A Multilayer Perceptron Replaces a Feedback Linearization Controller in a Nonlinear Servomechanism

José F. Haffner, Ney T. Meyrer, José N. Amaral and Luís F. A. Pereira

Abstract— **A Feedback Linearizing Controller (FLC) is used to train a multilayer perceptron control a DC motor. After training, the multilayer perceptron replaces the FLC and yields significantly better performance in the presence of state-measurement noise, load disturbances and parameter variations. Simulation results also indicate that the neural network based controller is better able to cope with input saturation resulting from an overly demanding reference model specification.**

Keywords— **Multilayer Perceptron, Neural Network Control, DC motor control, Feedback Linearization.**

I. Introduction

NONLINEAR systems whose dynamics are described by smooth functions can be well controlled using input/output feedback linearizing control techniques. These techniques allow the cancelation of inherent nonlinearities, resulting in a controlled system with a linear input/output behavior that tracks a specified reference model. Applying feedback linearization usually requires an accurate dynamic model and full state measurement. Since these often are not available in real systems, measurement errors and parameter variations can degrade the performance of linearizing controllers.

In this paper we propose to train a neural network to mimic the behavior of a linearizing controller applied to a nonlinear system. Using a mathematical model of the nonlinear system to be controlled, we design a linearizing controller. Applying an excitation to the linearized process inputs, we generate the data necessary to train the neural network. After the network is trained, we simulate situations that could occur in the physical system such as parameter variations, state measurement noise, and load disturbances. We then compare the performance of the original linearizing controller with the performance of the neural network. A pleasant surprise is that when such changes are present in the system the performance of the neural network is *better* than the performance of the original controller. Our results demonstrate the usefulness of some features of neural networks, such as their capability to approximate nonlinear functions and their robustness in presence of noise [3], [7].

We examine the problem of reference tracking and load disturbance rejection in a permanent magnetic field DC motor. Simulation results indicate good control performance using the neural network in the presence of noise and parameter variations. Experimental results are presented in section V. The servomechanism model and the Feedback Linearizing Controller are presented in section II. The neural network learning process is discussed in section III. The Feedback Linearizing Controller with Neural Network in section IV.

Programa de Pós Graduação em Engenharia Elétrica, Pontifícia Universidade Católica do Rio Grande do Sul, http:www.ee.pucrs.br, Porto Alegre - RS - Brazil.

Submited to *1998 IEEE International Joint Conference on Neural Networks*, Anchorage, Alaska, May, 1998.

II. Model and Feedback Linearizing Controller

The servomechanism studied in this research consists of a permanent magnetic field DC motor connected to a load via a gear train. The nonlinear load is a shaft with a mass. The motor axis and the shaft form a 90 degree angle. Our simulation study considers that the speed reduction is ideal, i.e., the gear train has no backlash, all connecting shafts are rigid, and the load can be calculated considering a mass m concentrated in a point at the end of a shaft of negligible mass of length l. The state variables that describe the servosystem are motor position $x_1(t)$, motor speed $x_2(t)$, and armature current $x_3(t)$. The system inputs are the armature voltage $u(t)$ and the torque disturbance $d(t)$. Remaining variables are described in section V.

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \left(\frac{g}{l}\right)\sin x_1 - \left(\frac{B}{J+ml^2}\right)x_2 \\
&\quad + \left(\frac{k_i N}{J+ml^2}\right)x_3 - \left(\frac{N}{J+ml^2}\right)d \\
\dot{x}_3 &= \left(\frac{-k_b N}{L_\mathrm{a}}\right)x_2 - \left(\frac{R_\mathrm{a}}{L_\mathrm{a}}\right)x_3 + \left(\frac{1}{L_a}\right)u \\
y &= x_1
\end{aligned}
$$

A feedback linearizing controller with a reference model and state feedback is used to control the system position. This control is shown in Figure 1 and described in [5]. The load disturbance is not considered in the controller design.

