We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

A New Approach of the Online Tuning Gain Scheduling Nonlinear PID Controller Using Neural Network

Ho Pham Huy ANH[1] and Nguyen Thanh Nam[2]
[1]Corresponding author, Ho Chi Minh City University of Technology, Ho Chi Minh City,
Viet Nam
(Tel: +84-908229736; Email: hphanh@hcmut.edu.vn)
[2]DCSELAB, Viet Nam National University Ho Chi Minh City (VNU-HCM), Viet Nam
(Tel: +84-0908150134; E-mail: thanhnam@vnuhcm.edu.vn)

Abstract

This chapter presents the design, development and implementation of a novel proposed online-tuning Gain Scheduling Dynamic Neural PID (DNN-PID) Controller using neural network suitable for real-time manipulator control applications. The unique feature of the novel DNN-PID controller is that it has highly simple and dynamic self-organizing structure, fast online-tuning speed, good generalization and flexibility in online-updating. The proposed adaptive algorithm focuses on fast and efficiently optimizing Gain Scheduling and PID weighting parameters of Neural MLPNN model used in DNN-PID controller. This approach is employed to implement the DNN-PID controller with a view of controlling the joint angle position of the highly nonlinear pneumatic artificial muscle (PAM) manipulator in real-time through Real-Time Windows Target run in MATLAB SIMULINK® environment. The performance of this novel proposed controller was found to be outperforming in comparison with conventional PID controller. These results can be applied to control other highly nonlinear SISO and MIMO systems.

Keywords: highly nonlinear PAM manipulator, proposed online tuning Gain Scheduling Dynamic Nonlinear PID controller (DNN-PID), real-time joint angle position control, fast online tuning back propagation (BP) algorithm, pneumatic artificial muscle (PAM) actuator.

1. Introduction

The compliant manipulator was used to replace monotonous and dangerous tasks, which has enhanced lots of researchers to develop more and more intelligent controllers for human-friendly industrial manipulators. Due to uncertainties, it is difficult to obtain a precise mathematical model for robot manipulators. Hence conventional control methodologies find it difficult or impossible to handle un-modeled dynamics of a robot manipulator. Furthermore, most of conventional control methods, for example PID

controllers, are based on mathematical and statistical procedures for modeling the system and estimation of optimal controller parameters. In practice, such manipulator is often highly non-linear and a mathematical model may be difficult to derive. Thus, as to accommodate system uncertainties and variations, learning methods and adaptive intelligent techniques must be incorporated.

Due to their highly nonlinear nature and time-varying parameters, PAM robot arms present a challenging nonlinear model problem. Approaches to PAM control have included PID control, adaptive control (Lilly, 2003), nonlinear optimal predictive control (Reynolds *et al.*, 2003), variable structure control (Repperger *et al.*, 1998; Medrano-Cerda *et al.*,1995), gain scheduling (Repperger *et al.*,1999), and various soft computing approaches including neural network Kohonen training algorithm control (Hesselroth *et al.*,1994), neural network + nonlinear PID controller (Ahn and Thanh, 2005), and neuro-fuzzy/genetic control (Chan *et al.*, 2003; Lilly *et al.*, 2003). Balasubramanian *et al.*, (2003a) applied the fuzzy model to identify the dynamic characteristics of PAM and later applied the nonlinear fuzzy model to model and to control of the PAM system. Lilly (2003) presented a direct continuous-time adaptive control technique and applied it to control joint angle in a single-joint arm. Tsagarakis *et al.* (2000) developed an improved model for PAM. Hesselroth *et al.* (1994) presented a neural network that controlled a five-link robot using back propagation to learn the correct control over a period of time. Repperger *et al.* (1999) applied a gain scheduling model-based controller to a single vertically hanging PAM. Chan *et al.*, (2003) and Lilly *et al.*, (2003) introduced a fuzzy P+ID controller and an evolutionary fuzzy controller, respectively, for the PAM system. The novel feature is a new method of identifying fuzzy models from experimental data using evolutionary techniques. Unfortunately, these fuzzy models are clumsy and have only been tested in simulation studies. (Ahn and Anh, 2006) applied a modified genetic algorithm (MGA) for optimizing the parameters of a linear ARX model of the PAM manipulator which can be modified online with an adaptive self-tuning control algorithm, and then (Ahn and Anh, 2007b) successfully applied recurrent neural networks (RNN) for optimizing the parameters of neural NARX model of the PAM robot arm. Recently, we (Ahn and Anh, 2009) successfully applied the modified genetic algorithm (MGA) for optimizing the parameters of the NARX fuzzy model of the PAM robot arm.

Although these control systems were partially successful in obtaining smooth actuator motion in response to input signals, the manipulator must be controlled slowly in order to get stable and accurate position control. Furthermore the external inertia load was also assumed to be constant or slowly varying. It is because PAM manipulators are multivariable non-linear coupled systems and frequently subjected to structured and/or unstructured uncertainties even in a well-structured setting for industrial use or human-friendly applications as well.

To overcome these drawbacks, the proposed online tuning DNN-PID algorithm in this chapter is a newly developed algorithm that has the following good features such as highly simple and dynamic self-organizing structure, fast learning speed, good generalization and flexibility in learning. The proposed online tuning DNN-PID controller is employed to compensate for environmental variations such as payload mass and time-varying parameters during the operation process. By virtue of on-line training by back propagation (BP) learning algorithm and then auto-tuned gain scheduling K and PID weighting values K_p, K_i and K_d, it learns well the nonlinear robot arm dynamics and simultaneously makes control decisions to both of joints of the robot arm. In effect, it offers an exciting on-line estimation scheme.

This chapter composes of the section 1 for introducing related works in PAM robot arm control. The section 2 presents procedure of design an online tuning gain scheduling DNN-PID controller for the 2-axes PAM robot arm. The section 3 presents and analyses experiment studies and results. Finally, the conclusion belongs to the section 4.

