

-
-
-
-
-

IF-705 – Automação Inteligente

Redes Neurais para Modelagem e Controle

Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
aluizioa@cin.ufpe.br



-
-
-
-
-

Sumário

- Introdução
- Neuro-Modelagem e Arquitetura de Controle
- Estratégias de Identificação
 - Modelo Básico (ou Direto) da Planta (MDP);
 - Modelo Inverso da Planta (MIP);
 - Modelo Inverso Especializado da Planta (MIEP);
 - Modelo de Operador (MO).
- Identificação: Representação
- Identificação por Redes Neurais Supervisionadas

Introdução

- Modelagem e controle de sistemas empregam cada vez mais algoritmos de inteligência computacional;
- Aplicações em Controle Inteligente (CI) requerem algoritmos capazes de:
 - *Operar* em ambiente mal-definido e variante no tempo;
 - *Adaptar-se* a mudanças na dinâmica da planta ou do processo bem como ao efeitos do ambiente;
 - *Aprender* informações relevantes do meio-ambiente;
 - *Adicionar* poucas restrições à dinâmica da planta.

Introdução

- As características anteriores são essenciais para operação de sistema em ambientes incertos, com o mínimo de intervenções externas.
- O aprendizado humano possui as características desejadas. O desafio é dotar as máquinas com tais características.
- A capacidade de adaptação autônoma da computação inteligente oferece à área de Engenharia de Controle (EC):
 - Melhoria no desempenho de sistemas de controle devido à sua capacidade de adaptação;
 - Aumento da qualidade da solução;
 - Possibilidade de autonomia para os sistemas;
 - Diminuição dos custos de projeto e custos operacionais.

Introdução

- Em geral, algoritmos de inteligência computacional utilizam:
 - Linearização da planta e modelos de controladores;
 - Definição de diferentes valores de parâmetros para permitir flexibilidade e adaptação ao sistema.
- RNAs podem ser aplicadas a sistemas *não-lineares*.
- Um sistema de controle inteligente pode ser implementado com diferentes tipos de RNAs e de algoritmos de aprendizagem:
 - A escolha do sistema de aprendizagem (abordagem e arquitetura) depende fundamentalmente do problema em questão.

Neuro-Modelagem e Arquitetura de Controle

- Blocos de aprendizagem são empregados nos algoritmos de modelagem e controle neuro-inspirados.
- Modelagem e arquiteturas de controle são em geral independentes da abordagem de aprendizagem a ser empregada:
 - Algumas abordagens podem ser mais adequadas que outras;
 - A habilidade para modelagem e aprendizagem determina a aplicabilidade de uma escolha.
- O grau de satisfação de um algoritmo de aprendizagem com respeito aos pontos acima determina sua adequação para modelagem e controle por aprendizagem online.

