y_{d2}$ , através da manipulação dos caudais  $v_1$  e  $v_2$ .

## Configuração Experimental

Na realização das várias experiências considera-se que apenas o tanque  $T_2$  abastece o reservatório  $R$ , isto é, as válvulas  $v_{10}$ ,  $v_{20}$  e  $v_{30}$  estão fechadas, encontrando-se abertas as válvulas  $v_{13}$ ,  $v_{23}$  e  $v_2$ .

$$\begin{aligned} v_{10} &= v_{20} = v_{30} \text{ (fechadas)} \\ v_{13} &= v_{23} = v_2 \text{ (abertas)} \end{aligned} \quad (5.3-1)$$

Na prática, as acções de controlo  $u_i$ ,  $i=1,2$ , que determinam a potência a ser fornecida a cada um dos actuadores (bombas), são enviadas a partir do computador através de um conversor D/A (Digital/Analógico), numa gama de  $[0, 10]$  Volts. A estes valores mínimos e máximos correspondem respectivamente os caudal volúmico de  $v_i = 0.0 \text{ m}^3/\text{s}$  e de  $v_i = 0.0001 \text{ m}^3/\text{s}$ .

A leitura de cada um dos níveis  $h_i$ ,  $i=1,2,3$ , compreendida entre  $h = 0.0 \text{ metros}$  e  $h = 0.6 \text{ metros}$  (altura máxima dos tanques) é efectuada pelos sensores, recebido de um conversor A/D (Analógico/Digital), como um sinal em tensão, no intervalo  $[-10, 10]$  Volts.

Nas experiências realizadas considerou-se para intervalo de amostragem  $T = 1 \text{ segundos}$ .

## Modelização Matemática Baseada em Primeiros Princípios

De forma a obter uma descrição matemática para a dinâmica do processo são aplicadas as leis fundamentais da física, no caso particular balanços mássicos, a cada um dos três tanques. Considerando a configuração acima referida (5.3-1), obtém-se as seguintes equações diferenciais (5.3-2), que caracterizam a evolução da altura do líquido em cada um dos tanques  $T_1$ ,  $T_2$  e  $T_3$ .

$$\frac{dh_1(t)}{dt} = \frac{1}{A} [ v_1 - f_{13} ] \quad (5.3-2a)$$

$$\frac{dh_2(t)}{dt} = \frac{1}{A} [v_2 + f_{32} - f_{20}] \quad (5.3-2b)$$

$$\frac{dh_3(t)}{dt} = \frac{1}{A} [f_{13} - f_{32}] \quad (5.3-2c)$$

As variáveis  $f_{ij}$  definem caudais volúmicos, dados por (5.3-3).

$$f_{13} = \xi_{13} S \operatorname{sgn}(h_1 - h_3) \sqrt{2g |h_1 - h_3|} \quad (5.3-3a)$$

$$f_{32} = \xi_{32} S \operatorname{sgn}(h_3 - h_2) \sqrt{2g |h_3 - h_2|} \quad (5.3-3b)$$

$$f_{20} = \xi_{20} S \sqrt{2gh_2} \quad (5.3-3c)$$

O parâmetro  $A$  define a secção transversal (circular) de cada um dos tanques e  $S$  a secção transversal das tubagens de interconexão.  $\xi_{ij} \in [0,1]$  é um coeficiente adimensional de escoamento,  $g$  é a constante de atracção universal,  $\operatorname{sgn}(\cdot)$  define a função sinal e  $|\cdot|$  define a função valor absoluto.

Note-se que o modelo acima descrito, (5.3-2) e (5.3-3), define na verdade um modelo de estados do tipo *affine*, (5.3-4a).

$$\dot{x}(t) = a(x(t)) + b(x(t)) v(t) \quad (5.3-4a)$$

$$y(t) = C x(t) \quad (5.3-4b)$$

O estado  $x(t)$  é definido pelo nível nos tanques  $h(t)$ , sendo  $f(\cdot)$ ,  $g(\cdot)$  e  $C$  definidos por (5.3-5).

$$a(x(t)) = \frac{1}{A} \begin{bmatrix} -f_{13} \\ f_{32} - f_{20} \\ f_{13} - f_{32} \end{bmatrix} \quad (5.3-5a)$$

$$b(x(t)) = \frac{1}{A} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (5.3-5b)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.3-5c)$$

Os parâmetros que caracterizam o processo laboratorial *DTS200*, obtidos experimentalmente, apresentam-se na Tabela 5.3-1.

Parâmetro	Descrição	Valor
$\xi_{13}$	Coeficiente de escoamento	0.70
$\xi_{32}$	Coeficiente de escoamento	0.70
$\xi_{20}$	Coeficiente de escoamento	0.61
$A$	Área da secção transversal de cada tanque	$0.0154 \text{ m}^2$
$S$	Área da secção das tubagens de ligação	$5 \times 10^{-5} \text{ m}^2$
$g$	Constante de atracção universal	$9.81 \text{ m s}^{-2}$

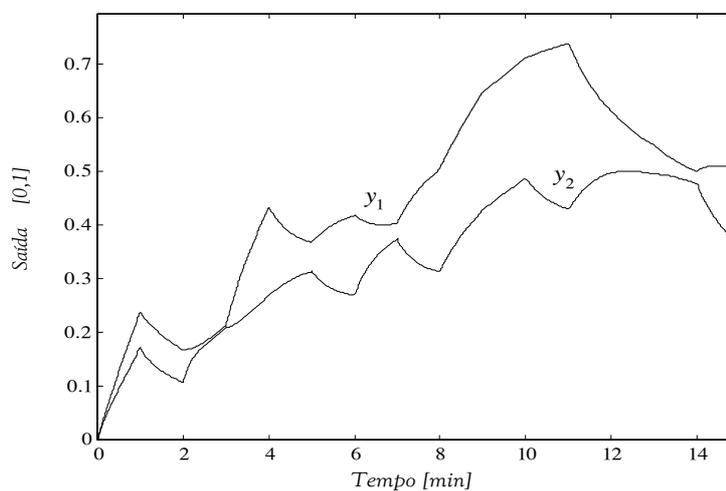
Tabela 5.3-1: Parâmetros do processo DTS200.

### 5.3.2 Identificação Experimental

A obtenção do modelo neuronal é efectuada a partir de um conjunto de dados de entrada-saída, admitindo o desconhecimento da descrição física do processo.

É importante que os dados sejam representativos do processo, o que neste caso significa que as entradas (abertura das bombas) devem ser manipuladas de forma a que as saídas (nível do líquido nos tanques) sejam representativas das regiões de interesse. É também essencial que a variação seja suficientemente rica, isto é, que os dados sejam variados tanto em frequência (de forma a capturar a dinâmica) como em amplitude (de forma a lidar com não linearidades).

Para a entrada considerada Figura 5.3-3(b), mostra-se a evolução dinâmica da saída do processo, Figura 5.3-3(a).

(b)- Saídas do processo ( $y_1$  e  $y_2$ ).

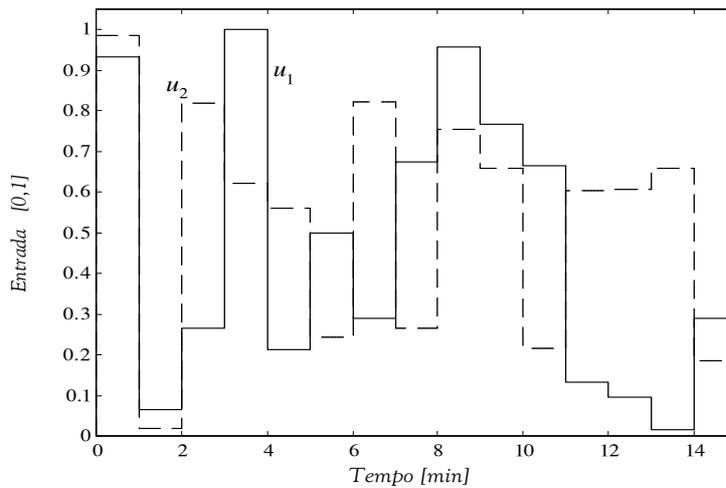
(b)- Ações de controlo ( $u_1$  e  $u_2$ ).

Figura 5.3-3: Sequência de dados de treino.

Tal como nos casos de estudo anteriores, foi usado o algoritmo de Levenberg-Marquardt para determinar os parâmetros da rede neuronal, admitindo como número de neurónios na camada interna  $n = 3$ , justificado em parte pelo conhecimento existente acerca da ordem do sistema, (5.3-2, que se sabe ser de terceira ordem.

No processo de treino, as matrizes do modelo neuronal foram inicializadas tendo em conta o conhecimento dinâmico existente acerca do processo. Assim, uma vez que o processo é caracterizado por uma dinâmica com características aproximadamente integrais foram, numa primeira fase, considerados para valores próprios da matriz  $A$  do modelo neuronal valores próximos do limite de estabilidade ( $z \approx 1$ , no círculo de raio unitário). As restantes matrizes  $B$ ,  $C$  e  $D$  foram inicializadas da seguinte forma:  $B$  e  $C$  aleatoriamente, segundo uma distribuição normal, no intervalo  $[0.0, 0.05]$  e  $D = 0.01 \times A$ .

Foram ainda utilizadas as expressões de normalização (5.3-6) e (5.3-7).

$$y_n = -0.05 y(k) + 0.05 \quad (5.3-6)$$

$$u_n = 0.1 u(k) \quad (5.3-7)$$

As variáveis  $y_n$  e  $u_n$  representam, respectivamente, os valores considerados como entradas e saídas utilizadas pela rede neuronal, ambos pertencentes ao intervalo  $[0, 1]$ .

Na Figura 5.3-4 mostra-se a evolução da saída real do processo ( $y$ ), da saída do modelo neuronal ( $y_n$ ) e da saída de um modelo linear ( $y_l$ ), obtido usando a função *ARX* do Matlab (*Control Toolbox*, 1998).

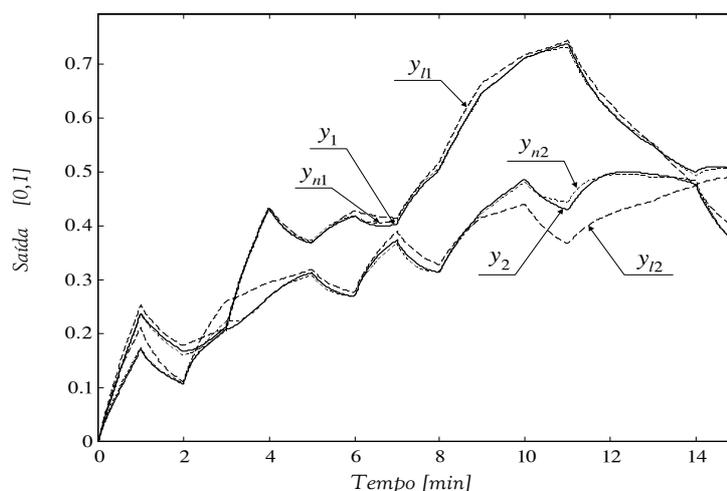


Figura 5.3-4: Identificação; comparação da saída real ( $y$ ) com a do modelo neuronal ( $y_n$ ) e saída de um modelo linear ( $y_l$ )

Da análise da Figura 5.3-4 pode-se concluir-se a capacidade do modelo neuronal em captar satisfatoriamente a dinâmica do processo, visível na proximidade entre a saída do processo real ( $y$ ) e do modelo neuronal ( $y_n$ ). Caso se considere um modelo linear, como pode também ser observado na Figura 5.3-4, não é possível obter resultados tão satisfatórios, o que reflecte a natureza não linear do processo em questão.

### 5.3.3 Controlo por Realimentação de Variáveis de Estado

Consideram-se neste grupo de experiências duas situações distintas. Primeiro, o projecto de um controlador linear, assumindo que se conhece a descrição física do sistema (5.3-2) e que os estados são acessíveis. Em segundo lugar, o projecto do controlador neuronal, usando apenas informação entrada-saída e não admitindo qualquer conhecimento físico sobre o processo.

#### Controlador Linear

Considere-se então como primeira situação o projecto de um controlador linear, baseado numa realimentação de variáveis de estado. Para tal procede-se à linearização do modelo físico, dado em (5.3-2), de forma a obter um modelo linear, a partir do qual seja possível projectar o controlador.

### Linearização

Da linearização do modelo não linear, resultante da aplicação de primeiros princípios e considerando o regime de operação apresentado na Tabela 5.3-2, é possível obter uma representação linear, no domínio contínuo.

Parâmetro	Descrição	Valor
$u_1$	Acção de controlo 1	0.244
$u_2$	Acção de controlo 1	0.281
$y_1$	Nível no tanque 1	0.200
$y_2$	Nível no tanque 2	0.150
$y_3$	Nível no tanque 3	0.175

Tabela 5.3-2: Regime estacionário considerado.

Da discretização do modelo linear contínuo, utilizando para intervalo de discretização  $T = 1$  segundos, resultou o modelo linear descrito em (5.3-8a) e caracterizado pelos parâmetros definidos em (5.3-9).

$$x(k+1) = \underline{A} x(k) + \underline{B} u(k) \quad (5.3-8a)$$

$$y(k) = \underline{C} x(k) \quad (5.3-8b)$$

$$\underline{A} = \begin{bmatrix} 0.9526 & 0.0000 & 0.0474 \\ 0.0000 & 0.9136 & 0.0638 \\ 0.0474 & 0.0638 & 0.8889 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} 0.0065 & 0.0000 \\ 0.0000 & 0.0065 \\ 0.0000 & 0.0000 \end{bmatrix} \quad (5.3-9)$$

$$\underline{C} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \end{bmatrix}$$

### Controlador Linear

Com base neste modelo linear é projectado um controlador por realimentação de variáveis de estado (Henriques e Dourado, 1999a). Neste caso a lei de controlo é definida por (5.3-10),

$$u(k) = G y_d(k) + F x(k) \quad (5.3-10)$$

em que as matrizes  $F \in \mathcal{R}^{n_u \times n_n}$  e  $G \in \mathcal{R}^{n_u \times n_y}$  são determinadas de forma a posicionar os valores próprios do sistema em malha fechada em localizações desejadas e simultaneamente garantir que a entrada de referência  $y_{d_i}$  influencie apenas a saída  $y_i$  (condição de desacoplamento).  $n_u$ ,  $n_y$ , e  $n_n$  definem, respectivamente, o número de entradas, o número de saídas e o número de estados do processo.

Usando os resultados de Falb e Wolovith (1967), determinam-se as matrizes  $F$  e  $G$  a partir de (5.3-11) e (5.3-12).

$$F = B^{*-1} \left[ \sum_{k=0}^{\lambda} M_k C A^k + A^* \right] \quad (5.3-11)$$

$$G = B^{*-1} \quad (5.3-12)$$

Em que

$$B^* = \begin{bmatrix} C_1 A^{d_1} B \\ \vdots \\ C_{n_y} A^{d_{n_y}} B \end{bmatrix} \quad (5.3-13)$$

$$A^* = \begin{bmatrix} C_1 A^{d_1+1} \\ \vdots \\ C_{n_y} A^{d_{n_y}} \end{bmatrix} \quad (5.3-14)$$

Na equação (5.3-11) as matrizes  $M_k$  são determinadas de forma a especificar os valores próprios do sistema em malha fechada em localizações desejadas. O parâmetro  $\lambda$  é dado por  $\lambda = \max(d_i)$ , em que os elementos  $d_1, \dots, d_{n_y}$  são definidos por (5.3-15), e definem o grau relativo de cada uma das saídas, definidos neste caso (linear) por (5.3-15) como um valor particular de (@\*\*).

$$d_i = \min \{ j : C_i A^j B \neq 0 \} \quad (5.3-15)$$

ou

$$d_i = n_y - 1 \text{ se } C_i A^j B = 0 \quad \forall j$$

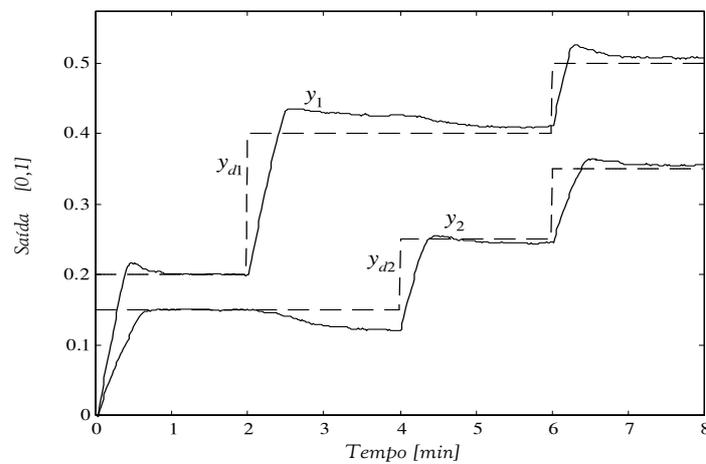
Da aplicação deste algoritmo, equações (5.3-10)-(5.3-15), ao sistema linear (5.3-8) e (5.3-9), e especificando para valores próprios os valores de 0.70 para ambos os sistemas

desacoplados,  $y_{d1} \leftrightarrow y_1$  e  $y_{d2} \leftrightarrow y_2$ , resulta a lei de controlo, caracterizada pelas matrizes definidas em (5.3-10).

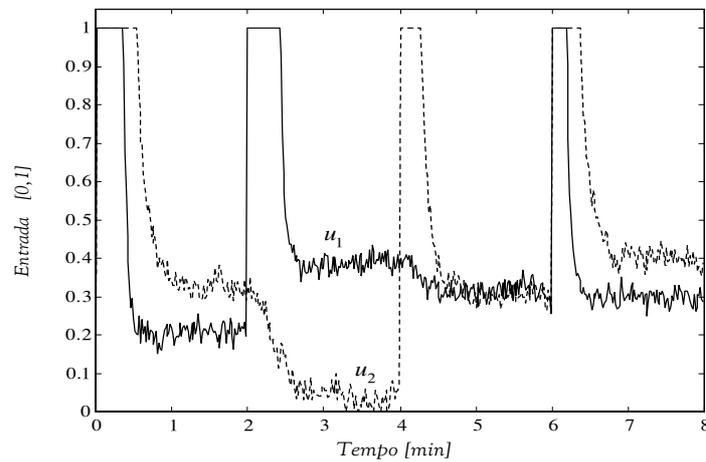
$$F = \begin{bmatrix} -38.9036 & 0.0000 & -7.2964 \\ 0.0000 & -32.8939 & -9.8197 \end{bmatrix} \quad (5.3-16)$$

$$G = \begin{bmatrix} 46.2000 & 0.0000 \\ 0.0000 & 46.2000 \end{bmatrix}$$

Na Figura 5.3-5 mostra-se, depois de implementado este controlador, os resultados experimentais obtidos no processo DTS200.



(a)- Saídas desejadas ( $y_{d1}$  e  $y_{d2}$ ) e do processo ( $y_1$  e  $y_2$ ).



(b)- Ações de controlo ( $u_1$  e  $u_2$ ).

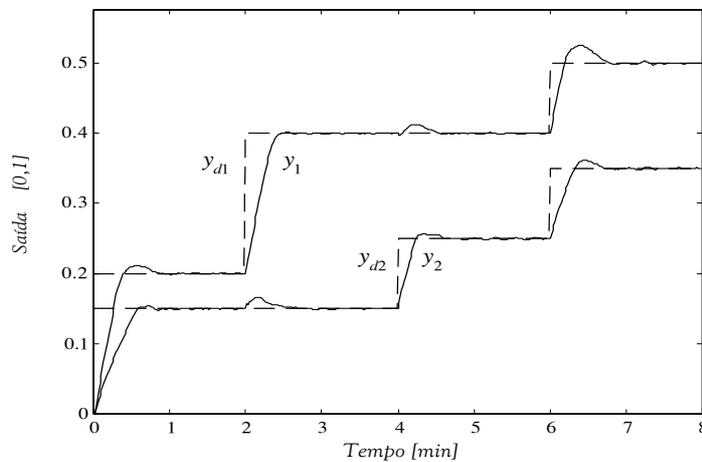
Figura 5.3-5: Controlo linear usando realimentação de variáveis de estado.

Da análise da Figura 5.3-5 é visível que para valores de saída desejados próximos daqueles para o qual foi obtido o modelo linear, é possível obter excelentes resultados. Contudo, para valores de referência diferentes o controlador tem dificuldades em anular o erro em regime final, facto que se justifica quer por imprecisões dos parâmetros do modelo de primeiros princípios, quer da linearização considerada.

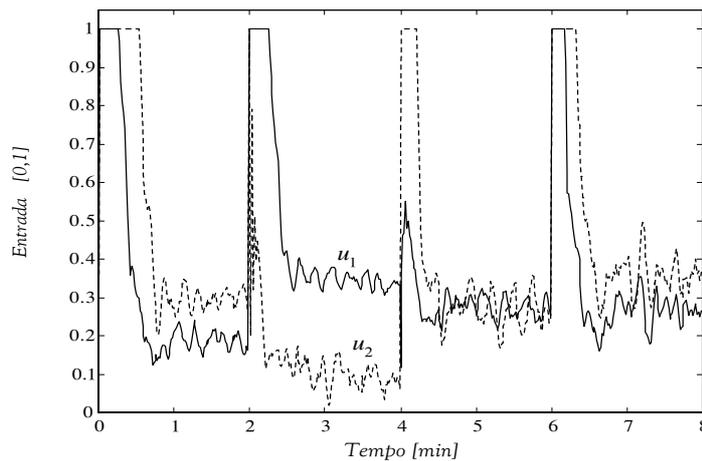
## Regulação da Saída Baseada no Modelo Neuronal

Considera-se agora a metodologia de controlo *RSRN* desenvolvida nesta dissertação, especificando os seguintes parâmetros  $\Lambda = 0.9 I_3$ ,  $A_0 = 0.2 I_3$  e  $Q_0 = 0.25 I_3$ .

Os resultados da aplicação prática do controlador ao processo *DTS200* mostram-se na Figura 5.3-6.



(a)- Saídas desejadas ( $y_{d1}$  e  $y_{d2}$ ) e do processo ( $y_1$  e  $y_2$ ).



(b)- Acções de controlo ( $u_1$  e  $u_2$ ).

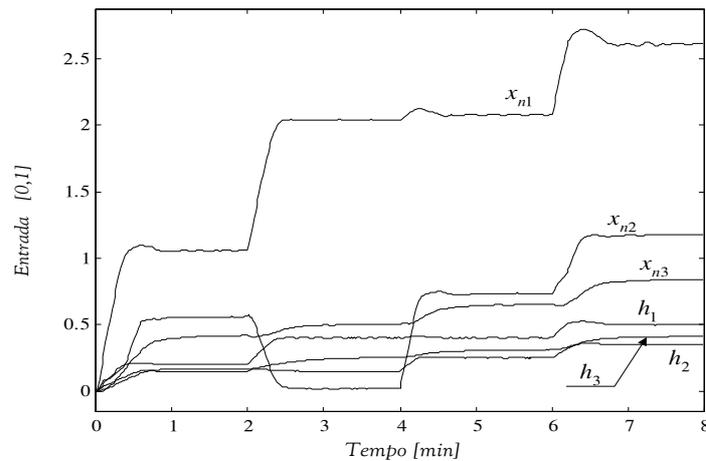
(c)- Estados reais ( $h$ ) e do modelo neuronal ( $x_n$ ).

Figura 5.3-6: Controlo neuronal adaptativo.

Da sua análise é possível concluir que, para o caso considerado a técnica de regulação baseada num modelo neuronal é capaz de controlar convenientemente os níveis  $y_1$  e  $y_2$  em cada dos tanques  $T_1$  e  $T_2$  (Figura 5.3-6a). Por comparação com a experiência realizada utilizando o controlador convencional fixo (Figura 5.3-5a), é visível a capacidade do regulador em eliminar o erro em regime final.

Como se mostra na Figura 5.3-6(c) os valores reais dos estados não coincidem com os valores dos estados da rede neuronal (saída da unidade escondida). Como se havia referido a rede neuronal não é de facto um observador, não sendo seu objectivo determinar os estados reais do processo. No entanto, não é por esta razão que o controlador deixa de cumprir o objectivo, uma vez que toda a estratégia se baseia, como se referiu na secção @3.4, não na convergência de estados mas na convergência da saída, o que de facto se verifica, Figura 5.3-6(b).

Entendida a questão de outra forma, e uma vez que se sabe existir uma infinidade de possibilidades de representação de modelos de estados que descrevem a mesma relação entrada-saída de um processo, a rede obtida não é senão uma dessas representações.

Na Tabela 5.3-3 quantifica-se o desempenho de cada um dos controladores, linear e neuronal, para cada uma das entradas e saídas, sendo os critérios especificados por (@5.1-7) e (@5.1-8).

Critério	Linear	Neuronal
$\Sigma e_{n1}^2$	0.8482	0.6656
$\Sigma e_{n2}^2$	0.4765	0.3585
$\Sigma \Delta u_1^2$	2.6935	2.6938

$\Sigma \Delta u_2^2$	2.7850	4.6801
-----------------------	--------	--------

Tabela 5.3-3: Comparação das técnicas de controlo por realimentação de variáveis de estado e RSRN.

Da análise da Tabela 5.3-3 e a partir da observação da Figura 5.3-5 e da Figura 5.3-6 pode concluir-se que ambos os controladores apresentam um comportamento semelhante. Como diferença principal refere-se o facto de o controlador fixo não permitir anular o erro em regime final para regimes de operação para o qual não foi sintonizado.

Refira-se que para este sistema particular um controlador que use apenas informação da saída, de que é exemplo o típico *PI* considerado nos casos de estudo anteriores, tem algumas dificuldades em controlar convenientemente o processo. Uma vez que o sistema apresenta um comportamento do tipo integral, a aplicação de um simples controlador proporcional (*P*) permite alcançar um comportamento dinâmico aceitável não conseguindo contudo anular o erro em regime final. A introdução da componente integral (*PI*) apesar de permitir anular o erro em regime final conduz na prática a que o comportamento do sistema se caracterize por uma das seguintes situações: demasiado lenta ou demasiado oscilatória, quando se considera respectivamente uma componente integral pequena ou elevada, sendo difícil de conseguir uma situação de compromisso aceitável.

Serve esta explicação para, para justificar por um lado a razão pelo qual não foi utilizado como técnica de comparação um controlador *PI*, e por outro justificar os resultados aceitáveis devidos à utilização de técnicas de realimentação de variáveis de estado. Este último facto justifica-se pelo uso do tipo de informação que se admite conhecer: um modelo de estados que descreve o comportamento do processo e também a acessibilidades desses mesmos estados.

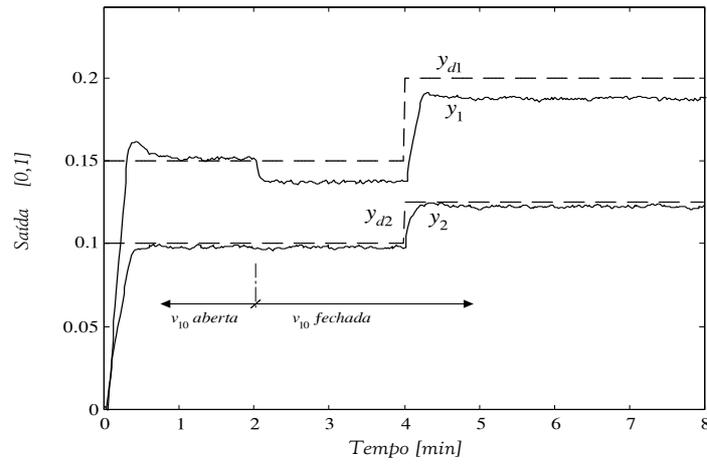
No entanto este é um processo muito particular, em que os estados são acessíveis sendo, na maioria dos casos, inviável a sua aplicação uma vez que a informação disponível é apenas de entrada-saída. Neste caso uma solução consiste em proceder à determinação indirecta dos estados, ou seja, à implementação de um observador.

A metodologia *RSRN* é um exemplo desta abordagem e, conforme mostrado nas experiências realizadas, a metodologia neuronal desenvolvida permite ultrapassar as dificuldades inerentes à não acessibilidade dos estados.

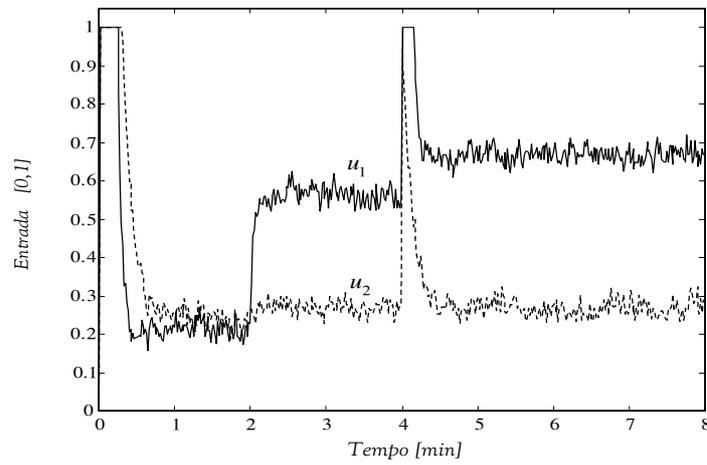
### 5.3.4 Simulação de Fugas

Considera-se agora o caso em que são introduzidas variações na dinâmica do processo sob a forma de fugas. Concretamente, com o objectivo de avaliar a robustez de cada controlador, é aberta a válvula  $v_{10}$ , que conecta directamente o tanque  $T_1$  ao reservatório, a partir do instante  $t = 20$  segundos.

Figura 5.3-7 e Figura 5.3-8 mostra-se o desempenho de cada um dos controladores, realimentação de variáveis de estado e *RSRN*, nesta situação.

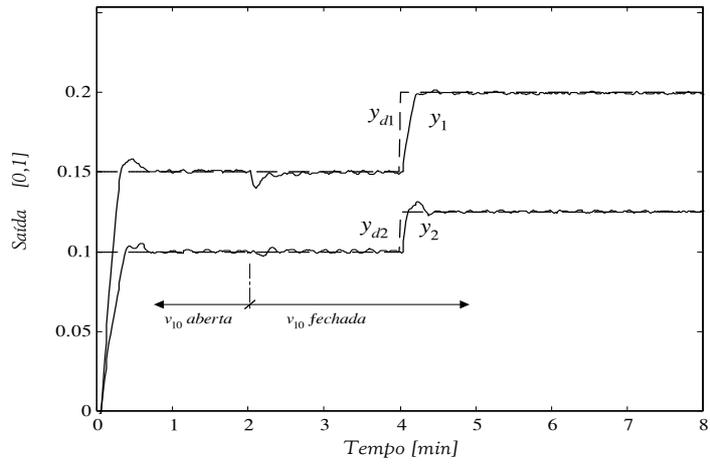


(a)- Saídas desejadas ( $y_{d1}$  e  $y_{d2}$ ) e do processo ( $y_1$  e  $y_2$ ).

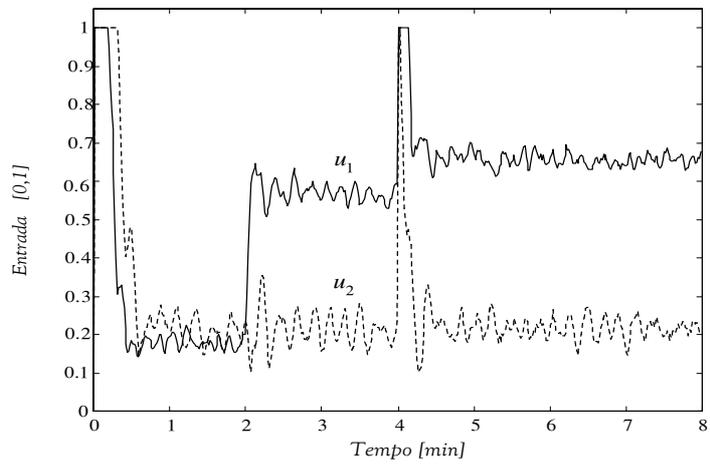


(b)- Ações de controle ( $u_1$  e  $u_2$ ).

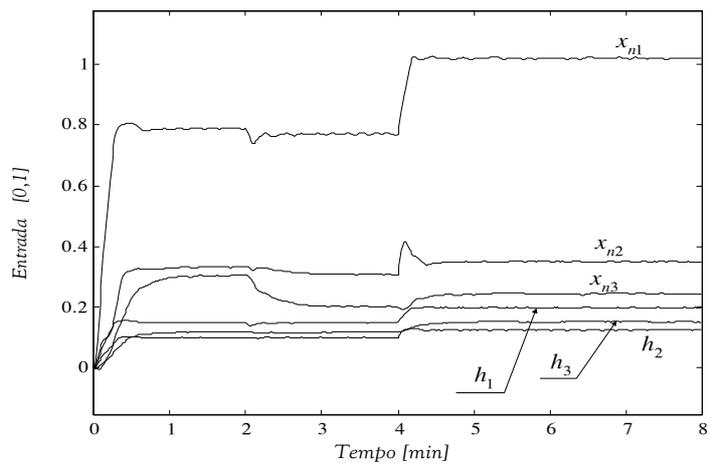
Figura 5.3-7: Controlador linear usando realimentação de variáveis de estado.



(a)- Saídas desejadas ( $y_{d1}$  e  $y_{d2}$ ) e do processo ( $y_1$  e  $y_2$ ).



(b)- Acções de controlo ( $u_1$  e  $u_2$ ).



(c)- Estados reais ( $h$ ) e do modelo neuronal ( $x_n$ ).

Figura 5.3-8: Controlador adaptativo neuronal.

Da análise das experiências efectuadas verifica-se que apenas a metodologia adaptativa cumpre o seu papel de uma forma satisfatória, verificando-se para o caso do controlador fixo (realimentação de variáveis de estado) a existência de erro em regime final, facto já verificado na experiência apresentada Figura 5.3-5(a). A existência de erro em regime final, Figura 5.3-7(a), é facilmente justificada, uma vez que o modelo dado por (5.3-2) e (5.3-3) é definido para uma configuração (5.3-1) que não tem em conta a abertura da válvula  $v_{10}$ .

Relativamente ao controlador adaptativo *RSRN*, a sua capacidade de adaptação permite-lhe efectuar convenientemente o controlo do processo, compensando o efeito que provoca na dinâmica do processo a abertura da válvula.

Quantifica-se na Tabela 5.3-4 o desempenho de cada um dos controladores, linear por realimentação de variáveis de estado e *RSRN*.

Critério	Linear	RSRN
$\Sigma e_{n_1}^2$	0.2048	0.1506
$\Sigma e_{n_2}^2$	0.0793	0.0730
$\Sigma \Delta u_1^2$	1.3760	1.6760
$\Sigma \Delta u_2^2$	1.8257	1.9573

Tabela 5.3-4: Comparação das técnicas de realimentação de estados e *RSRN*.

Da sua análise salienta-se por um lado o erro de controlo superior para a saída  $y_1$  (onde é introduzida a perturbação) do que para a segunda saída,  $y_2$ , em ambos os controladores. Ainda em relação a este aspecto referem-se os resultados superiores, expressos pelo menor critério de erro, proporcionado pela metodologia de controlo neuronal, como já tinha sido observado.

No entanto, a metodologia neuronal conduz a um ligeiro aumento de esforço de actuação (bombas), visível no critério  $\Sigma \Delta u^2$ .

### 5.3.5 Conclusão

A ideia principal a reter destas experiências não é tanto a comparação dos resultados que resultaram das duas metodologias, linear e neuronal, mas sim dos aspectos que estão na base de cada uma delas para obter os mesmos resultados.

Assim, os resultados conseguidos pela primeira metodologia justificam-se pelo tipo de informação utilizada: é baseada no conhecimento dos estados do processo e na sua descrição segundo um modelo de primeiros princípios.

Os resultados alcançados pela abordagem neuronal devem-se principalmente à estrutura utilizada e à sua capacidade de adaptação. Através destas é possível capturar por um lado a dinâmica do processo e por outro proceder ao projecto de um controlador.

Apesar dos resultados terem sido, até certo ponto semelhantes, neste processo em particular, a questão que se coloca é o que sucederia caso a informação disponível fosse apenas informação entrada-saída, o que na verdade acontece a maioria dos processos reais. Naturalmente inviabilizaria a primeira metodologia, reconhecendo-se contudo o aspecto genérico da metodologia *RSRN*.

Concluindo, este caso de estudo permitiu verificar a utilidade e eficácia de modelos neuronais inseridos em estruturas adaptativas de controlo, para sistemas cuja modelização não possa ser obtida com exactidão, mas que se reconhece as vantagens de controlo de técnicas de realimentação de variáveis de estado.

# 5.4

## Conclusão

Neste capítulo foram efectuados estudos experimentais e de simulação no sentido de validar a metodologia de controlo desenvolvida *RSRN*, quando aplicada a sistemas com características distintas.

Assim, o primeiro caso de estudo foi fundamentalmente orientado para investigar aspectos relativos à inicialização de parâmetros correspondentes aos procedimentos de identificação e de controlo que caracterizam a metodologia *RSRN*. Examinou-se, além disso, a sua capacidade de adaptação face a variações, quer no regime de operação, quer na dinâmica do processo. Pela sua versatilidade em considerar este tipo de variações, usou-se o processo térmico *PT326*, um processo laboratorial de reduzidas dimensões e de características interessantes. Comprovou-se experimentalmente a eficácia do sistema de controlo neuronal *RSRN*, em comparação com uma técnica de controlo linear *PI*.

O segundo caso de estudo teve por objectivo principal investigar a capacidade da metodologia *RSRN* em lidar com processos não lineares, de que é o processo químico *CSTR* um *benchmark*. O desempenho do controlador foi comparado com um controlador *PI*, sem dúvida o mais utilizado na indústria para este tipo de processos.

No terceiro caso de estudo, é usado o processo laboratorial *DTS200*, tendo o objectivo principal sido investigar o desempenho da estratégia de controlo aplicada a sistemas multi-variáveis. Tem ainda o propósito de mostrar a capacidade do controlador adaptativo *RSRN* em desempenhar as funções para o qual foi desenvolvido, isto é, o projecto de controladores baseados em modelos de estados, sem na verdade assumir a sua acessibilidade, mas apenas informação de entrada-saída. O desempenho da metodologia proposta foi comparado com um controlador convencional, projectado a partir de um conhecimento detalhado do processo (acessibilidade de estados e modelo de primeiros princípios).

Efectuando uma análise geral aos três casos de estudo considerados, e com base nos resultados obtidos, pode-se concluir acerca das capacidades da metodologia adaptativa *RSRN* para o controlo de sistemas não lineares multi-variáveis, variantes no tempo e sujeitos a perturbações.

Uma vez que é uma metodologia genérica que tem em consideração o aspecto importante da estabilidade e ainda porque requer meios computacionais aceitáveis, reconhece-se a viabilidade da sua aplicação prática no controlo de sistemas industriais.

# 6.

## Conclusão e Perspectivas

*“How new information concerning the unknown plant is to be collected and stored, how it is to be used to detect and classify the environment at any instant of time, and how the controller architecture is to be organized for fast, accurate and stable response, are the subjects of ongoing research.”*

K. Narendra, 1998.

**A** crescente complexidade dos processos industriais e a procura em responder às suas exigências tem motivado a comunidade do controlo na procura de novas teorias e soluções.

A par destes desafios, as capacidades dos sistemas computacionais tem evoluído de uma forma impressionante. É hoje em dia possível introduzir, a custos reduzidos, computadores em sistemas de controlo, com elevadas capacidades de cálculo e capazes de processar e armazenar grandes quantidades de informação. Torna-se assim viável a implementação de algoritmos sofisticados, capazes de utilizar, cada vez mais eficazmente, toda a informação disponível.

Neste contexto, a aplicação de técnicas de *soft computing* tem tido, sem dúvida, um impacto significativo. E, se durante o passado recente têm, por um lado, sido desenvolvidos esforços no sentido de compreender as possibilidades destas metodologias e de permitir a sua incorporação em estruturas de controlo, por outro, a investigação de

teorias e métodos convencionais baseados em modelos matemáticos, tem também alcançado resultados significativos.

