15th Triennial World Congress, Barcelona, Spain

NEURAL MODELING FOR DIESEL ENGINE CONTROL

Xavier DOVIFAAZ, * **Mustapha OULADSINE,** ** **Ahmed RACHID,** *** **Gérard BLOCH** ****

* *Centre de Recherche en Automatique de Nancy (CRAN), ESSTIN, 2 rue Jean Lamour, 54000 Vandoeuvre, France.*
xavier.dovifaaz@esstin.uhp-nancy.fr
** *LSIS, IUSPIM, Domaine Universitaire St Jérôme, Av. de l'Escadrille Normandie-Niemen, 13013 Marseille, France.*
*** *Centre de Robotique, d'Electrotechnique et d'Automatique, 7 rue du Moulin Neuf, 80000 Amiens, France.*
**** *Centre de Recherche en Automatique de Nancy (CRAN), ESSTIN, 2 rue Jean Lamour, 54000 Vandoeuvre, France.*

Abstract : This paper presents a neural model of a Diesel engine and its use for control. It describes the method used to design the neural model, from the physical equations that concern the engine. This model is then used in a specialized training scheme to control the engine. A trajectory tracking of the engine speed with and without pollution constraints is simulated.

Keywords: neural-network models, neural control, Diesel engines, automotive control, automotive emissions.

1. INTRODUCTION

Neural techniques are used in many automatic domains like modeling (Pham, 1995) (Chen *et al.*, 1990) (Narendra and Parthasarathy, 1990), control (Adetona *et al.*, 2000) (Chen *et al.*, 1999) (Hafner *et al.*, 1999), vision and diagnosis. Indeed, neural networks bring important benefits by suppressing some theoretical difficulties that appears when applying some classical techniques on complex systems. As they include themselves nonlinearities in their structure, they can describe or control complex nonlinear systems with precision. Thanks to neural techniques, a large class of nonlinear systems can be modeled or controlled with a priori few theoretical knowledge compared to classical techniques.

In this paper, we applied neural techniques to model and control a turbocharger Diesel engine. Our objective is to construct a model that can be used to control the Diesel engine. Particularly, we want to control the engine speed and the opacity of the exhaust gas that characterize one type of pollution. More precisely, the control should allow to reduce the peaks of opacity that occur during engine acceleration. Neural networks are used because they can replace the complex and nonlinear thermodynamic, mechanical and chemical equations that describe the Diesel engine (Blanke and Andersen, 1985). This paper is presented as follows. section 1 presents the neural network for modeling the behaviour of the engine. Section 2 presents some the neural control simulation results using the model presented in section 1.

2. NEURAL NETWORK MODELING

2.1 *Structure of the engine model*

It can be decomposed into subsystems as presented in figure (1). The atmospheric air goes through the compressor, the air intake manifold, and the combustion chamber. The injection pump inject fuel in the combustion chamber while the valves are closed, and the mixture burns. The gases produced by the explosion passe through the exhaust manifold and the turbine and are ejected out away. Five states have been modeled : the engine speed R (rpm), the intake manifold pressure P (kPa), the inlet air flow $\dot{m}$ (kg/s), the fuel flow $\dot{m}_f$ (kg/s) and the opacity of the exhaust gas $\mathcal{O}_p$ (%). This work mainly concerns the engine speed and the opacity. The only command that we consider is the position signal of the injection pump rack, T (V).

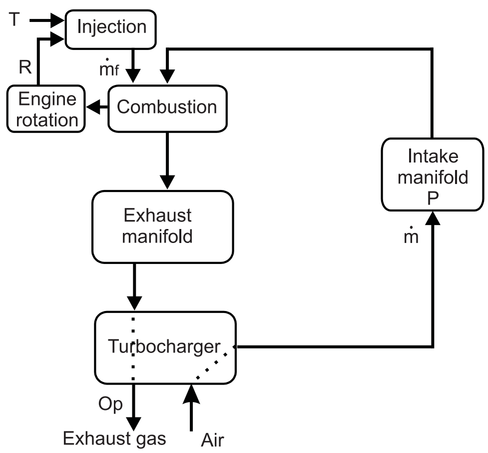

FIG. 1. Turbocharged diesel engine plant

The physical relations that describe the internal combustion engine (see for instance (Blanke and Andersen, 1985) (Kao and Moskwa, 1995)) are used to design the neural model of the engine. Among the different possibilities, the Diesel engine speed, the intake manifold pressure and the exhaust gas opacity can be described with the following relations :

