



# Robust control based on adaptive neural network for Rotary inverted pendulum with oscillation compensation

Seyed Hassan Zabihifar<sup>1</sup> · Arkady Semenovich Yushchenko<sup>1</sup> · Hamed Navvabi<sup>1,2</sup>

Received: 23 May 2019 / Accepted: 24 February 2020 / Published online: 6 March 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

A new stable adaptive neural network (ANN) control scheme for the Furuta pendulum, as a two-degree-of-freedom underactuated nonlinear system, is proposed in this paper. This approach aims to address the control problem of the Furuta pendulum in the steady state and also in the presence of external disturbances. The adaptive classical control laws such as e-modification present some limitations in particular when oscillations are presented in the input. To avoid this problem, two ANNs are implemented using filtered tracking error in the control loop. The first one is a single hidden layer network, used to approximate the equivalent control online, and the second is the feed-forward network, used to minimize the oscillations. The goal of the control is to bring the pendulum close to the upright position in the presence of the various uncertainties and being able to compensate oscillations and external disturbances. The main purpose of the second ANN is to minimize the chattering phenomenon and response time by finding the optimal control input signal, which also leads to the reduction of energy consumption. The learning algorithms of the two ANNs are obtained using the direct Lyapunov stability method. The simulation results are given to highlight the performances of the proposed control scheme.

**Keywords** Adaptive neural network · Oscillation compensation · Rotary inverted pendulum · Furuta

## 1 Introduction

Stabilization of a Furuta pendulum has been considered as an active research area to control the engineers' system. The rotary inverted pendulum is a popular test bed for the class of underactuated mechanical systems. Early researches on rotary inverted pendulum were motivated by the need to design controllers to balance the rockets during a

vertical take-off [1]. Nevertheless, the control algorithm developed for a rotating pendulum system can be easily extended for any other two-degree-of-freedom unstable underactuated system (e.g., Acrobot, pendubot, inertia wheel pendulum, cart-pole) [2].

The Furuta pendulum is a two-degree-of-freedom system with only one actuator. This is an inverted pendulum, classified as a nonlinear, nonminimum phase and underactuated system. The structure is composed of an arm, attached to a motor, rotating in the horizontal plane. At the end of the arm, a pendulum is attached with a free rotational movement in the vertical plane. The motion control of such systems becomes difficult because the control of the overall system should be achieved from the actuated joints to the nonactuated joints [3]. Moreover, the presence of extraneous disturbance in the system has made the control design more complicated.

The rotary inverted pendulum (Fig. 1), which was first introduced by Furuta et al. [4], contains well-known underactuated dynamics, and many reports about its stabilization can be found. Most of the controls of the rotary inverted pendulum fall into one of the several categories. For example, some have considered the problem of

---

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s00521-020-04821-x>) contains supplementary material, which is available to authorized users.

---

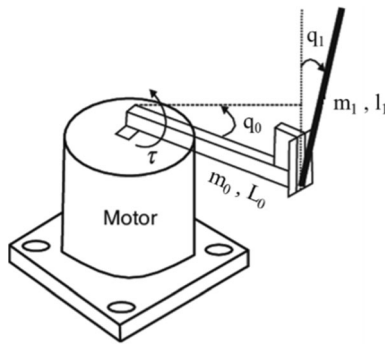
✉ Seyed Hassan Zabihifar  
zabihifar@student.bmstu.ru

Arkady Semenovich Yushchenko  
yusch@bmstu.ru

Hamed Navvabi  
navvabi\_hamed@mecheng.iust.ac.ir

<sup>1</sup> School of Mechatronic and Robotic, Bauman Moscow State Technical University (BMSTU), Moscow, Russia

<sup>2</sup> School of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran



**Fig. 1** Furuta pendulum system

stabilizing the pendulum around the unstable vertical position [5–9]. Some swung the pendulum from its hanging position to its upright vertical position [10–13]. Some tried to create oscillations around its unstable vertical position [14, 15], and some tried to track the trajectory of the arm, while the pendulum is in the upright vertical position [3, 16]. Therefore, three classical control objectives have been discussed in the literature as follows: (1) swing-up; (2) stabilization; and (3) trajectory tracking.

This study aims to consider the control problem of swinging up the pendulum to its upright vertical position and to stabilize the pendulum around that point. The adaptive neural network has been applied to achieve these control goals due to their approximation property, and the sliding surface brings us robustness property. Eliminating the chattering phenomenon in steady-state mode and optimization of the control input signal are the main purpose of this novel control scheme.

Regarding the stabilization problem, neural networks have been used in various pendulum-type systems [17, 18]. The robustness property of neural networks has been demonstrated using either real-time experiments or numerical simulations. In [19], adaptive neural network control for unknown nonlinear systems was proposed. The approach was applied to a car pendulum that provided tracking of the pendulum without considering the cart's position. In [20], a dynamic Takagi–Sugeno–Kang-type, radial basis function-based neural–fuzzy system was proposed for online estimation of an ideal controller. Although the controller can solve a tracking problem, it was applied to the stabilization of the car–pole system; however, similar to [19], the boundedness of the cart position was not shown. In [21], a method based on neural networks with output feedback control was applied to address the tracking problem for a spherical inverted pendulum.

Several robust controllers were proposed for dealing with uncertainties and disturbances in the Furuta system. Yu et al. [22] proposed a robust controller to stabilize the Furuta pendulum under bounded perturbation. Khanesar et al. [23] used a fuzzy sliding controller to drive a rotary inverted pendulum

to the vertical position subject to bound uncertainties and disturbances. Park et al. [17] presented a swing-up and stabilization control with coupled sliding mode control. In [24], an adaptive RBF network-based NN controller has been proposed, which combines with the SMC robust compensator to control problems for two-link robot manipulator; in [5], an adaptive controller has been proposed to balance a rotary inverted pendulum with time-varying uncertainties; and finally in [3], the tracking control of pendulum-type systems has been discussed using neural networks.

In these approaches, oscillations in outputs or control input signals have not been eliminated completely, and strong fluctuations exist in control input signals when putting the system in an unstable situation in addition to the high energy consumption. Some have steady-state errors or more fluctuations when the external disturbance has been disappeared.

In these approaches, oscillations in outputs or control input signals have not been eliminated completely, and strong fluctuations exist in control input signals when putting the system in an unstable situation in addition to the high energy consumption. Some have steady-state errors or more fluctuations when the external disturbance has been disappeared.

In this paper, the control input has been smoothed in a steady-state mode using a new approach, and the oscillation of the system has been eliminated by using filtered tracking error and a new scheme of adaptive neural network algorithm. Removing discontinuity terms in control law and using a dual neural network make a fast adaptation while reasonable control input is needed. Robustness of the system has been demonstrated by applying external disturbance to drive out the system from the equilibrium position. The results show that the novel adaptive neural network controller performs well, while the dynamic model of the system is not needed to design the controller. Finally, the results have been compared with previous works to highlight the controller improvement with the new presented scheme. Finally, all of these results are verified by ADAMS model (software in the loop) and the last section gives the conclusions.

The main contributions are as follows:

- Introducing a new filtered tracking error to stabilize the inverted pendulum in the vertical position and keep the arm stable with the zero velocity.
- Introducing a new control scheme for an underactuated system using dual adaptive neural network.
- Proposing a new weight adaptation laws based on e-modification technique for the corrective neural network and introducing a pseudo-sigmoid activation function in the second neural network for oscillation compensation.

## 2 System dynamics

The dynamic model of the Furuta pendulum in Euler–Lagrange form [25, 26] can be written as:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G_m(q) = U \tag{1}$$

where  $q = [q_0 \ q_1]^T \in \mathbb{R}^2$  is a vector of joint positions,  $M(q) \in \mathbb{R}^{2 \times 2}$  is the symmetric positive definite inertia matrix,  $c(q, \dot{q})\dot{q} \in \mathbb{R}^2$  is the vector of centripetal and Coriolis torques,  $G_m(q) \in \mathbb{R}^2$  is the vector of gravitational torques, and  $U = [u \ 0]^T \in \mathbb{R}^2$  is the vector of input torques, with  $u \in \mathbb{R}$  being the torque applied to the arm. In particular, the model of the Furuta pendulum has the following components:

$$q = \begin{bmatrix} q_0 \\ q_1 \end{bmatrix}, \quad M(q) = \begin{bmatrix} I_0 + m_1(L_0^2 + l_1^2 \sin^2 q_1) & m_1 l_1 L_0 \cos q_1 \\ m_1 l_1 L_0 \cos q_1 & J_1 + m_1 l_1^2 \end{bmatrix}$$

$$C(q, \dot{q}) = \begin{bmatrix} \frac{1}{2} m_1 l_1^2 \sin(2q_1) \dot{q}_1 & -m_1 l_1 L_0 \sin q_1 \dot{q}_1 + \frac{1}{2} m_1 l_1^2 \sin(2q_1) \dot{q}_0 \\ -\frac{1}{2} m_1 l_1^2 \sin(2q_1) \dot{q}_0 & 0 \end{bmatrix}$$

$$G_m(q) = \begin{bmatrix} 0 \\ -m_1 g l_1 \sin q_1 \end{bmatrix}, \quad U = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

The coordinate system and notations are described in Fig. 1. We will assume that friction is negligible.