2. Control System

2.1. Experimental apparatus

The PAM manipulator used in this paper is a two-axis, closed-loop activated with 2 antagonistic PAM pairs which are pneumatic driven controlled through 2 proportional valves. Each of the 2-axes provides a different motion and contributes to 1 degree of freedom of the PAM manipulator (Fig. 1). In this paper, the 1st joint of the PAM manipulator is fixed and proposed online tuning Gain Scheduling neural DNN-PID control algorithm is applied to control the joint angle position of the 2nd joint of the PAM manipulator. A general configuration of the investigated 2-axes PAM manipulator shown through the schematic diagram of the 2-axes PAM robot arm and the experimental apparatus presented in Fig.1 and Fig.2, respectively.
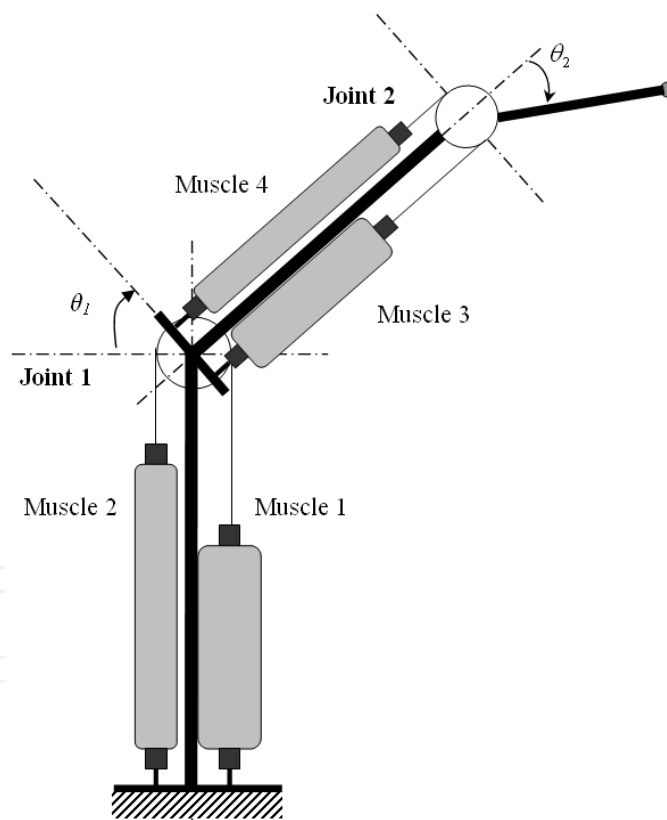

Fig. 1. Working principle of the 2-axes PAM robot arm.

The experiment system is illustrated in Fig.2. The air pressure proportional valve manufactured by FESTO Corporation is used. The angle encoder sensor is used to measure the output angle of the joint. The entire system is a closed loop system through computer. First, initial control voltage value $u_0(t)$=5[V] is sent to proportional valve as to inflate the

artificial muscles with air pressure at P_0 (initial pressure) to render the joint initial status. Second, by changing the control output $u(t)$ from the D/A converter, we could set the air pressures of the two artificial muscles at $(P_0 + \Delta P)$ and $(P_0 - \Delta P)$, respectively. As a result, the joint is forced to rotate for a certain angle. Then we can measure the joint angle rotation through the rotary encoder and the counter board and send it back to PC to have a closed loop control system.
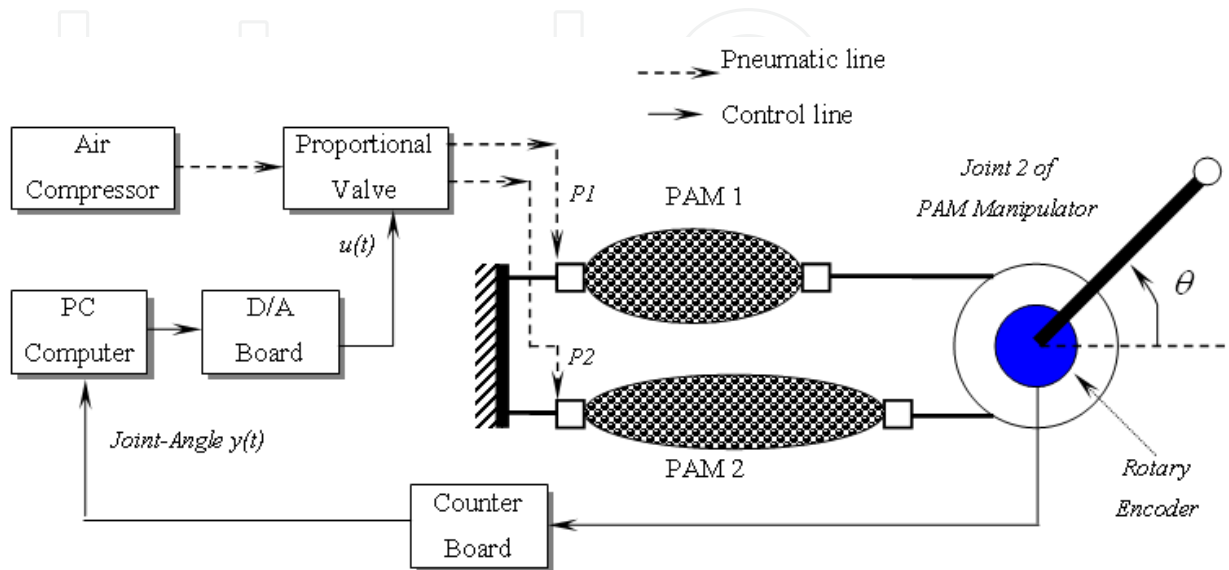
Fig. 2. Experimental Set-up Configuration of the PAM robot arm.
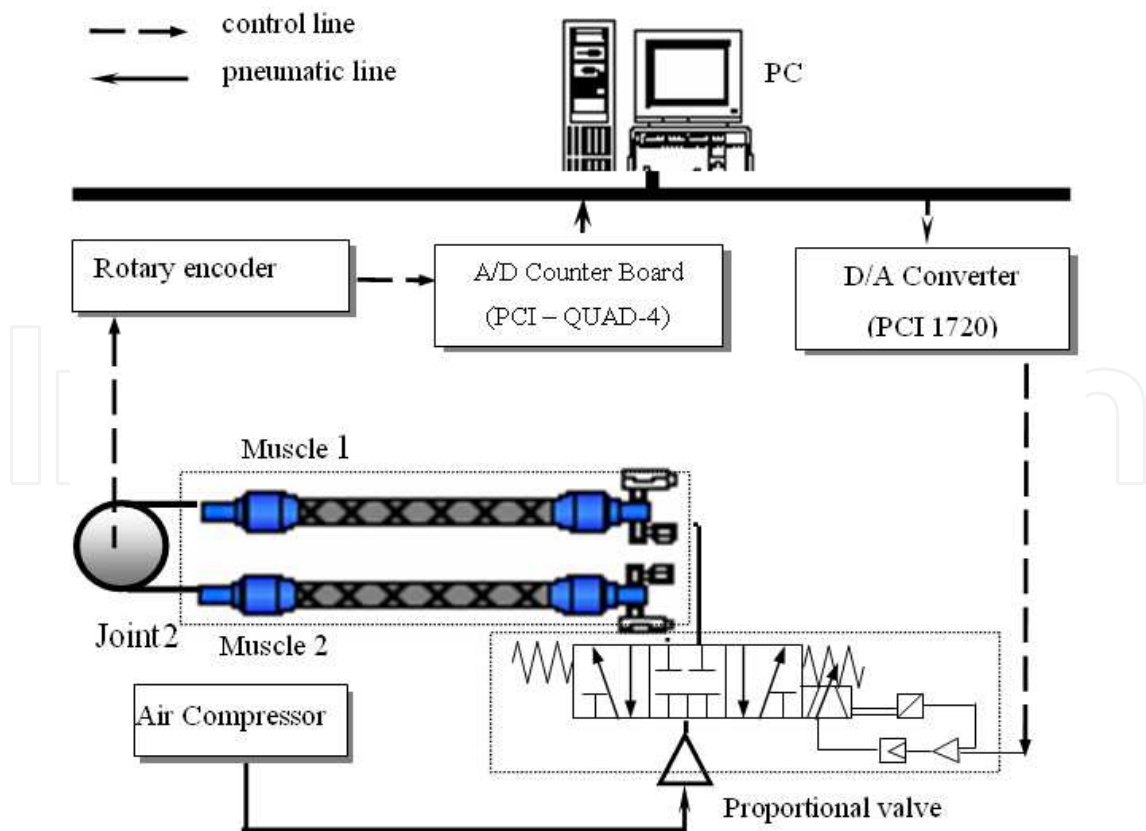
Fig. 3. Schematic diagram of the experimental apparatus.