$$\begin{cases} \dfrac{dR}{dt} = f_R(R, P, T) \\ \dfrac{dP}{dt} = f_P(R, P) \\ \mathcal{O}_p = f_{Op}(R, \dot{m}, \dot{m}_f) \end{cases} \qquad (1)$$

It can be noticed that on one hand the dynamic of speed is mainly due to the engine inertia and that the speed value naturally depends on the injected fuel rate and thus on the injection pump position T. On the another hand, the pressure derivative depends on the intake manifold air flows. In the present case, as no external load is applied to the engine, the turbocharger as a weak influence on the engine. This explains that the air flows, and thus the pressure derivative, mainly depends on the engine speed and on the pressure. As regards the opacity, it depends on the ratio between the air and the fuel quantities used in the combustion. This partially explains why the opacity depends on the air flow and the fuel injection pump position.

These relations are used to construct the neural model used to estimate the speed, the pressure and the opacity. The first step consists of the first order discretization of the previous equations :

$$\begin{aligned} R(k) &= NN_R \;\; (R(k-1), ..., R(k-n_{RR}), \\ &\qquad P(k-1), ..., P(k-n_{PR}), \\ &\qquad T(k-1), ..., T(k-n_{TR})) \\ P(k) &= NN_P \;\; (R(k-1), ..., R(k-n_{RP}), \\ &\qquad P(k)..., P(k-n_{PP})) \\ \mathcal{O}_p(k) &= NN_{op} \;\; (\dot{m}_f(k), R(k), \dot{m}(k)) \end{aligned} \qquad (2)$$

where n_{RR}, n_{TR}, n_{RP}, n_{PR}, n_{PP} and n_{PP} are the model orders that must be identified and where NN_R, NN_P and NN_{op} represent the functions modeled by the neural networks for respectively the estimation of the engine speed, the pressure and the opacity.

Concerning the opacity model, some modifications are needed. Firstly, the opacity is measured at the exhaust of the Diesel engine. This means that there is some delay between opacity and the other variables and that there is some dynamics due to the transportation of the gas. Secondly, the opacity depends on the injected fuel flow. Some works (Blanke and Andersen, 1985) shows that this quantity mainly depends on the engine speed and on the injection pump position :

$$\dot{m}_f(k) = f(R(k), T(k)) \qquad (3)$$

With these considerations the opacity can be expressed as below :

$$\begin{aligned} \mathcal{O}_p(k) &= NN_{op} \;\; (\mathcal{O}_p(k-1), ..., \mathcal{O}_p(k-n_{op}), \\ &\qquad T(k-d), R(k-d), \dot{m}(k-d)) \end{aligned} \qquad (4)$$

where d is the delay mentioned above and n_{op} the order of the opacity model.

One objective of the modeling is to control the engine. However, the model is not easy to use in control due to the interlocking of the models of speed and pressure. This explain why some simplification was proceeded. On one hand, the speed at time k depends on the command T, on the speed at previous times, and on the pressure. In the other hand, this pressure depends on speed at previous times. Thus we considered that this is possible to express the speed as a neural function depending on the command and speed only.