- $I_0$  Inertia of the arm
- $L_0$  Total length of the arm
- $m_1$  Mass of the pendulum
- $l_1$  Distance to the center of gravity of the pendulum
- $J_1$  Inertia of the pendulum around its center of gravity
- $q_0$  Rotational angle of the arm
- $q_1$  Rotational angle of the pendulum
- $u$  Input torque applied on the arm
- $g$  The gravity

### 2.1 Problem formulation

The controller is required to serve a twofold control objective. The first objective is to stabilize the pendulum in its upright position at the origin from an initial condition in the upper half plane (i.e.,  $q_1 \in (-\frac{\pi}{2}, \frac{\pi}{2})$ ). The second objective of the controller is to ensure the proper orientation control of the arm ( $q_0$ ). In addition, the controller must possess adequate disturbance rejection ability to offer satisfactory control performance in an uncertain environment. The rotary inverted pendulum has dynamics from (1). Define the tracking error  $e(t)$  by

$$\begin{aligned} e_0 &= q_{0_d} - q_0 \\ e_1 &= q_{1_d} - q_1 \end{aligned} \tag{2}$$

where the desired arm position  $q_d(t)$  is twice differentiable and bounded for all time  $t \geq 0$  in the sense

$$\|q_d(t)\|, \|\dot{q}_d(t)\|, \|\ddot{q}_d(t)\| \leq \zeta \tag{3}$$

where  $\zeta$  is a positive constant.

The Furuta pendulum model in (1) can be written in the following form [3]:

$$\begin{aligned} \frac{d}{dt} q_0 &= \dot{q}_0 \\ \frac{d}{dt} \dot{q}_0 &= f_0 + g_0 u \\ \frac{d}{dt} q_1 &= \dot{q}_1 \\ \frac{d}{dt} \dot{q}_1 &= f_1 + g_1 u \end{aligned} \tag{4}$$

where

$$\begin{aligned} f_0 &= \frac{1}{\det M(q)} [M_{22} Z_1 - M_{12} Z_2] \\ f_1 &= \frac{1}{\det M(q)} [-M_{21} Z_1 + M_{11} Z_2] \\ Z_1 &= -C_{11} \dot{q}_0 - C_{12} \dot{q}_1 \\ Z_2 &= -C_{21} \dot{q}_0 - C_{22} \dot{q}_1 + m_1 g l_1 \sin \theta_1 \\ g_0 &= \frac{M_{22}}{\det M(q)} \\ g_1 &= \frac{M_{21}}{\det M(q)} \end{aligned} \tag{5}$$

with  $M_{ij}$  and  $C_{ij}$ , which are the elements of the inertia matrix  $M(q)$  and the Coriolis matrix  $C(q, \dot{q})$ , respectively. The system given by (4) can be written in terms of the tracking error (2) as follows:

$$\dot{e}_0 = \dot{q}_{0_d} - \dot{q}_0 \tag{6}$$

$$\ddot{e}_0 = \ddot{q}_{0_d} - f_0 - g_0 u \tag{7}$$

$$\dot{e}_1 = \dot{q}_{1_d} - \dot{q}_1 \tag{8}$$

$$\ddot{e}_1 = \ddot{q}_{1_d} - f_1 - g_1 u \tag{9}$$

which describe the open-loop system with  $\dot{q}_{1_d} = 0, \ddot{q}_{1_d} = 0$ . Here, it is important to introduce a proper function of the error to gain the control goal. Then, we propose an output function  $r(t) \in \mathbb{R}$  as a filtered tracking error given by

$$r = \dot{e}_0 + \dot{e}_1 + \lambda e_1 \tag{10}$$

where  $\lambda > 0$  is a positive definite design parameter. By computing its time derivative, one can obtain

$$\dot{r} = \ddot{e}_0 + \ddot{e}_1 + \lambda \dot{e}_1 \tag{11}$$

substituting (7) and (9) into (11) and simplifying

$$\begin{aligned} \dot{r} &= \ddot{q}_{d0} - \dot{f}_0 - g_0u - f_1 - g_1u - \lambda\dot{q}_1 \\ &= F - Gu \end{aligned} \tag{12}$$

where

$$\begin{aligned} F &= \ddot{q}_{d0} - \lambda\dot{q}_1 - \dot{f}_0 - f_1 \\ G &= (g_0 + g_1) \end{aligned} \tag{13}$$

Equation (12) can also be rewritten as:

$$\frac{\dot{r}}{G} = \frac{F}{G} - u \tag{14}$$

where the function  $G(q_1)$  is strictly positive for all  $|q_1| < \cos^{-1}(\frac{J_1+m_1l_1^2}{m_1l_1L_0})$  and is bounded and continuous such that

$$\frac{1}{2} \left| \frac{\dot{G}}{G^2} \right| \leq \mu \tag{15}$$

where  $\mu$  is a positive constant value [3].

A two-layer neural network can estimate any nonlinear, continuous and unknown function [27]. Now, according to the universal approximation property of NN, there is a two-layer NN such that:

$$\frac{F}{G} = f(x) = W^T \sigma(V^T x) + \epsilon \tag{16}$$

which  $V, W$  are the NN weights,  $\sigma$  is a sigmoid activation function,  $x$  is the input vector of the neural network and the approximation error  $\epsilon$  bounded on a compact set by

$$\|\epsilon\| < \epsilon_N$$

Now, let an NN estimate of  $f(x)$  be given by

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \tag{17}$$

where  $\hat{V}, \hat{W}$  are the matrix of input and output NN weights, respectively, which should be specified given the tuning algorithm. Note that  $\hat{V}, \hat{W}$  are estimates of the ideal weight values.

The problem is to define a control law, which can estimate the system dynamics and is robust enough to compensate disturbances while reaching the control goal. Oscillation compensation is another goal for the control system.

### 3 Controller structure

In this section, SHL networks will perform the approximation of the corresponding command. According to the universal approximation theory, the neural network of the SHL type can estimate any nonlinear, continuous, unknown function [27]. Due to the iterative nature of the neural

network’s training mechanism and due to the high order of complexity of the dynamic model, the neural network may take a relatively long time to converge, which may lead to unstable dynamics or unsatisfactory performance. Henceforth, a robustifying term  $F_r$  is introduced that is corresponding to a PD controller and injects damping into the system.

$$F_r = K_v r \tag{18}$$

with  $K_v$  being a positive constant gain. Therefore, a novel adaptive neural-network-based controller is proposed by the following expression:

$$U = U^* + U_{cor} \tag{19}$$

$$U^* = \hat{W}^T \sigma(\hat{V}^T x) + K_v r \tag{20}$$

where  $U^*$  is the equivalent controller, which contains SHL neural network and  $F_r$ . The  $U_{cor}$  is the ideal corrective control that will be presented in the next section.

The optimal weight matrices  $\hat{V}, \hat{W}$  are unknown, and it is necessary to estimate them by an adaptation mechanism so that the output feedback control law can be realized. The matrices  $\hat{W}$  and  $\hat{V}$  are the estimation of  $W$  and  $V$ , respectively. Assumption: On any compact subset of  $\mathfrak{R}^n$ , the ideal NN weights are bounded so that  $W_F \leq W_m, V_F \leq V_m$ . with  $W_m$  and  $V_m$  are unknown positive constants,  $\|\cdot\|_F$  is the Frobenius norm, and the weight deviations or weight estimation errors are defined as

$$\hat{V} = V - \hat{V} \quad \hat{W} = W - \hat{W}$$

Define the hidden layer output error for a given  $x$  as

$$\tilde{\sigma} = \sigma - \hat{\sigma} \equiv \sigma(V^T x) - \sigma(\hat{V}^T x) \tag{21}$$

The Taylor series expansion of  $\sigma(x)$  for a given  $x$  may be written as

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^2 \tag{22}$$

with

$$\sigma'(z) \equiv \left. \frac{d\sigma(z)}{dz} \right|_{z=z} \tag{23}$$

the Jacobian matrix. The  $O(z)^2$  denotes terms of order two. Denoting  $\hat{\sigma}' = \sigma'(\hat{V}^T x)$ , we have [28]

$$\tilde{\sigma} = \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^2 = \hat{\sigma}' \tilde{V}^T x + O(\tilde{V}^T x)^2 \tag{24}$$

E-modification technique usually used in robust adaptive control, which is applied for improving the robustness of the controller in the presence of the NN approximation error [29]. Hence, the weight adaptation laws based on e-modification technique has been provided by [28]:

$$\begin{aligned} \dot{\hat{W}} &= F_w(\hat{\sigma}r^T - k\|r\|\hat{W}) \\ \dot{\hat{V}} &= F_v(xr^T\hat{W}^T\hat{\sigma}' - k\|r\|\hat{V}) \end{aligned} \tag{25}$$

with  $F_w > 0$ ,  $F_v > 0$  are the adaptive gains, and  $k$  is a positive constant. Then, the tracking error  $r(t)$  approaches to zero with  $t$ , and the weight estimates  $\hat{V}$ ,  $\hat{W}$  are bounded.