The experimental apparatus is shown in Fig.3. The hardware includes an IBM compatible PC (Pentium 1.7 GHz) which sends the control voltage signal *u(t)* to control the proportional valve (FESTO, MPYE-5-1/8HF-710B), through a D/A board (ADVANTECH, PCI 1720 card) which change digital signal from PC to analog voltage *u(t)*. The rotating torque is generated by the pneumatic pressure difference supplied from air-compressor between the antagonistic artificial muscles. Consequently, the 2nd joint of PAM manipulator will be rotated. The joint angle, θ [deg], is detected by a rotary encoder (METRONIX, H40-8-3600ZO) with a resolution of 0.1[deg] and fed back to the computer through an 32-bit counter board (COMPUTING MEASUREMENT, PCI QUAD-4 card) which changes digital pulse signals to joint angle value *y(t)*. The external inertia load could be changed from 0.5[kg] to 2[kg], which is a 400 (%) change with respect to the minimum inertia load condition. The experiments are conducted under the pressure of 4[bar] and all control software is coded in MATLAB-SIMULINK with C-mex S-function.

Table 1 presents the configuration of the hardware set-up installed from Fig.2 and Fig.3 as to control of the 2nd joint of the PAM manipulator using the novel proposed online tuning Gain Scheduling DNN-PID control algorithm.

No.	Name	Model name	Company
1	Proportional valve	MPYE-5-1/8HF-710 B	FESTO
2	Pneumatic artificial muscle (x2)	MAS-10-N-220-AA-MCFK	FESTO
3	D/A board	PCI 1720	ADVANTECH
4	A/D board	PCI QUAD-4	COMPUTING MEASUREMENT
5	Rotary encoder	H40-8-3600ZO	METRONIX

Table 1. Lists of the experimental hardware set-up.

2.2. Controller design

The structure of the newly proposed online tuning Gain Scheduling DNN-PID control algorithm using neural network is shown in Fig. 4. This control algorithm is a new one and has the characteristics such as simple structure and little computation time, compared with the previous neural network controller using auto-tuning method (Ahn K.K., Thanh T.D.C., 2005). This system with the set point filter and controller using neural network can solve the problems, which were mentioned in the introduction and is also useful for the PAM manipulator with nonlinearity properties.
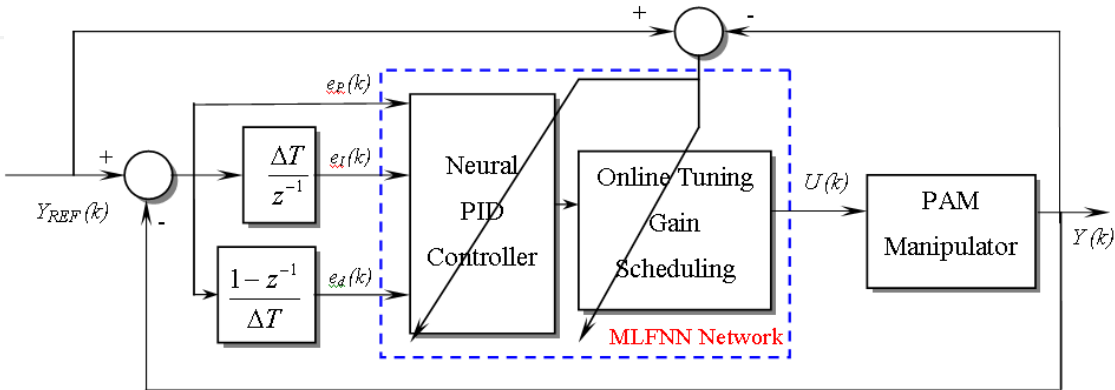

Fig. 4. Block diagram of proposed online tuning gain scheduling DNN-PID position control system.

