Article title: Design and Control of a Vertically Moving Base Inverted Pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm

Authors: Mustefa Jibril[1], Messay Tadese[2], Reta Degefa[3]

Affiliations: School of Electrical and Computer Engineering[1]

Orcid ids: 0000-0002-3165-2410[1]

Contact e-mail: mustefazinet1981@gmail.com

License information: This work has been published open access under Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at https://www.scienceopen.com/.

Preprint statement: This article is a preprint and has not been peer-reviewed, under consideration and submitted to ScienceOpen Preprints for open peer review.

DOI: 10.14293/S2199-1006.1.SOR-.PPVNR50.v1

Preprint first posted online: 27 July 2020

Keywords: Inverted pendulum, Nonlinear autoregressive moving average L2
**Design and Control of a Vertically Moving Base Inverted Pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm**

Mustefa Jibril\(^1\), Messay Tadese\(^2\), Reta Degefa\(^3\)

\(^1\) Msc, School of Electrical & Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia
\(^2\) Msc, School of Electrical & Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia
\(^3\) Msc, School of Electrical & Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia

mustefa.jibril@du.edu.et

**Abstract**

In this paper, a vertically moving base inverted pendulum control analysis has been done using Matlab/Simulink Toolbox. Because the vertically moving base inverted pendulum system is nonlinear and highly unstable, a feedback control system is used to make the system controlled and stable. A nonlinear autoregressive moving average L2 controller which is a family of Neural Network controller is used with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm to improve the stability of the pendulum. Comparison of the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm for tracking a desired angular position of the system using a step and random input signals and a promising results have been obtained successfully.

**Keywords:** Inverted pendulum, Nonlinear autoregressive moving average L2

**1. Introduction**

An inverted pendulum is a oscillator which has its crowd above its pivot point. It is often implemented with the pivot kernel mounted on a base that can move vertically. The inverted pendulum is a classic problem in control system and is widely used as a measure for experimenting control algorithms (PID controllers, neural networks, fuzzy control, genetic algorithms, etc.). Variations on this problem include multiple links, allowing the motion of the base to be commanded while arranging the pendulum, and evenness the cart-pendulum system on a see-saw. The inverted pendulum is related to rocket or missile guidance, where propulsion is actuated at the beds of a tall vehicle. The understanding of a similar funeral is built in the technology of Segway, a self-balancing conveyance device. The largest implemented use are on huge lifting cranes on shipyards. When stirring the shipping containers back and forth, the cranes demand the bins accordingly so that it never swings or sways. It always stays perfectly positioned under the worker even when moving or stopping quickly. Another resources that an inverted pendulum may be stabilized, without any feedback or control mechanism, is by oscillating the support rapidly up and down. If the oscillation is sufficiently strong (in terms of its acceleration and amplitude) then the inverted pendulum can recover from perturbations in a strikingly counterintuitive manner.

**2. System Discription of the Pendulum**
The vertically moving base inverted pendulum schematic diagram is shown in Figure 1. The rod is considered massless. The pointmass at the end of the rod is denoted by $m$. The rod has a length $l$.

![Diagram of Vertically Moving Base Inverted Pendulum](image)

Figure 1 Vertically moving base inverted pendulum system

The equation of motion for a vertically moving base inverted pendulum is derived using the Lagrangian. The position of the point mass is now given by:

$$\left(l \sin \theta, y + l \cos \theta \right) \quad (1)$$

and the velocity is found by taking the first derivative of the position:

$$v^2 = \dot{y}^2 - 2l \dot{\theta} \dot{y} \sin \theta + l^2 \dot{\theta}^2 \quad (2)$$

The Lagrangian for this system can be written as:

$$L = \frac{1}{2} m \left(\ddot{y}^2 - 2l \dot{\theta} \dot{y} \sin \theta + l^2 \dot{\theta}^2\right) - mg \left(y + l \cos \theta\right) \quad (3)$$

and the equation of motion follows from:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta}\right) = 0$$

resulting in:

$$l \ddot{\theta} = \dot{y} \sin \theta + g \sin \theta \quad (4)$$

For small angle approximation: $\theta \ll 1$, Equation (4) becomes

$$l \ddot{\theta} = \dot{y} \theta + g \theta \quad (5)$$
The system is nonlinear and the block diagram of the system is shown in Figure 2 below.