Neuro-Modelagem e Arquitetura de Controle

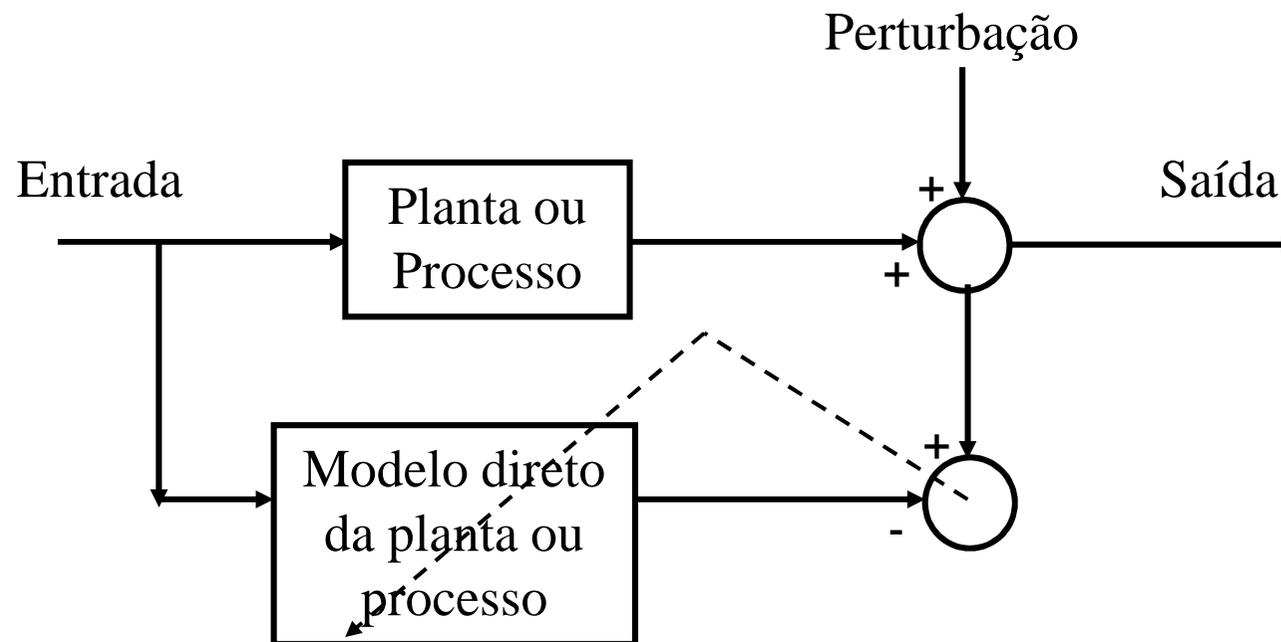
- Em controle convencional, a modelagem da planta ou processo se utiliza de conhecimento de física, química ou biologia:
 - O modelo deve ser acurado para que o controlador possa assegurar estabilidade e outras características relevantes ao sistema.
- Alguns sistemas que podem não ter modelos acurados:
 - Processos químicos;
 - Robôs em ambiente não-estruturado;
 - Aeronave em movimento sujeita a forças incertas.
- Alternativa: Usar dados experimentais, entrada-saída da planta ou processo. A determinação de um modelo vindo de dados experimentais se chama identificação de sistemas.

Estratégias de Neuro-identificação

- Há quatro arquiteturas principais nos módulos de aprendizagem para modelar a planta:
 - Modelo Básico (ou Direto) da Planta (MDP);
 - Modelo Inverso da Planta (MIP);
 - Modelo de Operador (MO);
 - Modelo Inverso Especializado da Planta (MIEP).
- O sinal de entrada deve varrer todo o espaço de entrada da rede.
- Nos três primeiros casos, a saída obtida é comparada à desejada, produzida pela planta. Portanto, aprendizagem supervisionada pode ser utilizada para treinar os pesos da rede.
 - Apenas no Modelo Inverso Especializado da Planta, a saída da rede é a entrada da planta e o erro é considerado na saída desta.

Estratégias de Neuro-identificação: Modelo Direto da Planta

- Arquiteturas do Modelo Direto da planta ou processo:



Estratégias de Neuro-identificação: Modelo Direto da Planta

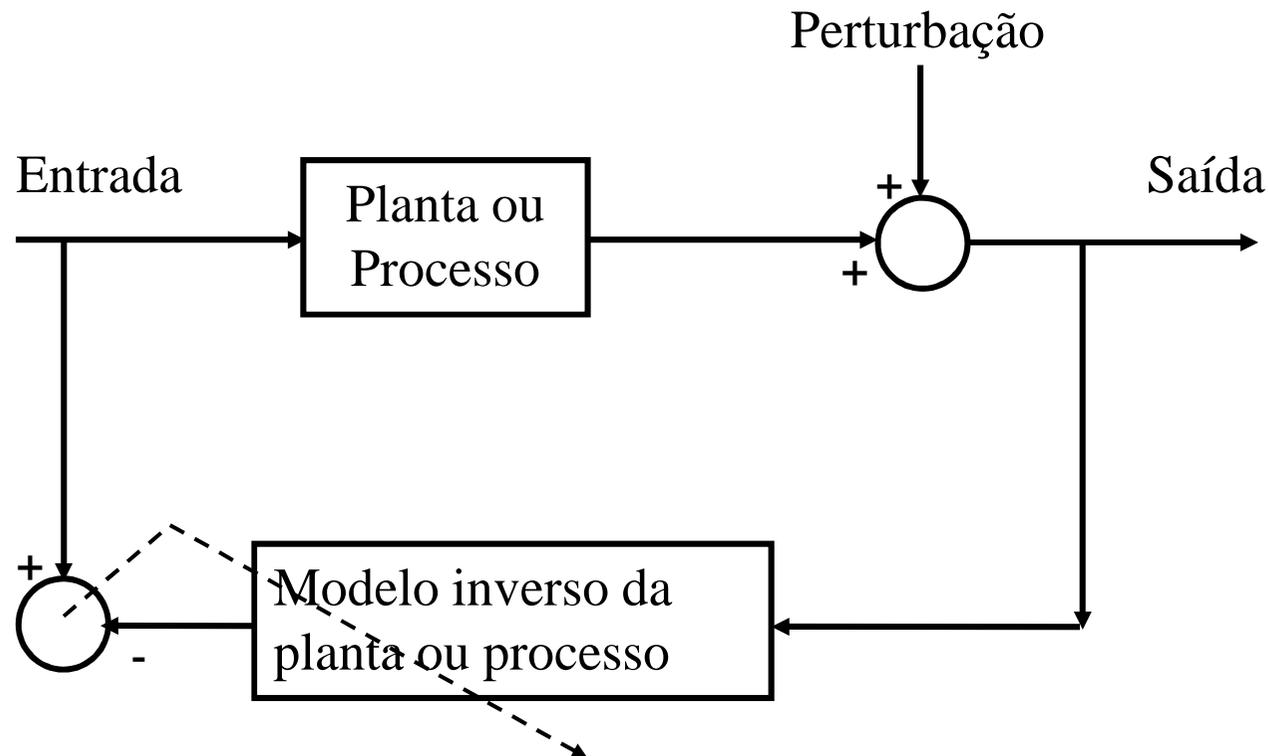
- Modelagem Básico da Planta (Modelo Direto da Planta)
 - O objetivo é modelar (identificar) o funcionamento da planta ou processo considerando suas entradas e saídas como pares para o treinamento.
 - A função de custo busca minimizar a esperança do erro quadrático:

$$J_y = E(\varepsilon_y^2(t)), \quad \varepsilon_y(t) = \hat{y}(t) - y(t)$$

onde $y(t)$ é a saída da rede e $\hat{y}(t)$ é a saída da planta.

Estratégias de Neuro-identificação: Modelo Inverso da Planta

- Arquiteturas do Modelo Inverso da planta ou processo:



Estratégias de Neuro-identificação: Modelo Inverso da Planta

- Modelagem Inversa da Planta
 - O objetivo é formular um *controlador* de forma que a arquitetura como um todo (controlador/planta) tenha uma unidade de função de transferência;
 - A planta precisa ser total ou localmente inversível;
 - A função de custo do controle tenta minimizar a esperança do erro quadrático:

$$J_u = E(\varepsilon_u^2(t)), \quad \varepsilon_u(t) = \hat{u}(t) - u(t)$$

onde $u(t)$ é a saída de controle da rede e $\hat{u}(t)$ é o sinal de controle medido.

Estratégias de Neuro-identificação: Modelo Inverso da Planta

- Modelagem Inversa da Planta:
 - O erro do sinal de controle é relacionado com o erro da planta de um sistema SISO por uma primeira aproximação:

$$\varepsilon_y(t) \approx \frac{dy(t)}{du(t)} \varepsilon_u(t)$$

onde dy/du é o Jacobiano da planta.

- Então a função de custo da resposta da Planta será:

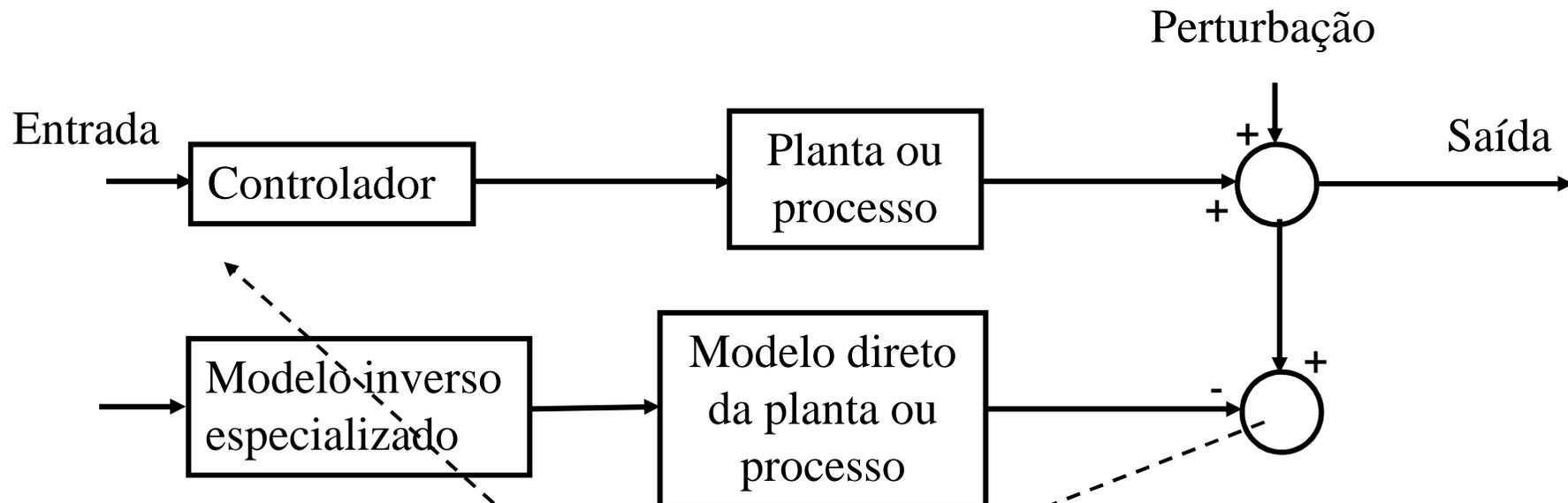
$$J_y = E(\varepsilon_y^2(t)) \approx E\left[\left(\frac{dy(t)}{du(t)}\right)^2 \varepsilon_u^2(t)\right]$$