The block diagram of proposed online tuning Gain Scheduling DNN-PID control based on Multi-Layer Feed-Forward Neural Network (MLFNN) composed of three layers is shown in Figure 5.
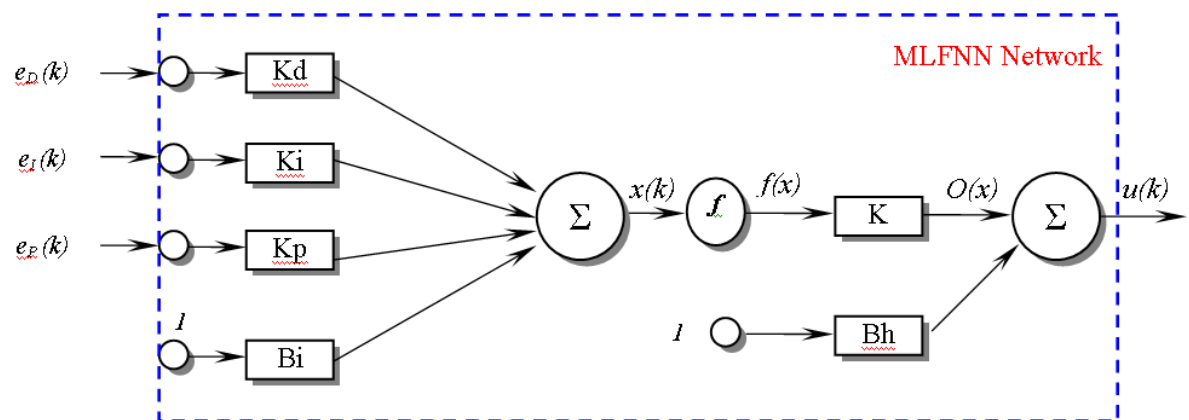

Fig. 5. Structure of MLFNN network system used in proposed online tuning DNN-PID controller.

The structure of the newly proposed online tuning Gain Scheduling DNN-PID control algorithm using Multi-Layer Feed-forward Neural Network (MLFNN) is shown in Fig.5. This control algorithm is a new one and has the characteristics such as simple structure, little computation time and more robust control, compared with the previous neural network controller using auto-tuning method (Ahn K.K., Thanh T.D.C., 2005).

From Figures 4 and 5, a control input u applied to the 2nd joints of the 2-axes PAM manipulator can be obtained from the following equation.

$$u = K\,f(x) + B_h \tag{1}$$

with x is input of Hyperbolic Tangent function $f(.)$ which is presented in Equation (2), K and B_h are the bias weighting values of input layer and hidden layer respectively. The Hyperbolic Tangent function $f(.)$ has a nonlinear relationship as explained in the following equation.

$$f(x) = \frac{\left(1 - e^{-x}\right)}{\left(1 + e^{-x}\right)} \tag{2}$$

The block diagram of proposed online tuning Gain Scheduling DNN-PID control based on Multi-Layer Feed-Forward Neural Network (MLFNN) composed of three layers is shown in Figure 5. In this figure, K, K_p, K_i and K_d, are scheduling, proportional, integral and derivative gain while e_p, e_i and e_d are system error between desired set-point output and output of joint of the PAM manipulator, integral of the system error and the difference of the system error, respectively.

MLFNN network is trained online by the fast learning back propagation (FLBP) algorithm as to minimize the system error between desired set-point output and output of joint of the PAM manipulator.

From Figure 5, the input signal of the Hyperbolic Tangent function *f(.)* becomes

$$x(k) = K_p(k)e_p(k) + K_i(k)e_i(k) + K_d(k)e_d(k) + B_i(k)$$
$$O(k) = f(x(k))$$
$$u(k) = K(k)O(k) + B_h(k)$$

(3)

with

$$e_p(k) = y_{REF}(k) - y(k)$$
$$e_i(k) = e_p(k).\Delta T$$
$$e_d(k) = \frac{e_p(k)(1 - z^{-1})}{\Delta T}$$

(4)

ΔT is the sampling time, z is the operator of Z-Transform, k is the discrete sequence, $y_{REF}(k)$ and $y(k)$ are the desired set-point output and output of joint of the PAM manipulator. Furthermore, B_i, K_p, K_i and K_d are weighting values of Input layer and B_h *and* K are weighting values of Hidden layer. These weighting values will be tuned online by fast learning back propagation (FLBP) algorithm.

As to online tuning the gain scheduling K and PID parameters K_p, K_i and K_d, the gradient descent method used in FLBP learning algorithm using the following equations were applied.

$$K(k+1) = K(k) - \eta \frac{\partial E(k)}{\partial K}$$
$$K_p(k+1) = K_p(k) - \eta_p \frac{\partial E(k)}{\partial K_p}$$
$$K_i(k+1) = K_i(k) - \eta_i \frac{\partial E(k)}{\partial K_i}$$
$$K_d(k+1) = K_d(k) - \eta_d \frac{\partial E(k)}{\partial K_d}$$

(5)

and the Bias weighting values $B_i(k)$ and $B_h(k)$ are updated as follows:

$$B_i(k+1) = B_i(k) - \eta_{Bi} \frac{\partial E(k)}{\partial B_i}$$
$$B_h(k+1) = B_h(k) - \eta_{Bh} \frac{\partial E(k)}{\partial B_h}$$

(6)

where η, η_p, η_i, η_d, η_{Bi} and η_{Bh} are learning rate values determining the convergence speed of updated weighting values; E(k) is the error defined by the gradient descent method as follows

$$E(k) = \frac{1}{2}\left(y_{REF}(k) - y(k)\right)^2 \tag{7}$$

Apply the chain rule with equation 5 and 6, it leads to

$$\frac{\partial E(k)}{\partial K} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial K}$$

$$\frac{\partial E(k)}{\partial K_p} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_p} \tag{8}$$

$$\frac{\partial E(k)}{\partial K_i} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_i}$$

$$\frac{\partial E(k)}{\partial K_d} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_d}$$

And

$$\frac{\partial E(k)}{\partial B_i} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial B_i}$$

$$\frac{\partial E(k)}{\partial B_h} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial B_h} \tag{9}$$

From equations 1, 3 and 6, the following equations can be derived

$$\frac{\partial E(k)}{\partial y} = -\left(y_{REF}(k) - y(k)\right) = -e_p(k)$$

$$\frac{\partial y(k)}{\partial u} \approx \frac{\Delta y}{\Delta u} = \frac{\left(y(k) - y(k-1)\right)}{\left(u(k) - u(k-1)\right)} = \Delta$$

$$\frac{\partial u(k)}{\partial O} = K \tag{10}$$

$$\frac{\partial O(k)}{\partial x} = f'\left(x(k)\right)$$

$$\frac{\partial x(k)}{\partial B_i} = 1; \frac{\partial x(k)}{\partial K_p} = e_p(k); \frac{\partial x(k)}{\partial K_i} = e_i(k); \frac{\partial x(k)}{\partial K_d} = e_d(k)$$

$$\frac{\partial u(k)}{\partial K} = O(k); \frac{\partial u(k)}{\partial B_h} = 1$$

