Journal of Engineering and Applied Sciences
MEDWELL PUBLICATIONS

Inverted Pendulum Control using NARMA-L2 with Resilient Backpropagation and Levenberg Marquardt Backpropagation Training Algorithm

Mustefa Jibril, Messay Tadese and Nuriye Hassen
School of Electrical and Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia

Key words: Inverted pendulum, NARMA-L2, resilient backpropagation, Levenberg Marquardt backpropagation

Corresponding Author:
Mustefa Jibril
School of Electrical and Computer Engineering, Dire Dawa Institute of Technology, Dire Dawa, Ethiopia

Page No.: 324-330
Volume: 16, Issue 10, 2021
ISSN: 1816-949x
Journal of Engineering and Applied Sciences

Abstract: In this study, the performance of inverted pendulum has been Investigated using neural network control theory. The proposed controllers used in this study are NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm controllers. The mathematical model of Inverted Pendulum on a Cart driving mechanism have been done successfully. Comparison of an inverted pendulum with NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm controllers for a control target deviation of an angle from vertical of the inverted pendulum using two input signals (step and random). The simulation result shows that the inverted pendulum with NARMA-L2 with resilient backpropagation controller to have a small rise time, settling time and percentage overshoot in the step response and having a good response in the random response too. Finally, the inverted pendulum with with NARMA-L2 with resilient backpropagation controller shows the best performance in the overall simulation result.

INTRODUCTION

An inverted pendulum is a pendulum that has its center of mass above its pivot point. It is unstable and without additional help will fall over. It can be suspended stably in this inverted position by using a control system to monitor the angle of the pole and move the pivot point horizontally back under the center of mass when it starts to fall over, keeping it balanced. The inverted pendulum is a classic problem in dynamics and control theory and is used as a benchmark for testing control strategies. It is often implemented with the pivot point mounted on a cart that can move horizontally under control of an electronic servo system as shown in the photo; this is called a cart and pole apparatus. Most applications limit the pendulum to 1 degree of freedom by affixing the pole to an axis of rotation. Whereas a normal pendulum is stable when hanging downwards, an inverted pendulum is inherently unstable and must be actively balanced in order to remain upright; this can be done either by applying a torque at the pivot point by moving the pivot point horizontally as part of a feedback system, changing the rate of rotation of a mass mounted on the pendulum on an axis parallel to the pivot axis and thereby generating a net torque on the pendulum, or by oscillating the pivot point vertically. A simple demonstration of moving the pivot point in a feedback system is achieved by balancing an upturned broomstick on the end of one's finger[1].

A second type of inverted pendulum is a tiltmeter for tall structures which consists of a wire anchored to the bottom of the foundation and attached to a float in a pool of oil at the top of the structure that has devices for measuring movement of the neutral position of the float away from its original position.

A pendulum with its bob hanging directly below the support pivot is at a stable equilibrium point; there is no torque on the pendulum, so, it will remain motionless and if displaced from this position will experience a restoring torque which returns it toward the equilibrium position. A pendulum with its bob in an inverted position, supported on a rigid rod directly above the pivot, 180° from its stable equilibrium position, is at an unstable equilibrium point. At this point again there is no torque on the pendulum but the slightest displacement away from this position will cause a gravitation torque on the pendulum which will accelerate it away from equilibrium and it will fall over[2].

In order to stabilize a pendulum in this inverted position, a feedback control system can be used which monitors the pendulum's angle and moves the position of the pivot point sideways when the pendulum starts to fall over, to keep it balanced. The inverted pendulum is a classic problem in dynamics and control theory and is widely used as a benchmark for testing control algorithms (PID controllers, state space representation, neural networks, fuzzy control, genetic algorithms, etc.). Variations on this problem include multiple links, allowing the motion of the cart to be commanded while maintaining the pendulum and balancing the cart-pendulum system on a see-saw. The inverted pendulum is related to rocket or missile guidance where the center of gravity is located behind the center of drag causing aerodynamic instability. The understanding of a similar problem can be shown by simple robotics in the form of a balancing cart. Balancing an upturned broomstick on the end of one's finger is a simple demonstration and the problem is solved by self-balancing personal transporters such as the Segway PT, the self-balancing hoverboard and the self-balancing unicycle.

