

Health Management, Fault Diagnosis and Supervisory Control for Intelligent Engineering Systems

Part 4

Neural Networks for Fault Diagnosis

Silvio Simani

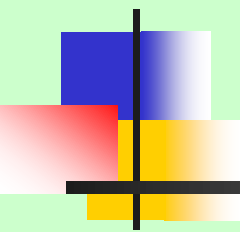
E-mail: silvio.simani@unife.it

URL: www.silviosimani.it/talks.html

References

Textbook (*suggested*)

- *Neural Networks for Identification, Prediction, and Control*, by Duc Truong Pham and Xing Liu. Springer Verlag; (December 1995). ISBN: 3540199594
- *Nonlinear Identification and Control: A Neural Network Approach*, by G. P. Liu. Springer Verlag; (October 2001). ISBN: 1852333421.
- *Multi-Objective Optimization using Evolutionary Algorithms*, by Deb Kalyanmoy. John Wiley & Sons, Ltd, Chichester, England, 2001.

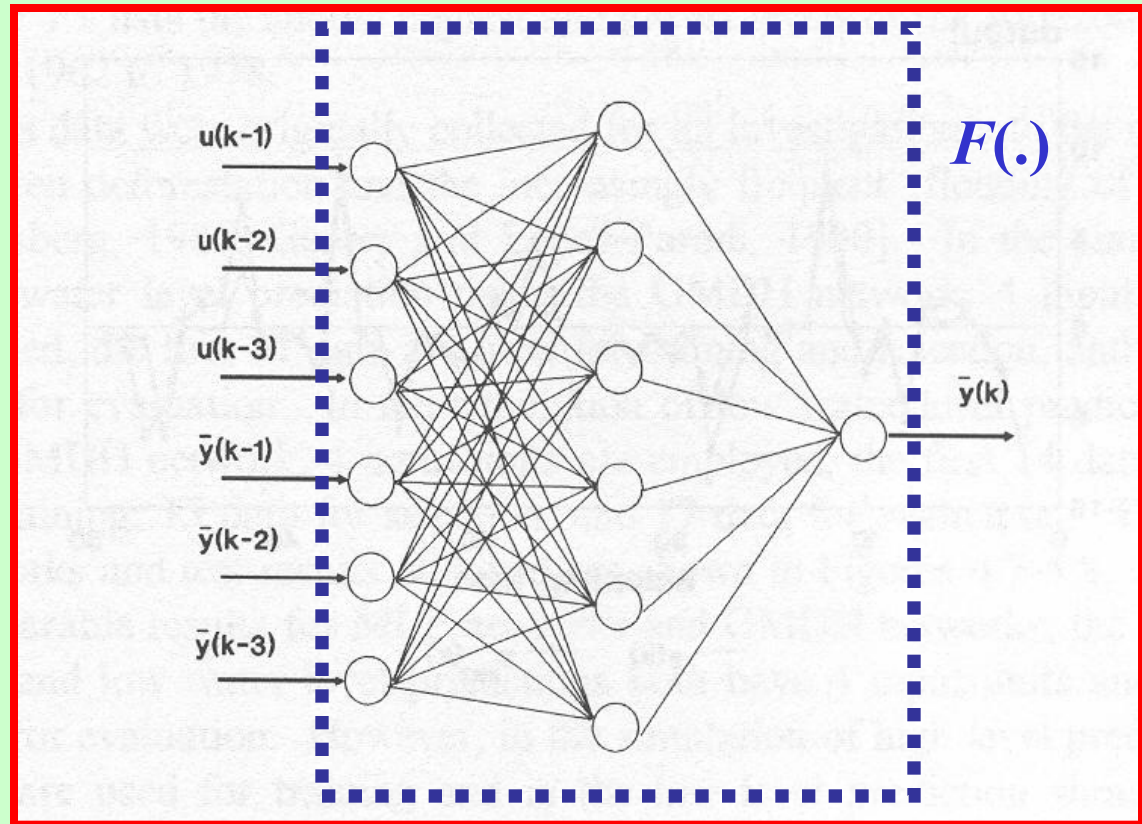


Application Examples

Neural Networks for
Fault Diagnosis of Nonlinear Processes

Nonlinear *Dynamic* System

- Take a static NN
- From static to dynamic NN
- "Quasi-static" NN
- Add inputs, outputs and delayed signals



$$\tilde{y}(k) = F(u(k-1), u(k-2), u(k-3), \tilde{y}(k-1), \tilde{y}(k-2), \tilde{y}(k-3))$$

