

# Relatório de Estágio

**Estágio para licenciatura no tema**

Técnicas de Simulação, Controlo e Optimização para Sistemas Autónomos-  
Aplicações na Robótica e em Processos de Manufactura

**PRODEP Proc. N° 4.3/7.04/92 / 93**

**Autor**

Alfredo Manuel de Oliveira Martins

**Junho de 1993**

01527

424

Universidade do Porto  
Faculdade de Engenharia  
Biblioteca 4

621.3(047.3)/LEEC 1992/MAR 2

25 09 09

# **ISR - Instituto de Sistemas e Robótica**

**DEEC - R. dos Bragas, 4099 Porto Codex, Portugal**

**Fax: 351 - 2- 319280**

**Tel: 351 -2 - 2007505**

---

**REFERÊNCIA DO PROJECTO 4.3/1.56/92**

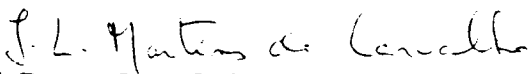
## **Parecer Técnico**

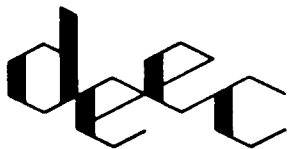
Acompanhei os trabalhos de estágio realizados no Pólo do Porto do Instituto de Sistemas e Robótica pelo Lic<sup>o</sup> Alfredo Manuel de Oliveira Martins, bem como apreciei os trabalhos produzidos e apresentados em Congressos Internacionais e o relatório oportunamente apresentado.

Com base no acompanhamento referido, considero que aqueles trabalhos reflectem o trabalho efectivamente realizado, revelando elevado empenhamento do estagiário.

Sou portanto de parecer que foram plenamente atingidos os objectivos inicialmente propostos no plano de trabalhos,

O Supervisor do Estágio no Instituto de Sistemas e Robótica

  
Prof. Doutor Jorge L. Martins de Carvalho



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Departamento de Engenharia Electrotécnica e de Computadores

Rua dos Bragas, 4099 Porto Codex, PORTUGAL  
Telef. 361-2-317105/107/412/457 · Telex 27323 FEUP P · Telefax 361-2-319280

**Acção de Formação Prodep 4.3/7.04/92**

Formando: Alfredo Manuel de Oliveira Martins

Orientador Científico: Fernando Manuel Ferreira Lobo Pereira

Este relatório corresponde à apreciação científica das actividades desenvolvidas pelo Lic<sup>o</sup> Alfredo Manuel de Oliveira Martins durante o segundo semestre de 1992.

O trabalho desenvolvido foi de elevado mérito e executado com grande dedicação.

Os trabalhos realizados e o relatório apresentados pelo formando reflectem, com rigor a natureza dos trabalhos executados.

Além do empenhamento e das excelentes qualidades do estagiário concorreram para o sucesso deste trabalho, os meios disponibilizados.

O Supervisor do Estágio na FEUP

Fernando M. F. Lobo Pereira  
Prof. Aux. da FEUP

# **Modelização de Veículos Submarinos Autónomos**

*Relatório apresentado para o estágio do PRODEP sob o tema:  
Técnicas de Simulação, Controlo e Optimização para Sistemas Autónomos-  
Aplicações na Robótica e em Processos de Manufatura*

*Realizado por: Alfredo Manuel de Oliveira Martins*

**ISR-PORTO**

## Índice

1 Actividade desenvolvida .....	1
1.1 Estudos orientados efectuados .....	1
1.2 Disciplinas frequentadas .....	2
1.3 Participações em Conferências.....	2
1.4 Estágios .....	2
2. Apêndices .....	3
2.1 Projecto Realizado .....	3

## **1 Actividade desenvolvida**

### **1.1 Estudos orientados efectuados**

Estudo de técnicas de simulação, controlo e optimização para sistemas autónomos-aplicações na robótica e em processos de manufactura.

Verificou-se num campo específico da robótica móvel, inúmeras carências no que diz respeito a técnicas e à teoria base, que permitam um desenvolvimento a nível industrial deste tipo de aplicações. Este campo: Robótica móvel submarina, pela sua grande possibilidade de intervenção nas áreas económicas e científicas, é pois uma área que urge aprofundar.

Várias técnicas e paradigmas estavam disponíveis para o estudo de veículos submarinos autónomos. Nomeadamente no que diz respeito a controlo: Controlo Robusto, Controlo Multivariável, Controlo " Sliding-Modes", Controlo Adaptativo, Redes Neurais e Teoria clássica de Sistemas. Em identificação e modelização temos várias áreas da Física: Hidrostática, Hidrodinâmica e Mecânica, e áreas do conhecimento tais como: Métodos de estimação de parâmetros, construção de modelos estocásticos e ainda técnicas de simulação.

Levantamento de resultados teóricos disponíveis e análise das respectivas limitações, nomeadamente em termos de aplicações.

Para o efeito efectuaram-se pesquisas bibliográficas exaustivas em várias bibliotecas sobre os temas em consideração. Esta pesquisa foi particularmente frutuosa porque permitiu coligir e agrupar muitos documentos sobre as áreas estudadas que se encontravam dispersos e por várias instituições nacionais e internacionais.

Modelização e estudo de técnicas disponíveis tais como redes neurais sob supervisão do orientador, para ultrapassar as limitações encontradas no controlo de veículos submarinos autónomo.

No âmbito dos trabalhos desta fase foi feito um levantamento do estado da arte actual, da teoria redes neurais e a sua aplicação em controlo, nomeadamente ao de veículos submarinos autónomos.

## **1.2 Disciplinas frequentadas**

Controlo Ótimo

Identificação e Modelização de Sistemas

Métodos de Optimização

Robótica

Análise de Sistemas Informáticos

## **1.3 Participações em Conferências**

Participação em diversas palestras sobre modelização e controlo de veículos submarinos autónomos, efectuadas na FEUP pelo Prof. Anthony Healey da Naval Postgraduate School (EUA).

Participação num breve curso leccionado pelo Prof. Carlos Pedreira da Pontifícia Universidade Católica do Rio de Janeiro, sobre Teoria de Redes Neurais.

Participação no " Workshop on Autonomous Underwater Vehicles " realizado no Porto e durante o qual foram abordados temas por várias personalidades pioneiras na área, nomeadamente o Prof. James Bellingham do MIT (EUA), Prof. Anthony Healey, Prof. Thor Fossen NIT (Noruega).

## **1.4 Estágios**

O estágio no Instituto de Sistemas e Robótica permitiu, por um lado, o acesso a uma rede de computação necessária ao desenvolvimento de algumas fases do trabalho e, por outro, tornou possível a participação em projectos de I&D nacionais e europeus que endereçam aspectos práticos dos tópicos abordados no âmbito deste trabalho.



## **2. Apêndices**

### **2.1 Projecto Realizado**

# **Modelização de Veículos Submarinos Autónomos**

*Relatório apresentado para o estágio do PRODEP sob o tema:  
Técnicas de Simulação, Controlo e Optimização para Sistemas Autónomos-  
Aplicações na Robótica e em Processos de Manufatura*

**ISR-PORTO**

1. Introdução.....	3
2. Modelo do AUV.....	4
2.1 Sistemas de coordenadas.....	4
2.2 O estado do veículo.....	4
2.3 Ângulos de Euler.....	5
2.4 Hidrostática.....	7
2.5 Hidrodinâmica.....	10
2.6 Actuadores .....	13
2.7 Sensores .....	14
2.8 Resumo do modelo .....	15
3. Conclusões e Perspectivas Futuras.....	21
4. Bibliografia.....	22

# 1. Introdução

Para exploração do meio subaquático existem poucos meios. Estes incluem: o uso de mergulhadores (autónomos ou não), submersíveis tripulados, submersíveis não tripulados e controlados remotamente e veículos submarinos autónomos.

Os submersíveis tripulados, existentes hoje em dia, deparam também com o problema da profundidade, e da sua maleabilidade. Apesar de já se ter feito descer veículos tripulados à profundidade de dezenas de quilómetros (fossa das Marianas no Oceano Pacífico) as condições de operacionalidade em que isto foi feito deixam muito a desejar. Restam os veículos robóticos submarinos (autónomos ou não). Estes pela sua grande flexibilidade permitem explorar ambientes hostis aos mergulhadores tais como locais poluídos, descer a profundidades consideráveis e efectuar o mais variado tipo de tarefas, tais como as de medição de grandezas do meio, inspecção e manutenção, filmagem, etc.

Nesta classe de veículos destacam-se os veículos submarinos autónomos (doravante designados por AUV's da sigla inglesa "Autonomous Underwater Vehicles") que têm ainda a facilidade de trabalhar em meios onde a comunicação entre o veículo e um operador é bastante dificultada, ou de efectuar missões de uma forma muito mais eficiente. A capacidade dos AUV's de serem programados para uma missão específica e depois a executarem trazendo se necessário resultados, faz deles um meio necessário e promissor na exploração subaquática.

No entanto, a tecnologia destes veículos encontra-se ainda na sua infância. Não existem sistemas precisos e baratos para navegação, o conhecimento das leis hidrodinâmicas que governam o comportamento dos AUV's é, ou ainda incompleto ou de uma complexidade excessiva., a própria tecnologia de construção está ainda a dar os seus primeiros passos e o problema genérico do controlo e gestão de missão está ainda pouco explorado.

Neste contexto, surge a necessidade de se conseguirem boas soluções para um controlo eficaz nas mais diversas situações. Estas podem incluir desconhecimento do meio onde o AUV opera (correntes, pressões, temperaturas, topologia do fundo oceânico, etc), desconhecimento de características do próprio AUV (incertezas nos modelos) e situações inesperadas tais como obstáculos moveis ou situações de avaria.

Este trabalho foca dentro do vasto tema que é o controlo de AUV's uma pequena parte que diz respeito ao controlo a baixo nível (aquilo que geralmente é estudado na teoria de controlo).

Mais precisamente pretende-se modelizar um veículo submarino autónomo e estudar diversas estratégias para o controlo nomeadamente redes neuronais.

Iremos muitas vezes utilizar nomenclatura em língua inglesa, dado ser esta a de uso comum.

## 2. Modelo do AUV

Para se poder efectuar um estudo sobre o controlo de AUV's é necessário possuir um modelo para o veículo. Só assim poderemos estudar os efeitos das metodologias propostas. A necessidade da existência de modelos põe-se-nos, quer face à impossibilidade de experimentar diferentes estratégias num veículo real (modelo como base de simulação), quer pela necessidade de um modelo comportamental que fundamente e justifique diversas abordagens a propor.

Vamos nesta secção introduzir brevemente o aspecto da modelização de AUV's. O modelo introduzido, com variantes pontuais (descritas quando existirem) será a base das nossas considerações futuras.

### 2.1 Sistemas de coordenadas

No estudo do veículo, vamos considerar dois sistemas de coordenadas (cujo o uso é comum na literatura, sobre modelização de AUV's). Vamos também seguir a notação usada pela "Society of Naval Architects and Marine Engineers" .

Um sistema de coordenadas, é exterior ao veículo e corresponde àquele que em física é usualmente denominado como "referencial inercial". Iremos designar este referencial como: GLOBAL. Este sistema de coordenadas é fixo no meio exterior ao veículo e normalmente encontra-se referido à Terra. Neste caso desprezam-se os efeitos da rotação e da progressão do planeta no espaço. Se os quiséssemos incluir no modelo do AUV, este referencial tomaria como base as coordenadas de um grupo distante de estrelas no universo (note-se que de uma forma geral os efeitos directos dos movimentos da Terra sobre os AUV's, são negligíveis comparados com outras forças que neles actuam.

O outro sistema de coordenadas encontra-se solidário com o veículo. A origem deste referencial não necessita de encontrar em nenhum ponto especial do veículo, embora seja usual coloca-la no "centro geométrico". Por razões práticas, é usual fazer coincidir a origem deste sistema com a intersecção dos planos de simetria do veículo (se os houver, e para facilitar os cálculos momentos de inércia). A este sistema de coordenadas chamaremos de BODY FIXED (da nomenclatura inglesa, referindo-se a ser fixo com o veículo).

### 2.2 O estado do veículo

O *estado* de um sistema, corresponde a toda a informação que num dado instante pode descrever a situação do sistema. De uma forma geral, conhecidas as leis que governam o sistema, conhecido o seu estado num instante, conhecemos todo o seu passado e o seu futuro.

No caso do AUV o estado vai corresponder um vector com seis componentes. Este estado pode ser dado quer no referencial GLOBAL, quer no BODY FIXED.

As seis componentes correspondem à caracterização da velocidade linear do veículo (nas três dimensões do espaço) e à sua velocidade angular (ou seja a taxa de variação da atitude do veículo, isto é, três velocidades angulares).

À posição do veículo no referencial GLOBAL, nos três eixos iremos de: X, Y e Z. Fazendo uso da convenção em que o sentido positivo para o eixo Z é para baixo. Logo  $\dot{X}$ ,  $\dot{Y}$  e  $\dot{Z}$  correspondem à componente da velocidade ( $\underline{v}$  na Figura 2.1) nas três direcções. Estas três componentes no referencial BODY FIXED são u, v e w ("surge", "sway" e "heave").

As três componentes da velocidade angular do veículo ( $\underline{\omega}$  na Figura 2.1), no referencial BODY FIXED são p, q e r, respectivamente: "roll rate", "pitch rate" e "yaw rate". No GLOBAL, iremos usar as taxas de variação dos ângulos de Euler (ver secção seguinte):  $\dot{\Phi}$ ,  $\dot{\theta}$  e  $\dot{\Psi}$ .

Na Figura 2.1 podemos observar além da indicação das componentes da velocidade, também os vectores posição  $R_o$  e  $R_g$ , respectivamente da origem do referencial BODY FIXED e do centro de gravidade do veículo.

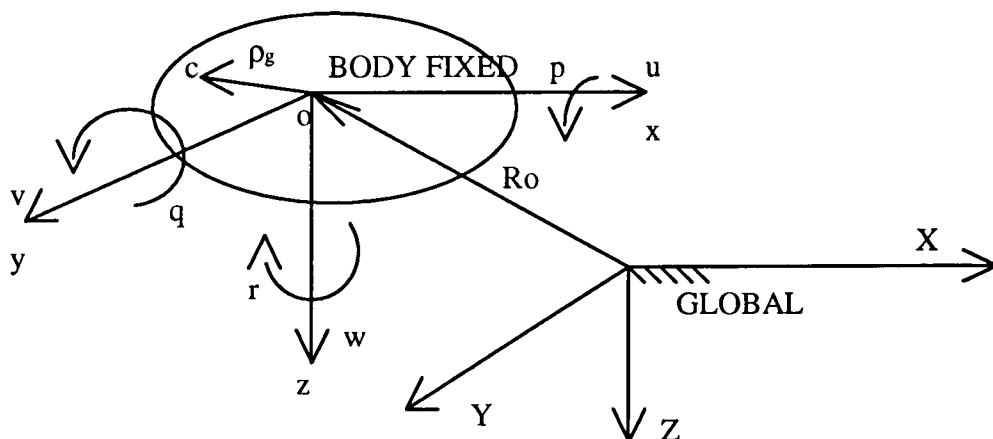


Figura 2.1

### 2.3 Ângulos de Euler

Existem várias formas de descrever a orientação de um referencial em relação outro. Iremos aqui introduzir, a aplicação dos "ângulos de Euler", para transformar a orientação do AUV, descrita no sistema BODY FIXED para o sistema de coordenadas GLOBAL.

Os ângulos de Euler  $\Psi, \theta$  e  $\Phi$  ("azimuth", "elevation" e "spin" na literatura inglesa), descrevem três rotações aplicadas a um referencial para o posicionar com os eixos paralelos a outro.

No nosso caso estamos interessados em transformar a velocidade linear e angular do veículo descrita no referencial BODY FIXED, para a sua descrição no referencial GLOBAL.

Suponha-se os dois sistemas de coordenadas com origem coincidente (basta efectuar uma translação do BODY FIXED para tal acontecer). Os ângulos de Euler descrevem como, rodando o referencial BODY FIXED este fica com os eixos coincidentes com o GLOBAL.

A rotação é efectuada de acordo com a convenção seguinte:

Roda-se de  $\Psi$  em torno do eixo Z.

Roda-se de  $\theta$  em torno do eixo Y

Roda-se de  $\Phi$  em torno do eixo X

Teremos pois que  $\dot{R}_o$  pode ser obtido aplicando uma matriz de transformação ao vector  $[u,v,w]^T$ . Esta matriz de transformação vai ser função dos ângulos de Euler que descrevem a orientação do referencial BODY FIXED em termos do GLOBAL.

$$\dot{R}_o = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = T_1(\Theta) \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

onde  $\Theta$  é o vector com os ângulos de Euler

$$\Theta = \begin{bmatrix} \Phi \\ \theta \\ \Psi \end{bmatrix}$$

De igual forma se pode obter uma transformação entre a velocidade angular descrita no referencial BODY FIXED  $[p \ q \ r]^T$  e as derivadas dos ângulos de Euler  $\dot{\Theta}$

$$\dot{\Theta} = \begin{bmatrix} \dot{\Phi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} = T_2(\Theta) \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Teremos assim que o estado descrito no referencial GLOBAL pode ser obtido do descrito no BODY FIXED por aplicação de uma matriz de transformação:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\Phi} \\ \dot{\theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

Através de alguns cálculos [Fossen 91] pode-se chegar às seguintes expressões para as matrizes  $T_1$  e  $T_2$ :

$$T_1 = \begin{bmatrix} \cos \Psi \cos \theta & -\sin \Psi \cos \Phi + \cos \Psi \sin \theta \sin \Phi & \sin \Phi \sin \Psi + \cos \Psi \cos \Phi \sin \theta \\ \sin \Psi \cos \theta & \cos \Psi \cos \Phi + \sin \Phi \sin \theta \sin \Psi & -\cos \Psi \sin \Phi + \sin \theta \sin \Psi \cos \Phi \\ -\sin \theta & \cos \theta \sin \Phi & \cos \theta \cos \Phi \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 1 & 0 & \cos \Phi \tan \theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi / \cos \theta & \cos \Phi / \cos \theta \end{bmatrix}$$

A transformação inversa pode ser obtida por aplicação das matrizes  $T_1^{-1}$  e  $T_2^{-1}$ .