Resulta daqui, que o controlo automático enfrenta presentemente os seguintes paradigmas fundamentais: o integro-diferencial, que tem os seus princípios em teorias convencionais, e o paradigma da informação, distinguindo-se neste caso aspectos quantitativos e qualitativos, para os quais as técnicas de *soft computing* se mostram particularmente adequadas. Assiste-se actualmente a esforços no sentido de estabelecer pontes entre os vários paradigmas e de desenvolver abordagens unificadoras. A concepção de sistemas de controlo híbrido, que combinem métodos convencionais com ferramentas de *soft computing*, pode dar uma contribuição para o estabelecimento destas teorias unificadoras. Esta dissertação pretendeu caminhar nesta direcção.

# 6.1

## Conclusão

Desde as suas origens, as técnicas de *soft computing* têm mostrado potencialidades e o seu sucesso tem sido inquestionável na solução prática de problemas para os quais os controladores convencionais se têm revelado menos adequados. Contudo, e apesar dos avanços entretanto operados, a investigação destas técnicas encontra-se ainda numa fase de maturação não existindo ainda resultados bem estabelecidos, tal como existem para os sistemas de controlo convencionais. Motivados pelas suas potencialidades, pretendeu-se desenvolver nesta dissertação sistemas de controlo que incorporassem técnicas de *soft computing* mas, que ao mesmo tempo, tivessem em conta aspectos bem estabelecidos da teoria de controlo convencional.

Um conceito que tem dominado a teoria do controlo é a designada teoria de controlo com base num modelo. De facto, os métodos de identificação e controlo convencionais baseiam-se na existência de um modelo matemático que, de alguma forma, descreve o sistema que se pretende controlar. É neste contexto que se evidenciam as capacidades das redes neuronais. Desta forma, as redes neuronais não devem ser vistas como uma alternativa mas sim como um complemento, isto é, uma extensão, às técnicas convencionais de controlo.

Uma das características essenciais das redes neuronais, que fundamentou a sua inclusão neste trabalho foi, sem dúvida, a sua generalidade. A sua utilização permite estabelecer uma abordagem sistemática para o projecto de um controlador: numa primeira fase é desenvolvido um modelo neuronal, com base em informação entrada-saída e, numa segunda etapa, usando técnicas de controlo não linear, é efectuado o projecto de um controlador baseado no modelo desenvolvido. Torna-se assim possível desenvolver sistemas de controlo genéricos, de forma a poderem ser encarados com viabilidade de aplicação prática em processos industriais.

## Objectivos Alcançados

O objectivo desta dissertação consistiu no desenvolvimento de um sistema de controlo adaptativo para sistemas não lineares. Naturalmente que o trabalho realizado representou apenas uma tentativa de contribuir nesta direcção, no entanto, as soluções propostas conseguem, assim se julga, cumprir os objectivos inicialmente traçados.

Assim, abordou-se no capítulo 2 o problema da aplicação das redes neuronais à modelização de processos. Entre outras, foram revistas questões relativas a estruturas

neuronal e respectivos algoritmos de estimação de parâmetros. Deu-se especial atenção às redes recorrentes, uma vez que se mostram particularmente adequadas à modelização de processos dinâmicos. Em particular, propôs-se uma estrutura neuronal recorrente do tipo *affine* como uma tentativa de estabelecer um compromisso entre dois objectivos fundamentais: por um lado, a sua aplicação ao problema de modelização de sistemas não lineares e, por outro, facilitar o desenvolvimento de técnicas de estimação de parâmetros e de projecto de controladores.

No capítulo 3 tratou-se o problema da aplicação das redes neuronais ao controlo. Foram revistos aspectos genéricos das técnicas adaptativas e de estruturas de controlo que incluem redes neuronais, distinguindo-se fundamentalmente duas classes: uma em que a rede neuronal é ela própria um controlador e outra, em que a rede serve como modelo ao projecto de um controlador. Foram introduzidos conceitos de controlo não linear e, em particular, a técnica de linearização retroactiva, a qual é utilizada na maioria dos trabalhos que combinam conceitos rigorosos de controlo com redes neuronais.

No capítulo 4 propôs-se a estrutura de controlo *RSRN- Regulação da Saída com Redes Neuronal Recorrentes* combinando-se uma rede neuronal recorrente com a teoria da regulação da saída, uma técnica de controlo não linear. A partir desta abordagem foi proposto um algoritmo que considera uma estimação simultânea de parâmetros e de estados, usando conceitos da teoria de estabilidade de Lyapunov e da teoria de observação não linear. Foi revista a teoria de regulação da saída e, tirando partido da especificidade da rede recorrente, foi proposto um algoritmo capaz de produzir uma solução para o problema de regulação.

## Casos de Estudo

O trabalho prático desenvolvido e apresentado no capítulo 5 mostrou que as redes neuronais recorrentes têm um elevado potencial, não só em tarefas de modelização de processos, mas também para serem incorporadas em sistemas de controlo. Relativamente aos resultados práticos das experiências efectuadas, estas permitiram concluir sobre as capacidades da metodologia *RSRN* em sistemas de controlo, nomeadamente:

- i) aspectos de generalidade e sistematização, uma vez que a metodologia foi possível de aplicar de uma forma sistemática a processos com diferentes características;
- ii) capacidade da metodologia em ser aplicada a processos não lineares, de que foi exemplo o processo químico *CSTR*;
- iii) capacidade de adaptação face a variações, quer no regime de operação, quer na dinâmica do processo, principalmente verificadas nas experiências realizadas no processo térmico;
- iv) capacidade da metodologia em ser aplicada a sistemas multi-variáveis, de que foi exemplo o processo dos três tanques acoplados.

# 6.2

## Perspectivas

A realização de um trabalho deste tipo dá naturalmente origem a várias ideias. E se algumas delas foram assunto de investigação, outras há que permanecem em aberto e consequentemente darão continuidade a este trabalho.

Relativamente ao trabalho realizado, referem-se alguns aspectos que se prendem com a modelização, a estimação, o controlo e a estabilidade.

### Modelização

A rede neuronal utilizada caracteriza-se por constituir uma estrutura *affine*, para a qual não se conhecem ainda resultados de aproximação. Poder-se-á pois, no futuro, desenvolver esforços no sentido de demonstrar as suas capacidades de aproximação.

Idealmente um sistema de controlo deve permitir, face a alterações nas condições do ambiente e do próprio processo, ajustar os parâmetros de modo a reagir adequadamente a essas alterações. Se a implementação de um mecanismo de adaptação soluciona em parte o problema, por outro reconhece-se a conveniência da substituição do conceito adaptativo pelo de aprendizagem. Assim, a utilização de técnicas de estimação, em que o controlador de alguma forma “*aprende*” e não esquece, deve ser equacionada. Neste contexto, o uso de redes do tipo (*RBFNN*) poderá constituir um ponto de partida para uma investigação futura. Devido à sua localidade, as redes *RBFNN* permitem evitar questões de conflitos de dados, permitindo efectuar um ajuste de parâmetros em regiões do espaço sem no entanto afectar outras regiões.

### Estimação de Parâmetros

Em relação ao método de estimação de parâmetros, garante-se apenas a convergência do erro de estimação, especificado a partir da função de Lyapunov considerada.

$$V(\boldsymbol{\varepsilon}(k)) = \boldsymbol{\varepsilon}(k)^T \mathbf{P} \boldsymbol{\varepsilon}(k),$$

Poder-se-á no futuro investigar a extensão desta última função por uma função do tipo,

$$V(\boldsymbol{\varepsilon}(k)) = \boldsymbol{\varepsilon}(k)^T \mathbf{P} \boldsymbol{\varepsilon}(k) + \text{tr}\{ \mathbf{W}(k)^T \mathbf{W}(k) \}$$

tal como é considerado, por exemplo, nos trabalhos de (§Rovithakis *et al.*, 1996), (§Poznyak *et al.*, 1999). Desta forma é possível ter em conta a limitação dos valores dos pesos das matrizes de ligação, que se pode relacionar com a garantia da qualidade dos modelos neuronais. Por outras palavras, ao efectuar-se a estimação em-linha dos parâmetros da rede, não se garante que os aspectos relativos à dinâmica do modelo neuronal inicialmente obtido não sejam degradados. Apesar deste aspecto não ser particularmente importante para a metodologia considerada, poder-se-ia mostrar relevante caso o modelo neuronal fosse inserido numa estrutura de controlo preditivo.

Embora tenham sido investigados, e mesmo implementados na prática, aspectos relativos a teorias robustas de observação (*Sliding Mode Observers*), estes acabaram por não merecer atenção especial nesta dissertação. Formulou-se no entanto o problema (secção 4.2) e conclui-se sobre a necessidade da resolução de um sistema de equações não lineares, para o qual não foi possível dar solução. Julga-se que a inclusão destas teorias, adequadas particularmente a incertezas, será um aspecto a ponderar seriamente.

## Controlo e Estabilidade

A primeira questão de interesse a investigar futuramente reside nas limitações impostas. Consideram-se apenas sistemas cujo número de entradas é igual ao número de saídas, restrição cuja eliminação poderá constituir tema de investigação futura. Refere-se neste contexto o trabalho de (§Castillo-Toledo *et al.* 1993b).

Relativamente à metodologia particular desenvolvida espera-se poder, muito em breve, aplicá-la ao sistema de colectores solares existente na Plataforma Solar de Almería, estando já a ser desenvolvidos esforços para o efeito. Validar-se-á assim a sua eficácia em sistemas à escala industrial, esperando-se reforçar significativamente a sua viabilidade de aplicação a processos industriais.

O problema da estabilidade é nesta dissertação tratado admitindo, além das condições para os teoremas 1 e 2, o princípio de separação, sendo este último, no entanto, válido apenas para sistema lineares. Recorde-se que a estrutura da regulação da saída permite a aplicação do um princípio da separação, pois a acção de controlo é tida em conta directamente para caracterizar a estabilidade em malha fechada, o que permite ir mais além do que o princípio de separação básico.

Existem contudo trabalhos, de que é exemplo o de Lin e Byrnes (§1995), que estabelecem resultados precisos para a existência do princípio de separação, caso o observador cumpra determinados requisitos, nomeadamente, que se caracterize por ser exponencialmente estável. Poder-se-á pois avançar neste sentido, averiguando em que condições se poderá garantir que a rede neuronal constitua ela própria um observador com estas características.

## Comentários Finais

A evolução verificada nas tecnologias de informação e o constante desenvolvimento das capacidades computacionais tem proporcionado os meios para o desenvolvimento e aplicação prática de novas teorias e novos métodos. É hoje em dia possível desenvolver algoritmos sofisticados, originando novas perspectivas e desafios nos diversos campos da engenharia, designadamente no domínio do controlo automático.

O conceito de *soft computing*, emergiu como uma associação de metodologias de computação, capazes de proporcionar as bases para a concepção e projecto de sistemas e através delas, criar conhecimento para a tomada de decisão a partir de grandes quantidades de dados e de diferentes tipos de informação, metodologias estas possíveis de serem por isso aplicadas a problemas de controlo, modelização e decisão em sistemas complexos.

O controlo automático deixa de ser apenas abordado segundo o ponto de vista quantitativo, a que esteve associado desde as suas origens, e passa a incorporar também aspectos qualitativos. Surge o conceito de *Controlo Inteligente*, intimamente ligado ao conceito de *soft computing* e às metodologias que o constituem, entre as quais os sistemas difusos e as redes neuronais.

O surgimento destas novas metodologias a par dos desenvolvimentos efectuados na teoria convencional originou, naturalmente, alguma competição entre elas, constituindo os desenvolvimentos daí resultantes oportunidades únicas para a evolução do controlo automático. Assiste-se actualmente a esforços no sentido de estabelecer pontes entre elas, de forma a desenvolver abordagens unificadoras que tirem partido das sinergias das várias metodologias.

Torna-se cada vez mais evidente que o maior desafio que se coloca actualmente à comunidade do controlo consiste no desenvolvimento de ferramentas que permitam não só explorar as poderosas capacidades computacionais, mas também, que potenciem a sua utilização eficaz no processamento de toda a informação disponível.

A forma como a informação recolhida sobre o processo é armazenada e, principalmente, como é usada para classificar tanto o processo como o ambiente e como a partir dela é extraído conhecimento de forma a tomar decisões, ditarão os avanços futuros no controlo automático.



# Apêndice A

Apresentam-se neste apêndice os teoremas de estabilidade de Lyapunov, o princípio de invariância de La Salle e alguns resultados relativos a normas de vectores e matrizes e suas propriedades fundamentais.

## A.1 Estabilidade de Lyapunov e Princípio de Invariância de La Salle

(Slotine e Li, 1991)

### Definição

Uma função contínua  $V(x)$  diz-se definida positiva se

$$\begin{aligned} V(0) &= 0 \\ V(x) &> 0 \text{ se } x \neq 0 \end{aligned} \tag{A.1-1}$$

Uma função contínua  $V(x)$  diz-se semi-definida positiva se

$$\begin{aligned} V(0) &= 0 \\ V(x) &\geq 0 \text{ se } x \neq 0 \end{aligned} \tag{A.1-2}$$

### A1.1 Estabilidade Segundo Lyapunov

#### Estabilidade Local

Se para uma dada região existir uma função escalar  $V(x)$ , com derivadas parciais contínuas, tal que

- i)  $V(x)$  é positiva definida,  $V(x) > 0$ ,
- ii)  $\dot{V}(x)$  é negativa semi-definida,  $\dot{V}(x) \leq 0$ ,

então o ponto de equilíbrio  $x = 0$  é estável. Se a condição ii) for substituída por

- ii)  $\dot{V}(x)$  é negativa definida,  $\dot{V}(x) < 0$ ,

então o ponto de equilíbrio  $x = 0$  diz-se assintoticamente estável.

#### Estabilidade Global

Para uma função escalar  $V(x)$ , com derivadas parciais contínuas, a origem ( $x = 0$ ) é assintoticamente estável de uma forma global se

- i)  $V(x)$  é positiva definida,  $V(x) > 0$ ,
- ii)  $\dot{V}(x)$  é negativa definida,  $\dot{V}(x) < 0$ ,
- iii)  $V(x) \rightarrow \infty$  se  $\|x\| \rightarrow \infty$ .

**Nota:**

As definições apresentadas para sistemas contínuos são válidas para sistemas definidos no domínio discreto substituindo a derivada na segunda condição (ii)

$$\dot{V}(x) \text{ por } \Delta V(x(k)) = V(x(k)) - V(x(k-1))$$

### A.1.2 Princípio de Invariância de La Salle

No caso em que a função de Lyapunov  $V(x)$  é caracterizada por  $\dot{V}(x) \leq 0$  (semi-definida) não é possível estabelecer conclusões acerca de estabilidade assintótica. Neste caso é útil o princípio de invariância de La Salle:

Seja  $V(x)$  uma função escalar com derivadas parciais contínuas. Assuma-se que para uma certa região  $\Omega$  da origem

- i)  $V(x)$  é positiva definida (localmente),  $V(x) > 0$ ,
- ii)  $\dot{V}(x)$  é negativa semi-definida,  $\dot{V}(x) \leq 0$ ,
- iii) o conjunto definido por  $\dot{V}(x) = 0$  contém apenas como solução a trajectória  $x = 0$ .

Então o ponto de equilíbrio  $x = 0$  é assintoticamente estável (localmente).

## A.2 Normas

(Kahlil, 1996)

### A.2.1 Definições

Por definição a norma,  $\|\cdot\|$ , de um vector  $x \in \mathfrak{R}^n$  é um valor real positivo que satisfaz:

- i)  $\|x\| > 0$ .
- ii)  $\|x\| = 0$  se e só se  $x = 0$ .
- iii)  $\|x + y\| \leq \|x\| + \|y\| \forall x, y \in \mathfrak{R}^n$ .
- iv)  $\|\alpha x\| = |\alpha| \|x\|$  com  $\alpha \in \mathfrak{R}$  e  $x \in \mathfrak{R}^n$ .

#### Vectores

Uma classe de normas, que verifica as condições anteriores, é a que se define por:

$$\|x\|_p = (\|x_1\|^p + \|x_2\|^p + \dots + \|x_n\|^p)^{1/p} \quad (\text{A.1-3})$$

e

$$\|x\|_\infty = \max_i \|x_i\| \quad (\text{A.1-4})$$

As normas mais comuns são definidas para os casos de  $p = 1$  e  $p = 2$ , esta última conhecida também por norma Euclidian.

$$\|x\|_1 = \|x_1\| + \|x_2\| + \dots + \|x_n\| = \sum_{i=1}^n \|x_i\| \quad (\text{A.1-5})$$

$$\|x\|_2 = (\|x_1\|^2 + \|x_2\|^2 + \dots + \|x_n\|^2)^{1/2} = (x^T x)^{1/2} = \sqrt{x^T x} \quad (\text{A.1-6})$$

#### Matrizes

Uma matriz  $A \in \mathfrak{R}^{m,n}$  define um mapeamento linear  $y = Ax$  de  $x \in \mathfrak{R}^n$  para  $y \in \mathfrak{R}^m$ . A norma induzida da matriz  $A$  é, por definição:

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \quad (\text{A.1-7})$$

em que  $\sup$  denota supremo.

Casos particulares de  $p = 1, 2, \infty$ .

$$\|A\|_1 = \max_j \sum_{i=1}^m \|a_{ij}\| \quad (\text{A.1-8})$$

$$\|A\|_2 = (\lambda_{\max}(A^T A))^{1/2} = \sqrt{\lambda_{\max}(A^T A)} \quad (\text{A.1-9})$$

$$\|A\|_\infty = \max_i \sum_{j=1}^n \|a_{ij}\| \quad (\text{A.1-10})$$

Em (A.1-9)  $\lambda_{\max}(A^T A)$  denota o máximo dos valores próprios da matriz  $(A^T A)$ .

Define-se

$$\sigma_i^2 = \lambda_i(A^T A) \quad (\text{A.1-11})$$

em que  $\lambda_i$  define um valor próprio e  $\sigma_i$  um valor singular. Desta definição

$$\sigma_{\max}^2 = \lambda_{\max}(A^T A) \quad (\text{A.1-12})$$

$$\sigma_{\max} = \|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

A norma de Frobenius é definida por

$$\|A\|_F = \sqrt{\text{tr}(A^T A)} = \sqrt{\sum_{i,j} a_{ij}^2} \quad (\text{A.1-13})$$

$$\|A\|_F^2 = \sum_i \sigma_i^2 \quad (\text{A.1-14})$$

em que  $\text{tr}(A)$  denota o traço da matriz  $A$ .

## A.2.2 Propiedades

### Vectores

$$\|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2 \quad (\text{A.1-15})$$

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n} \|x\|_\infty \quad (\text{A.1-16})$$

$$\|x\|_\infty \leq \|x\|_1 \leq n \|x\|_\infty \quad (\text{A.1-17})$$

### Matrices

$$\|A B\|_p \leq \|A\|_p \|B\|_p \quad (\text{A.1-18})$$

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \quad (\text{A.1-19})$$

$$\frac{1}{n} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{m} \|A\|_\infty \quad (\text{A.1-20})$$

$$\frac{1}{m} \|A\|_1 \leq \|A\|_2 \leq \sqrt{m} \|A\|_\infty \quad (\text{A.1-21})$$

### Matrices Positivas definidas (P)

$$\|P\| \leq \lambda_{\max}(P) \quad (\text{A.1-22})$$

$$\|P\| \geq \lambda_{\min}(P) \quad (\text{A.1-23})$$

$$x^T P x \leq \lambda_{\max}(P) x^T x = \lambda_{\max}(P) \|x\|_2^2 \quad (\text{A.1-24})$$

$$x^T P x \geq \lambda_{\min}(P) x^T x = \lambda_{\min}(P) \|x\|_2^2 \quad (\text{A.1-25})$$

# Apêndice B

Nesta dissertação procurou-se, desde o seu início, o desenvolvimento de arquitecturas híbridas que pudessem combinar as capacidades de técnicas ditas inteligentes com resultados bem conhecidos de teoria de controlo, tendo nesse sentido, além de estudos teóricos, sido conduzidos vários estudos experimentais.

Apresentam-se neste apêndice alguns dos trabalhos desenvolvidos ao longo desta dissertação que em parte contribuíram para esse fim.

## B.1

*Experimental On-Line Learning For a Benchmark Process,*  
Henriques, J., J. Vitor, C. Pereira, A. Dourado,  
*CONTROLO98, 3rd Portuguese Conference on Automatic Control, 193-198,*  
Coimbra, Portugal, 1998.

## B.2

*Supervision And C-Means Clustering of PID Controllers For A Solar Power Plant,*  
Henriques, J., A. Cardoso e A. Dourado,  
*International Journal of Approximate Reasoning, Vol. 22, pp. 73-91,*  
1999.

## B.3

*Application of A Recurrent Neural Network In On-Line Modelling Of Real-Time Systems,*  
Henriques, J., P. Gil, A. Dourado, H. Ramos,  
*ESIT99 – European Symposium on Intelligent Techniques,*  
Crete, Grécia, 1999.

## B.4

*Multivariable Adaptive Control Using An Observer Based On A Recurrent Neural Network,*  
Henriques, J., A. Dourado  
*International Journal of Adaptive Control and Signal Processing, Vol. 13, pp. 241-259, 1999.*

**B.5***Non-Linear Multivariable Predictive Control: Neural versus First Principle Modelling Approach*

Henriques, J., P. Gil, A. Dourado

Submitted to the Journal of Systems and Control Engineering,  
1999.

No primeiro deles (B.1) são realizadas comparações experimentais entre metodologias adaptativas de controlo, que incorporam na sua estrutura modelos difusos ou modelos neuronais. Foi objectivo principal investigar as vantagens e desvantagens de cada uma das metodologias.

No segundo trabalho (B.2) apresenta-se uma estrutura de controlo em que se propõe um sistema difuso a funcionar *em-linha* como supervisor a um conjunto de controladores *PID*, previamente sintonizados recorrendo a redes neuronais. Concilia-se assim na mesma abordagem aspectos convencionais (controladores *PID*) com técnicas de inteligência artificial (sistemas difusos e redes neuronais). Salienta-se ainda a sua aplicação prática a um processo de escala industrial.

No terceiro trabalho (B.3) investiga-se experimentalmente, com algum pormenor, as capacidades das redes neuronais recorrentes quando aplicadas na modelização *em-linha* de processos dinâmicos.

Assumindo a existência de um modelo neuronal recorrente, capaz de capturar a dinâmica do processo a controlar, considera-se nos restantes trabalhos o problema de projecto de um controlador com base no modelo não linear obtido. Em todos eles, além da investigação teórica efectuada, refere-se a sua validação experimental em processos laboratoriais.

Assim, no quarto trabalho (B.4) considera-se a possibilidade de aplicação da teoria de controlo linear. Para o efeito considera-se uma linearização local do modelo neuronal não linear, interpreta-se este último como um observador, e empregam-se métodos de controlo por realimentação de variáveis de estado.

No último trabalho (B.5) são investigadas as capacidades de preditivas de modelos neuronais recorrentes, em particular incorporados como modelo de predição em esquemas de controlo preditivo.

## EXPERIMENTAL ON-LINE LEARNING FOR A BENCHMARK PROCESS

J. Henriques, J. Victor, C. Pereira and A. Dourado

CISUC - Centro de Informática e Sistemas da Universidade de Coimbra  
Departamento de Engenharia Informática  
Pinhal de Marrocos, 3030 Coimbra, Portugal  
jh, zevitor, cpereira, dourado@dei.uc.pt

**Abstract:** Three controllers based on neural networks and fuzzy systems are developed and implemented for an experimental benchmark process. They are tested under several conditions in order to assess their learning capabilities. To evaluate their behaviour some performance indexes are used. The experimental results show that this kind of controllers, included in the broader class of intelligent controllers, are promising techniques.

Keywords: Intelligent control, On-line learning, Fuzzy systems, Neural networks, Experimental evaluation.

### 1. INTRODUCTION

Three controllers based on neural networks and fuzzy systems are developed and implemented on an experimental benchmark process composed of PT 326 and PCS 327 (Feedback, 1985). The PT 326 is a nonlinear process with variable time constant and variable transport lag. This type of process is found in many industrial systems such as furnaces or air conditioning. The PCS 327 is a process simulator used to introduce additional poles and pure time delays on process PT 326. The following controllers are implemented and compared:

*i)* A neural RBF (Radial Basis Function) controller with on-line learning of radial function centres and output layer weights. The centres of the gaussian functions are on-line learned with adaptive determined rate and the weights of the output layer evaluated by a recursive least squares algorithm with directional forgetting factor.

*ii)* A hybrid neural controller (HNC) (observer based on a recurrent Elman's Neural Network + pole placement). The neural network acts as a state observer on the non-linear process and the observed states are used in a pole placement control algorithm. The

network parameters are trained on-line and used by the adaptive control algorithm.

*iii)* A fuzzy adaptive controller (modified SOC). The control strategy is modified according to changes in the process or in the control environment by adaptation of the consequent of each rule, considering the value of a performance measure. The mechanism of adaptation in each iteration can be carried out for all the fired rules or only for those that have fired above a certain threshold.

This paper is organised as follows. In section 2 are described the studied control methodologies. In section 3, for several conditions, experimental results for the benchmark process are presented. The potentiality of the different controllers is evaluated by using some performance indexes. Finally in section 4 some comments and conclusions are presented.

## 2. CONTROL METHODOLOGIES

The system to be controlled, is assumed to be a SISO (Single Input, Single Output) described by a non linear input/output discrete time model, in which the output  $y(k)$ , at a particular discrete time  $k$ , can be defined as a non linear function of past outputs  $y(k-1)$ ,  $y(k-2)$ ,... and past inputs  $u(k-1)$ ,  $u(k-2)$ , ...

$$y(k) = \varphi \{y(k-1), y(k-2), \Lambda, u(k-1), u(k-2), \Lambda\} \quad (1)$$

An alternative representation is the use of a discrete state space model

$$X(k) = f [X(k-1), u(k-1)] \quad (2a)$$

$$y(k) = h [X(k)] \quad (b)$$

where  $f$  and  $h$  are non linear functions and  $X(k)$  is the state vector.

### 2.1 Neural RBF controller

In most generic sense, a RBFNN is any network that has radial symmetric activation functions. The output of a hidden neuron is a function of the distance between an input vector and the centre of the function.

Given an input vector of dimension  $p$   $V \in \mathfrak{R}^p$ , the output of the network  $\hat{y}$  is described by:

$$\hat{y} = \sum_{i=1}^m w_i \varphi_i (\|V - c_i\|), \quad c_i \in \mathfrak{R}^p \quad (3)$$

in which  $\omega_i$  ( $i = 1, 2, \dots, m$ ) are the network weights,  $\|\cdot\|$  denotes de Euclidean norm,  $c_i$  are the basis function centres and  $\varphi_i$  is the radial basis function. This function can be selected in one of many different ways. In this work the Gaussian exponential function is used (4), where  $\sigma_i$  defines the width of the receptive field.

$$\varphi_i : \mathfrak{R}^p \rightarrow \mathfrak{R}, \varphi_i = \exp\left(-\frac{\|V - c_i\|^2}{\sigma_i^2}\right) \quad (4)$$

In the proposed on-line learning procedure (Pereira *et al.*, 1996), all the parameters of the NN are identified on-line in two steps. The first step applies the recursive K-means clustering in a modified version of the technique proposed by Moody and Darken (Moody and Darken, 1989), which is essentially a competitive learning algorithm of Kohonen type. In this work it is introduced a monitoring procedure to update the learning rate. In the second step the selective forgetting factor is applied to the estimation of the weights of the output layer.

*Calculation of the centres and widths of the Gaussian functions with adaptive learning rate.* Since at each period of time the network input exists only in some regions of the input space, it is reasonable to allocate the centres of the Gaussian functions in that regions, using a clustering algorithm, like for example the K-means clustering. A modified recursive version is

applied here. The initial centres are chosen randomly. The following steps compose the method:

*Step 1.* Read the next input vector,  $V(k)$ .

*Step 2.* Modify only the closest centre according to

$$c_i(k) = c_i(k-1) + \beta(k) [V(k) - c_i(k-1)] \quad (5)$$

*Step 3.* Compute the sum of squared error (SSE) for the last  $N$  sampling periods:

$$SSE = \sum_{k=1}^N e(k)^2 = \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \quad (6)$$

*Step 4.* If the SSE is greater than  $\epsilon_c$  then:

$$\beta(k+1) = \beta_c \quad (7)$$

otherwise:

$$\beta(k+1) = 0.9\beta(k) \quad (8)$$

where  $\beta(k)$  represents the learning rate,  $0 < \beta(k) < 1$ ,  $\beta_c$  is a positive constant value for the learning rate. In case the modelling error over a period of time (window of  $N$  samples) is greater than a prefixed constant  $\epsilon_c$ , the adaptive learning rate remains constant, otherwise it decreases slowly to zero. The widths of the Gaussian functions are determined by the "P-Nearest Neighbour" method (Moody and Darken, 1989).

*Solving the Weight Matrix.* Due to the linearity of the error function with respect to the weights, the weight matrix can be solved such that the error is minimised in the sense of recursive least squares algorithm. However if the input is sufficiently exciting, the elements of the covariance matrix converge to zero, loosing the tracking capability if the system is time-variant. A selective forgetting algorithm solves this problem and guarantees that the covariance matrix is bounded from above and below.

*Control Loop.* The learning procedure described is used to train the neural RBF controller. The control structure proposed is shown in Fig. 1, with monitoring and adaptive training. The RBFNN is used to model the inverse of the process. The training signal is the difference between the reference and the process output. The feedback signal is a measure of the accuracy of the inverse model.  $F$  is a first order filter to smooth the error signal, introducing robustness into the control system.

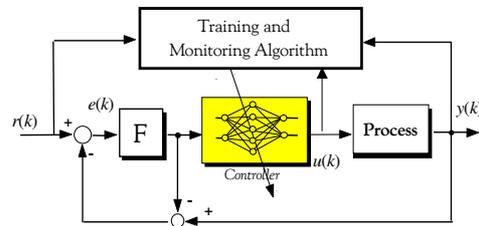


Fig. 1. Real-time adaptive RBF control structure

### 2.2 Hybrid Neural Controller

In this methodology an Elman's NN (Pham and Xing, 1995) is used for modelling a general non-linear process. An Elman's network is a recursive one, which is described by the following difference equation

$$\hat{X}(k) = \sigma[W^{xc}X^c(k), W^{xu}u(k-1)] \quad (9a)$$

$$X^c(k) = \alpha X^c(k-1) + \hat{X}(k-1) \quad (b)$$

$$\hat{y}(k) = W^{yx}\hat{X}(k) \quad (c)$$

where  $\sigma\{\cdot\}$  is a non-linear function (usually sigmoidal type) and  $\alpha$  is a constant (self-connection). The interconnection matrices are  $W^{xu} \in \mathfrak{R}^{n,p}$ ,  $W^{xc} \in \mathfrak{R}^{n,n}$  and  $W^{yx} \in \mathfrak{R}^{q,n}$ , respectively, the context-hidden layer, the input-hidden layer and the hidden-output layer. For more details see, (Henriques and Dourado, 1998).

*Learning methodology.* There are several training algorithms that have been proposed to adjust the DRNN weights: the real time recurrent algorithm from Williams, Narendra's dynamic backpropagation (BP), and the Werbos' BP through time (BTT) (Werbos, 1990), which is used in this work. However, due to the computational complexity of the BTT, a natural simplification is obtained by truncating the BP of information to a fixed number (N) of prior samples on a sliding window mode. The network is duplicated N samples in order to describe the behaviour on the considered time window, and standard BP algorithm can be applied through the global static network. In this work N=4, establishing a compromise between computational complexity and system order representation. The performance criterion to be minimised is the SSE (eq. 6), in the interval  $[k - NK, k]$ . The computation of the gradient of  $J^{total}$  is obtained after the network has been run through the interval  $[k - NK, k]$ . The total gradient is

$$\frac{\partial J^{total}[k - N, k]}{\partial W_{ij}} = \sum_{\tau=k-N+1}^k \delta_i(\tau) \hat{X}_j(\tau-1) \quad (10)$$

where the values  $\epsilon(\tau)$  and  $\delta(\tau)$  are computed recursively using the following equations

$$\epsilon_L(\tau) = e_L(\tau) \quad (11a)$$

$$\delta_L(\tau) = \sigma' [W^{xc}X^c(k), W^{xu}U(k-1)] \epsilon_L(\tau) \quad (b)$$

$$\epsilon_L(\tau-1) = e_L(\tau-1) + \sum_i W_{i\tau} \delta_L(\tau) \quad (c)$$

These equations represent the well-known BP in which target values are specified for units in all layers (L) and not only in the last layer.

*Controller Design.* The neural network is a non-linear model. However it is possible to derive a linear model by computing the variations from the outputs with respect to the inputs of the network.

$$\hat{X}(k) = A\hat{X}(k-1) + Bu(k-1) \quad (12)$$

$$\hat{y}(k) = C\hat{X}(k) \quad (b)$$

where the A, B and C are matrices defined by

$$A = \frac{\partial \sigma\{\cdot\}}{\partial \hat{X}(k-1)} = W^{xc} \sigma'\{\cdot\} \quad (13a)$$

$$B = \frac{\partial \sigma\{\cdot\}}{\partial u(k-1)} = W^{xu} \sigma'\{\cdot\} \quad (b)$$

$$C = W^{yx} \quad (c)$$

The controller uses only input/output knowledge described by the equation

$$A(q^{-1})y(k) = B(q^{-1})u(k) \quad (14)$$

where A and B are polynomials in the backward shift operator  $q^{-1}$ . The controller is a general FGH one described by:

$$F(q^{-1})u(k) = H(q^{-1})r(k) - G(q^{-1})y(k) \quad (15)$$

where  $r(k)$  is the reference, F, G and H are polynomials determined such that the closed loop poles (solution of  $AF + BG$ ) are placed in desired locations (Åström, 1995). Fig 3 shows the control structure.

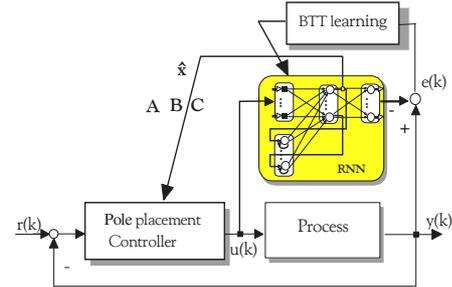


Fig. 2. Self-tuning control structure with a DRNN.

### 2.3 Modified SOC

The self-organising controller (SOC) was introduced by Procyk and Mamdani and is a well-known method of adaptation for fuzzy controllers. In Fig. 3 is presented the schematic representation of the SOC (Procyk and Mamdani, 1979).

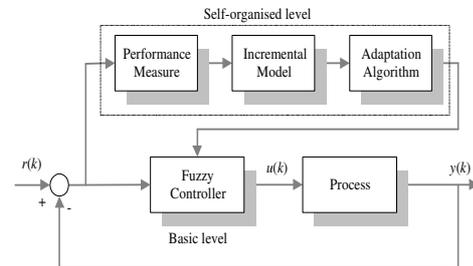


Fig. 3. Self-organising controller scheme.

The adaptation mechanism consists of a performance measure and an adaptive algorithm that modifies the rules of the fuzzy controller based on the performance measure, using an incremental process model.

*Performance measure.* The performance measure is similar to the fuzzy controller. They are based on the same inputs, the error and the change of error, except for the fact that the conclusions of the rules represents a performance measure instead of a control action. The numerical value of the performance measure  $p(k)$  is obtained from a decision table,

$$p(k) = f(e(k), \Delta e(k)) \quad (16)$$

where  $f$  is a look-up table. It represents a reference model of the process under control. The more the process deviates from the implicit reference model, the worse the performance is classified.

*Incremental process model.* Since the performance measure represents a measure of how to modify the controller output, a process model is needed to relate these changes to control signal variations. An incremental model  $M$  is determined by:

$$M = TJ \quad (17)$$

where  $J$  is the Jacobian of the process model and  $T$  is the sampling time. When  $M$  is obtained, the changes to the controller outputs have to be determined. The reinforcement values  $c(k)$  are calculated by means of the performance measure  $p(k)$  in combination with  $M$

$$c(k) = M^{-1} p(k) \quad (18)$$

*Adaptation algorithm.* When a fuzzy controller is regarded as a combination of a look-up table and an interpolation algorithm, a major simplification of the SOC is possible (Shao, 1988). The simplification consists of changing the consequent of the fired rules at sample  $k$ . Like in the original scheme there must be some knowledge about the time delays of the process in order to update the rules efficiently. The resulting controller can be interpreted as being based on Sugeno rules, with constant consequents and a weighted sum for defuzzification (Jager, 1995).

If the adaptation delay is  $m$  samples, the rule  $k$  that has been fired  $m$  samples ago, is described by (19)

$$\text{Rule } k: \text{ If } X_1 \text{ is } A_1 \text{ and } X_n \text{ is } A_n \quad (19)$$

$$\text{then } u(k) = u(k-m)$$

is changed to (20)

$$\text{Rule } k': \text{ If } X_1 \text{ is } A_1 \text{ and } X_n \text{ is } A_n \quad (20)$$

$$\text{then } u(k) = u(k-m) + c(k)$$

by the adaptation mechanism, where  $c(k)$  is given by eq. (18). It is possible to update all rules that fired  $m$  samples ago, or using a certain threshold above that the fired strength of a rule should be. There is also the possibility of establishing a confidence level between the old rule consequent and the reinforcement values in the rule update.

### 3. EXPERIMENTAL RESULTS

The three methods were applied to an airstream heating system and their performance evaluated and compared.

#### 3.1 Process PT326 and Process Simulator PCS327

The experimental problem considered involves the temperature control of a laboratory process, the PT 326, Fig. 4. Its function is similar to a hair dryer. Air is forced to circulate by a fan blower through a tube and heated at the inlet. This is a nonlinear process with a pure time delay, which depends on the position of the temperature sensor element and the air flow rate  $\Omega$ . The system input  $u(k)$ , is the voltage on the heating device, which consists of a mesh of resistor wires, and the output,  $y(k)$ , is the outlet air temperature. The PCS 327 is provided to operate as an external controller. It has also some additional analogue modules enabling the study of several effects: pure time delays, transfer lags and nonlinear effects in the control action.

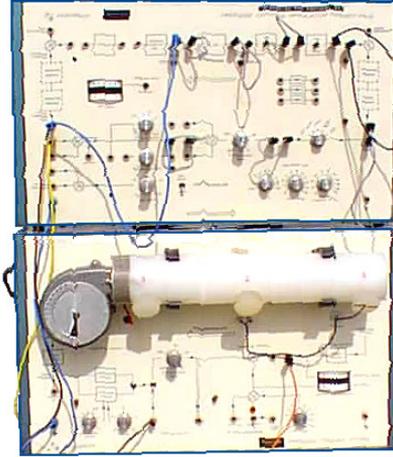


Fig. 4. The Laboratory Process PT 326 and PCS 327.

In all experiments, the following parameters were chosen:

*PT326:* Blower inlet throttle  $\Omega = 30^\circ$ , Sampling time: 150 ms, Detector probe in the position 279 mm.

*RBF Controller:*  $\alpha_{min} = 0.01$ ,  $\alpha_{max} = 0.1$ ,  $P(0) = 0.05$ ,  $I_6$ ,  $e_c = 1$ ,  $\beta(k) = 0.6$  and  $N = 20$ .

*HNC:* Matrices weights are chosen randomly in the interval  $[0, 1]$ . Activation functions: sigmoidal; BTT method: learning rate = 0.02, pole location at 0.91.

*Modified SOC:* The fuzzy PI controller included in the SOC structure has the following configuration: Number of linguistic terms: 7, Membership functions: triangular, Rule base: generic fuzzy PI (initially empty), Inference mechanism: max-min,

Defuzzification method: height, Scaling factors:  $K_e = 0.16$ ;  $K_{\Delta e} = 1.30$ ;  $K_{\Delta u} = 0.15$ . The value for constant  $m$  is set to 2.

The initial configuration was kept the same during the several experiments.

### 3.2 Experiments

Three groups of experiments were carried out in order to evaluate the performance of the controllers with respect to set point tracking, load and actuator disturbances and process dynamics change.

*Tracking Problem.* Let the reference be initially a square wave switching at sample 400 to a sinusoidal wave, with a period of 60 seconds (400 samples). Fig. 5, Fig. 6 and Fig. 7 show the process behaviour for the different controllers.