### 3.1 Corrective control

According to the use of the filtered tracking error, some oscillations may appear in steady-state mode. Although in other works, a sign function has been used to correct the controller, the discontinuity makes some perturbations specifically when an external disturbance has been applied. The corrective control is designed to eliminate the chattering phenomenon, which is provided by the discontinuous terms; for that reason, we proposed a continuous function  $Z(\cdot)$  given by:

$$Z(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{26}$$

Therefore, the ideal expression of the corrective term will be:

$$U_{cor}^* = BZ(\alpha^T x^e) \tag{27}$$

where the gain  $B$  and  $\alpha = [\lambda_1 \ \lambda_2]^T$  present the ideal output layer and hidden layer weights, respectively,  $x^e = [e_1 \ \dot{e}_1]^T$  is the input vector, and  $Z(\cdot)$  is the activation function defined in Eq. (26). A second neural network, which is shown in Fig. 2, is added to the control scheme. The purpose here is estimating the ideal corrective term.

The output of this network is represented by:

$$U_{cor}^* = \hat{B}Z(\hat{\alpha}^T x^e) + \varepsilon_c \tag{28}$$

where  $\hat{B}$  and  $\hat{\alpha}$  are the estimation of  $B$  and  $\alpha$ , respectively, and  $\varepsilon_c$  is the reconstruction error. The weights  $B$  and  $\alpha$  are unknown, so it is judicious to find a way of adapting them. Therefore, the real estimate of the corrective term by the ANN presented in Fig. 2 has the following structure:

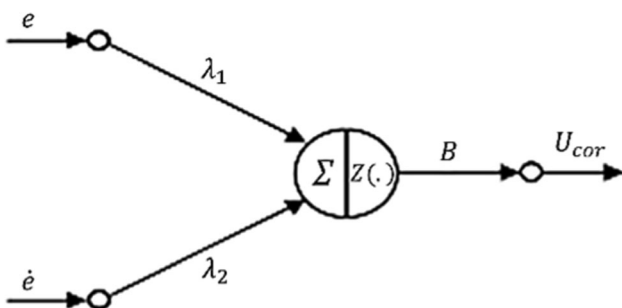


Fig. 2 Second network to estimate the correction term

$$U_{cor} = \hat{B}Z(\hat{\alpha}^T x^e) \tag{29}$$

The Taylor series expansion of the corrective term estimation error is given by:

$$\tilde{U}_{cor} = BZ - \hat{B}\hat{Z} = \tilde{B}\hat{Z} + B\tilde{Z} = \tilde{B}\hat{Z} + \hat{B}\hat{Z}'\tilde{\alpha}^T x^e + w_g \tag{30}$$

where  $\tilde{B} = B - \hat{B}$ ,  $\tilde{\alpha} = \alpha - \hat{\alpha}$  are the error parameters. The term  $w_g$  presents the approximation error given by:

$$w_g = \varepsilon_c + \tilde{B}\hat{Z}'\tilde{\alpha}^T x^e + \tilde{B}O(\tilde{\alpha}^T x^e)^2 \tag{31}$$

Assumption:  $\|w_g\| \leq \bar{w}_g$ ,  $\bar{w}_g$  is an unknown positive constant and  $|B| \leq B_m, \|\alpha\|_F \leq \alpha_m$  with  $\alpha_m$  and  $B_m$  are unknown positive constants. The proposed adaptation laws for adjusting weights are given by the following equations that will be proofed by the Lyapunov function in (49).

$$\begin{aligned} \dot{\hat{B}} &= F_B \hat{Z}(\alpha^T x^e) r_e \\ \dot{\hat{\alpha}} &= F_\alpha (x^e r_e \hat{B} \hat{Z}' + k_\alpha |r_e| (\bar{\alpha} - \hat{\alpha})) \end{aligned} \tag{32}$$

where  $r_e = \dot{e}_1 + \delta e_1$ ,  $F_B = F_B^T > 0$ ,  $F_\alpha = F_\alpha^T > 0$ ,  $k_\alpha > 0$ . The vector  $\bar{\alpha}$  is selected as follows:

$$\bar{\alpha} = \left[ \frac{p}{|e_1| + \varepsilon} \quad 1 \right]^T, \quad p, \varepsilon > 0 \tag{33}$$

where  $\bar{\alpha} \leq \alpha_n$ ,  $\alpha_n$  is a positive constant. The structure of the controller is shown in Fig. 3.

### 3.2 Error system dynamics

The control input from (19), (20), and (29) is defined as

$$U = \hat{W}^T \sigma(\hat{V}^T x) + K_v r + \hat{B}Z(\hat{\alpha}^T x^e) \tag{34}$$

Using the control law (34), the closed-loop filtered error dynamics becomes

$$\frac{\dot{r}}{G} = -K_v r + W^T \sigma(V^T x) - \hat{W}^T \sigma(\hat{V}^T x) - \hat{B}Z(\hat{\alpha}^T x^e) + \varepsilon \tag{35}$$

Adding and subtracting  $W^T \hat{\sigma}, B\hat{Z}$  yields

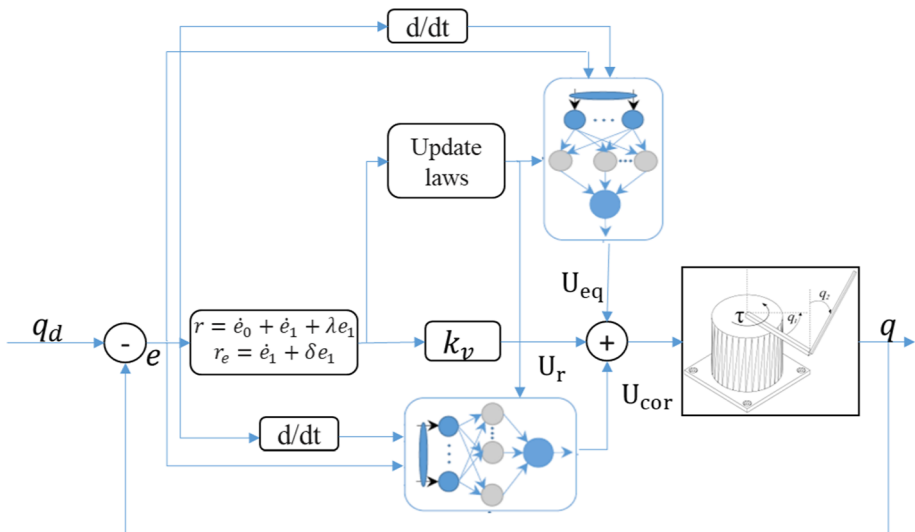
$$\frac{\dot{r}}{G} = -K_v r + \tilde{W}^T \hat{\sigma} + W^T \hat{\sigma} + \hat{B}\hat{Z} - B\hat{Z} + \varepsilon \tag{36}$$

Moreover, now adding and subtracting  $\hat{W}^T \tilde{\sigma}, B\tilde{Z}$  yields

$$\frac{\dot{r}}{G} = -K_v r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} + \tilde{W}^T \tilde{\sigma} + \hat{B}\hat{Z} + B\tilde{Z} - BZ + \varepsilon \tag{37}$$

The key step here is the use of the Taylor series approximation for  $\tilde{\sigma}, \tilde{Z}$ , according to which the closed-loop error system is

Fig. 3 Control system structure



$$\frac{\dot{r}}{G} = -K_v r + \tilde{W}^T \hat{\sigma} + \tilde{W}^T \hat{\sigma}' \tilde{V}^T x + \tilde{B} \tilde{Z} + \tilde{B} \tilde{Z}' \tilde{\alpha} x^e + w_1 \tag{38}$$

where the disturbance terms are

$$w_1 = \tilde{W}^T \hat{\sigma}' \tilde{V}^T x - BZ + w_g + \tilde{W}^T O(\tilde{V}^T x)^2 + \varepsilon \tag{39}$$

