Smoothing the Control Action for NARMA-L2 Controllers

Arjpolson Pukrittayakamee arjpolson@chula.com

Orlando De Jesús dorland@okstate.edu

Martin T. Hagan mhagan@okstate.edu

Oklahoma State University, School of Electrical and Computer Engineering,
202 Engineering South, Stillwater, OK, 74078-5032, USA

ABSTRACT

There have been many approaches to the use of neural networks in control systems, such as the NARMA-L2 Controller, introduced by Narendra and Mukhopadhyay. One disadvantage of this controller is that it often produces chattering in the control action. In this paper we investigate the addition of linear feedback to the NARMA-L2 controller in order to smooth the control action. We show that under certain circumstances this linear feedback can cause problems of stability and increased steady state errors, especially when the plant model is inaccurate. These problems can be reduced significantly by the use of appropriate design strategies, which are presented in this paper.

1. INTRODUCTION

Many neural network architectures have been applied successfully for the identification and control of dynamic systems ([1], [3]). One popular neural network control architecture is the NARMA-L2 controller [4]. Training is straightforward for this method, because the controller is simply a rearrangement of a neural network plant model, which is trained offline, in batch form; there is no separate dynamic training [2] for the controller. Another advantage of NARMA-L2 is that the only online computation is a forward pass through the neural network controller. The drawback of this method is that the plant must either be in companion form, or be capable of approximation by a companion form model [6]. In addition, there is often chattering in the control action. This chattering can be reduced by the addition of linear feedback, as we will discuss in this paper.

In Section 2 of this paper we will present the basic NARMA-L2 controller and will demonstrate its operation on a simple magnetic levitation system. We will also illustrate the chattering that typically occurs in the control action. Section 3 shows how linear feedback, when added to NARMA-L2 control, can reduce the control action chattering. The resulting controller is an approximation to feedback linearization control [6]. In Section 4, we analyze the stability and steady state performance of the modified NARMA-L2 control.

2. NARMA-L2 CONTROL

NARMA-L2 control transforms nonlinear system dynamics into linear dynamics by canceling the nonlinearities. The controller is simply a rearrangement of the neural network plant model, which is trained offline, in batch form. The only online computation is a forward pass through the neural network controller. The drawback of this method is that the plant must either be in companion form, or be capable of approximation by a companion form model.

2.1 Controller Description

To identify the system to be controlled we use the approximate NARMA-L2 model to represent the system in companion form [4]:

$$\begin{aligned} y(k+1) = {} & f[y(k), y(k-1), \ldots, y(k-n+1), \\ & u(k-1), \ldots, u(k-m+1)] \\ & + g[y(k), y(k-1), \ldots, y(k-n+1), \\ & u(k-1), \ldots, u(k-m+1)] \cdot u(k) \end{aligned} \tag{1}$$

where $f(\ )$ and $g(\ )$ are approximated using neural networks. Using this NARMA-L2 model, we can define the controller

$$\begin{aligned} u(k) &= \frac{y_r(k+1) - f[\mathbf{y}_n(k), \mathbf{u}_m(k-1)]}{g[\mathbf{y}_n(k), \mathbf{u}_m(k-1)]} \\ \mathbf{y}_n(k) &= [y(k), \ldots, y(k-n+1)]^T \\ \mathbf{u}_m(k-1) &= [u(k-1), u(k-2), \ldots, u(k-m)]^T \end{aligned} \tag{2}$$

where $y_r(k+1)$ is the reference signal to be tracked. This controller can be implemented using the previously identified NARMA-L2 plant model. If the system can be represented in companion form, and if $f(\ )$ and $g(\ )$ are accurately approximated, the system output will equal the reference model output:

$$y(k+1) = y_r(k+1)\,. \tag{3}$$

In the next section we will demonstrate the operation of the NARMA-L2 controller on a magnetic levitation system.