Example of Quasi-static NN
with 3 delayed inputs and outputs

Nonlinear System Identification

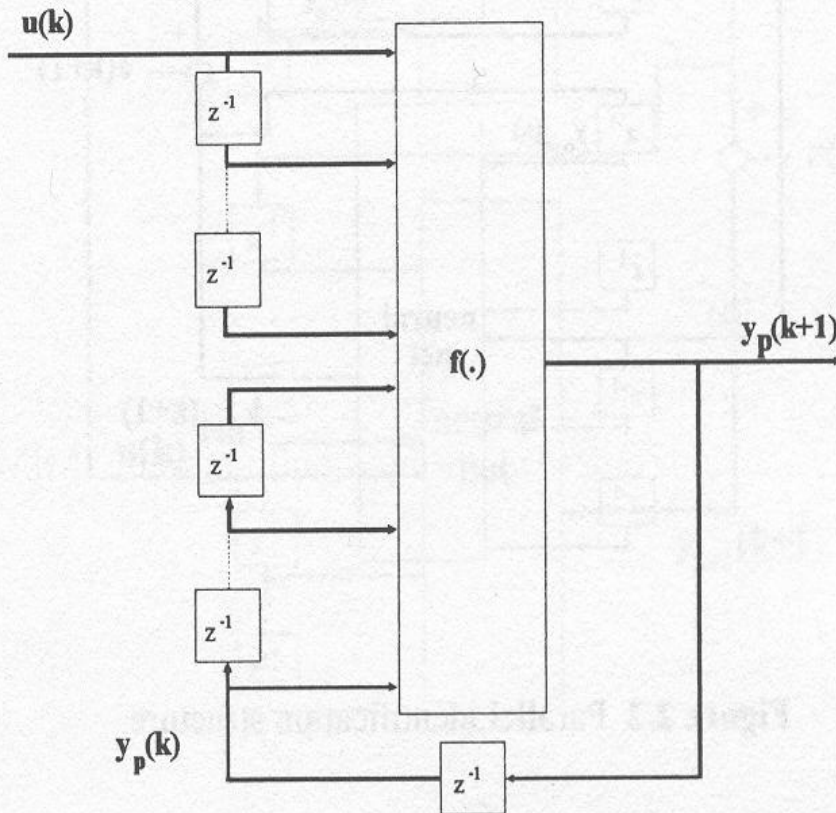


Figure 2.1 Input-output model

- $f(\cdot)$, unknown target function
- Nonlinear dynamic model
- Approximated via a quasi-static NN
- Nonlinear dynamic system identification
- Recall "*linear system identification*"