MATERIALS AND METHODS

Mathematical model of the inverted pendulum: The free body diagram of the inverted pendulum is shown in Fig. 1.

Summing the forces in the free body diagram of the cart in the horizontal direction, you get the following equation of motion:

$$M\ddot{z} + D\dot{z} + Q = F \tag{1}$$

The force exerted in the horizontal direction due to the moment on the pendulum is determined as follows:

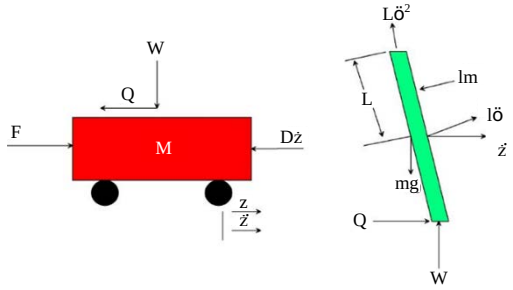

Fig. 1: Free body diagrams of the inverted pendulum system

$$\tau = r \times F = I\ddot{\theta}$$
$$F = \frac{I\ddot{\theta}}{r}$$
$$= \frac{mI^2\ddot{\theta}}{I}$$
$$= mI\ddot{\theta}$$

Component of this force in the direction of Q is $mI\ddot{\theta}\cos\theta$. The component of the centripetal force acting along the horizontal axis is as follows:

$$F = \frac{I\dot{\theta}^2}{r}$$
$$= \frac{mI^2\dot{\theta}^2}{I}$$
$$= mI\dot{\theta}^2$$

Component of this force in the direction of Q is $mI\dot{\theta}^2\sin\theta$. Summing the forces in the Free Body Diagram of the pendulum in the horizontal direction, you can get an equation for Q:

$$Q = m\ddot{z} + mI\ddot{\theta}\cos\theta - mI\dot{\theta}^2\sin\theta \tag{2}$$

If you substitute this Eq. 2 into the first equation (1), you get the first equation of motion for this system:

$$(M+m)\ddot{z} + D\dot{z} + mI\ddot{\theta}\cos\theta - mI\dot{\theta}^2\sin\theta = F \tag{3}$$

To get the second equation of motion, sum the forces perpendicular to the pendulum. This axis is chosen to simplify mathematical complexity. Solving the system along this axis ends up saving you a lot of algebra. Just as the previous equation is obtained, the vertical components of those forces are considered here to get the following equation[3]:

$$W\sin\theta + Q\cos\theta - mg\sin\theta = mI\ddot{\theta} + m\ddot{z}\cos\theta \tag{4}$$

To get rid of the P and N terms in the equation above, sum the moments around the centroid of the pendulum to get the following equation:

$$-WI\sin\theta - QI\cos\theta = I\ddot{\theta} \tag{5}$$

Combining these last two equations, you get the second dynamic equation:

$$\left(I + mI^2\right)\ddot{\theta} + mgI\sin\theta = -mI\ddot{z}\cos\theta$$

The set of equations completely defining the dynamics of the inverted pendulum are:

$$\left(M + m\right)\ddot{z} + D\dot{z} + mI\ddot{\theta}\cos\theta - mI\dot{\theta}^2\sin\theta = F$$
$$\left(I + mI^2\right)\ddot{\theta} + mgI\sin\theta = -mI\ddot{z}\cos\theta$$

These two equations are non-linear and need to be linearized for the operating range. Since, the pendulum is being stabilized at an unstable equilibrium position which is 'Pi' radians from the stable equilibrium position, this set of equations should be linearized about theta = Pi. Assume that theta = Pi+ø (where ø represents a small angle from the vertical upward direction). Therefore, cos (theta) = -1, sin (theta) = -ø and (d (theta)/dt)^2 = 0. After linearization the two equations of motion become (where u represents the input):

$$\left(M + m\right)\ddot{z} + D\dot{z} - mI\ddot{\phi} = u$$
$$\left(I + mI^2\right)\ddot{\phi} - mgI\phi = mI\ddot{z}$$

The transfer function of inverted pendulum, a DC motor, Cart and Cart driving system will be:

$$\frac{\Phi(s)}{E(s)} = \frac{1.3s}{0.272s^3 + 0.34s^2 - 0.8s - 1}$$

Where:
E(s) = Error voltage and
Φ(s) = Angular position of the pendulum

The parameters of the system are shown in Table 1.