As seen, the convergence of “r” to zero implies convergence of the tracking error and its derivative to zero. So, the objective of control is summarized in the synthesis of a control law that allows the convergence of the filtered error to zero [28]. Let the desired trajectory  $q_d(t)$  be bounded by  $q_b$ . Assume the disturbance term  $w_1$  in (38) equals zero. The stability of closed-loop system has been proved by defining the Lyapunov function candidate  $L$  as below. The proof is illustrated in “Appendix.”

$$L = \frac{1}{2} \frac{r^2}{G} + \frac{1}{2} \text{tr}\{\tilde{W}^T F_w^{-1} \tilde{W}\} + \frac{1}{2} \text{tr}\{\tilde{V}^T F_v^{-1} \tilde{V}\} + \frac{1}{2} \text{tr}\{\tilde{B}^T F_B^{-1} \tilde{B}\} + \frac{1}{2} \text{tr}\{\tilde{\alpha}^T F_\alpha^{-1} \tilde{\alpha}\} \tag{40}$$

### 4 Simulation studies and performance comparison

The simulation results are introduced in order to make it possible to see the operation of the proposed adaptive neuro-controller and control input signals. The results of the swing-up and disturbance rejection tests are introduced. A simulation comparison wherein three controllers are compared to the new controller is performed. The model

parameters are briefly described, and the controllers used in the comparison are introduced.

#### 4.1 Controllers used in the comparison

A simulation study has been performed to assess the performance of the new controller (34). Specifically, a linear controller and the adaptive neural network schemes in [16, 30] were implemented in real time.

The schemes in [3, 16, 30] have an adaptive neural network component and a linear PD or PID component. In addition, in [3] there is a nonlinear term that eliminates the adaptation error introduced by the neural network, but that discontinued term introduces some fluctuations, specifically when the disturbance happens. In the proposed controller, the second adaptive neural network has been used to eliminate the oscillation and the adaptation error. Also in disturbance investigation, the external disturbance should be applied on the pendulum to observe the ability of the controller to keep it stable, while in some of the previous works, it has been applied as a torque summation in the control input of the first link. In [3], the initial angle of the pendulum has been defined zero, while here, we propose a nonzero initial angle to highlight the ability of the proposed scheme to swing up and also to confirm the robustness against external disturbances. It is important to observe that the proposed adaptation laws (25) and (32) of the new scheme (34) are derived such that the time derivative of the Lyapunov function  $L$  in (49) is negative, and some of the other schemes use the back-propagation adaptation without any motivation given by the closed-loop system analysis.

The adaptive neural network controller proposed by Moreno [3] is given by

$$\tau = -\tilde{W}^T \sigma(\tilde{V}^T x) - k_p y - \delta \text{sign}(y) \tag{41}$$

where the constants  $k_p$  and  $\delta$  are positive,  $\hat{V}$  is the matrix of estimated input weights, and  $\hat{W}$  is a vector of the estimated output weights. Output function  $y(t)$  is given by

$$y = \dot{e}_1 + \Delta_1 e_1 + \dot{e}_2 + \Delta_2 e_2 \tag{42}$$

where  $\Delta_1 = 3, \Delta_2 = 8$  are positive constants and  $k_p = 1.05, \delta = 0.035$ . The number of neurons in the hidden layer was  $L = 10$ . The adaptation laws were implemented with  $\alpha = 1, N = 1.05$ , and  $R = 8.53$ . The weights of the neural network were initialized zero. It is worth mentioning that the presented approach by Moreno has been compared to the PID algorithm and the results show the superiority of Moreno’s approach to PID algorithm. Therefore, the PID has not been discussed in this paper.

In the tests, we also consider the adaptive neural network controller proposed by Chaoui and Sicard [16].

$$\tau = \tau_{NN} + k_D s \tag{43}$$

where  $\tau_{NN}$  is the output of a two-layer neural network, six neurons in the hidden layer and one neuron in the output layer; that its, input and output weights are obtained by the back-propagation algorithm, which minimizes the signal  $s$  given by

$$s = [1 - \lambda][\dot{e}_1 + \psi_1 e_1] + \lambda[\dot{e}_2 + \psi_2 e_2] \tag{44}$$

with  $0 < \lambda < 1, \psi_1, \psi_2 > 0$ , and  $k_D > 0$ . Specifically, (43) was implemented with  $\lambda = 0.5, \psi_1 = 20, \psi_2 = 30$ , and  $k_D = 1.757$ .

See the original works [3, 16] for further details on the controllers (41) and (43), respectively.

### 4.2 Simulation

The simulation of the Furuta pendulum has been conducted on a MATLAB®/Simulink with the following initial conditions: ( $q_0 = 0, q_1 = 30^\circ$ ). The actual values of the system parameters are presented in Table 1. For the proposed controller, the number of neurons in the hidden layer is 7, and initial values for  $W$  and  $V$  have been defined zero. Determining the number of hidden layer neurons required for the best approximation is an open problem for general fully connected two-layer NN [28]. Therefore, the number of neurons in the hidden layer is defined by the trial-and-error approach, and it can be selected from 3 to 11 for a good performance. Changing this parameter (Figs. 10, 11)

**Table 1** System parameters

$m_0 = 990$ gr	$l_0 = 55$ cm	$q_0 = 0$
$m_1 = 225$ gr	$l_1 = 25$ cm	$q_1 = 30^\circ$

in this area causes little changes in the results and can be easily modified. The initial adaptive coefficients in (25) are assumed to be  $F_w = 100, F_v = 50, k = 1$  and in (32) is assumed to be  $F_B = 5, F_\alpha = 2, k_\alpha = 1$ . We suppose that  $\lambda = 10, K_v = 1, \delta = 2$  and initial values for  $B, \lambda_1, \lambda_2$  are zero.

To stabilize the system, the arm speed should also be controlled, and by applying the zero as the desired state for it, a stable system has been achieved. Therefore, the state variables are considered as follows:

$$\begin{aligned} X &= [e_1 \cdot \dot{e}_1 \cdot e_{1\text{ delay}} \cdot \dot{e}_0 \cdot \ddot{e}_0] \\ e_0 &= q_0^{\text{desired}} - q_0 \\ e_1 &= q_1^{\text{desired}} - q_1 \end{aligned} \tag{45}$$

The reason for introducing  $e_{1\text{ delay}}$  is that since the neural network controller itself is a nonmemory device, some delayed signals must be introduced in order to control the output to depend not only on the current input (error in our case) but also on past inputs. Under some circumstances, by properly introducing time delays into the control channel, the control performance of some practical systems can be improved [31]. As discussed in [32], using mixed current and delayed states can significantly reduce both internal oscillations of the offshore platform and required control force. In this paper, we consider only one simple delayed signal. However, in principle, multiple delayed signals can be introduced.

According to Eq. (25), its parameters are calculated as follows.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{46}$$

Then,  $\sigma'(x)$  is calculated as follows [28, 33, 34]:

$$\sigma'(x) = \sigma(x)\sigma(-x) \tag{47}$$

### 4.3 Results

The following shows the obtained results. For simplicity, the proposed controller in (34) is labeled as “new,” the neural network controller proposed by [3] in (41) is labeled as “Moreno,” and the neural network controller introduced by [16] in (43) is labeled as “Sicard.” The results of the three implemented controllers to reach the inverted pendulum from 30 degrees to vertical position (zero degree) without external disturbances are shown in Figs. 4 and 5. The maximum absolute value of the pendulum position error and the root mean square (RMS) value of the error are compared in Table 2. The settling time has been defined when the signal crosses into and remains in the 2%-tolerance region around the state level. In maximum error quantification, the first-time interval is considered after 5 s