The relative degree of the nonlinear process to be controlled is found by successively differentiating the plant output y with relation to time. The successively differentiation stops when the plant input u appears in the equation of the derivative. For the system under study we obtain

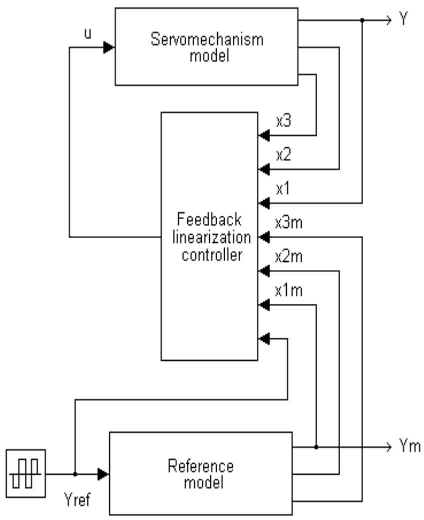

Fig. 1. Feedback Linearization

the following equations:

$$
\begin{aligned}
y &= x_1 \\
\dot{y} &= x_2 \\
\ddot{y} &= \left(\frac{g}{l}\right)\sin x_1 - \left(\frac{B}{J+ml^2}\right)x_2 + \left(\frac{k_i N}{J+ml^2}\right)x_3 && (1) \\
\dddot{y} &= Cu + f(x) && (2)
\end{aligned}
$$

in which

$$
\begin{aligned}
C &= \frac{k_i N}{(J+ml^2)L_a} \\
f(x) &= \left[\frac{-gB}{(J+ml^2)l}\right]\sin x_1 \\
&\quad + \left[\frac{B^2}{(J+ml^2)} - \frac{k_i k_b N^2}{(J+ml^2)L_a}\right]x_2 + \\
&\quad + \left(\frac{g}{l}\right)x_2\cos x_1 - \left[\frac{k_i R_a N}{(J+ml^2)L_a}\right]x_3 && (3)
\end{aligned}
$$

and in which $f(x)$ represents state-dependent system nonlinearities. Model parameters are listed in Table I.

¿From equation 2 we conclude that the servomechanism model has relative degree three. To build a linearizing controller based on a reference model, we must design a third order stable linear reference model. The parameters of the reference model, α_1, α_2, and α_3, are determined to obtain a desired time constant and to limit overshoot. The time constant and maximum overshoot specified must be compatible with the dynamics of the servomechanism. The states of the reference model are position (x_{1m}), speed (x_{2m}), and acceleration (x_{3m}). With this choice of states

Symbol	Variable Name	Value	Unit
R_a	armature resistance	0.83	Ohm
L_a	armature inductance	0.63	mH
k_i	torque constant	0.0182	N m/A
k_b	back EMF constant	0.0182	V s/rad
J	motor inertia	8.32×10^{-6}	kg m^2
B	damping constant	0.0009	N m s/rad
N	gear ratio	5.9	None
E_a	armature voltage	12	V
I_a	armature current		A

TABLE I

Model Parameters

for the model, we obtain equations 4 to 7.

$$
\begin{aligned}
y_m &= x_{1m} && (4) \\
\dot{y}_m &= \dot{x}_{1m} = x_{2m} && (5) \\
\ddot{y}_m &= \dot{x}_{2m} = x_{3m} && (6) \\
\dddot{y}_m &= \dot{x}_{3m} = -\alpha_3 x_{1m} - \alpha_2 x_{2m} - \alpha_1 x_{3m} + \alpha_3 y_{ref} && (7)
\end{aligned}
$$

in which y_{ref} is the reference position.

The linearizing control law is defined in equation 8. The inherent nonlinearities of the system, expressed by $f(x)$ in equation 3, are completely canceled by the linearizing control. The dynamical behavior of the controlled system will be given considering $\dddot{y} = \vartheta$, where ϑ is defined in equation 9.

$$
\begin{aligned}
u &= \frac{1}{C}(\vartheta - f(x)) && (8) \\
\vartheta &= \dddot{y}_m + \alpha_1(\ddot{y}_m - \ddot{y}) + \alpha_2(\dot{y}_m - \dot{y}) + \alpha_3(y_m - y) && (9)
\end{aligned}
$$

Substituting equations 4 to 7 in equation 9, the output of the nonlinear system y is governed by the following differential equation:

$$\dddot{y}_m = -\alpha_1\ddot{y} - \alpha_2\dot{y} - \alpha_3 y + \alpha_3 y_{ref}$$

The equations used to describe the process do not take into consideration the magnetic flux saturation or the limitation in the electronic amplifier used to drive the DC motor; however, the performance of the linearizing control will be degraded when these limitations are violated. To account for these constraints in the simulation studies, we introduce the saturation function in the linearizing controller shown in Figure 2. The parameters of the reference model must be chosen to maintain the control voltage away from the saturation region.