Note-se que para valores de  $\theta = \pm 90^\circ$ , a matriz  $T_2$  não é invertível. Esta limitação não nos põe problemas na prática, já que o AUV nunca mergulha e emerge com ângulo de "pitch" de  $90^\circ$  (posição com a frente do veículo apontada verticalmente para cima ou para baixo, nunca desce e sobe "a pique").

Temos pois com as relações acima definidas uma forma de descrever a cinemática do AUV.

## 2.4 Hidrostática

Vamos considerar o veículo como um corpo rígido imerso num fluido.

Podemos pois dizer que ele está sujeito a diversas forças exteriores. Nesta secção não iremos tratar das forças hidrodinâmicas.

Podemos obter :

$$R_g = R_o + \rho_g$$



Dado que :

$$\dot{R}_O = v$$

Então derivando  $R_g$  em ordem ao tempo teremos:

$$R_g = (v + \omega \times \rho_g)$$

Pela segunda lei de Newton obtém-se:

$$m(\dot{v} + \dot{\omega} \times \rho_g + \omega \times v + \omega \times \omega \times \rho_g) = F_e \quad (2.1)$$

Onde  $F_e$  contem a soma de todas as forças exteriores. O termo  $\dot{\omega} \times \rho_g$  é a aceleração angular,  $\omega \times v$  é a aceleração de Coriolis e  $\omega \times \omega \times \rho_g$  é a aceleração centrífuga.

Nestas equações temos as forças exteriores, de igual forma podemos escrever equações semelhantes para os momentos exteriores.

O momento angular  $H$ , pode ser calculado por :

$$H = I\omega$$

A matriz  $I$  é a matriz dos momentos de inércia e é diagonal no caso de o ponto para onde é calculada, estar na intersecção dos eixos de simetria do veiculo. Daí o interesse que a origem do referencial BODY FIXED esteja na intersecção dos eixos de simetria.

A matriz  $I$  tem pois dimensão 3\*3:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Os elementos da diagonal principal desta matriz são calculados por:

$$I_{xx} = \int_v (y^2 + z^2) \rho dv$$

$$I_{yy} = \int_v (x^2 + z^2) \rho dv$$

$$I_{zz} = \int_v (x^2 + y^2) \rho dv$$

Para os outros elementos:

$$I_{xy} = \int_v xy\rho dv$$

$$I_{xz} = \int_v xz\rho dv$$

$$I_{yz} = \int_v yz\rho dv$$

Onde  $\rho$  é a densidade dos elementos infinitésimas de volume  $dv$ .

Estes valores são normalmente determinados experimentalmente. E no nosso caso, dado que o AUV é usualmente simétrico, a matriz  $I$  é diagonal.

Derivando o momento angular no ponto origem de BODY FIXED, temos:

$$\dot{H} = I_o \dot{\omega} + \omega \times (I_o \omega) = Me - \rho_g \cdot Fe$$

O termo  $-\rho_g \cdot Fe$  é o contributo dado pelas forças exteriores para os momentos devido ao facto destas forças estarem aplicadas num ponto (centro de massa), diferente da origem de BODY FIXED. Logo quando estes pontos coincidem, o contributo das forças exteriores para os momentos é nulo.

Temos pois os momentos externos dados por:

$$I\dot{\omega} + \rho_g \times m\dot{v} + m\rho_g \times \omega \times v + \omega \times (I\omega) = Me \quad (2.2)$$

As equações (2.1) e (2.2) dão-nos a descrição da hidrostática do veiculo e são ao todo 6 equações, uma para cada grau de liberdade. É usual denotar as componentes de  $Fe$  e  $Me$  por:

$$Fe = \begin{bmatrix} X_F \\ Y_F \\ Z_F \end{bmatrix}$$

$$Me = \begin{bmatrix} K_F \\ M_F \\ N_F \end{bmatrix}$$

Onde :

$X_F$	Velocidade "surge"
$Y_F$	Velocidade "sway"
$Z_F$	Velocidade "heave"
$K_F$	Momento "roll"

$M_F$	Momento "pitch"
$N_F$	Momento "yaw"

Definindo:

$$\bar{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}$$

$$\bar{\rho} = \begin{bmatrix} 0 & -\rho_{gz} & \rho_{gy} \\ \rho_{gz} & 0 & -\rho_{gx} \\ -\rho_{gy} & \rho_{gx} & 0 \end{bmatrix}$$

Obtemos a equação matricial:

$$\begin{bmatrix} \bar{M} & m\bar{\rho} \\ m\bar{\rho} & I_o \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} m(\omega \times v + \omega \times \omega \times \rho_g) \\ m(\rho_g \times \omega \times v) + \omega \times (I_o \omega) \end{bmatrix} = \begin{bmatrix} F_e \\ M_e \end{bmatrix} \quad (2.3)$$

## 2.5 Hidrodinâmica

A descrição da hidrodinâmica do AUV, é bastante complexa. É principalmente devido aos efeitos hidrodinâmicos que o controlo do veículo se torna numa tarefa difícil.

Vários efeitos hidrodinâmicos surgem com o movimento do veículo. Este ao mover-se desloca massa de água circundante, o que produz forças e momentos sobre o AUV. Além disso são induzidas ondas na água circundante, estas ondas alteram o meio onde o AUV se desloca. A diferença de posição para o centro de impulsão e o centro de gravidade causa a existência de forças e momentos de restabelecimento que tendem a levar o veículo para uma dada atitude. A variação de pressão causada pela velocidade relativa do veículo face à água causa a existência de forças de "drag" paralelas à direcção de movimento e forças de "lift" perpendiculares a esta. O movimento do AUV junto a outras superfícies dá origem a forças de sucção causadas por diminuição de pressão entre a superfície externa e a do AUV. Dado estar a mover-se imerso num fluido, o veículo sofre forças de atrito viscoso. Outros efeitos hidrodinâmicos são os devidos às características dos propulsores.

Na equação (2.3) podemos incluir estes efeitos hidrodinâmicos, vamos considerar que  $M_e$  e  $F_e$  podem ser expressas em termos das forças de propulsão, do amortecimento hidrodinâmico, das forças de restabelecimento (devidas à diferença de posições entre a aplicação da impulsão e da gravidade) e aos efeitos da massa de água deslocada pelo movimento do veículo, usualmente designados de massa acrescentada.

Se  $U_e$  for o vector de controlo que inclui  $F_e$  e  $M_e$  podemos obter:

$$U_e = u_p - M_A \dot{x} - C_A(x)x - D(x)x - D_s(x)u_s - G(\Theta) \quad (2.4)$$

Onde  $x$  é o vector estado que inclui a velocidade linear e angular (de dimensão 6).  $M_A$  é a matriz de inércia acrescentada,  $C_A$  é a matriz com os efeitos centrífugos e de Coriolis devidos à massa acrescentada.  $D$  é a matriz de amortecimento hidrodinâmico.  $G$  é o vector que inclui os efeitos restauradores da força gravítica depende da posição do veículo como a seguir se vai ver. Se o nosso veículo incluir superfícies de controlo, existe também  $u_s$  que é o vector dos angulos das superfícies e  $D_s$  que é a matriz de amortecimento hidrodinâmico para as superfícies.

Na Figura 2.2 podemos observar o efeito da diferença entre os pontos de aplicação da força da gravidade e da impulsão hidrostática em "roll".

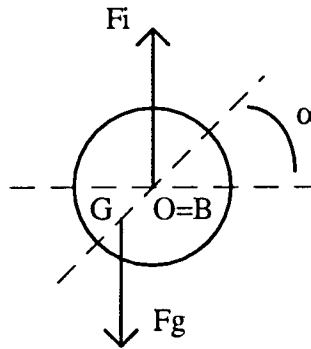


Figura 2.2

O momento sobre o eixo  $xx'$  é dado por:

$$M = -F_g z_g \text{sen}\alpha + F_i z_g \text{sen}\alpha$$

Supondo que o ponto de aplicação (B) da força de impulsão ( $F_i$ ) coincide com a origem de BODY FIXED, teremos o momento sobre o eixo  $xx'$ , dado por:

$$M = -F_g z_g \text{sen}\alpha$$

Para convertermos as forças  $F_i$  e  $F_g$  que se encontram definidas no referencial GLOBAL para o BODY FIXED, necessitamos de aplicar a matriz de transformação  $T^{-1}$ .

Note-se que o facto de sabermos que  $F_i$  e  $F_g$  se encontram na direcção vertical, isto significa que só têm componentes no eixo  $zz'$  do referencial GLOBAL.

De uma forma genérica, sem que B coincida com G teremos:

$$M = \rho_g \cdot T^{-1} \begin{bmatrix} 0 \\ 0 \\ F_g \end{bmatrix} + \rho_g \cdot T^{-1} \begin{bmatrix} 0 \\ 0 \\ -F_i \end{bmatrix}$$

Logo teremos o vector G dado por:

$$G = \begin{bmatrix} (F_g - F_i) \text{sen} \theta \\ -(F_g - F_i) \text{cos} \theta \text{sen} \phi \\ -(F_g - F_i) \text{cos} \theta \text{cos} \phi \\ -(y_G F_g - y_B F_i) \text{cos} \theta \text{cos} \phi + (z_G F_g - z_B F_i) \text{cos} \theta \text{sen} \phi \\ (z_G F_g - z_B F_i) \text{sen} \theta + (x_G F_g - x_B F_i) \text{cos} \theta \text{cos} \phi \\ -(x_G F_g - x_B F_i) \text{cos} \theta \text{sen} \phi - (y_G F_g - y_B F_i) \text{sen} \theta \end{bmatrix} \quad (2.5)$$

As matrizes  $M_A$  e  $C_A$  contêm elementos que são coeficientes hidrodinâmicos do veículo. Estes coeficientes são determinados experimentalmente.

$$M_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (2.6)$$

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & c^{15} & c^{16} \\ 0 & 0 & 0 & c^{24} & 0 & c^{26} \\ 0 & 0 & 0 & c^{34} & c^{35} & 0 \\ 0 & -c^{24} & -c^{34} & 0 & c^{45} & c^{46} \\ -c^{15} & 0 & -c^{35} & -c^{45} & 0 & c^{56} \\ -c^{16} & -c^{26} & 0 & -c^{46} & -c^{56} & 0 \end{bmatrix} \quad (2.7)$$

onde:

$$\begin{aligned}
c^{15} &= -X_{\dot{w}}u - Y_{\dot{w}}v - Z_{\dot{w}}w - Z_{\dot{p}}p - Z_{\dot{q}}q - Z_{\dot{r}}r \\
c^{16} &= X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r \\
c^{24} &= X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r \\
c^{26} &= -X_{\dot{u}}u - X_{\dot{v}}v - X_{\dot{w}}w - X_{\dot{p}}p - X_{\dot{q}}q - X_{\dot{r}}r \\
c^{34} &= -X_{\dot{v}}u - Y_{\dot{v}}v - Y_{\dot{w}}w - Y_{\dot{p}}p - Y_{\dot{q}}q - Y_{\dot{r}}r \\
c^{35} &= X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r \\
c^{45} &= -X_{\dot{r}}u - Y_{\dot{r}}v - Z_{\dot{r}}w - K_{\dot{r}}p - M_{\dot{r}}q - N_{\dot{r}}r \\
c^{46} &= X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r \\
c^{56} &= -X_{\dot{p}}u - Y_{\dot{p}}v - Z_{\dot{p}}w - K_{\dot{p}}p - K_{\dot{q}}q - K_{\dot{r}}r
\end{aligned} \tag{2.8}$$

Os efeitos do atrito viscoso podem ser descritos aproximadamente como proporcionais ao quadrado da velocidade. Teremos pois que a matriz D terá elementos que são coeficientes hidrodinâmicos que dependem do quadrado da velocidade (respectiva a cada elemento).

## 2.6 Actuadores

Como actuadores no AUV, temos as superfícies de controlo, os propulsores de controlo e os propulsores principais.

Geralmente considera-se a dinâmica das superfícies de controlo como desprezável. Apenas se inclui o efeito hidrodinâmico das superfícies no comportamento do veículo. Basicamente temos vários coeficientes hidrodinâmicos que vão afectar várias variáveis de estado. A influência das superfícies, faz-se sentir dado irem afectar os termos de "drag" e de "lift". A força de "drag" depende da área lateral do veículo que se encontra exposta ao fluido na direcção do movimento. O "drag" tende a contraria o movimento do veículo. A força de "lift" é perpendicular á direcção de movimento e também depende da atitude do veículo.

Ao incluirmos superfícies estamos a aumentar o "drag" já que a sua área vai-se somar à área lateral do veículo. Podemos obter por exemplo de uma forma simplificada a expressão da força de "surge" (movimento "para a frente"), como:

$$X_F = X_{u|u|} \cdot u|u| + F_{prop} + X_{\delta_r \delta_r} \delta_r^2$$

Onde  $X_{u|u|}$  é o termo de drag devido á superfície lateral do veículo,  $X_{prop}$  é a força de impulsão devida aos motores e  $X_{\delta_r \delta_r} \delta_r^2$  é o "drag" acrescentado pelo leme do AUV.

Podemos observar na expressão seguinte a descrição detalhada do valor de  $X_F$  para o caso de termos dois lemes um de "stern" (leme de profundidade) e um "rudder" (direcção).

$$\begin{aligned}
 X_F = & m(v \cdot r - w \cdot q + X_q(q^2 + r^2) + m((-y_g \cdot p \cdot q) - z_g \cdot p \cdot r) \\
 & + X_{pp} \cdot p^2 + X_{qq} \cdot q^2 + X_{rr} \cdot r^2 + X_{pr} \cdot p \cdot r + X_{wp} \cdot w \cdot q + \\
 & X_{vp} \cdot v \cdot p + X_{vr} \cdot v \cdot r - (F_g - F_i) \cdot \text{sen}\theta - X_{u|u} \cdot u \cdot |u| + u \cdot q \cdot (X_{q\delta_s} \cdot \delta_s) \\
 & + X_{\delta_s\delta_s} \cdot \delta_s^2 + X_{\delta_r\delta_r} \cdot \delta_r^2 \cdot u \cdot |u| + F_{prop}
 \end{aligned}$$

O problema de descrever a força de propulsão originada pelos motores, é complexo devido aos efeitos hidrodinâmicos existentes no fluido que atravessa as hélices. O movimento do fluido junto dos motores é turbulento e logo a sua descrição é de maior dificuldade. Note-se que nas descrições dos efeitos hidrodinâmicos acima referidos, pelo uso de coeficientes que são em geral determinados experimentalmente, faz-se uma aproximação quanto ao comportamento hidrodinâmico do veículo.

Se as hélices se encontrarem dentro de um tubo, os efeitos de turbulência são acrescidos.

A força e momento causados pelos propulsores pode ser dada por:

$$u_p = b(x, n)$$

Onde  $x$  é o estado e  $n$  é a velocidade de rotação do motor. A função  $b(\cdot)$  é uma função não linear. Pode-se observar em [Fossen 91] um estudo detalhado desta função.

No nosso modelo, vamos desprezar os efeitos hidrodinâmicos dos propulsores face à hidrodinâmica total do veículo.

## 2.7 Sensores

O AUV, pode ser equipado com inúmeros tipos de sensores. Podemos ter sonares para efectuar reconhecimento e navegação (por exemplo pode-se determinar a posição do AUV por triangulação numa zona delimitada por balizas). É necessária a existência de sensores de velocidade angular e de atitude para medir as variáveis de estado do veículo. O problema mais difícil diz respeito à medida das velocidades lineares ( $u$ ,  $v$ ,  $w$ ). Normalmente só se dispõe de sensores que medem a velocidade do veículo em relação à água. Isto significa que o efeito das correntes pode afectar significativamente os valores medidos. Além disso os sensores de velocidade em relação à água têm normalmente uma "dead zone", zona de velocidades para os quais não medem a velocidade. Isto acontece a muito baixas velocidades, e uma vez que o AUV pode deslocar-se a baixas velocidades, esta não-linearidade pode ser significativa. Uma alternativa são os sensores de efeito de Doppler que medem a velocidade do veículo relativamente

ao fundo oceânico. O seu problema é além do seu elevado preço, a sua dependência face ao tipo de fundo que pode introduzir erros nas leituras e a falta de robustez nas medidas originada por esse motivo.

Não vamos derivar aqui as equações da dinâmica dos vários sensores, já que ou a dinâmica pode-se desprezar, ou se não for caso disso o controlo deve incluir suficiente robustez para não ser significativamente afectado pela dinâmica dos sensores. Vamos admitir que temos sempre disponível a informação do estado do veículo. Tal pode não ser necessariamente verdade, mas pode-se sempre dispor de estimadores de estado que com maior ou menor qualidade nos podem fornecer uma estimativa do estado do AUV. Os erros existentes nesta estimativa devem ser tratados pelo controlo do veículo, que deve dispor de robustez necessária para não só eliminar os efeitos dos erros de estimativa de estado, como também os efeitos do desconhecimento exacto da planta e das variações dos parâmetros desta.

## 2.8 Resumo do modelo

Nas secções anteriores estivemos a analisar a modelização de um AUV sob diversos aspectos, vamos aqui apresentar as equações de um modelo para o veículo.

Já foi estabelecido que o estado do veículo tem 6 componentes:

$$x = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}$$

Estas componentes são as componentes da velocidade linear e angular do veículo no seu referencial (BODY FIXED).

As equações (2.3) e (2.4) dão-nos a hidrostática e a hidrodinâmica do AUV. Na equação (2.4) temos os termos  $u_p$  e  $u_s$  que são os vectores de controlo (motores propulsores e superfícies de controlo respectivamente).