For the model to be complete, the orders and the number of nodes in each hidden layer, have to be identified. For each order value, the neural network is trained for a given node number and a criterion is calculated. The criterion values allows to select an optimal order. The neural network is then trained with this order, but for several values of the node number in the hidden layer. A criterion analysis give the final node number and thus the final network. This process is repeated for each network, what leads to the final models of speed and opacity :

$$
\begin{array}{rcll}
R(k) & = & NN_w & (R(k-1), R(k-2), T(k-1)) \\
\mathcal{O}_p(k) & = & NN_{Op} & (\mathcal{O}_p(k-1), T(k-4), \\
 & & & R(k-4), \dot{m}(k-4)) \\
P(k) & = & NN_P & (P(k-1), R(k-1))
\end{array}
\quad (5)
$$

The complete model (shown by figure 2) consists then of several interconnected multilayer perceptrons composed of several inputs, an output and a single hidden layer. One of them reconstructs the engine speed from the command T. The neural model of speed, pressure, air flow and opacity respectively contains 4, 3, 2 and 5 hidden neurons in their hidden layer. These numbers give an idea of the model complexity. It is not surprising to find that the air flow model contains only 2 nodes, since the air flow is almost a linear function of speed (when no external load is applied to the engine). On the other hand, the opacity model includes 5 nodes, since the opacity is generally described by complex functions. These recurrent neural networks have to be trained using data of the command T, the speed R, the air flow $\dot{m}$ and the opacity $\mathcal{O}_p$.

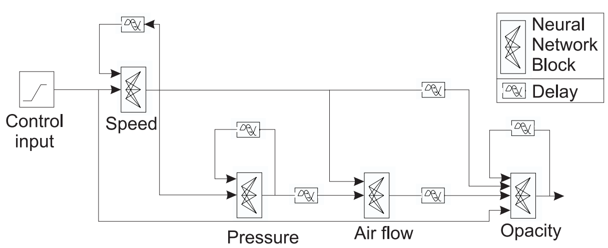

FIG. 2. Neural model structure

2.2 *Experimental results*

The following figures present some estimation results. The method of training used is the Levenberg-Marquardt algorithm (Söberg *et al.*, 1995) (Bloch *et al.*, 1996), which is not detailed here. Figures (4), (5) and (6) present the measurements and estimations respectively for the speed, the pressure and the opacity used to identify the weights of the neural networks. These measurements were generated using the control profile shown on figure (3). In order to validate the model, another temporal set for the input T (figure (7)) was applied to the real system and the neural model. The corresponding measurements and estimations of speed, pressure and opacity are given by the figures (8), (9) and (10). It should be noticed that the corresponding sample time of the data is fixed to 0.1 s. Its value is chosen sufficiently small for the model to be able to reproduce the dynamics of the engine. However, its bottom value is limited by the acquisition system capability. The reducing of the sample time to a too small value is furthermore useless because the model is a mean-value model.

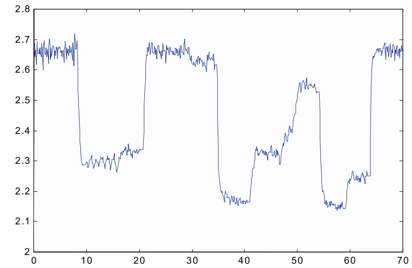

FIG. 3. Control input (V), identification data.

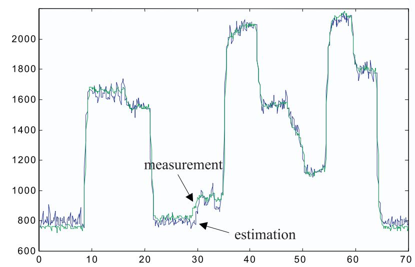

FIG. 4. Measurements and estimates of speed (rpm), identification data.

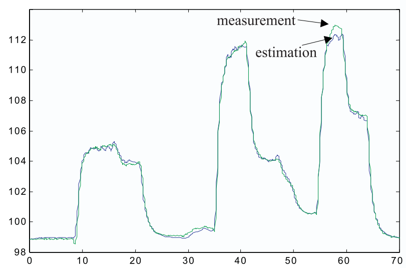

FIG. 5. Measurements and estimates of pressure (kPa), identification data.