III. Neural Network Training

The scheme used to train the multilayer perceptron is shown in Figure 2. Random probing signals are generated and presented to the servosystem. A set of data, describing the system input/output relationship, is collected and used in the multilayer perceptron learning process.

For the generation of the training data, the control voltage must be kept out of the saturation region. To cover the whole operation range, a noisy signal is added to the control signal to generate a persistently exciting signal [4], [6]. A more detailed description of the multilayer perceptron learning process is found in [1], [3].

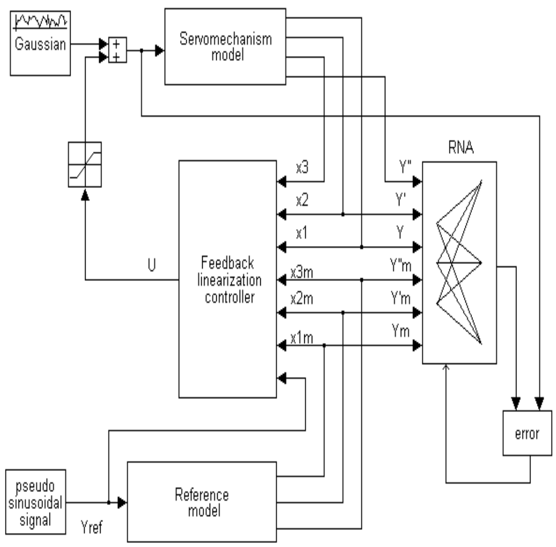

Fig. 2. Neural Network Training

The multilayer perceptron has 6 inputs, one hidden layer with 25 neurons and one output. It is trained with the Levenberg Marquardt algorithm from the MATLAB Neural Network Toolbox. To generate the training data, we apply a pseudo-sinusoidal signal to the input of the reference model as well as to the feedback linearizing controller (see Figure 2). This signal reaches zero at a constant frequency. Every time the signal reaches zero a random coin is flipped to determined whether the signal phase is advanced 180 degrees. Also at this point a new amplitude for the signal is randomly selected. Thirty seconds of this signal were generated. Gaussian noise was added to the control voltage as shown in Figure 2.

IV. Feedback Linearization Controller with Neural Network

The multilayer perceptron is trained with a data set composed of the output of the plant and the output of the model. The inputs of the multilayer perceptron are y, $\dot{y}$, $\ddot{y}$, y_{m}, $\dot{y}_{\mathrm{m}}$, and $\ddot{y}_{\mathrm{m}}$. The single output of the MLP is u. The network is trained until the output error is below an established threshold. After training, the network yields a good approximation of the control law and system dynamics.

This strategy requires that position, speed, and acceleration of the motor be available. The position and speed are easily measured from the system through encoders. Both the acceleration $\ddot{y}$ and the armature current of the DC motor x_3 might be difficult or costly to obtain in practice. Low *et al.* proposed that the acceleration be computed using equation 10 [3]. Observe that this computation involves only position, speed and acceraltion of the motor shaft and the position, speed and acceleration of the reference model.

$$\ddot{y}_{\mathrm{e}} = \ddot{y}_{\mathrm{m}} + K_1(\dot{y}_{\mathrm{m}} - \dot{y}) + K_2(y_{\mathrm{m}} - y) \qquad (10)$$

Figure 3 shows a block diagram in which the neural network has replaced the feedback linearizing controller.