Nonlinear System Identification

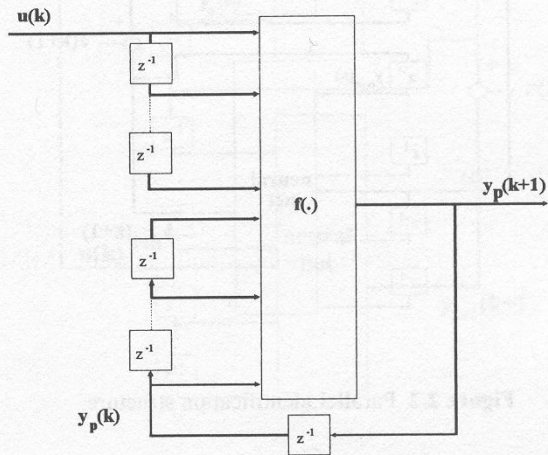
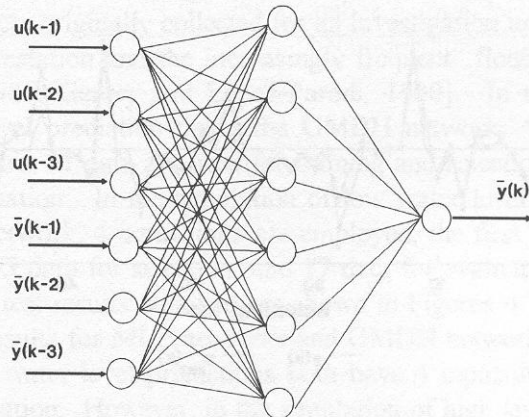
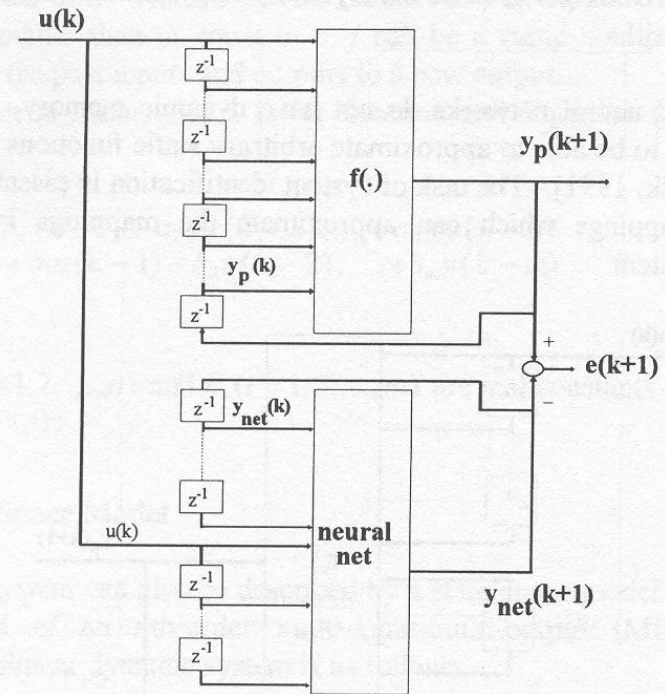
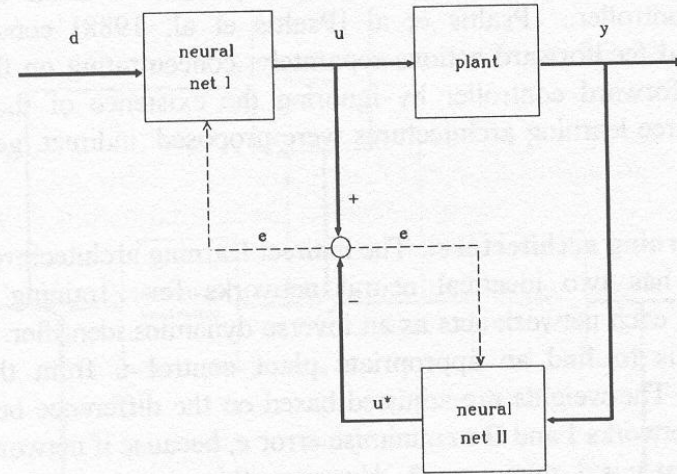
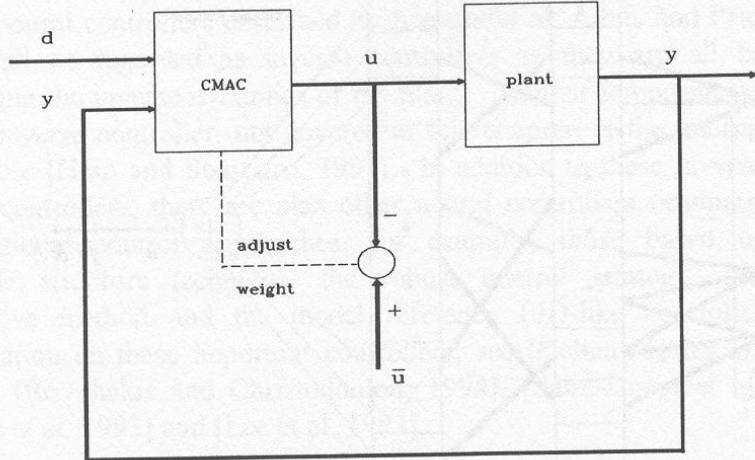


Figure 2.1 Input-output model



Target function: $y_p(k+1) = f(.)$
 Identified function: $y_{NET}(k+1) = F(.)$
 Estimation error: $e(k+1)$

Nonlinear System Neural Control



d : reference/desired response
 y : system output/desired output
 u : system input/controller output
 $\bar{u}$: desired controller input
 u^* : NN output
 e : controller/network error

The goal of training is to find an appropriate plant control u from the desired response d . The weights are adjusted based on the difference between the outputs of the networks I & II to minimise e . If network I is trained so that $y = d$, then $u = u^*$. Networks act as inverse dynamics identifiers.