Podemos passar alguns termos de  $U_e$  para o lado esquerdo da equação (2.3):

$$M_{est} \dot{x} + C_{est} x + M_A \dot{x} + C_A x + D(x) x + G(\Theta) x = u_p - D_s u_s \quad (2.9)$$

As matrizes  $M_A$  e  $C_A$  são as indicadas na equação (2.4), a matriz  $M_{est}$  corresponde à matriz existente em (2.3) que está a multiplicar pela derivada do estado e  $C_{est}$  corresponde à matriz composta pelos termos que multiplicam pelo estado na equação (2.3). O termo  $C_{est} x$  é a expansão da segunda matriz na equação (2.3).

A matriz  $M_{est}$  é dada por:



$$M_{est} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{xz} \\ mz_G & 0 & mx_G & -I_{xy} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{xz} & -I_{yz} & I_z \end{bmatrix}$$

e  $C_{est}$ :

$$C_{est} = \begin{bmatrix} 0 & -mr & mq & m(y_G q + z_G r) & -mx_G q & -mx_G r \\ mr & 0 & -mp & -my_G p & m(z_G r + x_G p) & -my_G r \\ -mq & mp & 0 & -mz_G p & -mz_G q & m(x_G p + y_G q) \\ -m(y_G q + z_G r) & my_G p & mz_G p & 0 & -I_{yz} q - I_{xz} p + I_x r & I_{yz} r + I_{xy} p - I_y q \\ mx_G & -m(z_G r + x_G p) & mz_G q & I_{yz} q + I_{xz} p - I_x r & 0 & -I_{xz} r - I_{xy} q + I_y p \\ mx_G r & my_G r & -m(x_G p + y_G q) & -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_y p & 0 \end{bmatrix}$$

A matriz MA é a matriz de massa acrescentada definida em (2.6). A matriz CA de efeitos de Coriolis e centrífugos CA pode ser [Fossen 1991] simplificada a partir de (2.7) e (2.8) para:

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}} w & Y_{\dot{v}} v \\ 0 & 0 & 0 & Z_{\dot{w}} w & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & -Y_{\dot{v}} v & X_{\dot{u}} u & 0 \\ 0 & -Z_{\dot{w}} w & Y_{\dot{v}} v & 0 & -N_{\dot{r}} r & M_{\dot{q}} q \\ Z_{\dot{w}} w & 0 & -X_{\dot{u}} u & N_{\dot{r}} r & 0 & -K_{\dot{p}} p \\ -Y_{\dot{v}} v & X_{\dot{u}} u & 0 & -M_{\dot{q}} q & K_{\dot{p}} p & 0 \end{bmatrix}$$

A matriz D contém os efeitos de atrito viscoso (efeitos de "drag"). Experimentalmente conclui-se que estes efeitos podem ser descritos com uma boa aproximação por um termo linear e um termo quadrático de drag. Devido a ser difícil calcular, e de se poder na maior parte das aplicações, ignorar os termos fora da diagonal principal de D (termos cruzados), a matriz D é usualmente descrita como sendo uma matriz diagonal. Estes elementos podem-se determinar experimentalmente e logo a matriz D toma a forma:

$$D(x) = \begin{bmatrix} X_u + X_{|u|}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{|v|}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + Z_{|w|}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + K_{|p|}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + M_{|q|}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{|r|}|r| \end{bmatrix}$$

A matriz  $D_s$  com os termos de "drag" para as superfícies de controlo, toma também uma forma semelhante, contendo os coeficientes hidrodinâmicos lineares e quadráticos para cada superfície.

Podemos também agrupar  $M_{est}$  e  $MA$  numa única matriz  $M$ , e da mesma forma teremos  $C_{est}$  e  $CA$  agrupadas em  $C$ . Assim a equação (2.9) fica:

$$M\dot{x} + Cx + D(x)x + G(\Theta)x = u_p - D_s u_s \quad (2.10)$$

onde  $G$  é dada por (2.5),  $D$  foi descrita acima  $M$  é dada por:

$$M = \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{w}} & -X_{\dot{p}} & mz_G - X_{\dot{q}} & -my_G - X_{\dot{r}} \\ -X_{\dot{v}} & m - Y_{\dot{v}} & -Y_{\dot{w}} & -mz_G - Y_{\dot{p}} & -Y_{\dot{q}} & mx_G - Y_{\dot{r}} \\ -X_{\dot{w}} & -Y_{\dot{w}} & m - Z_{\dot{w}} & my_G - Z_{\dot{p}} & -mx_G - Z_{\dot{q}} & -Z_{\dot{r}} \\ -X_{\dot{p}} & -mz_G - Y_{\dot{p}} & my_G - Z_{\dot{p}} & I_x - K_{\dot{p}} & -I_{xy} - K_{\dot{q}} & -I_{xz} - K_{\dot{r}} \\ mz_G - X_{\dot{q}} & -Y_{\dot{q}} & mx_G - Z_{\dot{q}} & -I_{xy} - K_{\dot{q}} & I_y - M_{\dot{q}} & -I_{yz} - M_{\dot{r}} \\ -my_G - X_{\dot{r}} & mx_G - Y_{\dot{r}} & -Z_{\dot{r}} & -I_{xz} - K_{\dot{r}} & -I_{yz} - M_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix}$$

e  $C$  por:

$$C = \begin{bmatrix} 0 & -mr & mq & m(y_c q + z_c r) & -mx_c q - Z_w w & -mx_c r - Y_v v \\ mr & 0 & -mp & -my_c p + Z_w w & m(z_c r + x_c p) & -my_c r - X_u u \\ -mq & mp & 0 & -mz_c p - Y_v v & -mz_c q + X_u u & m(x_c p + y_c q) \\ -m(y_c q + z_c r) & my_c p - Z_w w & mz_c p + Y_v v & 0 & -lyzq - lxp + lxr - N_r & lyzr + lxp - lyq + M_q q \\ mx_c + Z_w w & -m(z_c r + x_c p) & mz_c q - X_u u & lyzq + lxp - lxr + N_r & 0 & -lxz - lxyq + lxp - K_p p \\ mx_c r - Y_v v & my_c r + X_u u & -m(x_c p + y_c q) & -lyzr - lxp + lyq - M_q q & lxr + lxyq - lxp + K_p p & 0 \end{bmatrix}$$

No lado direito da equação (2.10) temos o controlo. Vamos supor a existência no nosso modelo de 3 superfícies de controlo (3 ângulos independentes, não necessariamente só 3 superfícies), "stern", "rudder" e "spin". A primeira controla o ângulo de ataque do AUV (rotação sobre o eixo  $yy'$ ), a segunda é o leme de direcção (rotação sobre o eixo  $zz'$ ) e a última controla o "roll" do veículo (constituída por lemes de profundidade de ambos os lados do veículo e com ângulos simétricos, faz a rotação sobre o eixo dos  $xx'$ ). Vamos também considerar a existência de 6 motores para controlo e propulsão do AUV, estes motores supõem-se dispostos simetricamente aos eixos de simetria do AUV (eixos de BODY FIXED).

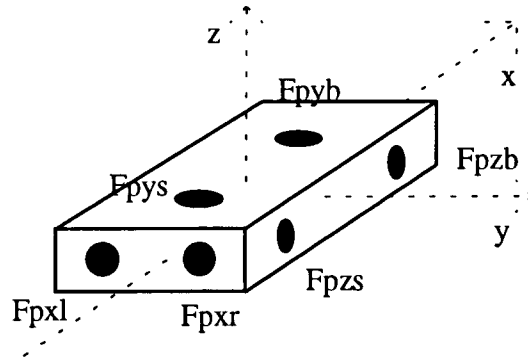


Figura 2.3

Dado ser algo complicada a modelização dos motores (ver [Silvestre 91] ) vamos supor já como controlo disponível as forças provocadas por cada motor. Dado que  $D_s$  vai depender também do quadrado do vector de controlo, vamos exprimir o lado direito de (2.10) por duas matrizes  $B_1$  e  $B_2$  que vão multiplicar o controlo e o quadrado do controlo respectivamente.

Teremos pois como vector controlo:

$$\mathbf{u} = \begin{bmatrix} F_{pxl} \\ F_{pxr} \\ F_{pys} \\ F_{pyb} \\ F_{pzs} \\ F_{pzb} \\ \delta_{sp} \\ \delta_s \\ \delta_r \end{bmatrix}$$

Onde os índices "l" e "r" nas forças indicam motor esquerdo e direito respectivamente, e "s" e "b" (de "stern" e "bow"), indicam traseiro e dianteiro respectivamente. Nas superfícies, "sp" é a de "spin", "s" é a de "stern" (leme de profundidade) e "r" a de "rudder" (leme de direcção).

Ficamos pois com o modelo final do AUV dado por:

$$M\dot{x} + Cx + D(x)x + G(\Theta)x = B_1u + B_2u^2 \quad (2.10)$$

sendo  $l_x$ ,  $l_y$  e  $l_z$  as distâncias dos motores ao eixo respectivo, temos  $B_1$  dada por:

$$B_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & X_{\delta_{sp}} & X_{\delta_s} & X_{\delta_r} \\ 0 & 0 & 1 & 1 & 0 & 0 & Y_{\delta_{sp}} & Y_{\delta_s} & Y_{\delta_r} \\ 0 & 0 & 0 & 0 & 1 & 1 & Z_{\delta_{sp}} & Z_{\delta_s} & Z_{\delta_r} \\ l_x & l_x & 0 & 0 & 0 & 0 & K_{\delta_{sp}} & K_{\delta_s} & K_{\delta_r} \\ 0 & 0 & l_y & l_y & 0 & 0 & M_{\delta_{sp}} & M_{\delta_s} & M_{\delta_r} \\ 0 & 0 & 0 & 0 & l_z & l_z & N_{\delta_{sp}} & N_{\delta_s} & N_{\delta_r} \end{bmatrix}$$

e  $B_2$  é dada por:

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & X_{\delta^2_{sp}} & X_{\delta^2_s} & X_{\delta^2_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & Y_{\delta^2_{sp}} & Y_{\delta^2_s} & Y_{\delta^2_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & Z_{\delta^2_{sp}} & Z_{\delta^2_s} & Z_{\delta^2_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{\delta^2_{sp}} & K_{\delta^2_s} & K_{\delta^2_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & M_{\delta^2_{sp}} & M_{\delta^2_s} & M_{\delta^2_r} \\ 0 & 0 & 0 & 0 & 0 & 0 & N_{\delta^2_{sp}} & N_{\delta^2_s} & N_{\delta^2_r} \end{bmatrix}$$

Devemos notar que o modelo aqui apresentado, já contém bastantes simplificações, nomeadamente na consideração dos efeitos hidrodinâmicos, em que são feitas aproximações. No entanto constitui uma boa aproximação para o comportamento do veículo em termos práticos.

### **3. Conclusões e Perspectivas Futuras**

Esta modelização de um veículo submarino autónomo, permite efectuar simulações e testar vários paradigmas de controlo, usando como base o modelo aqui proposto.

Pensamos no entanto que este trabalho constitui um bom ponto de partida para estudos futuros no campo aliciante do controlo de AUV's e que algumas ideias aqui avançadas podem ser promissoras no futuro.

## 4. Bibliografia

- [Fossen 91] Thor Inge Fossen, "*Nonlinear modelling and control of underwater vehicles*", Ph. D. Disert., Norwegian Institute of Technology, June 1991.
- [Fossen 93] Thor Inge Fossen, "*An introduction to nonlinear modelling, stability and control of underwater vehicles*", Tech. Report. , Norwegian Institute of Technology, 1993.
- [Slotine 91] Jean-Jacques E. Slotine, Weiping Li, "*Aplied Nonlinear Control*", Prentice Hall, 1991
- [Maciej 89] J. M Maciejowski, "*Multivariable feedback design*", Addison Wesley, 1989
- [Cristi 90] Roberto Cristi, Fotis Papoulias, Anthony Healey, "*Adaptive sliding mode control of autonomous vehicles in the dive plane*", IEEE Journ. Oceanic Eng., vol 15, no3, Jul. 1990
- [Almeida 93] Alfredo Martins, Jose Almeida, "*Redes neuronais em controlo - um estado da arte*", 1993
- [Silvestre 91] Carlos Silvestre, "*Modelação e controlo de veículos submarinos autónomos*", Tese de Mestrado, IST, Junho de 1991
- [Ljung 87] Lennart Ljung, "*System identification*", Prentice Hall, 1987

**Estado da arte das Redes Neurais e as suas aplicações em  
Controlo**

**ISR-PORTO**

Introdução .....	3
Arquitecturas .....	5
Elementos básicos .....	5
NÃO RECORRENTES .....	6
MULTILAYER FF .....	6
MAPA KOHONEN .....	7
CMAC .....	8
RADIAL BASIS FUNCTION NETWORKS .....	9
RECORRENTES .....	10
HOPFIELD .....	10
COUNTERPROPAGATION .....	10
Aprendizagem .....	12
SUPERVISIONADA .....	12
BACKPROPAGATION .....	12
HOPFIELD .....	13
BOLTZMAN MACHINE .....	14
CMAC .....	14
ALGORITMOS GENÉTICOS .....	14
COUNTERPROPAGATION .....	16
RADIAL BASIS FUNCTION NETWORKS .....	16
NÃO SUPERVISIONADA .....	17
COMPETITIVA .....	17
KOHONEN .....	17
Aplicações em controlo .....	18
IDENTIFICAÇÃO .....	18
CONTROLO ADAPTATIVO .....	22
CONTROLO ÓPTIMO .....	27
DETECÇÃO DE FALHAS .....	27
CONTROLO HIERÁRQUICO .....	29
EXPERT SYSTEMS .....	29
OUTROS MÉTODOS DE CONTROLO .....	30
Bibliografia .....	32



## Introdução

As redes neuronais são hoje em dia uma ferramenta válida para o controlo e identificação de sistemas. Neste trabalho vamos apresentar um estudo sobre a utilização de redes neuronais em identificação e controlo. Não vamos fazer um estudo exaustivo da situação actual do tema redes neuronais. Vamo-nos apenas cingir ao estudo destas em aplicações de controlo.

O estudo das redes neuronais desenvolveu-se nos últimos anos da década de oitenta com o artigo de Rumelhart [Rumelhart 1986], isto após quase duas décadas de desinteresse. Este campo de investigação iniciou-se na década de quarenta com o objectivo de simular o funcionamento do cérebro humano. Após vários desenvolvimentos na área, esta estagna graças ao estudo efectuado por Minsky e Paper [Minsky 1969]. Neste eram levantadas questões quanto à classe de problemas que poderiam ser resolvidos por redes neuronais e quanto à necessidade da existência, de mais de uma camada para a resolução de problemas complexos. No entanto por esta altura não eram conhecidos métodos eficientes de treino para redes com várias camadas. Foi graças a Rumelhart (que popularizou um método apresentado por Werbos [Werbos 1974] ), que se passou a poder treinar redes com camadas escondidas, levando a investigação nesta área a um grande desenvolvimento. Isto após um artigo de Hopfield onde eram enunciadas as possibilidades das redes neuronais no campo da optimização [Hopfield 1982].

O uso de redes neuronais (doravante designadas como RN) para controlo teve especialmente um grande desenvolvimento também na década de oitenta. Inicialmente, e como as RN tentavam simular o funcionamento do cérebro, as funções de controlo muscular deste, eram também estudadas. Na época do nascimento das RN, a teoria de controlo tinha grandes desenvolvimentos, nomeadamente o controlo de sistemas lineares com o desenvolvimento dos métodos de resposta em frequência e do lugar de raízes nas décadas de 40 e 50. Nos fins da década de 50 métodos como o espaço de estados proporcionavam o estudo de sistemas com múltiplas entradas e múltiplas saídas ( MIMO) [Ogata 1970]. Nesta época o estudo dos sistemas lineares era muito activo ,e existindo, boas técnicas para análise de problemas deste tipo, o a necessidade das RN não era premente.

Com o incremento do estudo de sistemas não-lineares e a não existência de métodos eficientes para este tipo de problemas, as características das RN tornavam-nas muito atraentes. No entanto os problemas básicos de aprendizagem ainda não tinham sido resolvidos, pelo que só recentemente com a explosão do estudo das RN, é que a sua aplicação no controlo levou também um forte desenvolvimento.

Os motivos que levam a aplicação das RN em controlo são a sua robustez, adaptabilidade e a capacidade de efectuarem mapeamentos não-lineares e logo a sua aplicabilidade ao controlo de sistemas não-lineares.

Este estudo encontra-se dividido em três partes. Na primeira são apresentadas as diversas arquitecturas de redes neuronais, existentes e com aplicações a controlo. Na segunda parte são apresentados os diversos métodos de treino das redes (métodos estes que normalmente estão associados a uma certa arquitectura). Na ultima parte são enunciadas aplicações em controlo, nomeadamente a forma como as RN são usadas em identificação e controlo de sistemas.

## Arquitecturas

### Elementos básicos

A rede neuronal é constituída por unidades básicas de processamento [Lippman 1988], aqui designadas por neurónios. Cada neurónio possui várias entradas a que estão associados pesos, um somador que efectua a soma pesada das entradas e à saída deste uma função não-linear (normalmente o degrau ou uma versão diferenciável deste, que é o sigmoide) Fig 1.

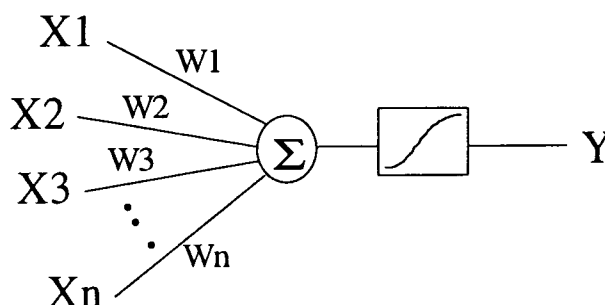


Figura 1

As redes contêm vários neurónios ligados por forma às saídas de uns estarem ligadas às entradas de outros.