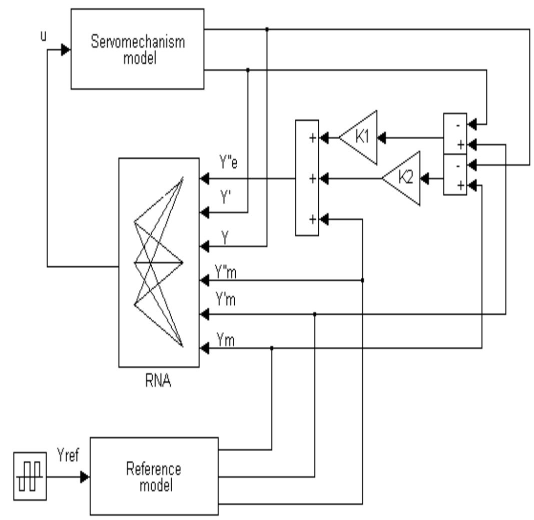

Fig. 3. Feedback Linearization with Neural Network

V. Results

The simulation results were obtained with Matlab's SIMULINK using a sample period of 1 ms. The characteristics of the motor were taken from the Pittman catalog (reference number GM9234).

The load applied to the motor axis is a shaft of length $l = 0.2$ m, mass $m = 0.2$ kg, and we used a gravity acceleration $g = 9.8\ \mathrm{m/s^2}$. The parameters for our third order reference model are $\alpha_1 = 124.70$, $\alpha_2 = 2304$, and $\alpha_3 = 14965$. The gains $K_1 = 1000$ and $K_2 = 8000$ were used in equation 10. To take into account the effect of magnetic saturation, the armature voltage was limited to $\pm$ 20 V. The reference input y_{ref} is a square wave with frequency of 1 rad/s and amplitude of $\pm$ 1.5.

In the discussion and graphs presented in this section the term "linearizing controller" is used to refer to the original feedback linearizing controller strategy, while the term "neural network" represents the situation in which the controller has been replaced by the trained multilayer perceptron as described in section IV.

Figure 4 provides a comparison between the linearizing controller and the neural network when noise is added to each state. The additive noise has zero mean. The variances of the noise begin added to the measurement of position, speed and armature current are $\sigma_p^2 = 0.15\ \mathrm{rad}^2$, $\sigma_s^2 = 30\ (\mathrm{rad/s})^2$ and $\sigma_i^2 = 0.5\ \mathrm{A}^2$, respectively. The responses of both control strategies are very similar.

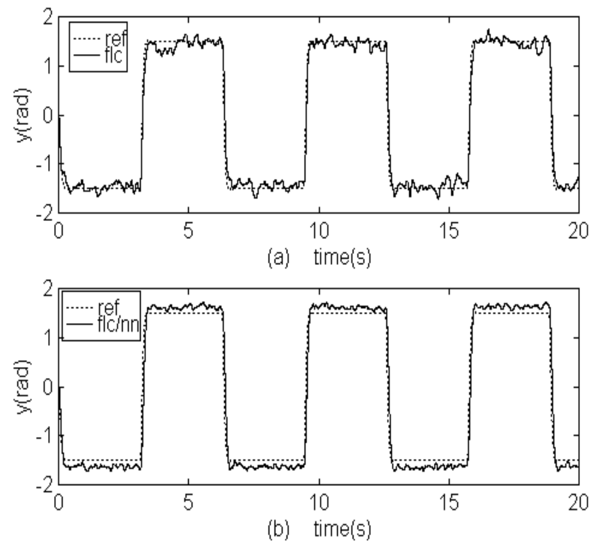

Fig. 4. Response with noise in the measured states: (a) Feedback Linearizing Controller. (b) Neural Network.

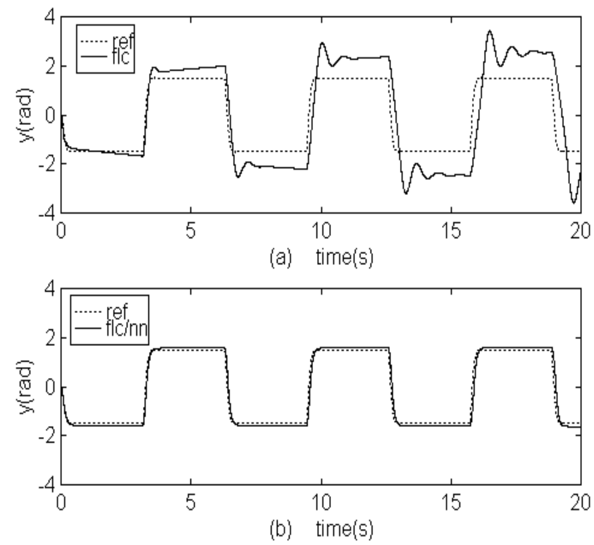

Fig. 5. Response with continuous variation of R_{a} (100%) and L_{a} (10%): (a) Feedback Linearizing Controller. (b) Neural Network.

