

Detecção e diagnóstico de falhas em robôs manipuladores via redes neurais artificiais

Renato Tinós

Orientador: Prof. Dr. Marco H. Terra

Sumário

- ▼ Introdução
- ▼ Detecção e diagnóstico de falhas (DDF) em sistemas dinâmicos
- ▼ Redes neurais artificiais (RNA)
- ▼ Robôs manipuladores
- ▼ Sistema de DDF via redes neurais artificiais
- ▼ Resultados
 - Manipulador planar com 2 graus de liberdade
 - Manipulador Puma 560
- ▼ Conclusões

Introdução

Falhas em sistemas dinâmicos:

- ▼ podem acarretar perdas de desempenho
- ▼ podem por em risco os equipamentos e o pessoal envolvido
- ▼ ocorrem: de modo abrupto ou de modo lento
- ▼ podem ser encobertas pela ação dos controladores
- ▼ pode ser entendida como qualquer tipo de mal funcionamento que ocasione perdas de desempenho
- ▼ Tolerância a falhas: redundância física e redundância em paralelo

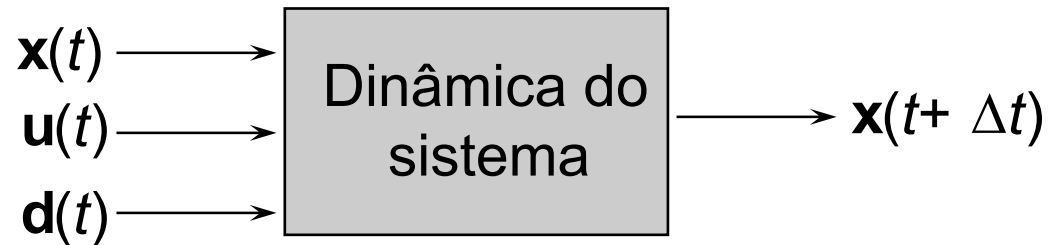
DDF em Sistemas Dinâmicos

- ▼ as técnicas de DDF utilizam processamento de informações das variáveis do processo
- ▼ através de DDF pode-se ter sistemas tolerantes a falhas: sistemas reconfiguráveis

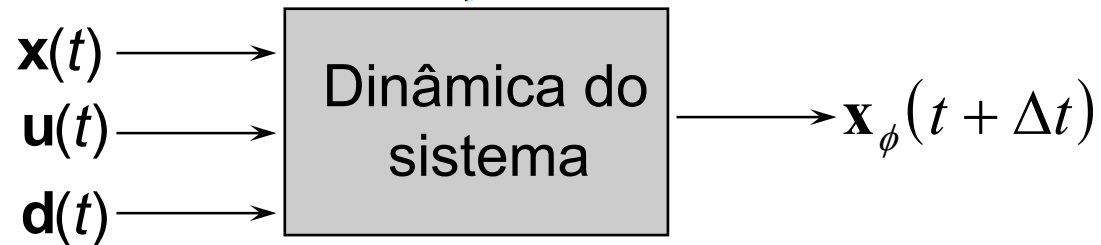
Técnicas de DDF:

- ▼ através de redundância analítica:
 - uso do modelo matemático (erros de modelagem)
- ▼ através de técnicas de inteligência artificial (IA)

Conceito de Geração de Resíduos

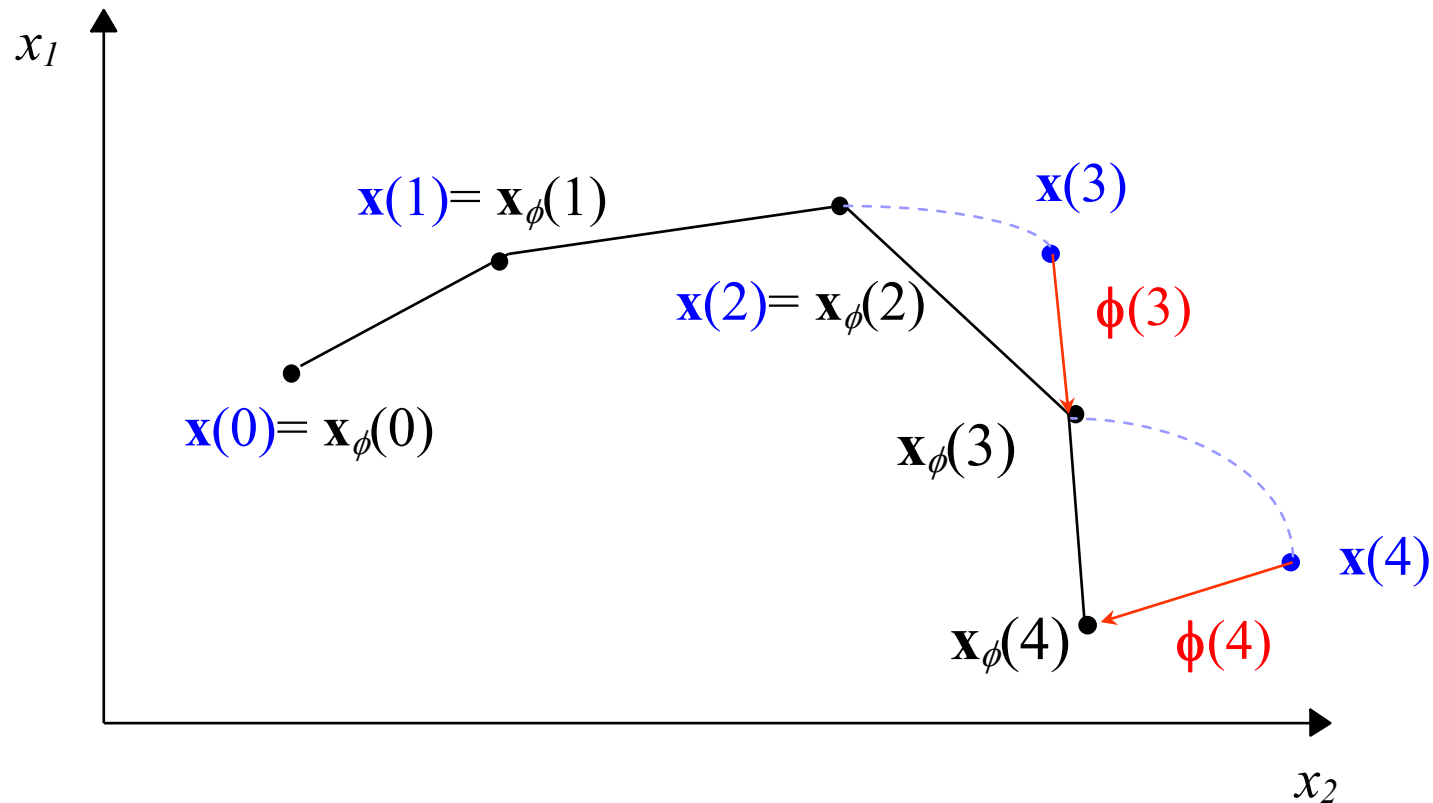


falha



Conceito de Geração de Resíduos

$$\phi(t + \Delta t) = \mathbf{x}_\phi(t + \Delta t) - \mathbf{x}(t + \Delta t)$$



DDF via redundância analítica

- ▼ uso do modelo matemático do sistema

Alguns enfoques:

- ▼ por paridade de estados
- ▼ por observadores dedicados
- ▼ por identificação paramétrica

Problemas:

- ▼ erros de modelagem
- ▼ a modelagem pode ser difícil ou impossível

Soluções:

- ▼ uso de técnicas robustas

DDF utilizando técnicas de IA

- ▼ baseada na classificação das variáveis medidas
- ▼ baseada no conceito de geração de resíduos

Técnicas de IA utilizadas:

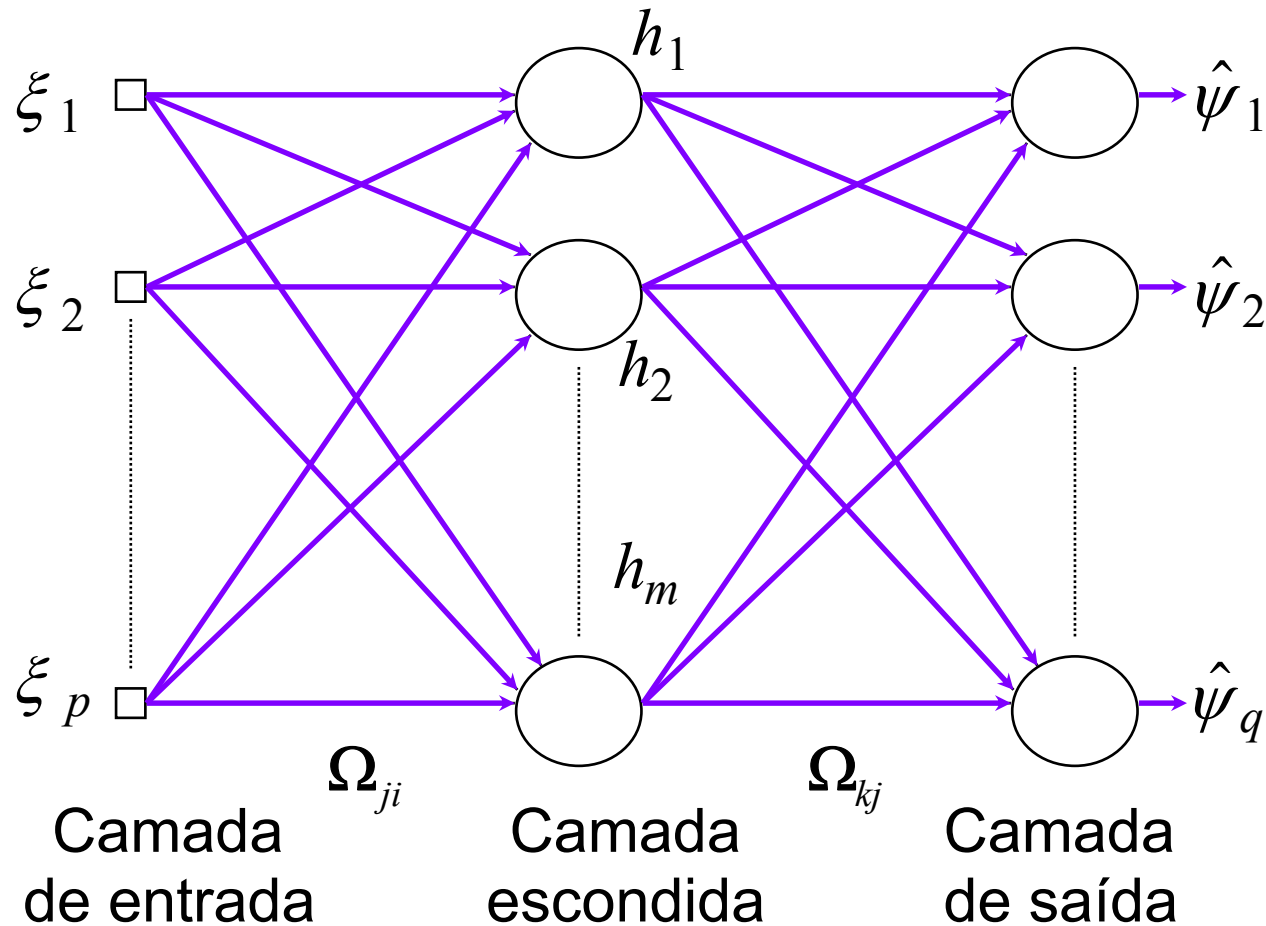
- ▼ sistemas especialistas
- ▼ lógica nebulosa (*fuzzy*)
- ▼ redes neurais artificiais (RNA)

Usando RNA e o conceito de geração de resíduos:

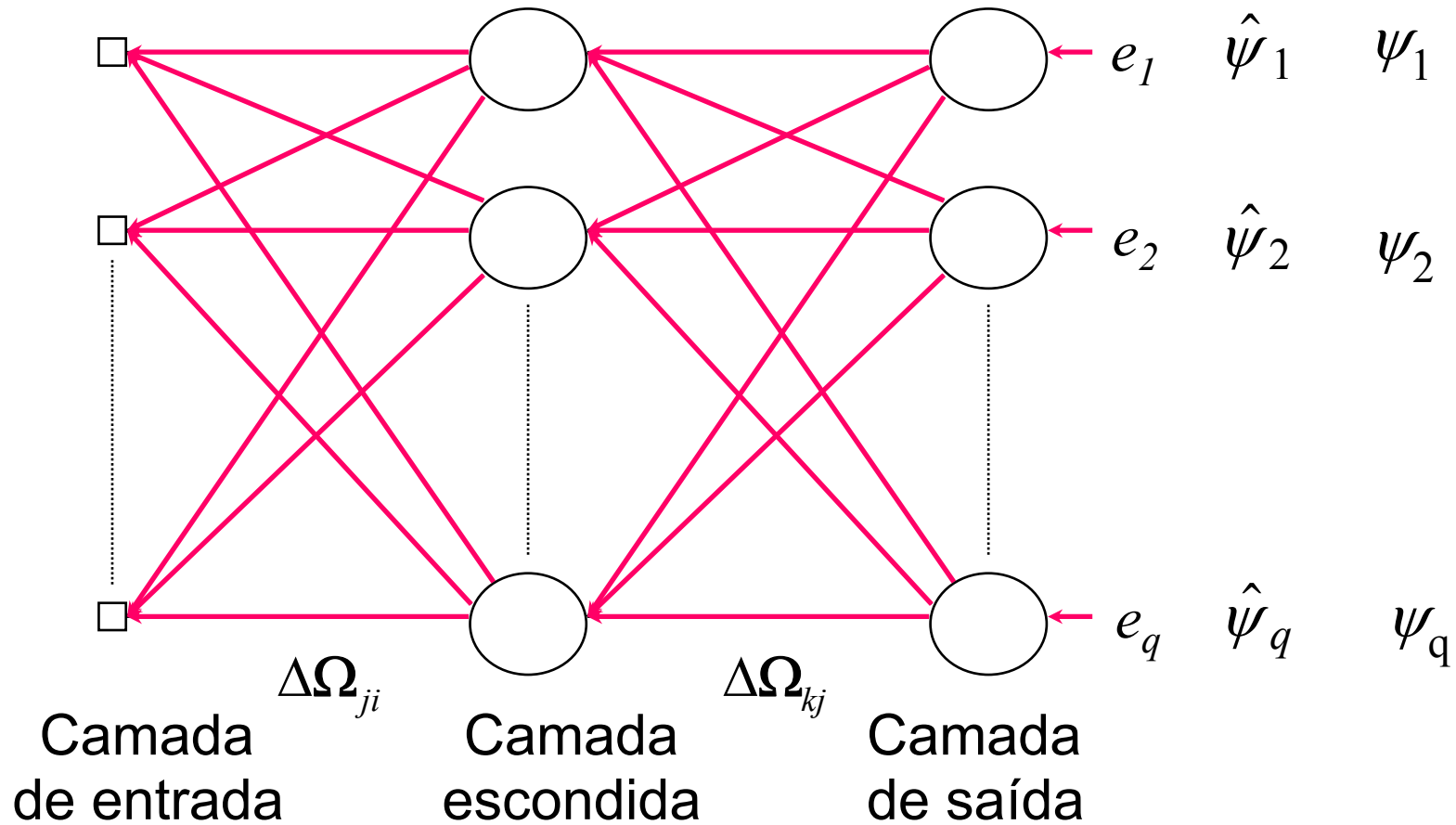
- ▼ Mapeamento: RNA como aproximador de funções
- ▼ Análise dos resíduos: RNA como classificador

Redes neurais artificiais

- ▼ *perceptron* multicamadas (MLP) treinado por retropropagação do erro (*backpropagation*)