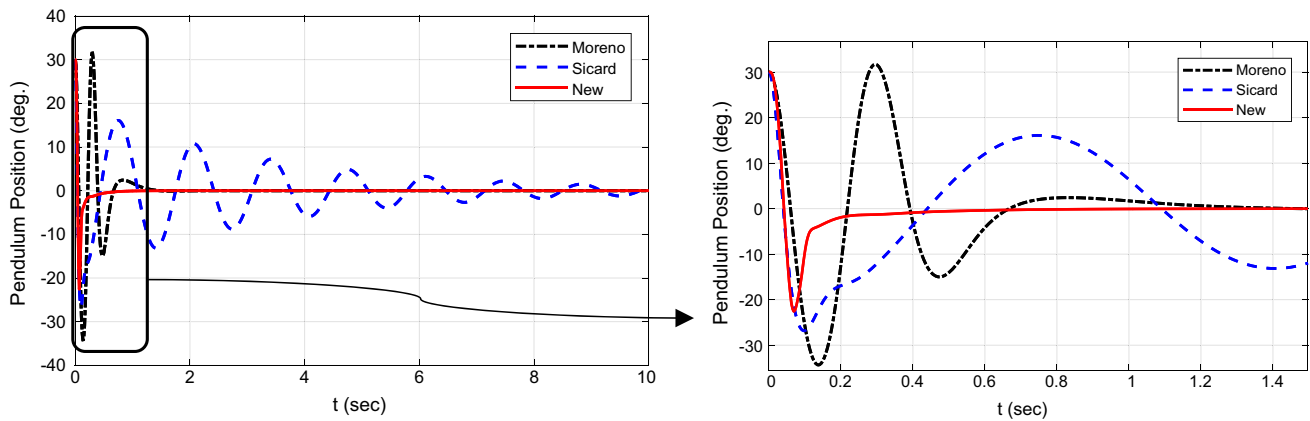


Fig. 4 Angle of Pendulum position in comparison with other implemented controllers and the magnified part of it

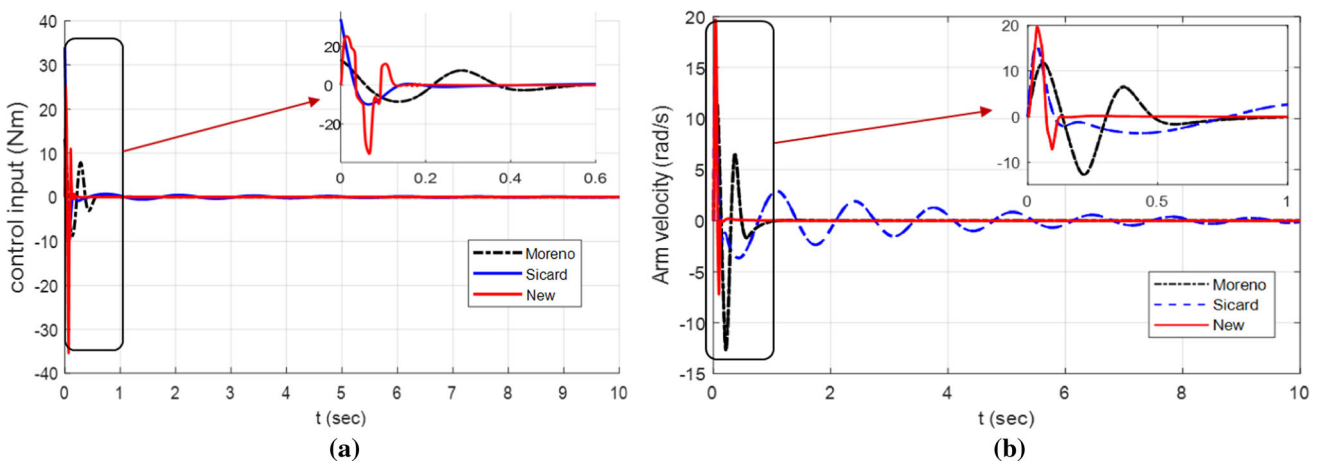


Fig. 5 Comparison of three implemented controllers **a** the control input and **b** the angular velocity of the arm

Table 2 Quantification of error

	Sicard	Moreno	New
Max $\{ e_1(t) \}$ ( $^\circ$ ), $5 < t < 10$	3.9842	$3.1010e - 5$	<b><math>7.5377e - 9</math></b>
Max $\{ e_1(t) \}$ ( $^\circ$ ), $2 < t < 10$	10.7567	0.0186	<b>0.0013</b>
RMS $\{e_1(t)\}$ ( $^\circ$ )	6.1463	5.0352	<b>1.9126</b>
Settling time $q_1(t)$ (2%) (s)	11.019	1.214	<b>0.485</b>

Bold values show the best results in each row of the table

when the pendulum is in a stable position, and the maximum of the error has been highlighted. The second time interval ( $2 < t < 10$ ) shows the ability of the proposed approach to reach a stable position in a shorter time.

Figure 4 shows that, in 0.2 s, the controller can adapt to parameters with a completely unknown model for the controller, and control the underactuated system well. Also, the control input is within the acceptable range (Fig. 5a). The velocity of the arm tends to be zero while the oscillation has not been produced (Fig. 5b). The first second of

the diagrams has been magnified to highlight the performance of the proposed controller in comparison with others. The fast adaptation and less fluctuation in pendulum position and arm velocity are the advantages of the new controller. Table 2 presents a quantification of the results obtained during the tests, therein showing a comparison of the maximum absolute value of the error  $e_1(t)$ , the RMS value of the error  $e_1(t)$ , for each algorithm. A better performance is obtained with the new controller because the index RMS $\{e(t)\}$ , the settling time, and the maximum absolute of the error are the lowest.

To check the robustness of the controller, an external disturbance as a torque is applied to the pendulum for 0.1 s with  $F_d = 0.5$  NM and the controller performance are evaluated. The other variables with time are displayed in Figs. 6 and 7. Quantification of the error with disturbance has been evaluated in Table 3.

As seen in Figs. 6 and 7, the disturbance applied within 0.1 s changed the pendulum angle significantly from the steady state. However, the proposed controller is robust enough to keep the states in the neighborhoods of the



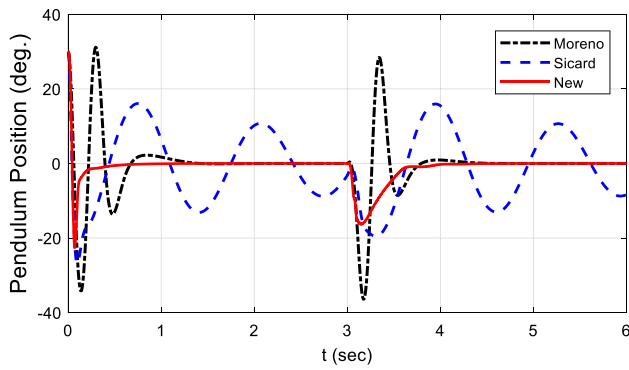


Fig. 6 Pendulum position with external disturbance

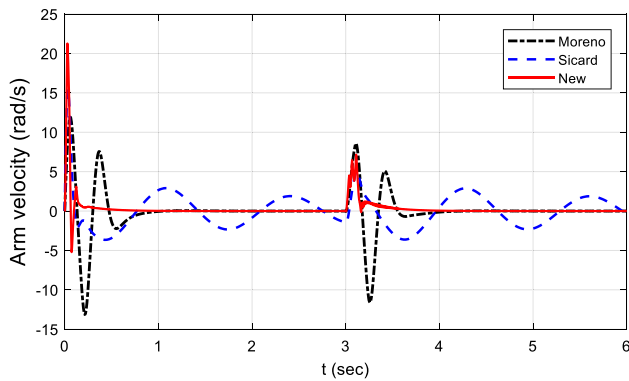


Fig. 7 The arm velocity with the external disturbance

Table 3 Quantification of error with external disturbance

	Sicard	Moreno	New
RMS error (deg.)	10.3346	8.5528	<b>3.8937</b>

Bold value shows the best result

equilibrium points and to return them to the desired values immediately when the disturbance disappears. The applied torque is a step function with positive value, which makes a positive deviation in pendulum’s angle if there is no controller. But here, because of fast adaptation of the controller parameters, the movement of the arm prevented the deviation of the pendulum in that direction, and after a very small movement in positive direction, the controller makes an input torque and pendulum deviates in minus direction and then tends to be in the vertical stable position. The results show that the controller tries to do does not produce a big error like other approaches and the settling time after disturbance highlights the controller adaptability. The RMS values of the errors in Table 3 show the ability of the proposed controller in disturbance rejection in comparison with other approaches. The absence of extreme fluctuations in the results shows good performance of the controller in an underactuated system.