In Figure 5, system responses using the linearizing controller and the neural network are compared in a simulation incorporating variations in the electrical parameters of the motor (R_{a} and L_{a}). During this simulation the armature resistance R_a and the armature inductance L_a are changed linearly for a period of $20s$. At the end of $20s$, the R_a has doubled and L_a has changed by 10% of its nominal value. These ranges of variation reflect the actual variation of these parameters in practice. Figure 5 shows that the linearizing controller is unable to track the reference signal in the presence of parametric variations without offset. This result is consistent with discussion found in textbooks [2], [5]. The neural network was able to accurately follow the reference signal in spite of the parametric variations, with much less offset.

Figure 6 compares both controllers with a load disturbance of 0.05 Nm[1]. Because the disturbance was not taken into account in the design of the linearizing controller, it is not able to track the reference signal when load disturbance is present. In spite of the load disturbance not being included in the generation of the data for the neural network training, Figure 6 shows that the trained neural network was able to accurately follow the reference even when such a disturbance is present. This result showcases the advantage of the generalization capability of the neural network in this application.

If a linearizing controller is designed with a model that is too fast for the controlled servomechanism, the controller will try to follow the model by increasing the control voltage. Because of the limitation in the DC motor driver along with the magnetic flux saturation, the performance of the controlled system might be degraded. This effect is shown in the upper chart of Figure 7. For the simulation shown in

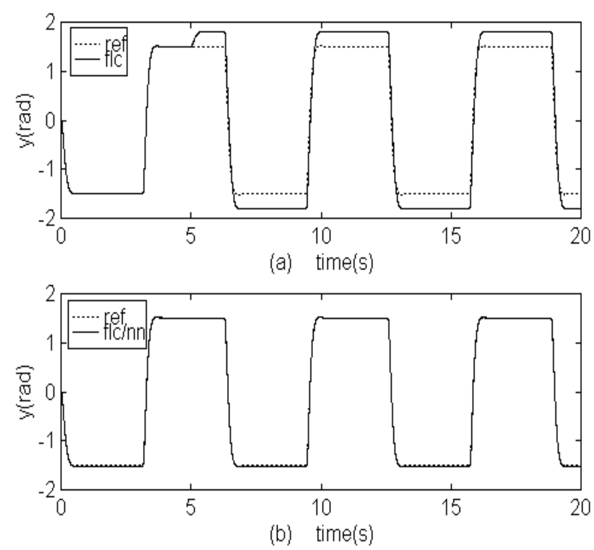

Fig. 6. Response with load disturbance: (a) Feedback Linearizing Controller. (b) Neural Network. Observe the introduction of the load disturbance at time $t = 5s$.

this figure, the reference model was made ten times faster than the model used in the simulations of Figures 4, 5, and 6. The neural network was trained to yield a control voltage within an acceptable interval. Consequently, when the same (faster) model is used in the neural network controller, the armature voltage will remain below the saturation limits. Thus the controller will delay, but will not degrade significantly, the response to the reference signal.

Finally we consider a situation in which the neural network controller is subjected to a load disturbance $(0,05$ Nm$)$ and variation in the motor parameters [2]. The

[1] The load disturbance of 0.05 Nm was chosen so that the resulting load does not exceed the maximum torque specified for the motor.