Perceptron multicamadas



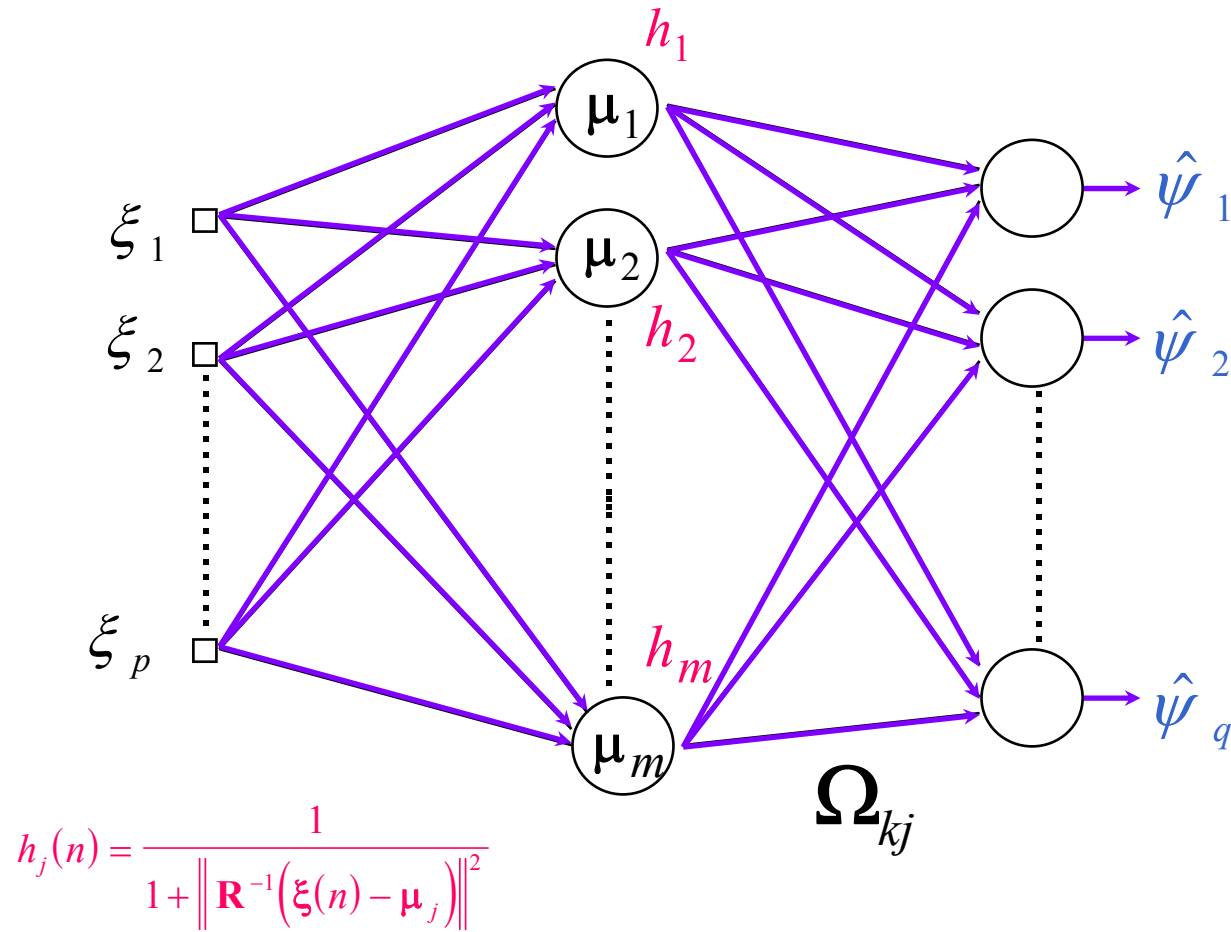
Perceptron multicamadas

- ▼ a relação entrada/saída do MLP define um mapeamento de um espaço p -dimensional para um espaço q -dimensional, que é continuamente diferenciável - teorema A1 [CYBENKO, 1989].
- ▼ Para o problema de classificação o MLP com treinamento por retropropagação produz bordas de decisão.

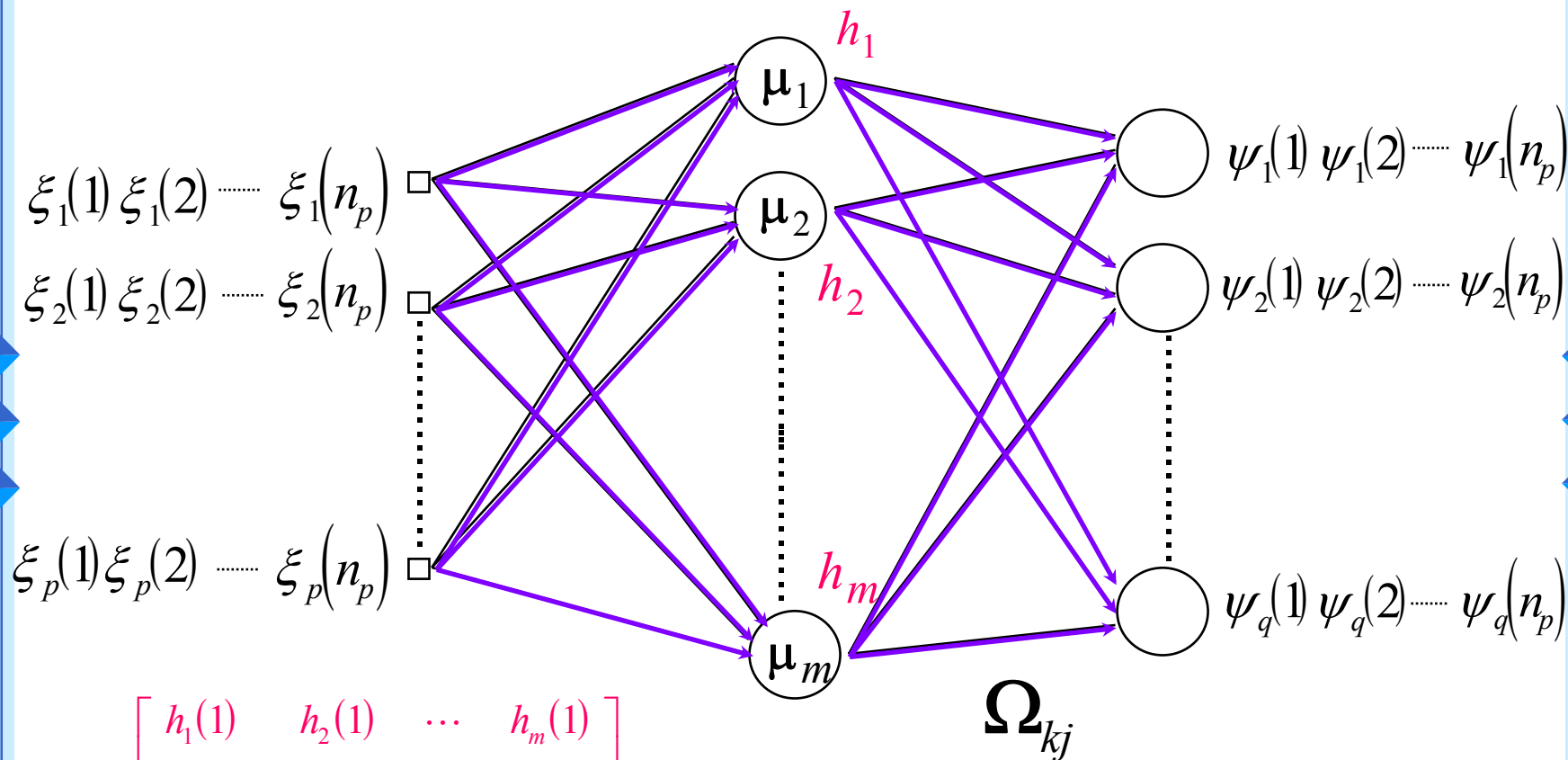
Problemas para a classificação em DDF:

- ▼ as bordas não estão em posições conservadoras
- ▼ nas áreas sem padrões de treinamento a classificação é arbitrária

Rede com Função de Base Radial (Rede RBF)



Rede com Função de Base Radial (Rede RBF)



$$\mathbf{H} = \begin{bmatrix} h_1(1) & h_2(1) & \cdots & h_m(1) \\ h_1(2) & h_2(2) & \cdots & h_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(n_p) & h_2(n_p) & \cdots & h_m(n_p) \end{bmatrix}$$

Rede RBF

- ▼ a classificação é feita de acordo com a distância do padrão a ser classificado e os vetores de centros
- ▼ o treinamento é rápido pois o modelo, após a determinação das unidades radiais, é linear
- ▼ inexistência de mínimos locais no cálculo dos pesos
- ▼ pode ser usada também para aproximação de funções, no entanto, espaços de entradas com dimensão alta podem tornar o treinamento complicado
- ▼ aqui, 4 métodos de treinamento foram utilizados

1) Forward Selection (FS)

- ▼ emprega seleção de subconjuntos para escolher os centros das unidades radiais a partir dos padrões de treinamento
- ▼ começa com um subconjunto vazio e adiciona em cada passo o padrão que mais reduz a soma do erro médio quadrático
- ▼ a seleção termina quando um critério de parada é alcançado (aqui, o erro GCV)
- ▼ depois da escolha dos centros, a matriz de pesos pode ser calculada

$$\hat{\Omega} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \Psi$$

2) Global ridge regression (GRR)

- ▼ é equivalente à decaimento de pesos, usado em RNA
- ▼ emprega todos os padrões de treinamento como centros das unidades radiais, mas penaliza os pesos grandes
- ▼ um único termo de penalidade (λ) é aplicado em todas as unidades radiais

$$C_k = \sum_{n=1}^{n_p} \left(\psi_k(n) - \hat{\psi}_k(n) \right)^2 + \lambda \sum_{j=1}^m \omega_{kj}^2$$

$$\hat{\Omega} = \left(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}_m \right)^{-1} \mathbf{H}^T \Psi$$

3) Local ridge regression (LRR)