2.2 Magnetic Levitation System (MagLev)

In this test problem, the objective is to control the position of a magnet suspended above an electromagnet, where the magnet is constrained so that it can only move in the vertical direction. The equation of motion is:

$$\frac{d^2y(t)}{dt^2} = -g + \frac{\alpha}{M}\frac{i^2(t)\,\mathrm{sgn}(i(t))}{y(t)} - \frac{\beta}{M}\frac{dy(t)}{dt}$$

where $y(t)$ is the distance of the magnet above the electromagnet, $i(t)$ is the current flowing in the electromagnet, M is the mass of the magnet, and g is the gravitational constant. The parameter β is a viscous friction coefficient that is determined by the material in which the magnet moves, and α is a field strength constant that is determined by the number of turns of wire on the electromagnet and the strength of the magnet.

The first step in using NARMA-L2 control is the identification of the plant model, in which multilayer networks are used to approximate $f(\ )$ and $g(\ )$. The inputs to these networks are delayed plant inputs and outputs ($\mathbf{u}_m(k-1)$ and $\mathbf{y}_n(k)$). Table 1 shows the plant identification parameters for the magnetic levitation system. The input signal for the system identification consists of a series of pulses of random amplitude and width (see Figure 1). The first two rows of the table indicate the ranges for the amplitudes and widths of the pulses. The last three rows define the architecture of the identification network. The $f(\ )$ and $g(\ )$ networks have the same architecture, with one hidden layer.

Table 1: Parameters for MagLev Plant Identification

	MagLev
Input range	(0, 4)
Input interval	(0.05, 5)
Sample time	0.01
Delayed inputs (m)	3
Delayed outputs (n)	3
Hidden layer size	10

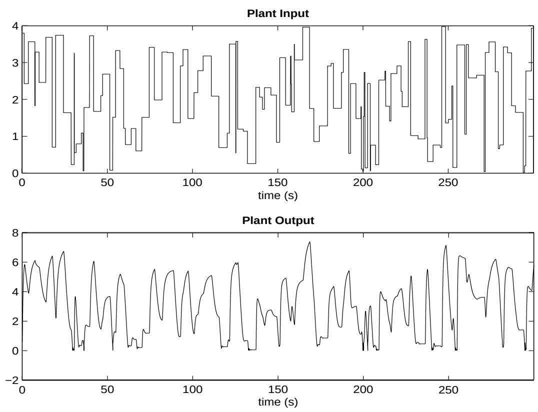

Figure 1. Sample Input and Output Signals for Identification

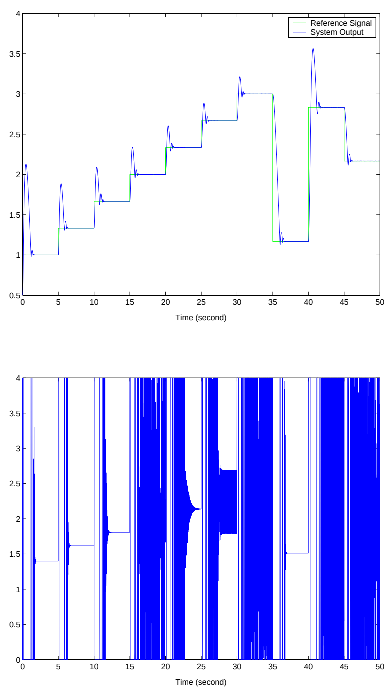

Figure 2. MagLev response and control action for the NARMA-L2 controller.

Figure 2 shows the system response and the control action for the NARMA-L2 controller when the reference signal is a series of step functions. The system tracks the reference, but the control action demonstrates significant chattering. In the next section, we will add linear feedback to the NARMA-L2 controller to reduce the chattering.

3. SMOOTHING THE CONTROL ACTION

The NARMA-L2 controller is very similar to the feedback linearization controller [6]. The main difference between the two is that the feedback linearization controller adds a linear feedback term. If we modify Eq. (2) to include the linear feedback, we have

$$u(k) = \frac{c_0 y_r(k+1) - f[\mathbf{y}_n(k), \mathbf{u}_m(k-1)] - \mathbf{d}^T \mathbf{y}_p(k)}{g[\mathbf{y}_n(k), \mathbf{u}_m(k-1)]} \qquad (4)$$

$$\mathbf{d} = [d_1, \ldots, d_p]^T$$

If $f(\ )$ and $g(\ )$ are accurately approximated, the system output will satisfy the following linear difference equation:

$$y(k+1) = c_0 y_r(k+1) - \mathbf{d}^T \mathbf{y}_p(k)$$

$$\text{or} \qquad (5)$$

$$y(k) + d_1 y(k-1) + \ldots + d_p y(k-p) = c_0 y_r(k)$$

Taking z-transforms, we have

$$Y(z) = \frac{c_0}{D(z)} Y_r(z) \qquad (6)$$

$$D(z) = 1 + d_1 z^{-1} + \ldots + d_p z^{-p}$$

By appropriately setting the roots of $D(z)$ inside the unit circle, we can have a response $y(k)$ that is a smoothed version of $y_r(k)$. This means that $u(k)$ will also be smoothed, and we can reduce the chattering found in the NARMA-L2 control. If the parameter c_0 is assigned to be

$$c_0 = D(1) = 1 + d_1 + \ldots + d_p \qquad (7)$$

then we can be assured of zero steady state error to a step input.

Figure 3 and Figure 4 show the responses and the control action for the smoothed NARMA-L2 controller on the MagLev system. For these examples, we have chosen the order of the linear feedback, p, to be 1. For Figure 3 the closed loop pole (root of $D(z)$) is set to 0.5, and for Figure 4 the pole is set to 0.95. As the pole moves toward 1, the response is smoother and the chattering in the control action is diminished. This responses should be compared to Figure 2, which is the standard NARMA-L2 response, and corresponds to having the closed loop pole set to 0.

There is clearly a trade-off to be made here between the reduction in chattering and the speed in the overall response. As the closed loop poles are moved toward 1, the chattering is reduced, but the system settling time is increased. There are also other considerations of stability and steady state error that are caused by the fact that the functions $f(\ )$ and $g(\ )$ are only approximated. As we will show in the next section, even when the roots of $D(z)$ are inside the unit circle, equilibrium points of the closed loop system can become unstable because of errors in the approximations of $f(\ )$ and $g(\ )$.

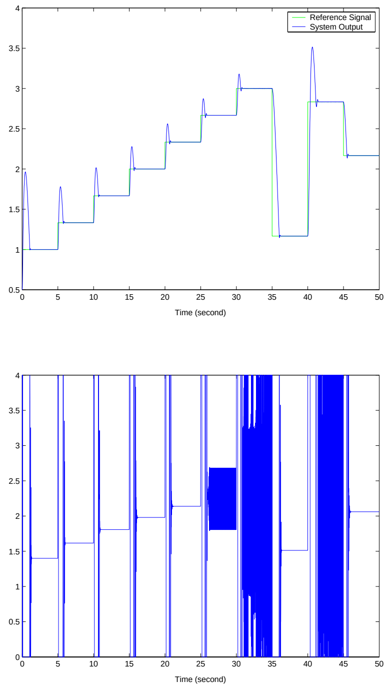

Figure 3. MagLev response and control action for the NARMA-L2 controller with linear feedback (pole=0.5).

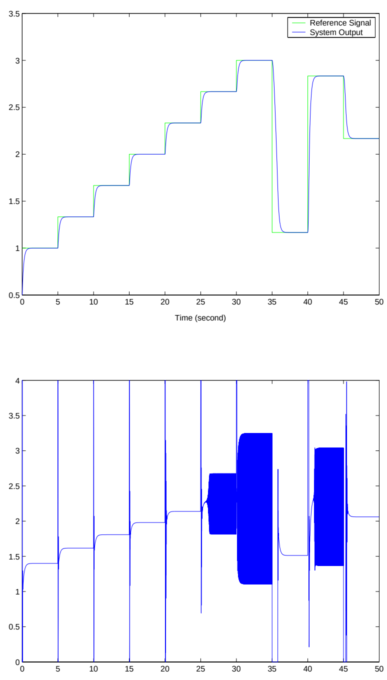

Figure 4. MagLev response and control action for the NARMA-L2 controller with linear feedback (pole=0.95).