Estratégias de Neuro-identificação: Modelo Inverso da Planta

- Modelagem Inversa da Planta :
 - Para planta não-linear, os valores do Jacobiano (dy/du) devem variar, e haverá necessidade de ponderar o erro do controle.
 - Se o erro de saída não for correlacionado com o Jacobiano da planta, então:
$$J_y = E(\varepsilon_y^2(t)) \approx E\left[\left(\frac{dy(t)}{du(t)}\right)^2\right] E[\varepsilon_u^2(t)] \approx E\left[\left(\frac{dy(t)}{du(t)}\right)^2\right] J_u$$
 - E o Jacobiano poderá ser incorporado à taxa de treinamento.

Estratégias de Neuro-identificação: Modelo Inverso Especializado da Planta

- Arquitetura do Modelo Inverso Especializado da Planta ou Processo
 - Após aprender o modelo da planta:



Estratégias de Neuro-identificação: Modelo Inverso Especializado da Planta

- Modelagem Inversa Especializada da Planta:
 - De forma semelhante ao MIP, propõe um modelo inverso de planta com uma unidade de função de transferência.
 - O método é diferente daquele anteriormente descrito:
 - Constrói-se um Modelo Direto de Planta (MDP);
 - A diferença entre a resposta da MDP e da saída desejada forma um sinal de erro, que é retro-propagado para ajustar os parâmetros livres do Modelo Inverso.
 - A principal vantagem desta abordagem é o chamado *goal driven*. Isto é, tende a não aprender apenas ruídos.

Estratégias de Neuro-identificação: Modelo Inverso Especializado da Planta

- Modelagem Inversa Especializada da Planta:
 - Para aplicações on-line, o erro de saída da planta força o MIP a se deslocar para regiões inexploradas no espaço das entradas, enquanto o MIEP só aprende quando o sinal de controle for suficientemente excitado.
 - Outra grande diferença o MIEP e o MIP é que o primeiro tenta minimizar o *erro quadrático da planta*, enquanto o segundo minimiza o *erro quadrático do controle*;
 - Novamente, se a planta for não-linear, então como no MIEP, haverá uma incompatibilidade entre a verdadeira planta inversa e o modelo adaptativo.

Estratégias de Neuro-identificação: Modelo Inverso Especializado da Planta

- Modelagem de Planta Inversa Especializada:
 - Arquiteturas de Modelagem Inversa podem ser utilizadas para *sintetizar* controladores, porém podem não ser tão robustos quanto controladores de aprendizagem alternativos, devido a falta de informação de realimentação.

Estratégias de Neuro-identificação: Modelo do Operador

- Modelagem de Operadores:
 - O controlador é sintetizado pela aprendizagem de um especialista humano (operador do sistema);
 - O algoritmo de aprendizagem é rodado em paralelo a um operador habilidoso da planta:
 - As respostas do operador formam a saída desejada da RNA, que pode ser treinada com esses dados.
 - Em geral, o sinal que entra na rede precisa ser previamente filtrado pois existe muito ruído decorrente do fato que vários sinais de atuação diferentes resultam em sinais de entradas da rede similares.