Table 1: Parameters of the inverted pendulum

Model parameter	Symbols	Symbol's value
Mass of the Cart	M	1 (kg)
Mass of the pendulum	m	0.3 (kg)
Friction of the Cart	D	0.000 (N/m/sec)
Length of pendulum to center of gravity	L	0.26 (m)
Moment of inertia (Pendulum)	I	0.007 (kg m^{-2})
Radius of pulley	r	0.04 (m)
Force applied to the cart	F	
Cart position coordinate	z	
Pendulum angle with the vertical	φ	

Proposed controllers design

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)[4]. 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 cancelling the non-linearities. 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[5].

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[6].

The first step is to pick out a styles association to use. One standard pattern this is used to symbolize fashionable discrete-time nonlinear system is the Nonlinear Autoregressive-Moving Average (NARMA) model:

$$y(k+d) = N\begin{bmatrix} y(k), y(k-1), ..., \\ y(k-n+1), u(k), u(k-1), ..., \\ u(k-n+1) \end{bmatrix} \tag{6}$$

Where:
u(k) = The system input
y(k) = 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) = yr(k+d) the subsequent step is to expand a nonlinear controller of the form:

$$u(k) = G\begin{bmatrix} y(k), y(k-1), ..., \\ y(k-n+1), y_r(k+d), \\ u(k-1), ..., u(k-m+1) \end{bmatrix} \tag{7}$$

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[7]:

$$\hat{y}(k+d)=f\begin{bmatrix}y(k),y(k-1),...,\\ y(k-n+1),u(k-1),...,\\ u(k-m+1)\end{bmatrix}+ g\begin{bmatrix}y(k),y(k-1),...,y(k-n+1),\\ u(k-1),...,u(k-m+1)\end{bmatrix}u(k) \quad (8)$$

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) = yr(k+d). The resulting controller would have the form:

$$u(k)=\frac{\begin{bmatrix}y_r(k+d)-f\begin{pmatrix}y(k),y(k-1),...,\\ y(k-n+1),u(k-1),.....,\\ u(k-n+1)\end{pmatrix}\end{bmatrix}}{g\begin{bmatrix}y(k),y(k-1),...,y(k-n+1),\\ u(k-1),.....,u(k-n+1)\end{bmatrix}} \quad (9)$$

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[8]:

$$y(k+d)=f\begin{bmatrix}y(k),y(k-1),..,y(k-n+1),\\ u(k-1),..,u(k-m+1)\end{bmatrix}+ g\begin{bmatrix}y(k),y(k-1),..,y\\ (k-n+1),u(k-1),..,u(k-m+1)\end{bmatrix}u(k+1) \quad (10)$$

where d≥2. Figure 2 shows the structure of a neural network representation. Using the NARMA-L2 model, you can obtain the controller:

$$u(k+1)=\frac{\begin{bmatrix}y_r(k+d)-f\begin{pmatrix}y(k),y(k-1),...,\\ y(k-n+1),u(k),.....,\\ u(k-n+1)\end{pmatrix}\end{bmatrix}}{g\begin{bmatrix}y(k),y(k-1),...,y(k-n+1),\\ u(k),.....,u(k-n+1)\end{bmatrix}} \quad (11)$$

which is realizable for d≥2. Figure 3 shows the block diagram of the NARMA-L2 controller. This controller can be implemented with the formerly diagnosed NARMA-L2 plant model, as shown in Fig. 4. Table 2 illustrates the network architecture, training data and training parameters of the proposed controllers.