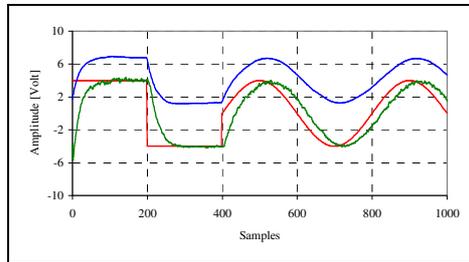


Fig. 5. Neural RBF Controller.

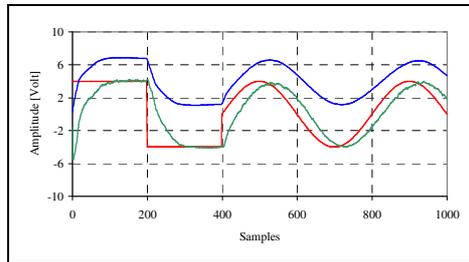


Fig. 6. Hybrid Neural Controller.

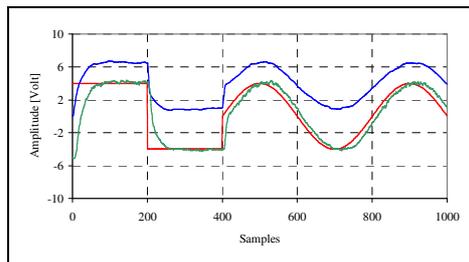


Fig. 7. Modified SOC.

As can be seen the three controllers performed considerably well in tracking the reference.

*Regulation Problem.* This experiment addressed the disturbance rejection problem. Variations were considered in the flow rate of input air:  $\Omega=10^\circ$  at sample 200 and reset to its initial value,  $\Omega=30^\circ$ , at sample 400. Between samples 600 and 700, 1.0 Volt is added to the control signal, and between samples 800 and 900 the same value is subtracted. The reference is a constant wave with amplitude 4.0 Volts. Fig. 8, Fig. 9 and Fig. 10 present the process behaviour.

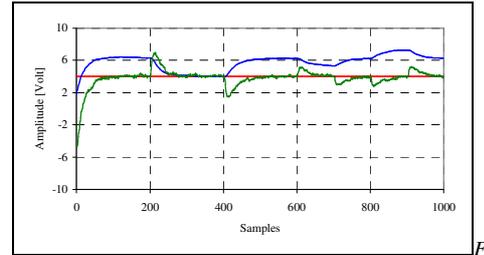


Fig. 8. Neural RBF Controller.

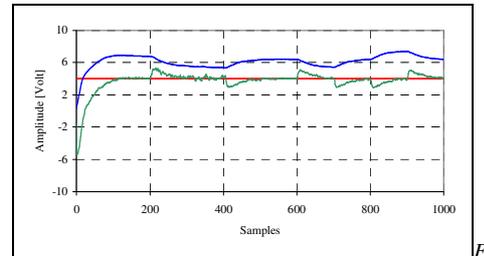


Fig. 9. Hybrid Neural Controller.

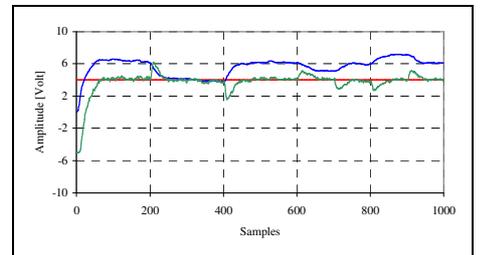


Fig. 10. Modified SOC.

The controllers perform well, rejecting disturbances and maintaining the output at its set point.

*Process Dynamics Change.* In this final group of experiments the introduction of a pure time delay and additional pole is tested. The first can be seen as a non-linear effect and the second as a linear dynamics change. Assuming a square wave reference, at sample 400, a pure time delay of one second and a pole with a time constant equal to one second is added. The results of the experiments are presented in Fig. 11, Fig. 12 and Fig. 13.

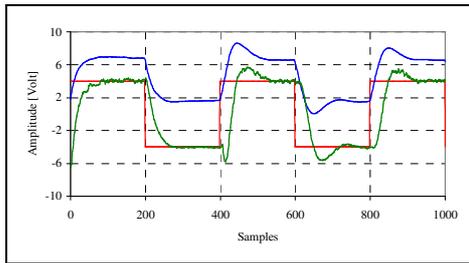


Fig. 11. Neural RBF Controller.

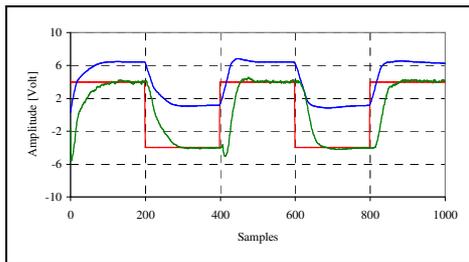


Fig. 12. Hybrid Neural Controller.

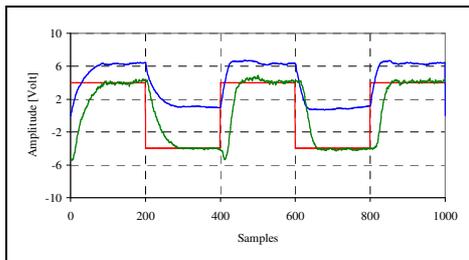


Fig. 13. Modified SOC.

The three controllers performed considerably well in the presence of dynamics change.

### 3.3 Performance Evaluation

In order to assess more precisely the controllers' behaviour four performance indexes are used:  $SSE$  (eq. 6),  $SISE$ ,  $SSC$  and  $SSCC$  - Table 1.

$$SISE = \sum_{k=0}^n ke_k^2 ; SSC = \sum_{k=0}^n u_k^2 ; \quad (21)$$

$$SSCC = \sum_{k=1}^n (u_k - u_{k-1})^2$$

The initial parameters for each controller were obtained for the worst case (process dynamics change). For the other experiments better performance could be achieved if an appropriate initial configuration were considered. In this case the controllers robustness was preferable relatively to particular performances.

The table analysis shows that the performance indexes are very similar for the controllers in each experiment. In fact none of the controllers' can be clearly considered superior.

Table 1 Performance indexes.

Controller	Performance Indexes			
	$SSE$	$SISE$	$SSC$	$SSCC$
<i>Tracking problem</i>				
RBF	3 475.1	996 706.5	23 058.2	2.2
HNC	4 415.8	1 211 076.0	21 900.4	2.2
SOC	2 734.3	445 843.1	20 803.4	5.5
<i>Regulation problem</i>				
RBF	1 308.7	194 420.3	33 860.4	0.8
HNC	1 638.3	127 887.2	38 555.8	1.0
SOC	1 896.9	155 804.0	32 741.7	1.7
<i>Process dynamics change</i>				
RBF	8	3 886 068.3	29010.3	5.0
	372.1			
HNC	8	3 886 969.6	23729.6	3.7
	770.1			
SOC	9	3 684 782.2	23065.8	5.1
	015.9			

## 4. CONCLUSIONS

These kinds of controllers, included in the broader class of intelligent controllers, have a large amount of potentialities. They do not require a big amount of previous knowledge about the controlled process (although they allow to integrate that knowledge) and can be used in real-time control.

Concerning the experiments, the three controllers have shown acceptable performances. However real sized industrial applications are needed to assess a final conclusion.

*Acknowledgement:* This work was partially financed by JNICT/PRAXIS XXI.

## REFERENCES

- Åström, K. and B. Wittenmark (1995). *Adaptive control* (2<sup>nd</sup> Edition). Addison Wesley.
- Feedback (1985) - Process Trainer PT 326 / PCS 327 - Technical Information
- Henriques, J. and A. Dourado (1998). Adaptive Control Using a Recurrent Neural Network Observer. *Control98* - Coimbra, September 9-11.
- Jager, R. (1995). Fuzzy Logic in Control. Ph.D. Thesis, Delft Univ. of Technology, The Netherlands.
- Moody, J., and Darken, C. Fast learning in networks of locally-tuned processing units. *Neural Computation*, vol 1, N° 2, pp. 281-294, 1989.
- Pereira, C., J. Henriques, B. Ribeiro, A. Dourado, Real Time Adaptive Training of RBFNN using a Selective Forgetting Algorithm, EANN96, London, June 1996
- Pham, D. and L. Xing (1995). Dynamic System Identification Using Elman and Jordan Networks. *Neural Networks for Chemical Engineers*, Editor A. Bulsari, Chap. 23.
- Procyk, T. and E. Mamdani (1979). A Linguistic Self-Organising Process Controller. *Automatica*, vol. 15, pp. 15-30
- Shao, S. (1988). Fuzzy self-organising controller and its application for dynamic processes. *Fuzzy Sets and Systems*, vol. 26, pp. 151-164.
- Werbos, P (1990). Backpropagation trough time: what it does and how do it. *Proc IEEE*, vol. 78, pp. 1550-1560.



ELSEVIER

International Journal of Approximate Reasoning 22 (1999) 73-91

INTERNATIONAL JOURNAL OF  
APPROXIMATE  
REASONING

# Supervision and c-Means clustering of PID controllers for a solar power plant

Jorge Henriques<sup>\*1</sup>, Alberto Cardoso<sup>2</sup>, António Dourado<sup>3</sup>

*CISUC - Centro de Informática e Sistemas da Universidade de Coimbra, Departamento de Engenharia Informática, Pólo II, UC 3030 Coimbra Portugal*

Received 1 October 1998; accepted 1 February 1999

## Abstract

A hierarchical control strategy consisting on a supervisor switching of PID controllers simplified using the C-Means clustering technique is presented. This control approach was developed and applied on the distributed collector field of a solar power plant. The main characteristic of a solar power plant is that the primary energy source, the solar radiation, can not be manipulated. It varies throughout the day, causing changes in plant dynamics conducting to distinct main several operating points. Around each specific operating point it is possible to design a particular PID controller that allows an acceptable behaviour. However, due to the complex behaviour of the plant, poor transient performance is achieved in another operating points. To overcome this problem, the supervisor, based on real data available from the plant, provides a way to switch between several PID controllers tuned for the nominal operating points, which represents the range of interest. Each of these PID controllers have been previously off-line tuned, using a neural network approach, that combines a dynamic recurrent non-linear neural network model with a pole placement control design. To reduce the number of local controllers, to be used by the supervisor, the c-Means clustering technique was used. Simulation and experimental results, obtained at Plataforma Solar de Almería (in Spain), are presented showing the effectiveness of the proposed approach. © Elsevier Science Inc. All rights reserved.

\*Tel: +351-39-790000; fax +351-39-701266; e-mail: jh@dei.uc.pt

<sup>1</sup> The experiments were carried out within the project "Innovative Training Horizons in Applied Solar Thermal and Chemical Technologies" (C.N.:ERBFMGECT950023) supported by the European Union Program "Training and Mobility of Researchers" and promoted by CIEMAT-IER. The authors would like to express their gratitude to the personnel of the Plataforma Solar de Almería, in particular Pedro Balsa. The authors also acknowledge suggestions by the reviewers that have helped considerably in the improving of this paper. This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under program PRAXIS XXI.

<sup>1</sup> E-mail: alberto@dei.uc.pt

<sup>2</sup> E-mail: dourado@dei.uc.pt

0888-613X/99/\$ - see front matter © 1999 elsevier Science Inc. All rights reserved.

PII: S 0 8 8 8 - 6 1 3 X ( 9 9 ) 0 0 0 1 7 - 1

*Keywords:* Fuzzy supervisor; Clustering; Switching control; PID Control; Neural networks; Solar power plants.

---

## 1. Introduction

The main control requirement in a solar power plant, in order to be able to use the heated oil for power production, is to maintain the outlet oil temperature from the collector field at a constant pre-specified value. The fundamental feature of the plant is that its primary energy source, the solar radiation, is a measurable disturbance and can not be influenced by the control system. Moreover, since the solar radiation changes substantially during plant operation, due to the daily solar cycle, atmospheric conditions, such as a cloud cover, humidity and turbidity, this leads to significant variations in the dynamic characteristics (e.g. the time constant and the time delay) of the field, corresponding to different operation conditions. Therefore, it is difficult to obtain a satisfactory performance over the total operating range with a static controller.

This paper presents the application of a hierarchical control strategy to the distributed collector field of a solar power plant at the Plataforma Solar de Almería. The Acurex distributed solar collector field of this solar plant is well described in literature [6,7] and is located at Tabernas, in south of Spain. The field consists of 480 distributed solar collectors arranged in 20 rows, which form 10 parallel loops. Each loop is 172 m long and the total aperture surface is 2672 m<sup>2</sup>. The plant is able to provide 1.2 MW peak of thermal power. A picture and a schematic diagram are shown, respectively, in Figs. 1 and 2.

Each collector uses parabolic mirrors to concentrate solar radiation in a receiver tube. Synthetic oil is pumped through the receiver tube and picks up the heat transferred through the tube walls. The inlet oil, at temperature  $T_{in}$ , is pumped from the bottom of the storage tank flowing through the collector field where its temperature is raised. Next, the heated fluid is introduced into the storage tank, from the top, to be used for electrical energy generation or feeding a heat exchanger in the desalination plant. The manipulated variable in the solar plant is the oil flow rate,  $Q_{in}$ , and the main goal is to regulate the outlet field oil temperature,  $T_{out}$ , at a desired value,  $T_{ref}$ . The main disturbances are the solar radiation,  $Q_{rr}$ , and the inlet oil temperature.

To deal with the several operating points, that characterises the plant behaviour, some control strategies have been proposed. One of them applies adaptive control schemes, using local linear models of the plant, which mimic changes during the operation [7,5]. In reference [8], experimental results when measurable disturbances of the plant (solar radiation and inlet oil temperature)



Figure 1. Acurex Solar Collector Field.

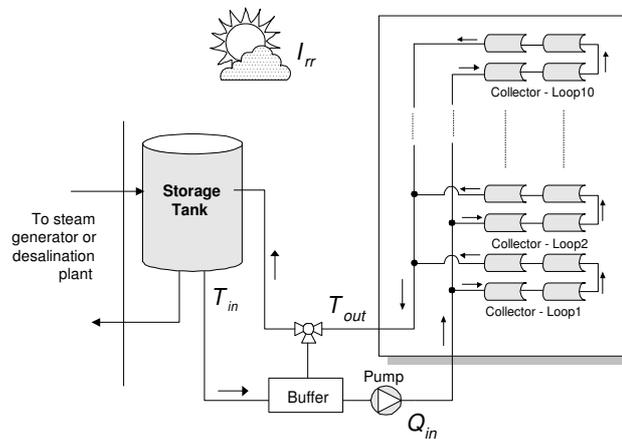


Figure 2. Schematic diagram of the Acurex Solar Collector Field.

are included in the design of a predictive MUSMAR-controller, are presented. Also fuzzy controllers (e.g. incremental PI fuzzy controller) have been tested at this plant [21,3]. In some of these contributions a feedforward compensation term, obtained from the static behaviour of the plant, has been proposed to compensate the effects of the radiation and the difference between the desired output temperature and the inlet temperature. Another possibility is to apply

an indirect adaptive controller, using a certain predictive search strategy based on a mathematical model of the process [19,18]. Another alternative might be the commissioning of a switching controller using different models of the plant for different operating points [20]; a similar strategy has been followed in this work.

There are several ways to design a controller for each local model. Traditional PID controllers have some well known advantages, such as the dynamic performance reached in some nominal operating conditions, reliability based on possible stability studies and their industrial widespread use [9]. They are simple to implement and they can successfully regulate many industrial processes with various specifications.

In order to improve the performance of the PID controllers, several strategies have been proposed, such as adaptive and supervising techniques. One can consider the following main reasons for using a supervisor [12]:

- (i) adaptive controllers are able to cope with the most of the cases that leave the PID under-optimal, but they require specialised design methods using some a priori model structure knowledge [16];
- (ii) PID supervisors are easier to implement because they need very little knowledge about the process;
- (iii) the combination of a linear PID control law with a supervisory strategy can lead to a highly non-linear control law; then it can allow increasing significantly the robustness of the control system [1];
- iv) finally, the supervisor can provide an interface with the user for expressing precisely the specifications in terms of close loop performances.

One possibility to implement the supervisor is the fuzzy logic methodology (some examples are described in Ref. [13]). Fuzzy logic, [23], have been considered as an effective tool to deal with disturbances and uncertainties in terms of vagueness, ignorance, and imprecision.

In this work the hierarchical control strategy is based on a PID control technique with a fuzzy logic switching supervisor. The supervisor has been derived using a fuzzy method to implement on-line the switching between each PID controller, accordingly to the measured conditions. The local PID controllers have been previously tuned off-line using a neural network approach, that combines a dynamic recurrent non-linear neural network model with a pole placement control design. For reducing the number of local controllers, a c-Means clustering technique was applied.

The paper is organised as follows. Section 2 describes the neural network approach to design the PID controllers and the classification technique is presented. The fuzzy switching supervisory, providing the mechanism to switch between the PID controllers, is also described. In section 3 some simulations and experimental results are presented. Finally, some conclusions are embodied in section 4.

## 2. Control Methodology

### 2.1 - PID Design and Tuning

Neural networks have been successfully applied for the modelling and control of non-linear systems. The dynamic recurrent neural networks, which involve dynamic elements and feedback connections, have been considered more suitable for modelling dynamical systems [11]. In this work, a recurrent Elman's neural network was used to obtain each nominal model. The corresponding nominal parameters of each PID controller were obtained with a pole placement algorithm, assuming a linearisation of each nominal neural model. Additionally, in order to improve the controllers performance, the experience acquired from the plant behaviour can be used to further adjust the PID parameters.

For the controller design, the available plant data, obtained by experimental results from other researchers, was used. Due to the difficulty to obtain data covering all range of possible operating conditions, some controllers were designed using data from a simulation plant model [2]. In this case the following relation [2], characterising the steady state behaviour, was used

For the controller design, the available plant data, obtained by experimental results from other researchers, was used. Due to the difficulty to obtain data covering all range of possible operating conditions, some controllers were designed using simulation data from a plant model. In this case the following relation [2], characterising the steady state behaviour, was used

$$Q_{in} = \frac{0.7869I_{rr} - 0.485(T_{out} - 151.5) - 80.7}{(T_{out} - T_{in})} \quad (1)$$

It is assumed that each nominal model, is described by a general non-linear discrete state equation in the form:

$$x(k) = f \{ x(k-1), Q_{in}(k-1) \} \quad (2)$$

$$T_{out}(k) = h \{ x(k) \} \quad (3)$$

where  $x(k) \in \mathfrak{R}^n$ , represents a state vector at discrete time k and  $f$  and  $h$  are non-linear functions. The ability of a recurrent Elman's network to approximate a discrete time non-linear system is used [17]. No a priori assumptions are made about the process, with the exception that the maximum value for its order (n) is known in advance. An Elman's network can be described by the following equations:

$$x^h(k) = \sigma \{ W^{xc} x^c(k), W^{xu} Q_{in}(k-1) \} \quad (4)$$

$$x^c(k) = \lambda x^c(k-1) + x^h(k-1) \quad (5)$$

$$T_{out}(k) = W^{yx} x^h(k) \quad (6)$$

where  $\sigma\{\cdot\}$  is a hyperbolic tangent non-linear function,  $x^c(k)$  is a context unit and  $x^h(k)$  is a hidden unit. The context units are locally recurrent and a multiplicative constant,  $\lambda$ , decreases the values as they are fed back. This constant determines the memory depth, i.e., how long a given value fed to the context

unit will be *remembered*. The interconnection matrices,  $W^{xu} \in \mathfrak{R}^{1,n}$ ,  $W^{xc} \in \mathfrak{R}^{n,n}$  and  $W^{yx} \in \mathfrak{R}^{n,1}$ , which define the interconnection paths for the context-hidden layer, input-hidden layer and hidden-output layer respectively, are evaluated from the truncated Werbos' backpropagation through time algorithm [22]. e to the computational complexity of the BTT, a natural simplification was obtained by truncating the backpropagation of the information to a fixed number of prior time steps, on a sliding window mode. From the non-linear neural model, Eqs. (4)-(6) it is possible to derive a linear model by computing the derivatives from the output ( $T_{out}$ ) with respect to the input ( $Q_{in}$ ), extracting the actual linearised parameters [10]. Following this approach a discrete time LTI (*linear time invariant*) SISO (*single input single output*) system can be obtained in the form of a standard discrete time state space model:

$$x(k) = W^A x(k-1) + W^B Q_{in}(k-1) \quad (7)$$

$$T_{out}(k) = W^C x(k) \quad (8)$$

$W^A$ ,  $W^B$  and  $W^C$  are matrices of dimension  $(n,n)$ ,  $(n,1)$  and  $(1,n)$  respectively. Due to the identifiability of the controller parameters, a second order system ( $n=2$ ) was considered. Adopting an input-output representation, the system described by Eqs. (7) and (8) can be defined by

$$P(q^{-1}) = \frac{T_{out}(k)}{Q_{in}(k)} = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} = \frac{B(q^{-1})}{A(q^{-1})} \quad (9)$$

where  $q^{-1}$  is the backward shift operator. Concerning the PID controller, it is defined by:

$$F(q^{-1}) Q_{in}(k) = G(q^{-1}) e(k) \quad (10)$$

The output error,  $e(k)$ , is defined as the difference between the desired outlet oil temperature,  $T_{ref}$ , and the actual outlet oil temperature,  $T_{out}$ .

$$e(k) = T_{ref}(k) - T_{out}(k) \quad (11)$$

$F(q^{-1})$  and  $G(q^{-1})$  are polynomials in the form

$$F(q^{-1}) = 1 + f_1 q^{-1} + f_2 q^{-2} = (1 - q^{-1})(1 + f q^{-1}) \quad (12)$$

$$G(q^{-1}) = g_o + g_1 q^{-1} + g_2 q^{-2} \quad (13)$$

Then, the closed loop transfer function is given by

$$T_{out}(k) = \frac{BG(q^{-1})}{AF(q^{-1}) + BG(q^{-1})} T_{ref}(k) \quad (14)$$

The PID polynomials are computed such that the closed loop poles, obtained from the solution of Diophantine equation (15), are placed in the desired locations  $p_1$  and  $p_2$ .

$$AF(q^{-1}) + BG(q^{-1}) = A_m \tag{15}$$

$$A_m = (1 - p_1q^{-1})(1 - p_2q^{-1}) = 1 + \gamma_1q^{-1} + \gamma_2q^{-2} \tag{16}$$

Finally the PID parameters are evaluated by the following equation

$$\theta = \Gamma^{-1}\Omega \tag{17}$$

where  $\mathfrak{K}$ ,  $\mathfrak{S}$  and  $\theta$  are matrices defined by

$$\theta = \begin{bmatrix} f_1 \\ f_2 \\ g_0 \\ g_1 \\ g_2 \end{bmatrix} \tag{18}$$

$$\Gamma = \begin{bmatrix} 1 & 0 & b_1 & 0 & 0 \\ a_1 & 1 & b_2 & b_1 & 0 \\ a_2 & a_1 & 0 & b_2 & b_1 \\ 0 & a_2 & 0 & 0 & b_2 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

$$\Omega = \begin{bmatrix} \gamma_1 - a_1 \\ \gamma_2 - a_2 \\ 0 \\ 0 \\ -1 \end{bmatrix} \tag{20}$$

### 2.2 – Supervisor Design

To formulate the design problem it is assumed that it is possible to represent the plant dynamics by a number of characteristics behaviours  $M_1, M_2, \dots, M_N$  and that, for each  $M_i$ , a corresponding model  $P_i$  may be derived, where each  $P_i$  is called a nominal model. In order to obtain a desired performance when the plant is operating under conditions  $M_i$ , it is assumed that a nominal controller  $C_i$ , can also be designed. The overall structure of the fuzzy switching supervisor PID control is shown in Figure 3.

The fuzzy supervisor consists of three stages: the fuzzification, the fuzzy rule base and the inference procedure. The first one converts the numerical values of the solar radiation ( $I_{rr}$ ) and the reference temperature ( $T_{ref}$ ), which mainly define the nominal operating points, into linguistic variables. The fuzzy rule base defines the switching control strategy. The selected controller is obtained by the defuzzification part, which chooses among rules that have been fired simultaneously.

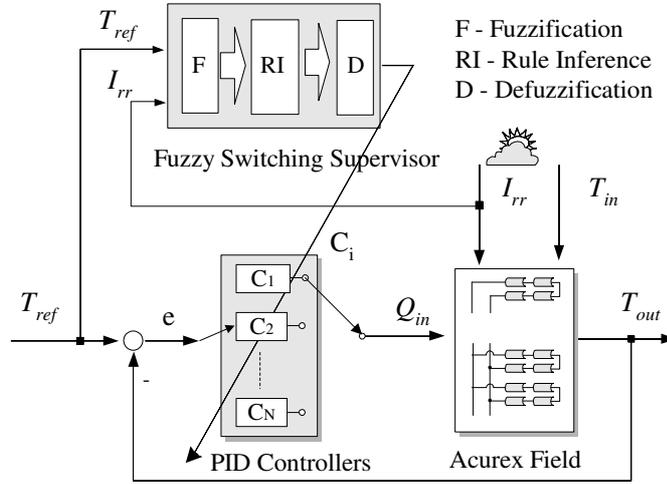


Figure 3. Schematic diagram of fuzzy switching supervisor PID control.

An intuitive method, issued from the knowledge of the operators and of the authors themselves, was used to derive the membership functions for the linguistic variables: solar radiation and reference temperature. The universes of discourse  $I$  and  $R$ , respectively for the solar radiation and reference temperature, were defined by:

$$I \equiv [600 W / m^2, 1000 W / m^2] \tag{21}$$

$$R \equiv [230 C, 270 C] \tag{22}$$

For both variables it was assumed five linguistic terms

$$\{VS, SM, NO, LA, VL\} = \{Very\ Small, Small, Normal, Large, Very\ Large\} \tag{23}$$

The fuzzy sets  $SI$  and  $SR$  are defined as a set of ordered pairs:

$$SI = \{I_{rr}, \mu_I(I_{rr}) \mid I_{rr} \in I\} \tag{24}$$

$$SR = \{T_{ref}, \mu_R(T_{ref}) \mid T_{ref} \in R\} \tag{25}$$

The membership functions were considered triangular, as shown in Figure 4 and Figure 5.

The rule base, that defines the strategy how to do the switching between controllers, is drawn in table of Fig. 6. The two inputs, assuming to describe the actual nominal condition of the solar plant, are the solar radiation and reference temperature, the output  $i = 1, \dots, 25$  is the index that identifies the select controller,  $C_i$ .

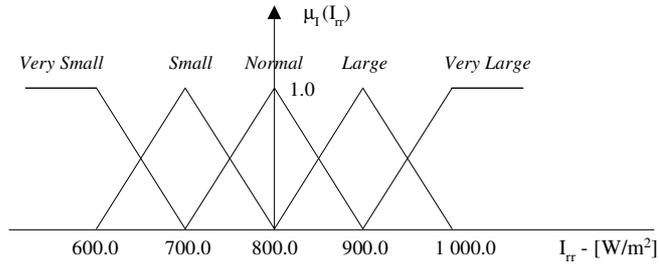


Figure 4. Solar Radiation Membership function.

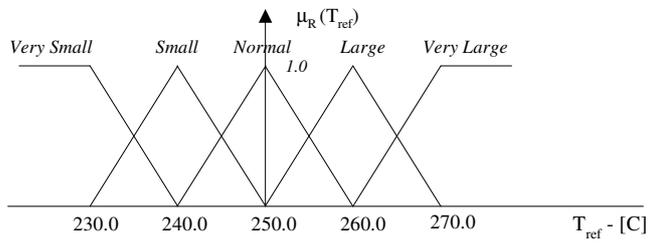


Figure 5. Reference Temperature Membership function.

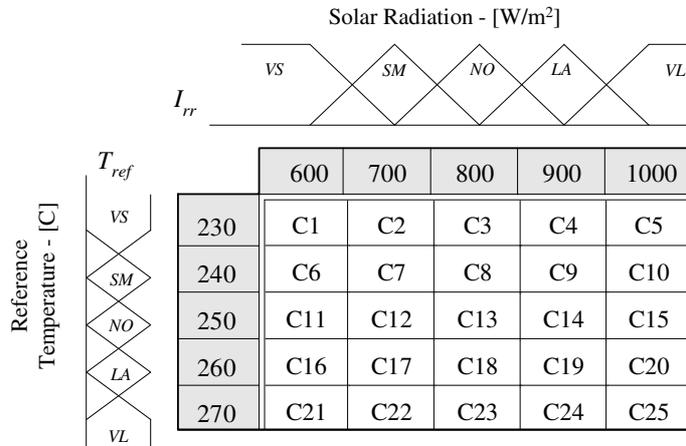


Figure 6. Switching Supervisor Rule Base.

In general a Takagi-Sugeno fuzzy system is described by a set of rules in the form

$$\begin{aligned}
 \text{Rule } i: & \text{ IF } (x \text{ is } X) \text{ AND } (y \text{ is } Y) \\
 & \text{ THEN } (z \text{ is } Z)
 \end{aligned}
 \tag{26}$$

where  $X$  and  $Y$  are fuzzy sets in the antecedent and  $Z$  is a fuzzy set in the consequent. In a Mamdani type inference, after the aggregation process, it is common to have a fuzzy set for each output variable that needs defuzzification. However, it is possible to use a singleton as the output membership function rather than a fuzzy set. Following this idea the switching supervisor rule base (table shown in Fig. 6 can be described by a set of rules in the following way

$$\begin{aligned} \text{Rule } i: & \text{ IF } (I_{rr} \text{ is } SI^{(i)}) \text{ AND } (T_{ref} \text{ is } SR^{(i)}) \\ & \text{ THEN } \alpha_i = \mu_I^{(i)}(I_{rr}) \otimes \mu_R^{(i)}(T_{ref}) \end{aligned} \quad (27)$$

where  $SI^{(i)}$  and  $SR^{(i)}$  define, respectively, the linguistic values of variables  $I_{rr}$  and  $T_{ref}$  for the  $i$  rule.  $\mu_I^{(i)}(I_{rr})$  and  $\mu_R^{(i)}(T_{ref})$  define the respective membership degrees.

The aggregation of the consequents of the fired rules, determine  $i$  such that  $\alpha_i = \alpha_{\max} = \max \{ \alpha_1, \dots, \alpha_K, \dots, \alpha_{25} \}$ . This index identifies the controller to be selected. This very simple procedure proved to be effective.

Usually only one controller will be active. In the case that two controllers were activated it is assumed that the rule which correspond to the highest reference temperature membership function,  $\mu_R^{(i)}(T_{ref})$ , will be the selected one.

Fig. 7 shows the fuzzy switching approach, presented above, for a particular situation ( $I_{rr} = 620 \text{ W/m}^2$  and  $T_{ref} = 267 \text{ C}$ ). Using the proposed fuzzy model, the selected controller was the  $C_{21}$ .

### 2.3 – Clustering and Supervisor Simplification

The purpose of clustering is to classify a given data set  $X = \{x_1, x_2, \dots, x_N\}$  into homogeneous groups of data  $V = \{v_1, v_2, \dots, v_M\}$  with  $1 \leq M \leq N$  (if  $M=1$ , all data belongs to the same class and if  $M=N$ , each data sample defines a class). In this work, homogeneous data means the controllers parameters that have similar values. The c-Means clustering algorithm [4] (Bezdek, 1981) is an extremely powerful classification method which minimises the Euclidean distance between each data point (controller coefficients) and its cluster center. The number of clusters ( $M$ ) should reflect the level of knowledge of the system under consideration, or the level of generality in the user's description of the system.

The data to be classified are the parameters of the nominal PID controllers  $C_i$   $i = 1, \dots, 25$  (which can be interpreted as 3D points, see Figure 8). The number of clusters (distinct controller classes) considered, was  $M=7$ . After applying the c-Means clustering procedure, the initial twenty five controllers (symbol + in Figure 8) were reduced to seven classes (symbol  $\circ$  in Figure 8).

By reducing the number of controllers, the task of re-tuning of the nominal controllers can be simplified. The final rule base, which defines the supervision task, is now on Fig. 9..

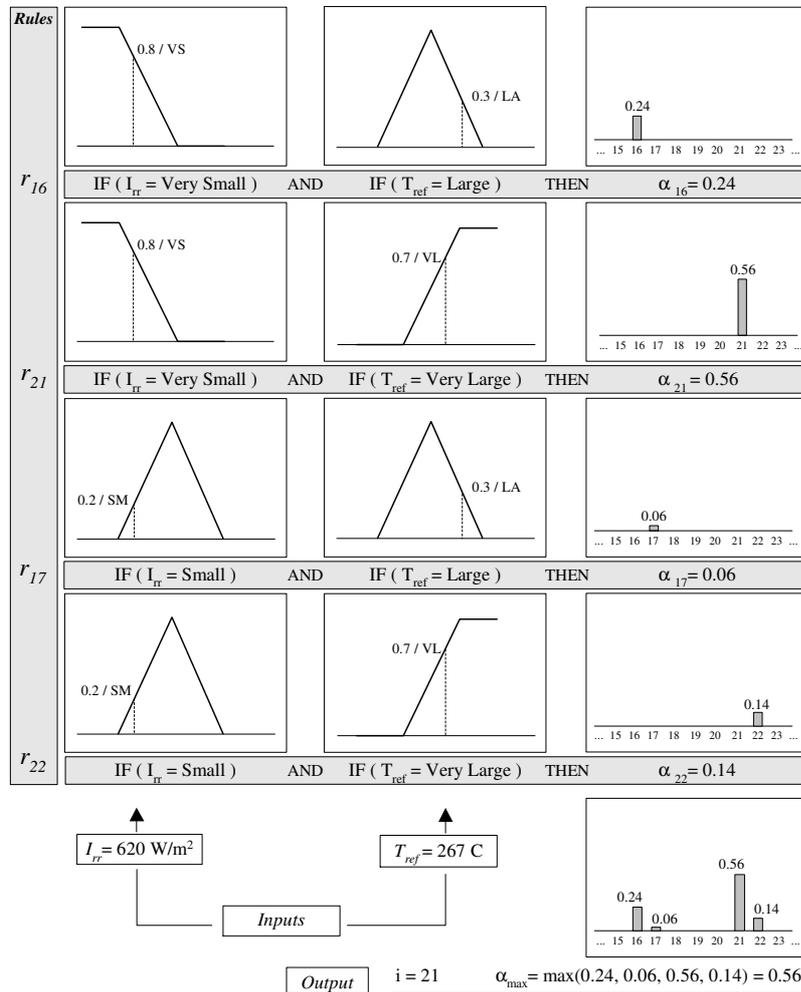


Figure 7. Example of the fuzzy inference mechanism.

### 3. RESULTS

The experiments were carried out in the Acurex Solar Collectors Field of the *Plataforma Solar de Almería* (PSA) from 6 to 7 July 1998. The proposed control was implemented in C code and operates over a software developed at PSA [14] also in C code.

The effectiveness of the developed approach was first tested using a non-linear distributed parameter model of the Acurex field, developed at the University of Sevilla [2]. Also the behaviours of the proposed approach and of a fixed PID controller were compared.

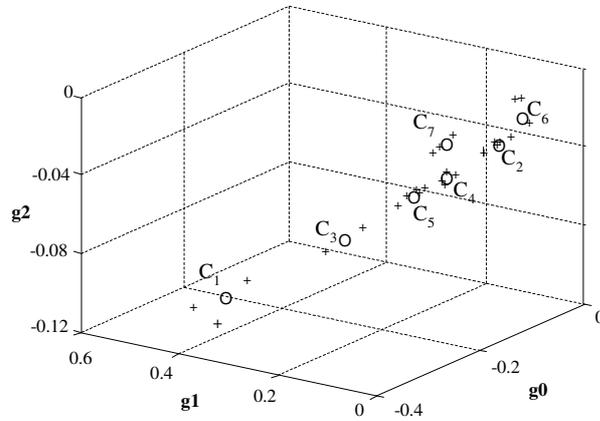


Figure 8. Clustering of the PID controllers.

		Solar Radiation - [W/m <sup>2</sup> ]				
		600	700	800	900	1000
Reference Temperature - [C]	230	C5	C5	C2	C6	C6
	240	C5	C1	C5	C2	C2
	250	C7	C5	C1	C4	C2
	260	C7	C2	C4	C3	C4
	270	C6	C7	C1	C4	C3

Figure 9. Simplified Supervisor Rule Base.

The described experiments intended to show the results in the following situations: (i) several changes in the points of operation, by setting different reference temperatures (it is not possible to manipulate the solar radiation), and (ii) the rejection capabilities of the controller to disturbances in the inlet oil temperature and when the addition/suppression of a collector loop from operation was done.

The effect of strong disturbances caused by large passing clouds, which produce drastic changes in the direct solar radiation level, was not possible to test due to the clear conditions during the experiments. In all the experiments the sampling time was 15 seconds and the output temperature ( $T_{out}$ ) was considered as the maximum temperature of all the loops (another usual strategy is to assume the average value).

3.1 - Simulations

The performance of the proposed strategy was compared with a fixed PID controller. The fixed controller was designed for the operating point  $I_{rr} = 800 W / m^2$ ,  $T_{ref} = 240 C$ , in other words, the controller  $C_5$  was implemented. The data used to perform the present simulation ( $T_{in}$  and  $I_{rr}$ ) was obtained on 6 July 1998 (the solar radiation evolution is presented in Figure 12).

From the simulation it can be seen that the fixed PID controller is well designed for its nominal conditions (approximately from 10h10m to 10h50m, from 12h15m to 12h50m and from 14h10m to the end of the time simulation), but for other conditions, the performance deteriorates.

The proposed strategy, on the other hand, deals quite well with these nominal conditions variations, providing appreciable improvements in the performance of the closed loop system (Figure 11). The selected controller, computed by the fuzzy supervisor, and the solar radiation evolution, are shown in Figure 12.

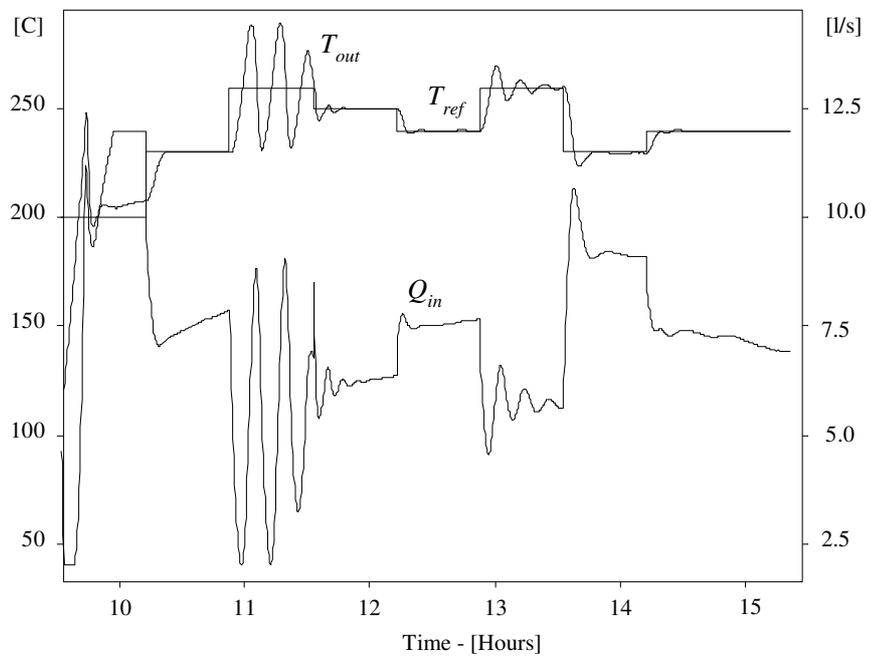


Figure 10. Simulation 1: time response using a fixed PID controller (C5).