4. ANALYSIS OF SMOOTHED NARMA-L2

The analysis of the previous section assumed that the approximations for $f(\ )$ and $g(\ )$ were exact. In practice these approximations will only be valid within the regions where training data exists. In addition, even where training data does exist, there will be some small approximation errors. In this section we will investigate the effect of these approximation errors on the response of the smoothed NARMA-L2 controller. In particular, we will investigate stability and steady state errors of the closed loop system, and how they are affected by both the approximation errors and the roots of $D(z)$.

Let's begin by modifying Eq. (4) to indicate that $f(\ )$ and $g(\ )$ are being approximated:

$$u(k) = \frac{c_0 y_r(k+1) - \hat{f}[\mathbf{y}_n(k), \mathbf{u}_m(k-1)] - \mathbf{d}^T\mathbf{y}_p(k)}{\hat{g}[\mathbf{y}_n(k), \mathbf{u}_m(k-1)]} \tag{8}$$

If we substitute this equation into Eq. (1), we find

$$\begin{aligned} y(k+1) &= f + g \cdot \left[\frac{c_0 y_r(k+1) - \hat{f} - \mathbf{d}^T\mathbf{y}_p(k)}{\hat{g}}\right] \\ &= \left[f - \frac{g}{\hat{g}} \cdot \hat{f}\right] + \frac{g}{\hat{g}} \cdot c_0 y_r(k+1) - \frac{g}{\hat{g}} \cdot \mathbf{d}^T\mathbf{y}_p(k) \end{aligned} \tag{9}$$

where we have removed the arguments for $f(\ )$ and $g(\ )$ in order to simplify the notation. Define the following variables:

$$e_g = \frac{g}{\hat{g}} \qquad e_f = f - e_g \cdot \hat{f}, \tag{10}$$

then Eq. (9) can be rewritten as

$$y(k+1) = e_f + e_g \cdot c_0 y_r(k+1) - e_g \cdot \mathbf{d}^T\mathbf{y}_p(k). \tag{11}$$

We now want to check the stability and steady state error associated with a specific equilibrium point. In other words, y_r is set to a constant value, and we want to check the location of the corresponding equilibrium point for y, and its stability. For this situation, we will assume that e_f and e_g are approximately constant.

Eq. (11) can then be rewritten as

$$\begin{aligned} y(k) + e_g \cdot \mathbf{d}^T\mathbf{y}_p(k-1) &= e_g \cdot c_0 y_r(k) + e_f \\ y(k) + e_g d_1 y(k-1) + \ldots + e_g d_p y(k-p) &= e_g \cdot c_0 y_r(k) + e_f \end{aligned} \tag{12}$$

Taking z-transforms we have

$$\begin{aligned} Y(z) &= \frac{1}{D_e(z)}\left[e_g c_0 Y_r(z) + \frac{z e_f}{z-1}\right] \\ D_e(z) &= 1 + e_g d_1 z^{-1} + \ldots + e_g d_p z^{-p} \end{aligned} \tag{13}$$

Although the roots of $D(z)$ are designed to fall inside the unit circle, it is the roots of $D_e(z)$ that will determine the stability of the equilibrium point for the closed loop system. If we place the roots of $D(z)$ too close to the unit circle (in order to eliminate chattering), then the roots of $D_e(z)$ may fall outside the unit circle.

If the roots of $D_e(z)$ fall inside the unit circle, then we can find the steady state response to a step input of amplitude A ($y_r(k) = A\mu_s(k)$) from the following equation.

$$y_{ss} = \frac{e_g c_0 A + e_f}{D_e(1)}$$

$$D_e(1) = 1 + e_g \sum_{i=1}^{p} d_i \qquad (14)$$

If $e_g = 1$, $e_f = 0$ and c_0 is set using Eq. (7), then $y_{ss} = A$ and the steady state error will be zero.

We can use Eq. (14) to determine the values of e_g and e_f, which would then allow us to predict the stability and steady state error as we move the roots of $D(z)$ closer to the unit circle. To determine these values, we can evaluate the steady state error for two different settings of $D(z)$. This will give us two equations in two unknowns.