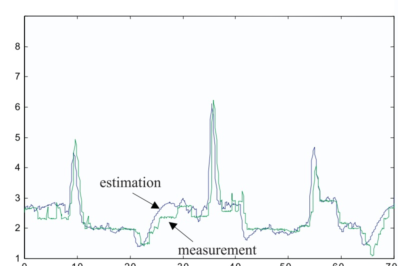

FIG. 6. Measurement and estimates of opacity (%), identification data.

Even if the estimations are given by a complete simulation neural model whose the single input is the position T, the estimates of speed, pressure and opacity are near the measurements. The engine model reproduces the static and the dynamical behaviour of the system with a good precision. Figures (6) and (10) show that peaks and static levels of opacity are well estimated, despite the dynamics and nonlinearities.

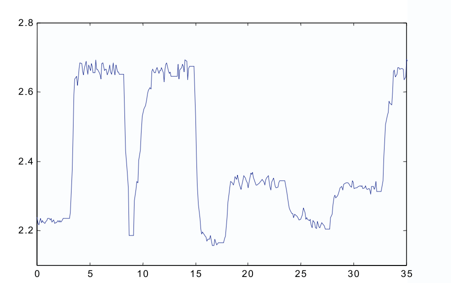

FIG. 7. Control input (V), validation data.

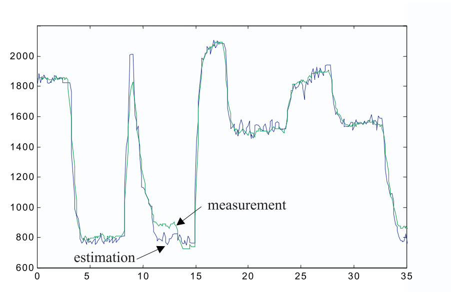

FIG. 8. Measurement and estimates of speed, validation data.

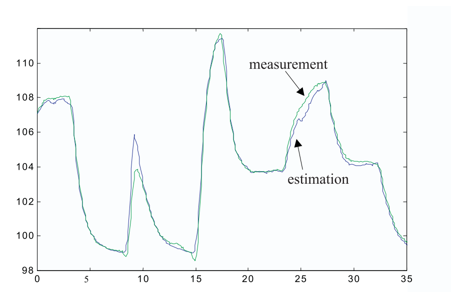

FIG. 9. Measurements and estimates of pressure (kPa), validation data.

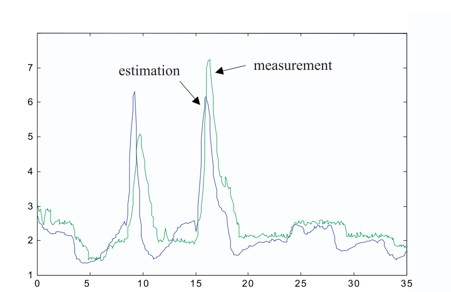

FIG. 10. Measurement and estimates of opacity (%), validation data.

3. NEURAL CONTROL

3.1 *Introduction and theory*

The benefits of neural networks for modeling and control are of the same order. They allow to control complexes non-linear processes, in an optimal manner. Several control schemes using the neural networks are presented in literature, like predictive control (Eaton *et al.*, 1994) (Soloway and Haley, 1996), internal model control (Rivals and Personnaz, 2000) (Hunt and Sbarbaro, 1991), inverse control (He *et al.*, 1999) and optimal control (Plumer, 1996). Several strategies of the controller training have been proposed. In our application, the approach used for constructing the control is the "specialized training", credited to (Psaltis *et al.*, 1988) and whose the principle is given in figure (11) :