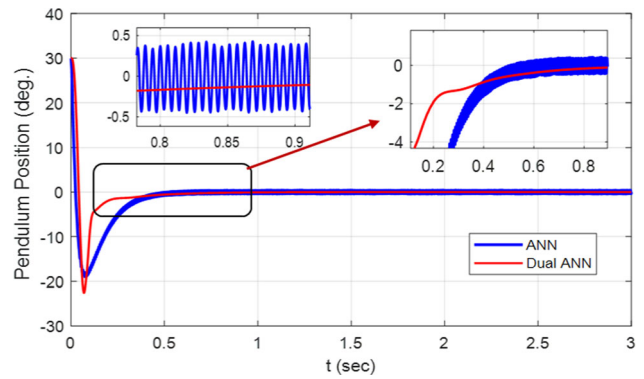


Fig. 8 The angle of pendulum position. ANN controller in comparison with proposed ANN controller in which oscillation is compensated

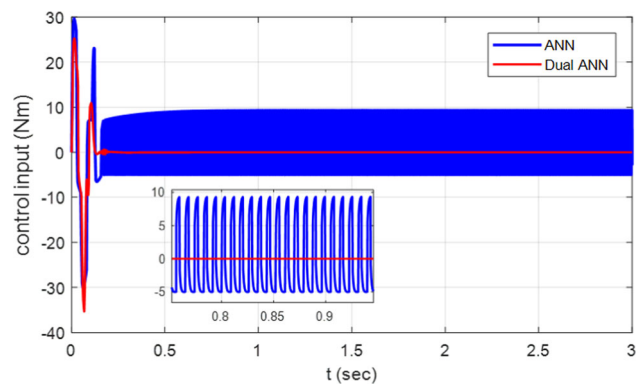


Fig. 9 Control input. ANN controller and proposed ANN controller in which oscillation is compensated

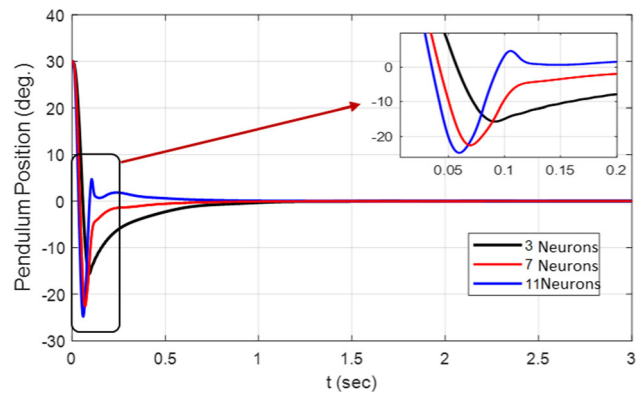


Fig. 10 Angle of pendulum position in a number of hidden neuron analysis

### 4.4 Corrective control analysis

In the following section, adding  $U_{cor}$ , which is the second part of the controller in (19), has been investigated. One of the advantages of the presented approach is using a dual neural network to eliminate oscillations that are emerged by a discontinuity of the sign function used in other

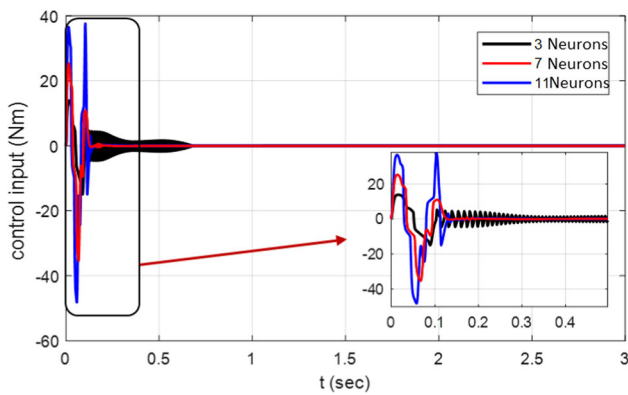


Fig. 11 Control input in a number of hidden neuron analysis

approaches like [3, 35]. To show the effect of the  $U_{cor}$  in oscillation compensation, the results with  $U_{cor}$  and without that are shown in Figs. 8 and 9. Actually, the controller  $U_2$  that is compared to (19) is defined by:

$$U_2 = \hat{W}^T \sigma(\hat{V}^T x) + K_v r + \text{sign}(r) \tag{48}$$

Figures 8 and 9 show the performance of the proposed algorithm in comparison with the ANN controller without  $U_{cor}$ . The proposed control scheme has compensated the oscillation caused by uncertainties and discontinuity. Although by multiplying a coefficient to the sign function,

the pendulum position oscillation can decrease (not eliminated at all), but the control input value increases so that it becomes unreasonable. Therefore, the figures suggest that the performance of the proposed algorithm is clearly better than the performance of the ANN controller formulated in (44).

### 4.5 Number of hidden neuron analysis

The number of hidden neurons in the neural network was studied in the following. As it has been referred to before, the number of neurons in the hidden layer is defined by the trial-and-error approach. In Figs. 10 and 11, the results have been analyzed when the number of hidden neurons in the hidden layer is changing from 3 to 11. The three sets of neurons have been introduced to show the efficiency of each case in pendulum position and control input values.

As seen in Figs. 10 and 11, three variants (3 neurons, 7 neurons, and 11 neurons) of choosing a number of hidden neurons have been analyzed. It is clearly shown that the 7 neurons in the hidden layer make better performance in position error and also in control input values.

The controller is able to drive the Furuta pendulum system toward its desired equilibrium points (i.e.,  $X = 0$ ) in an efficient manner. Simulation results clearly show the

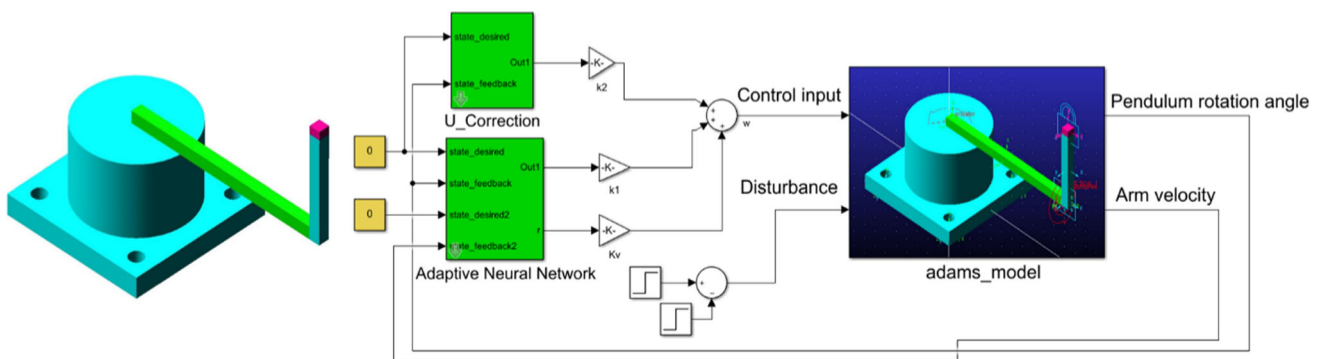


Fig. 12 ANN control of the system in ADAMS and Simulink (software in the loop)

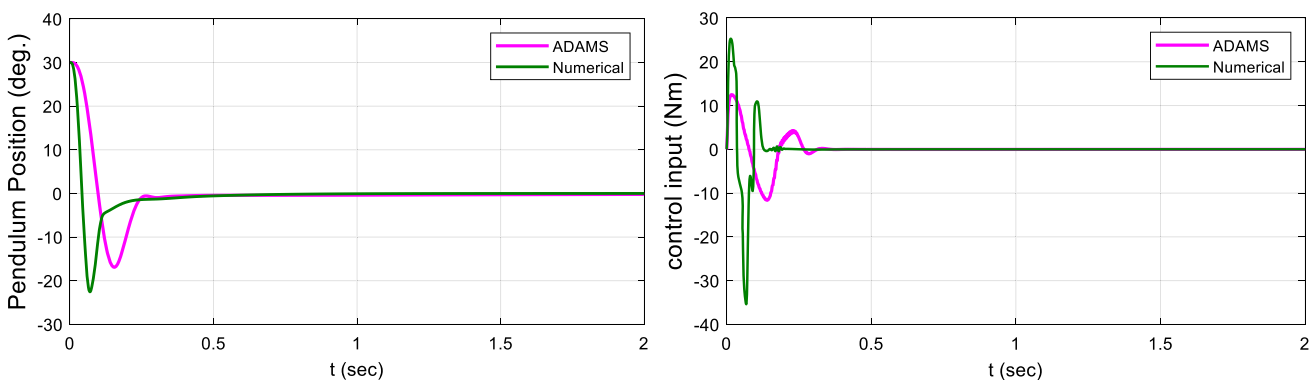


Fig. 13 Comparison of results for ADAMS and numerical model

stabilizing ability of the controller. Thus, the algorithm can be used for either two-degree-of-freedom systems with a slight design modification.