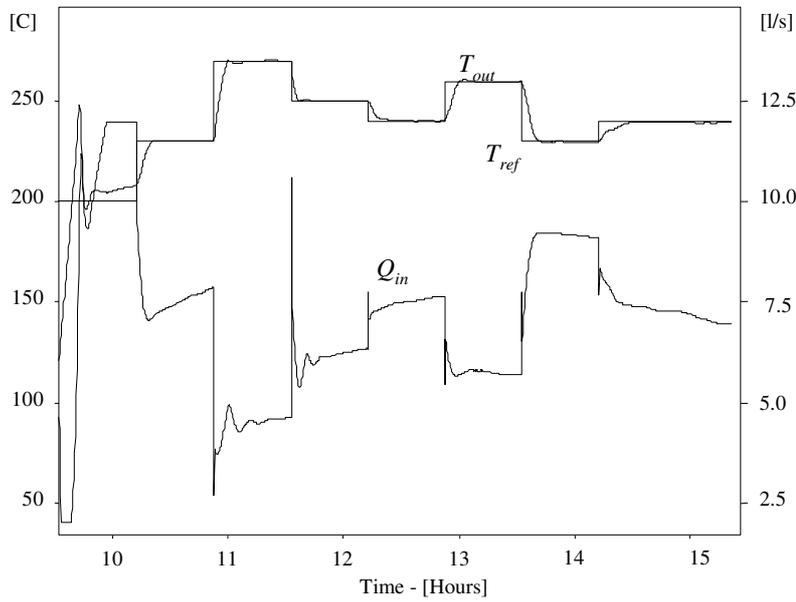


Figure 11. Simulaton2: time response using the proposed strategy.

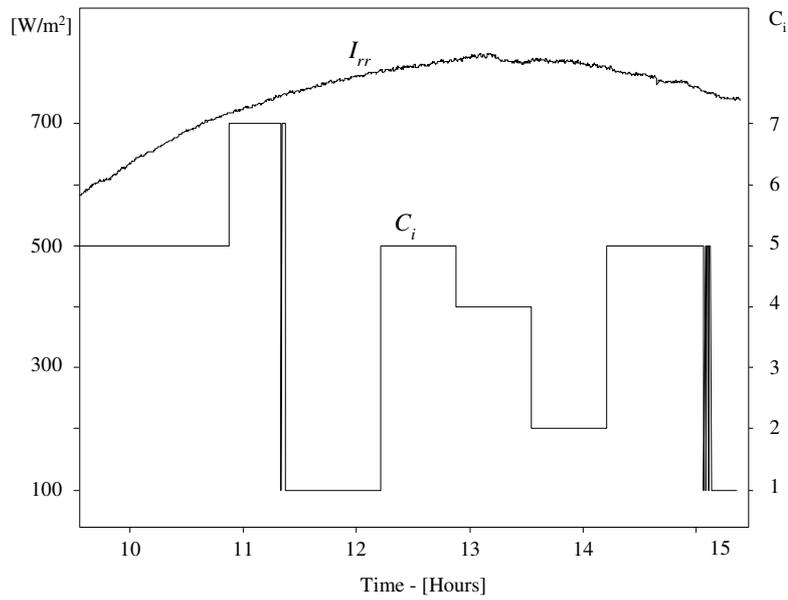


Figure 12. Simulation2: solar radiation and selected controller - 6 July 1998.

3.2 - Experiments

The first experiment was carried out on 6 July 1998. The proposed control strategy was tested to cope with the changes in plant dynamics. Figure 13 shows the results in which several reference temperatures changes have been performed. Also, in order to show the rejection capabilities of the proposed control, a change in the inlet oil temperature was done at instant 13h25m.

As can be seen the behaviour is quite good. The response presents almost no oscillations neither overshoot and settles for the new value of the reference temperature in about 15 minutes. The disturbance rejection capabilities of the controller were also acceptable.

From the use of a PID controller, it should be expected a zero steady state error. The actual error, for instance at instant 12h10m, is justified by the evolution of solar radiation in its daily cycle, which acts as a load disturbance (approximately a ramp) in the output temperature and is not adequately compensated by the integral action of the PID. This experiment has shown that the fuzzy supervisor mechanism, which makes the switching between the PID controllers (shown in Figure 14), provides a good control strategy.

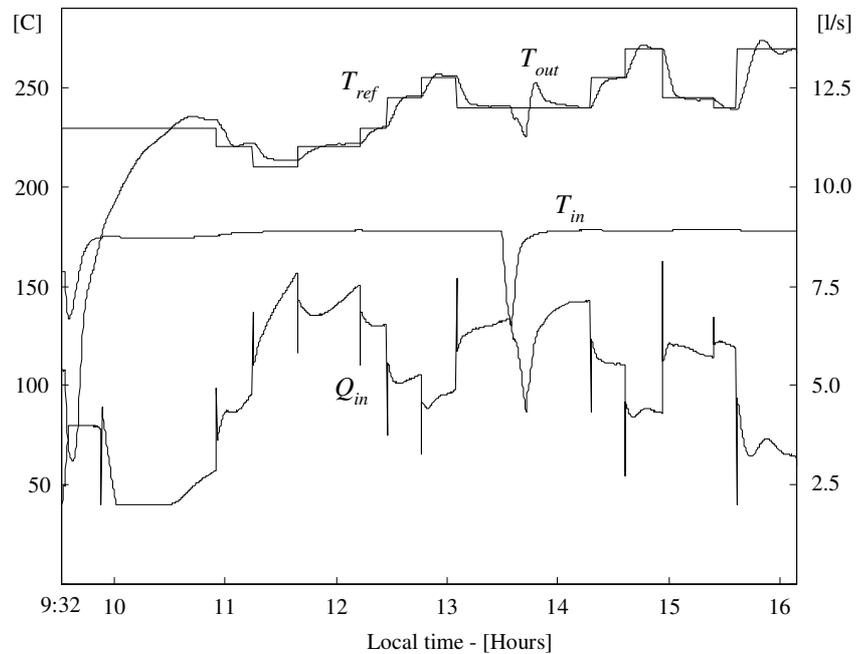


Figure 13. Experiment 1: results obtained on 6 July 1998.

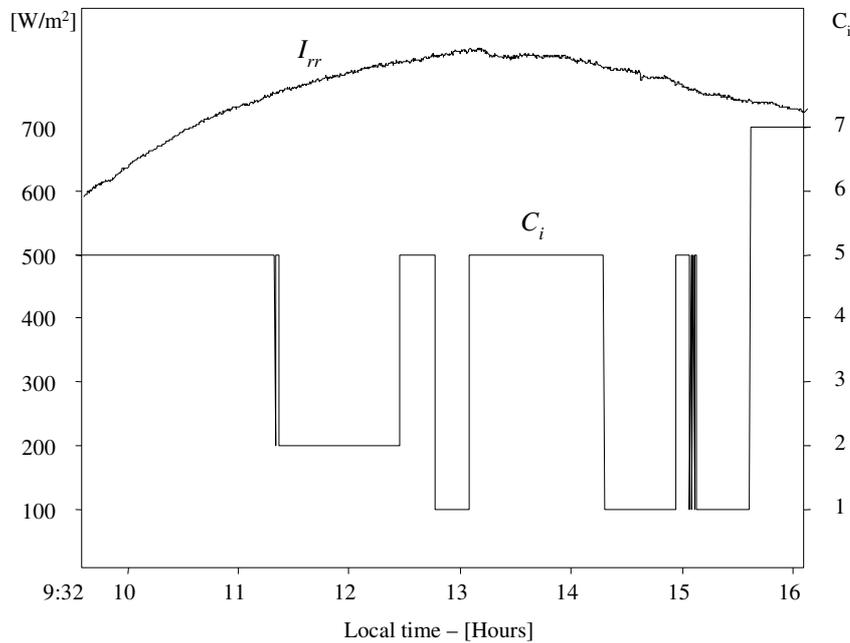


Figure 14. Experiment 1: solar radiation and selected controller - 6 July 1998.

The experiment 2 was carried out on 7 July 1998 and the results can be seen in Figure 15 and Figure 16. The set point reference temperature was changed. Also at instant 14h55m a collector loop was suppressed from the field and added again at instant 15h12m.

After an initial phase the outlet oil temperature achieves the reference. The results are very acceptable in face of the different operating points and the addition/suppression of a collector loop.

Concerning the fuzzy switching supervisor and observing the results, it can be concluded that the adequate controller has been selected to each operating point.

#### 4. Conclusions

A PID based hierarchical control combining a switching supervisor strategy and the c-Means clustering technique has been presented and applied to a distributed collector field of a solar power plant. The process is characterised by different operating conditions, depending on the changes in dynamics caused by variation of the solar radiation, reference temperature, and plant characteristics. Based on a neural network approach off-line trained with experimental data, a set of nominal PID controllers tuned for different operating points have been designed. Next, the number of nominal PID controllers have been reduced employing a c-Means clustering technique.

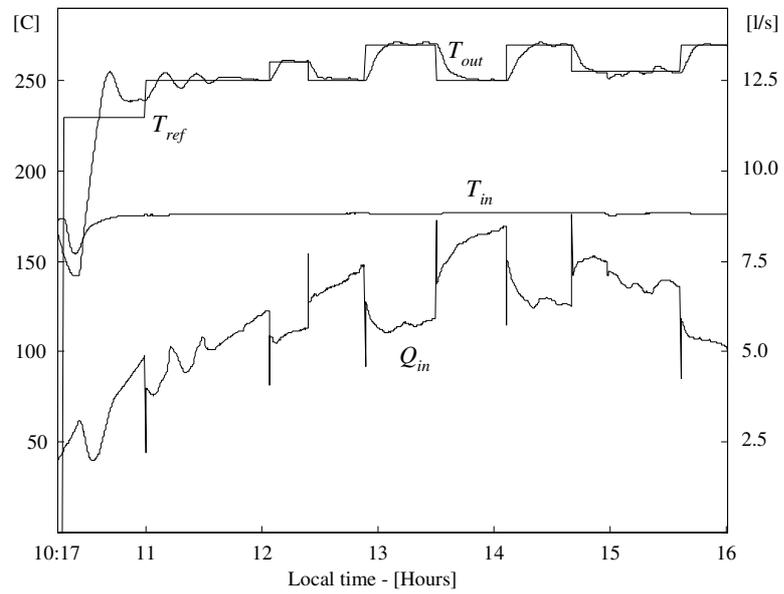


Figure 15. Experiment 2: results obtained on 7 July 1998.

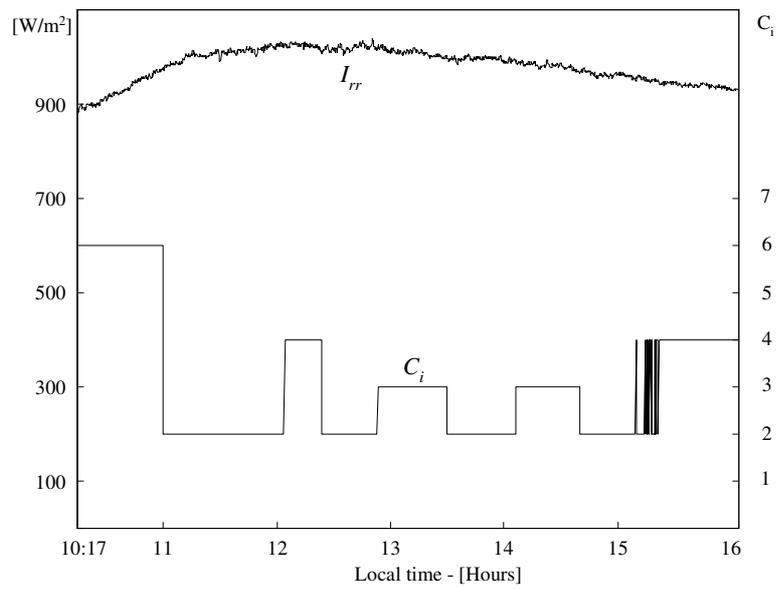


Figure 16. Experiment 2: solar radiation and selected controller - 7 July 1998.

Simulation results show that the closed loop system achieves a good performance and show the improvements achievable by the proposed strategy with respect to the use of just one controller. Experimental results confirm the simulation results and show that the system has robustness with respect to changes in solar radiation, inlet oil temperature and operating conditions.

## References

- [1] Babuska, R., P. Horacek (1992). Laboratory evaluation of fuzzy controllers. *Pre-prints of the IFAC/IFIP/IMACS International Symposium of Artificial Intelligence in Real Time Control*, Delft, Netherlands, June 18-18, 263-268.
- [2] Berenguel, M., E. Camacho, F. Rubio (1993). Simulation Software Package for the Acurex. *Internal Report*, E.S.I.de Sevilla.
- [3] Berenguel, M., E. Camacho, F. Rubio, P. Luk (1997). Incremental fuzzy PI control of a solar power plant. *IEE Proc. Control Theory and Applications*, 144 (6), 596-604.
- [4] Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Functions Algorithms*. Plenum, New York.
- [5] Camacho, E., F. Rubio, F. Hughes (1992). Self-Tuning control of a solar power plant with a distributed collector field. *IEEE Control Systems Magazine*, 12 (2), 72-78.
- [6] Camacho, E., F. Rubio, J. Gutierrez (1988). Modelling and simulation of a Solar Power Plant with distributed collector system. *IFAC Symposium Power Systems, Modelling and Control Applications*, Brussels.
- [7] Camacho, E., M. Berenguel, F. Rubio (1994). Application of a gain scheduling generalised predictive controller to a solar power plant. *Control Engineer Practice*, 2 (2), 227-238.
- [8] Coito, F., J. Lemos, R. Silva, E. Mosca (1997). Adaptive control of a solar energy plant: Exploiting accessible disturbances. *Int. Journal of Adaptive Control and Signal Processing*, 11 (4), 327-342.
- [9] French, I., C. Cox, M. Willis, G. Montague (1992). Intelligent tuning of PI controllers for bioprocess application. *Pre-prints of the IFAC/IFIP/IMACS Internat. Symposium on Artificial Intelligence in Real-Time Control*, Delft, Netherlands, 697-702.
- [10] Henriques, J. A. Dourado (1998). A Multivariable adaptive control using a recurrent neural network. *EANN98 - Engineering Applications of Neural Networks*, Gibraltar, 9-12 June 1998, 118-121.
- [11] Jin, L., N. Nikiforuk, M. Gupta (1995). Approximation of discrete time state space trajectories using dynamic recurrent networks. *IEEE Trans. Automatic Control*, 40 (7), 1266-1270.
- [12] Ketata, R., D. Geest, D. Titli (1995). Fuzzy controller: design, evaluation, parallel and hierarchical combination with a PID controller. *Fuzzy Sets and Systems*, 71, 113-229.
- [13] Lee, C. (1990). Fuzzy logic in control systems: fuzzy logic controller - parts 1 and 2. *IEEE Trans. Systems Man Cybernetics*, 20, 404-435.
- [14] López, J. (1996). Acurex Filed D.A.S. and Control System. *Internal Report 08/96*, Plataforma Solar de Almería.
- [15] Mamdani, E. (1974). Application of fuzzy algorithms for control of a simple dynamic plant. *IEE Proc. D*, 121, 1585-1588.
- [16] Oliveira, P., P. Lima, J. Sentiero (1991). Fuzzy Supervision on Intelligent Control Systems - *European Control Conference ECC91*, Grenoble, France, 1226-1231.
- [17] Pham, D., L. Xing (1995). Dynamic System Identification Using Elman and Jordan Networks. *Neural Networks for Chemical Engineers*, Editor A. Bulsari, Chap. 23, 572-591.

- [18] Pickhardt, R., H. Unbehauen (1994). An indirect adaptive predictive controller for linear and non-linear plants. *Proceedings of the American Control Conference*, 2585-2589.
- [19] Pickhardt, R., R. Silva (1998). Application of a non-linear predictive controller to a solar power plant. *Proceedings of the IEEE Conference on Control Applications*, Trieste.
- [20] Rubio, F., M. Berenguel, E. Camacho (1995). Fuzzy logic control of a solar power plant. *IEEE Transactions on Fuzzy Systems*, 3 (4), 459-468.
- [21] Takagi, T, M. Sugeno (1985). Fuzzy identification of systems and its application to modelling and control. *IEE Trans. Man. Cybernetics*, 15, 116-132.
- [22] Werbos, P. (1990). Backpropagation through time: what it does and how do it. *Proc. IEEE*, 78, 1550-1560.
- Ying, H. (1998). Constructing Non-linear Variable Gain Controllers via the Takagi-Sugeno Fuzzy Control. *IEEE Transactions on Fuzzy Systems*, 6 (2), 226-234.
- [23] Zadeh, L. (1965). Fuzzy sets. *Inform. Control*, 8, 338-353.



## APPLICATION OF A RECURRENT NEURAL NETWORK IN ON-LINE MODELLING OF REAL-TIME SYSTEMS

Jorge Henriques<sup>†\*</sup>, Paulo Gil<sup>†‡</sup>, António Dourado<sup>†</sup> and H. Duarte-Ramos<sup>‡</sup>  
{jh, pgil, dourado}@dei.uc.pt                      hdr@mail.fct.unl.pt

<sup>†</sup>CISUC - Informatics Engineering Department, UC, Pólo II, 3030 Coimbra - Portugal  
Phone: +351 39 790000 - Fax: +351 39 701266

<sup>‡</sup>Electrical Engineering Department, UNL, 2825 Monte da Caparica - Portugal  
Phone: +351 1 2948545 - Fax: +351 1 2948532

**ABSTRACT:** Given the universal approximation properties, simplicity as well its intrinsic analogy to the non-linear state space form, a recurrent Elman network is derived and applied for modelling non-linear plants. Learning is implemented on-line, based on input and output data and using a truncated backpropagation through time algorithm. Regarding its structural simplicity, previous knowledge based on a linear description of the plant to be modelled might be used for initialising the network weights. The main goal of this work is to emphasise the potential benefits of this architecture for real-time identification. Experimental results collected from a laboratory heating system, for several operating conditions, confirm the viability and effectiveness of the proposed methodology.

**Keywords:** Recurrent Elman networks; modelling; on-line learning; real-time systems.

### 1. INTRODUCTION

It has been recognised since early that neural networks (NN) offer a number of potential benefits for application in the field of control engineering, particularly for modelling non-linear systems. Some appealing features of NN are its ability for learning through examples, they do not require any *a priori* knowledge and can approximate arbitrary well any non-linear continuous function, Hornik, and White (1989). Among the several architectures found in literature, recurrent neural networks (RNN) involving dynamic elements and internal feedback connections, have been considered as more suitable for this purposes than feedforward networks, Linkens and Nyongesa (1996).

In the last few years, various works have been presented showing that recurrent neural networks are quite effective in modelling non-linear dynamical systems, Parlos and Atiya (1994), Draye and Libert (1996), Kosmatopoulos and Iannou (1995). The critical issue in the application of RNN is the choice of the network architecture, that is, the number and type of neurons, the location of feedback loops and the development of a suitable training algorithm.

Despite the great potential that dynamic recurrent neural networks hold, Jin and Gupta (1995), a successful practical application is conditioned by several drawbacks. Among them, are clearly identifiable the computational efficiency of the learning stage, which depends on the initial weights; the information content of what is learned, which depends on the data set; since they are black-boxes models, the real structure of the plant is not captured and hence not accessible, Filho and Soares (1988).

Considering the number of recurrent neural topologies and training algorithms available, the choice of an appropriate pair (architecture, learning) is intimately dependent on the purposes and can be decisive for its success, e.g. the non-linear control schemes with NN identification. In this field, usually it is possible to derive at least an approximate linear model of the plant. Thus, it seems quite appropriate to incorporate available information from the approximate model into the NN initialisation, instead of choosing the weighting values randomly. In addition, for those plants having strong non-linear features or structural changes or even unmodelled dynamics an on-line learning with NN is of capital importance.

In the present work the combination of a modified recurrent Elman network with a truncated backpropagation through time algorithm (BTT) is applied for modelling non-linear plants. It is intended with this approach to profit from the simplicity of Elman networks and the fast training provided by the truncated BTT algorithm in order to develop a working practical strategy for real-time applications.

---

\* Author to whom correspondence should be addressed

## 2. ELMAN NETWORK: MODEL AND TRAINING

For modelling purposes it is assumed that the plant to be controlled is a multivariable plant, with  $m$  inputs and  $q$  outputs, described by a general non-linear input-output discrete time state space model:

$$x(k+1) = f \{ x(k), u(k) \} \quad (1)$$

$$y(k) = g \{ x(k) \} \quad (2)$$

where  $f: \mathfrak{R}^{n+p} \rightarrow \mathfrak{R}^n$  and  $g: \mathfrak{R}^n \rightarrow \mathfrak{R}^q$  are non-linear functions;  $u(k) \in \mathfrak{R}^m$ ,  $y(k) \in \mathfrak{R}^q$  and  $x(k) \in \mathfrak{R}^n$  are, respectively, the input vector, the output vector and the state vector, at a discrete time  $k$ .

### 2.1 - ELMAN NETWORK ARCHITECTURE AND DYNAMICS

Elman (1990) has proposed a partially recurrent network, where the feedforward connections are modifiable and the recurrent connections are fixed. Additionally to the input and the output units, the Elman network has a hidden unit,  $x^h(k) \in \mathfrak{R}^n$  and a context unit,  $x^c(k) \in \mathfrak{R}^n$ .  $W^x \in \mathfrak{R}^{n \times n}$ ,  $W^u \in \mathfrak{R}^{n \times p}$  and  $W^y \in \mathfrak{R}^{q \times n}$  are the interconnection matrices, respectively, for the context-hidden layer, input-hidden layer and hidden-output layer. Theoretically, an Elman network with  $n$  hidden units is able to represent a  $n^{\text{th}}$  order dynamic system. However, due to practical difficulties with the identification of higher order systems, some modifications have been proposed. In Pham and Xing (1995) a self-connection  $\alpha \in \mathfrak{R}^+$  in the context unit is introduced, Figure 1, improving its memorisation ability.

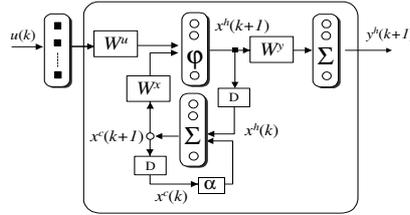


Figure 1: Block diagram of a modified Elman network.

The dynamics of the modified Elman neural network is described by the difference equations (3)-(6).

$$s(k+1) = W^x x^c(k+1) + W^u u(k) \quad (3)$$

$$x^h(k+1) = \varphi \{ s(k+1) \} \quad (4)$$

$$x^c(k+1) = x^h(k) + \alpha x^c(k) \quad (5)$$

$$y^h(k+1) = W^y x^h(k+1) \quad (6)$$

where  $s(k) \in \mathfrak{R}^n$  is an intermediate variable and  $\varphi(\cdot)$  is an hyperbolic tangent function.

### 2.2 - LEARNING METHODOLOGY

The main difficulty related to the recursive training of recurrent networks arises from the fact that the output of the network and its partial derivatives with respect to the weights depend on the inputs since the beginning of the training process and on the initial state of the network. Therefore, a rigorous computation of the gradient, which implies taking into account all the past history is not practical. In this work, however, the gradient is approximated considering a finite number  $N$  of previous sampling periods. The training is defined on a sliding window mode, Henriques and Dourado (1998), where the identification criterion in the horizon  $[k-N, K, k]$  is defined by:

$$E(k) = \sum_{i=k-N}^k e(i)^2 \quad (7)$$

where  $e(k) \in \mathfrak{R}^q$  is the modelling error at time step  $k$ , given by:

$$e(k) = y(k) - y^h(k) \quad (8)$$

Several training algorithms have been proposed to adjust the weight values in recurrent networks. Examples of these methods are the dynamic backpropagation from Narendra and Parthasarathy (1991), the real time recurrent algorithm from Williams and Ziepsler (1995) and the backpropagation through time from Werbos (1990), among others. This latest method is considered in the present work. For updating  $W^x$ ,  $W^u$  and  $W^y$  a gradient type algorithm is used as follows:

$$\Delta W^i(k+1) = \rho_m \Delta W^i(k) - \mu_m (1 - \rho_m) \frac{\partial E(k)}{\partial W^i(k)}, i = u, x, y \quad (9)$$

where  $\mu_m \in \mathfrak{R}^+$  is the learning rate and  $\rho_m \in \mathfrak{R}^+$  is a momentum term.

The recurrent network is expanded into a multilayer feedforward network, being a new layer added at each time step. The computation of the derivatives in (9) is then performed as in the standard feedforward backpropagation network case, Rumelhart and Williams (1986), according to (10), (11) and (12).

$$\frac{\partial E(k)}{\partial W^x} = \sum_{i=k-N}^k \delta^h(i) x^c(i)^T \quad (10)$$

$$\frac{\partial E(k)}{\partial W^u} = \sum_{i=k-N}^k \delta^h(i) u(i-1)^T \quad (11)$$

$$\frac{\partial E(k)}{\partial W^y} = \sum_{i=k-N}^k \delta^y(i) x^h(i)^T \quad (12)$$

The values of  $\delta^h(i) \in \mathfrak{R}^n$  and  $\delta^y(i) \in \mathfrak{R}^q$  are computed recursively for  $i \in [k-N, k-1]$  by (13)-(17).

$$\varepsilon^y(i) = \delta^y(i) = e(i) \quad (13)$$

$$\varepsilon^h(i) = W^{yT} \delta^y(i) + I_n \delta^c(i+1) \quad (14)$$

$$\delta^h(i) = \varepsilon^h(i) \otimes \varphi'\{s(i)\} \quad (15)$$

$$\varepsilon^c(i) = \alpha I_n \delta^c(i+1) + W^{xT} \delta^h(i) \quad (16)$$

$$\delta^c(i) = \varepsilon^c(i) \quad (17)$$

The process is initialised at time  $k$  according to:

$$\varepsilon^y(k) = \delta^y(k) = e(k) = y(k) - y^h(k) \quad (18)$$

$$\delta^h(k) = \varepsilon^h(k) \otimes \varphi'\{s(k)\} \quad (19)$$

$$\varepsilon^c(k) = W^{xT} \delta^h(k) \quad (20)$$

The symbol  $\otimes$  denotes the multiplication element by element,  $I_n$  is an identity matrix and  $\varphi'(\cdot)$  is the derivative of the hyperbolic tangent function  $\varphi(\cdot)$ .

### 2.3 – INITIALISATION

Considering the non-linear neural network (3)-(6), a linear model can be derived by a Taylor series expansion of (21) and neglecting the context unit contribution, yielding (22).

$$x^h(k+1) = \varphi \left\{ W^x x^h(k), \alpha W^x x^c(k), W^u u(k) \right\} \quad (21)$$

$$x^h(k+1) = W^A x^h(k) + W^B u(k) \quad (22)$$

where

$$W^A = \frac{\partial \varphi\{\cdot\}}{\partial x^h(k)} = \text{diag}[\varphi'\{\cdot\}] W^x \quad (23)$$

$$W^B = \frac{\partial \varphi\{\cdot\}}{\partial u(k)} = \text{diag}[\varphi'\{\cdot\}] W^u \quad (24)$$

Having a linear state space model describing the process to be identified (25)-(26) it is possible to provide previous knowledge to the network in order to initialise its weights, given the relationship between equations (22) and (25) and equations (26) and (6).

$$x(k+1) = A x(k) + B u(k) \quad (25)$$

$$y(k) = C x(k) \quad (26)$$

### 3. EXPERIMENTAL RESULTS

#### 3.1 – THE HEATING SYSTEM

In the laboratory process, depicted in Figure 2, air is forced to circulate by a fan blower through a duct and is heated at its inlet. This is a non-linear process with a pure time delay, which depends on the position of the temperature sensor element and the air flow rate, depending on the damper position  $\Omega$ . The system input  $u(k)$ , is the voltage on the heating device, which consists of a mesh of resistor wires, and the output,  $y(k)$ , is the outlet air temperature.

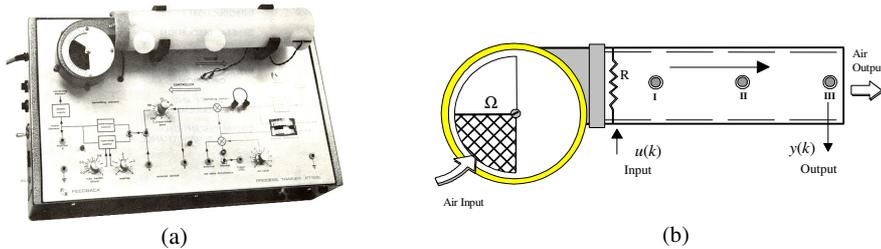


Figure 2: The Laboratory Process. (a) picture; (b) schematic diagram.

#### 3.2 - EXPERIMENTS

In order to assess the performance of the proposed on-line identification strategy a set of experiments was carried out. The experiments have been conducted using a common PC and the algorithms have been implemented in C code. For sampling time was chosen 0.25 second. In the experiments input variations and plant dynamics change have been considered. Additionally, the capability of prediction when the process identification is turned-off was also investigated. In all experiments the air damper was positioned at  $\Omega = 10^\circ$  and the sensor located at position III. Regarding the neural network, the number of hidden units (equal to the number of context units) is equal to three,  $n = 3$ . For the truncated backpropagation learning algorithm the following parameters were chosen as:  $\mu_m = 0.02$ ,  $\rho_m = 0.3$ ,  $\alpha = 0.4$  and  $N = 6$ .

In Figure 3 are depicted the on-line identification with previous knowledge initialisation (a) and randomly initialisation (b). As can be seen, despite both initialisations are performing quite well, the one being carried out with previous knowledge exhibits a faster initial learning, as was expected.

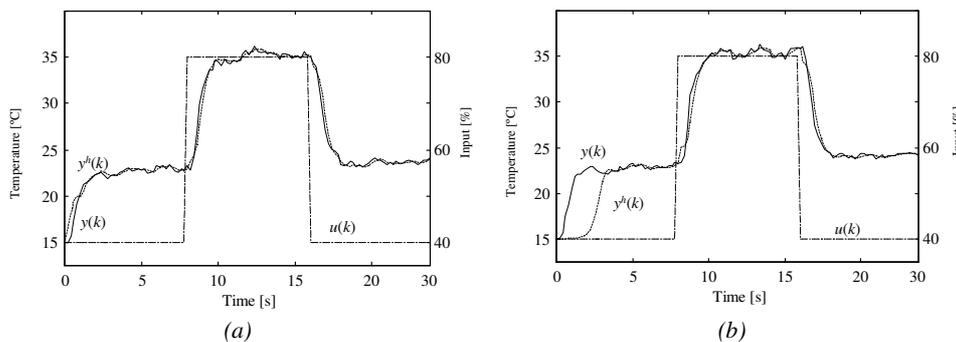


Figure 3: Identification (a) initial knowledge; (b) randomly initialisation.

For different inputs, such as, square and sinusoidal waves, the proposed approach give results quite satisfactory as illustrated in Figure 4, even when varying the relevant characteristics of the waves (amplitude and frequency).

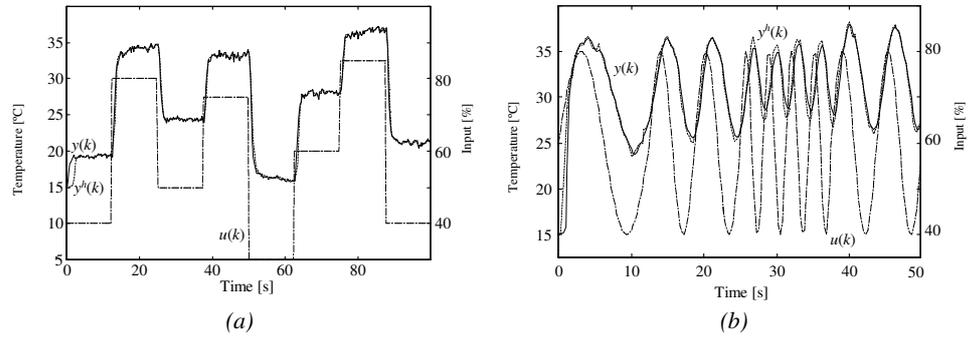


Figure 4: Identification (a) input square wave; (b) sinusoidal wave.

In Figure 5 is illustrated the utilisation of neural modelling for prediction. In (a) an on-line identification with a sliding window  $N = 6$  was carried out up to the instant 25 second, being the output prediction performed from instant 25 to 50 second, freezing the network weights. In (b) an off-line identification was performed at instant 37.5 second with data collected over one wave period,  $N = 100$  (25 second), and the output predicted from this time on with this static neural model. As expected, the neural model is intimately dependent on the data set used for learning and thus can only be applied accurately for prediction in the same conditions of the data set training.

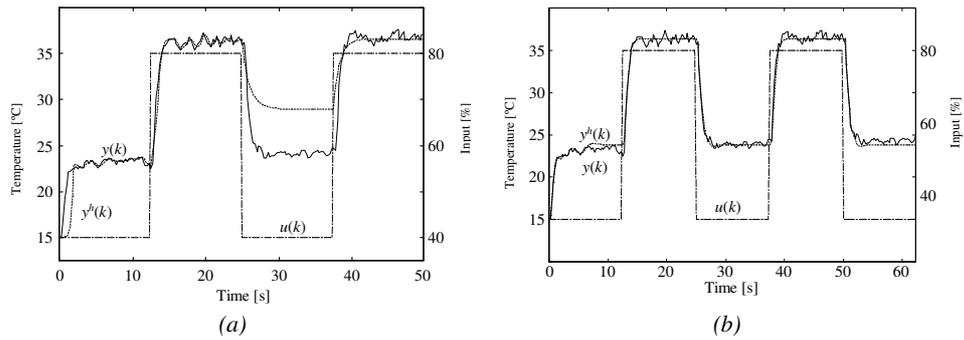


Figure 5: Identification and prediction (a) window size equal to 6; (b) window size equal to 100.

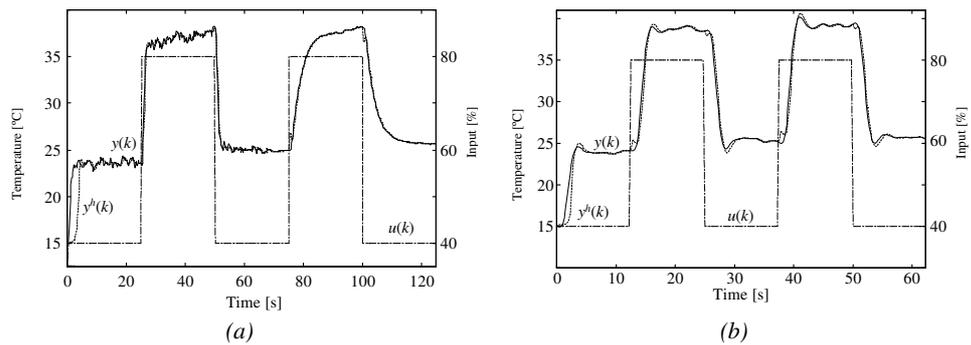


Figure 6 : Identification (a) additional first order system; (b) additional time delay.

Finally, in Figure 6 is illustrated the plant dynamics modification with the inclusion of an additional first order system (constant time equal to 3 second) at instant 70 second, figure (a), and an additional one second delay, figure (b). As can be observed, the on-line neural learning performs very satisfactory in both cases, which reveals clearly the adaptive features of this methodology. Despite the identification behaviour in figure (b) is satisfactory, there exist at the beginning of each input transition an anticipation of the neural output. This is justified by the fact that the number of hidden unit is not enough for capturing the dynamics accurately. Experiments carried out using a higher number of hidden units reveal a superior performance and is not observed the reported behaviour.

#### 4. CONCLUSIONS

In this paper the application of a modified recurrent Elman network as general tool for modelling together with a truncated backpropagation through time algorithm for training have been presented. Given the simplicity of this network topology along with the fast training provided by the proposed learning algorithm, this approach is proved from the experiments on the laboratory plant, to be a feasible alternative for real-time identification. In this context, given the adaptive features revealed by the Elman networks as well their ability for modelling any kind of non-linearities in the form of non-linear state space representation, they are bringing a valuable added-value to the control field and particularly to those strategies using an explicit model of the plant to be controlled.

#### ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under the program PRAXIS XXI.

#### REFERENCES

- Draye, J., Pavisic, D., Libert, G., 1996, "Dynamic recurrent neural networks: a dynamical analysis", *IEEE Trans. on Systems Man and Cybernetics*, Part B, vol 26, n° 5, pp. 692-706.
- Elman, J., 1990, "Finding Structure in time", *Cognitive Science*, 14, pp 1789-211.
- Filho, B., Cabral, E. Soares, A., 1998, "A new approach to artificial neural networks", *IEEE Trans. on Neural Networks*, vol 9, n° 6, pp. 1167-1179.
- Henriques J.; Dourado, A., 1998, "A Multivariable adaptive control using a recurrent neural network" *Proceedings of Eann98 – Eng. Applications of Neural Networks*, Gibraltar, 9-12 June, pp 118-121.
- Hornik, K.; Stinchcombe, M.; White, H., 1989, "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, pp 359-366.
- Jin, L.; Nikiforuk, P.; Gupta, M., 1995, "Approximation of discrete time state space trajectories using dynamic recurrent networks", *IEEE Trans. Automatic Control*, 40, n° 7, pp 1266-1270.
- Kosmatopoulos, E., Polycarpou, M., Iannou, A., 1995, "High-order neural network structures for identification of dynamical systems", *IEEE Trans. on Neural Networks*, vol 6, n° 2, pp. 422-431.
- Linkens, D. Nyongesa, Y., 1996, "Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications", *IEE Proc. Control Theory App.*, 134 (4), pp 367-385.
- Narendra, K.; Parthasarathy, K., 1991, "Gradient methods for the optimisation of dynamical systems containing neural networks", *IEEE Trans. on Neural Networks*, 2, n° 2, pp 252-262.
- Parlos, A., Chong, K., Atiya, A., 1994, "Application of the recurrent multilayer perceptron In modelling complex process dynamics", *IEEE Trans. on Neural Networks*, vol 5, n° 2, pp. 255-266.
- Pham D.; Xing, L., 1995, "Dynamic System Identification Using Elman and Jordan Networks", *Neural Networks for Chemical Engineers*, Editor A. Bulsari, Chap. 23, pp 572-591.
- Rumelhart, D.; Hinton, G.; Williams, R., 1986, "Learning internal representations by error propagation", *Explorations in the microstructure of cognition. Vol 1: Foundations*, Cambridge, MIT Press/Bradford books.
- Werbos, P., 1990, "Backpropagation through time, what it does and how do it", *Proc. IEEE*, 78, pp 1550-1560.
- Williams, R.; Zipser, D., 1995, "Gradient-based learning algorithms for recurrent networks and their computational complexity", *Backpropagation*, Edit by Yves Chauvin and D. Rumelhart, Chap.13, pp 433-486.

## MULTIVARIABLE ADAPTIVE CONTROL USING AN OBSERVER BASED ON A RECURRENT NEURAL NETWORK

J. HENRIQUES\* AND A. DOURADO

*CISUC - Centro de Informática e Sistemas da Universidade de Coimbra, Departamento de Engenharia Informática  
Pólo II, UC, 3030 Coimbra - Portugal*

### SUMMARY

A real time learning control technique for a general non-linear multivariable process is presented and applied to a laboratory plant. The proposed technique is a hybrid approach, which combines the ability of a recurrent neural network for modelling purposes and a linear pole placement control law to design the controller, providing a bridge between the field of neural networks and the well known linear adaptive control methods.

An Elman type recurrent neural network strategy is introduced to model the behaviour of the non-linear plant, using available input-output data (an unmeasurable states problem is assumed). Following a linearization technique a linear time varying state space model is obtained, which allows simultaneous estimation of parameters and states. Once the neural model is linearized, some well-established standard linear control strategies can be applied. With simultaneous online training of the neural network and controller synthesis, the resulting structure is an indirect adaptive self-tuning strategy.

The identification and control performances of the proposed approach are investigated on a non-linear multivariable three tanks laboratory system. Experimental results show the effectiveness of the proposed hybrid structure. Copyright © 1999 John Wiley & Sons, Ltd.

**Keywords:** Recurrent neural networks, non-linear observer, adaptive control, decoupling.

### 1. INTRODUCTION

It has been recognized that neural networks (NN) offer a number of potential benefits for application in the field of control engineering, particularly in modelling the dynamics of non-linear multivariable systems. Recurrent neural networks (RNN), which involve dynamic elements and have feedback connections, have been considered more suitable for this purpose than feedforward networks<sup>1</sup>. It has been shown,<sup>2,3</sup> in continuous and in discrete time respectively, that a RNN may be used to approximate, with an arbitrary precision, the behavior of a non-linear dynamical system.