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 feed-forward 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

Table 2: Neural network parameters

Network architecture	Values	Variables	Values
Size of hidden layer	6	Delayed plant input	2
Sample interval(sec)	1	Delayed plant output	3
Training data			
Training sample	100	Maximum Plant output	3
Maximum plant input	1	Minimum Plant output	1
Minimum plant input	1	Max interval value (sec)	3
Min interval value (sec)			1.5
Training parameters			
Training epochs			100

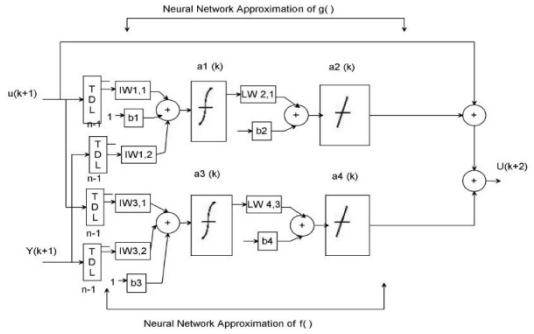

Fig. 2: The structure of a neural network representation

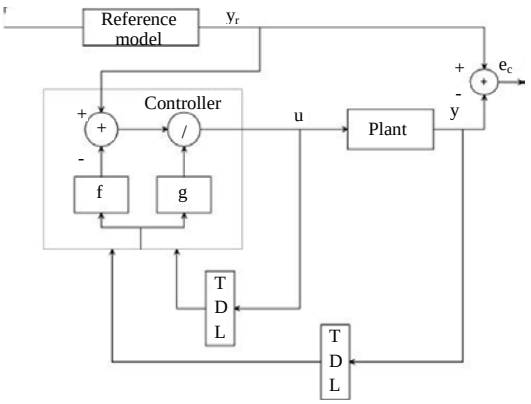

Fig. 3: Block diagram of the NARMA-L2 controller

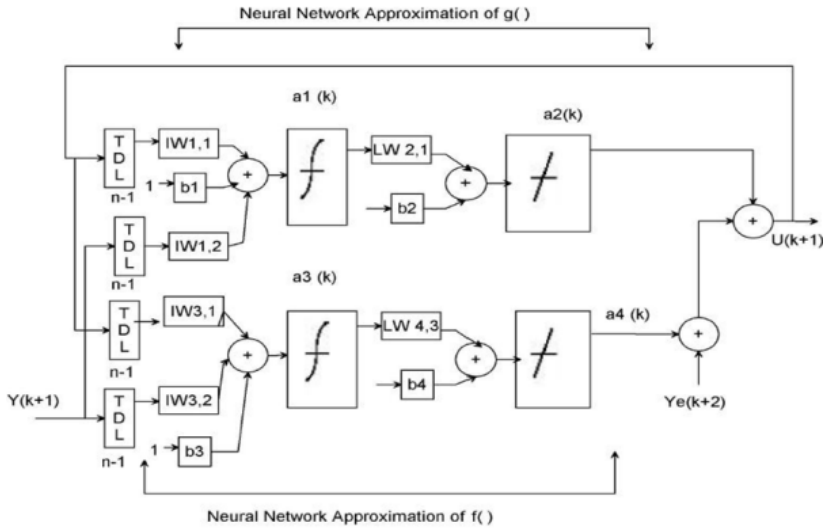

Fig. 4: Previously identified NARMA-L2 plant model

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 μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ 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, μ 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.

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.

RESULTS AND DISCUSSION

Comparison of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for step input signal: The simulink model of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for step input signal is shown in Fig. 5.

The inverted pendulum withthe proposed controllers simulation result are shown in Fig. 6. The data of the rise time, percentage overshoot, settling time and peak value is shown in Table 3[9].

As Table 3 shows that the inverted pendulum with an oscillatory base using NARMA-L2 with resilient backpropagation algorithm improves the performance of the system by minimizing the rise time, percentage overshoot and settling time.

Comparison of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for random input signal: The simulink model of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for random input signal is shown in Fig. 7[10].