### 5 ADAMS simulation

In order to verify numerical results, dynamic model of the Furuta is simulated in the commercial software, ADAMS/View package of MSC software. Robot Furuta with all of the details, as like as real working condition of the robot, is modeled in ADAMS, which is shown in Fig. 12. Coulomb friction effect, Stiction and Sliding, are considered in all of the joints. Coefficients of static and dynamic friction are set to 0.5 and 0.3, respectively. All the robot dimensions and inertia parameters are assumed the same as in Table 1. The absolute velocity threshold for the transition from dynamic friction to static friction is assumed to be 0.1 mm/sec. The effect of viscous friction using a damper on the joints is considered. Damping coefficient is chosen 0.2, so that the software model performs the same as the experimental setup.

By using “Controls” plugin in ADAMS/View, dynamic model of the system is exported to MATLAB/Simulink environment (Fig. 12) and the proposed controller is implemented on the robot to evaluate the results. The

controller parameters are the same as Sect. 4.2. The results of the implemented controller on ADAMS model without external disturbance have been evaluated. To compare the ADAMS model with the numerical model, the results are shown in Fig. 13.

As seen in Fig. 13, both numerical and ADAMS models are almost the same and some small differences exist because of the friction, which is modeled perfectly in ADAMS model. The quantification of error for 10 s simulation has been presented in Table 4.

An external disturbance as a torque is applied to the pendulum for 0.1 s with  $F_d = 0.5$  NM (Fig. 14) as like as numerical model in Fig. 6.

As it is shown in Fig. 13, because of the friction existence in the ADAMS model, deviation of the angular position of the pendulum is not too much in comparison with numerical model. The results show that the proposed approach is working on numerical and simulated physical models as well and the performance is comparable with previous works.

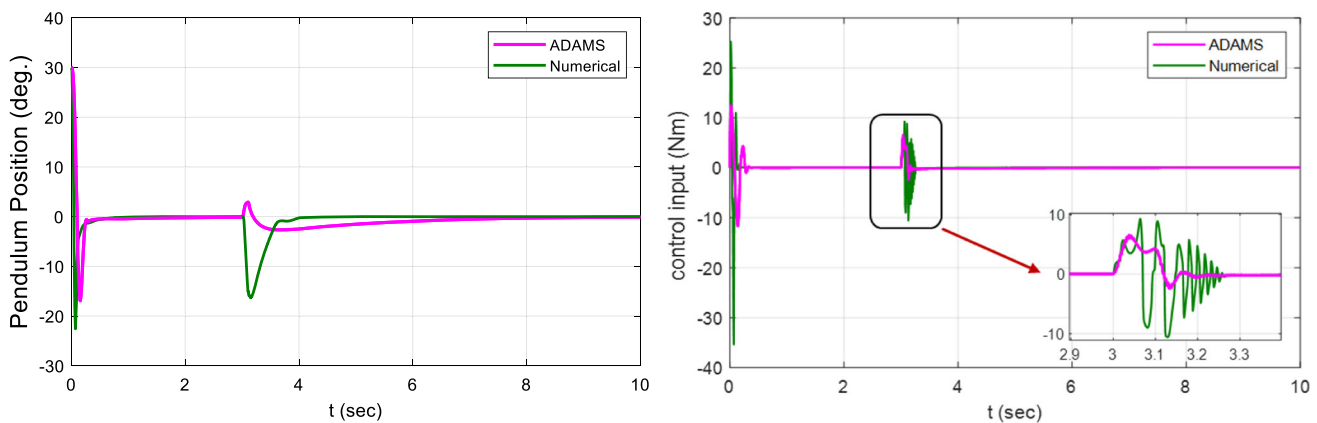
### 6 Conclusion

This paper showed the development and application of an adaptive neural network control scheme to drive the Furuta pendulum system from an initial condition toward its upright vertical position and to stabilize the system at that point. The  $U_{cor}$  has been introduced to compensate the oscillation of the pendulum when it reaches the steady-state point. The controller was derived from the universal approximation property of neural networks, and weight adaptation laws were designed. The simulation results clearly indicate the effectiveness of the proposed control law in an uncertain nonlinear underactuated system. The results have been compared with previous works to high-

**Table 4** Quantification of error for numerical and ADAMS model

	ADAMS	Numerical
Max $\{ e_1(t) \}$ (°), $5 < t < 10$	0.037	<b>7.5377e - 9</b>
Max $\{ e_1(t) \}$ (°), $2 < t < 10$	0.2191	<b>0.0013</b>
RMS $\{e_1(t)\}$ (°)	2.6178	<b>1.9126</b>
Settling time $q_1(t)$ (2%) (s)	<b>0.38</b>	0.485

Bold values show the best results in each row of the table



**Fig. 14** Pendulum position with external disturbance for ADAMS and numerical model

light the performance of the proposed approach. The controller simultaneously stabilizes the arm’s orientation angle and angular position of the pendulum. The controller is able to provide robust, nonfluctuation performance in the presence of parametric uncertainty and extraneous disturbances. In addition, the proposed algorithm can be easily extended for any other two-degree-of-freedom underactuated system. Numerical results are verified with software in the loop simulation of the robot in ADAMS software. Authors are planning to apply other activation functions and different ordering of the adaptive neural network to improve the performance and to apply the controller in other complex mechanisms for future works.

### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## Appendix

### Proof of the closed-loop system

Define the Lyapunov function candidate

$$L = \frac{1}{2}r^2 + \frac{1}{2}\text{tr}\{\tilde{W}^T F_w^{-1} \tilde{W}\} + \frac{1}{2}\text{tr}\{\tilde{V}^T F_v^{-1} \tilde{V}\} + \frac{1}{2}\text{tr}\{\tilde{B}^T F_B^{-1} \tilde{B}\} + \frac{1}{2}\text{tr}\{\tilde{\alpha}^T F_\alpha^{-1} \tilde{\alpha}\} \tag{49}$$

where  $L$  is the proposed Lyapunov function. Differentiating yields

$$\dot{L} = r \frac{\dot{r}}{G} - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{tr}\{\tilde{W}^T F_w^{-1} \dot{\tilde{W}}\} + \text{tr}\{\tilde{V}^T F_v^{-1} \dot{\tilde{V}}\} + \text{tr}\{\tilde{B}^T F_B^{-1} \dot{\tilde{B}}\} + \text{tr}\{\tilde{\alpha}^T F_\alpha^{-1} \dot{\tilde{\alpha}}\} \tag{50}$$

Whence substitution from (38) (with  $w_1 = 0$ ) yields

$$\begin{aligned} \dot{L} = & -K_v r^2 - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{tr}\{\tilde{W}^T (F_w^{-1} \dot{\tilde{W}} + \hat{\sigma} r^T)\} \\ & + \text{tr}\{\tilde{V}^T (F_v^{-1} \dot{\tilde{V}} + x r^T \tilde{W}^T \hat{\sigma}')\} + \text{tr}\{\tilde{B}^T (F_B^{-1} \dot{\tilde{B}} - r^T \dot{Z})\} \\ & + \text{tr}\{\tilde{\alpha}^T (F_\alpha^{-1} \dot{\tilde{\alpha}} + x^e r^T \hat{B} \dot{Z})\} \end{aligned} \tag{51}$$

Since  $\hat{W} = W - \tilde{W}$ , the  $W$  is constant, so  $d\frac{\hat{W}}{dt} = -d\tilde{W}/dt$ , as for  $V$ ,  $B$  and  $\alpha$ , the tuning rules from (32) and (25) yield.

$$\begin{aligned} \dot{L} = & -K_v r^2 - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{tr}\{\tilde{W}^T (\text{kr}\tilde{W})\} + \text{tr}\{\tilde{V}^T (\text{kr}\tilde{V})\} \\ & + \text{tr}\{\tilde{B}^T \dot{Z}(r - r_e)\} + \text{tr}\{\tilde{\alpha}^T (x^e \hat{B} \dot{Z}(r - r_e) - k_\alpha |r_e| (\bar{\alpha} - \hat{\alpha}))\} \end{aligned} \tag{52}$$

$$\begin{aligned} \dot{L} = & -K_v r^2 - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{krtr}\{\tilde{W}^T (W - \tilde{W})\} + \text{krtr}\{\tilde{V}^T (V - \tilde{V})\} \\ & + \text{tr}\{\tilde{B}^T \dot{Z}(r - r_e)\} + \text{tr}\{\tilde{\alpha}^T (x^e \hat{B} \dot{Z}(r - r_e) - k_\alpha |r_e| (\bar{\alpha} - \hat{\alpha}))\} \end{aligned} \tag{53}$$