\*Correspondance to: J. Henriques, CISUC – Centro de Informática e Sistemas da Universidade de Coimbra, Departamento de Engenharia Informática, Pólo II, UC – 3030 Coimbra – Portugal : jh@dei.uc.pt

Contract/grant sponsor: Portuguese Ministry of Science and Technology (MCT)  
Contract/grant number: PRAXIS XXI

CCC 0890-6327/99/040241-19\$17.50

Copyright © 1999 John Wiley & Sons, Ltd.

A main difference between NN controllers and linear controllers is that in NN controllers the models are parameterized by using a non-linear model instead of linear mappings. This non-linear mapping makes the control problem much harder to design and analyze. It would be of great interest to profit from the well established linear control theory to ease these task, as is done in this paper. The present work consists on a hybrid scheme, and the main objective is to investigate the combination of the potentiality of NN to model dynamical systems and the well known theory of linear control.

This ability of NN to model non-linear systems and their capability of learning trough examples has led to the development of several neural based control strategies<sup>4,5</sup>. In particular, great attention has been paid to the computational features of non-linear NN for adaptive control. The idea of applying NN in this field appeared in<sup>6</sup> and<sup>7</sup>. It has been proposed in reference 7 that a more general flexible control law could be provided using the non-linear function approximation property of NN. Such a NN controller could be trained on-line to provide a non-linear control law, capable of adequate control of the non-linear plant over the operating range. Chen<sup>6</sup> has proposed a scheme for combining backpropagation neural networks with self-tuning adaptive control for the *affine* class of non-linear systems. Chen and Khalil<sup>8</sup> has presented an analytical study of the use of multilayer neural networks (MNN) in the control of the *affine* class of non-linear discrete time systems. Jin *et al.*<sup>9</sup> have proposed an adaptive scheme for a general class of unknown MIMO (Multiple Input Multiple Output) discrete time non-linear systems. Based on the neural network model an extension of the concept of input-output linearisation is used to synthesize a control technique for model control purposes. In reference 10 the application of a neural network, for direct model reference adaptive control, is investigated. In particular, it is proposed an hybrid control structure consisting of the neural network in parallel with a linear control law. Delgado<sup>11</sup> has proposed a generalization of the Hopfield NN to identify and control the *affine* class of control systems. A static feedback that cancels the non-linear terms of the plant is proposed, yielding a linear plant that can then be controlled using a standard PID controller.

In practice most of control applications have to deal with unmeasurable states, using only input-output information, which justify the use of input-output models. However, the use of a state vector representation, for a dynamical system, is particularly suited for control purposes. To overcome this problem, some works have appeared that suggest using NN to solve state estimation and identification problems for non-linear systems. Sørensen<sup>12</sup> has proposed a neural network to act as an observer for a multivariable, non-linear dynamic system. Training is performed off-line and the design is inspired in the linear observer technique. Kim *et al.*<sup>13</sup> have presented a generalized dynamic RNN to design an adaptive observer for a particular class of non-linear systems, where a NN is inserted in the feedback path to capture the non-linearity of the observed system. Puskorius and Feldkamp<sup>14</sup> have presented a decoupled extended Kalman filter algorithm for training a RNN. He suggested that RNN trained by Kalman filter methods could combine the traditional features of state space controllers and observers. In reference 15 a prediction error algorithm for neural state space models is developed, both for the deterministic and stochastic cases. For the stochastic case a predictor with direct parameterization of the Kalman gain by a neural network architecture is proposed. In reference 16 for the state estimation problem a non-linear filter is assigned a given structure in which the values of certain number of parameters have to be determined, using a non-linear programming so as to minimize a cost function. The filter structure is implemented by means of a multilayer feedforward neural network in which the unknown parameters are the weights of the network.

A well know technique to deal with non-linear control systems is based on a linearization of the non-linear model. At a given operating point, a nominal linear model of the plant is obtained and the controller is design using some well-established standard linear control strategy. Concerning NN models linearization some works have been proposed. Ahmed and Tasaddup<sup>17</sup> proposed a control scheme based on plant linearisation at each operating point. Following this author, since control design for linear systems has been well developed, it is natural to make use of it in non-linear plants. Sørensen<sup>18</sup> has shown the possibility to extract online, from a neural model, the actual linearized parameters by taking the derivatives of the outputs with respect to the inputs. Using the features of this strategy for parameter estimation, a conventional pole placement adaptive controller can be adopted for non-linear control. In reference 15 a linear fractional transformation representation is derived, making possible to interpret the non-linear neural networks as a nominal linear model. This linear model is then used in the design of a standard robust control scheme.

In this paper a simple hybrid strategy for combining the approximation ability of a recurrent Elman's network and a decoupling pole placement control law is investigated. A standard linear state space model of the form  $x(t + 1) = Ax(t) + Bu(t)$  is derived from a linearization technique and a state space feedback controller,  $u(t) = Gr(t) - Fx(t)$ , combining simultaneously decoupling and pole placement design is then applied.

This paper is organized as follows. In section 2 the architecture of the particular recurrent NN is described. The modeling problem is stated in a general input-output problem and a variation of the backpropagation through time algorithm is proposed to adjust the weights of the network. In section 3 an observer technique, allowing simultaneous estimation of parameters and states is derived. Based on the linearization of the network a time varying state space linear model is obtained. The control parameters are updated using a decoupling pole placement control law resulting in an indirect adaptive self-tuning scheme. Following this, a strategy for incorporating initial knowledge in neural networks is presented. In section 4, in order to illustrate the applicability of the proposed method, control experiments on a three interconnected tanks system are performed. The analysis of the results shows the effectiveness of the proposed approach. Finally, section 5 concludes the paper.

## 2. SYSTEM IDENTIFICATION WITH RECURRENT NETWORKS

### 2.1. Recurrent Networks

In general a recurrent network is composed of  $m$  neuron units, and  $p$  external inputs. Let  $x_k(t)$  denote the state of the  $k^{th}$  neuron unit in the network at time  $t$ ,  $u_k(t)$  denote the  $k^{th}$  element of the external input in the network at time  $t$  and let  $v_k(t)$  be defined as:

$$v_k(t) = \begin{cases} u_k(t) & \text{if } k \in U \\ x_k(t) & \text{if } k \in X \end{cases} \quad (1)$$

where  $U$  denotes the set of indices  $k$  such that  $u_k(t)$  is an external input and  $X$  denotes the set of indices for which  $x_k(t)$  is the neuron unit in the network. Or, in vector notation:

$$v = \begin{bmatrix} u \\ x \end{bmatrix} \quad (2)$$

where  $u$  and  $x$  represent two vectors, respectively with  $m$  and  $p$  elements.

The dynamics of each neuron unit is described by the difference equations (3) and (4), where every self-connection in the network is assumed to have a delay of one time step (D), that is,

$$s_k(t+1) = \sum_{i \in U} w_{ki}^u u_i(t) + \sum_{i \in X} w_{ki}^x x_i(t) \quad (3)$$

$$x_k(t+1) = \varphi \{ s_k(t+1) \} \quad (4)$$

In the above equation  $w_{ki}^x$  represents the weight of the connection from the  $i^{\text{th}}$  neuron to the  $k^{\text{th}}$  neuron,  $w_{ki}^u$  represents the weight of the connection from the  $i^{\text{th}}$  input neuron to the  $k^{\text{th}}$  neuron,  $s_k(t)$  is an intermediate variable and  $\varphi$  is a non-linear function, an hyperbolic tangent, given by (5):

$$\varphi(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}, a \in \Re \quad (5)$$

Equations (3) and (4) can be rewritten as equation (6):

$$x_k(t+1) = \varphi \left\{ \sum_{i \in U \cup i \in X} w_{ki} v_i(t) \right\} \quad (6)$$

where  $w_{ki}$  is the  $ki^{\text{th}}$  component of the complete weight matrix  $W$ . This equation can be rewritten, similarly to a non-linear state space model as equation (7)

$$x(t+1) = \varphi \{ x(t), u(t), W \} \quad (7)$$

Figure 1 shows a recurrent network with three neuron units and two external inputs. For this particular example the neuron outputs are given by (8), with  $i=1,2,3$ .

$$x_i(t+1) = \varphi \{ w_{i1} u_1(t) + w_{i2} u_2(t) + w_{i3} x_1(t) + w_{i4} x_2(t) + w_{i5} x_3(t) \} \quad (8)$$

## 2.2 - Backpropagation Through Time

The neural network model considered is employed to approximate the behavior of a general non-linear plant. Assuming that  $d(t)$  denotes the desired plant neuron output values at instant  $t$ , the objective is to find the correct weight values  $W$  such that the squared error between the neurons of the network and the desired states, defined by (9), is minimized:

$$E(t) = \frac{1}{2} \sum_{k \in X} \{ e_k(t) \}^2 \quad (9)$$

The error term  $e_k(t)$  is computed as

$$e_k(t) = d_k(t) - x_k(t) \quad (10)$$

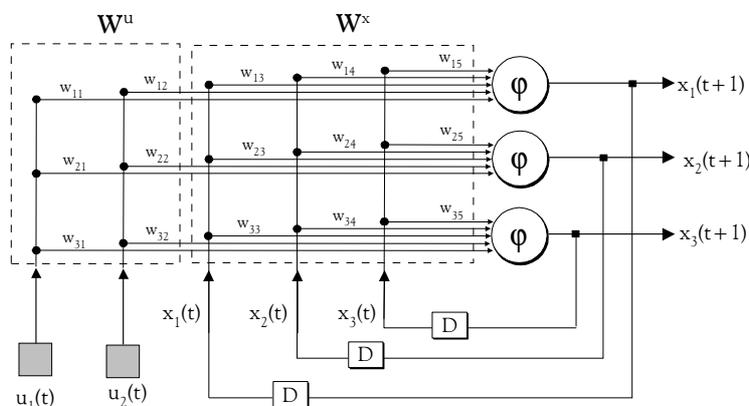


Figure 1 - Example of a recurrent network with two inputs and three neuron units.

Several training algorithms have been proposed to adjust the values of  $W$  in recurrent networks. Examples of these methods are the Narendra's dynamic backpropagation<sup>19</sup>, the real time recurrent algorithm of Williams and Ziepsen<sup>20</sup> and the Werbos' backpropagation through time (BTT)<sup>21</sup>. These methods use a gradient based learning algorithm and involve the computation of partial derivatives or sensitivity functions. Using a gradient based learning algorithm, the weights changes are derived from (11)

$$\Delta w_{ij}(t+1) = w_{ij}(t+1) - w_{ij}(t) \tag{11}$$

and are updated each time in the following way (equation (12)):

$$w_{ij}(t+1) = w_{ij}(t) - \mu \frac{\partial E(t)}{\partial w_{ij}(t)} \tag{12}$$

where  $\mu$  is a positive learning rate parameter. The backpropagation through time (BTT) is a gradient type algorithm, based on an extension of the standard backpropagation (BP) for feedforward networks<sup>22</sup>, and is applied in this work. A recurrent network is expanded into a multilayer feedforward network, where a new layer is added for each time step. The computation of the derivatives is then performed as in a standard feedforward backpropagation network. Figure 2 shows an unfolded feedforward network consisting of  $N+1$  layers, which corresponds to a recurrent network's run from time  $t_0$  to  $t_N$ .

Using the standard BP, the computation of the gradient of  $E(t_N)$ , at time  $t_N$ , is given by (13):

$$\frac{\partial E(t_N)}{\partial w_{kj}} = \sum_{t=t_1}^{t_N} \delta_k(t) v_j(t-1) \tag{13}$$

The values of  $\delta_k(t)$ , the sensitivity to the unit's net inputs, are computed recursively for  $k \in X$  and for  $t \in [t_1, t_N]$  by equations (14) and (15):

$$\delta_k(t) = \varepsilon_k(t) \cdot \varphi' \{ s_k(t) \} \tag{14}$$

$$\varepsilon_k(t-1) = \sum_{i \in U \cup i \in X} w_{ik} \delta_i(t) \tag{15}$$

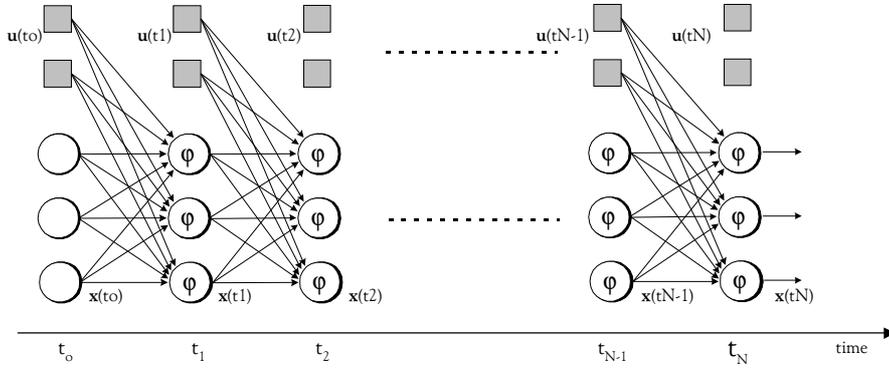


Figure 2 - A schematic diagram for an unrolled unfolded network.

The process start at time  $t_N$  with

$$\varepsilon_k(t_N) = e_k(t_N) = d_k(t_N) - x_k(t_N) \tag{16}$$

Using this algorithm only the final target values, at instant  $t_N$ , are specified. If the target values  $d(t)$  are given for the units in other layers, than the last, the total gradient evaluated from  $t \in [t_1, t_N]$ , is obtained using the same procedure, modifying equation (15) to equation (17):

$$\varepsilon_k(t-1) = e_k(t-1) + \sum_{i \in U \cup i \in X} w_{ik} \delta_i(t) \tag{17}$$

Theoretically this algorithm makes use of potentially unbounded history storage, and therefore it is sometimes denoted by BTT( $\infty$ ). A common way to simplify the computational requirements is obtained by truncating the backpropagation of information to a fixed number of prior time steps. The resulting algorithm is the truncated BTT, BTT(N), where N represents the number of prior time steps considered. Following this algorithm some dependencies in the network are ignored. However, as will be seen in the next section, if the dependence of past information exponentially decays backward in time, this simplification can give a reasonable approximation for the true gradient.

Normally the standard steepest descent method tends not only to be slow but also ineffective at finding a good solution. To avoid this, several variations/new algorithms of the basic algorithm that provide significant speedup have been proposed. Examples of these variations are the use of momentum, variable learning rate, conjugate gradient and the Levenberg-Marquardt method. Here, only a momentum term is used, and the equation (11) results in (18):

$$\Delta w_{ij}(t+1) = \eta \Delta w_{ij}(t) - \mu (1 - \eta) \frac{\partial E(t)}{\partial w_{ij}(t)} \tag{18}$$

where  $\eta$  is a momentum factor, a positive constant. Though this learning method is very simple it has performed very well on various experiments.

An important problem in the application of neural networks to control, is the selection of an appropriate architecture. Due to some of its features, like the ability to approximate the dynamics of a discrete time non-linear system, and due to its interpretation as a non-linear state space model, an Elman network is considered in this work. For modelling purposes a multivariable plant with  $p$  inputs and  $q$  outputs is assumed, which is described by a non-linear input-output discrete time model. The output vector  $y(t) \in \mathfrak{R}^q$ , at a particular discrete time  $t$ , can be specified as a non-linear function of past outputs and past inputs:

$$y(t) = \phi \{ y(t-1), y(t-2), \Lambda, u(t-1), u(t-2), \Lambda \} \tag{19}$$

where  $\phi$  is a non-linear function and  $u(t) \in P$  is the input vector, at instant  $t$ . For control purposes, the use of a state vector representation is particularly suited. To deal with this situations (modelling based and input-output information and design controller based on state variables) an observer technique will be introduced, and a state space non-linear discrete state equation is then assumed in the form (20) and (21)

$$x(t+1) = f \{ x(t), u(t) \} \tag{20}$$

$$y(t+1) = g \{ x(t+1) \} \tag{21}$$

where  $x(t) \in n$  represents the state vector and  $f$  and  $g$  are non-linear functions.

Elman has proposed a recurrent network<sup>23</sup>, where the feedforward connections are modifiable and the recurrent connections are fixed. In Figure 3 the block diagram of an Elman network is shown.

The input  $u(t) \in P$  and the output  $y(t) \in q$  interact with the environment while the hidden unit  $x^h(t) \in n$  and the context unit  $x^c(t) \in n$  do not. The interconnection matrices respectively for the context-hidden layer, input-hidden layer and hidden-output layer are  $W^x \in n,n$ ,  $W^u \in n,p$  and  $W^y \in q,n$ . The dynamics of the neural network is described by the following difference equations (24).

$$x^h(t+1) = \phi \{ W^x x^c(t+1) + W^u u(t) \} \tag{22}$$

$$y(t+1) = W^y x^h(t+1) \tag{23}$$

$$x^c(t+1) = x^h(t) + \alpha x^c(t) \tag{24}$$

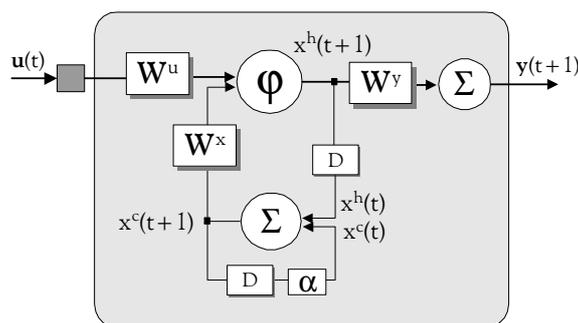


Figure 3 - Block diagram of an Elman network.

where  $\varphi$  is the hyperbolic tangent function (5), and the unrestricted function  $f(x)=x$  is used for the neurons in the output layer. The context units are locally recurrent (i.e., they feedback into themselves). The local recurrence decreases the values by a multiplicative constant  $\alpha$  (time constant) as they are fed back. This constant determines the memory depth (i.e., how long a given value fed to the context unit will be "remembered"). If an augmented state,  $\hat{x} \in \mathfrak{R}^{2n}$ , is defined by

$$\hat{x}(t) = \begin{bmatrix} x^h(t) \\ x^c(t) \end{bmatrix} \quad (25)$$

the equations (22-24) can be rewritten by

$$\hat{x}(t+1) = \varphi \left\{ \hat{x}(t), u(t), W^x, W^u \right\} \quad (26)$$

$$\hat{y}(t+1) = g \left\{ \hat{x}(t+1), W^y \right\} \quad (27)$$

Equations (26) and (27) can be seen as a state space model, similar to the non-linear system defined in (20) and (21).

#### 2.4 - Learning in Elman Networks

The evaluation of dynamic derivatives of recurrent network's outputs with respect to its weights is a computationally expensive procedure, since the derivatives cannot be evaluated from the standard backpropagation. Introducing this architecture Elman investigated the effect of cutting temporal history on backpropagation through time. By truncating the backpropagation of the information to a fixed number of prior time steps, on a sliding window mode, it is possible to save computation time and storage, but one may loose in terms of accuracy<sup>24</sup>.

Using equation (24) and considering N prior time steps, the context unit, at instant  $t$ , can be described as

$$x^c(t) = x^h(t-1) + \alpha x^h(t-2) + \Lambda + \alpha^{N-1} x^h(t-N) + \alpha^N x^c(t-N) \quad (28)$$

or

$$x^c(t) = \alpha^N x^c(t-N) + \sum_{i=1}^N \alpha^{i-1} x^h(t-i) \quad (29)$$

This equation shows that the output of the context unit is an integration of the output of the hidden unit. A value of  $\alpha$  near to 1 enables the context unit to remember more past outputs, and a value near to 0 enables the context unit to forget rapidly past information, similarly to a forgetting factor. For  $0 < \alpha < 1$ , the dependence of past information exponentially decays backward in time making the use of the truncated BTT an acceptable approximation of the true BTT in case the values of  $\alpha$  and N are correctly chosen, since (30) is verified.

$$\alpha^N \rightarrow 0 \quad (30)$$

### 3. CONTROL STRATEGY

#### 3.1 - Observer Design Using Elman Networks

Taking the non-linear neural network, described by the equations (22)-(24), instantaneous linear models can be derived. Combining the equations (22) and (24), equation (31) is obtained.

$$x^h(t+1) = \varphi \left\{ W^x x^h(t) + \alpha W^x x^c(t) + W^u u(t) \right\} \quad (31)$$

Computing the derivatives from the 'output' to the 'inputs', it is seen that equation (31) results in

$$x^h(t+1) = \frac{\partial \varphi\{\cdot\}}{\partial x^h(t)} x^h(t) + \frac{\partial \varphi\{\cdot\}}{\partial x^c(t)} x^c(t) + \frac{\partial \varphi\{\cdot\}}{\partial u(t)} u(t) \quad (32)$$

where

$$\frac{\partial \varphi\{\cdot\}}{\partial x^h(t)} = \text{diag}[\varphi'\{\cdot\}] W^x = \bar{A} \quad (33)$$

$$\frac{\partial \varphi\{\cdot\}}{\partial x^c(t)} = \alpha \text{diag}[\varphi'\{\cdot\}] W^x = \alpha \bar{A} \quad (34)$$

$$\frac{\partial \varphi\{\cdot\}}{\partial u(t)} = \text{diag}[\varphi'\{\cdot\}] W^u = \bar{B} \quad (35)$$

Using the augmented state  $\hat{x} \in \mathfrak{R}^{2n}$ , defined by equation (25), a linear state space model of the neural network can be rewritten by

$$\hat{x}(t+1) = W^A \hat{x}(t) + W^B u(t) \quad (36)$$

$$\hat{y}(t+1) = W^C \hat{x}(t+1) \quad (37)$$

where the matrices  $W^A \in \mathfrak{R}^{2n,2n}$ ,  $W^B \in \mathfrak{R}^{2n,p}$  and  $W^C \in \mathfrak{R}^{q,2n}$  are defined respectively by

$$W^A = \begin{bmatrix} \bar{A} & \alpha \bar{A} \\ I_{n,n} & \alpha I_{n,n} \end{bmatrix} \quad (38)$$

$$W^B = \begin{bmatrix} \bar{B} \\ \emptyset_{n,p} \end{bmatrix} \quad (39)$$

$$W^C = \begin{bmatrix} W^y & \emptyset_{q,n} \end{bmatrix} \quad (40)$$

The matrices  $\emptyset_{m,n}$  and  $I_{n,n}$  represent, respectively, a zero and an identity matrix of dimension  $(m,n)$  and  $(n,n)$ . The matrix  $\text{diag}[\varphi'\{\cdot\}]$  is a  $(n,n)$  diagonal one with elements

$$[\varphi'_1(t) \varphi'_2(t) \Lambda \varphi'_n(t)]$$

This linear state space model can act as an observer for the non-linear process defined in equations (22)-(24). The  $n$  first elements in the state vector  $\hat{x}(t)$  consist of the hidden states  $x^h(t)$  and the last elements consist of the context states  $x^c(t)$ , resulting in a non-minimal state dimension. The number of estimated states is twice the assumed order of the process, which can be seen as a disadvantage of the proposed estimator. An advantage is that no a priori assumptions are made about the process, with the exception that the maximum value for its order,  $n$ , is known in advance. Another advantage is the ability for incorporating prior modelling information into neural networks. Prior knowledge about a plant typically takes the form of a linear state space equation or a linear transfer function (which are equivalent for controllable and observable linear systems), of the form

$$x(t+1) = A x(t) + B u(t) \quad (41)$$

$$y(t+1) = C x(t+1) \quad (42)$$



the  $i^{th}$  input  $r_i(t)$  affects only the  $i^{th}$  output  $y_i(t)$ . Falb and Wollovith<sup>25</sup> have established this decoupling pole placement control law. Let  $d_1, \dots, d_q$  be define by

$$d_i = \min \left\{ j: C_i A^j B \neq 0 \right\}$$

or

$$d_i = n-1 \text{ if } C_i A^j B = 0 \text{ for all } j$$

Then the  $F$  and  $G$  matrices are computed respectively by

$$F = B^{*-1} \left[ \sum_{k=0}^{\lambda} \{ M_k C A^k + A^* \} \right]$$

$$G = B^{*-1}$$

where

$$B^* = \begin{bmatrix} C_1 A^{d_1} B \\ M \\ C_q A^{d_q} B \end{bmatrix}$$

$$A^* = \begin{bmatrix} C_1 A^{d_1+1} \\ M \\ C_q A^{d_q+1} \end{bmatrix}$$

The constant  $\lambda = \max \{d_i\}$  and the  $M_k$  matrices are suitably chosen to specify the closed-loop pole location. The number of closed-loop poles ( $np$ ) which can be specified is given by

$$np = q + \sum_{i=1}^q d_i$$

In order to improve the stability and to assure a zero steady state error a modification of the original algorithm was considered in this work. The closed-loop poles of the decoupled system were specified to be at  $z=1$  (the eigenvalues are chosen such  $eig(A-BF) = 1$ ) and an additional feedback path is introduced. For each subsystem

$$r_i(t) = \beta_i [r_i(t) - y_i(t)]$$

where  $\beta_i$  is a positive constant. This strategy is similar to control an open loop system which has an integral action. A block diagram for a two input-two output case is presented in Figure 5.

#### 4. MULTIVARIABLE CONTROL OF A THREE TANK SYSTEM

The DTS200 three-tank system<sup>26</sup> is a non-linear system composed by three plexiglas tanks interconnected in series by two connecting pipes. A picture and a schematic diagram of this tank system are shown, respectively, in Figure 6 and 7. The liquid leaving T2 is collected in a reservoir from which pumps 1 and 2 supply the tanks T1 and T2. The three tanks are equipped with piezo-resistive pressure transducer for measuring the level of the liquid (usually distilled water).

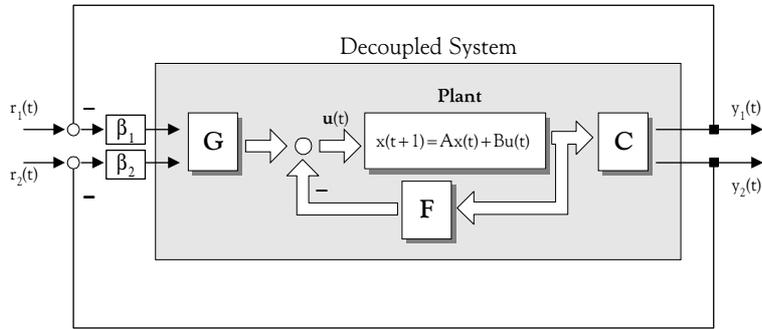


Figure 5 - Structure of the decoupling controller.

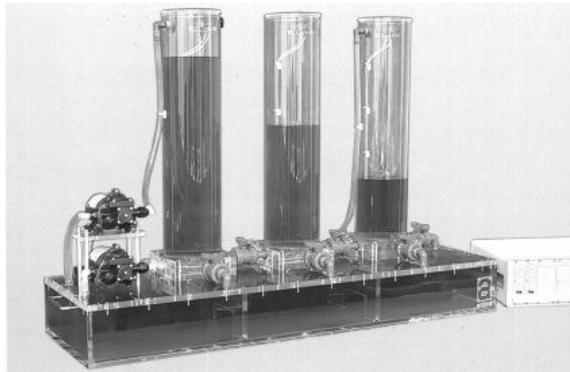


Figure 6 - A picture of the DTS200 (Courtesy of Amira).

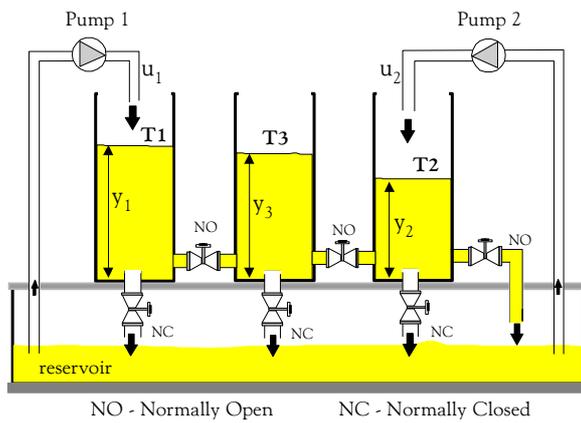


Figure 7 - Schematic diagram of the DTS200.

A digital controller manipulates the flow rate, such that the levels in the tanks T1 and T2 can be set independently. The connecting pipes and the tanks are additionally equipped with manually adjustable valves and outlets for the purpose of simulating clogs as well as leaks. There are three measurable state variables,  $x_1(t)$ ,  $x_2(t)$  and  $x_3(t)$  which, for control purposes, are assumed to be unmeasurable. The goal of the control system is to control the level  $y_1(t)$  and  $x_2(t)$  by adjusting the flow rate  $u_1(t)$  and  $u_2(t)$ .

4.1 - Control Specifications

For modelling purposes the three tank system process is assumed to be described as a non-linear input-output discrete time model, as in equation (19). The number of inputs is two ( $p=2$ ), equal to the number of outputs ( $q=2$ ). For control purposes, due to the observer technique used, the system is assumed to be described by equation (36) and (37). In this case a third order model was considered ( $n=3$ ).

For the identification task, using the Elman network, the following parameters were employed: learning rate  $\mu=0.01$  (equation 18), momentum  $\eta=0.6$  (equation 18), self-connection  $\alpha=0.6$  (equation 24), window size  $N=4$  (equation 28) and  $a=1.5$  (equation 5).

A priori knowledge about the process was assumed, in the form of a linear state space equation ( $W^A$ ,  $W^B$ ,  $W^C$ ), which was employed to initialize the weights of the RNN, as seen in equations (43)-(45). The sampling time used was 1.5 seconds and to prevent possible long training times, the maximum number of iterations in the learning task, for each sample, was limited to 20 iterations. The F and G matrices are computed with the goal that the pump flow  $u_1(t)$  affects only the level  $y_1(t)$  and the pump flow  $u_2(t)$  affects only the level tank  $y_2(t)$ . In the experiments, the parameters  $\beta_i$  (equation 54) were chosen to be equal for both subsystems  $\beta_i=0.3$ .

4.2 - Experimental Results

Two experiments were carried out in order to evaluate the performance of the proposed strategy with respect to set-point tracking. In the first experiment the states are assumed to be unmeasurable. The herein proposed observer, which allows simultaneous estimation of parameters and states using only input-output information, is implemented. The second is carried out to compare the performance of the first one with a situation where the state variables are accessible. The state variables are the tank levels  $x_1(t)$ ,  $x_2(t)$ , and  $x_3(t)$  with  $y_1(t)=x_1(t)$  and  $y_2(t)=x_2(t)$ . In this second experiment the observer only estimates the parameters values  $W^A$  and  $W^B$ . The output matrix  $W^C$  is fixed and is equal to

$$W^C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Also, in the training of the Elman network, the error is defined by  $e_k(t)=x_k(t)-x_k^h(t)$ , for  $k=1,2,3$  where  $x_k(t)$  is the actual state and  $x_k^h(t)$  is the estimated state (hidden state in the Elman network).

Experiment 1

Figure 8 shows the set-point/output (liquid levels), Figure 9 shows the actual states (assume to be unmeasurable) and the estimated states. Figure 10 shows the command inputs (pumps flow variations).

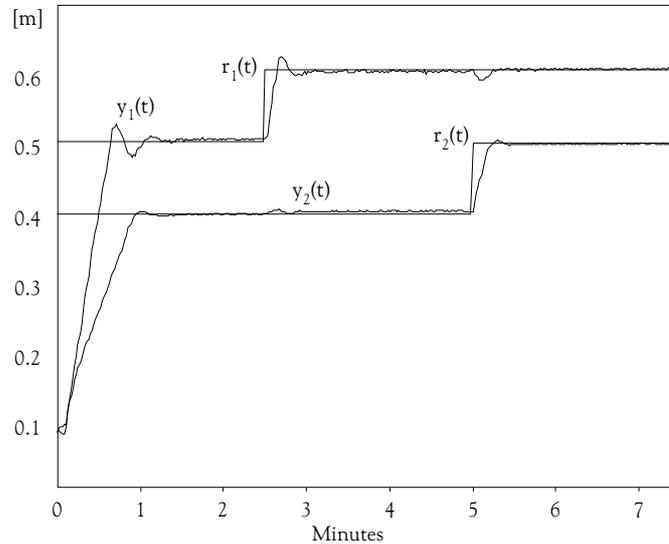


Figure 8 - Set-point/Output - unmeasurable state problem.

In Figure 8 it can be observed that a very acceptable control performance is obtained by combining on-line a nonlinear neural network estimation task (states and parameters) with a conventional pole placement controller.

As depicted in Figure 9 the estimated states do not converge to the actual ones. This may be understood by the fact that only input-output information is used and that no knowledge about the plant is assumed (usually, the design of an observer assumes the knowledge of the output matrix,  $C$ ). However, the controller based on this estimated states and parameters performs well since it is able to in maintain the outputs at the desired values.

#### Experiment 2

Figure 11 shows the set-point/output (liquid levels), Figure 12 depicts the actual states (assume to be accessible) and the estimated states. In Figure 13 can be seen the command inputs (pumps flow variations).

In Figure 12 the good behaviour of the identification of the Elman network is illustrated. As can be seen the neural model performs considerably well in tracking the actual states ( $x^h(t) \rightarrow x(t)$ ).

#### 4.3. Comments

Although for the first experiment the controller does not use knowledge of the state variables it performs reasonably well in comparison with the second controller, where the states are fully accessible. As can be seen for this particular experiment, whenever the system states are not

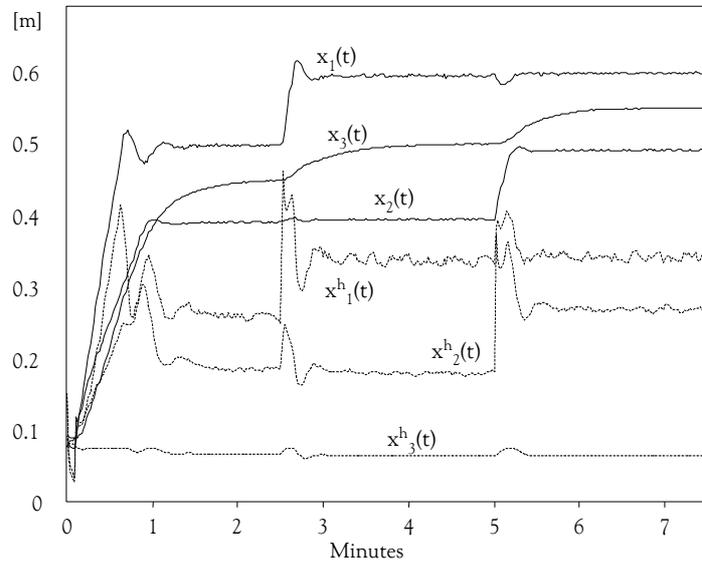


Figure 9 - Actual states/Estimated states - unmeasurable state problem.

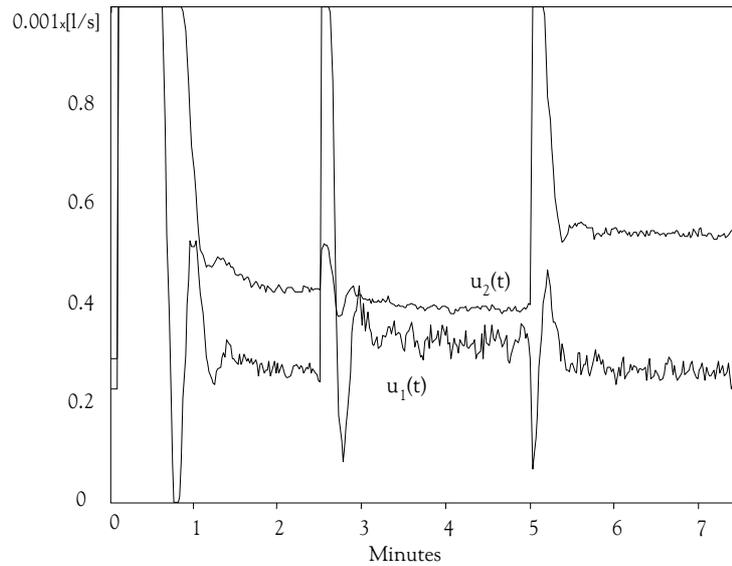


Figure 10 - Command inputs - unmeasurable state problem.

accessible the proposed technique seems to be an interesting approach. Thus the difficulties of the implementation of the conventional fully-state feedback system can be avoided using this neural control scheme.

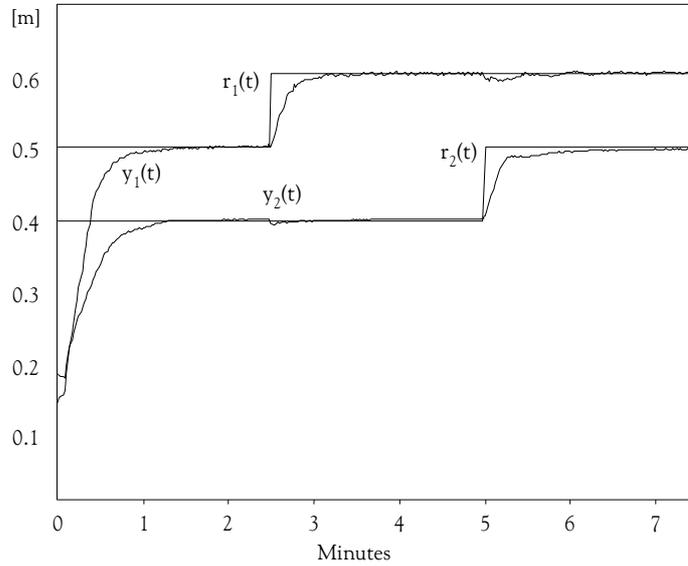


Figure 11 - Set-point/Output - Accessible state problem.

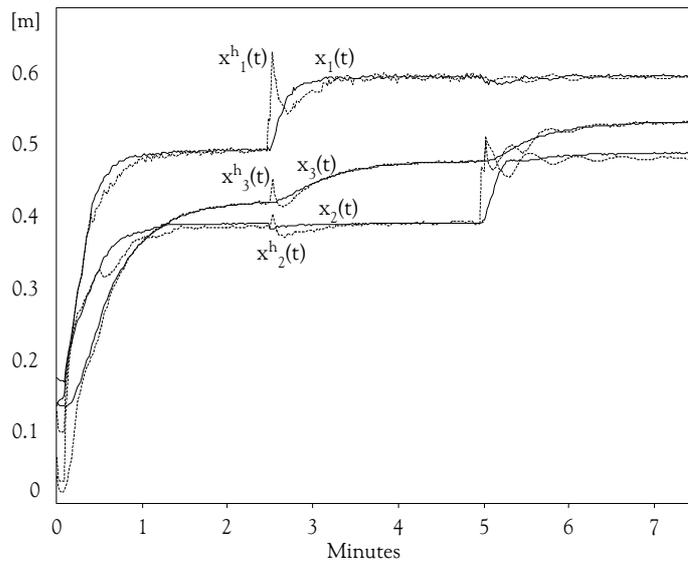


Figure 12 - Actual states/Estimated states - Accessible state problem.

Although a theoretical proof of the stability problem has not been addressed, a locally asymptotically stability for the closed loop system can be assured, even if there is error in the identified parameters. In fact the values of the closed loop poles are placed, at each sampling time,

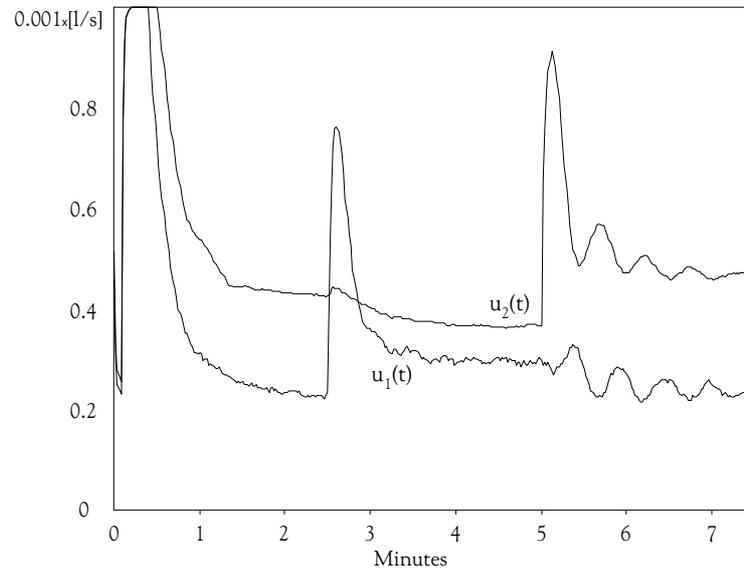


Figure 13 - Command inputs - Accessible state problem.

inside and away from the unity circle. This creates a stability margin such that possible errors in the identified parameters do not push the system outside the stability region.