- ▼ LRR aplica em cada unidade radial um termo de penalidade diferente

$$C_k = \sum_{n=1}^{n_p} \left(\psi_k(n) - \hat{\psi}_k(n) \right)^2 + \sum_{j=1}^m \lambda_j^2 \omega_{kj}^2$$

$$\hat{\Omega} = \left(\mathbf{H}^T \mathbf{H} + \Lambda \right)^{-1} \mathbf{H}^T \Psi$$

Λ é uma matriz diagonal formada pelos termos de penalidade individuais

4) Mapa Auto-organizável de Kohonen (MAOK)

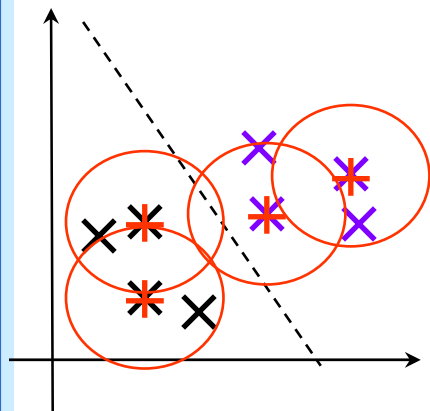
- ▼ o conjunto de treinamento é separado de acordo com as diferentes classes
- ▼ inicialmente todos os padrões de cada classe são escolhidos como centros das unidades radiais
- ▼ para cada classe, a unidade com a maior ativação é selecionada

$$h_c(t) = \max_j \{h_j(t)\}$$

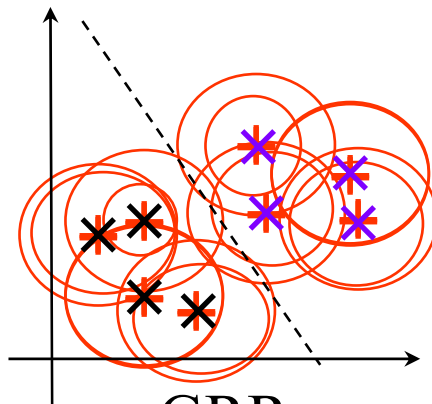
- ▼ o centro da unidade radial é atualizado de acordo com

$$\mu_j(t+1) = \mu_j(t) + \alpha(t)\beta(t)[\xi(t) - \mu_j(t)]$$

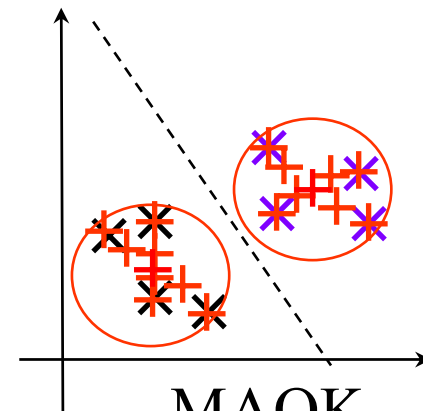
- ▼ se o número de iterações é suficientemente grande e os parâmetros são escolhidos apropriadamente, os centros das unidades radiais nos mesmos aglomerados deverão se mover para a mesma posição
- ▼ como várias unidades radiais têm centros nas mesmas posições, estas unidades podem ser agrupadas
- ▼ assim, a complexidade da rede é reduzida já que o número de parâmetros adaptativos decresce
- ▼ o próximo passo é agrupar as unidades radiais de cada classe em uma única rede RBF e calcular a matriz de pesos ótima



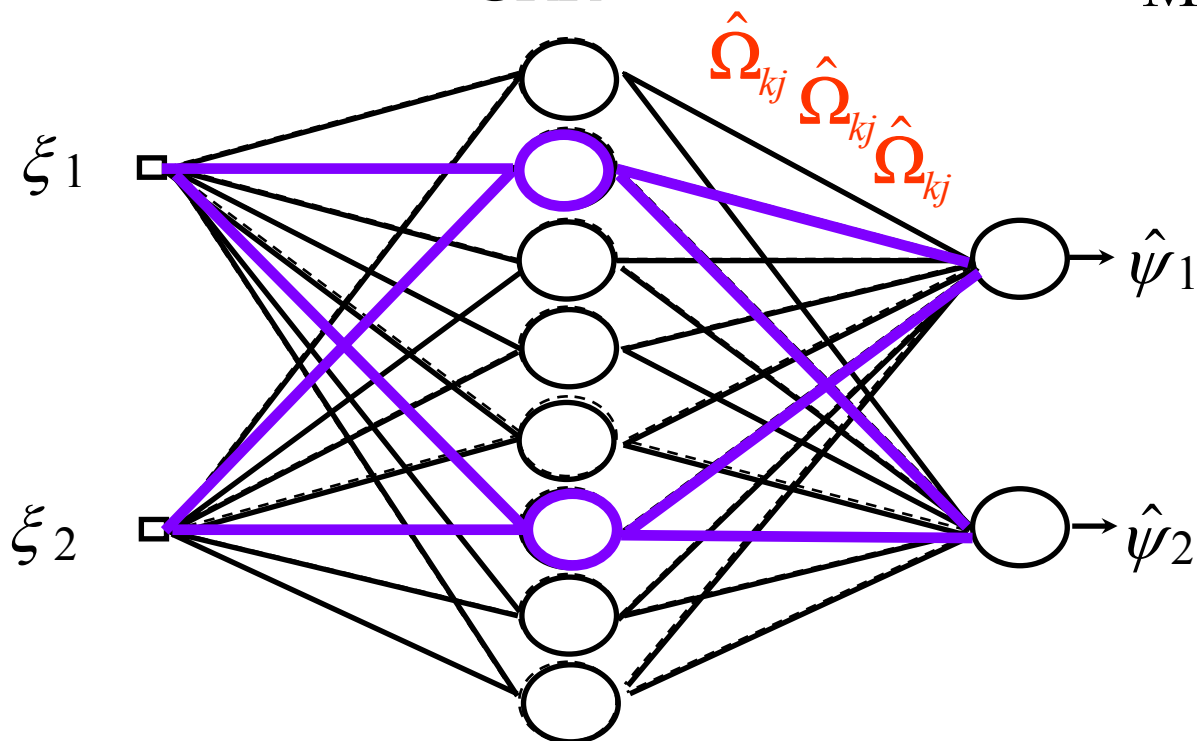
FS



GRR



MAOK



Robôs manipuladores

A dinâmica de um robô manipulador é dada por:

$$\ddot{\theta} = \mathbf{M}(\theta, t)^{-1} [\tau - \mathbf{v}(\theta, \dot{\theta}, t) - \mathbf{g}(\theta, t) - \mathbf{z}(\theta, \dot{\theta}, t) - \mathbf{d}(t)]$$

Falhas em robôs manipuladores:

- ▶ podem gerar movimentos descontrolados que podem causar sérios danos ao robô e ao ambiente de trabalho
- ▶ Em medicina, exploração espacial e ambientes hostis, as falhas podem causar acidentes irreversíveis

DDF em robôs manipuladores

- ▼ geralmente utiliza-se redundância analítica para gerar os resíduos

Para análise dos resíduos, pode-se citar:

- ▼ utiliza-se *threshold* fixos: problemas de alarmes falsos
- ▼ utiliza-se *thresholds* variáveis: (Visinsky *et al.*, 1995), (Schneider e Frank, 1996) e (Naughton *et al.*, 1996)

O esquema aqui utilizado faz uso de um MLP para gerar os resíduos e uma rede RBF para produzir *thresholds* variáveis

DDF em robôs manipuladores via RNA

Aplicando-se a Integração de Euler na eq. dinâmica:

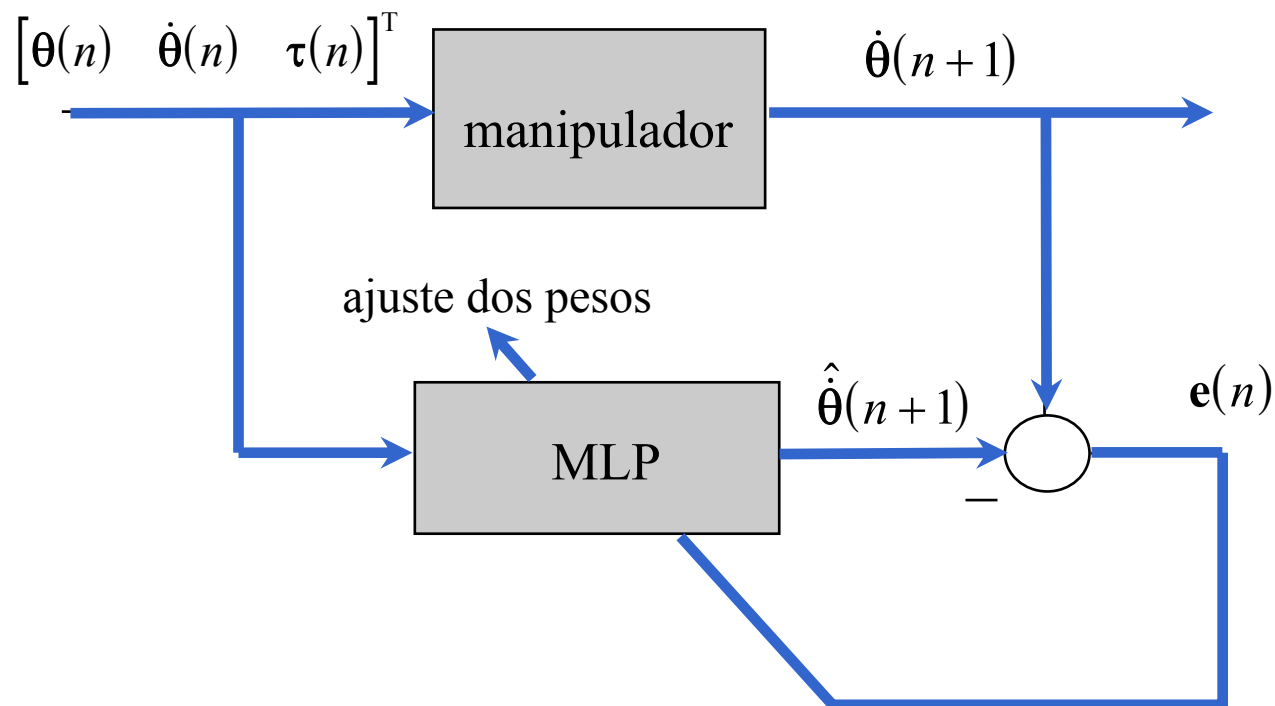
$$\dot{\theta}(t + \Delta t) = \mathbf{M}(\theta, t)^{-1} [\tau - \mathbf{v}(\theta, \dot{\theta}, t) - \mathbf{g}(\theta, t) - \mathbf{z}(\theta, \dot{\theta}, t) - \mathbf{d}(t)] \Delta t + \dot{\theta}(t)$$

- ▶ O MLP deve reproduzir a função não-linear:

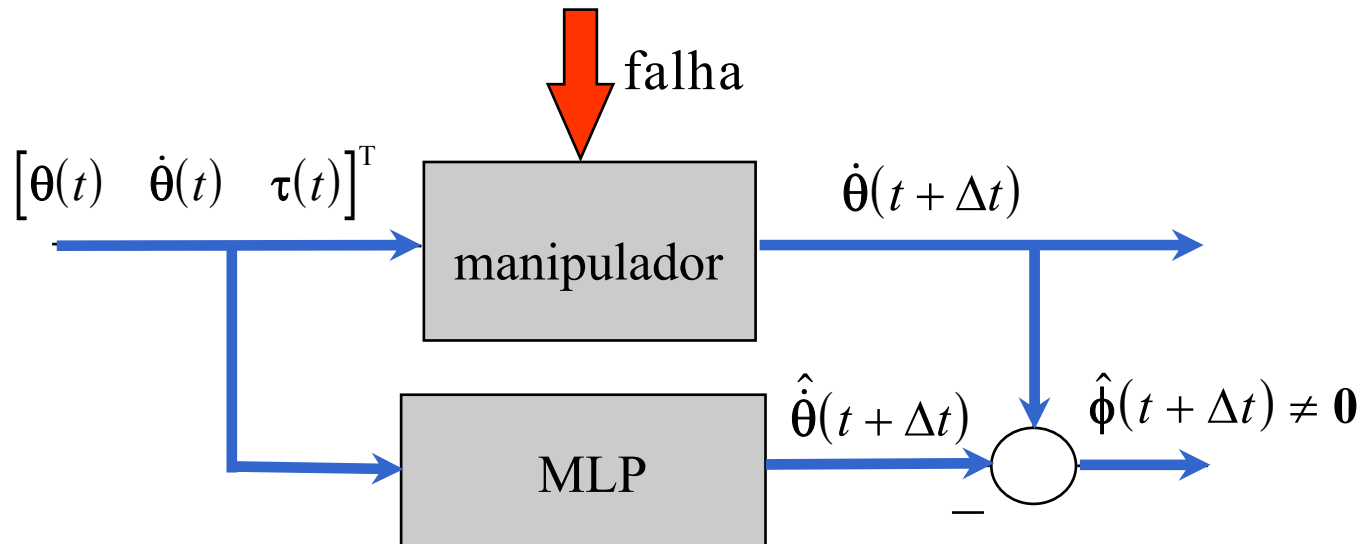
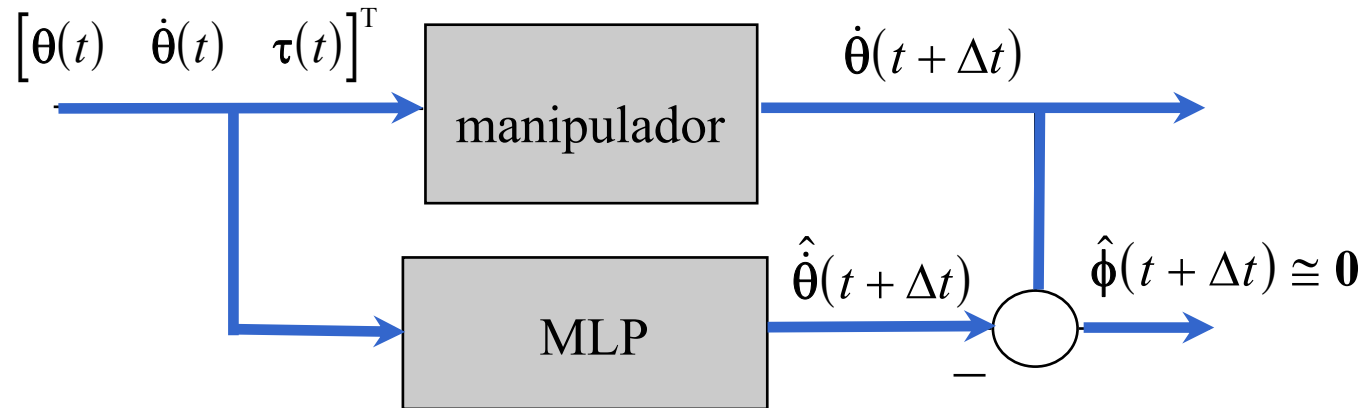
$$\dot{\theta}_{t+\Delta t} = f(\theta_t, \dot{\theta}_t, \tau_t)$$

- ▶ A rede RBF tem como entradas os dados do resíduo e de velocidades nas juntas e como saídas o vetor de estados das falhas