O uso das RN compreende duas fases: o treino e a execução. Na primeira a rede é treinada por forma aos pesos serem ajustados de forma a satisfazerem um dado objectivo. Na segunda a rede é usada para cumprir a função que lhe foi destinada. Note-se que também a rede pode ser treinada on-line. É geralmente a primeira fase que demora mais tempo dada a sua natureza iterativa. Na segunda os cálculos são bastante simples pelo que esta fase é bastante mais rápida.

Dado serem constituídas por muitos elementos de pequena capacidade de cálculo, as RN são modelizadas favoravelmente por sistemas massivamente paralelos. E o campo dos computadores massivamente paralelos tem tido forte desenvolvimento recentemente [Zorpette1992]. Existindo já hoje em dia hardware destinado a redes neuronais [Eberhart 1990] o que facilita o seu uso e aplicação.

Nós consideramos como redes recorrentes aquelas em que, de alguma forma, o sinal que sai de um dado neurónio influencia a sua entrada, i.e. existe feedback. Não só são recorrentes as redes onde a saída liga à entrada, mas também, as com ligações laterais, onde a saída de um neurónio é também uma sua entrada.

## NÃO RECORRENTES

### MULTILAYER FF

A primeira arquitectura de redes que apareceu, foi a rede só com uma camada de neurónios [Roseblatt 1967], [Widrow 1960]. A "multilayer feedforward" [Rumelhart 1986], [Werbos 1974], ou seja rede com várias camadas, é uma extensão da rede só com uma camada.

Este tipo de redes permite resolver certo tipo de problemas que a rede com uma única camada não consegue [Minsky 1969], [Nguyen 1990]. Uma rede com uma camada de neurónios, classifica as suas entradas em várias classes de saída (combinações possíveis para as saídas). Quando o problema a resolver não é linearmente separável, i.e. as classes de saída não podem ser obtidas por separação do espaço de entrada, por um único hiperplano, o problema não é resolúvel por redes de uma única camada (limitação demonstrada por [Minsky 1969]). Um caso clássico deste tipo de problemas é o do XOR (Fig. 2), com uma camada não é possível implementar esta função por uma RN.

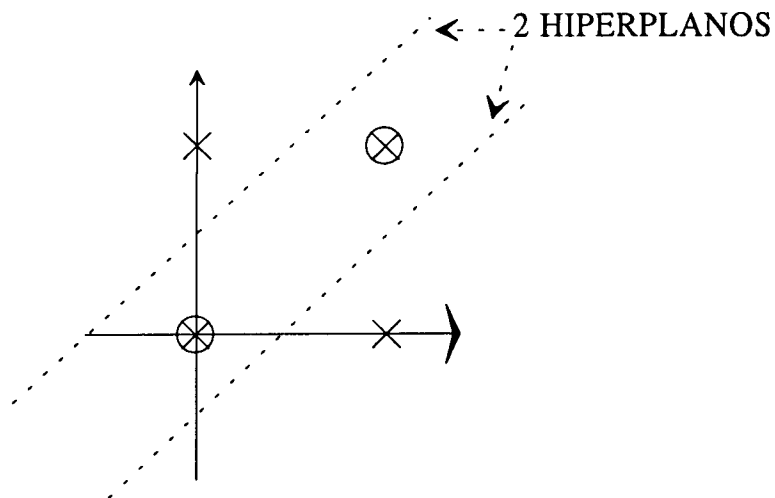


Figura 2

Ao introduzirmos mais uma camada, as "partes" aproximadas da função completa, obtidas pelos neurónios da 1ª camada, são combinadas para formar a aproximação completa [Nguyen 1990]. Em [Hornik 1989], [Cybenko 1989] pode-se encontrar estudos sobre a capacidade de aproximação das redes neuronais.

Nas redes multilayer feedforward, os neurónios estão organizados em várias camadas (Fig. 3).

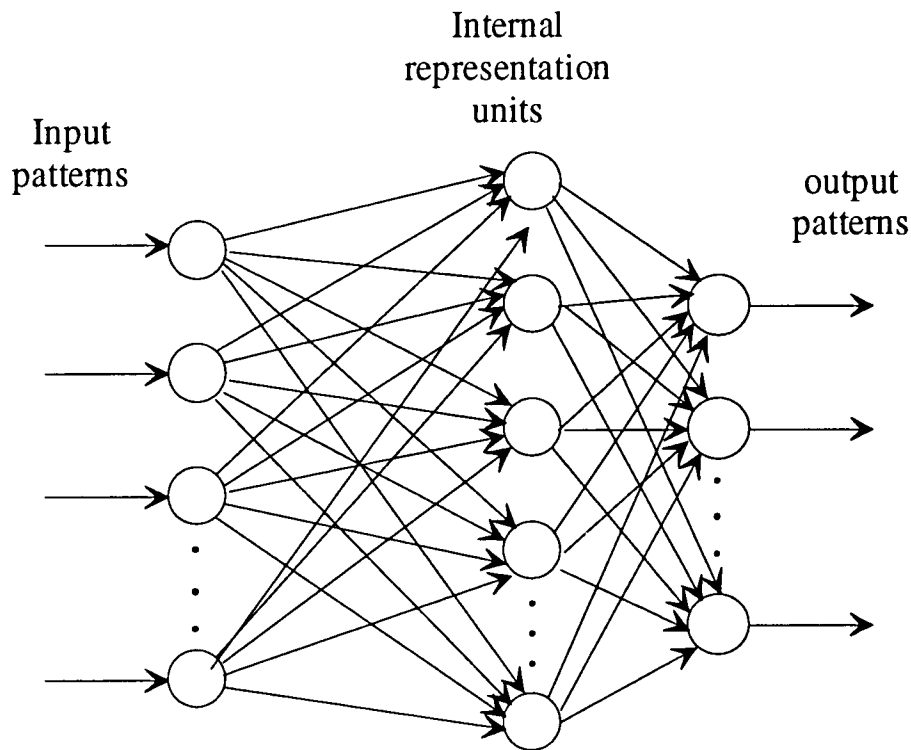


Figura 3.

Nestas as saídas dos neurónios de uma camada ligam às entradas dos neurónios da camada seguinte (exceptuando obviamente as camadas de saída e de entrada). Não existe "feedback", i.e. não há ligação entre as saídas de uma camada posterior (mais próxima da saída) e as entradas de uma camada anterior. Por isso este tipo de redes é classificada como de não-recorrentes.

Apesar de se saber que redes com uma camada escondida resolvem problemas convexos, e que redes com duas ou mais camadas escondidas resolvem problemas de complexidade arbitrária [Almeida 1990a], não existem resultados teóricos sobre o número de camadas e neurónios ideal para a resolução de um dado problema (existem alguns estudos sobre o assunto tais como [Hornik 1989], [Cybenko 1989]).

O método para o ajuste dos pesos para estas rede é normalmente o " error backpropagation" que será descrito mais adiante.

### MAPA KOHONEN

Os "self-organizing maps" foram descritos inicialmente por [Kohonen 1987]. Estes usam uma malha bidimensional de nós [Kohonen 1987], [Kohonen 1990], [Lippman 1987] (Fig. 4), onde cada nó representa o centro de um agrupamento ("cluster") dos dados de treino.

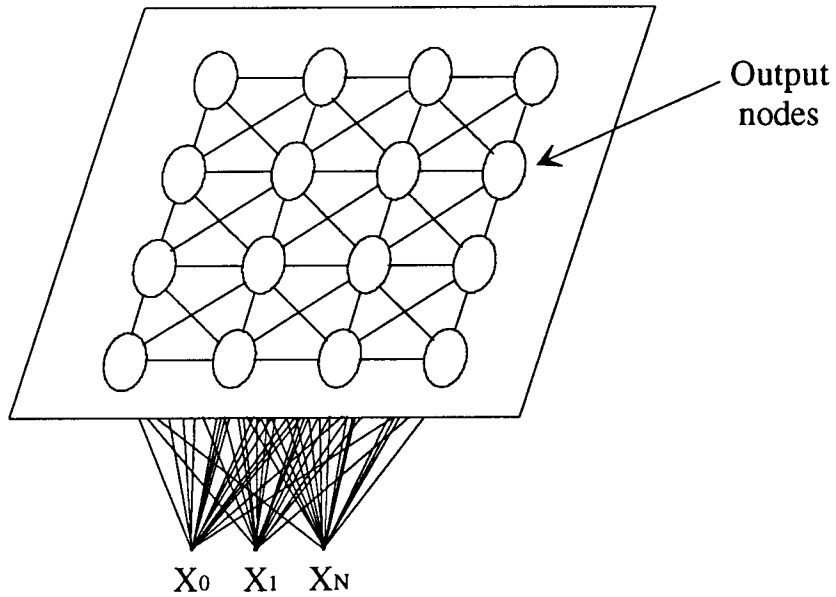


Figura 4

Assim após o treino, apenas o agrupamento com o centro mais próximo da entrada tem nível alto (está activo). Assim temos que os dados de entrada são quantizados no numero de neurónios existentes. É possível utilizarem-se várias camadas de neurónios hierarquicamente por forma a termos diversos graus de quantização. Assim a um dado conjunto de neurónios na primeira camada corresponderia um na segunda, que faria um agrupamento a um nível mais elevado. O método de ajustamento dos pesos é apresentado na secção seguinte.

### CMAC

Neste tipo de redes (inicialmente proposto por [Albus 1975]) usa-se o conceito de aproximação local. Têm neurónios que à entrada possuem campos receptivos, i.e. existem funções que mapeiam as entradas possíveis para os pesos. Estas funções só tomam o valor de um para um dado subconjunto dos dados de entrada (Fig. 5).

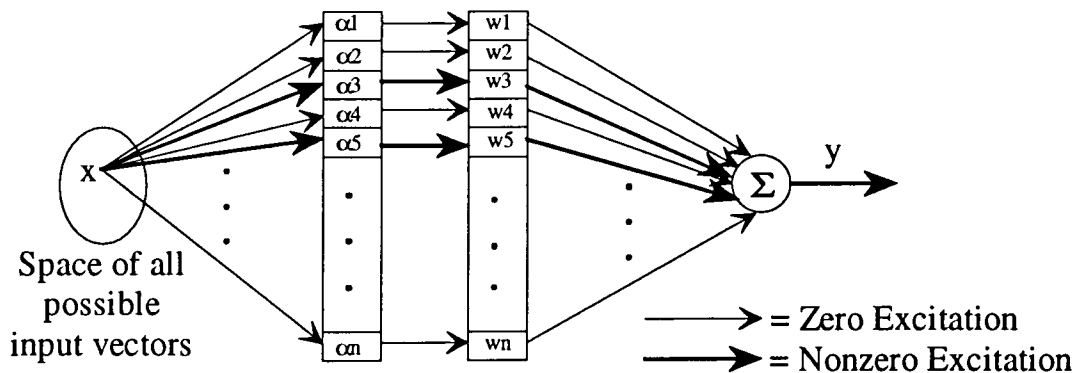


Figura 5

Estas funções podem ser quer rectangulares ou de ordem mais elevada como B-splines, as vantagens do uso deste tipo de funções podem ser encontradas em [Lane 1992].

### RADIAL BASIS FUNCTION NETWORKS

Este tipo de redes não é constituído pelos neurónios básicos, apresentados no início desta secção. Pode-se dizer que contém só duas camadas: uma contendo "radial basis units" e outra contendo somadores. Só para esta é que existem pesos (ver Fig 6).

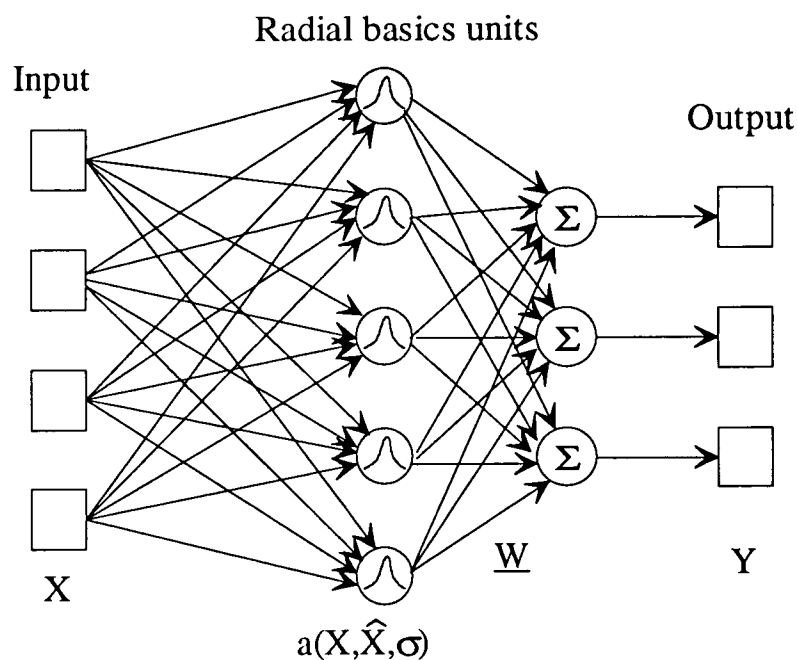


Figura 6

As funções das "radial basis units" ao contrário das multilayer networks, são não monotónicas, e as entradas são apresentadas directamente sem pesos.

A vantagem deste tipo de redes é um particionamento do espaço de entrada mais eficiente [Leonard 1991].

Na sua aprendizagem descrita adiante, não só são calculados pesos, mas também o "centro" e variância das funções das "radial basis units" (normalmente estas funções são gaussianas pelo que teremos que calcular a média e a variância).

O particionamento do espaço de entrada é conseguido por escolha das funções da primeira camada de modo a formarem fronteiras de decisão.

Estas redes são obviamente não-recorrentes dado não existir nenhum tipo de "feedback".

## RECORRENTES

### HOPFIELD

No campo das redes recorrentes, o modelo mais vulgarizado é o apresentado por Hopfield em [Hopfield 1982]. Este artigo veio chamar a atenção para as RN na década de oitenta e juntamente com [Rumelhart 1986] despoletou o interesse pela área. Este tipo de redes está essencialmente vocacionado para optimização (Hopfield apresentou uma aplicação das redes ao problema do caixeiro viajante, em [Tagliarini 1991] temos um caso de optimização combinatoria apresentado de uma forma bastante geral). Existem também aplicações em controlo tais como identificação de sistemas [Chu 1990] que serão apresentadas mais adiante.

Na rede de Hopfield todos os neurónios encontram-se ligados a todos (Fig 7).

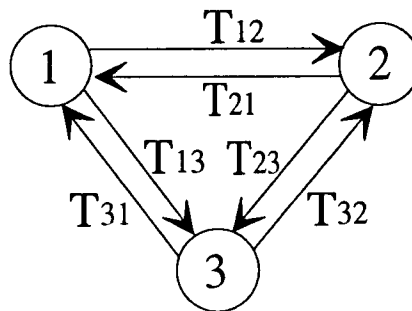


Figura 7

Esta rede é o caso mais conhecido de "content adressable memory" (CAM). Aqui são os estados dos neurónios (normalmente a rede de Hopfield é binária, ou seja, os neurónios ou estão activos ou inactivos, embora possa existir redes contínuas [Chu 1990], [Hopfield 1984]), que significam a saída da rede. Ao apresentar-se um dado padrão de entrada à rede, o que significa um conjunto de estados dos neurónios, a rede "decai" para um outro padrão que foi inicialmente treinado. Este constitui um mínimo local de uma função de energia [Hopfield 1982], [Hopfield 1984] que Hopfield [Hopfield 1982] provou que era minimizada (localmente), pela rede. O método de actualização dos pesos é apresentado na secção seguinte.

### COUNTERPROPAGATION

Esta arquitectura usa dois tipos diferentes de camadas de neurónios. Foi apresentada por Hetch-Nielsen [Hetch-Nielsen 1987] para tomar partido da arquitectura paralela das redes neuronais. Este tipo de redes usa a ideia de codificar as entradas [Hunt 1992] (pela camada competitiva). Primeiro uma com uma camada competitiva, e depois uma camada de saída. Cada camada está ligada a seguinte com todas as ligações possíveis. (Fig 8).



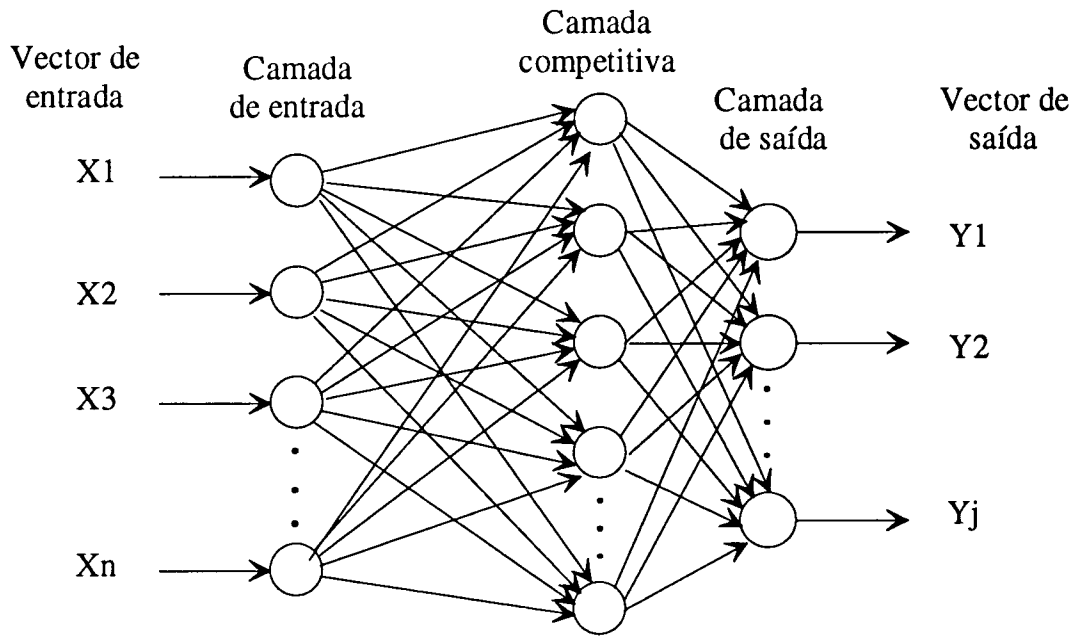


Figura 8

As unidades da camada escondida competem por responder à entrada, a unidade que vence a competição que influencia os valores na camada de saída, os outros neurónios que perdem a competição não influenciam a saída. Existem pesos quer na camada de competição quer na de saída. Este tipo de comportamento dá a este tipo de redes, características de memória associativa [Dayhoff 1990]. Combina características da rede de neurónios tipo feedforward e as características dos mapas de Kohonen.