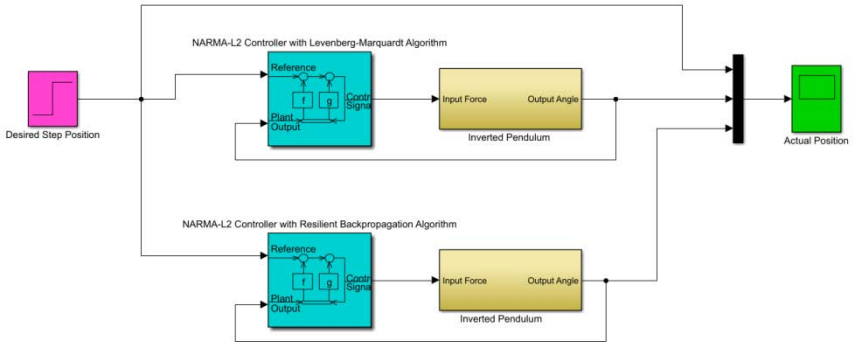

Fig. 5: Simulink model of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for step input signal

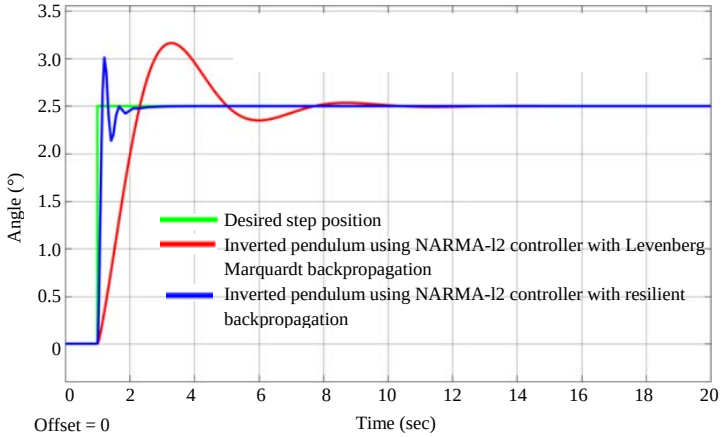

Fig. 6: Step response simulation result

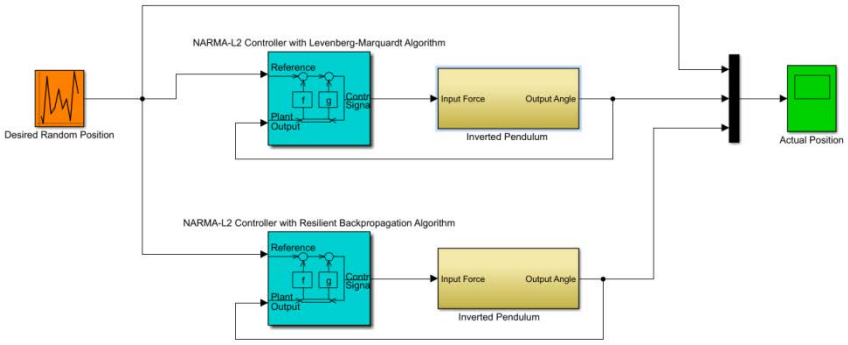

Fig. 7: Simulink model of the inverted pendulum using NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm for random input signal

The inverted pendulum with the proposed controllers simulation result are shown in Fig. 8. Figure 8 shows that the inverted pendulum with an oscillatory base using NARMA-L2 with resilient backpropagation algorithm improves the performance of the system by minimizing the percentage overshoot and tracking the reference signal[11].