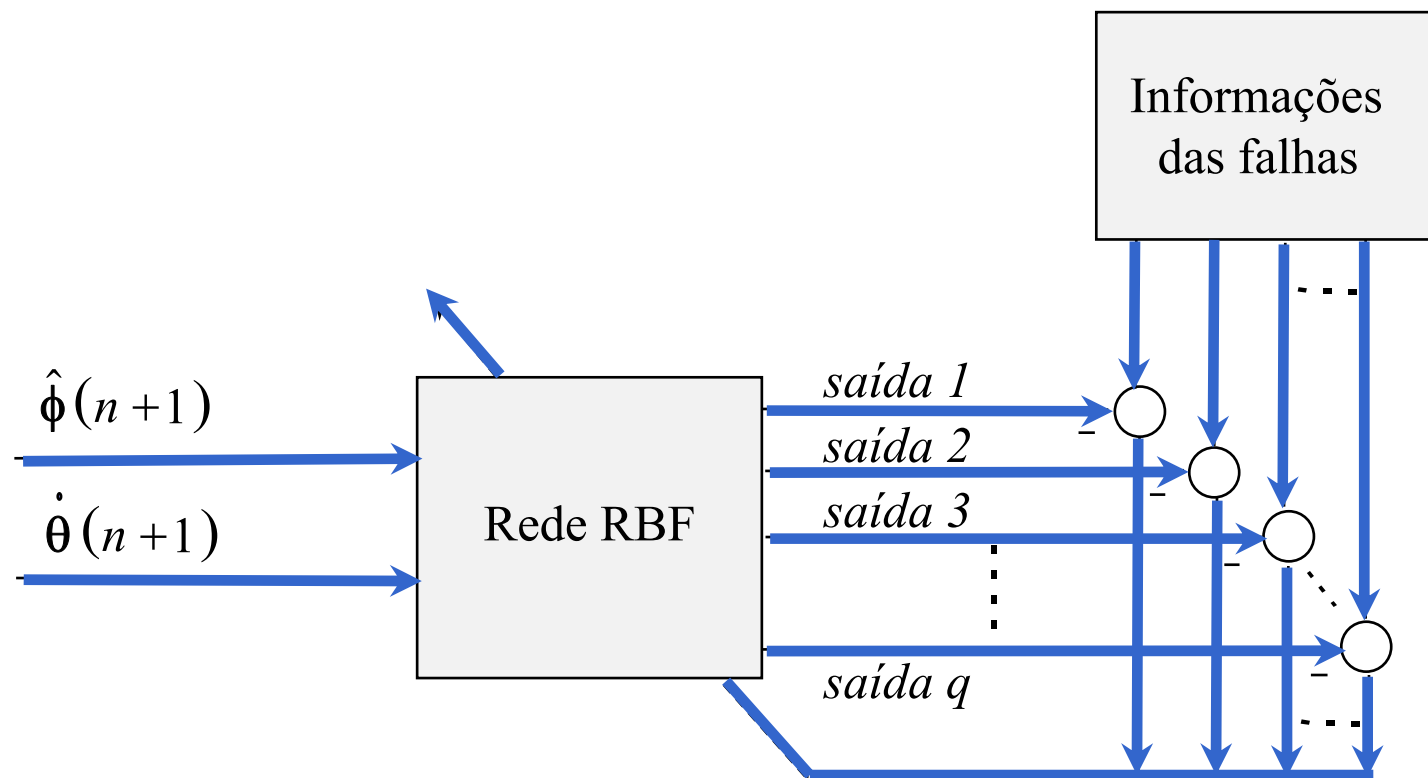
Geração de Resíduos - Treinamento



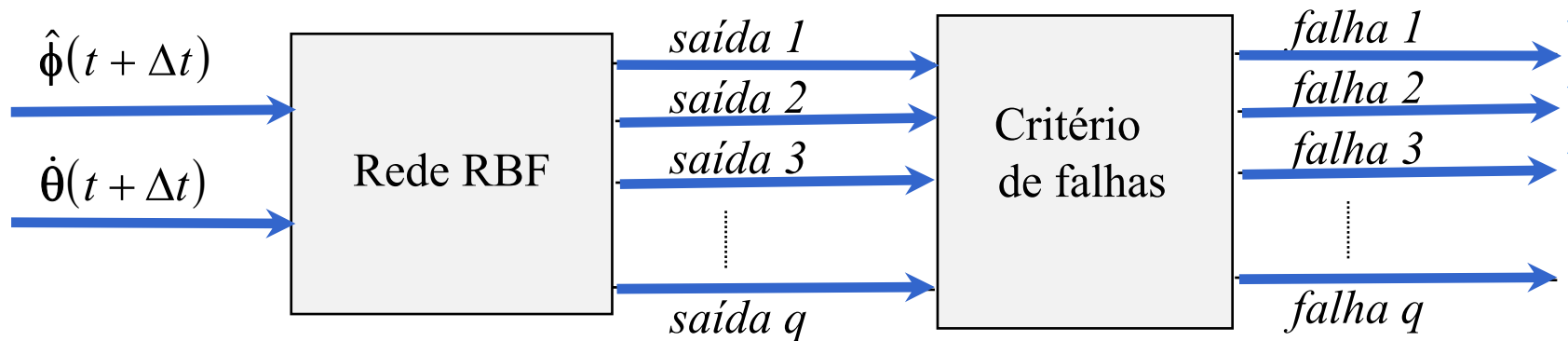
Geração de Resíduos - Operação



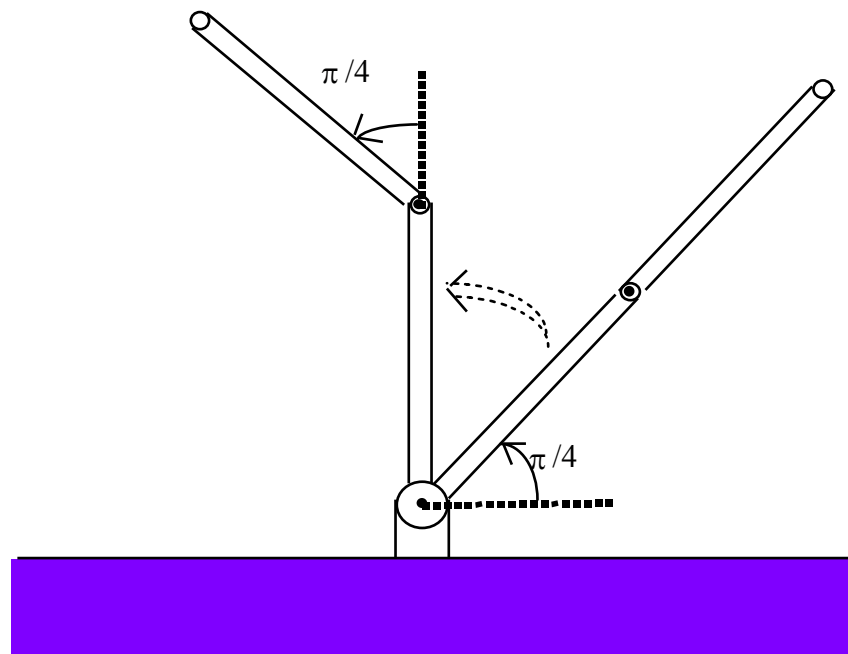
Análise dos Resíduos - Treinamento



Análise dos Resíduos - Operação



Resultados: robô manipulador planar com 2 graus de liberdade



- ▼ O MLP é treinado com 10 trajetórias com 50 amostras cada

Resultados: robô manipulador planar com 2 graus de liberdade

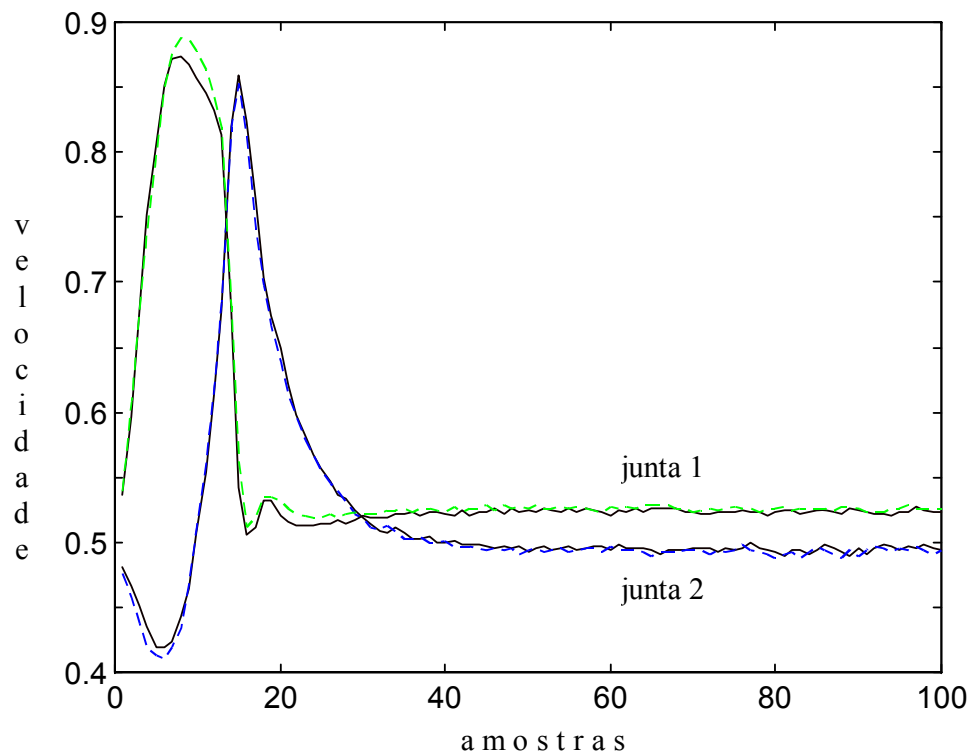


Figura. Velocidades das juntas (tracejadas) e saídas do MLP (contínuas) para uma trajetória não-treinada livre de falhas

Resultados - Robo com 2 graus de liberdade

Falhas:

- ▼ Falha 1: travamento da junta 1
- ▼ Falha 2: travamento da junta 2

Critério de falhas adotado: 5 saídas consecutivas da rede RBF maiores que 0,5

- ▼ A rede RBF é treinada apresentado-se 9 trajetórias com 40 amostras cada para os dois tipos de falha e para operação normal (Total: 27 trajetórias)
- ▼ Foram usados os 3 métodos para treinamento da rede RBF: 1 único alarme falso nos testes.