Define the matrix of all the NN weights as

$$T \equiv \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \tag{54}$$

Assumption: On any compact subset of  $\mathfrak{R}^n$ , the ideal NN weights are bounded so that

$$\|T\|_F \leq T_B$$

with  $T_B$  known, and  $\|\cdot\|_F$  is the Frobenius norm. Then

$$\begin{aligned} \dot{L} = & -K_v r^2 - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{krtr}\{\tilde{T}^T (T - \tilde{T})\} \\ & + (r - r_e) \text{tr}\{\tilde{B}^T \dot{Z} + \tilde{\alpha}^T x^e \hat{B} \dot{Z}\} - k_\alpha |r_e| \text{tr}\{(\bar{\alpha} - \hat{\alpha})\} \end{aligned} \tag{55}$$

Since  $\text{tr}\{\tilde{T}^T (T - \tilde{T})\} \leq \|\tilde{T}\|_F \|T\|_F - \|\tilde{T}\|_F^2$ , the results are as follows[28]:

$$\begin{aligned} \dot{L} \leq & -r K_v r - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 + \text{kr} \cdot \tilde{T}_F (T_B - \tilde{T}_F) \\ & + (r - r_e) \text{tr}\{\tilde{B}^T \dot{Z} + \tilde{\alpha}^T x^e \hat{B} \dot{Z}\} - k_\alpha |r_e| \text{tr}\{(\bar{\alpha} - \hat{\alpha})\} \\ \dot{L} \leq & -r K_v r - \frac{1}{2} \frac{\dot{G}}{G^2} r^2 - \text{kr} \cdot (\tilde{T}_F - D)^2 + \text{kr} D^2 \\ & + (r - r_e) \text{tr}\{\tilde{B}^T \dot{Z} + \tilde{\alpha}^T x^e \hat{B} \dot{Z}\} - k_\alpha |r_e| \text{tr}\{(\bar{\alpha} - \hat{\alpha})\} \\ \dot{L} \leq & -\frac{1}{2} \frac{\dot{G}}{G^2} r^2 - \text{kr} \cdot (\tilde{Z}_F - D)^2 - r(K_v r - kD^2) \\ & - \left(\bar{w}_g + k_\alpha \frac{\alpha_m^2}{4} + k_\alpha \tilde{\alpha} \tilde{\alpha}\right) |r_e| \\ \dot{L} \leq & -\frac{1}{2} \frac{\dot{G}}{G^2} r^2 - \text{kr} \cdot (\tilde{Z}_F - D)^2 - r(K_v r - kD^2) \\ & - \left(\bar{w}_g + k_\alpha \frac{\alpha_m^2}{4}\right) |r_e| - k_\alpha |r_e| \alpha_n^2 \end{aligned} \tag{56}$$

where  $D = \frac{T_B}{2}$ . Suppose that  $k > 0$ , using the inequality in (15), the first term is negative; then, we prove that the Lyapunov first-time derivative becomes negative if:

$$r > \frac{kD^2}{K_v}$$

Then, we prove that the Lyapunov first-time derivative will be:

$$\dot{L} \leq 0$$

which guarantees the stability of closed-loop system [3, 27, 28].

## References

- Kurode S, Asif C (2011) Swing-up and stabilization of rotary inverted pendulum. In: Preprints of the 18th IFAC world congress, Milano (Italy)
- Shubhobrata R, Ranjit kumar B (2013) Stabilization of furuta pendulum: a backstepping based hierarchical sliding mode approach with disturbance estimation. In: 7th International conference on intelligent systems and control (ISCO)
- Moreno-Valenzuela J (2016) Adaptive neural network control for the trajectory tracking of the furuta pendulum. *IEEE Trans Cybern* 46:3439–3452
- Furuta K, Yamakita M, Kobayashi S (1991) Swing-up control of inverted pendulum. In: International conference on industrial electronics, control and instrumentation
- Huang Y-F, Chen A-C (2013) Adaptive control of rotary inverted pendulum system with time-varying uncertainties. Springer, Berlin, pp 1112–1114
- Acosta J (2010) Furuta's pendulum: a conservative nonlinear model for theory and practice. *Math Probl Eng* 2010:742894. <https://doi.org/10.1155/2010/742894>
- Aguilar-Ibañez C, Suárez-Castañón M, Gutiérrez- Frias O (2010) The direct Lyapunov method for the stabilization of the Furuta pendulum. *Int J Control* 83(11):2285–2293
- Turker T, Gorgun H, Cansever G (2012) Lyapunov's direct method for stabilization of the Furuta pendulum. *Turk J Electr Eng Comput Sci* 120(1):99–110
- de Jesús RJ (2018) Discrete time control based in neural networks for pendulums. *Appl Soft Comput J* 68:821–832
- Astrom K, Aracil J, Gordillo F (2008) A family of smooth controller for swinging up a pendulum. *Automatica* 36(2):1841–1848
- Gordillo F, Acosta J, Aracil J (2003) A new swing-up law for the Furuta pendulum. *Int J Control* 76(8):836–844
- Hera P, Freidovich L, Shiriaev A, Mettin U (2009) New approach for swing up the Furuta pendulum: theory and experiments. *Mechatronics* 19:1240–1250
- Hercus R, Wong K-Y (2013) Control of an inverted pendulum using the NeuraBase. In: International conference on neural information processing
- Aguilar L, Boiko I, Fridman L, Iriarte R (2009) Generating self-excited oscillations via two-relay controller. *IEEE Trans Autom Control* 54(2):416–420
- Freidovich L, Shiriaev A, Gordillo F, Gomez-Estern F, Aracil J (2009) Partial-energy-shaping control for orbital stabilization of high-frequency oscillations of the Furuta pendulum. *IEEE Trans Control Syst Technol* 17(4):853–858
- Sicard P, Chaoui H (2011) Motion and balance neural control of inverted pendulums with nonlinear friction and disturbance. In: Proceedings of the 24th Canadian conference on electrical and computer engineering (CCECE), Niagara Falls, Canada
- Nelson J, Kraft LG (1994) Real-time control of an inverted pendulum system using complementary neural network and optimal techniques. In: American control conference, Baltimore
- Cong S, Liang Y (2009) PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems. *IEEE Trans Ind Electron* 56:3872–3879
- Hsu C-F (2014) Adaptive backstepping Elman-based neural control for unknown nonlinear systems. *Neurocomputing* 136:170–179
- Hsu C-F, Lin C-M, Yeh R-G (2013) Supervisory adaptive dynamic RBF-based neural-fuzzy control system design for unknown nonlinear systems. *Appl Soft Comput* 13:1620–1626
- Ping Z (2013) Tracking problems of a spherical inverted pendulum via neural network enhanced design. *Neurocomputing* 106:137–147
- Yu C, Wang F, Lu Y (2010) Robust control of a Furuta pendulum. In: SICE annual conference
- Khanesar M, Teshnehlab M, Shoorehdeli M (2007) Fuzzy sliding mode control of rotary inverted pendulum. In: IEEE international conference on computational cybernetics
- Van Cuong P, Nan WY (2016) Adaptive trajectory tracking neural network control with robust compensator for robot manipulators. *Nat Comput Appl* 27:525–536
- Fantoni I (2002) Non-linear control for underactuated mechanical systems. Springer, London
- Olfati-Saber R (1999) Fixed point controllers and stabilization of the cart-pole system and the rotating pendulum. In: Proceedings of the 38th IEEE conference on decision and control, vol 2, No. December, pp 1174–1181
- Boudjedir H, Yacef F, Bouhalim O, Rizoug A (2012) Dual neural network for adaptive sliding mode control of quadrotor helicopter stabilization. *Int J Inf Sci Techn (IJIST)* 2:1–14
- Lewis FW, Jagannathan S, Yesildirak A (1998) Neural network control of robot manipulators and non-linear systems. Taylor & Francis, London
- Shuzhi S, Ge S, Hang CC, Lee TH, Zhang T (2002) Stable adaptive neural network control. Springer, New York
- Kim S, Jung SS (2008) Control experiment of a wheel-driven mobile inverted pendulum using neural network. *IEEE Trans Control Syst Technol* 16(2):297–303
- Zhang B-L, Han Q-L, Zhang X-M (2017) Recent advances in vibration control of offshore platforms. *Nonlinear Dyn* 89(2):755–771
- Zhang B-L, Han Q-L, Zhang X-M (2014) Sliding mode control with mixed current and delayed states for offshore steel jacket platforms. *IEEE Trans Control Syst Technol* 22:1769–1783
- George S, Feng L (2001) Adaptive neural network control by adaptive interaction. In: Proceedings of the American control conference
- Robert DB, Feng L (1999) Adaptive interaction and its application to neural networks. *Inf Sci* 121(3):201–215
- Kurode S, Asif C (2011) Swing-up and stabilization of rotary inverted pendulum using sliding modes. In: IFAC, Milano

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.