From equations 8, 9 and 10, the following resulting equations can be derived

$$\frac{\partial E(k)}{\partial K} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial K} = -e_p(k)\Delta O(k)$$

$$\frac{\partial E(k)}{\partial K_p} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_p} = -e_p(k)\Delta K f'(x(k))e_p(k) = -\Delta K f'(x)e_p^2(k) \quad (11)$$

$$\frac{\partial E(k)}{\partial K_i} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_i} = -e_p(k)\Delta K f'(x(k))e_i(k) = -\Delta K f'(x)e_p(k)e_i(k)$$

$$\frac{\partial E(k)}{\partial K_d} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial K_d} = -e_p(k)\Delta K f'(x(k))e_d(k) = -\Delta K f'(x)e_p(k)e_d(k)$$

and

$$\frac{\partial E(k)}{\partial B_i} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial O}\frac{\partial O(k)}{\partial x}\frac{\partial x(k)}{\partial B_i} = -e_p(k)\Delta K f'(x(k))\mathbb{1} = -\Delta K f'(x)e_p(k)$$

$$\frac{\partial E(k)}{\partial B_h} = \frac{\partial E(k)}{\partial y}\frac{\partial y(k)}{\partial u}\frac{\partial u(k)}{\partial B_h} = -e_p(k)\Delta\mathbb{1} = -\Delta e_p(k) \quad (12)$$

and with

$$f'(x) = 2.\frac{e^{-x}}{\left(1+e^{-x}\right)^2} \quad (13)$$

From equation 5 and 6, the final equations for online tuning gain scheduling K and PID parameters K_p, K_i and K_d are expressed as follows:

$$K(k+1) = K(k) + \eta.e_p(k)\Delta.O(k)$$

$$K_p(k+1) = K_p(k) + \eta_p.e_p^2(k)\Delta K.\frac{2e^{-x}}{\left(1+e^{-x}\right)^2}$$

$$K_i(k+1) = K_i(k) + \eta_i.e_p(k)e_i(k)\Delta K\frac{2e^{-x}}{\left(1+e^{-x}\right)^2} \quad (14)$$

$$K_d(k+1) = K_d(k) + \eta_d.e_p(k)e_i(k)\Delta K\frac{2e^{-x}}{\left(1+e^{-x}\right)^2}$$

and the Bias weighting values $B_i(k)$ and $B_h(k)$ are updated as follows:

$$B_i(k+1) = B_i(k) + \eta_{Bi}.e_p(k)\Delta K \frac{2e^{-x}}{\left(1+e^{-x}\right)^2} \tag{15}$$

$$B_h(k+1) = B_h(k) + \eta_{Bh}e_p(k)\Delta$$

3. Experimental Results

The performance of proposed online tuning gain scheduling DNN-PID control scheme is verified on joint angle position control of the 2nd joint of the 2-axes PAM robot arm. Fig.3 and Fig.4 describes the working diagram of this control scheme.

Fig.6 presents the experiment SIMULINK diagram of proposed online tuning DNN-PID control algorithm run in Real-time Windows Target with DYNAMIC_NEURAL_PID being subsystems written in C then compiled and run in real-time C-mex file. Three initial PID parameters K_P, K_I, K_D and gain scheduling G value are chosen by trial and error method and determined as G=0.8; K_P=0.09; K_I=0.089 and K_D=0.07.
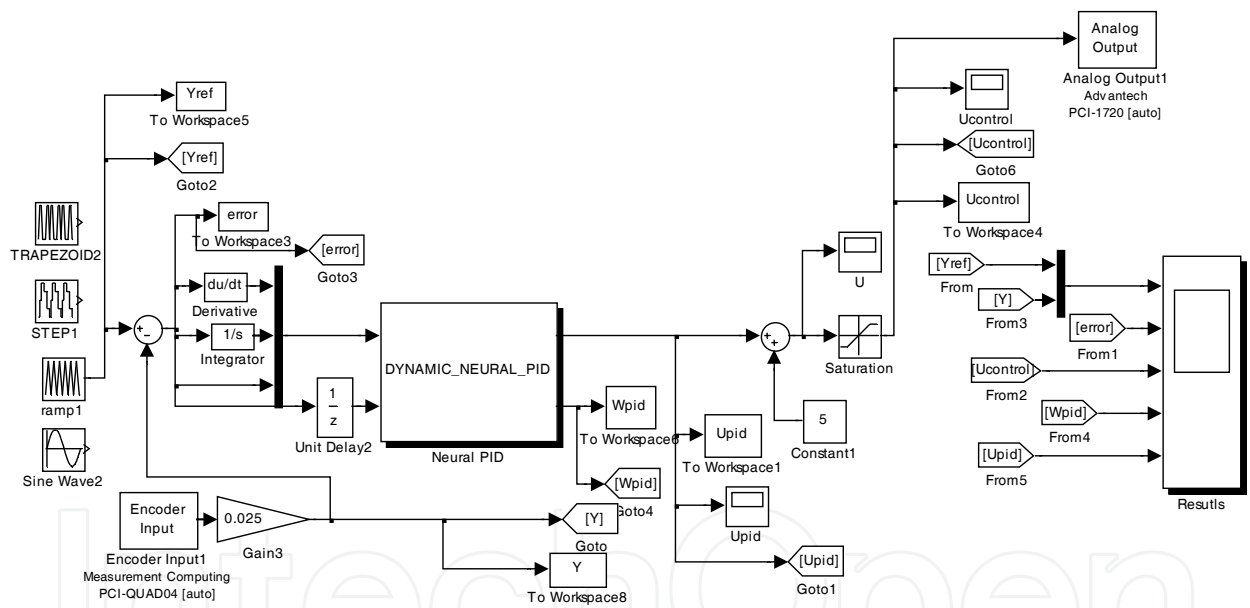
Fig. 6. Experiment SIMULINK Model of PAM robot arm control using online tuning DNN-PID control.