![Figure 2 Block diagram of vertically moving base inverted pendulum](image)

The system parameters are shown in Table 1 below.

| No | Parameter                          | Symbol | Value       |
|----|------------------------------------|--------|-------------|
| 1  | Mass of the pendulum               | \( m \) | 0.6 Kg      |
| 2  | Pendulum rod length                | \( l \) | 0.45 m      |
| 3  | Acceleration due to gravity        | \( g \) | 10 m/s^2    |

### 3. Proposed Controllers Design

#### 3.1 Design of NARMA-L2 Controller

The neuro controller described on this phase is cited through two different names: response linearization control and NARMA-L2 manipulate. It is known as comments linearization when the plant shape has a specific form (associate form). It is known as NARMA-L2 manipulate while the fortification mold may be approximated by using the same form. The vital principle of this type of control is to convert nonlinear design system into linear dynamics with the aid of canceling the nonlinearities. This phase starts off evolved with the aid of submitting the associate system form and presentation how you may use a neural community to become aware of this model. Then it describes how the identified neural network model may be used to broaden a controller.

#### 3.1.1 Identification of the NARMA-L2 Model:

The first step in the use of feedback linearization (or NARMA-L2) manipulate is to identify the design to be controlled. You train a neural network to represent the forward dynamics of the system.

The first step is to pick out a styles association to use. One standard patterns this is used to symbolize fashionable discrete-time nonlinear system is the nonlinear autoregressive-moving average (NARMA) model:

\[
y(k+d) = N[y(k), y(k-1),..., y(k-n+1), u(k), u(k-1),..., u(k-n+1)]
\]  

(6)
Where \( u(k) \) is the system input, and \( y(k) \) is the system output. For the identification section, you can teach a neural network to approximate the nonlinear function \( N \). If you want the system output to follow some reference trajectory \( y(k+d) = y_r(k+d) \) the subsequent step is to expand a nonlinear controller of the form:

\[
 u(k) = G\left[ y(k), y(k-1), \ldots, y(k-n+1), y_r(k+d), u(k-1), \ldots, u(k-m+1) \right] \tag{7}
\]

The trouble with the usage of this controller is that in case you need to teach a neural network to create the characteristic \( G \) to minimize mean square blunders, you need to apply dynamic returned propagation. This can be pretty sluggish. One answer is to apply approximate models to symbolize the system. The controller used on this section is based totally at the NARMA-L2 approximate model:

\[
 \tilde{y}(k+d) = f\left[ y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1) \right] + g\left[ y(k), y(k-1), \ldots, y(k-n+1) \right] u(k) \tag{8}
\]

This model is in associate shape, wherein the next controller input \( u(k) \) is not contained in the nonlinearity. The gain of this form is that you could resolve for the control input that causes the system output to comply with the reference \( y(k+d) = y_r(k+d) \). The resulting controller would have the form

\[
 u(k) = \frac{y_r(k+d) - f\left[ y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-n+1) \right]}{g\left[ y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-n+1) \right]} \tag{9}
\]

Using this equation immediately can motive awareness problems, due to the fact you ought to determine the control input \( u(k) \) primarily based on the output at the same time, \( y(k) \). So, rather, use the model

\[
 y(k+d) = f\left[ y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1) \right] + g\left[ y(k), y(k-1), \ldots, y(k-n+1), u(k-1), \ldots, u(k-m+1) \right] u(k+1) \tag{10}
\]

Where \( d \geq 2 \). Figure 3 shows the structure of a neural network representation.
Figure 3. The structure of a neural network representation.

Using the NARMA-L2 model, you can obtain the controller

$$u(k+1) = \frac{y(k+d) - f(y(k), y(k-1), ..., y(k-n+1), u(k), ..., u(k-n+1))}{g(y(k), y(k-1), ..., y(k-n+1), u(k), ..., u(k-n+1))}$$ (11)

Which is realizable for $d \geq 2$. Figure 4 shows the block diagram of the NARMA-L2 controller.

Figure 4. Block diagram of the NARMA-L2 controller

This controller can be implemented with the formerly diagnosed NARMA-L2 plant model, as shown in Figure 5 below.

Figure 5. Previously identified NARMA-L2 plant model

Table 2 illustrates the network architecture, training data and training parameters of the proposed controllers.
Table 2 Neural network Parameters

| Network Architecture |          |          |          |
|----------------------|----------|----------|----------|
| Size of hidden layer | 9        | Delayed  | 3        |
|                      |          | plant    |          |
|                      |          | input    |          |
| Sample interval(sec) | 0.01     | Delayed  | 2        |
|                      |          | plant    |          |
|                      |          | output   |          |

| Training Data        |          |          |          |
|----------------------|----------|----------|----------|
| Training sample      | 100      | Maximum  | 1        |
|                      |          | Plant    |          |
|                      |          | output   |          |
| Maximum Plant input  | 4        | Minimum  | 0        |
|                      |          | Plant    |          |
|                      |          | output   |          |
| Minimum Plant input  | 1        | Max      | 01       |
|                      |          | interval | value     |
|                      |          | (sec)    | 0.1      |

| Training Parameters  |          |          |          |
|----------------------|----------|----------|----------|
| Training Epochs      | 100      |          |          |

### 3.2 Levenberg-Marquardt Algorithm

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as

\[ H = J^T J \quad (12) \]

and the gradient can be computed as

\[ g = J^T e \quad (13) \]

where \( J \) is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and \( e \) is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

\[ x_{k+1} = x_k - \left[ J^T J + \mu I \right]^{-1} J^T e \quad (14) \]

When the scalar \( \mu \) is zero, this is just Newton's method, using the approximate Hessian matrix. When \( \mu \) is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, \( \mu \) is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm.