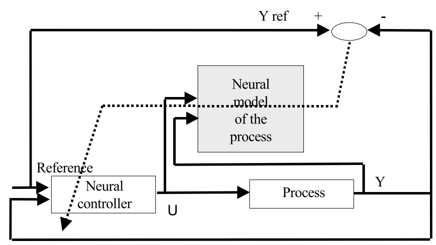

FIG. 11. Architecture of training for control

It uses a direct neural model of the process and the residual between the reference and the system output to update the controller weights. More precisely, the training consist of minimizing a criterion using the Jacobians of the system estimated by the neural model of the system. In the first step, only the speed is controlled. The criterion thus contains only the engine speed variable. In the second step, the opacity is taken into account by the controller. In this case, the criterion to minimize contains the engine speed and the opacity :

$$J(W) = \frac{1}{2}\sum_{k=1}^{N} \Big(\eta_y (Y_{ref} - Y(W,k))^2 + \eta_z (Z_{ref}(k) - Z(W,k))^2\Big) \qquad (6)$$

where the variables are Y and Z and the respective references Y_{ref} and Z_{ref}. η_y and η_z are weighting factors.

A possible structure for the controller can be as follows :

$$\begin{aligned} U(k) = \; & NN_u \Big(Y_{ref}(k+1), Y(k), \ldots, Y(k-n_y+1), \\ & Z_{ref}(k+1), Z(k), \ldots, Z(k-n_y+1), \\ & U(k-1), \ldots, U(k-n_u)\Big) \end{aligned} \qquad (7)$$

The general rule for calculating recursively the controller parameters (the weights) is :

$$W^i = W^{i-1} - \left[R(W^{i-1})\right]^{-1} J'(W^{i-1}) \qquad (8)$$

where W^i denotes the ith updating of the weights parameter vector $W = (w_1 \cdots w_n)^T$ and where the gradient of the criterion, J' is defined by : $J'(W) = \left(\frac{\partial J}{\partial w_1} \quad \frac{\partial J}{\partial w_2} \quad \cdots \quad \frac{\partial J}{\partial w_n}\right)^T$

In the Gauss-Newton algorithm, R is an approximation of the Hessian matrix. It allows a fast and efficient convergence of the criterion to the minimum. The recursive algorithm is then given by the following equalities :

$$\begin{cases} W^{k+1} = W^k - P_k \Big(e_y(W^k, k+1)\, \Psi_y(W^k, k+1) + \\ \qquad \eta_z\, e_z(W^k, k+1)\, \Psi_z(W^k, k+1) \Big) \\ M = \Big(P_k - \dfrac{P_k\, \Psi_y(W^k, k+1)\, \Psi_y^T(W^k, k+1)\, P_k}{1 + \Psi_y^T(W^k, k+1)\, P_k\, \Psi_y(W^k, k+1)} \Big) \\ P_{k+1} = \Big(M - \dfrac{M\, \Psi_z(W^k, k+1)\, \Psi_z^T(W^k, k+1)\, M}{1 + \Psi_z^T(W^k, k+1)\, M\, \Psi_z(W^k, k+1)} \Big) \end{cases} \tag{9}$$

where e_y and e_z are the output prediction errors respectively for the speed and the opacity :

$$e_y(W, k+1) = Y_{ref}(k+1) - Y(W, k+1)$$

$$e_z(W, k+1) = Z_{ref}(k+1) - Z(W, k+1)$$

and where the vectors Ψ_y and Ψ_z are defined by :

$$\Psi_y(W, k+1) = \left(\frac{\partial e_y(W, k+1)}{\partial w_1} \cdots \frac{\partial e_y(W, k+1)}{\partial w_n} \right)^T$$

$$\Psi_z(W, k+1) = \left(\frac{\partial e_z(W, k+1)}{\partial w_1} \cdots \frac{\partial e_z(W, k+1)}{\partial w_n} \right)^T$$

The derivatives $\frac{\partial e_y}{\partial w_i}$ and $\frac{\partial e_z}{\partial w_i}$ are given by :

$$\begin{aligned} \frac{\partial e_y(W, k+1)}{\partial w_i} &= -\frac{\partial Y(W, k+1)}{\partial U(W, k)} \frac{dU(W, k)}{dw_i} \\ \frac{\partial e_z(W, k+1)}{\partial w_i} &= -\frac{\partial Z(W, k+1)}{\partial U(W, k)} \frac{dU(W, k)}{dw_i} \end{aligned} \tag{10}$$