## Aprendizagem

Considera-se aqui que aprendizagem supervisionada é aquela onde alguém treina a rede com um dado objectivo. Isto inclui não só o caso de backpropagation mas também o das redes de Hopfield [Lippman 1987], [Kahna 1990], apesar de nestas não termos pares entrada-saída. Mas nas últimas, o utilizador ao apresentar as entradas, está no fundo a dizer para que padrões quer que a rede "caia".

### SUPERVISIONADA

#### BACKPROPAGATION

Este é o método mais usado para treino de redes neuronais. Numa forma mais básica, aplica-se a redes multilayer feedforward. É uma extensão do método apresentado por Widrow e Hoff (regra delta) [Widrow 1960] para uma única camada de neurónios. Neste método os pesos eram ajustados de acordo com o erro na saída dos neurónios. Este erro era a diferença entre a saída obtida e a saída desejada. Ao erro era multiplicado um coeficiente que consistia no passo do algoritmo de aprendizagem.

Para a aprendizagem de redes com múltiplas camadas não existia na altura da regra delta nenhum método eficiente para treino. Foi Werbos que em 1974 [Werbos 1974], desenvolveu bases teóricas para um método de cálculo de derivadas parciais ordenadas, que deu origem mais tarde à backpropagation apresentada por Rumelhart [Rumelhart 1986].

A ideia básica da backpropagation é actualizar os pesos da última camada de acordo com a regra delta e para as outras camadas ir propagando o erro para trás. Ou seja, da saída para a entrada. O erro é propagado para trás na rede indo alterar os pesos de acordo com a influência destes na saída [Rumelhart 1986], [Almeida 1990b]. Este erro é passado para trás, sendo calculado para cada neurónio nas camadas escondidas. Depende dos erros nos neurónios da camada seguinte, dos pesos que ligam o neurónio em questão à camada seguinte e da derivada da função não-linear do neurónio (daí a necessidade que esta seja diferenciável e logo a utilização do sigmoide em vez do degrau). Pode-se encontrar a expressão desta regra de aprendizagem, também chamada regra delta generalizada, em [Rumelhart 1986], [Hopfield 1982], [Khanna 1990], [Dayhoff 1990], [Lippmann 1987].

A propagação deste erro para trás, ou seja o cálculo das derivadas parciais do erro final em ordem aos pesos [Werbos 1974], [Rumelhart 1986], pode ser feito com uma outra rede neuronal (chamada de rede de sensibilidade em [Narendra 1990], [Narendra 1991b] ou rede de propagação de erro [Almeida 1990a]) que vai calculando estas derivadas. Esta rede tem os mesmos pesos que a outra e possui como funções não-lineares nos neurónios, as derivadas das funções na outra rede [Almeida 1990a], [Narendra 1990], [Narendra 1991b].

O ajuste dos pesos pode ser feito em batch ou imediatamente [Porto 1992], [Almeida 1990a] isto corresponde a actualizar os pesos ao fim da apresentação de um grupo de padrões (chamado método do

gradiente estocástico), em função do erro do conjunto, ou a, actualizar os pesos a seguir à apresentação de cada padrão. Algumas optimizações ao método de backpropagation foram propostas. Nomeadamente a introdução de termos de momento [Rumelhart 1986] (semelhantes ao seu equivalente na física, favorecendo o passo nas direcções em que não há oscilações e travando em caso contrario) e a adaptação do passo na aprendizagem [Almeida 1990b] que fazem com que o algoritmo mude de taxa de aprendizagem, conforme a função erro a minimizar é muito ou pouco "inclinada". Isto tenta obviar ao facto de, perto de um mínimo o gradiente do erro ser baixo e logo a convergência ser mais lenta [Dayhoff 1990], [Almeida 1990b].

Rumelhart [Rumelhart 1986], sugere que uma vez que o método de backpropagation não pode ser aplicado para redes recorrentes (redes com feedback), estas podem no entanto ser substituídas por redes feedforward com tantas camadas quantos instantes de tempo.

### **HOPFIELD**

Na rede de Hopfield é minimizada uma função de energia (ver a expressão em [Hopfield 1982], [Lippman 1987], [Hunt 1992], [Lippman 1988], [Dayhoff 1990]). Ao apresentar-se um padrão de entrada a rede decai para um mínimo local dessa função. Assim o treino da rede consiste em atribuir os pesos de tal forma, que padrões por nós desejados, fiquem como mínimos locais dessa função. Desta forma ao apresentarmos um padrão semelhante a rede dá-nos o padrão memorizado mais próximo ("content adressable memory"). Podemos pois recuperar informação a partir de informação incompleta. Para ajustar os pesos usa-se a regra de Hebb (descrita em [Hunt 1992]), esta regra faz com que a cada padrão de treino corresponda um ponto de equilíbrio da função de energia. Note-se que esta regra gera automaticamente os pontos de equilíbrio, colocando-os em posições desconhecidas à priori pelo utilizador [Hunt 1992].

Para a execução, é apresentada uma entrada (conjunto de estados dos neurónios) e os estados são actualizados. Para isso, escolhe-se aleatoriamente um neurónio cujo o estado é alterado se isso significar uma diminuição do valor da função energia, caso contrário permanece na mesma. Assim eventualmente a rede decai para o mínimo mais próximo da entrada. Nas redes de Hopfield continuas [Hopfield 1984], [Chu 1990] o método de mudança dos estados usando uma semelhança com a equação de carga de um circuito RC, pode ser encontrado em [Hopfield 1984] e [Chu 1990]. Nestas as saídas podem tomar valores contínuos que são função dos seus estados [Chu 1990].

Dado que na aprendizagem as entradas apresentadas são também futuras saídas da rede, estamos no fundo a indicar o resultado que queremos obter pelo que à semelhança de [Lippmann 1987] também consideramos este tipo de aprendizagem como supervisionada.

## **BOLTZMAN MACHINE**

Aqui a ideia é de usar o conceito de temperatura que existe na física das moléculas gasosas (estudos efectuados por Boltzman no fim do século passado) para a aprendizagem nas redes de Hopfield. Esta metodologia baseada em simulated annealing foi introduzida por Hinton, Sejnowski e Ackley [Hinton 1984] (a que eles designaram como máquina de Boltzmann) como uma alteração ao método de disparo das unidades da rede proposta por Hopfield [Hopfield 1982].

O método aqui, ao contrário de Hopfield (que dispara as unidades com uma probabilidade fixa) consiste em disparar as unidades (unidade com o seu estado a ser actualizado) de acordo com probabilidades calculadas a partir da equação de Boltzmann [Hinton 1984]. Nesta entram dois parâmetros: a activação e a temperatura. A temperatura mede a "instabilidade" do sistema, ou seja com temperaturas pequenas as probabilidades de escapar a mínimos locais baixam. A activação é a diferença de "energia" (função energia na rede de Hopfield [Hopfield 1982]) entre os estados activo e inactivo de um neurónio [Hinton 1984]. Hinton refere que o modelo de um problema (no fundo a rede faz a modelização de um sistema) não pode ser um modelo perfeito, no entanto a existência de regularidades no sistema a modelizar torna o modelo bom [Hinton 1984].

Com este método de selecção de neurónios para actualização, consegue-se numa rede de Hopfield com melhores capacidades de fuga a mínimos locais.

## **CMAC**

A aprendizagem deste tipo de redes consiste no "update" dos pesos para os quais os campos receptivos tomam valor um. Desta forma apenas um subconjunto dos pesos é ajustado em cada iteração [Hunt 1992], [Lane 1992]. Com isto consegue-se que a aprendizagem seja mais rápida do que no caso da BP, onde todo o conjunto de pesos é ajustado em cada iteração. Ao ajustar-se apenas uma pequena parte dos pesos, estamos a fazer aproximações locais ao contrário das globais na BP. Assim com aproximações locais ao treinarmos a rede para um dado padrão de entrada não "destruímos" tanto o que foi aprendido para outro padrão diferente. A sensibilidade para a ordem de apresentação dos dados de treino é pois reduzida [Hunt 1992].

## **ALGORITMOS GENÉTICOS**

Este tipo de algoritmos encontra a sua base na teoria da evolução das espécies. Foi com Holland [Holland 1962 e 1975] que a teoria dos algoritmos genéticos teve início.

Esta baseando-se nos conceitos de genes, mutação e cruzamento da natureza, adaptou-os para o seu uso em algoritmos de optimização e pesquisa.

Nos algoritmos genéticos usa-se uma função de avaliação ("fitness") [Menczer 1990] da selecção escolhida para determinar se esta sobrevive ou não face às progenitoras [Porto 1992] (i.e. se a nova selecção é ou não mais apta a representar os nossos objectivos). No caso das redes neuronais uma

selecção consiste num conjunto de pesos, e a sua avaliação indica com que qualidade estes representam o mapeamento entre entradas e saídas.

Inicialmente o algoritmo [Menczer 1990] selecciona aleatoriamente uma dada população. Seguidamente, inicia-se um ciclo onde são seleccionados um ou dois indivíduos, segundo uma função de distribuição. Depois é aplicado um operador ao indivíduo (mutação), ou aos indivíduos (cruzamento), escolhidos. Ao resultado do operador, o filho, é avaliado o "fitness", e é escolhido da população um indivíduo (segundo uma dada função de probabilidade), que vai ser substituído por ele. Finalmente são actualizados os "fitness", a função de probabilidade é normalizada e repete-se o ciclo.

Como mutação entende-se uma alteração de alguma percentagem dos pesos. Esta percentagem é determinada por um parâmetro que mede a localidade mutação. Cada valor mudado está correlacionado com o valor anterior e contém uma alteração aleatória (temos expressão da mutação em [Menczer 1990]).

No cruzamento temos dois vectores de pesos a cruzar, em dois pontos dos vectores faz-se um "corte", de modo a que o vector cruzado contenha alguns pesos de um dos progenitores e alguns do outro (Fig. 9).

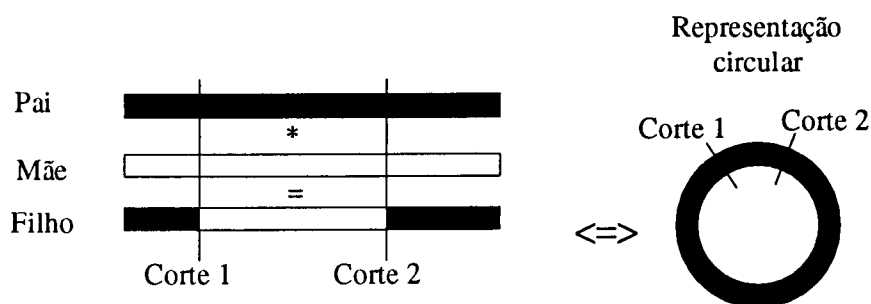


Figura 9

A mutação é boa para exploração e não para convergência para mínimos. Na medida em que, possui um carácter aleatório, permite-lhe explorar novas situações e fugir a eventuais mínimos locais (é um pouco semelhante ao processo "agitar o sistema" em simulated annealing). O crossover ao incorporar as características dos pais revela-se um modo eficiente de exploração em busca de mínimos (pretende-se recombinar as características favoráveis dos progenitores).

Todas estas características tornam este algoritmo bom para espaços de pesquisa largos, complexos e razoavelmente desconhecidos [Jong 1988].

Tem como limitações a lentidão de convergência [Jong 1988] (note-se a rapidez de convergência dos seus congéneres naturais).

## COUNTERPROPAGATION

O treino deste tipo de redes inclui o ajuste dos pesos da camada competitiva e o ajuste da camada de saída. Dado que usamos durante o treino pares entrada-saída, este tipo de aprendizagem é considerado como supervisionada. Primeiro os pesos da camada competitiva são ajustados, só os pesos das ligações que ligam à unidade vencedora é que são actualizados [Dayhoff 1990]. Os pesos da camada de saída que se encontram ligados à unidade vencedora na primeira camada, são actualizados, os outros permanecem inalterados. O ajuste dos pesos na saída é feito com base no erro obtido entre a saída produzida e a saída desejada.

## RADIAL BASIS FUNCTION NETWORKS

A aprendizagem deste tipo de redes (cuja arquitectura foi descrita acima) compreende a determinação não só dos pesos associados aos somadores na saída, mas também a determinação da média e do desvio padrão das funções da primeira camada.

Os "centros" de cada unidade da primeira camada (média) são determinados por particionamento dos dados de treino em subconjuntos, de forma a cada "centro" de um subconjunto seja atribuído a uma unidade na rede [Leonard 1991]. Os pontos de entrada são atribuídos ao subconjunto (e logo unidade) com o "centro" mais próximo.

Os factores de escala (desvio padrão) que determinam a largura de cada função na camada escondida (1ª camada), são determinados de modo a que cada valor seja maior do que a distância até ao próximo centro (ou seja as funções sobrepõem-se parcialmente). Por outro lado devem ter um valor não muito excessivo de forma a não representarem pontos de entrada que correspondam a outra unidade. Uma heurística para determinar o seu valor ("P-nearest neighbor") é dada em [Leonard 1991]. Para determinar os pesos usa-se uma regressão linear de mínimos quadrados. Esta regressão relaciona as saídas com os valores de activação das unidades da primeira camada e não com as entradas. Pode-se encontrar em [Leonard 1991] um método para a obtenção destes pesos.

Face à "backpropagation" o treino é mais rápido [Leonard 1991], e são usadas usando métodos de optimização linear [Leonard 1991], com garantia de solução global [Hunt 1992], são bons aproximadores (dado que particionam bem o espaço de entrada). Têm no entanto o problema de estarem demasiado ligadas ao tamanho do espaço de entrada, levando a redes de grande dimensão para muitos dados de treino [Hunt 1992].

## **NÃO SUPERVISIONADA COMPETITIVA**

Na aprendizagem competitiva, temos pelo menos uma camada de neurónios que efectua a competição. Podemos ter por exemplo 2 camadas: 1 de entrada e uma competitiva [Rumelhart 1985]. Primeiro calcula-se as somas pesadas das entradas para cada neurónio competitivo. O que tiver soma maior vence. E só este vê os seus pesos actualizados [Dayhoff 1990], [Rumelhart 1985].

Podemos ter mais do que duas camadas, aqui a ideia central é de que, após o calculo de um dado valor (normalmente a soma pesada das entradas), para cada neurónio, o que tiver o maior ( ou menor ) sai vencedor. E só os pesos dos vencedores são reforçados. Este reforço faz com que certas características dos dados de entrada sejam apreendidas na rede.

Este tipo de aprendizagem à semelhança dos mapas de Kohonen descritos a seguir (que também realizam aprendizagem competitiva) classifica os dados de entrada por si só (aprendizagem não-supervisionada). Agrupa-os segundo classes em função de semelhanças neles detectadas. Estas semelhanças são detectadas pela rede, o que torna este tipo de redes bom para análise de grande quantidades de dados e para encontrar neles relações.

### **KOHONEN**

A aprendizagem consiste no ajuste dos pesos. Este ajuste [Kohonen 1990], [Lippman 1987] é feito de forma não-supervisionada. Assim começando com valores iniciais aleatórios, os pesos são ajustados em todos os nós na vizinhança do nó, cuja distância é mínima da entrada, o ajuste é feito através da diferença entre a entrada e o valor do peso, esta afectada de um ganho de diminui no tempo. De igual forma a largura da vizinhança diminui com o tempo. Assim inicialmente são feitos ajustes grossos e com a passagem do tempo os ajustes tornam-se mais finos.

Este tipo de aprendizagem e rede leva à sua utilização em agrupamento ou quantização de um conjunto de dados grande [Lippmann 1987]. Aqui a aprendizagem é competitiva dado que são os nós que estão mais próximo da entrada apresentada que vêm os seus pesos ajustados. Por outro lado é não-supervisionada dado que nos limitamos a apresentar entradas e o mapa ajusta os pesos de forma a agrupar de alguma forma essas entradas (este agrupamento não é inicialmente conhecido, e de uma forma geral é o resultado pretendido).

## Aplicações em controlo

### IDENTIFICAÇÃO

Certas propriedades das redes neuronais, tornam-nas em ferramentas úteis na identificação de sistemas.

Uma propriedade importante é sem dúvida a capacidade das redes neuronais multilayer aproximarem uma vasta gama de funções não-lineares [Cybenko 1989]. Em estudos teóricos, provou-se que qualquer função continua pode ser aproximada arbitrariamente bem [Cybenko 1989], [Hornik 1989]. Inclusive, Hornik demonstrou que uma RN multilayer feedforward com uma camada escondida, pode, sob certas condições, aproximar uma função e as suas derivadas [Hornik 1989]. Estas propriedades de aproximação são tanto mais úteis que pelo teorema de Kolmogorov, qualquer função de múltiplas variáveis pode ser calculada usando apenas somas e funções não lineares crescentes de uma só variável. Vários estudos focam RN e a sua relação com o teorema de Kolmogorov [Hunt 1992], [Kurková 1992], [Cotter 1992].

Outros estudos [Cardaliaget 1992], dão formulas explícitas para a aproximação de funções e suas derivadas, nomeadamente funções definidas de forma implícita.