### 5. CONCLUSIONS

An indirect adaptive control scheme for multivariable discrete non-linear systems, using recurrent networks, has been presented. This approach provides a bridge between the well known linear adaptive control methods and the field of neural networks.

Due to their properties, an Elman network was proposed, as the recurrent network responsible for the modelling task. This NN has some useful properties. Namely i) it can perform the identification of an arbitrary dynamic non-linear system, ii) it allows a fast on-line training with a relative small amount of memory and iii) it makes acceptable the use of the truncated backpropagation through time learning technique. Also, due to its particular architecture, iv) it can be interpreted as a non-linear state space model. Evaluating, at each sampling time, the derivatives of this non-linear neural model, a linear time variant state space model is obtained, which facilitates the use of familiar linear control techniques. In this work a state space controller, combining simultaneously a pole placement and a decoupling law, is employed.

The effectiveness of this strategy was validated on a multivariable laboratory process, the DTS200. Although the stability problem has not been addressed, the importance of rigorous analytical treatments of this problems are acknowledged. Further research is necessary on the stability of the closed loop, convergence and other issues related to feedback systems employing recurrent neural networks.

However, the results presented in this paper show that the combination of neural networks with linear control techniques can be an efficient tool to control non-linear systems. The design of complicated non-linear multivariable controllers (with possible unmeasurable states) and eventually the development of complex analytical models can be avoided by the application of strategies similar to the proposed one. It is believed that NN can be effectively used for control design of non-linear systems and they must be seen as an extension, rather than replacement, of linear identifiers and controllers that may be already working.

#### ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under program PRAXIS XXI. The authors also acknowledge helpful suggestions by the reviewers, which have been incorporated into the manuscript.

#### REFERENCES

1. Linkens, D. and H. Nyongesa - Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications - *IEE Proc. Control Theor. App.*, **143**, n° 4, 367-385, 1996.
2. Funahashi, K. and Y. Nakamura - Approximation of dynamical systems by continuous time recurrent networks - *Neural Networks*, **6**, 801-806, 1993.
3. Jin, L., P. Nikiforuk and M. Gupta - Approximation of discrete time state space trajectories using dynamic recurrent networks - *IEEE Trans. Automatic Control*, **40**, n° 7, 1266-1270, 1995.
4. Hunt K., D. Sbarbaro, R. Zbikowski and P. Gawthrop - Neural networks for control systems - a survey - *Automatica*, **28**, n° 6, 1083-1112, 1992.
5. Ribeiro, B. - On modeling, monitoring and control of MIMO non-linear systems with neural networks: An application to an industrial rotary lime kiln. - PhD Thesis (in Portuguese), 1995.
6. Chen, F-C - Back propagation neural networks for nonlinear self-tuning adaptive control - *IEEE Control System Magazine, Special Issue on Neural Networks for control* - 45-48, 1990.
7. Narendra K., and K. Parthasarathy - Identification and control of dynamical systems using neural networks - *IEEE Trans. Neural Networks*, **1**, 4-27, 1990.
8. Chen, F-C and H. Khalil - Adaptive control of a class of non-linear discrete-time systems using neural networks - *IEEE Trans. Automat. Control*, **40**, n° 7, 791-801, 1995.
9. Jin, L., P. Nikiforuk and M. Gupta - Adaptive control of discrete time non-linear system using recurrent neural networks - *IEE Proc. Control Theory Applications*, **141**, n° 3, 1994.
10. Lightbody, G, and G. Irwin - Direct neural model reference adaptive control - *IEE Proc. Control Theory Applications*, **142**, n° 1, 31-43, 1995.
11. Delgado, A. - Dynamic recurrent neural networks for system identification and control - *IEE Proc. Control Theory Applications*, **142**, n° 4, 307-314, 1995.
12. Sørensen, O. - Optimal control with neural networks - *Proceeding of IASTED - International Association of Science and Technology for Development*, Zurich, 90-95, 1991.
13. Kim, Y., F. Lewis and C. Abdallah - A Dynamic recurrent neural network based adaptive observer for a class of non-linear systems - *Automatica*, **33**, n° 8, pp 1539-1543, 1997.
14. Puskorius, G. and L. Feldkamp - Neuro-control of non-linear dynamic systems with Kalman filter trained recurrent networks - *IEEE Trans. Neural Networks*, **5**, n° 2, 279-297, 1994.
15. Suykens, J., J. Vandewallw and B. Moor - Artificial neural networks for modeling and control of non-linear systems - *Kluwer Academic Press*, 1996.
16. Parisini, T. and R. Zoppoli - Neural networks for non-linear state estimation - *Int. J. Robust Non-linear Control*, **4**, 231-148, 1994.
17. Ahmed, M. and I. Tasadduq - Neural net controller for non-linear plants: design approach through linearisation - *IEE Proc. Control Theory Applications*, **141**, n°5, 315-322, 1994.
18. Sørensen, O. - Non-linear pole placement control with a neural network - *EJC*, **2**, 36-43, 1996.
19. Narendra K. and K. Parthasarathy - Gradient methods for the optimization of dynamical systems containing neural networks - *IEEE Trans. on Neural Networks*, **2**, n° 2, 252-262, 1991.

20. Williams, R., D. Zipser - Gradient-based learning algorithms for recurrent networks and their computational complexity - *Backpropagation: Theory, architectures and applications*, Edit by Yves Chauvin and D. Rumelhart, Chap.13, 433-486, 1995.
21. Werbos, P. - Backpropagation through time: what it does and how do it - *Proc. IEEE*, **78**, 1550-1560, 1990.
22. Rumelhart, D. G. Hinton and R. Williams - Learning internal representations by error propagation - *Explorations in the microstructure of cognition. Vol 1: Foundations*, Cambridge, MIT Press/Bradford books, 1986.
23. Elman, J. - *Finding Structure in time* - University of California, San Diego, 1988.
24. Pham, D. and L. Xing - Dynamic system identification using Elman and Jordan networks - *Neural Networks for Chemical Engineers*, Edited by A. Bulsari, Chap. 23, 572-591, 1995.
25. Falb P. and W. Wolovich, - Decoupling in the Design and Synthesis of Multivariable Control Systems - *IEEE Trans. Automatic Control*, **12**, n° 6, 651-659, 1967.
26. Amira, Three Tank System DTS200: Laboratory Setup - *Amira GmbH*, 1996.



*Submitted to be published in  
Journal of Systems and Control Engineering  
January 1999*

## **Non-linear Multivariable Predictive Control: Neural *versus* First Principle Modelling Approach**

Jorge Henriques<sup>†\*</sup>, Paulo Gil<sup>†‡</sup>, António Dourado<sup>†</sup> and H. Duarte-Ramos<sup>‡</sup>

{jh,pgil,dourado}@dei.uc.pt hdr@mail.fct.unl.pt

<sup>†</sup>*CISUC - Centro de Informática e Sistemas da Universidade de Coimbra  
Informatics Engineering Department, UC  
Pólo II, Pinhal de Marrocos  
3030 Coimbra - Portugal  
Phone: +351 39 790000 - Fax: +351 39 701266*

<sup>‡</sup>*Electrical Engineering Department, UNL  
2825 Monte da Caparica - Portugal  
Phone: +351 1 2948545 - Fax: +351 1 2948532*

### **Abstract**

A hybrid control strategy integrating neural networks into the predictive control scheme and a first principle model predictive control technique for controlling purposes of non-linear plants are presented and compared.

The neural network approach involves a recurrent Elman network to capture the dynamics of the plant to be controlled being the learning stage implemented on-line using a modified backpropagation through time algorithm. The predictive control scheme is implemented by an iterative and repetitive optimisation process.

Within the first principle model predictive control framework a local linearisation of the mathematical model of the non-linear plant followed by a discrete-time approximation is performed at each sampling time. Next, a real-time open-loop linear constrained optimisation problem is solved with a standard quadratic programming algorithm.

Experimental results collected from an interconnected three-tanks system are presented and discussed for both control techniques.

**Keywords:** *Model predictive control; Elman networks; receding horizon; MIMO systems; optimisation.*

---

\* *Author to whom correspondence should be addressed*

## 1. Introduction

Model predictive control (MPC) has become over the recent years an increasingly accepted tool for controlling industrial processing plants. Although it was originally developed for power plant and petroleum refinery applications, it can in these days be found in a significant variety of industrial environments, such as, food processing, automotive, aerospace or pulp and paper, among others [1]. One major reason for this wide acceptance is undoubtedly related to its inherent ability to handle both multivariable input and output constraints while requiring that the resulting control be somehow optimal with respect to a specified cost function. The fundamental concepts concerning model predictive control can be traced back to 1963 and the work of Propoi where the moving horizon approach was for the first time proposed [2]. Yet, it was not until the late 1970s and the contribution of Richalet and co-authors [3], that, for the first time, has been reported and succeeded a model predictive implementation in the process control field, particularly for linear plants.

Recently, there have been some attempts to extend MPC techniques to non-linear systems. One such approach is based on a local linearisation, where successive linear approximations of the plant are used to predict future outputs [4]. Oliveira [5] in order to ensure the stability of the control system incorporates into the optimisation problem an extra contraction constraint, initially proposed by Polak and Yang [6], [7], leading to the so-called contractive MPC. Other possible way for achieving stability in the regulation problem considers a finite horizon terminal constraint that forces all states to be zero at the end of the prediction horizon [8]. Another alternative approach is proposed by Nevistić and Morari [9] where a combination of feedback linearisation and model predictive control is considered.

The success of the MPC technology is attributed to three relevant factors [1]. First and foremost is the incorporation of an explicit model into the control computation. This allows the controller to deal with the most significant features of the plant, depending on the accuracy of the mathematical model. Secondly, because it predicts future plant behaviour, the effects either of feedforward and feedback disturbances can be anticipated and hence adequately rejected. Finally, MPC methodologies have the potential of dealing with either input and output constraints in an explicit manner during the design and implementation stages. Nevertheless, owing to the complexity of the plant and/or lack of information about some of its parameters, deriving an appropriate and detailed first principle model may in many cases be an arduous task. This is the main drawback of this approach.

Given the learning capabilities of the neural networks (NN) [10], a potential way to overcome this issue may consist in the incorporation of NN as a non-linear black-box model. The learning features of NN, has led in recent years to the development of several neural based control strategies [11], [12], [13]. Examples are the direct inverse neuro control scheme [14], the specialised inverse neuro control strategy [15], the adaptive neuro control with backpropagation through time algorithm [16], [17], the internal model control [18] and the optimal neuro control [19].

Another viable control strategy involves NN within a predictive control framework. According to this approach several works have been reported referring feedforward neural networks (FNN) with external recurrence and off-line training. In this context, Donat *et al.* [20] successfully applied a multilayer FNN trained with a backpropagation algorithm together with an optimisation problem that is solved with a sequential quadratic programming method. Hao *et al.* [21] combine a FNN with the one-step-ahead predictive control scheme, being the present and future inputs obtained through a standard Quasi-Newton non-linear optimisation algorithm. Temeng *et al.* [22] propose a hybrid multivariable non-linear predictive control based on a FNN model and on the Fletcher variable metric method for solving the optimisation problem. Draeger *et al.* [23] included a FNN for non-linear prediction

in an extended standard dynamic matrix control algorithm. More recently, Chen [24] has proposed a three-layer FNN with hyperbolic tangent functions using the Levenberg-Marquardt algorithm for minimising a cost functional. Using also FNN, Tan and Cauwenberghe [25] presented a non-linear one-step-ahead control strategy. However, instead of solving a non-linear optimisation problem with a standard non-linear optimisation algorithm, the control action is obtained by a gradient descent algorithm, as in the learning stage.

Another approach is implementing recurrent neural networks (RNN) for modelling purposes in a predictive control structure. Zamarreño and Vega [26] proposed a neural non-linear adaptive control technique. A specific recurrent neural network is used for identification, being the control problem solved with a standard non-linear programming algorithm. Shaw and Doyle [27] presented a dynamic recurrent neural network to predict the outputs of a MIMO plant within the Internal Model Control framework.

In this paper a hybrid predictive control is compared with a standard MPC scheme. The first approach adopts the approximation capabilities of a recurrent Elman network for modelling purposes. The learning process is performed on-line with a modification of a backpropagation through time algorithm. Instead of one-step-ahead prediction proposed in [25], in the present work the sequence of control actions is computed extending the prediction outputs to a multi-step-ahead horizon.

Regarding the first principle model predictive control scheme, the non-linear real plant dynamics, at each time step, is first linearised and subjected to a time discretisation. Next, a standard constrained quadratic programming problem is solved to obtain the sequence of present and future manipulated variables. For both approaches only the first control action in the optimal control sequence is picked for implementation on the real plant.

The paper is organised as follows: in section 2 the MPC algorithm is revised briefly and both neural and receding horizon predictive control strategies are described; the recurrent Elman neural topology as well as the truncated backpropagation through time learning algorithm are presented. In section 3 the laboratory three-tanks system is characterised. Section 4 presents and discusses some experimental results carried out with both predictive techniques. Finally, the main conclusions are drawn in section 5.

## 2. Neural Network and Model Predictive Control

In the MPC technique, also known as receding horizon control (RHC), an explicit dynamic model of the plant is used to predict its outputs over some specified finite prediction horizon  $P$ , when the control actions are accordingly changed over some finite control horizon  $M$ .

The predictive control approach is a discrete time technique where, at time step  $k$ , an optimisation procedure computes on-line and in real-time the open-loop sequence of present and future control moves  $\{u(k|k), u(k+1|k), \dots, u(k+M-1|k)\}$ , such that the predicted outputs follow a predefined trajectory (Figure 1), [28] and taking into account constraints on the outputs and on the inputs. The first control action  $u(k|k)$  in the optimal control sequence is then picked for implementation on the real plant over the time interval  $[k, k+1]$ , being discarded the remaining sequence.

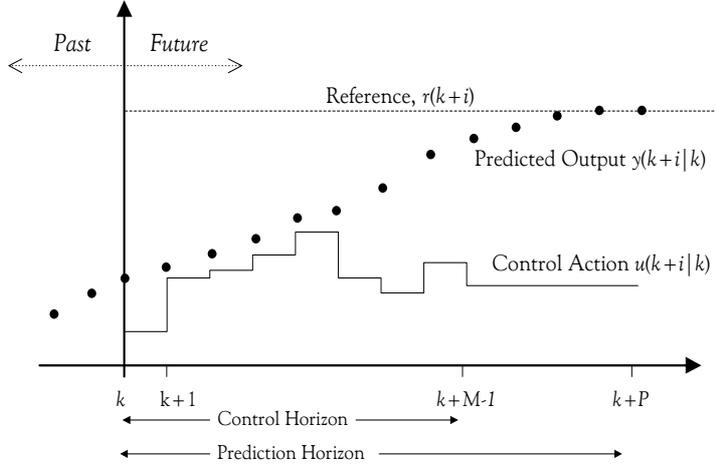


Figure 1 - RHC optimisation problem.

At the next sample time  $k+1$ , the prediction and control horizons are shifted ahead by one step and a new optimisation problem is solved using updated measurements from the process,  $y(k+1)$ , in conjunction with the control action  $u(k|k)$ . Thus, by repeatedly solve an open-loop optimisation problem with every initial conditions updated at each time step, the model predictive control strategy results in a closed-loop constrained optimal control technique.

The general open-loop optimisation problem can be formulated as

$$\min_{u(t)} V = \min_{u(t)} \int_{t_k}^{t_k+P} L(x(t), u(t), t) dt \quad (1)$$

subject to the system dynamics (2) and constraints (3).

$$\dot{x}(t) = f^P(x(t), u(t), t) \quad (2)$$

$$y(t) = g^P(x(t), u(t), t)$$

$$\begin{aligned} l(x(t), u(t)) &= 0 \\ c(x(t), u(t)) &\geq 0 \end{aligned} \quad (3)$$

$f^P : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^n$  and  $g^P : \mathfrak{R}^n \times \mathfrak{R}^m \rightarrow \mathfrak{R}^q$  are twice continuous differentiable;  $L$  is the Lagrangian;  $u(t) \in \mathfrak{R}^m$ ,  $y(t) \in \mathfrak{R}^q$  and  $x(t) \in \mathfrak{R}^n$  are, respectively, the inputs, outputs and states;  $c(\cdot)$  and  $l(\cdot)$  are functions characterising equality and inequality constraints;  $t_k$  is the discrete current time.

Depending on the cost function, the dynamic model and the constraints involved, several particular formulations can be stated in terms of the general formulation given above.

## 2.1 – Linear MPC Formulation

By linearising and discretising (2) a time-invariant discrete-time linear system to be controlled of the form (4) is obtained.

$$x(k+1) = \Phi x(k) + \Gamma u(k) + \eta \quad (4)$$

where  $\Phi \in \mathfrak{R}^{n \times n}$  and  $\Gamma \in \mathfrak{R}^{n \times m}$  are denoted the state and input matrices;  $x(k) \in \mathfrak{R}^n$  and  $u(k) \in \mathfrak{R}^m$  are the discrete state and control vectors;  $\eta \in \mathfrak{R}^n$  is the independent term vector;  $k$  is an integer sampling time index.

Assuming the plant as having all states measurable and considering a 2-norm as a performance index and linear constraints on the input and states, the open-loop optimal receding horizon control problem can be stated as follows

$$\min_{u(k), \mathbf{K}, u(k+M-1)} J = \min_{u(k), \mathbf{K}, u(k+M-1)} \left\{ \begin{array}{l} \sum_{i=1}^P \|x(k+i|k) - r(k+i)\|_{Q_i}^2 \\ + \sum_{i=0}^{M-1} \left[ \|u(k+i|k)\|_{R_i}^2 + \|\Delta u(k+i|k)\|_{S_i}^2 \right] \end{array} \right\} \quad (5)$$

subject to the system dynamics (4) and the following constraints:

$$\begin{aligned} x_{\min} &\leq x(k+i) \leq x_{\max}, \quad i = 1, \mathbf{K}, P \\ u_{\min} &\leq u(k+i) \leq u_{\max}, \quad i = 0, \mathbf{K}, M-1 \\ |\Delta u(k+i)| &\leq \Delta u_{\max}, \quad i = 0, \mathbf{K}, M-1 \\ |\Delta u(k+i)| &= 0, \quad i = M, \mathbf{K}, P-1 \end{aligned} \quad (6)$$

where  $\|\cdot\|_{\Omega}$  is the weighted Euclidean norm;  $Q_i \in \mathfrak{R}^{n \times n}$ ,  $R_i \in \mathfrak{R}^{m \times m}$  and  $S_i \in \mathfrak{R}^{m \times m}$  denote positive definite, symmetric weighting matrices;  $\Delta u \in \mathfrak{R}^m$  is the predicted control action increment;  $r(k) \in \mathfrak{R}^n$  denotes the discrete reference trajectory;  $i$  is the current discrete time index.

Since the optimal receding horizon formulation above is based on a quadratic cost function and assumes that all constraints are linear, then it may be cast into a quadratic programming problem (QP). In this form, the functional cost to be minimised is determined by a second-order Taylor series approximation for the objective functional  $J(\cdot)$ . For this type of optimisation problems, it is required to find a solution  $\Delta \tilde{u}^*$  to

$$\text{minimise } h^T \Delta \tilde{u} + \frac{1}{2} \Delta \tilde{u}^T H \Delta \tilde{u} \quad (7)$$

$$\text{subject to } A^T \Delta \tilde{u} \leq b \quad (8)$$

with  $A \in \mathfrak{R}^{mM \times (4mM + 2nP)}$ ,  $b \in \mathfrak{R}^{(4mM + 2nP)}$ . The gradient vector and the Hessian matrix of functional  $J(\cdot)$  are respectively  $h \in \mathfrak{R}^{mM}$  and  $H \in \mathfrak{R}^{mM \times mM}$ . The extended incremental input vector  $\Delta \tilde{u} \in \mathfrak{R}^{mM}$  is the vector of decision variables. If the Hessian matrix is positive

definite then the QP format cost function is strictly convex [29, Theorem 3.7] and consequently  $\Delta \tilde{u}^*$  is the unique global minimum for the optimisation problem [30, Theorem 2.1].

At each time step  $k$ , the gradient vector and the Hessian matrix of the specified performance index  $J(\cdot)$  are computed according to

$$h_l^T = 2 \left\{ x_o^T \left[ \sum_{i=l-1}^{P-1} (\Phi^{i+1})^T Q_{i+1} \sum_{q=0}^{i-l+1} \Phi^q \right] \Gamma + \left[ \Gamma u_o^T + \eta \right] \sum_{i=l-1}^{P-1} \left[ \sum_{q=0}^i (\Phi^q)^T Q_{i+1} \sum_{s=0}^{i-l+1} \Phi^s \right] \Gamma \right. \\ \left. - \left[ \sum_{i=l-1}^{P-1} (r(i+1))^T Q_{i+1} \sum_{q=0}^{i-l+1} \Phi^q \right] \Gamma + u_o^T \sum_{i=l-1}^{M-1} R_i \right\}, \quad l=1, K, M \quad (9)$$

$$H_{ll} = 2 \left\{ \Gamma^T \sum_{i=0}^{P-1} \left[ \sum_{q=0}^i (\Phi^q)^T Q_{i+1} \sum_{q=0}^i \Phi^q \right] \Gamma + \sum_{i=l-1}^{M-1} R_i + S_{l-1} \right\}, \quad l=1, K, M \quad (10)$$

$$H_{lp} = 2 \left\{ \Gamma^T \sum_{i=l-1}^{P-1} \left[ \sum_{q=0}^{i-l+1} (\Phi^q)^T Q_{i+1} \sum_{q=0}^{i-p+1} \Phi^q \right] \Gamma + \sum_{i=l-1}^{M-1} R_i \right\}, \quad l, p=1, K, M; \quad p \neq l$$

The constraint matrix  $A$  and constant vector  $b$  of equation (8), which embody the linear constraints on the inputs, on its increments and on the system states are given by

$$A^T = \begin{bmatrix} I_m & 0_m & \Lambda & 0_m & 0_m \\ 0_m & I_m & 0_m & \Lambda & 0_m \\ M & M & M & M & M \\ 0_m & 0_m & \Lambda & 0_m & I_m \\ -I_m & 0_m & \Lambda & 0_m & 0_m \\ 0_m & -I_m & 0_m & \Lambda & 0_m \\ M & M & M & M & M \\ 0_m & 0_m & \Lambda & 0_m & -I_m \\ I_m & 0_m & \Lambda & 0_m & 0_m \\ I_m & I_m & 0_m & \Lambda & 0_m \\ M & M & M & M & M \\ I_m & I_m & \Lambda & I_m & I_m \\ -I_m & 0_m & \Lambda & 0_m & 0_m \\ -I_m & -I_m & 0_m & \Lambda & 0_m \\ M & M & M & M & M \\ -I_m & -I_m & \Lambda & -I_m & -I_m \\ C_{11} & 0_{n \times m} & \Lambda & 0_{n \times m} & 0_{n \times m} \\ C_{21} & C_{22} & 0_{n \times m} & \Lambda & 0_{n \times m} \\ M & M & M & M & M \\ C_{M1} & C_{M2} & C_{M3} & \Lambda & C_{MM} \\ M & M & M & M & M \\ C_{P1} & C_{P2} & C_{P3} & \Lambda & C_{PM} \\ -C_{11} & 0_{n \times m} & \Lambda & 0_{n \times m} & 0_{n \times m} \\ -C_{21} & -C_{22} & 0_{n \times m} & \Lambda & 0_{n \times m} \\ M & M & M & M & M \\ -C_{M1} & -C_{M2} & -C_{M3} & \Lambda & -C_{MM} \\ M & M & M & M & M \\ -C_{P1} & -C_{P2} & -C_{P3} & \Lambda & -C_{PM} \end{bmatrix} \quad b = \begin{bmatrix} \Delta u_{UB}(0) \\ \Delta u_{UB}(1) \\ M \\ \Delta u_{UB}(M-1) \\ \Delta u_{UB}(0) \\ \Delta u_{UB}(1) \\ M \\ \Delta u_{UB}(M-1) \\ u_{UB}(0) - u_o \\ u_{UB}(1) - u_o \\ M \\ u_{UB}(M-1) - u_o \\ u_{LB}(0) + u_o \\ u_{LB}(1) + u_o \\ M \\ u_{LB}(M-1) + u_o \\ c_1^{x_{UB}} \\ c_2^{x_{UB}} \\ M \\ c_M^{x_{UB}} \\ M \\ c_P^{x_{UB}} \\ c_1^{x_{LB}} \\ c_2^{x_{LB}} \\ M \\ c_M^{x_{LB}} \\ M \\ c_P^{x_{LB}} \end{bmatrix} \quad (11)$$

with  $C_{ij}$  and  $c_i$  given by equations and (12) and (13), respectively;  $UB$  stands for “upper bound” and  $LB$  for “lower bound”;  $x_o$  is the state vector at time  $k$  and  $u_o$  the control action that has been applied to the plant at time  $k-1$ .

$$C_{ij} = \sum_{l=0}^{i-j-1} \Phi^l \Gamma, \quad \forall i=1, K, P, \forall j=0, K, M-1, j < i \quad (12)$$

$$c_i^{x_{UB}} = x_{UB} - \left[ \Phi^i x_o + \sum_{l=0}^{i-1} (\Phi^l \Gamma u_o + \eta) \right], \quad \forall i=1, K, P \quad (13)$$

$$c_i^{x_{LB}} = -x_{LB} + \left[ \Phi^i x_o + \sum_{l=0}^{i-1} (\Phi^l \Gamma u_o + \eta) \right], \quad \forall i=1, K, P$$

## 2.2. - Non-linear Neural Predictive Control

For modelling purposes the plant is assumed as multivariable described by a general non-linear discrete time state space model issued from the discretisation of (2).

$$x(k+1) = \phi_f \{x(k), u(k), k\} \quad (14)$$

$$y(k) = \phi_g \{x(k), u(k), k\} \quad (15)$$

where  $\phi_f : \mathfrak{R}^n \times \mathfrak{R}^m \times \mathfrak{R} \rightarrow \mathfrak{R}^n$  and  $\phi_g : \mathfrak{R}^n \times \mathfrak{R}^m \times \mathfrak{R} \rightarrow \mathfrak{R}^q$  are non-linear functions;  $u(k) \in \mathfrak{R}^m$ ,  $y(k) \in \mathfrak{R}^q$  and  $x(k) \in \mathfrak{R}^n$  are, respectively, the inputs, outputs and states at a discrete time  $k$ . As in the MPC formulation, the states are assumed to be directly observable.

### 2.2.1 - Elman Networks Topology

Recurrent neural networks comprising dynamic elements and feedback connections, are considered, since early, suitable for dealing with dynamical systems [31]. One of the useful characteristics of these topologies is that they can represent any order of delays implicitly rather than explicitly, as in feedforward networks with external recurrence. For recurrent networks it was proved that they may be used to approximate to any arbitrary precision a discrete time state space description, Jin *et al.* [31]. One of the simplest RNN is the Elman network.

Elman [32] proposed a partially recurrent network where the feedforward connections are modifiable and the recurrent connections are fixed. Additionally to the input and the output units the Elman network has a hidden unit,  $x^h(t) \in \mathfrak{R}^n$ , and a context unit,  $x^c(k) \in \mathfrak{R}^n$ , Figure 2. The interconnection matrices  $W^x \in \mathfrak{R}^{n \times n}$ ,  $W^u \in \mathfrak{R}^{n \times m}$  and  $W^y \in \mathfrak{R}^{q \times n}$  define, respectively, the interconnection weights for the context-hidden layer, for the input-hidden layer and for the hidden-output layer.

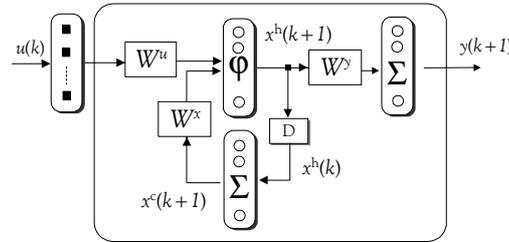


Figure 2 - Block diagram of an Elman network.

In the original architecture the context layer only holds a copy of the activation of the hidden units from the previous time step, being the trace of the entire history accumulated in the context unit. Due to practical difficulties related to the identification of higher order systems, some modifications have been proposed. In [33] a self-connection or a feedback gain  $\alpha \in \mathfrak{R}^+$  in the context units is incorporated, as depicted in Figure 3, improving the dynamic memorisation ability of the network.

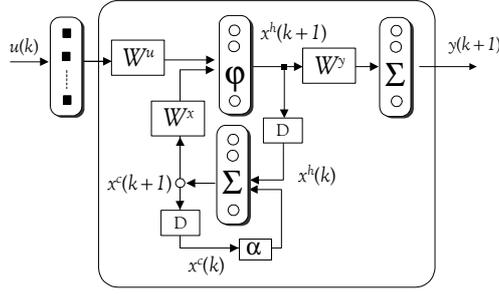


Figure 3 – Block diagram of the modified Elman network.

The dynamics of the modified Elman neural network is described by the difference equations (16)-(19).

$$s(k+1) = W^x x^c(k+1) + W^u u(k) \quad (16)$$

$$x^h(k+1) = \varphi \{ s(k+1) \} \quad (17)$$

$$x^c(k+1) = x^h(k) + \alpha x^c(k) \quad (18)$$

$$y(k+1) = W^y x^h(k+1) \quad (19)$$

The function  $f(x) = x$  is used for the output layer,  $s(k) \in \mathfrak{R}^n$  is an intermediate variable and  $\varphi(\cdot)$  is an hyperbolic tangent function given by

$$\varphi(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}, \quad a \in \mathfrak{R}^+ \quad (20)$$

If an augmented state,  $\bar{x}(k) \in \mathfrak{R}^{2n}$ , is defined by

$$\bar{x}(k) = \begin{bmatrix} x^h(k) \\ x^c(k) \end{bmatrix} \quad (21)$$

equations (16)-(19) can be rewritten as

$$\bar{x}(k+1) = \varphi \{ \bar{x}(k), u(k), W^x, W^u \} \quad (22)$$

$$y(k+1) = g \{ \bar{x}(k+1), W^y \} \quad (23)$$

These equations can be interpreted as a state space model, analogous to the non-linear system dynamics defined in (14) and (15).

### 2.2.2 - Learning Methodology

The network depicted in Figure 3 is used to approximate the behaviour of a general non-linear plant. Assuming that  $x_d(k) \in \mathfrak{R}^n$  denotes the plant state at time step  $k$ , the goal is to find  $W^x$  and  $W^u$  ( $W^y$  is known and fixed) such that the squared error between the output neurons and the desired states in the horizon  $[k-N, K, k]$ , defined by the criterion (24), is minimised.

$$E(k) = \sum_{i=k-N}^k e_m(i)^2 \quad (24)$$

where  $e_m(k) \in \mathfrak{R}^n$  is the modelling error at time step  $k$  given by (25).

$$e_m(k) = x_d(k) - x^h(k) \quad (25)$$

Several training algorithms have been proposed to adjust the weighting values in recurrent networks. Examples of these methods are the Narendra's dynamic backpropagation [34], the real time recurrent algorithm from Williams and Zipser [35] and the backpropagation through time algorithm (BTT) from Werbos [36], which is being considered in the present work. All these methods use a gradient based learning algorithm and involve the computation of partial derivatives or sensitivity functions. To updated  $W^x$  and  $W^u$  equation (26) is used.

$$\Delta W^i(k+1) = \rho_m \Delta W^i(k) - \mu_m (1 - \rho_m) \frac{\partial E(k)}{\partial W^i(k)}, \quad i = x, u \quad (26)$$

$\mu_m \in \mathfrak{R}^+$  is a learning rate;  $\rho_m \in \mathfrak{R}^+$  is a momentum term; the increment  $\Delta W$  is evaluated according to

$$\Delta W^i(k+1) = W^i(k+1) - W^i(k), \quad i = x, u \quad (27)$$

For evaluating  $W^x$  and  $W^u$  a simplification of the  $BTT(\infty)$  algorithm is considered in this work by truncating the infinity backpropagation of information to a finite number ( $N$ ) of prior time steps [37]. Thus using equation (18) and considering  $N$  prior time steps, the context unit at time step  $k$  can be described by

$$x^c(k) = \alpha^N x^c(k-N) + \sum_{i=1}^N \alpha^{i-1} x^h(k-i) \quad (28)$$

A value of  $\alpha$  near to 1 enables the context unit to remember more past information and a value near to 0 let the context unit to forget rapidly past data, similarly to a forgetting factor. For  $0 < \alpha < 1$ , the dependence of past information decays exponentially backward in time, making the use of the truncated  $BTT(N)$  an acceptable approximation of the true  $BTT(\infty)$ , in case of values  $\alpha$  and  $N$  are correctly chosen, being verified (29).

$$\alpha^N \rightarrow 0 \quad (29)$$

The BTT algorithm is based on an extension of the standard backpropagation for feedforward networks. A recurrent network is expanded into a multilayer feedforward network, being a new layer added at each time step. The computation of the derivatives is then performed as in a standard feedforward backpropagation network [38]. Figure 4 illustrates an unfolded Elman network consisting of  $N+1$  layers, corresponding to a recurrent network that has been running from time  $k-N$  to  $k$ .

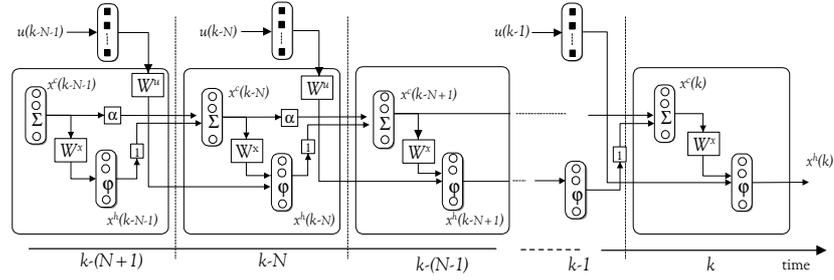


Figure 4 - Schematic diagram of an unfolded Elman recurrent network.

The computation of the gradient in equation (26), is accomplished with (30) and (31)

$$\frac{\partial E(k)}{\partial W^x} = \sum_{i=k-N}^k \delta^h(i) x^c(i)^T \quad (30)$$

$$\frac{\partial E(k)}{\partial W^u} = \sum_{i=k-N}^k \delta^h(i) u(i-1)^T \quad (31)$$

where the sensitivity to the units of the net inputs  $\delta^h(k) \in \mathfrak{R}^+$  is computed recursively for  $i = k$  to  $i = k - N$  according to

$$\varepsilon^h(i) = e_m(i) + \delta^c(i+1) \quad (32)$$

$$\delta^h(i) = \varepsilon^h(i) \otimes \varphi'\{s(i)\} \quad (33)$$

$$\varepsilon^c(i) = \alpha \delta^c(i+1) + W^{xT} \delta^h(i) \quad (34)$$

$$\delta^c(i) = \varepsilon^c(i) \quad (35)$$

The process starts at time  $k$  with

$$\varepsilon^h(k) = e_m(k) = x_d(k) - x^h(k) \quad (36)$$

$$\delta^h(k) = \varepsilon^h(k) \otimes \varphi'\{s(k)\} \quad (37)$$

$$\varepsilon^c(k) = W^{xT} \delta^h(k) \quad (38)$$

The symbol  $\otimes$  denotes an element to element multiplication and  $\varphi'(\cdot)$  is the derivative of the hyperbolic tangent function  $\varphi(\cdot)$ , given by

$$\varphi'(x) = \frac{4a}{(e^{ax} + e^{-ax})^2} \quad (39)$$

### 2.2.3 - Neural Network-based Predictive Control

The general principle underlying neural predictive control is the same as that of any standard MPC technique, that is, the existence of an explicit model of the plant and the computation of the manipulated variables by solving an optimisation problem.

The control action sequence  $\{u(k|k), K, u(k+M-1|k)\}$  is evaluated such that the predicted time response has certain desirable characteristics according to a pre-defined design criterion, as equation (5), with  $R_i = 0$ ,  $Q_i = I_m$  and  $S_i$  a constant weighting matrix.

The minimisation of the cost function  $J(\cdot)$  is performed repeatedly on-line at each sampling time, being the first input  $u(k|k)$  in the computed optimal sequence injected into the real plant, as in the standard MPC case. Since  $J(\cdot)$  is non-linearly dependent on the control action sequence, the minimisation of the criterion proceeds iteratively, as in the neural learning stage. For the  $i^{\text{th}}$  iteration the gradient descent algorithm is evaluated at each instant  $k$ , according to

$$u^{i+1}(k|k) = u^i(k|k) - \mu_p \frac{\partial J(\cdot)}{\partial u^i(k|k)} \quad (40)$$

where  $\mu_p \in \mathfrak{R}^+$  is the optimisation step. The derivative of the cost function (40) with respect to the present and future inputs  $\{u(k|k), K, u(k+M-1|k)\}$ , is given by

$$\frac{\partial J(\cdot)}{\partial u(k+j|k)} = 2 \left\{ \begin{array}{l} \sum_{i=1}^P e_p(k+i|k) \frac{\partial x(k+i|k)}{\partial u(k+j|k)} \\ -S [u(k+j-1|k) - 2u(k+j|k) + u(k+j+1|k)] \end{array} \right\} \quad (41)$$

where  $j = 0, K, M-1$ ;  $i = 1, K, P$ ;  $e_p(k+i|k) \in \mathfrak{R}^n$  is the prediction error at instant  $k+i$ , given by

$$e_p(k+i|k) = x(k+i|k) - r(k+i|k) \quad (42)$$

The derivative of the network outputs relative to the input sequence, in (41) is given by:

$$\frac{\partial x(k+i|k)}{\partial u(k+j|k)} = \begin{cases} 0 & \text{if } i \leq j \\ \sum_{l=1}^{i-1} \frac{\partial x(k+i|k)}{\partial x(k+l|k)} \frac{\partial x(k+l|k)}{\partial u(k+j|k)} + \frac{\partial x(k+i|k)}{\partial u(k+j|k)} & \text{if } i > j \end{cases} \quad (43)$$

With the current sequence of manipulated variables provided by the optimiser, the model evaluates the trajectory of the predicted outputs over the prediction horizon. These predictions are fed into the optimiser again where the objective function is evaluated and a new sequence of manipulated variables is calculated. This iterative task continues until the convergence is reached. Then, the control action  $u(k|k)$  is implemented on the real plant and the procedure is repeated at the next time step.

The optimisation problem is solved in each iteration with a gradient descent algorithm similar to the learning phase (section 2.2). Nevertheless, instead of computing the weights

$W^x$  and  $W^u$  while  $u(k|k)$  is kept fixed (26),  $u(k|k)$  is updated being  $W^x$  and  $W^u$  assumed to be constant (40).

### 3. The Laboratory Three-Tank System

The three-tank system used for experiments [39] comprises three plexiglas cylinders, being one of the tanks ( $T_3$ ) connected to the other two tanks by the means of circular cross section pipes equipped with manually adjustable ball valves, Figure 5. At the tank  $T_2$  is located the main outlet of the plant, which is connected to the collecting reservoir in a similar way by a circular cross section pipe and a outflow ball valve. Additionally, each tank is provided with a direct connection to the reservoir in order to enable the simulation of leaks by adjusting manually the respective ball valve.

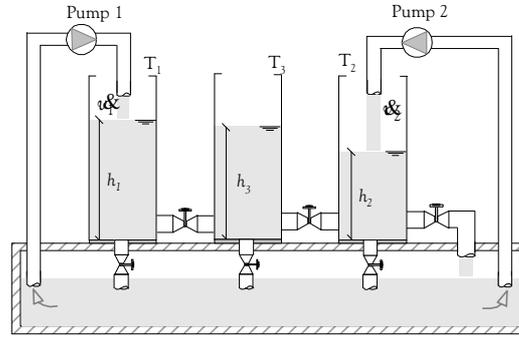


Figure 5 - Structure of the laboratory Three-Tank system.