Identificação: Representação de Plantas

- Sistema de equações de estado que representa um sistema de controle SISO contínuo no tempo:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t))$$

$$y(t) = g(\mathbf{x}(t))$$

- Para um sistema SISO discreto no tempo:

$$\mathbf{x}[k + 1] = \mathbf{f}(\mathbf{x}[k], u[k])$$

$$y[k] = h(\mathbf{x}[k])$$

onde $\mathbf{x}[k]$ é o vetor do estado da planta, $u[k]$ é o sinal de controle e $y[k]$ é a saída observada da planta.

Identificação: Representação de Plantas

- A maioria dos modelos de plantas, admite que o sinal de saída possui informações sobre estados anteriores da planta, ou seja:

$$y[k] = h(\mathbf{y}[k-1], \mathbf{u}[k])$$

onde $\mathbf{y}[k-1]$ é um vetor de comprimento n_y , formado pelas saídas passadas $y[k-1], \dots, y[k-n_y]$ e $\mathbf{u}[k]$ é um vetor de comprimento n_u+1 , formado pelas ações de controle presente e passadas $u[k], \dots, u[k-n_u]$.

Identificação: Representação de Plantas

- Resta escolher os valores de n_y e n_u apropriados:
 - Se forem muito grandes, haverá perda de generalização e convergência pobre devido a sobre estimativa;
 - Se forem muito pequenos, o modelo será incapaz de captar a dinâmica existente, afetando a estabilidade de aprendizagem do sistema de controle.

Identificação por Redes Neurais Supervisionadas

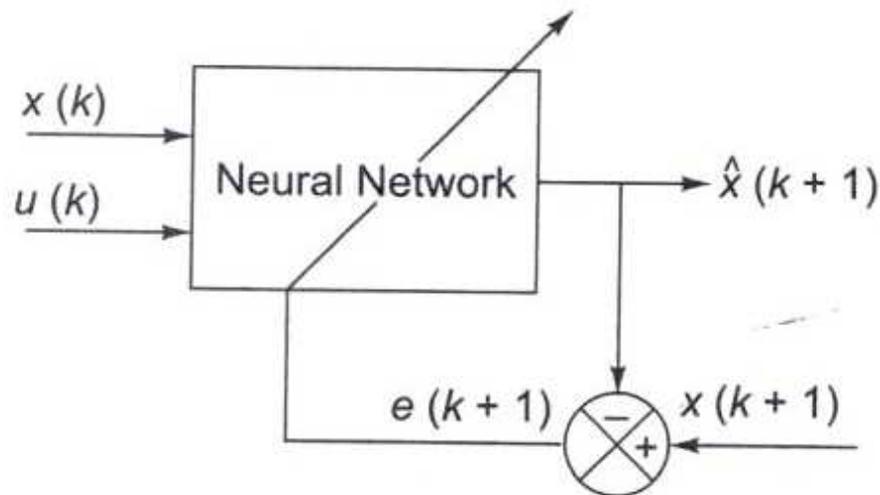
- Identificação (f desconhecido): Seja um sistema discreto, causal e invariante no tempo que é descrito pela equação $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$, com $\mathbf{u}(\cdot)$ sendo um sinal uniformemente limitado e a planta estável. Uma rede neural de alimentação direta $N(\cdot)$ aproxima f se a rede predisser uma saída estimada, dadas a sequência de entrada $\mathbf{u}(k)$ e o estado do sistema $\mathbf{x}(k)$, tal que:

$$\sum_{k=1}^P \|\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1)\| \leq \varepsilon, \quad \text{onde } \varepsilon < 0,$$

P é o número de instantes de tempo.

Identificação por Redes Neurais Supervisionadas

- Sistemas modelados por equações de diferenças e empregando redes neurais de alimentação direta:
 - A forma não afim: $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$
onde $\mathbf{u}(\cdot) \in R^m$ e $\mathbf{x}(\cdot) \in R^n$, são os vetores de entrada e de estado.



Non-affine system model using feed-forward networks

Identificação por Redes Neurais Supervisionadas

- Sistemas modelados por equações de diferenças e empregando redes neurais de alimentação direta:
 - A forma afim: $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{g}(\mathbf{x}(k)) \mathbf{u}(k)$
onde $\mathbf{u}(\cdot) \in R^m$ e $\mathbf{x}(\cdot) \in R^n$, são os vetores de entrada e de estado.