[2] In this experiment the variation in the parameters is the same used

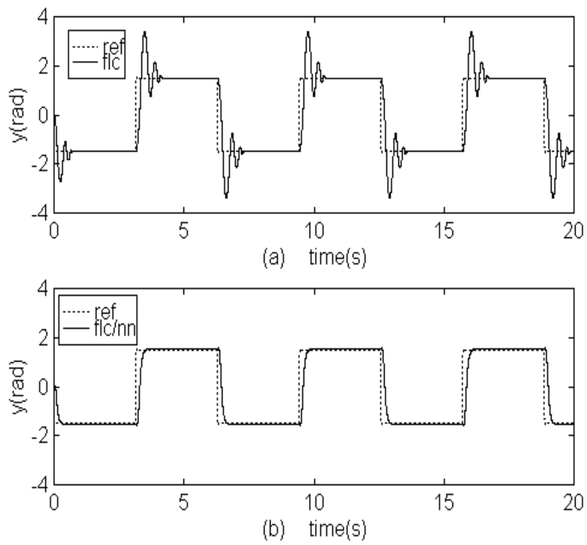

Fig. 7. Response with a 10 times faster Reference Model (a) Feedback Linearizing Controller. (b) Neural Network.

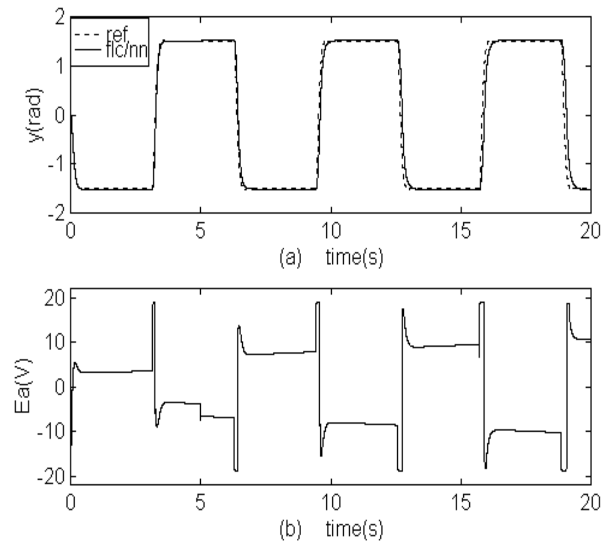

Fig. 8. Neural Network response with variation in plant parameters and load disturbance: (a) Position reference signal and Neural Network output. (b) Neural Network control signal. Observe in the control signal the introduction of the load disturbance at the time $t = 5s$.

neural network response is presented on the top graph of Figure 8. The bottom graph of this figure presents the actual control output of the neural network controller. Notice that this control signal increases with the variation in the parameters, especially in the armature resistance R_a.

VI. Conclusions

The replacement of the feedback linearizing controller with a multilayer perceptron trained with data extracted from the controller operation in a nonlinear servomechanism proved to be advantageous. The multilayer perceptron was better able to cope with variations in the plant parameters, load disturbance, and state measurement noise. It is interesting to notice that these measurement noise, parameter variations, and load disturbances were not present when the data to train the network was generated.

The modeling of the nonlinear system assumed a number of simplifications: the gear train has no backlash, the shafts are rigid, and the load is concentrated at a point. In practice, such simplifications might result in parameter variations of the process and unmodeled dynamics affecting the performance of a linearizing controller. Our results suggest that such effects would be mitigated with our neural network approach.

Acknowledgments

The authors would like to acknowledge the financial support from Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Apoio à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) and Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS).

in the simulation showed in Figure 5.

References

[1] J. F. Haffner, N. T. Meyrer, J. N. Amaral, and L. F. A. Pereira. Utilização de redes neurais artificiais para controle de posição de servomecanismo com carga não-linear. In *III Congresso Brasileiro de Redes Neurais*, pages 455–460, Florianópolis, 1997. L. Caloba and J. Barreto.

[2] Alberto Isidori. *Nonlinear Control Systems*. Springer-Verlag, 1985.

[3] T. S. Low, T. H. Lee, and H. K. Lim. A methodology for neural network training for control of drives with nonlinearities. *IEEE Trans. Ind. Electronics*, 39(2):243–249, April 1993.

[4] K. S. Narendra and A. M Annaswamy. *Stable Adaptive Systems* Prentice-Hall, Inc., 1989.

[5] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice-Hall, Inc., 1991.

[6] R. S. Sutton and P. J. Werbos. *Neural networks for control*. MIT Press, London, 1995.

[7] B. Widrow and E. Walach. *Adaptive Inverse Control*. Prentice Hall, Upper Saddle River, NJ, 1996.