Two of the tanks, namely,  $T_1$  and  $T_2$ , are fed with liquid, usually distilled water, which is pumped from the reservoir by two diaphragm pumps with fixed piston stroke and driven by a DC motor. In order to measure the current liquid level in each tank, the plant is equipped with piezoresistive differential pressure transducers.

For deriving the dynamics of the laboratory plant the mass flow rate balance equation is applied to all of the three tanks. When no leaks are considered, the mathematical model can be described by the following three non-linear equations:

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{A_T} \left[ \mathfrak{Q}_1 - \zeta_{13} S_t \operatorname{sgn}(h_1 - h_3) \sqrt{2g|h_1 - h_3|} \right] \\ \frac{dh_2}{dt} &= \frac{1}{A_T} \left[ \mathfrak{Q}_2 + \zeta_{32} S_t \operatorname{sgn}(h_3 - h_2) \sqrt{2g|h_3 - h_2|} - \zeta_{20} S_t \sqrt{2gh_2} \right] \\ \frac{dh_3}{dt} &= \frac{1}{A_T} \left[ \zeta_{13} S_t \operatorname{sgn}(h_1 - h_3) \sqrt{2g|h_1 - h_3|} - \zeta_{32} S_t \operatorname{sgn}(h_3 - h_2) \sqrt{2g|h_3 - h_2|} \right] \end{aligned} \quad (44)$$

where  $h_i, i=1,2,3$ , is the tank level;  $\mathfrak{Q}_i, i=1,2$ , denotes the flow rate;  $A_T$  and  $S_T$  are the cross section of each tank and the interconnecting pipes cross section, respectively;  $\zeta_{ij} \in [0, 1]$  is a dimensionless flow coefficient, with index  $j=0$  identifying the collecting reservoir;  $g$  is the gravity;  $\operatorname{sgn}$  stands for the sign of a given argument.

## 4. Experimental Results

In order to assess and compare the performance of both predictive controllers a set of experiments was carried out with the laboratory plant. The control algorithms were written in C code and running in a Pentium PC.

With respect to the neural strategy, an Elman network with two inputs ( $m = 2$ ) and three outputs ( $n = 3$ ) was implemented. The number of hidden units and context units is the same of the number of the plant states,  $n = 3$ .

For the identification task, the following parameters are chosen as: learning rate  $\mu_m = 0.02$  (26), momentum  $\rho_m = 0.4$  (26), self-connection  $\alpha = 0.6$  (18), window size  $N = 4$  (24) and  $\alpha = 0.4$  (20). Concerning the neural predictive scheme, the prediction horizon is  $P = 3$ , the control horizon  $M = 1$  and the optimisation step is  $\mu_p = 0.1$  (40).

The weighting matrices,  $Q_i$  and  $S_i$ , in the criterion (5), are chosen as:

$$Q_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad S_i = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix} \quad (45)$$

For implementing the first model-based predictive controller the non-linear continuous-time plant dynamics described by equation (44) is first subjected to a local linearisation based on a Taylor series expansion and neglecting the second and higher order terms. Following this linearisation, a discrete-time approximation is performed assuming a sampling time equal to 0.5 second invariantly, the same for both controllers, and considering that the system inputs are proceeded through a zero-order hold [40]. These steps are carried out sequential and repeatedly at each sampling time. The weighting matrices  $Q_i$ ,  $R_i$  and  $S_i$  in the objective functional (5) are chosen as

$$Q_i = \begin{cases} \begin{bmatrix} 10 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \Leftarrow i < P \\ \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \Leftarrow i = P \end{cases} \quad R = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \quad S = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (46)$$

The open-loop constrained optimal control problem, which is formulated as a Quadratic Programming problem (7) and (8) is solved using a MATLAB function (qp) [**Error! Reference source not found.**], and choosing for prediction horizon  $P = 5$  and control horizon  $M = 1$ . Since each pump flow rate and liquid level for each of the three tanks must be not only positive but also upper bounded, inequality constraints should be imposed in solving the optimal control problem. Thus, given the physical limitations of the real plant and the particular saturation characteristics of the actuators, the system states and the control actions are subjected to the following constraints:

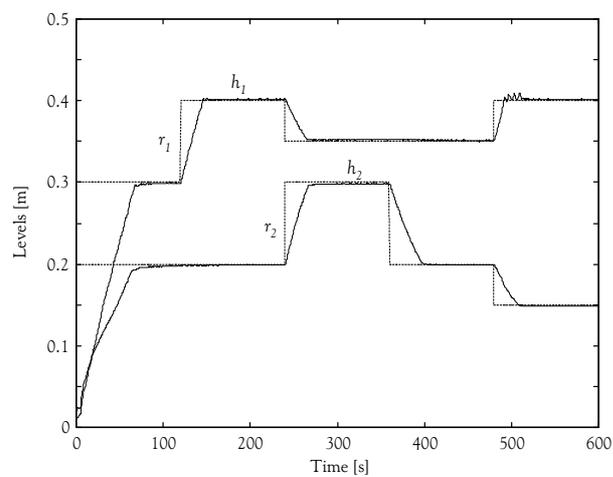
$$[0 \ 0 \ 0]^T \leq x \leq [0.6 \ 0.6 \ 0.6]^T \ m$$

$$[0 \ 0]^T \leq u \leq [97.50 \ 102.67]^T \times 10^{-6} \ m^3 s^{-1}$$

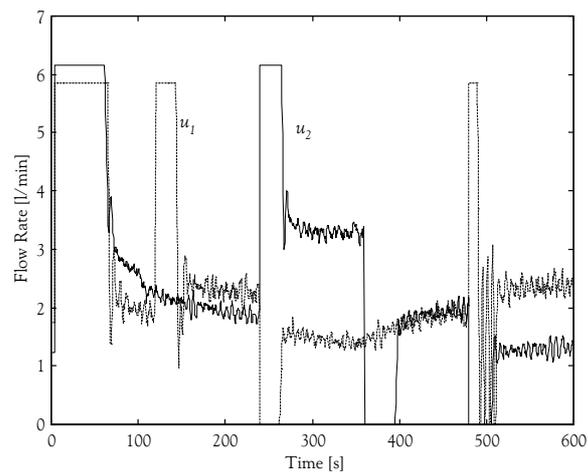
In order to prevent from excessive input changes, increments on the control actions are not merely penalised by the weighting matrix  $S$  in (5) but also constrained to

$$|\Delta u| \leq [16 \ 16]^T \times 10^{-6} \text{ m}^3 \text{ s}^{-1}.$$

In Figure 6 is depicted the set-point trajectory, the plant outputs and the control actions when a neural predictive strategy is implemented and tuned according to the parameters presented above. Previous knowledge of the plant dynamics as a linear state space equation was used to initialise the network weights. Figure 7 shows analogous results for the first principle model predictive controller implemented according to the tuning parameters above.

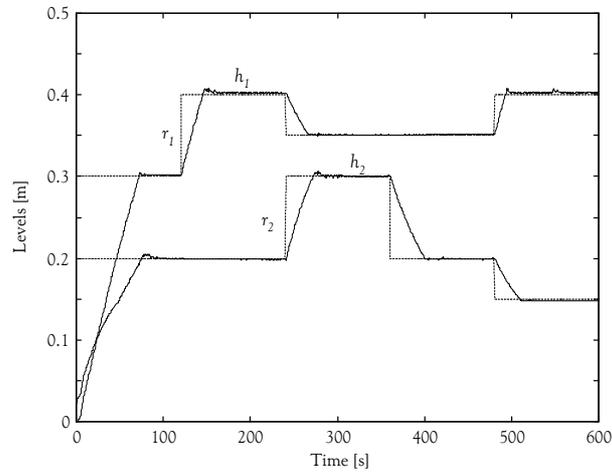


(a)

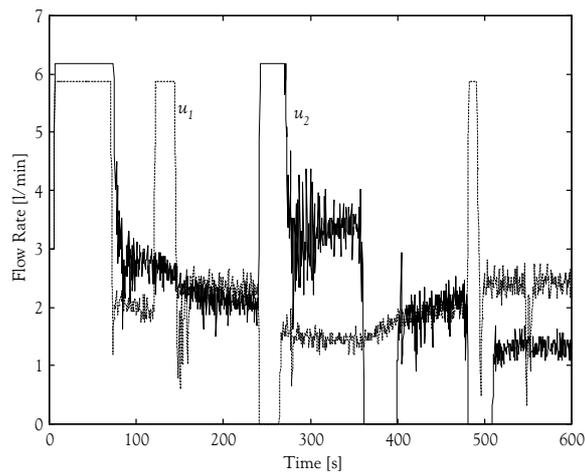


(b)

Figure 6 – Neural predictive control; (a) set-point trajectory and outputs; (b) control actions.



(a)

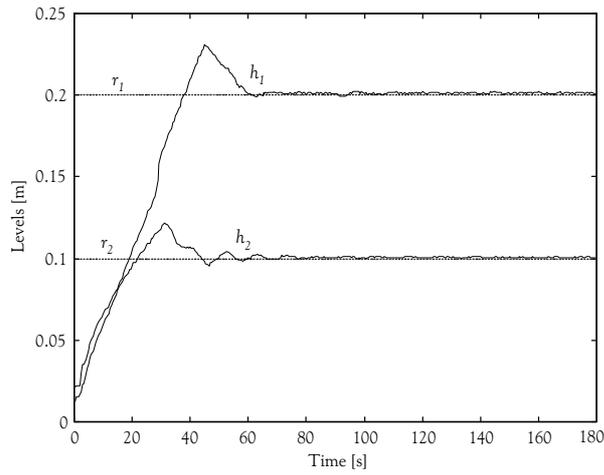


(b)

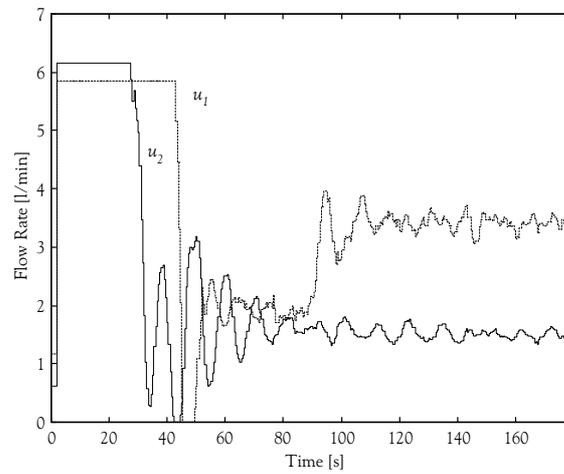
Figure 7 – First principle model predictive control; (a) set-point trajectory and outputs; (b) control actions.

As can be seen from Figure 7 and Figure 8, both controllers are able to follow adequately the set-point trajectory, regardless the set-point for each tank and the difference in level. However, with the first principle predictive technique the control error is closer to zero. In addition, the transient response exhibited by this control system is slightly faster than the neural predictive control system. These features are reflected in the corresponding control actions, where is observed a higher magnitude for the input changes. Nevertheless, this somehow nervous behaviour could be prevented by an adequate modification of the weighting matrices, particularly  $S$ .

In order to assess the robustness of each controller, one disturbance at instant 90 second was injected into the tank  $T_1$  by opening partially the valve connecting directly this tank to the reservoir. Additionally, no previous knowledge was incorporated into the neural predictive controller, being the initialisation implemented randomly (weighting matrices  $\in [0,1]$ ). The control system response for the disturbed plant is shown in Figure 8 and Figure 9, respectively, for the neural and first principle controllers.

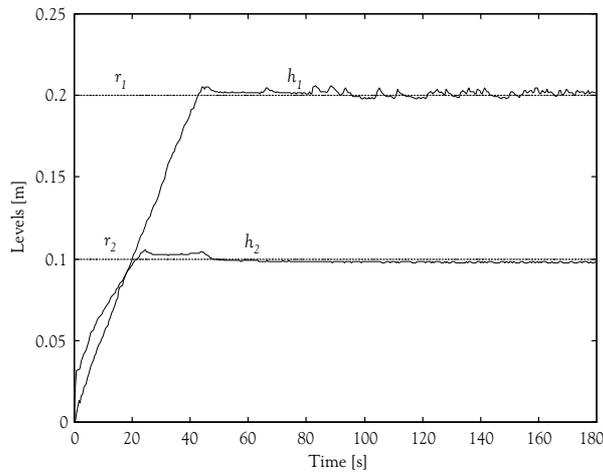


(a)

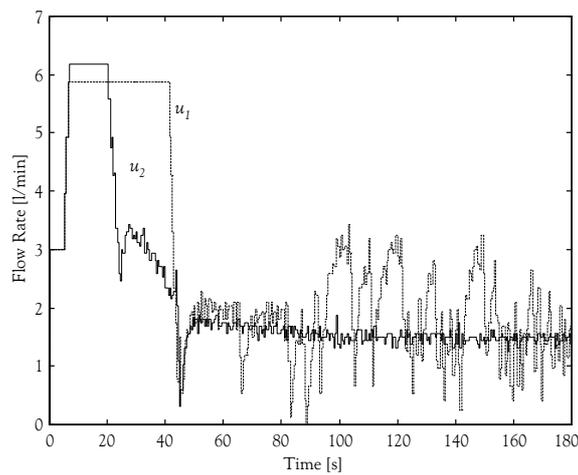


(b)

Figure 8 – Neural predictive control; (a) set-point trajectory and outputs; (b) control actions.



(a)



(b)

Figure 9 – First principle model predictive control; (a) set-point trajectory and outputs; (b) control actions.

With respect to the neural predictive controller, since no previous knowledge is provided there exists an initial period (approximately 60 second) where the network is gathering updated data from the plant and gradually improving the knowledge about its dynamics. After this initial period, the deviation from the set-point is quite tiny even after the injection of the disturbance. For the first principle model predictive controller the disturbance is also quite well rejected despite an higher magnitude of the input increments in the corresponding actuator (pump 1). This somehow frenetic behaviour exhibited by pump 1 is not so explicitly observed for the neural controller due to the learning abilities and adaptive characteristics of this strategy.

## 5. Conclusions

In this work two finite horizon predictive techniques have been implemented and applied for controlling a laboratory plant. One of the predictive strategies is based on an Elman network model for capturing the system dynamics. The other scheme incorporates explicitly a mathematical physical model and takes into account constraints on the inputs and the outputs. For solving the optimisation problem the neural technique uses a gradient descent method while the first principle model predictive control involves a quadratic programming algorithm.

The results obtained show that both approaches are quite viable even in the presence of an additional disturbance. However, for the operating conditions and the tuning parameters considered, the neural approach leads to a smoother functioning of the actuators, but with a slightly larger deviation, particularly when no disturbances are injected.

When the plant to be controlled is likely changing its parameters or having non-linearities or even unmodelled dynamics the on-line learning with neural network within a predictive control technique demonstrates to be not only quite effective but also preferable, given the adaptive features of this control scheme.

### Acknowledgements

This work was partially supported by the Portuguese Ministry of Science and Technology (MCT), under program PRAXIS XXI.

### References

- 1 **Qin, S. and Badgwell, T.** - An Overview of industrial model predictive control applications - Presented at *Non-linear MPC workshop*, Ascona, Switzerland, June 2-6, 1998.
- 2 **Garcia, C., Prett, D. and Morari, M.** - Model Predictive Control: Theory and Practice - a Survey. *Automatica*, vol 25, n<sup>o</sup>3, 335-348, 1989.
- 3 **Richalet, J., Rault, A., Testud, J. and Papon, J.** - Model predictive heuristic control: applications to an industrial process - *Automatica*, 14, 413-428, 1978.
- 4 **Garcia, C.** - Quadratic dynamic matrix control of non-linear process: an application to a batch reactor process - *AIChE Annual meeting*, San Francisco, 1984.
- 5 **Oliveira, S.** - Model predictive control (MPC) for constrained non-linear systems - PhD Thesis, CIT, Pasadena, CA, 1996.
- 6 **Polak, E. and Yang, T.** - Moving horizon control of linear systems with input saturation and plant uncertainty - Part 1 - *Int. J. Control*, 58 (3), 613-638, 1993.
- 7 **Polak, E. and Yang, T.** - Moving horizon control of linear systems with input saturation and plant uncertainty - Part 2 - *Int. J. Control*, 58 (3), 639-663, 1993.
- 8 **Kwon, W. and Pearson, A.** - A modified quadratic cost problem and feedback stabilisation of a linear system - *IEEE Trans. Automatic Control*, 22 (5), 838-842, 1977.
- 9 **Nevistić, V., and Morari, M.** - Constrained control of feedback-linearizable systems - *Proceedings of ECC95*, Rome, 1726-1731, 1995.
- 10 **Hornik, K., Stinchcombe, M. and White, H.** - Multilayer feedforward networks are universal approximators - *Neural Networks*, 2, 359-366, 1989.
- 11 **Hunt, K., Sbarbaro, D., Zbikowski, R. and Gawthrop, G.** - Neural networks for control systems: a survey *Automatica*, 28, n<sup>o</sup> 6, 1083-1112, 1992.
- 12 **Ribeiro, B.** - On modelling, monitoring and control of MIMO non-linear systems with neural networks: An application to an industrial rotary lime kiln - *PhD Thesis, Informatics Dep. Eng.* - University of Coimbra, 1995.
- 13 **Ronco, E. and Gawthrop, P.** - Neural networks for modelling and control - *Technical report csc97008 Centre for system and control, Department of mechanical engineering, University of Glasgow*, November 1997.
- 14 **Sanner, M. and Akin, D.** - Neuromorphic pitch attitude regulation of an underwater telerobot - *IEEE Control System Magazine*, 10, 62-67, 1990.

- 15 **Psaltis, D., Sideris, A. and Yamamura, A.** - A multilayered neural network controller - *IEEE Control System Magazine*, 8, 17-21, 1988.
- 16 **Narendra, K. and Parthasarathy, K.** - Identification and control of dynamic systems using neural networks *IEEE Trans. on Neural Networks*, 1, 4-27, 1990.
- 17 **Nguyen, D. and Widrow, B.** - Neural networks for self-learning control systems - *IEEE Control Systems Magazine*, April, 18-23, 1990.
- 18 **Hunt, K. and Sbarbaro, D.** - Neural networks for non-linear internal model control - *IEE Proc D*, 138, 431-438, 1991.
- 19 **Sørensen, O.** - Optimal control with neural networks - *Proceedings of the IASTED, International Conference on Artificial Intelligence Applications and Neural Networks, AINN91*, Zurich, July 1-3, vol 1, 90-95, 1991.
- 20 **Donat, S., Bhat, N. and McAvoy, T.** - Neural net based model predictive control - *Int. J. Control*, 54, 1453-1468, 1991.
- 21 **Hao, J., Tan, S. and J. Vandewalle** - One step ahead predictive control of non-linear systems by neural networks - *Proceedings of International Joint Conference on Neural Networks* - 2761-2764, 1993.
- 22 **Temeng, H., Schenelle, P. and McAvoy, T.** - Model predictive control of an industrial packed bed reactor using neural networks - *J. Proc. Cont.*, vol 5, n°1, 19-28, 1995.
- 23 **Draeger, A., Engell, S. and Ranke H.** - Model Predictive control using neural networks - *IEEE Control Systems*, 61-66, October 1995.
- 24 **Chen, J.** - Systematic derivations of model predictive control based on artificial neural networks - *Chemical Eng. Communications*, vol 164, 35-39, 1998.
- 25 **Tan, Y. and Cauwenberghe, A.** - Non-linear one step ahead control using neural networks: control strategy and stability design - *Automatica*, vol 32, n° 12, 1701-1706, 1996.
- 26 **Zamarreño, J. and Vega, P.** - Identification and predictive control of a melter unit from the sugar industry using recurrent neural nets - *Proceedings of Eann96, Engineering Applications of Neural Networks*, 119-126, 1996.
- 27 **Shaw, A. and Doyle, F.** - Multivariable non-linear control applications for a high purity distillation column using a recurrent dynamic neuron model - *Journal of Process Control*, Vol 7, n°4, 255-268, 1997.
- 28 **Morari, M.** - Advances in Model Predictive Control Conference - *D. Clarke ed., Oxford University Press*, 22- 27, 1994.
- 29 **Arora, J.** - Introduction to Optimum Design – *McGraw-Hill Book Co.*, 1989.
- 30 **Nash, S. and Sofer, A.** - Linear and Non-linear Programming - *McGraw-Hill, Industrial Engineering Series*, 1996.
- 31 **Jin, L., Nikiforuk, P. and Gupta, M.** - Approximation of discrete time state space trajectories using dynamic recurrent networks - *IEEE Trans. Automatic Control*, 40, n° 7, 1266-1270, 1995.
- 32 **Elman, J.** - Finding Structure in time - *Cognitive Science*, 14, 1789-211, 1990.
- 33 **Pham D. and Xing, L.** - Dynamic System Identification Using Elman and Jordan Networks - *Neural Networks for Chemical Engineers*, Editor A. Bulsari, Chap. 23, 1995.
- 34 **Narendra, K. and Parthasarathy, K.** - Gradient methods for the optimization of dynamical systems containing neural networks - *IEEE Trans. on Neural Networks*, 2, n° 2, 252-262, 1991.
- 35 **Williams, R. and Zipser, D.** - Gradient-based learning algorithms for recurrent networks and their computational complexity - *Backpropagation: Theory, architectures and applications*, Edit by Yves Chauvin and D. Rumelhart, Chap.13, 433-486, 1995.
- 36 **Werbos, P.** - Backpropagation through time, what it does and how do it - *Proc. IEEE*, 78, 1550-1560, 1990.
- 37 **Henriques J., and Dourado, A.** - A Multivariable adaptive control using a recurrent neural network *Proceedings of Eann98 - Engineering Applications of Neural Networks*, Gibraltar, 9-12 June, 118-121, 1998.
- 38 **Rumelhart, D., Hinton, G. and Williams, R.** - Learning internal representations by error propagation - *Explorations in the microstructure of cognition*. Vol 1 Foundations, Cambridge, MIT Press/Bradford books, 1986.
- 39 **Amira** - DTS200: Laboratory Setup Three-Tank System - *Amira GmbH*, 1996.
- 40 **Vaccaro, R.** - Digital Control: A State-Space Approach - *McGraw Hill International Editions*, 143-147, 1996.

# Bibliografia

- Agarwal, M., (1997) - A Systematic Classification of Neural Network Based Control, *IEEE Control Systems Magazine*, 4, 75-93.
- Ahmed, S. e A. Tasadduq, (1994a) - Block Partial Derivative and Its Application To Neural Net Based Direct Model Reference Adaptive Control, *IEE Proceedings on Control Theory and Applications*, 141, 5, 305-314.
- Ahmed, S. e A. Tasadduq, (1994b) - Neural Net Controller for Non-Linear Plants: Design Approach Through Linearization, *IEE Proceedings on Control Theory and Applications*, 141, 5, 315-322.
- Ahmed, M. e S. Riyaz, (2000) - Design of Dynamical Neural Observers, *IEE Proceedings on Control Theory and Applications*, 147, 3, 257-266.
- Albus, J., (1975a) - A New Approach To Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), *Journal of Dynamic Systems*, 220-227.
- Albus, J., (1975b) - Data Storage in The Cerebral Model Articulation controller, *Journal of Dynamic Systems*, 228-233.
- Almeida, L., (1987) - A Learning Rule for Asynchronous Perceptrons with Feedback in a Combinatorial Environment, *IEEE International Conference on Neural Networks*, Vol. 2, 609-618, San Diego, USA.
- Amira, (1996) - *Three Tank System DTS200: Laboratory Set-Up*, Amira GmbH, Duisburg.
- Anderson, J., (1972) - A Simple Neural Network Generating And Interactive Memory, *Mathematical BioSciences*, 14, 197-220.
- Anderson, B. e J. Moore, (1989) - *Optimal Control: Linear Quadratic Methods*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Antsaklis, P., (1994) - Defining Intelligent Control, Report of The Task Force On Intelligent Control, *IEEE Control Systems Magazine*, 14, 3, 58-66.
- Åström, K. e B. Wittenmark, (1973) - On Self Tuning Regulators, *Automatica*, 9, 468-478.
- Åström, K., (1983) - Theory and Application of Adaptive Control, *Automatica*, 19, 471-486.
- Åström, K. e B. Wittenmark, (1984) - *Computer Controlled Systems: Theory and Design*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Åström, K., (1989) - Toward Intelligent Control, *IEEE Control Systems Magazine*, 60- 64.
- Åström K. e B. Wittenmark, (1990) - *Computer-Controlled Systems: Theory and Design*, Prentice Hall International, NJ, USA.
- Åström, K. e B. Wittenmark, (1995) - *Adaptive Control*, Addison Wesley Publishing Company, USA.
- Babuska, R., J. Sousa e H. Verbruggen, (1999) - Predictive Control of Nonlinear Systms Based On Fuzzy and Neural Models, *ECC99 - 5rd European Control Conference*, Karlsruhe, Germany.

- Back, A. e A. Tsoi, (1991) - FIR and IIR Synapses: a New Neural Network Architecture for Time Series Modelling, *Neural Computation*, 3, 3, 375-385.
- Baker, W. e J. Farrell, (1992) - *An introduction to connectionist learning control systems*, Handbook of Intelligent Control, Chap. 2.
- Barron, A., (1993) - Universal Approximation Bounds for Super Positions of a Sigmoidal Function, *IEEE Transactions on Information Theory*, 39, 3, 930-945.
- Barto, A., R. Sutton e C. Anderson, (1983) - Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man and Cybernetics*, 13, 834-846.
- Baumman, W. e W. Rugh, (1986) - Feedback Control of Nonlinear Systems by Extended Linearization, *IEEE Transactions on Automatic Control*, 31, 1, 40-47.
- Beale, S. e B. Shafai, (1989) - Robust Control System Design with a Proportional Integral Observer, *International Journal of Control*, 50, 97-111.
- Beaufays, F. e E. Wan, (1994) - Relating Real Time Backpropagation and Backpropagation Through Time: An Application of Flow Graph Interreciprocity, *Neural Computation*, 6, 296-306.
- Bellman, R., (1957) - *Dynamic Programming*, Princeton University Press, NJ.
- Bertsekas, D. e P. Tsitsiklis, (1996) - *Neuro-Dynamic Programming*, Athena Scientific, Belmont, USA.
- Bhat, N. e T. McAvoy, (1990) - Use of Neural Nets for Dynamic Modelling and Control of Chemical Process Systems, *Computers Chemical Engineering*, 14, 5, 573-583.
- Bhattacharyya, S., (1976) - The Structure of Robust Observers, *IEEE Transactions on Automatic Control*, 21, 4, 581-588.
- Bilski, J. e L. Rutkowski, (1998) - A Fast Training Algorithm of Multilayer Perceptrons, *IEEE Transactions on Circuits and Systems II, Analogue and Digital Signal Processing*, 45, 6, 749-753.
- Bishop, C., (1995a) - *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, UK.
- Bishop, C., (1995b) - Real-time Control of a Tokamak Plasma Using Neural Networks, *Neural Computation*, 7, 206-217.
- Botto, M., (1996) - *A non-linear control approach for input constrained feedback linearized systems (PhD)*, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, Portugal.
- Botto, M., T. DenBoom, A. Krijgsman e J. Sá da Costa, (1999) - Predictive Control Based on Neural Network Models with IO Feedback Linearization, *International Journal of Control*, 72, 17, 1538-1554.
- Brdys, M, G. Kulawski e J. Quevedo, (1998) - Recurrent Networks for Non-Linear Adaptive Control, *IEE Proceedings on Control Theory and Applications*, 145, 2, 177-188.
- Brown, M., G. Lightbody e G. Irwin, (1997) - Nonlinear Internal Model Control Using Local Model Networks, *IEE Proceedings on Control Theory and Applications*, 144, 4, 505-514.
- Byrnes, C., W. Lin, W. e B. Ghosh, (1993) - Stabilization of Discrete Time Non-Linear Systems By Smooth State Feedback, *Systems & Control Letters*, 21, 255-263.
- Campolucci, P., A. Uncini e F. Piazza, (1997) - A Unifying View of Gradient Calculations and Learning for Locally Recurrent Neural Networks, *Italian workshop on neural networks*, Vietri sur Mare, Italy.
- Carpenter, G. e G. Grossberg, (1987) - A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition, *Computer Vision, Graphics and Image Processing*, 37, 54-115.
- Carr, J., (1981) - *Applications of Center Manifold Theory*, Springer Verlag, New York, USA.
- Casdagli, M., (1991) - State Space Reconstruction in Presence of Noise, *Physica D*, 51, 52-98.
- Castillo-Tolledo, B., S. DiGennaro, (1991) - Asymptotic Output Tracking for SISO Nonlinear Discrete Systems, *30th Conference on Decision and Control*, 1802-1806, Brighton, England.

- Castillo-Tolledo, B., S. DiGennaro e D. Normand-Cyrot, (1993a) - Non-linear Regulation for a Class of Discrete Time Systems, *Systems & Control Letters*, 20, 57-65.
- Castillo-Tolledo, B., (1993b) - Output Regulation of Nonlinear Systems with More Inputs Than Outputs, *International Journal of Control*, 57, 6, 1343-1356.
- Chadran, P., (1994) - Comments on Comparative Analysis of Backpropagation and The Extended Kalman Filter for Training Multilayer Perceptrons, *IEEE Transactions on Pattern Analysis Machine Intelligence*, 16, 8, 862-863.
- Chen, S. e S. Billings, (1989) - Recursive Prediction Error Parameter Estimation for Non-Linear Models, *International Journal of Control*, 49, 569-594.
- Chen, F., (1990) - Backpropagation Neural Networks for Non-Linear Self Tuning Adaptive Control, *IEEE Control Systems Magazine*, 4, 45-48.
- Chen, F. e H. Khalil, (1992) - Adaptive Control of Non-Linear Systems Using Neural Networks, *International Journal of Control*, 55, 6, 1299-1317.
- Chen, F. e H. Khalil, (1995) - Adaptive Control of a Class of Non-Linear Discrete Time Systems Using Neural Networks, *IEEE Transactions on Automatic Control*, 40, 5, 791-801.
- Chen, J. e R. Patton, (1999) - *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, USA.
- Choi, J., E. Misawa e G. Young, (1999) - A Study on Sliding Mode States Estimation, *Journal of Dynamical Systems, Measurement and Control*, 121, 255-260.
- Chu, S., R. Shoureshi e M. Tenorio, (1990) - Neural Networks for System Identification, *IEEE Control Systems Magazine*, 10, 4, 31-34.
- Chu, Y. e J. Huang, (1999) - A Neural Network Method for The Non-Linear Servomechanism Problem, *IEEE Transactions on Neural Networks*, 10, 6, 1412-1423.
- Clarke, D. e P. Gawthrop, (1975) - Self-Tuning Controller, *IEE Proceedings D*, 122, 9, 929-934.
- Cohen M. e S. Grossberg, (1983) - Absolute Stability of Global Pattern Formation and Parallel Memory Storage By Competitive Neural Networks, *IEEE Transactions on Systems, Man and Cybernetics*, 13, 5, 815-826.
- Cybenko, G., (1989) - Approximation by Superposition of a Sigmoidal Function, *Mathematics of Control, Signals and Systems*, 2, 3, 303-314.
- Davison, E., (1976) - The robust Control of a Servomechanism Problem for Linear Time-Invariant Multivariable Systems, *IEEE Transactions on Automatic Control*, 21, 25-34.
- De, S., A. Ghosh e S. Pal, (1995) - An Application of Genetic Algorithms To Evolve Hopfield Type Optimum Network Architectures for Object Extraction, *IEEE International Conference on Evolutionary Computing*, CD
- Delgado, A., C. Kambahmpati e K. Warwick, (1995) - Dynamic Recurrent Neural Networks for System Identification and Control, *IEE Proceedings on Control Theory and Applications*, 142, 4, 307-314.
- De Moor, B., P. Overshee e W. Favoreel, (1999) - *Algorithms for Subspace State Space System Identification*, Birkhauser Book series on Applied and Computational Control, Signal and Circuits, Chap. 6, 247-311, Biswa Datta Editor.
- DeVries, B. e J. Principe, (1992) - The Gamma Model: a New Neural Net Model for Temporal Processing, *Neural Networks*, 5, 4, 565-576.
- Dourado, A., (1997) - A proposal for an Information Technology Oriented Program in System and Control, *Proc. 47<sup>th</sup> IFAC Symposium on Advances in Control and Education*, 195-198, 14-16 Julho, Istambul, Turquia.
- Dourado, A., (2000) - Four Definitions as a Contribution to the Third Synthesis in Control, *Submetido para apresentação na conferência ECC01*.
- Draeger, A., S. Engell e H. Ranke, (1995) - Model Predictive Control Using Neural Networks, *IEEE Control Systems Magazine*, 10, 61-66.

- Edgart, B., (1980) - Unification of some Discrete-Time Adaptive Control Schemes, *IEEE Transactions on Automatic Control*, 25, 4, 693-697.
- Elman, J., (1990) - Finding Structure in Time, *Cognitive Science*, 14, 179-221.
- Fabri, S. e V. Kadiramanathan, (1996) - Dynamic Structure Neural Networks for Stable Adaptive Control of Non-Linear Systems, *IEEE Transactions on Neural Networks*, 7, 5, 1151-1167.
- Fairman, F. e R. Gupta, (1980) - Design of Multifunctional Reduced Order Observers, *International Journal of System Science*, 11, 1083-1094.
- Falb P. e W. Wollovith, (1967) - Decoupling in the Design and Synthesis of Multivariable Control Systems, *IEEE Transactions on Automatic Control*, 12, 6, 651-659.
- Fletcher, R., (1987) - *Practical Methods of Optimisation*, John Wiley & Sons, New York, USA.
- Francis, B. e W. Wonham, (1976) - The Internal Model Principle of Control Theory, *Automatica*, 12, 457-465.
- Francis, B., (1977) - The Linear Multivariable Regulator Problem, *SIAM Journal of Control Optimisation*, 14, 486-505.
- Frasconi, P., M. Gori e G. Soda, (1992) - Local Feedback Multilayered Networks, *Neural Computation*, 4, 1, 120-130.
- Frasconi, P. e M. Gori., (1996) - Computational Capabilities of Local-Feedback Recurrent Networks Acting as Finite-State Machines, *IEEE Transactions on Neural Networks*, 7, 6, 1521-1525.
- Fujii, S., Z. Kang e C. Zhou, (1999) - The Tracking of Mechanical Systems by an Adaptive Pole Placement Controller Combining with a Neural Network, *14th World Congress of IFAC*, 385-390, Beijing, China.
- Funahashi, K., (1989) - On the Approximate Realization of Continuous Mappings By Neural Networks, *Neural Networks*, 2, 3, 183-192.
- Funahashi, K. e Y. Nakamura, (1993) - Approximation of Dynamical Systems by Continuous Time Recurrent Networks, *Neural Networks*, 6, 801-806.
- Ge, S., C. Hang e T. Zhang, (1998) - Nonlinear Adaptive Control Using Neural Networks and Its Application to CSTR Systems, *Journal of Process Control*, 9, 313-323.
- Gil, P., J. Henriques, A. Dourado, H. Ramos, (1999) - Non-Linear Predictive Control Based On a Recurrent Neural Network, *ESIT99 - European Symposium on Intelligent Techniques*, Crete, Grece.
- Gill, P., W. Murray e M. Wright, (1981) - *Practical Optimisation*, Academic Press, London, UK.
- Gil, P., J. Henriques, A. Dourado, H. Ramos, (2000a) - Extended Neural Model Predictive Control of Non-Linear Systems, *ESIT99 - European Symposium on Intelligent Techniques*, Crete, Grece.
- Gil, P., J. Henriques, A. Dourado, H. Ramos, (2000b) - Constrained Neural Model Predictive Control with Guaranteed Free Offset, *IEEE International Conference on Industrial Electronics, Control and Instrumentation-IECON*, Nagoya, Japan.
- Girosi, F. e T. Poggio, (1989) - Representation Properties of Networks: Kolmogorov's Theorem is Irrelevant, *Neural Computation*, 1, 465-469.
- Glad, T., (1997) - Output Dead-Beat Control for Nonlinear Systems With One Zero at Infinity, *Systems & Control Letters*, 9, 249-255.
- Goh, C. e N. Edwards, (1994) - Synthesis of Optimal Feedback Controller By Neural Networks, *International Journal of Systems Science*, 25, 8, 1235-1248.
- Goodwin, G., P. Ramadge e P. Caines, (1980) - Discrete-Time Multivariable Adaptive Control, *IEEE Transactions on Automatic Control*, 25, 3, 449-453.
- Goodwin, G. e K. Sin., (1984) - *Adaptive Filtering Prediction and Control*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Gregory, P., (1958) - *Adaptive Flight Control*, Wright Air Development Center, 49-59.

- Grossberg, S., (1976) - Adaptive Pattern Classification and Universal Recoding: Parallel Development and Coding of Neural Feature Detectors, *Biological Cybernetics*, 23, 121-134.
- Gupta, M. e D. Rao, (1993) - Dynamic Neural Units with Application to the Control of Unknown Linear Systems, *Journal of Intelligent and Fuzzy Systems*, 1, 1, 73-92.
- Hagan, M. e M. Menhaj, (1994) - Training Feedforward Networks with The Marquardt Algorithm, *IEEE Transactions on Neural Networks*, 5, 6, 989-993.
- Hancock, E. e F. Fallside, (1991a) - *A Direct Control Method for a Class of Nonlinear Systems Using Neural Networks*, (Technical Report 65), Engineering Department, Cambridge University, Cambridge, England.
- Hancock, E. e F. Fallside, (1991b) - *Stable Control of Nonlinear Systems Using Neural Networks*, (Technical Report 81), Engineering Department, Cambridge University, Cambridge, England.
- Haskara, I., U. Ozguner e V. Utkin, (1998) - On Sliding Mode Observers Via Equivalent Control Approach, *International Journal of Control*, 71, 6, 1051-1067.
- Hassibi, B., D. Stork e G. Wolff, (1993) - Optimal Brain Surgeon and General Network Pruning, *IEEE International Conference on Neural Networks*, 2, 441-444, San Francisco, USA.
- Haykin, S., (1994) - *Neural Networks: a Comprehensive Foundation*, Macmillan Publishing company, New York, USA.
- H-Chen, H. A. Kremling e G. Allgower, (1995) - Nonlinear Predictive Control of a Benchmark CSTR, *3rd European Control Conference*, 3247-3252, Karlsruhe, Germany.
- Hebb, D., (1949) - *The Organization of The Behaviour*, John Wiley & Sons, New York, USA.
- Hecht-Nielsen, R., (1987) - Kolmogorov Mapping Neural Network Existence Theorem, *IEEE International Conference on Neural Networks*, Vol. 3, 11-14, San Diego, USA.
- Henriques J. e A. Dourado, (1998a) - A Multivariable Adaptive Control Using a Recurrent Neural Network, *EANN98, Engineering Applications of Neural Networks*, 118-121, Gibraltar, UK.
- Henriques, J., J. Vitor, C. Pereira, A. Dourado, (1998b) - Experimental on-line learning for a benchmark process, *CONTROLO98, 3rd Portuguese Conference on Automatic Control*, 193-198, Coimbra, Portugal.
- Henriques, J e A. Dourado, (1998c) - Adaptive Control Using a Recurrent Neural Network Observer, *CONTROLO98, 3rd Portuguese Conference on Automatic Control*, 583-488, Coimbra, Portugal.
- Henriques, J., A. Dourado, (1999a) - Multivariable Adaptive Control Using An Observer Based On a Recurrent Neural Network, *International Journal of Adaptive Control and Signal Processing*, 13, 241-259.
- Henriques, J., A. Cardoso e A. Dourado, (1999b) - Supervision and C-Means Clustering of PID Controllers for a Solar Power Plant, *International Journal of Approximate Reasoning*, 22, 73-91.
- Henriques, J., A. Dourado, (1999c) - A hybrid Neural-Decoupling Pole Placement Controller and Its Application, *ECC99 - 5rd European Control Conference*, Karlsruhe, Germany.
- Henriques, J., P. Gil, A. Dourado e H. Ramos, (1999d) - Non-linear Multivariable Predictive Control: Neural Versus First Principle Modelling Approach, *Submitted to the Journal of Systems and Control Engineering*.
- Henriques, J., P. Gil, A. Dourado e H. Ramos, (1999e) - Application of a Recurrent Neural Network in On-Line Modelling of Real-Time Systems, *ESIT99 - European Symposium on Intelligent Techniques*, Crete, Grece.
- Henriques, J., B. Castillo-Toledo, A. Titli, P. Gil e A. Dourado, (2000a) - *A Recurrent Neuronal Approach for The Nonlinear Discrete Time Output Regulation*, (Technical Report LAAS N°2000138/2000), LAAS, Toulouse, France.