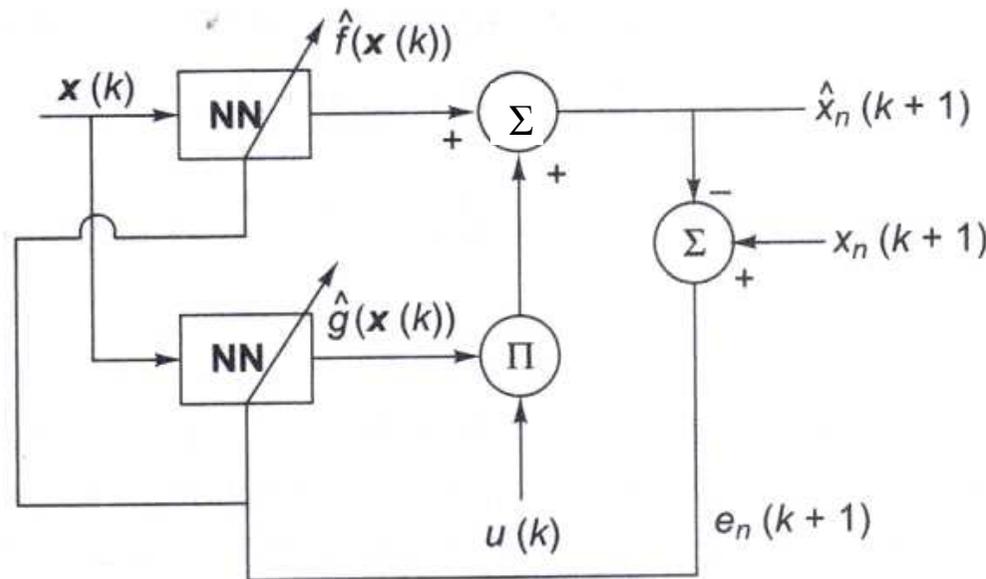


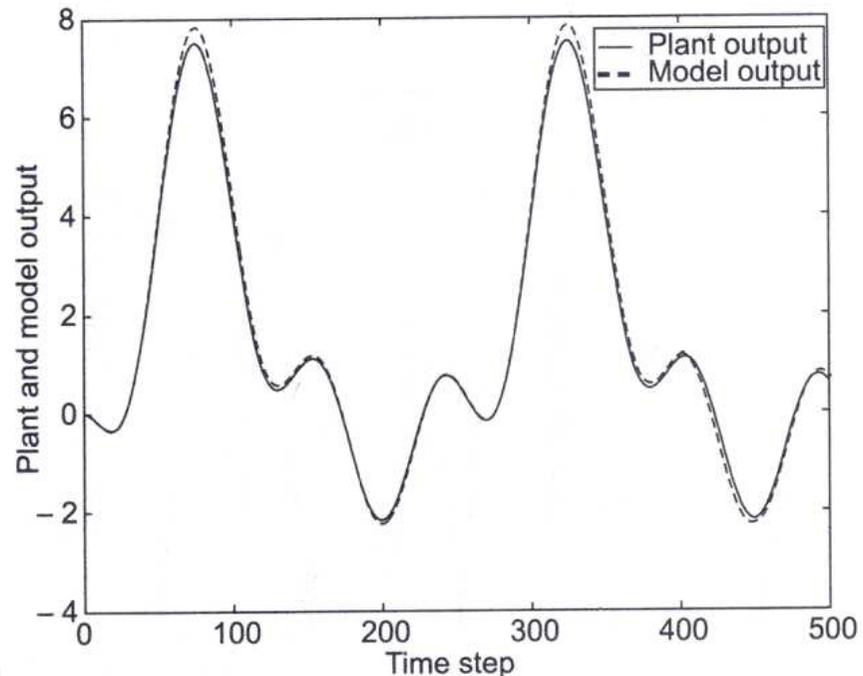
Fig. 2.35 Affine system model using feed-forward networks

Identificação por Redes Neurais Supervisionadas

Exemplo: Seja planta: $x(k+1) = 0,3x(k) + 0,6x(k-1) + g(u(k))$, onde $g(u) = 0,6 \sin(u) + 0,3 \sin(3\pi u) + 0,1 \sin(5\pi u)$; o sistema não-forçado linear é assintoticamente estável.