Esta propriedade de aproximação das RN, revela-se particularmente útil no caso dos sistemas não-lineares. O caso não linear não apresenta problemas extra de tratamento (ao contrário do que acontece com a generalidade da teoria de controlo).

Para identificação de sistemas, é frequente o uso de redes multilayer feedforward [Chen 1990c], [Narendra 1990], [Narendra 1991b], embora haja estudos [Chu 1990] do uso de redes de Hopfield, neste campo.

O problema de identificação põe-se em, determinar uma função  $P^*$  que aproxima o sistema desconhecido  $P$  a menos de um dado erro [Narendra 1990]. Assim não temos a planta, mas sim um conjunto de entradas  $u(k)$  para as quais temos as saídas  $y(k)$  (estamos a assumir aqui o caso discreto, note-se que é esta a situação usual pois pode-se passar de  $u(t)$  e  $y(t)$  para  $u(k)$  e  $y(k)$  por amostragem conveniente). Estes pares entrada-saída são obviamente úteis para o treino de uma rede. Podemos pois ter uma rede neuronal como modelo de identificação (Fig.10).

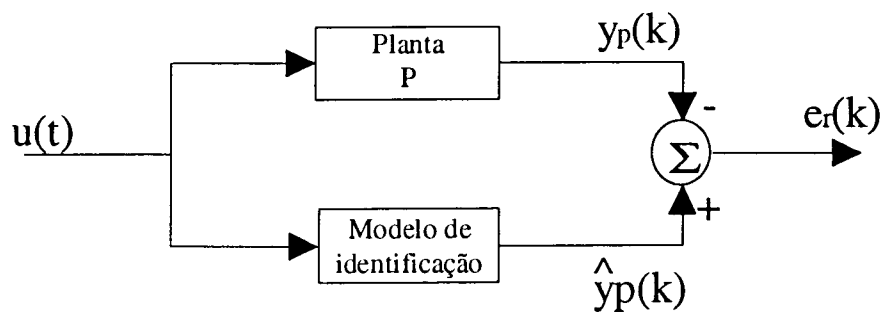


Figura 10



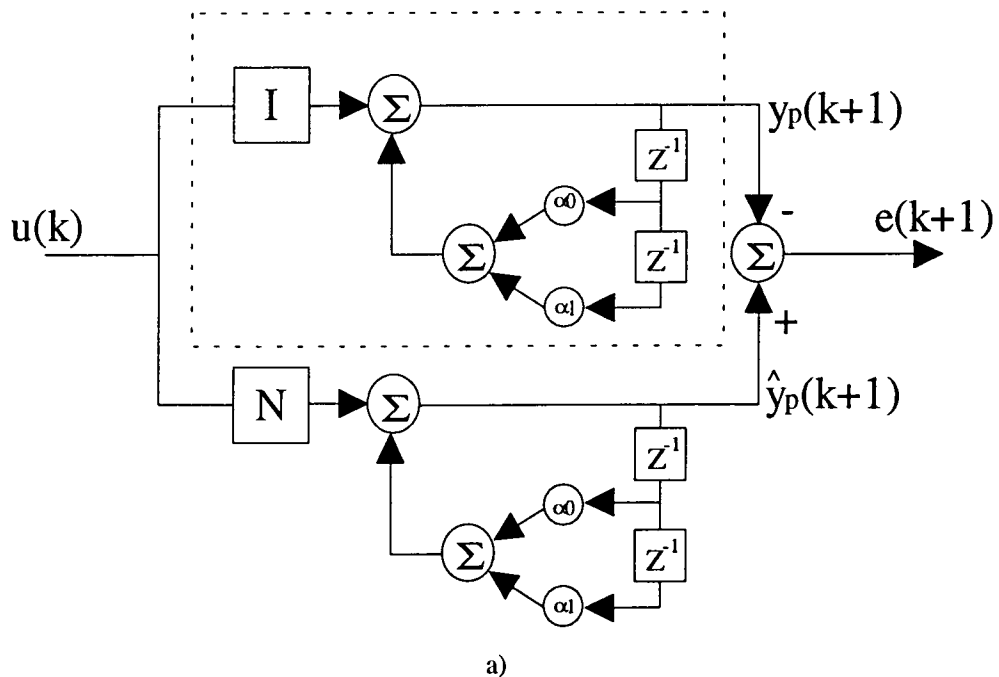
Teremos é que definir o que é o erro a minimizar durante o treino. Este erro, é o erro de identificação e consiste na diferença entre a saída produzida pela planta para um dado controlo, e a saída produzida pelo modelo. Deste modo ao minimizarmos este erro estamos a aproximar o modelo da planta. Esta minimização é feita ajustando os parâmetros do modelo, que aqui são os pesos da rede.

Tudo isto pressupõe que a planta pode ser aproximada por uma rede neuronal (o que é em grande parte garantido por [Cybenko 1989], [Hornik 1989] e outros).

O modelo de identificação poderá não consistir só numa rede neuronal, mas também num sistema constituído por uma RN e por uma parte linear (o que Narendra designa por rede neuronal generalizada [Narendra 1990] e [Narendra 1991b]). Este modelo é útil para aproximar sistemas não-lineares, dado ter elementos não-lineares (a rede) e termos lineares que aproximam a parte linear da dinâmica do sistema.

O treino da rede pode ser feito off-line, quanto estamos perante um caso identificação, sem problemas temporais, por exemplo o estudo de uma dada planta desconhecida. Ou pode ser feito on-line, como é o caso do controlo adaptativo com redes neuronais (ver mais adiante).

O modelo de identificação pode ser ou paralelo ou série-paralelo (ver equações em [Narendra 1990]). No primeiro é a saída do modelo que produz realimentação para o modelo (Fig. 11a), no segundo a realimentação do modelo vai-se buscar à saída da planta (Fig. 11b).



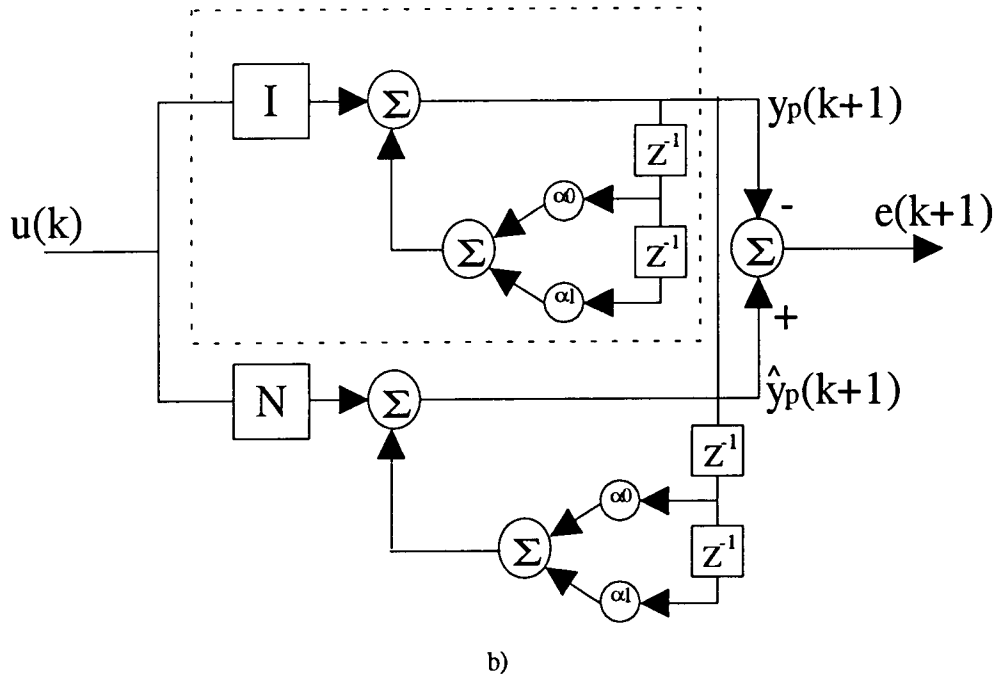


Fig. 11

O problema do modelo paralelo é que não tem garantias de estabilidade [Narendra 1990]. Isto mesmo na assunção (que é normalmente feita), de que a planta é BIBO ("bounded input - bounded output"), isto é, apresenta saídas limitadas (estável) na presença de entradas limitadas. Daí o uso do segundo modelo, por Narendra em [Narendra 1990] e [Narendra 1991b].

Este obteve bons resultados de simulação em [Narendra 1990] e [Narendra 1991b] (ver secção de simulação nos artigos), tendo casos onde por vezes é quase impossível distinguir a resposta do modelo da planta.

Outro caso de identificação de sistemas usando redes feedforward é o proposto por Chen [Chen 1990]. Neste estudo propõe-se a identificação de sistemas usando como modelo de identificação o modelo NARMAX ("non-linear autoregressive moving average with exogenous inputs"). Segundo Chen o modelo NARMAX é independente da forma da função não-linear, que serve de base à aproximação (função não linear das entrada, saída e ruído atrasado), podendo-se usar redes para a representação dessa função.

Chen propõe a simplificação de termos só ruído aditivo e não correlacionado (ver [Chen 1990], esta limitação é ainda significativa). Para o treino da rede, são propostos os algoritmos clássicos na teoria de identificação de sistemas, de predição de erro e de predição recursiva de erro [Chen 1990b] (também estudados em RN por Venugopal em [Yuh] para um caso de controlo, a ver mais adiante). Estes algoritmos de uma forma geral minimizam uma função de erro, e fazem-no segundo métodos de máxima descida (gradiente). Uma característica do uso de RN para o modelo NARMAX, salientada por Chen é a

de dar uma representação parcimoniosa, isto é, serem precisos menos parâmetros do que por exemplo no caso de aproximação polinomial [Chen 1990].

Usando ainda o modelo NARMAX, Chen propõe ainda uma aproximação usando "radial basis functions" (RBF). O método de aproximação com RBF é tradicionalmente usado para interpolação em espaços multidimensionais, daí a sua aplicação para a aproximação de funções. Este método pode ser aplicado usando RN. Estas têm uma configuração igual à apresentada na secção anterior de arquiteturas. O ajuste de parâmetros, consiste neste caso no cálculo dos centros (as funções podem ser outras além da de Gauss) e dos pesos (ver eq. 6 em [Chen 1990c]). Os centros ou são distribuídos uniformemente ou são retirados dos dados de entrada. Chen propõe um algoritmo iterativo: "orthogonal forward regression (OFR)", para o cálculo dos parâmetros.

O modelo usado no estudo [Chen 1990c] é o NARMAX, mas ao contrário do caso em [Chen 1990] não é feita a limitação em termos da "forma" do ruído. Isto é uma vantagem bastante grande já que, neste caso, o modelo já é aplicável a uma grande parte de casos reais. Note-se que, como é explicado em [Chen 1990c], o ruído geralmente é correlacionado. A desvantagem da aproximação com RBF é o maior número de parâmetros que normalmente necessita face à aproximação polinomial.

Apesar de menos vulgar, outra utilização de redes neuronais na identificação de sistemas é o uso de redes de Hopfield. Por exemplo: em [Chu 1990], é apresentado um caso em que uma rede de Hopfield é usada na identificação de sistemas. São medidas as entradas, variáveis de estado e derivadas das variáveis de estado, para a programação da rede. Os estados da rede convergem para os valores dos parâmetros do sistema. Outra aplicação sugerida por Chu em [Chu 1990] é o uso das redes de Hopfield para modelizar um sistema em termos dos seus coeficientes de Fourier (determinação das funções de transferência em frequência). Hopfield [Hopfield 1984], já havia mostrado a aplicação das RN para a decomposição de um sinal, daí a utilização das redes de Hopfield contínuas para neste caso, fazer-se uma decomposição, com funções de base ortogonais. A identificação por redes de Hopfield usa aqui um método de mínimos quadrados para determinar os pesos. A identificação da resposta em frequência, usa a regra de Widrow-Hoff com pequenas alterações (ver eq. 14 de [Chu 1990]) para modificar os pesos. Como entradas são usados senos e cossenos.

O uso das RN para identificação de sistemas, encontra-se frequentemente devido às capacidades de aproximação e adaptação das redes. Normalmente a identificação de sistemas com redes neuronais encontra-se associada ao controlo (nomeadamente o controlo adaptativo). Com o controlo através de RN, estas frequentemente são usadas para aprenderem a dinâmica (ou a dinâmica inversa) do sistema a controlar.

## CONTROLO ADAPTATIVO

O uso de redes neuronais para controlo adaptativo, é talvez o caso de maior número das aplicações das RN em controlo, como se pode comprovar pela sua referência em inúmeros artigos: [Narendra 1991a], [Werbos 1991], [Antsaklis 1992], [Bavarian 1988], [F.Chen 1990], [Psaltis 1988], [Tabary 1992], [Guez 1988], [Cooper 1992]

No controlo adaptativo a ideia central é de que, o controlador de alguma forma, detecta mudanças na planta e muda os seus parâmetros internos para uma melhor resposta.

Os sistemas adaptativos que fazem uso explícito de modelos para o controlo, são usualmente designados por MRAC ("Model Reference Adaptive Control"). Um esquema do controlo MRAC pode ser observado na figura 12.

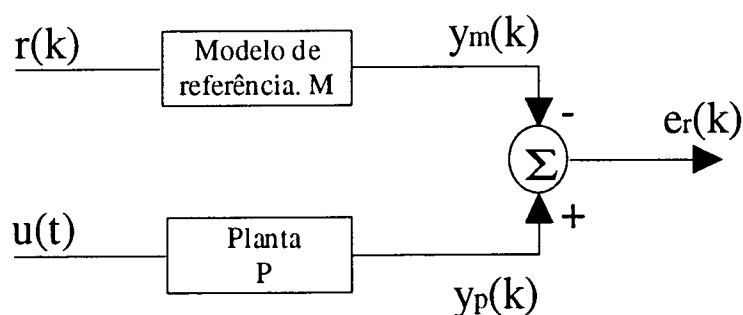
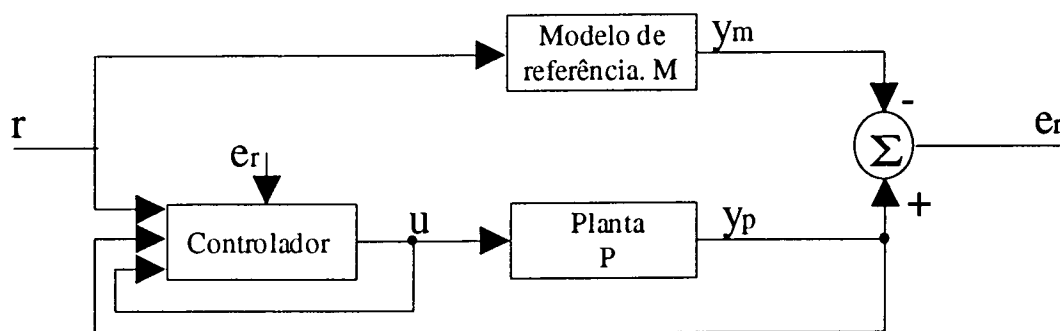


Figura 12

Podemos ver que o objectivo é determinar o controlo  $u(k)$  por forma a minimizar a diferença entre a saída da planta e a saída do modelo.

As RN podem ser usadas para efectuarem o controlador, como é o caso em que são usadas redes generalizadas [Narendra 1990], [Narendra 1991b] (designação dada por Narendra a sistemas contendo redes neuronais e sistemas lineares).

Existem dois tipos de controlo adaptativo: directo e indirecto (fig. 13a. e fig. 13b).



a)

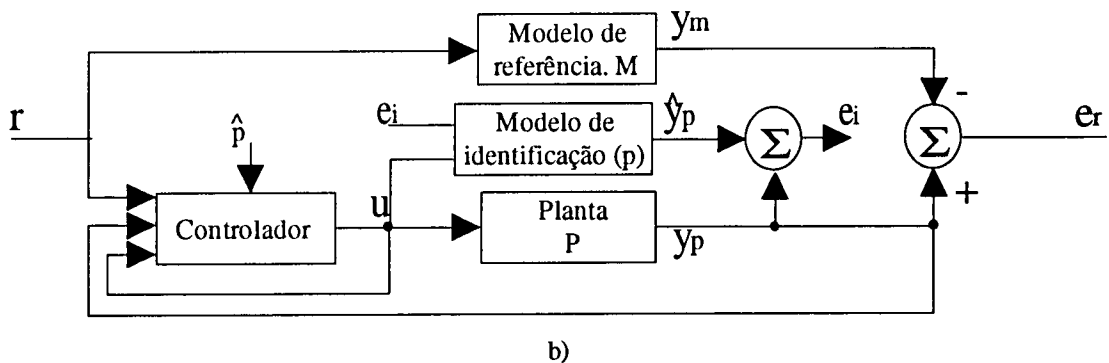


Figura 13

No controlo directo os parâmetros do controlador são ajustados para reduzir o erro de saída, no indirecto é suposto que o vector  $p^*$ , resultante da identificação da planta corresponde ao vector  $p$  dos parâmetros reais da planta. Isto é, usa-se a identificação do sistema em vez da planta real.

Provas da estabilidade do MRAC com controlo directo pode ser encontrada em [Narendra 1980] e em [Narendra 1980b] para o caso discreto.

Narendra sugere a aplicação do método indirecto com RN, isto porque no directo existe uma planta não-linear entre o controlador e o erro de saída [Narendra 1990].

Pode-se observar na Fig. 14 o esquema proposto por Narendra para o controlo adaptativo indirecto.