### 3.3 Resilient Backpropagation Algorithm


Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when you use steepest descent to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value.

4. Result and Discussion
4.1 Comparison of the Vertically Moving Base Inverted Pendulum using NARMA-L2 with Resilient Backpropagation and Levenberg Marquardt Backpropagation Algorithm for Step Input signal

The simulink model of the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation algorithm for step input signal is shown in Figure 6 below.

Figure 6 Simulink model of the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation algorithm for step input signal

The vertically moving base inverted pendulum subsystem and the simulation result are shown in Figure 7 and Figure 8 respectively.
Figure 7 vertically moving base inverted pendulum subsystem

Figure 8 Step response simulation result

The data of the rise time, percentage overshoot, settling time and peak value is shown in Table 3.

Table 3 Step response data

| No | Performance Data | Resilient backpropagation | Levenberg Marquardt backpropagation |
|----|------------------|----------------------------|-------------------------------------|
| 1  | Rise time        | 1.12 sec                   | 1.13 sec                            |
| 2  | Per. overshoot   | 8 %                        | 40 %                                |
| 3  | Settling time    | 1.8 sec                    | 4 sec                               |
| 4  | Peak value       | 2.7 Degree                 | 3.5 Degree                          |

As Table 3 shows that the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation algorithm improves the performance of the system by minimizing the rise time, percentage overshoot and settling time.
4.2 Comparison of the Vertically Moving Base Inverted Pendulum using NARMA-L2 with Resilient Backpropagation and Levenberg Marquardt Backpropagation Algorithm for Random Input signal

The simulink model of the vertically moving base inverted pendulum base using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation algorithm for random input signal is shown in Figure 9 below.

![Simulink Model](image)

Figure 9 Simulink model of the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation algorithm for random input signal

The vertically moving base inverted pendulum system simulation result are shown in Figure 10 below.

![Simulation Result](image)

Figure 10 Random input signal simulation result

Figure 10 shows that the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation algorithm improves the performance of the system by minimizing the percentage overshoot and tracking the reference signal.
5. Conclusion

The design and control of a vertically moving base inverted pendulum using a nonlinear autoregressive moving average L2 controller with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm have been analyzed. Comparison of the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation and Levenberg Marquardt backpropagation Training Algorithm for tracking a desired angular position of the system using a step and random input signals have been done. The step response simulation result shows that the vertically moving base inverted pendulum using NARMA-L2 with Resilient backpropagation algorithm improves the performance of the system by minimizing the rise time, percentage overshoot and settling time while the random input response simulation result shows that the vertically moving base inverted pendulum base using NARMA-L2 with Resilient backpropagation algorithm improves the performance of the system by minimizing the percentage overshoot and tracking the reference signal.

Reference

[1]. R. E. Grundy “The Kapitza Equation for the Inverted Pendulum” The Quarterly Journal of Mechanics and Applied Mathematics, Vol. 72, Issue 2, pp. 261-272, 2019.
[2]. Reetam Mondal et al. “Optimal Fractional Order $PI^{\alpha}D^{\beta}$ Controller for Stabilization of Cart Inverted Pendulum System: Experimental Results” Asian Journal of Control, Vol. 22, Issue 3, 2019.
[3]. Amira Tiga et al. “Nonlinear/Linear Switched Control of Inverted Pendulum System: Stability Analysis and Real-Time Implementation” Journal of Mathematical Problems in Engineering, Vol. 2019, Article ID 2391587, 10 pages, 2019.
[4]. Morasso P. et al. “Quiet Standing: The Single Inverted Pendulum Model is Not So Bad After All” PLOS ONE, Vol. 14, Issue 3, 2019.
[5]. Mohammed Rabah et al. “Comparison of Position Control of a Gyroscopic Inverted Pendulum using PID, Fuzzy Logic and Fuzzy PID Controllers” International Journal of Fuzzy Logic and Intelligent Systems, Vol. 18, Issue 2, pp. 103-110, 2018.
[6]. Beata K. et al. “Inverted Pendulum Model Linear Quadratic Regulator (LQR)” Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry and High-Energy Physics Experiments, 2018.
[7]. Elisa Sara V. et al. “Optimal Control of Inverted Pendulum System using PID Controller, LQR and MPC” IOP Conference Series: Materials Science and Engineering, Vol. 263, Issue 5, 2018.