Modelo neural: $\hat{x}(k+1) = 0,3x(k) + 0,6x(k-1) + N(u(k))$

Rede neural $N(u)$ de 2 camadas escondidas (20, 10 unidades), treinada por retropropagação com taxa de aprendizagem igual a 0,25. Entradas sorteadas de distribuição uniforme no intervalo $[-1, 1]$. A identificação que usa 500 passos de tempo (figura ao lado) tem RMS 0,003.



Identificação por Redes Neurais Supervisionadas

Exemplo: Se uma planta MIMO

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k)/[1+x_2^2(k)] \\ x_1(k)x_2(k)/[1+x_2^2(k)] \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

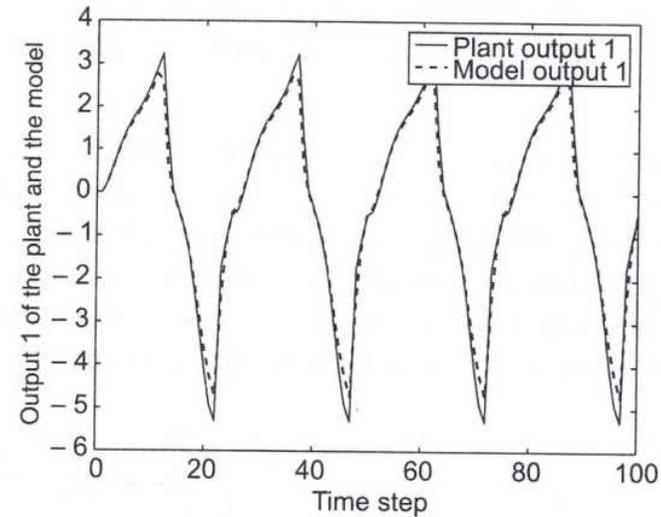
Com modelo neural considerado:

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} N^1(x_1(k), x_2(k)) \\ N^2(x_1(k), x_2(k)) \end{bmatrix} + \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

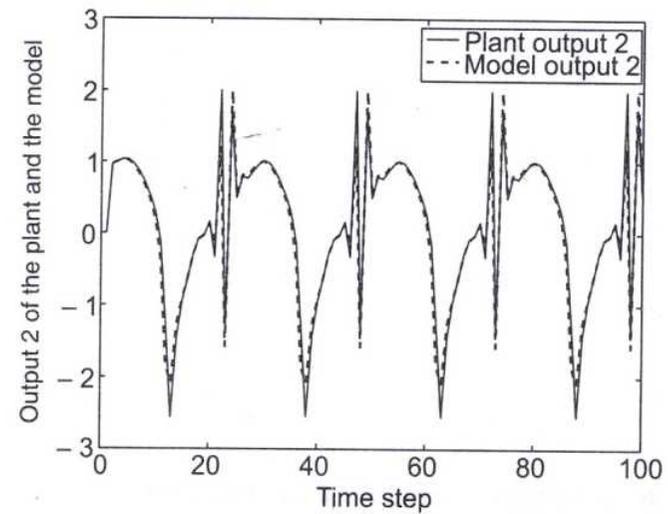
A identificação acontece com duas entradas aleatórias, u_1 e u_2 , extraídas de uma distribuição uniforme entre -1 e 1. N^1 e N^2 são duas redes neurais com duas camadas escondidas cada (15 e 10 unidades). A taxa de aprendizagem foi de 0,1. As redes foram treinadas por retropropagação.

Identificação por Redes Neurais Superivisionadas

As duas saídas:



(a)



(b)

Bibliografia Básica

- Behera, L. & Kar, I. (2010). *Intelligent Systems and Control: Principles and Applications*. Oxford University Press. ISBN13: 978-0-19-806315-5.
- Brown, M. & Harris, C. (1994). *Neurofuzzy Adaptive Modelling and Control*. Prentice-Hall (International Series in Systems and Control Engineering). ISBN-10: 0-13-134453-6.
- Hunt, K. J., Sbarbaro, D., Zbikowski, R., & Gawthrop, P. J. (1992). Neural networks for control systems - A survey. *Automatica*, 28 (6): 1083-1112.
- IEEE Task force on Intelligent Control: <http://www3.nd.edu/~pantsakl/Publications/162-RTFIC.pdf>
- Passino, K. M. (2010) Intelligent Control. In *The Control Systems Handbook*, Second Edition: Control System Advanced Methods. Edited by William S. Levine, 54-1 - 54-12, CRC Press