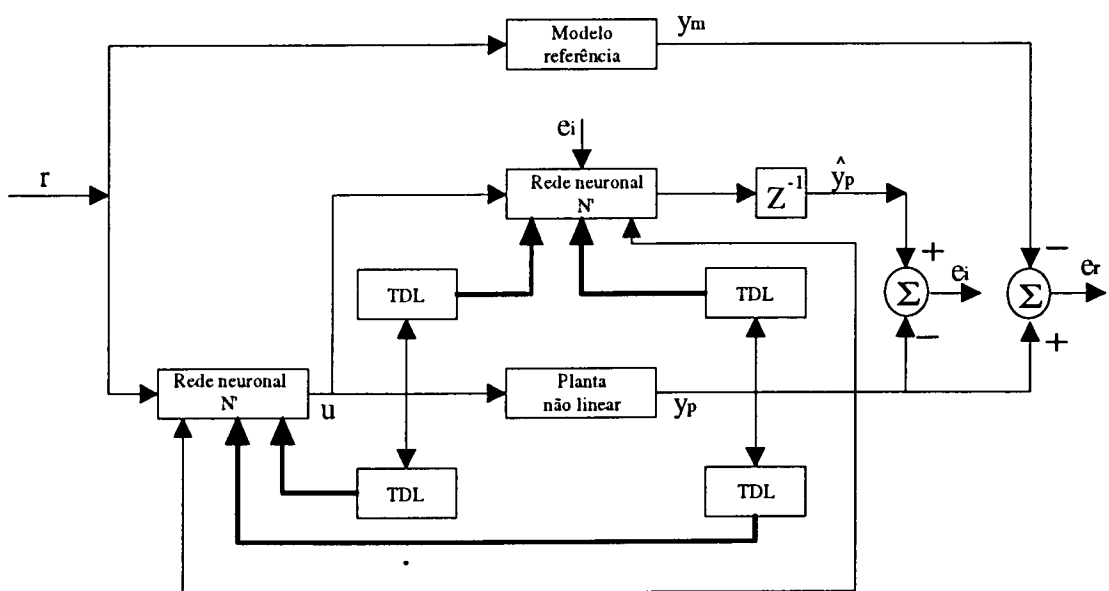
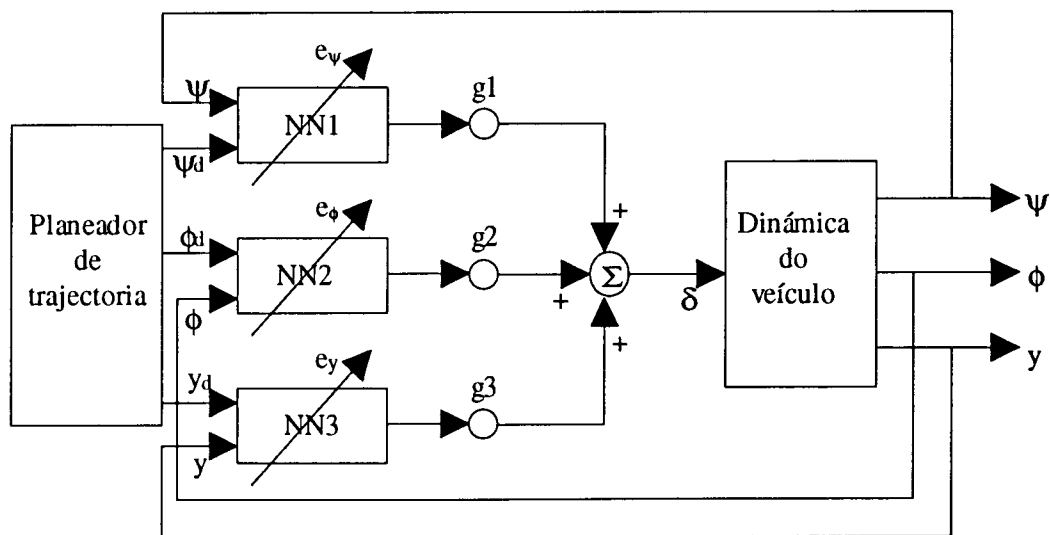


Figura 14

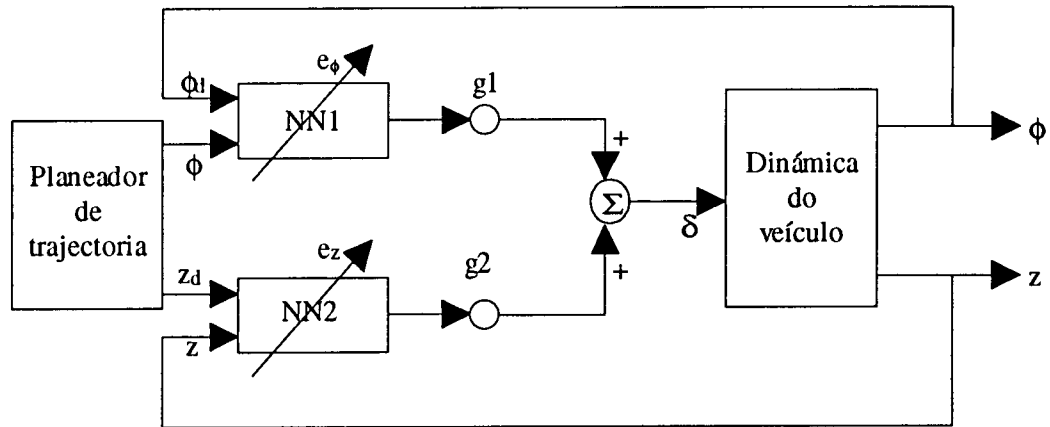
Podemos constatar que neste caso são usadas redes neuronais quer para a identificação do sistema, quer para realizar o controlador. Na figura TDF significa "tapped delay line", isto é um bloco que fornece

como saídas valores atrasados da sua entrada. Isto é usado pois Narendra propõe para a aprendizagem da rede o que lhe chama de dynamic backpropagation (ver [Narendra 1991b] para um tratamento mais detalhado, não só do método, mas também da identificação e controlo de sistemas). Isto é, o treino da rede não só com valores presentes, mas também com valores atrasados das entradas. Com controlo on-line a identificação e o controlo são usados simultaneamente.

Nem só o método indirecto é usado, segundo Venugopal [Venugopal ??], [Venugopal 1992] é usado o controlo directo. Aparentemente as objecções apresentadas em [Narendra 1990] não constituem problema para Venugopal, este sugere que para obviar à existência da planta não-linear desconhecida, um ganho que é colocado entre o controlador e a planta. Este ganho é proporcional ao inverso do jacobiano da dinâmica e é calculado também por ajuste durante o treino da rede. Em [Venugopal 1992] é apresentada a lei de ajuste deste ganho. Inicialmente [Venugopal ??], é apresentado um caso de controlo directo, em que o controlador consiste numa RN que aprende on-line (é treinada on-line com a referência desejada) as relações entre os ângulos de direcção, velocidades e os ângulos de "rudder" e "stern" que constituem o controlo da planta. A planta é um veiculo submarino autónomo. Apesar do sistema ser MIMO, são usadas 6 redes para implementar 5 controladores SISO que depois são combinados para obter as duas entradas de controlo na planta (ver figura 15).



a)



b)

Figura 15

Neste caso o controlo da velocidade para a frente, não é analisado. Este método de separação da dinâmica em várias relações SISO combinadas apresenta bons resultados de simulação (ver [Venugopal 1992]), se bem que seria de todo desejável a utilização de um controlador MIMO que incorporasse o acoplamento existente na dinâmica.

Dado que a rede aprende on-line, é necessário um compromisso entre o overshoot e a taxa de aprendizagem. Com uma taxa elevada temos maior rapidez de aprendizagem, e logo mais adaptabilidade. No entanto isso trás-nos maior overshoot e também problemas de estabilidade (em [Narendra 1990] explica-se a necessidade de termos uma taxa pequena para garantir estabilidade). Note-se que a complexidade da rede para o modelo linear do AUV é a mesma que para o não-linear, logo o uso de redes para modelos não-lineares é vantajoso.

Outro método para superar o problema levantado por Narendra, sobre o controlo directo, é o apresentado por Yuh [Yuh], [Yuh 1990]. Neste Yuh sugere a estimação do erro no controlo, com base no erro na saída da planta. Para esta estimação é usada uma "equação crítica". Esta consiste numa matriz que multiplica pelo erro na saída, obtendo-se como resultado o erro no controlo. É este o erro que vai ser minimizado por backpropagation no treino da rede. Em [Yuh] são estudados três métodos de aprendizagem para a rede: error backpropagation, recursive prediction error e uma versão modificada deste último. Dos três métodos o que apresentou melhores resultados foi backpropagation (os outros não se revelaram superiores em nenhum aspecto). No entanto Yuh sustenta sem provar ou pelo menos apresentar uma explicação, de que os métodos PRPE e MPRPE teram melhor resposta no caso de sistemas variantes no tempo, ou com inércia variável no tempo. Segundo Yuh a matriz crítica é dependente da taxa de amostragem na discretização da dinâmica e da matriz de inércia da planta (a planta também é um AUV), em [Yuh 1990] é apresentada a forma de calcular a eq. crítica (após outro estudo efectuado anteriormente por Yuh). Pelos resultados de simulação, pode-se observar que a aprendizagem on-line é de primordial importância. Existindo bastantes perturbações na dinâmica da

planta (o AUV movimenta-se num meio inconstante, turbulento e imprevisível) é necessária aprendizagem on-line. Isto mesmo pode ser comprovado pelos resultados de simulação, onde se vê que uma vez parado o treino da rede, esta já não consegue controlar o veículo. A limitação imposta por Yuh nesta estratégia de controlo é a existência e o conhecimento de um limite superior para a matriz de inércia do sistema, isto para que se possa calcular a matriz dos coeficientes críticos.

Sanner [Sanner 1990] propõe à semelhança de Yuh um método para ajustar os pesos da rede, com base num sinal de "crítica" que substitui o erro. No fundo esta crítica é obtida de modo semelhante ao de Yuh por cálculo a partir do erro na saída.

Nem só backpropagation é usada no controlo adaptativo, por exemplo: em [Barto 1983] temos o controlo de um sistema de "cart-pole" (pêndulo invertido, apoiado num carrinho em movimento), com o uso de "reinforcement", um sinal de recompensa que classifica o controlo (neste caso é ou 1 ou -1) e que é usado na actualização dos pesos de um neurónio para o controlo. O esquema apresentado tem ainda outro neurónio que com base nas entradas do primeiro e no valor de recompensa, determina a recompensa para o primeiro neurónio. Outro caso semelhante (também por um método de recompensa e numa planta "cart-pole") é o de [Andersen 1989], neste em vez de 2 neurónios temos duas RN de 2 camadas: uma que avalia o estado e outra que em função do estado avaliado produz o controlo.

Um caso de controlo, em que temos também equilíbrio é o de [Tabary 1992], neste caso o objectivo é manter a barra superior de uma bicicleta horizontal, variando o ângulo das duas forquilhas que é ajustável. O terreno supõe-se irregular. Um problema acrescido é o da impossibilidade de controlar todas as variáveis de estado. São usadas duas redes: uma "estática" que nos dá os ângulos supondo a bicicleta parada. A outra leva em conta os efeitos da velocidade e fornece a velocidade adequada, para a altura de um ponto da bicicleta, face ao solo (uma medida da irregularidade do terreno). O método de aprendizagem das redes é o de backpropagation.

Uma outra implementação de controlo adaptativo, é a da associação sensorial-motora [Kuperstein 1989], neste caso temos um braço articulado e um câmara stereo. Várias redes fazem a associação entre as imagens de um objecto e os sinais necessários para a sua manipulação (ver artigo para um esquema com o mapeamento efectuado pelas RN). As diferenças entre os dados adquiridos pelos sensores (câmaras) e a posição do braço, são usadas para a aprendizagem das redes.

Temos também controlo adaptativo através do uso de uma rede competitiva (um mapa de Kohonen) [Cooper 1992]. Para o treino da rede é apresentado um vector com as perturbações no sistema a ser controlado. A rede deve ajustar um controlador PI. Deve-se salientar que este artigo está longe de ser completo ou explícito. Na realidade não é sequer apresentada a forma de cálculo dos pesos, a origem dos dados de entrada e a forma de como o mapa de Kohonen altera os parâmetros do controlador PI.

Em [Kraft 1990] podemos ver uma comparação entre dois métodos clássicos de controlo adaptativo: "self-tuning regulator" e um MRAC baseado em funções de Lyapunov, e, uma rede CMAC. A rede apresenta resultados médios, e exceptuando a velocidade de convergência, em comparação com os outros



métodos é muito regular nas prestações. Se não apresenta algumas das muito boas características dos outros métodos, também não apresenta os seus grandes defeitos.

Uma outra discussão comparativa, neste caso entre o clássico MRAC e RN pode ser encontrada em [Guez 1988], saliente-se a capacidade das redes poderem adquirir o seu modelo de referência por exemplos (treino). Note-se também que esta característica das RN permite um maior desconhecimento da planta, já que embora não sabendo sequer um modelo de referência, os pares entrada-saída da planta são suficientes para treinar a rede.

## **CONTROLO ÓPTIMO**

O problema de controlo óptimo não é muitas vezes tratado por redes neuronais. Isto acontece na medida em que geralmente as RN não dão soluções garantidamente óptimas. Em [Sutton 1992] é sugerido um método de treino para obtermos controlo óptimo adaptativo numa rede neuronal. O método é "reinforcement learning", isto é, aprendizagem por recompensa. A ideia deste método de aprendizagem é a de alguma forma criticar o resultado das acções desencadeadas e logo diminuir ou acentuar a tendência para efectuar uma dada acção. É proposto o método de "Q-learning" que mantém estimativas sobre todos os pares estado-acção e usa estas estimativas para seleccionar acções.

Segundo Sutton, este método é controlo óptimo directo e adaptativo (sem demonstração). Não são apresentados resultados de simulação que sustentem as considerações teóricas desenvolvidas.

É natural que as RN não estejam vocacionadas para controlo óptimo. De facto as características das RN tornam-nas atractivas para outras aplicações na área de controlo.

## **DETECÇÃO DE FALHAS**

As capacidades de reconhecimento de padrões das RN, tornam a sua utilização na detecção de falhas, bastante proveitosa. Podemos encontrar em [Sartori 1992] um estudo (reduzido) da aplicação das RN não só na detecção de falhas mas também em controlo no geral. É apresentada uma metodologia genérica para o desenho e treino de redes multilayer.

Um caso de detecção de falhas é o de [Naidu 1990], para falhas no controlo de processos. É usada uma rede feedforward de três camadas, com um único nó de saída. Esta rede recebe as transformadas coseno do sinal de diagnostico e fornece um resultado indicando a gravidade da falha. O treino é feito com entradas aleatórias e por backpropagation, ou seja alguém no treino classifica as falhas.

Passino em [Passino 1989] apresenta um estudo sobre as capacidades das redes multilayer feedforward para conversão numérica-simbólica. Uma aplicação interessante é a da construção de uma "árvore" de falhas para um avião. A rede particiona o espaço de entrada em várias regiões de decisão. Assim decide quando há falhas (entradas, i.e. parâmetros observáveis do avião, em dados particionamentos do espaço de todos os valores possíveis) e qual é o tipo de falha.

A detecção de falhas não se restringe apenas às redes tendo como funções de activação o sigmoide. Uma rede de RBF ("Radial Basis Functions") pode também ser usada para a classificação de avarias [Leonard 1991]. Neste artigo são apontadas algumas das falhas da backpropagation, nomeadamente o facto de sob certas condições as superfícies de decisão não serem intuitivas ou robustas (note-se que a BP particiona o espaço de entrada de forma irregular, mas em contrapartida fá-lo geralmente com menos termos que as RBF).

Um outro caso que combina detecção de falhas com controlo de uma forma hierárquica é o apresentado em [Narendra 1991c].

Neste, assume-se que a planta pode encontrar-se num de um numero finito de configurações. O controlo do sistema é feito em três fases: na primeira monitoriza-se a resposta do sistema para estimar a configuração a que a planta pertence. Na segunda activa-se um controlador fixo para estabilizar o sistema nessa configuração e na terceira activa-se um controlador adaptativo para melhorar a resposta. Note-se que estas duas últimas fases constituem na prática uma só, pois o controlador fixo corresponde aos valores iniciais do controlador adaptativo. Na figura 16 pode-se observar um esquema de como é feito o controlo.

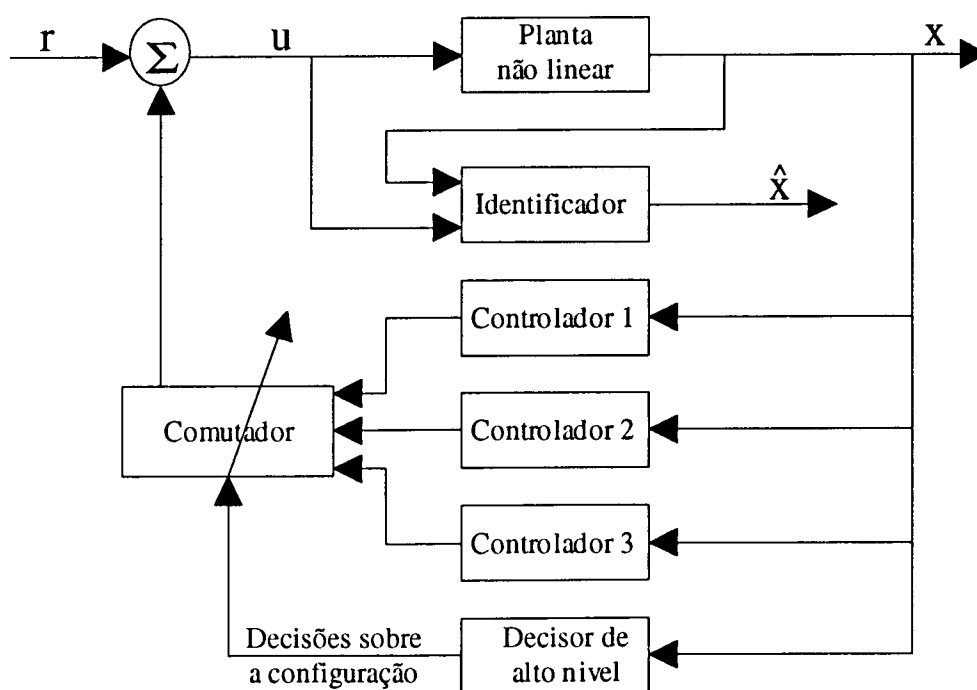


Figura 16

É usada uma RN para o decisor de alto-nível, este detecta a configuração da planta. O a rede identificadora faz a identificação do sistema para determinar os parâmetros desconhecidos do sistema isto para o caso de termos cada configuração parametrizada. Uma restrição que tem que se verificar é a

da incerteza nos parâmetros não afectam a detecção da configuração nem a estabilidade do sistema. Os controladores podem ou não serem implementados com RN, estes supõem-se já existirem e terem os seus valores iniciais determinados por treino off-line (no caso de RN). O treino da rede de alto-nível é feito off-line. Só a identificação e os controladores adaptativos (se for caso disso) são treinados em parte off-line e em parte on-line.