where the total derivatives $\frac{dU(W,k)}{dw_i}$ are estimated by the sum of partial derivatives :

$$\begin{aligned} \frac{dU(W, k)}{dw_i} &= \frac{\partial U(W, k)}{\partial w_i} + \\ &\sum_l \left(\frac{\partial U(W, k)}{\partial Y(W, k-l-1)} \frac{\partial Y(W, k-l-1)}{\partial w_i} \right) + \\ &\sum_m \left(\frac{\partial U(W, k)}{\partial Z(W, k-m-1)} \frac{\partial Z(W, k-m-1)}{\partial w_i} \right) \end{aligned} \tag{11}$$

In the following section, some simulation results of the control are presented. The aim is to validate the neural control of the engine speed and of the opacity.

3.2 *Application to the Diesel engine*

This section describes the simulation of the engine control with the use of the model presented in section 2. In a first case, the controller is trained to control the engine speed, while in a second case, the controller training includes the opacity to reduce pollution. In the controller training scheme, the real system is replaced by the model. The variables to control are the engine speed and the opacity. The criterion is thus defined by :

$$\begin{aligned} J(W) = \frac{1}{2} \sum_{k=1}^{N} \Big(&(R_{ref}(k) - R(W, k))^2 + \\ &\eta_{op}(\mathcal{O}_{p_{ref}}(k+d-1) - \mathcal{O}_p(W, k+d-1))^2 \Big) \end{aligned} \tag{12}$$

where R_{ref} is the speed reference and where $\mathcal{O}_{p_{ref}}$ represents the opacity constraint, defined such that the opacity is reduced during the transients. η_{op} is the opacity constraint weighting factor.

It should be noticed that the controller structure depends on the controlled output and is deduced from the model in (5). For instance, a speed controller output should be given by a function of the following form :

$$T(k+1) = f\ (R_{ref}(k), R(k), R(k-1)) \tag{13}$$

In the more general case where the speed and the opacity are controlled, T is given by the following neural function :

$$\begin{aligned} T(k+1) = NN_T\ (&R_{ref}(k), R(k), R(k-1), \\ &\mathcal{O}_{p_{ref}}(k+d+1), \mathcal{O}_p(k+d)) \end{aligned} \tag{14}$$

The criterion and the controller being defined, the training is processed. Data are previously normalized to avoid the divergence of the parameters and the default parameter values are given by a simple off line training. The in line training phase is then processed in several epochs, each containing the same data set.

Engine speed control

The controller is firstly trained to control the engine speed. This means that the factor η_{op} is 0 and that the controller training uses only the engine speed model. The control becomes a simple engine speed tracking without opacity constraint. For the sake of space, no figure is given. We just mention that the control performances are very good since there is no visible speed tracking error for the speed reference defined in figure (12). This allow to tackle the second step which is the control of speed with an opacity constraint.

Engine speed and opacity control

In order to take account of the opacity, η_{op} is not 0. Figures (12) and (13) present the simulated speed and opacity resulting from the neural control where $\eta_{op} = 0.8$. Figure (12) presents the speed output and its reference, while on figure (13) the opacity curve is compared to the opacity resulting from a control of engine speed only.

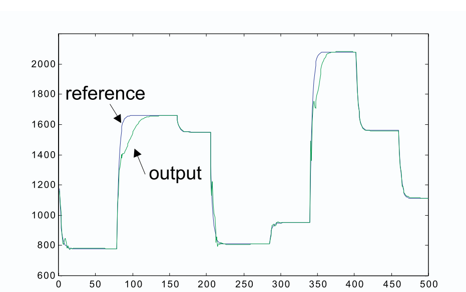

FIG. 12. Engine speed reference and output (rpm).