The experiment SIMULINK diagram of the 2nd joint of the 2-axes PAM robot arm position control using conventional PID controller in order to compare as to demonstrate the superiority of proposed control system. Three PID parameters K_P, K_I, K_D and gain scheduling K value of conventional PID controller are chosen by trial and error method.
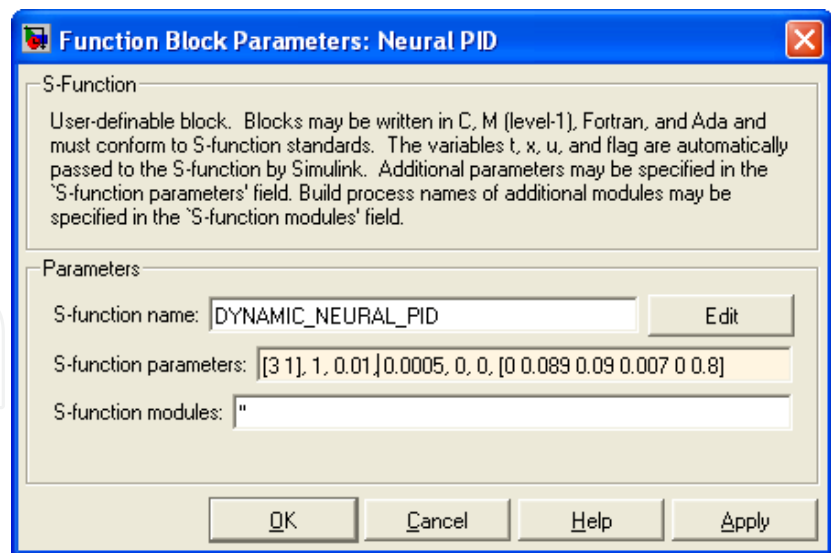
Fig. 7. Parameter configuration of DNN_PID subsystem used in proposed online tuning DNN-PID control

Fig. 7 shows that the parameter configuration of DYNAMIC_NEURAL_PID subsystem composes of seven parameters. The 1st vector parameter contains number of inputs and outputs of neural DYNAMIC_NEURAL_PID subsystem; the 2nd relates to the number of neurons of Hidden Layer used; the 3rd declares the step size used in real-time operation of PAM system; the 4th declares the learning rate value used in real-time operation of PAM manipulator; the 5th parameter contains logic value as to choose the sigmoid function (1) or the hyperbolic tangent function (0); the 6th parameter contains logic value as to choose the linear function (0) or the sigmoid/hyperbolic tangent function (1) of Output layer; and the 7th vector parameter contains the initial K, K_P, K_I, K_D weighting values and two initial bias weighting values B_i and B_h.

The final purpose of the PAM manipulator is to be used as an elbow and wrist rehabilitation robot device. Thus, the experiments were carried out with respect to 3 different waveforms as reference input (Triangular, Trapezoidal and Sinusoidal reference) with 2 different end-point Payloads (Load 0.5[kg] and Load 2[kg]) as to demonstrate the performance of novel proposed online tuning DNN-PID controller. Furthermore, the comparisons of control performance between the conventional PID and two different methods of the proposed online tuning DNN-PID controller were performed.

These two novel proposed methods compose of proposed online tuning DNN-PID-SIG and proposed online tuning DNN-PID-HYP. The 1st method possesses the activation function of hidden layer of DYNAMIC_NEURAL_PID subsystem being Sigmoid function and the 2nd method corresponds to the Hyperbolic Tangent function respectively.

The initial gain scheduling value G and PID controller parameters K_p, K_i and K_d were set to be $G = 0.8$, $Kp = 0.089$, $Ki = 0.09$, $Kd = 0.07$. These parameters of PID controller were obtained by trial-and-error through experiment. Forwardly, the two initial bias weighting values B_i and B_h are chosen equal 0.

First, the experiments were carried out to verify the effectiveness of the proposed online tuning DNN-PID controller using neural network with triangular reference input. Fig.8a shows the experimental results between the conventional PID controller and the proposed

nonlinear DNN-PID controller in 2 cases of Load 0.5[kg] and Load 2[kg] respectively. The online updating of each control parameter (G, Kp, Ki and Kd) in 2 cases of Load 0.5[kg] and Load 2[kg] was shown in Fig. 8b. In the experiment of the proposed online tuning DNN-PID controller, the initial values of G, Kp, Ki and Kd are set to be the same as that of conventional PID controller.

These figures show that thanks to the sophisticated online tuning of G, Kp, Ki and Kd, the error between desired reference y_{REF} and actual joint angle response y of the PAM manipulator continually optimized. Consequently, the minimized error decreases only in the range $\pm$ 0.5[deg] with both of proposed DNN-PID-SIG and DNN-PID-HYP in case of Load 0.5[kg]. The same good result is obtained with both of proposed DNN-PID-SIG and DNN-PID-HYP in case of Load 2[kg]. These results are really impressive in comparison with the bad and unchanged error of conventional PID controller ($\pm$ 1.5[deg] in case of Load 0.5[kg] and up to $\pm$ 2[deg] in case of Load 2[kg]). Furthermore, in case of Load 2[kg], Figure 8a shows that PID controller caused the PAM manipulator response being oscillatory and unstable. Otherwise, proposed online tuning DNN-PID controller continues to assert robust control to keep PAM manipulator response stable and accurate tracking.