## **CONTROLO HIERÁRQUICO**

As aplicações de redes em controlo hierárquico não são muitas. De facto não foi encontrada nenhuma aplicação onde tenhamos redes neuronais hierarquicamente a otimizar as prestações globais de um sistema.

Em [Rabelo 1992] é proposto uma arquitectura hierárquica com RN para o controlo de um braço de robot com 3 graus de liberdade. Este controlador não se encontra bem descrito não apresentando resultados de simulação suficientemente válidos. O único ponto de interesse é a referência a um método de criação dinâmica de nós para RN (note-se que nós com os pesos que lhe estão ligados a 0, são nós inexistentes).

Em [Kawato 1988] um sistema é proposto para o controlo voluntário de movimentos. Três níveis são considerados: determinação da trajectória desejada, transformação da trajectória de coordenadas visuais para as corporais e geração dos comandos para os motores. As RN são usadas para modelizarem o sistema motor, determinarem a trajectória e transformarem as coordenadas. Note-se que também neste caso a informação fornecida pelos autores não é completa, tendo o artigo um carácter um tanto genérico. Talvez o mais bem documentado caso (e mesmo assim não completamente) de controlo hierárquico com RN, seja o descrito na secção anterior, [Narendra 1991c]. Esta arquitectura apresenta uma estrutura hierárquica, embora não seja geral.

## **EXPERT SYSTEMS**

A integração de redes neuronais com sistemas periciais é rara, e por isso também o é no campo do controlo.

Podemos ver em [Handlman 1990] um exemplo desta integração. O sistema pericial tem a função de supervisionar o treino da rede neuronal. A rede escolhida é uma CMAC, devido às suas capacidades de rápida aprendizagem.

Não é possível avaliar correctamente este método devido à inexistência de dados comparativos com o caso de controlo só por RN.

## OUTROS MÉTODOS DE CONTROLO

Nem sempre o controlo tem de ser feito de forma convencional com um controlador antes da planta ou um modelo de identificação em paralelo.

Este é o caso de um interessante método para controlo, apresentado por Nguyen em [Nguyen 1990]. Esta arquitectura de controlo usa redes multilayer feedforward.

São usadas redes em duas tarefas distintas: uma rede controladora e uma rede emuladora (rede com o modelo do sistema, obtido por identificação).

É assumido que os estados da planta são directamente observáveis sem ruído.

Inicialmente o emulador é treinado, por forma a reproduzir a dinâmica da planta. Este emulador tem como saídas os estados, e como entradas os estados e as entradas da planta.

O objectivo é levar a planta de um estado inicial para um estado final determinados. Para treinar o controlador usa-se o processo descrito na figura 17.

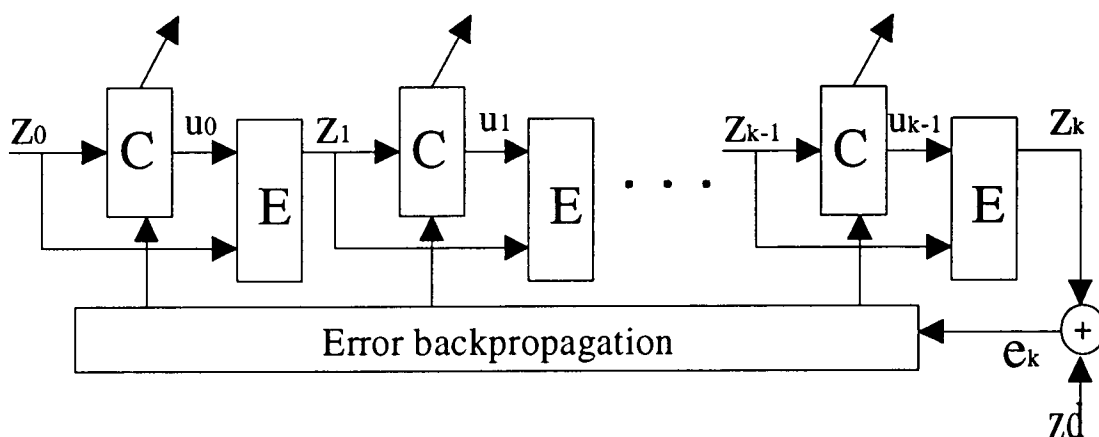


Figura 17

Este processo usa tantos pares de redes controladoras e emuladoras quanto o numero de passos especificado, para a passagem do estado inicial para o estado final. Isto é: inicialmente fornece-se o numero de passos que se pretende para a passagem dos estados inicial ao final, assim como estes estados. O método de aprendizagem é essencialmente o backpropagation. O emulador é treinado inicialmente e depois permanece fixo. São os pesos do controlador que são ajustados na segunda fase. Com a configuração da figura 17, vemos que introduzindo o estado inicial no primeiro par controlador-emulador, à saída deste temos o estado seguinte, e assim sucessivamente até atingirmos o estado final. Esta sucessão de estados é a solução pretendida para o problema.

Neste estudo é apresentada a simulação do estacionamento de um camião com atrelado. Este método revela-se promissor para a resolução de problemas de decisão sequencial.

## Bibliografia

- [Albus 1975] J. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)", *Trans. ASME- J. Dyn. Syst. Meas. Control*, Vol 97, pp220, 1975
- [Almeida 1990a] Luís B. Almeida, "Backpropagation in feedforward recurrent networks", *Neural Networks*, IEEE Computer Society Press, Bruce Shriver (ed.), 1990
- [Almeida 1990b] Luís B. Almeida, Fernando M. Silva, "Speeding up backpropagation", *Advanced Neural Computers*, North-Holland, R. Eckmiller (ed.), 1990
- [Andersen 1989] Charles W. Andersen, "Learning to control an inverted pendulum using neural networks", *IEEE Control Systems Mag.*, vol 9, no 3, Apr. 1989
- [Antsaklis 1992] Panos Antsaklis, "Neural networks in control systems", *IEEE Control Sys. Mag.*, Apr. 1992
- [Barto 1983] Andrew Barto, Richard S. Sutton, Charles W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems", *IEEE Trans. Systems Man Cybernetics*, vol 13, no 5, Sep 1983
- [Bavarian 1988] Beham Bavarian, "Introduction to neural networks for intelligent control", *IEEE Control Sys. Mag.*, Apr. 1988
- [Cardaliaguet 1992] Pierre Cardaliaguet, Guillaume Euvrard, "Approximation of function and its derivative with a neural network", *Neural Networks*, vol 5, 1992
- [Chen 1990] S. Chen, S. A. Billings, P. M. Grant, "Non-linear System identification using neural networks", *Int. Journal Control*, vol 51, no 6, John Wiley & Sons, 1990
- [Chen 1990b] S. Chen, S. A. Billings, P. M. Grant, "Parallel recursive prediction error algorithm for training layered neural networks", *Int. Journal Control*, vol 51, no 6, John Wiley & Sons, 1990

- [Chen 1990c] S. Chen, A. Billings, P. M. Grant, "*Practical identification of NARMAX models using radial basis functions*", Int. Journal Control, vol 51, no 6, John Wiley & Sons, 1990
- [F.Chen 1990] Fu-Chuang Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control", IEEE Control Systems Mag., vol 10, no 3, Apr. 1990
- [Chu 1990] S. Reynold Chu, Rahmat Shoureshi, Manuel Tenorio, "*Neural networks for system identification*", 1990
- [Cooper 1992] D. J. Cooper, L. Megan, R. F. Hindle Jr., "*Disturbance pattern classification and neuro-adaptive control*", IEEE Control Sys. Mag., Apr. 1992
- [Cotter 1992] Neil E. Cotter, Thierry J. Guillerom, "*The CMAC and a theorem of Kolmogorov*", Neural Networks, vol 5, pp221, 1992
- [Cybenko 1989] G. Cybenko, "*Approximation by superpositions of a sigmoidal function*", Math. Control Signals Syst., Vol. 2, pp303, 1989
- [Eberhart 1990] *Neural Network PC Tools*, Eds: Russell C. Eberhart, Roy W. Dobbins, Academic Press, 1990
- [Fitzpatrick 1988] J. Michael Fitzpatrick, John J Grefenstette, "*Genetic algorithms in noisy environments*", Machine Learning 3, pp108, Kluwer Academic Publisher, 1988
- [Fogel 1990] D. B. Fogel, J. W. Atimar, "*Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems*", Biological Cybernetics , no 63, pp114, 1990
- [Guez 1988] Allon Guez, James L. Eilbert, Moshe Kam, "*Neural network architecture for control*", IEEE Control Sys. Mag., Apr. 1988

- [Handlman 1990] David A. Handlman, Stephen H. Lane, Jack F. Gelfand, "*Integrating neural networks and knowledge-based systems for intelligent robotic control*", IEEE Control Systems Mag., vol 10, no 3, Apr. 1990
- [Hecht-Nielsen 87] R. Hecht-Nielsen, "*Counterpropagation networks*", Applied Optics, 26, pp4979
- [Hinton 1984] G. F. Hinton, T. J. Sejnowski, D. H. Ackley, "*Boltzman machines: constraint satisfaction networks that learn*", Tech. Report. CMU-CS-84-119, Dept. Comp. Sci. Carnegie-Mellon University., May 1984
- [Hopfield 1982] J. J. Hopfield, "*Neural networks and physical systems with emergent collective computational abilities*", Proc. Natural Acad. Sci. USA, vol 79, pp2554, Apr. 1982
- [Hopfield 1982] J. J. Hopfield, "*Neurons with graded response have collective computational properties like those of two-state neurons*", Proc. Natural Acad. Sci. USA, vol 81, pp3088, May 1984
- [Hornik 1989] K. Hornik, M. Stinchcombe, H. White, "*Multi-layer feedforward networks are universal approximators*", Neural Networks, vol 5, pp359, 1989
- [Hornik 1989] K. Hornik, M. Stinchcombe, H. White, "*Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks*", Neural Networks, vol 3, 1990
- [Hunt 1992] K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop, "*Neural networks for control systems - A survey*", Automatica, vol 28, no 6, 1992
- [Kawato 1988] Mitsuo Kawato, Yoji Uno, Michiaki Isobe, Ryogi Suzuki, "*Hierarchical neural network model for voluntary movement with application to robotics*", IEEE Control Sys. Mag., Apr. 1988
- [Kohonen 1987] Teuvo Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, 1987

- [Kohonen 1990] Teuvo Kohonen, "*The self-organizing map*", Proc. IEEE, Vol 78, No 9, Sept. 1990
- [Kraft 1990] L. Gordon Kraft, David P. Campagna, "A comparison between CMAC neural network control and two traditional adaptive control systems", IEEE Control Systems Mag., vol 10, no 3, Apr. 1990
- [Kuperstein 1989] Michael Kuperstein, Jorge Robinstein, "*Implementation of an adaptive neural controller for sensory-motor coordination*", IEEE Control Systems Mag., vol 9 no 3, Apr. 1989
- [Kurková 1992] Vera Kurková, "*Kolmogorov's theorem and multilayer neural networks*", Neural Networks, vol 5, pp501, 1992
- [Jong 1988] Kenneth de Jong, "*Learning with genetic algorithms: a overview*", Machine Learning 3, Kluwer Academic Publisher, 1988
- [Lane 1992] Stephen H. Lane, David A. Handelman, Jack J. Gelfand, "*Theory and development of higher-order CMAC neural networks*", IEEE Control Sys. Mag., Apr. 1992
- [Leonard 1991] James A. Leonard, Mark A. Kramer, "*Radial basis functions networks for classifying process fault*", IEEE Control Systems Mag., vol 11, no 3, Apr. 1991
- [Lippman 1987] Richard P. Lippmann, "*An Introduction to Computing with Neural Nets*", IEEE ASSP Mag, April 1987
- [Lippman 1988] Richard P. Lippmann, "*IS Over*", - 1988
- [Menczer 1990] Fillipo Menczer, Domenico Parisi, "*Evidence of hyperplanes in the genetic learning of neural networks*", Biological Cybernetics, no 66, 1990
- [Naidu 1990] Sinnasamy R. Naidu, Evangelos Zafiriou, Thomas J. McAvoy, "*Use of neural networks for sensor failure detection in a control system*", IEEE Control Systems Mag., 1990



- [Narendra 1980] Kumpati S. Narendra, Yuan-Hao Lin, "*Stable discrete adaptive control*", IEEE Trans. Autom. Control, Vol 25, No 3, Jun 1980
- [Narendra 1980b] Kumpati S. Narendra, Yuan-Hao Lin, Lena S. Valavani "*Stable adaptive controller design, part II: Proof of stability*", IEEE Trans. Autom. Control, Vol 25, no 3, Jun 1980
- [Narendra 1990] Kumpati S. Narendra, Kannan Parthasarathy, "*Identification and control of dynamical systems using neural networks*", IEEE Trans. Neural Networks, Vol 1, No 1, Mar 1990
- [Narendra 1991a] Kumpati S. Narendra, "*Intelligent control*", IEEE Control Syst. Mag., Jan. 1991
- [Narendra 1991b] Kumpati S. Narendra, Kannan Parthasarathy, "*Gradient methods for the optimization of dynamical systems containing neural networks*", IEEE Trans. Neural Networks, Vol 2, No 2, Mar 1991
- [Narendra 1991c] Kumpati S. Narendra, Snehasis Mukhopadhyay, "*Intelligent control using neural networks*", IEEE Control Syst. Mag., Apr. 1991
- [Nguyen 1990] Derrick H. Nguyen, Bernard Widrow, "*Neural networks for self-learning systems*", IEEE Control Systems Mag., vol 10, no 3, Apr. 1990
- [Ogata 1970] Katsuhiko Ogata, *Modern Control Eng.* , Prentice Hall, 1970
- [Passino 1989] Kevin M. Passino, Michael A. Sartori, Panos J. Antsaklis, "*Neural computing for numeric-to-symbol conversion in control systems*", IEEE Control Sys. Mag., Apr. 1989
- [Porto 1992] Vincent W. Porto, David B Fogel, "*Neural networks for AUV guidance control*", Sea Technology, Oct. 1992

- [Psaltis 1988] Demetri Psaltis, Athanasios Sideris, Alan A. Yamamoto, "*A multilayered neural network controller*", IEEE Control Sys. Mag., Apr. 1988
- [Rabelo 1992] Luis Rabelo, Xavier J. R. Avula, "*Hierarchical neuralcontroller architecture for robotic manipulation*", IEEE Control Sys. Mag., Apr. 1992
- [Rumelhart 1985] D. E. Rumelhart, D. Zipser, "*Feature discovery by competitive learning*", Cognitive Science, no 9, pp75, 1985
- [Rumelhart 1986] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "*Learning internal representations by error propagation*", Parallel Distributed Processing, Mit Press, 1986
- [Sanner 1990] Robert M. Sanner, David L Akin, "*Neuromorphic pitch attitude regulation of an underwater teleoperator*", IEEE Contr. Syst.Mag.,1990
- [Sartori 1992] Michael Sartori, Panos J. Antaklis, "*Implementations of learning control systems using neural networks*", IEEE Control Sys. Mag., Apr. 1992
- [Sutton 1992] Richard Sutton, Andrew G. Barto, Ronald J. Williams, "*Reinforcement Learning is direct adaptive optimal control*", IEEE Control Sys. Mag., Apr. 1992
- [Tagliarini 1991] G. A. Tagliarini, J. F. Christ, E. W. Page, "*Optimization using neural networks*", IEEE Trans. Computers, Vol 40, No 12, Dec. 1991
- [Tabary 1992] Guy Tabary, Isabelle Sallún, "*Control of a redundant articulated system by neural networks*", Neural Networks, vol 5, pp305 1992
- [Venugopal ??] K. P. Venugopal, A. S. Pandaya, R. Sudhakar, "*Adaptive neural networks controllers for autonomous under water vehicles*", ??, ??
- [Venugopal 1992] K. P. Venugopal, A. S. Pandaya, R. Sudhakar, "*On-line learnig control of autonomous underwater vehicles using feedforward neural networks*", IEEE Journal Oceanic Eng., vol 17, no 4, Oct 1992

- [Werbos 1974] Paul J. Werbos, "*Beyond regression: new tools for prediction and analysis in the behavior sciences*", Ph. D. Thesis, Harvard University, 1974
- [Werbos 1990] Paul J. Werbos, "*Backpropagation through time: what it does and how to do it*", Proc. IEEE, Vol 78, No 10, Oct. 1990
- [Werbos 1991] Paul J. Werbos, "*An overview of neural networks for control*", IEEE Control Systems Mag., vol 11, no 3, Apr. 1991
- [Yuh ] Junku Yuh, "*A learning control systems for underwater robotic vehicles*"
- [Yuh 1990] Junku Yuh, "*A neural net controller for underwater robotic vehicles*", IEEE Journal Oceanic Eng, vol 15, no3
- [Zorppette 1992] Glenn Zorppette, "*The power of parallelism*", IEEE Spectrum, Sept. 1992

Porto, Março 1993

Alfredo Manuel de Oliveira Martins



FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000101631