- Henriques, J., B. Castillo, A. Titli, P. Gil, A. Dourado, (2000b) - A Recurrent Neural Approach for The Non-Linear Discrete Time output regulation, *IEEE International Conference on Industrial Electronics, Control and Instrumentation-IECON*, Nagoya, Japan.
- Henriques, J., B. Castillo, A. Titli, P. Gil, A. Dourado, (2000c) - Learning and output Regulation with Recurrent Neural Networks, *CONTROLO2000 – 4th Portuguese Conference on Automatic Control*, Guimarães, Portugal.
- Hepburn, J. e W. Wonham, (1984) - Error Feedback and Internal Models on Differentiable Manifolds, *IEEE Transactions on Automatic Control*, 29, 5, 397-403.
- Hertz, J., A. Krogh, e R. Palmer, (1991) - *Introduction to the Theory of Neural Computation*, Addison Wesley Publishing Company, USA.
- Holland, J., (1975) - *Adaptation in Natural and Artificial Systems*, (Technical Report), University of Michigan Press, Michigan, USA.
- Hopfield, J., (1982) - Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *National Academy of Sciences*, 79, 2554-2558, USA.
- Hopfield, J., (1984) - Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons, *Proceedings of the National Academy of Sciences*, 81, 3088-3092, USA.
- Horne, B. e C. Giles, (1995) - *An Experimental Comparison of Recurrent Neural Networks*, Advances in Neural Information Processing Systems, Chap. 7, 697-704, G. Tesauero, D. Touretzky, e T. Leen Editores.
- Hornik, K., M. Stinchcombe e H. White, (1989) - Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, 2, 5, 359-366.
- Hornik, K., M. Stinchcombe e H. White, (1990) - Universal Approximation of An Unknown Mapping and Its Derivatives Using Multilayer Feedforward Networks, *Neural Networks*, 3, 551-560.
- Hornik, K., (1991) - Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, 4, 251-257.
- Hrycej, T., (1997) - *Neurocontrol: Towards on Industrial Control Methodology*, John Wiley & Sons, USA.
- Huang, J. e W. Rugh, (1990) - On a Nonlinear Servomechanism Problem, *Automatica*, 26, 6, 963-972.
- Huang, J. e W. Rugh, (1992a) - An Approximation Method for The Non-Linear Servomechanism Problem, *IEEE Transactions on Automatic Control*, 37, 9, 1395-1398.
- Huang, J. e W. Rugh, (1992b) - Stabilization on Zero-Error Manifolds and The Non-Linear Servomechanism Problem, *IEEE Transactions on Automatic Control*, 37, 7, 1009-1013.
- Huang, J., (1995) - Output Regulation of Non-linear Systems with Nonhyperbolic Zero Dynamics, *IEEE Transactions on Automatic Control*, 40, 8, 1497-1590.
- Hunt, R., R. Su e G. Meyer, (1983) - Design of Multi Input Multi Output Nonlinear Systems, *Differential Geometric Control Theory*, 268-298.
- Hunt K. e D. Sbarbaro, (1991) - Neural Networks for Non-Linear Internal Model Control, *IEE Proceedings D*, 138, 5, 431-438.
- Hunt, K., D. Sbarbaro, R. Zbikowski e P. Gawthrop, (1992) - Neural Networks for Control Systems: A Survey, *Automatica*, 28, 6, 1083-1112.
- Hussain, A., J. Soraghan e T. Durrani, (1997) - A New Adaptive Functional-Link Neural Network for Overcoming Co-Channel Interference, *IEEE Transactions on Communications*, 45, 10, 1358-1362.
- Hwang, C. e F. Sung, (1996) - Neuro Observer Controller Design for Non-Linear Dynamical Systems, *35th Conference on Decision and Control*, 3310-3315, Kobe, Japan.

- Isermann, R., K. Lachmann e D. Matk, (1992) - *Adaptive control system*, Prentice Hall International, USA.
- Isidori, A., (1985) - *Non-Linear Control Systems: An Intoduction*, Lecture Notes in Control Science, 72, Springer Verlag.
- Isidori, A. e C. Moog, (1996) - *On the Nonlinear Equivalent of the Notion of Transmission Zeros*, Modeling and Adaptive Control, 105, Springer Verlag, 445-471, C. Byrnes e A. Kurzhanski Editores.
- Isidori, A., (1989) - *Non-Linear Control Systems, 2nd Edition*, Springer Verlag, New York, USA.
- Isidori, A. e C. Byrnes, (1990) - *Output Regulation for Non-Linear Systems*, *IEEE Transactions on Automatic Control*, 35, 2, 131-140.
- Isidori, A., (1995) - *Non-Linear Control Systems, 3rd Edition*, Springer Verlag, New York, USA.
- Isidori, A., (1996) - *Non-Linear Output Regulation*, IEEE Press, Chap. 57, Sec. 3, 923-932, Control Handbook, CRC Press Inc.
- Jacobs, R., (1988) - *Increasead Rates of Convergence Through Learning Rate Adaptation*, *Neural Networks*, 1, 295-307.
- Jacquot, R., (1995) - *Modern Digital Control Systems*, Marcel Dekker, New York, USA.
- Jagannathan, S. e F. Lewis, (1996a) - *Multilayer Discrete Time Neural Net Controller with Guaranteed Performance*, *IEEE Transactions on Automatic Control*, 7, 107-130.
- Jagannathan, S., F. Lewis e O. Pastravanu, (1996b) - *Discrete Time Model Reference Adaptive Control of Non-Linear Dynamical Systems Using Neural Networks*, *International Journal of Control*, 64, 2, 217-239.
- Jakubczyk, B. e W. Respondek, (1980) - *On Linearization of Control Systems*, *Bulletin ofAcademie Polonaise Sci. ser. Sci. Math.*, 28, 517-522.
- Jervis, T. e W. Fitzgerald, (1993) - *Optimization Schemes of Neural Networks*, (Technical Report 144), Cambridge University, Engineering Department, Cambridge.
- Jie, H. e W. Rugh, (1988) - *On a Non-Linear Multivariable Servomechanism Problem*, (Technical Report ECE-88/04.2).
- Jin, L., P. Nikiforuk e M. Gupta, (1993) - *Direct Adaptive Output Tracking Control Using Multilayered Neural Networks*, *IEE Proceedings D*, 140, 6, 393-398.
- Jin, L., P. Nikiforuk e M. Gupta, (1994a) - *Adaptive Control of Discrete Time Non-Linear Systems Using Recurrent Neural Networks*, *IEE Proceedings on Control Theory and Applications*, 141, 3, 169-176.
- Jin, L., P. Nikiforuk e M. Gupta, (1994b) - *Dynamic Recurrent Neural Network for Control of Unknown Nonlinear Systems*, *Journal of Dynamical Systems, Measurement and Control*, 116, 567-576.
- Jin, L., P. Nikiforuk e M. Gupta, (1994c) - *Absolute Stability Conditions for Discrete-Time Recurrent Neural Networks*, *IEEE Transactions on Neural Networks*, 5, 6, 854-964.
- Jin, L., P. Nikiforuk e M. Gupta, (1995a) - *Approximation of Discrete Time State Space Trajectories Using Dynamic Recurrent Networks*, *IEEE Transactions on Automatic Control*, 40, 7, 1266-1270.
- Jin, L., P. Nikiforuk e M. Gupta, (1995b) - *Adaptive Model Reference Control of Discrete Time Non-Linear Systems Using Neural Networks*, *Control Theory and Advanced Technology* , 10, 4, 1379-1399.
- Jin, L., P. Nikiforuk e M. Gupta, (1995c) - *Fast Neural Learning and Control of Discrete Time Non-Linear Systems*, *IEEE Transactions on Systems, Man and Cybernetics*, 25, 3, 478-488.
- Jin, L. e M. Gupta, (1996) - *Globally asymptotical Stability of Discrete-Time Analog Neural Networks*, *IEEE Transactions on Neural Networks*, 7, 4, 1024-1031.
- Jin, L., M. Gupta e P. Nikiforuk, (1999a) - *Dynamic Recurrent Neural Networks for Approximation of Non-Linear Systems*, *14th World Congress of IFAC*, 45-50, Beijing, China.

- Jin, L., P. Nikiforuk e M. Gupta, (1999b) - Stable Dynamic Backpropagation Learning in Recurrent Neural Networks, *IEEE Transactions on Automatic Control*, 10, 6, 1321-1334.
- Johansson, E., F. Dowla, e D. Goodman, (1992) - Backpropagation Learning for Multilayer Feedforward Neural Networks Using The Conjugate Gradient Method, *International Journal of Neural Systems*, 2, 4, 291-301.
- Jordan, M., (1986) - Attractor Dynamics and Parallelism in a Connectionist Sequential Machine, *8th Annual conference of the Cognitive Science Society*, 531-546.
- Jordan, M., (1989) - Generic Constraints On Underspecified Target Trajectories, *International Joint Conference on Neural Networks*, 1, 217-225.
- Jordan, M. e D. Rumelhart, (1990) - Forward Models: Supervised Learning with a Distal Teacher, *Cognitive Science*, 16, 313-355.
- Kalman, R., (1960) - A New Approach to Linear Filtering and Prediction Problems, *Transactions on ASM Journal of Basic Engineering*, 82, 1, 35-45.
- Kalman, B. e S. Kwasny, (1997) - High Performance Training of Feedforward and Simple Recurrent Networks, *Neurocomputing*, 14, 1, 63-83.
- Karr, C., R. Belew e L. Booker, (1991) - Genetic Algorithm Based Fuzzy Control of Spacecraft Autonomous Rendezvous, *International Conference of Genetic Algorithms*, CD.
- Kawato, M., K. Furukawa e R. Suzuki, (1987) - A Hierarchical Neural Network Model for Control and Learning of Voluntary Movement, *Biological Cybernetics*, 57, 169-185.
- Khalil, H., (1992) - *Non-Linear Systems*, Macmillan Publishing company, New York, USA.
- Kim, K., S. Shafai e E. Kappos, (1989) - Proportional Integral Estimator, *SPIE Signal and Data Processing of Small Targets*, CD.
- Kim, Y., F. Lewis e C. Abdallah, (1997) - A Dynamic Recurrent Neural Network Based Adaptive Observer for a Class of Non-Linear Systems, *Automatica*, 33, 8, 1539-1543.
- Kohonen, T., (1972) - Correlation Matrix Memories, *IEEE Transactions On Computers*, 21, 353-359.
- Kohonen, T., (1987) - *Self-organizing and Associative Memory*, Springer Verlag, Berlin, Germany.
- Kolmogorov, A., (1957) - On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable an Addition, *Doklady Akademi Nauk SSSR*, 144, 679-681.
- Kopp, R. e R. Orford, (1963) - Linear Regression Applied To System Identification for Adaptive Control Systems, *American Institute of Aeronautics and Astronautics*, 1, 10, 2300-2306.
- Koshkouei, A e A. Zinober, (1999) - Sliding Mode Controller-Observer Systems with Unmatched Uncertainty, *14th World Congress of IFAC*, 193-198, Beijing, China.
- Kosmatopoulos, E., M. Polycarpou e P. Iannou, (1995) - High-order Neural Network Structures for Identification of Dynamical Systems, *IEEE Transactions on Neural Networks*, 6, 2, 422-431.
- Kremer, S., (1995) - On the Computational Power of Elman-Style Recurrent Networks, *IEEE Transactions on Neural Networks*, 6, 4, 1000-1004.
- Kremer, S., (1999a) - Identification of a Specific Limitation On Local-Feedback Recurrent Networks Acting As Mealy-Moore Machines, *IEEE Transactions on Neural Networks*, 10, 2, 433-438.
- Kremer, S., (1999b) - *Spatio Temporal Connectionist Networks: a Taxonomy and Review*, (Technical Report), Department of Computing and Information Science, University of Guelph, Guelph.
- Krener, A. e A. Isidori, (1983) - Linearization by Output Injection and Nonlinear Observers, *Systems & Control Letters*, 3, 47-52.
- Kulawski, G. e M. Brdys, (2000) - Stable Adaptive Control with Recurrent Networks, *Automatica*, 36, 5-22.
- Kurkova, V., (1991) - Kolmogorov's Theorem is Relevant, *Neural Computation*, 3, 617-622.

- Kurtz, M. e M. Henson, (1997) - Input-output Linearizing Control of Constrained Nonlinear Process, *Journal of Process Control*, 7, 1, 3-17.
- Landau, I., (1979) - *Adaptive Control, The Model Reference Approach*, Marcel Dekker, New York, USA.
- Landau, I., R. Lozano e M. Msaad, (1997) - *Adaptive Control: Communications and Control Engineering*, Springer Verlag, NY.
- Lang, K. A. Waibel e G. Hinton., (1990) - A time-Delay Neural Network Architecture for Isolated Word Recognition, *Neural Networks*, 3, 1, 23-44.
- Lapedes, A. e R. Farber, (1987) - *Non-linear Signal Processing Using Neural Networks Prediction and System Modelling*, (Technical Report LA-UR-262), Los Alamos National Laboratory, Los Alamos.
- Larimore, W e D. Sebord, (1999) - Automated multivariable system identification: basic principles and process control applications-(Tutorial Workshop), *ECC99 - 5rd European Control Conference*, Karlsruhe, Germany.
- Le Cun, Y., (1985) - Une Procédure D'apprendissage Pour Réseau à Seuil Assymétrique, *A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences*, 599-604, Paris, France.
- Le Cun, Y., J. Denker e S. Solla, (1990) - *Optimal Brain Damage*, Advances in Neural Information Processing Systems, Chap. 19, 598-604, M. Kaufman Editor.
- Leontaritis, I. e S. Billings, (1985) - Input-Output Parametric Models for Non-Linear Systems, Part I: Deterministic Non-Linear Systems, Part II: Stochastic Non-Linear Systems, *International Journal of Control*, 41, 2, 303-344.
- Levin, A. e K. Narendra, (1993) - Control of Non-Linear Dynamical Systems Using Neural Networks: Part I, Controllability and Stabilization, *IEEE Transactions on Neural Networks*, 4, 2, 192-206.
- Levin, A. e K. Narendra, (1996) - Control of Non-Linear Dynamical Systems Using Neural Networks: Part II, Observability, Identification and Control, *IEEE Transactions on Neural Networks*, 7, 1, 30-42.
- Li, Q. e S. Tso, (1998) - PID Tuning Using Neural Networks, *WAC*, 137.1-137.6, Anchorage, USA.
- Li, Q., R. Babuska e M. Verhaegen, (1999) - Adaptive Output Tracking of Nonlinear Systems Using neural networks, *14th World Congress of IFAC*, 339-344, Beijin, China.
- Lightbody, G. e G. Irwin, (1995) - Direct Neural Model Reference Adaptive Control, *IEE Proceedings on Control Theory and Applications*, 142, 1, 31-43.
- Lightbody, G. e G. Irwin, (1997) - Nonlinear Control Structures Based on Embedded Neural System Models, *IEEE Transactions on Neural Networks*, 8, 3, 553-567.
- Lin, W. e C. Byrnes, (1994) - Design of Discrete Time Non-Linear Control Systems Via Smooth Feedback, *IEEE Transactions on Automatic Control*, 39, 11, 2340-2346.
- Lin, T., B. Horne, P. Tino e C. Giles, (1996) - Learning Long Term Dependencies with NARX Recurrent Neural Networks, *IEEE Transactions on Neural Networks*, 7, 6, 1329-1338.
- Linder, S., (1997) - *Robust Qualitative and Quantitative Methods for Disturbance Rejection and Fault Accommodation (PhD)*, Department of Mechanical, Industrial and Manufacturing Engineering, Northeastern University, Boston, USA.
- Linkens, D. e H. Nyongesa, (1996) - Learning Systems in Intelligent Control: An Appraisal of Fuzzy, Neural and Genetic Algorithm Control Applications, *IEE Proceedings on Control Theory and Applications*, 143, 4, 367-385.
- Lixin, X., (1999) - Approximation Capability of Elman Neural Networks, *14th World Congress of IFAC*, 145-148, Beijin, China.

- Ljung, L. e S. Gunnarsson, (1990) - Adaptive Tracking in System Identification: A Survey, *Automatica*, 26, 1, 7-22.
- Ljung, L., (1999) - *System Identification: Theory for The User*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Lorentz, G., (1976) - *Approximation of Functions*, Chelsea Publishing, New York, USA.
- Lu, Y., (1996) - *Improved Recurrent Learning Algorithms (MsC)*, Department of Electronic Engineering, Hong Kong Polytechnic University, Hong Kong.
- Luenberger, D., (1966) - Observers for Multivariable Systems, *IEEE Transactions on Automatic Control*, 11, 2, 190-197.
- Luenberger, D., (1971) - An Introduction to Observers, *IEEE Transactions on Automatic Control*, 16, 596-602.
- Machado, R. e A. Rocha, (1992) - Evolutive Neural Networks, *IEEE International Conference on Fuzzy Systems*, 493-500, San Diego, USA.
- Mahmood, M., A. Sadjadi e R. Liou, (1992) - Fast Learning Process of Multilayer Neural Networks Using Recursive Least Squares Method, *IEEE Transactions on Signal Processing*, 40, 2, 446-450.
- Mahmoud, M e A. Bahnasawi, (1994) - Robust Observations of Uncertain Discrete Systems, *Control Theory and Advanced Technology*, 10, 2, 249-267.
- Mamdani E. e S. Assilian, (1975) - An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, *International Journal of Man-Machine Studies*, 7, 1, 1-12.
- Marquardt, D., (1963) - An Algorithm for Least-Squares Estimation of Non-Linear Parameters, *SIAM Journal of Applied Mathematical*.
- Matlab 5.2, (1998) - *The MathWorks Inc.*
- Mattews, M., (1990) - Neural Network Nonlinear Adaptive Filtering Using The Extended Kalman Algorithm, *IEEE International Neural Network Conference*, 115-119, Paris, France.
- McCulloch, W. e W. Pitts, (1943) - A Logical Calculus of The Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, 9, 127-147.
- Mendel, J., (1986) - *Lessons in Digital Estimation Theory*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Meredith, D., L. Karr e K. Kumar, (1993) - The Use of Genetic Algorithms in The Design of Fuzzy Logic Controllers, *3rd Workshop on Neural Networks*, 549-555, Houston, USA.
- Minsky, M. e S. Papert, (1969) - *Perceptrons*, MIT Press, Cambridge, USA.
- Moller, M., (1993) - A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning, *Neural Networks*, 6, 4, 525-533.
- Monaco, A. e D. Normand-Cyrot, (1987) - Minimum Phase Non-Linear Discrete Time Systems and Feedback Stabilization, *23th Conference on Decision and Control*, 979-986, Los Angeles, USA.
- Moody. M., (1989) - Fast Learning in Networks of Locally-Tuned Processing Units, *Neural Computation*, 1, 281-294.
- Morari, M. e E. Zafiriou, (1989) - *Robust Process Control*, Prentice Hall, Englewood Cliffs.
- Moreau, Y., (1998) - *Dynamical Neural Networks and Composition Networks for System Modelling (PhD)*, Department of Electrical Engineering Katholieke Universiteit Leuven, Leuven, Belgium.
- Narendra, K. e L. Valavani, (1980a) - Stable Adaptive Controller Design, *Direct Control, IEEE Transactions on Automatic Control*, 23, 570-583.
- Narendra, K. e L. Valavani, (1980b) - Stable Adaptive Controller Design, Part II, Proof of Stability, *IEEE Transactions on Automatic Control*, 25, 440-449.
- Narendra K. e K. Parthasarathy, (1990) - Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Transactions on Neural Networks*, 1, 1, 4-27.

- Narendra, K. e K. Parthasarathy, (1991) - Gradient Methods for The Optimisation of Dynamical Systems Containing Neural Networks, *IEEE Transactions on Neural Networks*, 2, 2, 252-262.
- Narendra, K., (1996) - Neural Networks for Control: Theory and Practice, *Proceedings of the IEEE*, 84, 10, 1385-1406.
- Nelson, L. e E. Stear, (1994) - The Simultaneous On-Line Estimation of Parameters and States in Linear Systems, *IEEE Transactions on Automatic Control*, 2, 94-98.
- Nerrand O., P. Ragot, L. Personnaz e G. Dreyfus, (1993) - Neural Networks and Non-Linear Adaptive Filtering: Unifying Concepts and New Algorithms, *Neural Computation*, 5, 165-199.
- Nguyen, D. e B. Widrow, (1990) - Neural Networks for Self Learning Control Systems, *IEEE Control Systems Magazine*, 10, 4, 18-23.
- Niesler, T. e J. Plessis, (1995) - Time Optimal Control By Means of Neural Networks, *IEEE Control Systems Magazine*, 10, 23-33.
- Nijmeijer H. e A. Van der Schaft, (1990) - *Nonlinear Dynamical Control Systems*, Springer Verlag, New York, USA.
- Omatu, S., M. Halid e R. Yusof, (1996) - *Neuro-Control and Its Applications*, Springer Verlag, London, UK.
- Omatu, S., T. Fujinaka, Y. Kishida e M. Yoshioka, (1999) - Self Tuning Neuro PID for MIMO Systems, *ECC99 - 5rd European Control Conference*, Karlsruhe, Germany.
- Paiva, R., (1999) - *Identificação Neuro Difusa: Aspectos de Interpretabilidade (MsC)*, Departamento de Engenharia Informática, Universidade de Coimbra, Coimbra, Portugal.
- Parks, P., (1966) - Lyapunov Redesign of Model Reference Adaptive Control systems, *IEEE Transactions on Automatic Control*, 11, 3, 362-367.
- Parkum, J., Poulsen, e N. Holst, (1992) - Recursive Forgetting Algorithms, *International Journal of Control*, 55, 109-128.
- Parlos, A., K. Chong e A. Atiya, (1994) - Application of the Recurrent Multilayer Perceptron in Modelling Complex Process Dynamics, *IEEE Transactions on Neural Networks*, 5, 2, 255-266.
- Passino, K., (1993) - Bridging the Gap Between Conventional and Intelligent Control, *IEEE Control Systems Magazine*, 6, 12-18.
- Pearlmutter, B., (1990) - Two New Learning Procedures for Recurrent Networks, *Neural Networks*, 3, 3, 99-101.
- Pearlmutter, B., (1995) - Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey, *IEEE Transactions on Neural Networks*, 6, 5, 1212-1228.
- Pedersen, M., (1997) - *Optimisation of Recurrent Neural Networks for Time Series Modelling (PhD)*, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.
- Pereira, C., J. Henriques, A. Dourado, (2000) - Adaptive RBFNN Versus Conventional Self-Tuning: Comparison of Two Parametric Model Approaches for Non-Linear Control, *Control Engineering Practice*, 8, 3-12.
- Pham, D. e L. Xing, (1995a) - *Neural Networks for Identification, Prediction and Control*, Springer Verlag, London, UK.
- Pham, D. e L. Xing, (1995b) - *Dynamic System Identification Using Elman and Jordan Networks*, Neural Networks for Chemical Engineers, Chap. 23, 572-591, A. Bulsari Editor.
- Pineda, F., (1987) - Generalization of Backpropagation to Recurrent Neural Networks, *Physical Review Letters*, 59, 2229-2232.
- Pintelon, R. e J. Schoukens, (1991) - *Identification of Linear Systems: A Practical Guideline To Accurate Modelling*, Pergamon Press, UK.
- Pollack, J., (1990) - Recursive Distributed Representations, *Artificial Intelligence*, 46, 77-105.

- Polycarpou, M. e P. Ioannou, (1992) - Modelling, Identification and Stable Adaptive Control of Continuous Time Non-Linear Dynamical Systems Using Neural Networks, *American Control Conference*, 36-40.
- Poznyak, A., W. Yu, E. Sanchez, e P. Perez, (1999) - Non-linear Adaptive Tracking Using Dynamic Neural Networks, *IEEE Transactions on Neural Networks*, 10, 6, 1402-1411.
- Press, H., S. Teukolsky, W. Vetterling e B. Flannery, (1992) - *Numerical Recipes in C*, University Press, Cambridge, USA.
- Psaltis, D., A. Sideris, e A. Yamamura, (1988) - A Multilayered Neural Network Controller, *IEEE Control Systems Magazine*, 8, 4, 17-20.
- Psichogios, D. e L. Ungar, (1991) - Direct and Indirect Model Based Using Artificial Neural Networks, *Industrial e Engineering Chemical Research*, 30, 12, 2564-2573.
- Puskorius, G. e L. Feldkamp, (1991) - Decoupled Extended Kalman Filter Training of Feedforward Layered Networks, *International Joint Conference on Neural Networks*, I, 771-777, Seattle, USA.
- Puskorius, G. e L. Feldkamp, (1994) - Neurocontrol of Non-Linear Dynamic Systems with Kalman Filter Trained Recurrent Networks, *IEEE Transactions on Neural Networks*, 5, 2, 272-297.
- Raghavan, S. e J. Hendrick, (1994) - Observer Design for a Class of Non-Linear Systems, *International Journal of Control*, 59, 2, 515-528.
- Rajamani, R., (1998) - Observers for Lipschitz Non-Linear Systems, *IEEE Transactions on Automatic Control*, 43, 3, 397-401.
- Ramos, J., (1998) - *Projecto e Aplicação De Controladores Difusos Em Tempo Real (MsC)*, Departamento de Engenharia Informática, Universidade de Coimbra, Coimbra, Portugal.
- Raol, J., (1995) - Parameter Estimation of State Space Models by Recurrent Neural Networks, *IEE Proceedings on Control Theory and Applications*, 142, 2, 114-118.
- Rayner, P., (1989) - A New Connectionist Model Based On a Non-Linear Adaptive Filter, *ICASSP*, 1191-1194.
- Reilly, J., (1983) - *Observer for Linear Systems*, Academic Press, New York, USA.
- Ribeiro, B., (1995) - *Contribuição Para a Modelização, Supervisão e Controlo de Sistemas Multi-Variáveis Não Lineares por Redes Neurais com Aplicação a um Forno De Cal (PhD)*, Departamento de Engenharia Informática, Universidade de Coimbra, Coimbra, Portugal.
- Rivals, I. e L. Personnaz, (1995a) - *Black-Box Modelling with State-Space Neural Networks*, Neural Adaptive Control Technology, Chap. 8, 237-263, R. Zbikowski e K. Hunt Editores.
- Rivals, I. e L. Personnaz, (1996) - Internal Model Control Using Neural Networks, *IEEE International Symposium on Industrial Electronics*, CD.
- Robinson, T. e F. Fallside, (1991) - A Recurrent Error Propagation Network Speech Recognition System, *Computer Speech and Language*, 5, 254-274.
- Rosenblatt, F., (1958) - The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain, *Psychological Review*, 65, 386-408.
- Rovithakis, G. e M. Christodoulou, (1994) - Adaptive Control of Unknown Plant Using Dynamical Neural Networks, *IEEE Transactions on Systems, Man and Cybernetics*, 24, 3, 400-412.
- Rovithakis, G. e M. Christodoulou, (1995) - Direct Adaptive Regulation of Unknown Non-Linear Dynamic Systems Via Dynamic Neural Networks, *IEEE Transactions on Systems, Man and Cybernetics*, 25, 12, 1578-1594.
- Rovithakis, G. e M. Christodoulou, (1996) - Direct Adaptive Regulation Using Dynamic Neural Networks: Application to DC Motors Speed Control, *Mathematics and Computers in Simulation*, 41, 53-62.
- Rovithakis, G. e M. Christodoulou, (1997) - Neural Adaptive Regulation of Unknown Non-Linear Dynamical Systems, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 27, 5, 810-822.

- Ruano, A., P. Fleming e D. Iones, (1992) - Connectionist Approach for PID Tuning, *IEE Proceedings D*, 129, 279-285.
- Ruck, D., S. Rogers, M. Kabrisky, P. Matbeck e M. Oxley, (1992) - Comparative Analysis of Backpropagation and The Extended Kalman Filter for Training Multilayer Perceptrons, *IEEE Transactions on Pattern Analysis Machine Intelligence*, 14, 6, 686-691.
- Rumelhart, D., G. Hinton e R. Williams, (1986) - *Learning Internal Representations by Error Propagation*, PDP Research Group, Parallel Distributed Processing, MIT Cambridge, USA.
- Saarinen, S., R. Bramley, R., e G. Cybenko, (1993) - Ill Conditioning in Neural Network Training Problems, *SIAM Journal of Scientific Computing*, 14, 693-714.
- Saberi, A. e A. Stoorvogel, (2000) - *On Output Regulation for Linear Systems*, (Technical Report), Department of Electrical & Computer Engineering, Rutgers University, NJ, USA.
- Saint-Donat, J. N. Bhat e T. McAvoy, (1991) - Neural Net Based Model Predictive Control, *International Journal of Control*, 54, 6, 1453-1468.
- Sanner, R. e J. Slotine, (1992) - Gaussian Networks for direct adaptive control, *IEEE Transactions on Neural Networks*, 3, 6, 837-863.
- Saerens M. e A. Soquet, (1991) - Neural Controller Based On Back-Propagation Algorithm, *IEE Proceedings, Part F: Radar and Signal Processing*, 138, 1, 55-62.
- Schmidhuber, J., (1992) - A Fixed Size Storage  $O(N^3)$  Time Complexity Learning Algorithm for Fully Recurrent Running Networks, *Neural Computation*, 4, 2, 243-248.
- Schreier, G., J. Ragot, R. Patton e P. Frank, (1999) - Observer Design for a Class of Non-Linear Systems, *ECC99 - 5rd European Control Conference*, 498-503, Karlsruhe, Germany.
- Shafai, B., B. Niemann e J. Stroustrup, (1996) - LTR design of Discrete-Time Proportional Integral Observers, *IEEE Transactions on Automatic Control*, 41, 7, 1056-1062.
- Shinskey, F., (1994) - *Feedback Controllers for The Process Industries*, McGraw Hill, New York, USA.
- Singhal, S. e L. Wu, (1989) - *Training Multilayer Perceptrons with The Extended Kalman Filter Algorithm*, Advances in Neural Information Processing Systems, Chap. 1, 133-140, M. Kaufmann Editor.
- Sjöberg, J., H. Hjalmarsson e L. Ljung, (1994) - Neural Networks in System Identification, *Proc. 10th Symposium on System Identification*, 1, 167-190, Copenhagen, Denmark.
- Sjöberg, J., Q. Zhang, L. Ljung. A. Bienveniste, B. Deylon, P. Glorennec, H. Hjalmarsson e A. Juditsky, (1995) - Non-linear Black Box Modelling in System Identification: A Unified Overview, *Automatica*, 31, 12, 1691-1724.
- Slotine, J., J. Hedrock e E. Misawa, (1987) - On Sliding Mode Observers for Non-Linear Systems, *ASME - Journal of Dynamic Systems Measurement and Control*, 109, 245-252.
- Soderström, T. e P. Stoica, (1989) - *System Identification*, Prentice Hall International, Hertfordshire.
- Sontag, E., (1992) - Neural Networks as a Systems Models and Controllers, *Trentelman H. L. and Willems J. C. Eds*, 339-380.
- Sørensen, O., (1991) - Optimal Control with Neural Networks, *IASTED, International Association of Science and Technology for Development*, 90-95.
- Sørensen, O., (1996) - Non-linear Pole Placement Control with a Neural Network, *European Journal of Control*, 2, 36-43.
- Sprecher, D., (1965) - On the Structure of Continuous Functions of Several Variables, *Transactions of the American Mathematical Society*, 115, 340-355.
- Srinivasan, B., U. Prasad e N. Rao, (1994) - Backpropagation Through Adjoints for The Identification of Non-Linear Dynamic Systems Using Recurrent Neural Models, *IEEE Transactions on Neural Networks*, 5, 2, 213-228.
- Stager, F. e M. Agarwal, (1997) - Three Methods for Speed up The Training of Feedforward and Feedback Perceptrons, *Neural Networks*, 10, 8, 1435-1443.

- Staib, W., (1993) - The Intelligent Arc Furnace: Neural Networks Revolutionise Steelmaking, *World Congress on Neural Networks*, 441-466, Portland.
- Stephanopoulos G. e C. Han, (1994) - Intelligent Systems in Process Engineering, *5th International Symposium on Process System Engineering*, Kyongju, Korea.
- Stone, M., (1948) - The generalized Weierstrass Approximation Theorem, *Mathematics Magazine*, 21, 167-183.
- Sureshbabu, N. e W. Rugh, (1995) - Output Regulation with Derivative Information, *IEEE Transactions on Automatic Control*, 40, 1755-1766.
- Sureshbabu, N. e W. Rugh, (1997) - On Output Regulation for Discrete Time Non-Linear Systems, *Automatica*, 33, 9, 1683-1689.
- Suykens, J., B. Moor e J. Vandewalle, (1995) - Nonlinear System Identification Using Neural State Space Models, Applicable to Robust Control Design, *International Journal of Control*, 62, 129-152.
- Suykens, J., J. Vandewalle e B. De Moor, (1996) - *Artificial Neural Networks for Modelling and Control of Non Linear Systems*, Kluwer Academic Publishers, The Netherlands.
- Suykens, K., B. DeMoor e J. Vandewalle, (1997) - NLq Theory: A Neural Control Framework with Global Asymptotic Stability Criteria, *Neural Networks*, 10, 615-637.
- Takens, F., (1981) - Detecting Strange Attractors in Turbulence, *Dynamical Systems and Turbulence*, 898, 366-381.
- TeBraake, H., E. Can, J. Scherpen e H. Verbruggen, (1997) - Control of Nonlinear Chemical Processes Using Neural Models and Feedback Linearization, *Computers Chemical Engineering*, 22, 4, 1113-1127.
- TeBraake, H., M. Botto, H. Can, J. Sá da Costa e H. Verbruggen, (1999) - Linear Predictive Control Based On Approximate Input-Output Feedback Linearization, *IEE Proceedings on Control Theory and Applications*, 146, 4, 295-300.
- Thau, F., (1973) - Observing the State of Non-Linear Dynamical Systems, *International Journal of control*, 17, 471-479.
- Tsoi, A. e A. Back, (1992) - An Adaptive Lattice Architecture for Dynamic Multiplayer Perceptrons, *Neural Computation*, 4, 922-931.
- Tsoi, A. e A. Back, (1997) - Discrete Time Recurrent Neural Architectures: a Unifying View, *Neural Computation*, 15, 183-223.
- Tsypkin, Y., (1971) - *Adaptation and Learning in Automatic Systems*, Academic Press, New York.
- Utkin, V., (1992) - *Sliding Modes in Control and Optimisation*, Springer Verlag, USA.
- Verma, B., (1997) - Fast Training of Multilayer Perceptrons, *IEEE Transactions on Automatic Control*, 8, 6, 1314-1320.
- Walcott, B. e S. Zak, (1987a) - Comparative Study of Non-Linear State Observation Techniques, *International Journal of Control*, 45, 6, 2109-2132.
- Walcott, B. e S. Zak, (1987b) - State Observation of Non-Linear Uncertain Dynamical Systems, *IEEE Transactions on Automatic Control*, 32, 2, 166-170.
- Wan, E. e A. Nelson, (1997) - *Dual Kalman Filtering Methods for Non-Linear Prediction, Estimation, and Smoothing*, Advances in Neural Information Processing Systems, Cambridge, USA.
- Wan, E. e A. Nelson, (2000) - *Dual EKF Methods*, (Technical Report), Department of Electrical and Computer Engineering, Oregon Graduate Institute, Oregon, USA.
- Wellstead, P., E. Zanker e J. Edmunds, (1979) - Pole Assignment Self-Tuning Regulator, *IEE Proceedings D*, 126, 781-787.
- Wellstead, P. e M. Zarrop, (1991) - *Self Tuning Systems, Control Signal and Signal Processing*, John Wiley & Sons, USA.

- Werbos, P., (1974) - *Beyond Regression: New Tools for Prediction and Analysis in The Behavioural Sciences (PhD)*, Harvard University, Harvard, USA.
- Werbos, P., (1990) - Overview of Design and Capabilities, *Neural Networks for Control*, 59-65.
- Werbos, P., (1990) - Backpropagation Through Time: What It Does and How To Do It, *IEEE, Special Issue on Neural Networks*, 78, 10, 1550-1560.
- Werbos, P., (1991) - An Overview of Neural Networks for Control, *IEEE Control Systems Magazine*, 11, 1, 40-41.
- White, D e D. Sofge, (1992) - *Handbook of Intelligent Control Neural, Fuzzy, and Adaptive Approaches*, Van Nostrand Reinhold, New York.
- Widrow, B. e M. Hoff , (1960) - Adaptive Switching Circuits, *IRE WESCON Convention Record*, 4, 96-104, New York, USA.
- Widrow B. e F. Smith, (1963) - Pattern-Recognizing Control Systems, *Computer and Information Sciences*, Spartan, Washington USA.
- Williams, R. e D. Zipser, (1989a) - A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, 1, 2, 270-280.
- Williams, R. e D. Zipser, (1989b) - Experimental Analysis of The Real Time Recurrent Learning Algorithm, *Connection Science*, 1, 1, 87-111.
- Williams, R., (1990a) - *Adaptive State Representation and Estimation Using Recurrent Connectionist Networks*, Neural Networks for Control, Chap. 4, 97-114, W. Miller, R. Sutton, P. Werbos Editores.
- Williams, R. e J. Peng, (1990b) - An Efficient Gradient Based Algorithm for Online Training of Recurrent Network Trajectories, *Neural Computation*, 2, 490-501.
- Williams, R. e D. Zipser, (1990c) - *Gradient Based Learning Algorithms for Recurrent Networks and Their Computational Complexity*, (Technical Report), College of Computer Science, Northeastern University, Boston, USA.
- Williams, R., (1992) - Training Recurrent Networks Using The Extended Kalman Filter, *International Joint Conference on Neural Networks*, 4, 241-246, Baltimore, USA.
- Williams, R. e D. Zipser, (1995) - *Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity*, Backpropagation: Theory, architectures and applications, Chap. 13, 433-486, Yves Chauvin and D. Rumelhart Editores.
- Yaz, E. e A. Azemi, (1993) - Variable Structure Observer with a Boundary Layer for Correlated Noise-Disturbance Models and Disturbance Minimization, *International Journal of control*, 57, 5, 1191-1206.
- Yesildirek, A. e F. Lewis, (1995) - Feedback Linearisation Using Neural Networks, *Automatica*, 31, 11, 1659-1664.
- Yu, W. e A. Poznyak e J. Alvarez, (1998) - *Robust Adaptive Control for Multicomponent Distillation Column Via Dynamic Neural Network*, (Technical Report), Seccion de Control Automatico, Cinvestav-IPN, Mexico.
- Yu, W. e A. Poznyak, (1999a) - Indirect Adaptive Control Via Parallel Dynamic Neural Networks, *IEE Proceedings on Control Theory and Applications*, 146, 1, 25-30.
- Yu, W., A. Poznyak e E. Sanchez, (1999b) - Neural Adaptive Control with Unmodelled Dynamic Compensation, *14th World Congress of IFAC*, 345-350, Beijing, China.
- Yu, W., M. Moreno e X. Li, (2000) - Observer Based Neuro Identifier, *IEE Proceedings on Control Theory and Applications*, 147, 2, 145-152.
- Zadeh, L., (1965) - Fuzzy Sets, *Information and Control*, 8, 338-353.
- Zadeh, L., (1994) - Fuzzy Logic, Neural Networks and Soft Computing, *Communications on the ACM*, 37, 3, 77-84.
- Zbikowski, R., (1994) - *Recurrent Neural Networks, Some Control Aspects (PhD)*, Faculty of Engineering of Glasgow, Glasgow University, Glasgow, UK.

- Zhu, Q., Z. Ma e K. Warwick, (1999) - Neural Network Enhanced Generalised Minimum Variance Self Tuning Controller for Non-Linear Discrete Time Systems, *IEE Proceedings on Control Theory and Applications*, 146, 4, 319-326.
- Zipser, D., (1989) - A Sub Grouping Strategy That Reduces Learning Complexity and Speeds Up Learning in Recurrent Networks, *Neural Computation*, 1, 4, 552-558.