We will demonstrate this analysis on the MagLev system, with the reference position set at 2. We simulated the closed-loop MagLev system at two different settings for $D(z)$. The first setting had three roots at 0.94; the second setting had three roots at 0.95. The results are as follows:

$$\lim_{k \to \infty} y^{[0.94]^3}(k) = 2.0288926 \qquad (15)$$

$$\lim_{k \to \infty} y^{[0.95]^3}(k) = 2.0453426 \qquad (16)$$

Using these two values and Eq. (14), we can compute

$$e_g = 0.99996517, e_f = -7.6906918 \times 10^{-5}. \qquad (17)$$

From these values we can determine the roots of $D_e(z)$ to be

$$p_1 = 0.92964$$
$$p_2 = 0.97512 + 0.02804j \qquad (18)$$
$$p_3 = 0.97512 - 0.02804j$$

when the roots of $D(z)$ are set to 0.96. The roots of $D_e(z)$ fall inside the unit circle, so we would expect this equilibrium point to be stable, and simulation demonstrates this to be true. If we perform a similar analysis for $D(z)$ containing three roots at 0.97, we find that the roots of $D_e(z)$ are located at

$$p_1 = 0.889147$$
$$p_2 = 1.00933 + 0.08327j \qquad (19)$$
$$p_3 = 1.00933 - 0.08327j$$

This indicates an unstable equilibrium point, which was verified by computer simulation.

Eq. (14) can also be used to predict the steady state error at a new choice for $D(z)$. To illustrate, consider again the MagLev system at the set point of 2, with the three roots of $D(z)$ set to 0.96. We know that the equilibrium point is stable, based on the pole locations given by Eq. (18). Using the values for e_f and e_g given in Eq. (17), we use Eq. (14) to compute the steady state response. The resulting calculation was 2.0733287. From the computer simulation we obtained a value of 2.0487326.

To summarize our analysis, errors in the approximation of the functions $f(\ )$ and $g(\ )$ can cause steady state errors in the system response, or even instabilities. The system is more likely to become unstable if the roots of $D(z)$ are near the unit circle. If it is necessary to place closed loop poles near the unit circle, it is critical that the network approximations for $f(\ )$ and $g(\ )$ be very accurate, which may require larger training sets. See [5] for a discussion of how to increase the training set to improve the approximation.

5. SUMMARY

This paper began with a review of the NARMA-L2 controller. This controller is attractive because it requires a minimal amount of computation, both for training of the neural networks and for implementation of the controller. The controller is simply a rearrangement of the neural network plant model, which is trained offline, in batch form. The only online computation is a forward pass through the neural network controller. A drawback of this method is that the plant must either be in companion form, or be capable of approximation by a companion form model. In addition, this controller is characterized by chattering in the control action.

Control chattering in the NARMA-L2 controller can be significantly reduced by adding linear feedback, as in the standard feedback linearization controller. By using the linear feedback to position the closed loop poles near the unit circle, the system response can be significantly smoothed and control chattering greatly reduced.

Because the system model is only approximate, the cancellation of nonlinearities will not be exact. The errors in the approximation can cause instabilities or steady state errors in the closed loop system response. For these reasons, the desired closed loop poles should not be placed too close to the unit circle.

6. REFERENCES

[1] Hagan, M.T. and Demuth, H.B., “Neural Networks for Control,” *Proceedings of the 1999 American Control Conference*, San Diego, CA, 1999, pp. 1642-1656.

[2] Hagan, M.T., De Jesús, O., and Schultz, R., “Training Recurrent Networks for Filtering and Control,” Chapter 12 in *Recurrent Neural Networks: Design and Applications*, L. Medsker and L.C. Jain, Eds., CRC Press, 1999, pp. 311-340.

[3] Hunt, K.J., Sbarbaro, D., and Zbikowski, R., Gawthrop, P.J., “Neural Networks for Control System - A Survey,” *Automatica*, Vol. 28, 1992, pp. 1083-1112.

[4] Narendra, K.S. and Mukhopadhyay, S., “Adaptive Control Using Neural Networks and Approximate Models,” *IEEE Transactions on Neural Networks* Vol. 8, 1997, pp. 475-485.

[5] De Jesus, O., Pukrittayakamee, A., and Hagan, M. T., “A Comparison of Neural Network Control Algorithms,” *Proceedings of the International Joint Conference on Neural Networks*, vol.4., 2001, 2626 -2631.

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