Resultados: Rede RBF

Erro médio quadrático da rede RBF para o conjunto de treinamento.

	saída 1 - rede RBF (Falha 1)	saída 2 - rede RBF (Falha 2)
FS	0.0008	0.0042
GRR	0.0015	0.0066
LRR	0.0013	0.0058
MAOK	0.0100	0.0212

Erro médio quadrático da rede RBF para o conjunto de teste.

	saída 1 - rede RBF (Falha 1)	saída 2 - rede RBF (Falha 2)
FS	0.0699	0.2727
GRR	0.0354	0.0829
LRR	0.0414	0.1233
MAOK	0.0273	0.0788

Resultados: método *global ridge regression*

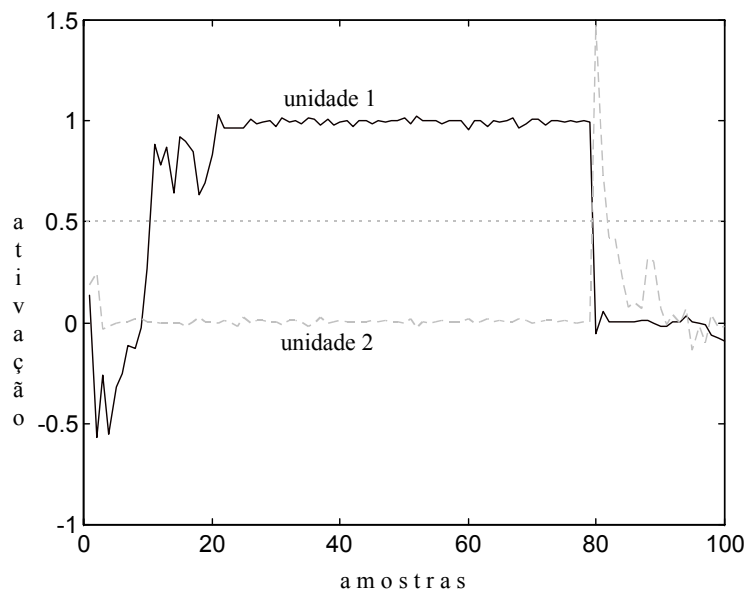
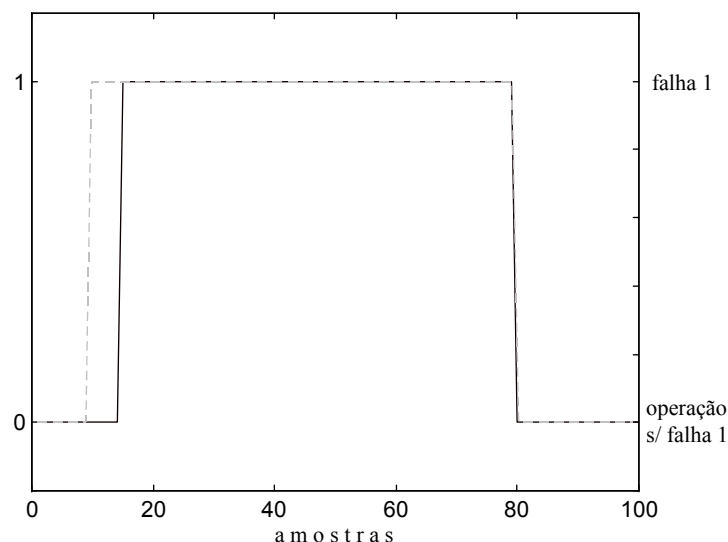


Figura. Saídas da rede RBF para uma trajetória não-treinada. A falha 1 ocorre entre as amostras 10 e 80

Figura. Detecção da falha 1 para a saída 1 da rede RBF vista na Figura ao lado



Resultados: método *local ridge regression*

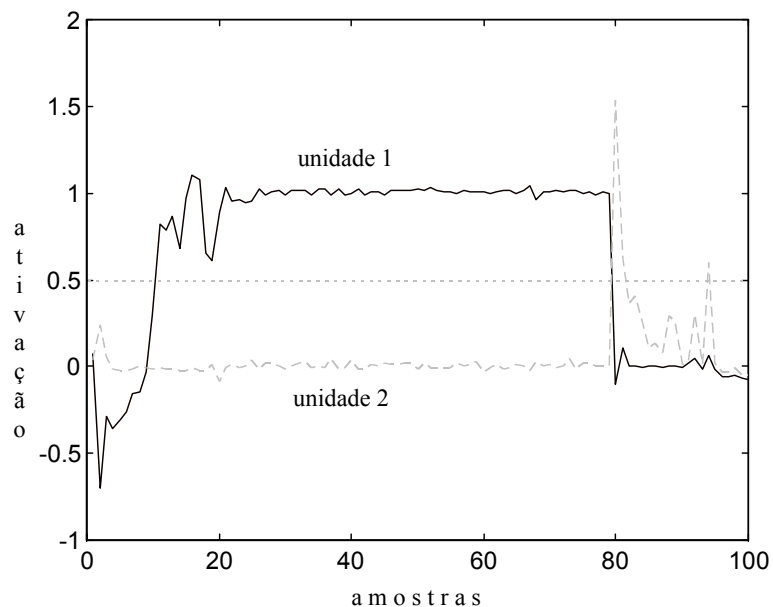
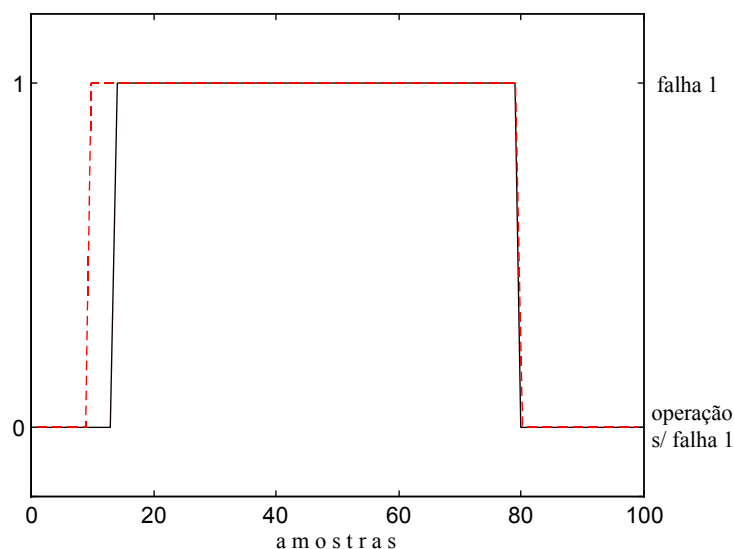


Figura. Saídas da rede RBF para uma trajetória não-treinada. A falha 1 ocorre entre as amostras 10 e 80

Figura. Detecção da falha 1 para a saída 1 da rede RBF vista na Figura ao lado



LRR

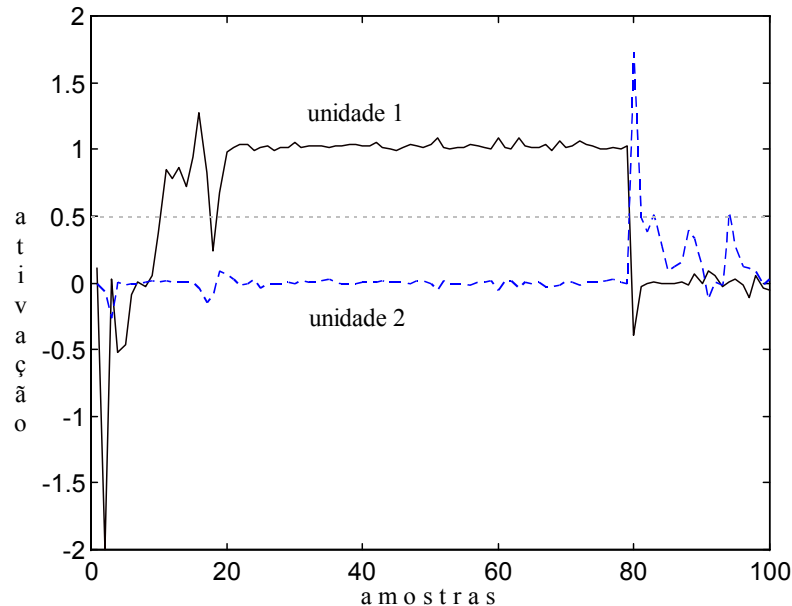


Figura. Saídas da rede RBF para uma trajetória não-treinada. A falha 1 ocorre entre as amostras 10 e 80