In comparison between proposed DNN-PID-SIG and DNN-PID-HYP, both of proposed control algorithms obtain the excellent robustness and accuracy as well and thus are considered the performance equivalent. However in initial stage, proposed DNN-PID-SIG possesses significant overshoot which may cause unstable to PAM manipulator in its initial operation.
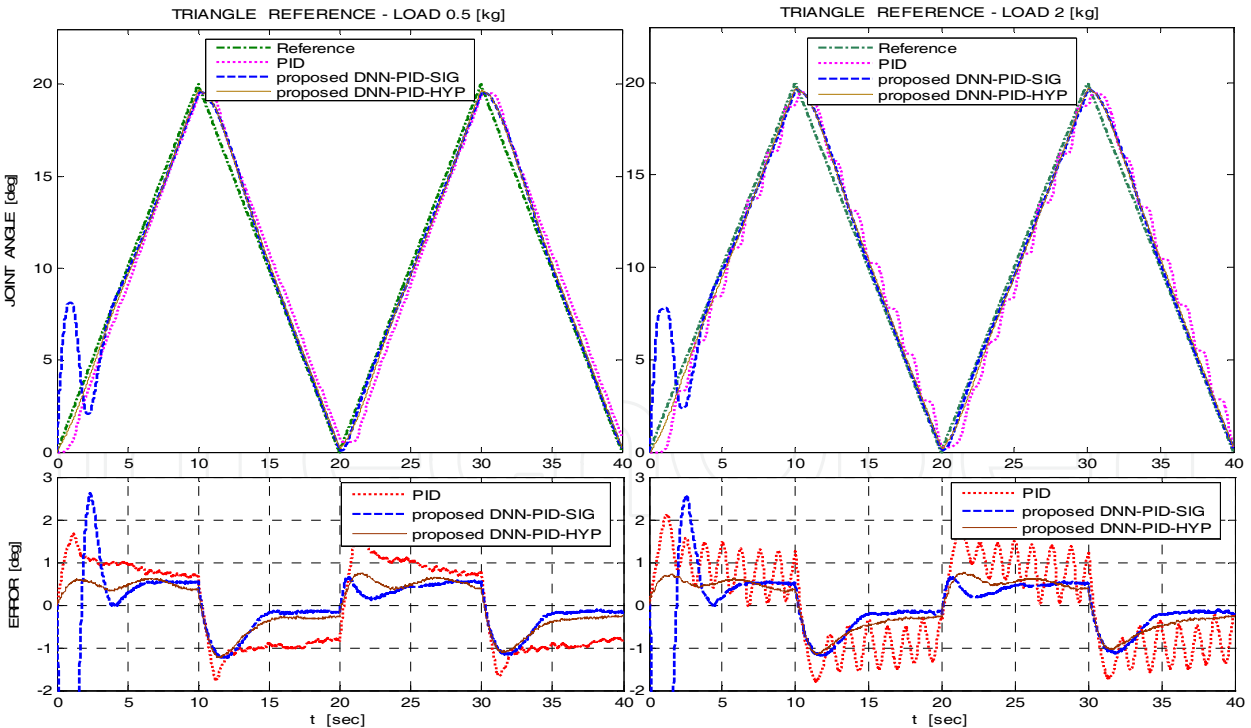

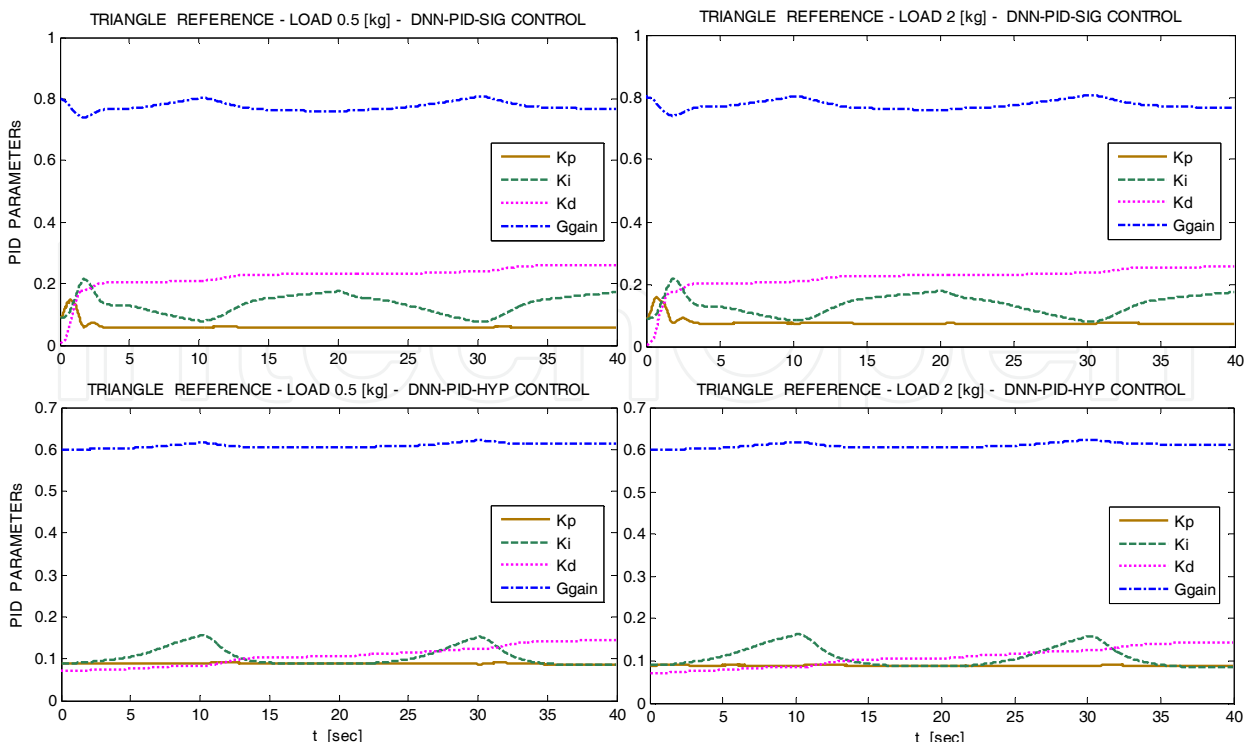Fig. 8a.Triangular response of the PAM robot arm – Load 0.5[kg] and Load 2[kg].

Fig. 8b. The online tuning convergence of proposed DNN-PID controller parameters with triangular reference.