Table 3: Step response data

Performance data	Resilient backpropagation	Levenberg Marquardt backpropagation
Rise time	1.12 (sec)	1.65 (sec)
Per. overshoot	20 (%)	32 (%)
Settling time	2.4 (sec)	10.5 (sec)
Peak value	3 (°)	3.3 (°)

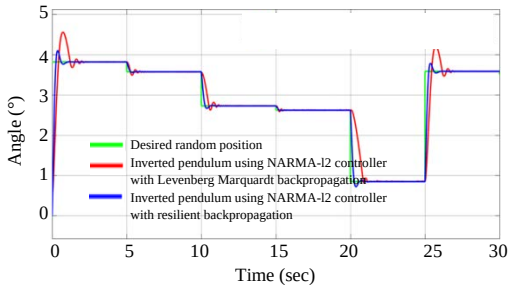

Fig. 8: Random input signal simulation result

CONCLUSION

In this study, performance investigation of an inverted pendulum system using neural network theory have been analysed and simulated successfully. The mathematical model of inverted pendulum on a Cart and Cart driving mechanism have been developed. The inverted pendulum with NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm controllers have been designed and the comparison of the inverted pendulum with NARMA-L2 with resilient backpropagation and Levenberg Marquardt backpropagation algorithm controllers using step and random input desired position signals have been done using MATLAB/Simulink. The simulation results prove that the inverted pendulum with NARMA-L2 with resilient backpropagation controller shows improvement in minimizing the rise time, settling time and percentage overshoot than the inverted pendulum NARMA-L2 with Levenberg Marquardt backpropagation algorithm controller. Finally, the comparative and simulation results prove the effectiveness of the inverted pendulum with NARMA-L2 with resilient backpropagation controller[12].

REFERENCES

01. Chawla, I., V. Chopra and A. Singla, 2021. Robust stabilization control of a spatial inverted pendulum using integral sliding mode controller. Int. J. Nonlinear Sci. Numer. Simul., 22: 183-195.
02. Iqbal, S., M. Ayyub and M. Sarfraz, 2020. Advanced fuzzy logic control application for tracking inverted pendulum. Int. J. Adv. Sci. Technol., 29: 8049-8057.
03. Maneetham, D. and P. Sutyasadi, 2020. System design for inverted pendulum using LQR control via IoT. Int. J. Simul. Multidiscip. Des. Optim., Vol. 11, 10.1051/smdo/2020007
04. Silik, Y. and U. Yaman, 2020. Control of rotary inverted pendulum by using on-off type of cold gas thrusters. Actuators, Vol. 9, No. 4. 10.3390/act9040095
05. Sanjeewa, S.D. and M. Parnichkun, 2019. Control of rotary double inverted pendulum system using mixed sensitivity H8 controller. Int. J. Adv. Rob. Syst., Vol. 16, No. 2. 10.1177/1729881419833273
06. EL-Hossainy, M., R. Shalaby and B. Abo-Zalam, 2019. Modified PDOS-based LQG controller for inverted pendulum. Menoufia J. Electron. Eng. Res., 28: 19-44.
07. Mehedi, I.M., U.M. Al-Saggaf, R. Mansouri and M. Bettayeb, 2019. Stabilization of a double inverted rotary pendulum through fractional order integral control scheme. Int. J. Adv. Rob. Syst., Vol. 16, No. 4. 10.1177/1729881419846741
08. Wu, B., C. Liu, X. Song and X. Wang, 2015. Design and implementation of the inverted pendulum optimal controller based on hybrid genetic algorithm. Proceedings of the 2015 International Conference on Automation, Mechanical Control and Computational Engineering, April 25-26, 2015, Atlantis Press, Jinan, China, pp: 1480-1486.
09. Yigit, I., 2017. Model free sliding mode stabilizing control of a real rotary inverted pendulum. J. Vibr. Control, 23: 1645-1662.
10. Bilgic, H.H., M.A. Sen and M. Kalyoncu, 2016. Tuning of LQR controller for an experimental inverted pendulum system based on the bees algorithm. J. Vibroengineering, 18: 3684-3694.
11. Franco, E., A. Astolfi and Y.F.R. Baena, 2018. Robust balancing control of flexible inverted-pendulum systems. Mech. Mach. Theory, 130: 539-551.
12. Chatterjee, S. and S.K. Das, 2018. An analytical formula for optimal tuning of the state feedback controller gains for the cart-inverted pendulum system. IFAC-PapersOnLine, 51: 668-672.