MAOK

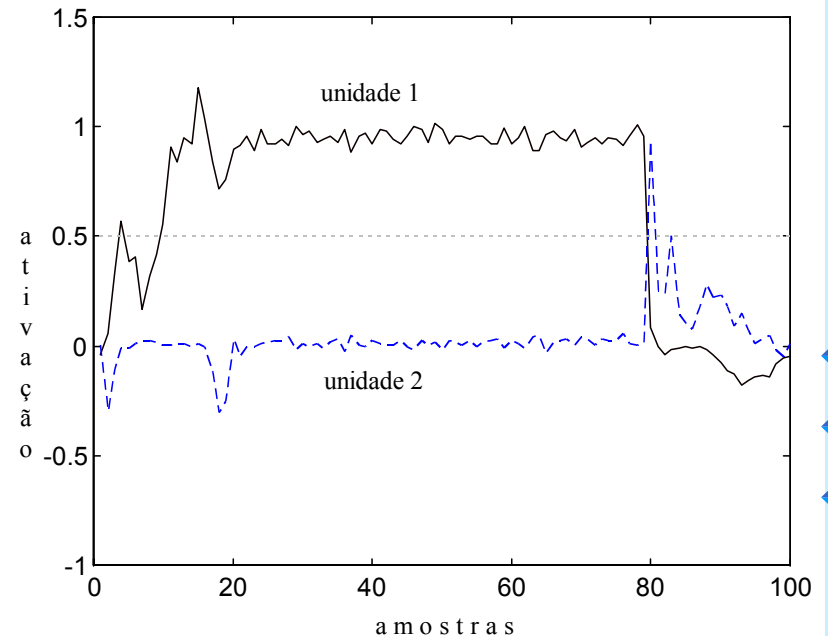
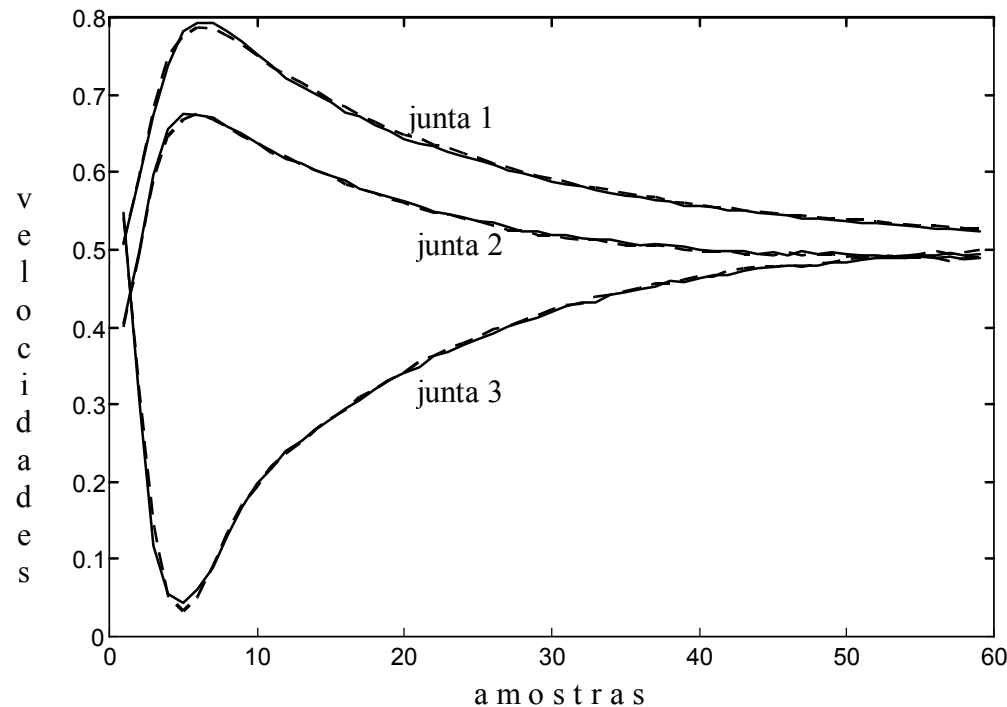


Figura. Saídas da rede RBF para uma trajetória não-treinada. A falha 1 ocorre entre as amostras 10 e 80

RESULTADOS: PUMA 560

- ▼ o MLP é treinado com 10 trajetórias com 50 amostras cada
- ▼ o MLP tem 9 entradas, 29 neurônios na camada escondida e 3 neurônios de saída



Resultados - Puma 560

- ▼ Falhas consideradas:
 - Falha 1: torque não é aplicado na junta 1
 - Falha 2: torque não é aplicado na junta 2
 - Falha 3: torque não é aplicado na junta 3
- ▼ A rede RBF foi treinada com 15 trajetórias com 12 amostras cada para as 3 falhas e para operação normal (720 padrões)
- ▼ Para os teste de validação, 30 trajetórias não-treinadas com 15 amostras cada são apresentadas 4 vezes (1800 padrões)
- ▼ critério de falhas: 3 amostras consecutivas $> 0,5$

Resultados: Rede RBF

Erro médio quadrático da rede RBF para o conjunto de treinamento.

	saída 1 - rede RBF (Falha 1)	saída 2 - rede RBF (Falha 2)	saída 3 - rede RBF (Falha 3)
FS	0.0040	0.0017	0.0017
GRR	0.0155	0.0075	0.0119
LRR	0.0061	0.0022	0.0030
MAOK	0.0096	0.0042	0.0092

Erro médio quadrático da rede RBF para o conjunto de teste.

	saída 1 - rede RBF (Falha 1)	saída 2 - rede RBF (Falha 2)	saída 3 - rede RBF (Falha 3)
FS	0.0712	0.0431	0.0347
GRR	0.0419	0.0371	0.0271
LRR	0.7930	241.7819	2.6576
MAOK	0.0462	0.0400	0.0293

	número de alarmes falsos	número de falhas não-detectadas
FS	8	2
GRR	1	0
LRR	7	1
MAOK	4	0

Resultados - Puma 560

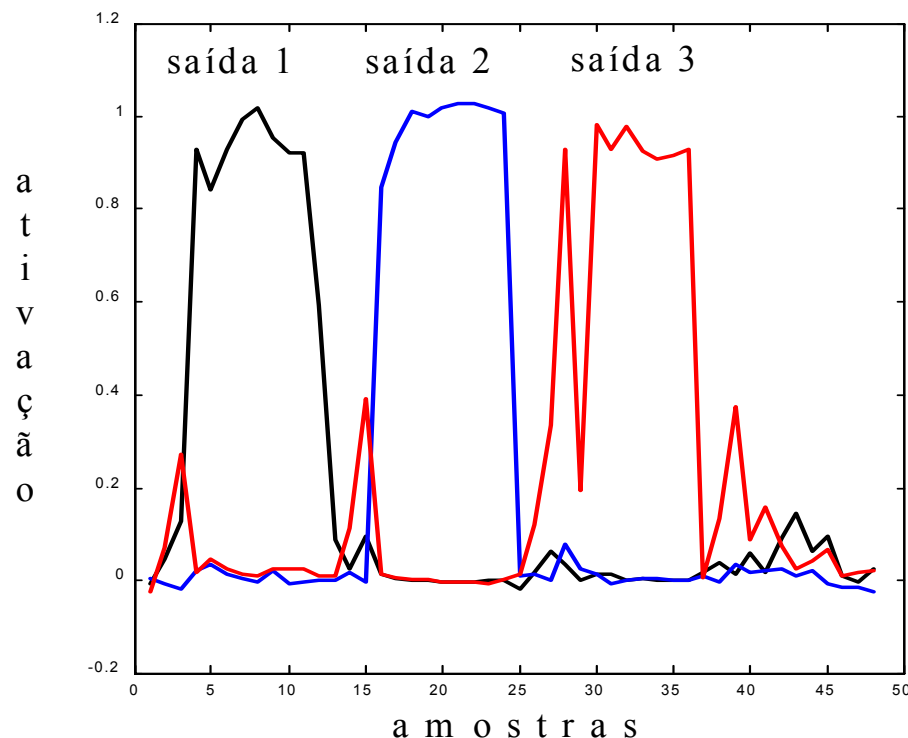


Figura. Saídas da rede RBF para 4 simulações de uma mesma trajetória não-treinada (12 amostras). Nas três primeiras simulações um tipo diferente de falha ocorre.

Conclusões

- ▼ o sistema de DDF via RNA apresenta bons resultados quando aplicado em um robô manipulador planar com 2 braços e no Puma 560
- ▼ o grande atrativo é que o sistema consegue detectar falhas que ocorrem em trajetórias não-treinadas
- ▼ para os métodos MAOK, GRR e LRR, as redes RBF apresentam sinais de saída mais suaves do que as redes treinadas pelo método FS
- ▼ o custo computacional do LRR é bastante alto
- ▼ o menor custo computacional ocorre no MAOK

Conclusões

- ▼ o método MAOK apresenta bons resultados quando comparados com os outros métodos
- ▼ um resultado interessante é que as redes treinadas pelo MAOK apresentam menor número de unidades radiais
- ▼ falhas cujo sistema apresente comportamentos semelhantes são difíceis de serem isoladas