The results show that a speed tracking error occurs during the transients (acceleration), which is

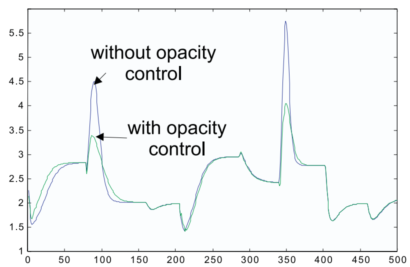

FIG. 13. Opacity reference and output (%).

due to the opacity constraint. In fact, in order to satisfy the opacity constraint, T is calculated by the neural controller such that less fuel is injected. This leads to a decrease of the acceleration and then to a speed tracking error during the transients. This error increases with the weighting factor η_{op}, while in the same time the peaks of opacity decrease.

4. CONCLUSIONS

In this article, we presented a new neural model of the Diesel engine used to design a neural control of speed and opacity. In one hand the neural networks allow to estimate the engine variables with good precision despite strong dynamics and nonlinearities. This is especially the case of the opacity. In the other hand, the results show that the use of a neural model of the engine allows to control the system, taking account of the nonlinearities, what is not obvious with the classical methods. This allows to consider with a great interest the use of neural networks in Diesel engine modeling and control.

5. REFERENCES

Adetona, O., E. Garcia and L.H. Keel (2000). A new method for the control of discrete nonlinear dynamic systems using neural networks. *IEEE Trans. on Neural Networks* **11(1)**, 102–112.

Blanke, M. and J.S. Andersen (1985). On modelling large two stroke diesel engines : News results from identification, bridge between control science and technology. *Proceedings of the Ninth Triennial World Congress of IFAC* **4**, 2015–2020.

Bloch, G., P. Thomas, M. Ouladsine and M. Lairi (1996). On several outlier-robust training rules of neural networks for identification of non-linear systems. *Neural Networks and Their Applications. Conference Proceedings, NEURA96* pp. 13–19.

Chen, S., S.A. Billings and P.M. Grant (1990). Non-linear system identification using neural networks. *International Journal of Control* **51**, 1191–1214.

Chen, Z., M. Lin and Z. Yan (1999). Neural network predictive adaptive controller based upon damped least square. *Proc. of the 14th World Congress of IFAC.*

Eaton, J.W., J.B. Rawlings and L.H. Ungar (1994). Stability of neural net based model predictive control. *Proc. of the American Control Conference* pp. 2481–2485.

Hafner, M., M. Schüler and R. Isermann (1999). Fast neural networks for diesel engine control design. *Proc. of th 14th world congres of IFAC.*

He, S., Q. Wu and N. Sepehri (1999). Control of base-excited inverted pendulums using a neural inverse modeling approach. *Proc. of 14th World Congress of IFAC* **J-3e-02-3**, 408–414.

Hunt, K.J. and D. Sbarbaro (1991). Neural networks for nonlinear internal model control. *IEEE Proc.* **138(5)**, 431–438.

Kao, M. and J.J. Moskwa (1995). Turbocharger diesel engine modeling for nonlinear engine control and state estimation. *ASME Trans., Dynamic Systems Measurement and Control* **117**, 20–30.

Narendra, K.S. and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions On Neural Networks* **1**, 4–27.

Pham, D.T. (1995). *Neural Networks for Identification, Prediction and Control.* Springer-Verlag London Limited.2nd edition. New York.

Plumer, E.S. (1996). Optimal control of terminal processes using neural networks. *IEEE Transactions On Neural Networks* **7**, 408–418.

Psaltis, D., A. Sideris and A.A. Yamamura (1988). A multilayer neural network controller. *IEEE Control Systems Magazine* **8(2)**, 17–21.

Rivals, I. and L. Personnaz (2000). Nonlinear internal model control using neural networks : application to processes with delay and design issues. *IEEE Trans. on Neural Networks* **11(1)**, 80–89.

Söberg, J., Z. Qinghua, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson and A. Juditsky (1995). Nonlinear black-box modeling in system identification : a unified overview. *Automatica* **31**, 1691–1724.

Soloway, D. and P.J. Haley (1996). Neural/generalized predictive control, a newton-raphson implementation. *Proc. of the 11th IEEE Int. Symposium on Intelligent Control* pp. 277–282.