Figure 8c shows the resulted shape of control voltage *U* applied to the joint of PAM manipulator, which is generated by the proposed online tuning DNN-PID controller as to assure the performance and the accuracy of the PAM manipulator response. This figure shows that PID controller generates an oscillatory and unstable control voltage in case of Load 2[kg]. On the contrary, proposed online tuning DNN-PID controller continues to robustly control with refined control voltage as to keep PAM manipulator response stable and accurate tracking.
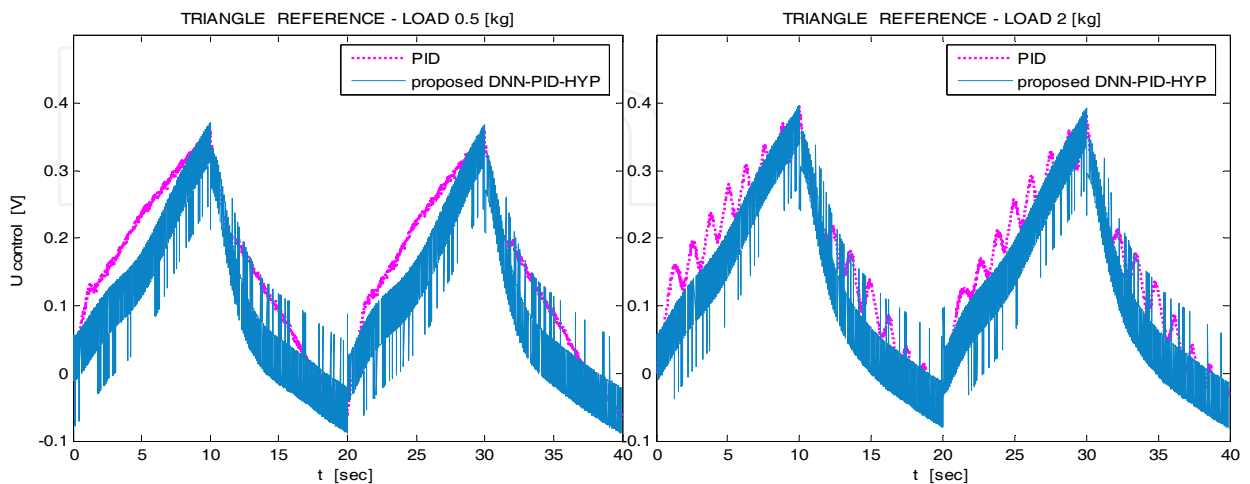
Fig. 8c.The voltage control applied to the PAM robot arm with triangular reference.

Forwardly, the experiments were carried out to verify the effectiveness of the proposed DNN-PID controller using neural network with trapezoidal reference input. Fig.9a shows the experimental results in comparison between the conventional PID controller and the two proposed nonlinear DNN-PID-SIG and DNN-PID-HYP controllers in 2 cases of Load 0.5[kg] and Load 2[kg] respectively. The online updating of each control parameter (G, Kp, Ki and Kd) in 2 cases of Load 0.5[kg] and Load 2[kg] was shown in Fig. 9b. In the experiment of the proposed online tuning DNN-PID controller, the initial values of G, Kp, Ki and Kd are set to be the same as that of conventional PID controller.
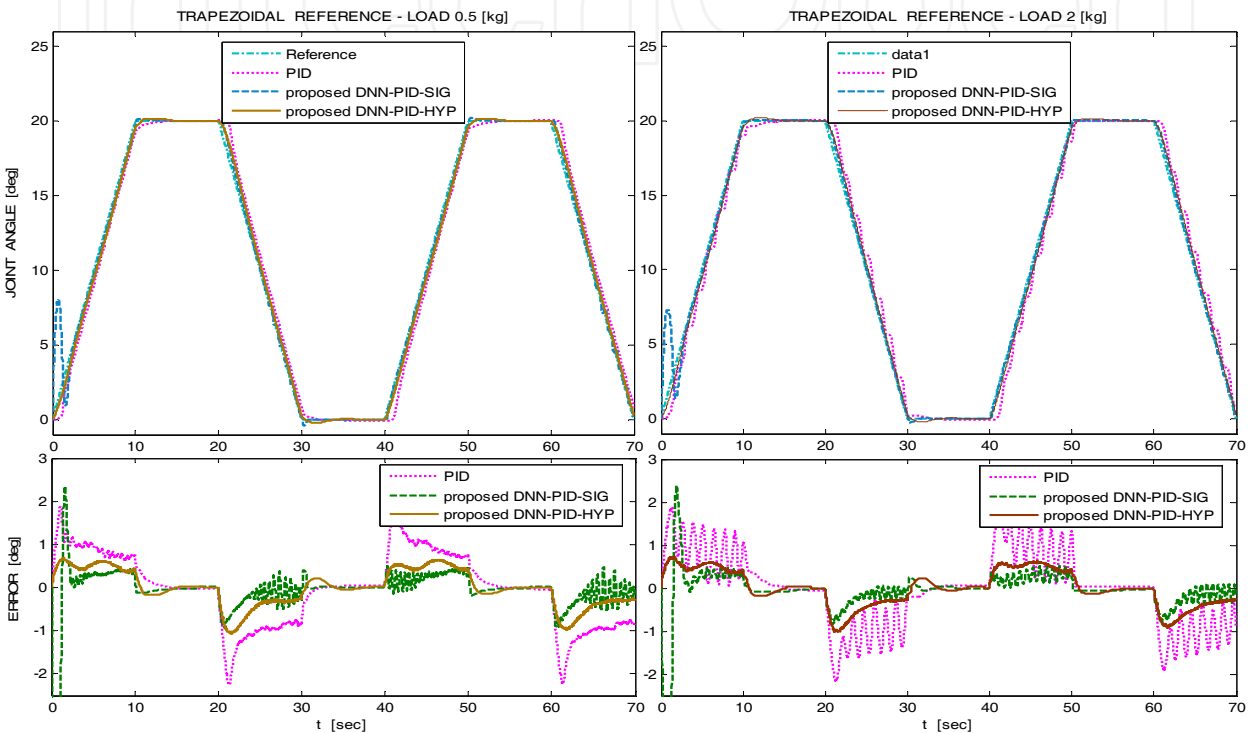
Fig. 9a. Trapezoidal response of both joints of the 2-axes PAM robot arm – Load 0.5[kg] and Load 2[kg].

These figures show that thanks to the refined online tuning of G, Kp, Ki and Kd, the error between desired reference y_{REF} and actual joint angle response y of the PAM manipulator continually optimized. Consequently, the minimized error decreases only in the range $\pm$ 0.5[deg] with both of proposed DNN-PID-SIG and DNN-PID-HYP in case of Load 0.5[kg]. The same good result is also obtained with both of proposed DNN-PID-SIG and DNN-PID-HYP in case of Load 2[kg]. These results are really superioir in comparison with the passive and unchanged error of conventional PID controller ($\pm$ 2[deg] in case of Load 0.5[kg] and up to $\pm$ 2.2[deg] in case of Load 2[kg]). Furthermore, in case of Load 2[kg], Figure 9a shows that PID controller caused the PAM manipulator response being oscillatory and unstable. On the contrary, proposed online tuning DNN-PID controller continues to assert robust control to keep PAM manipulator response stable and accurate tracking.
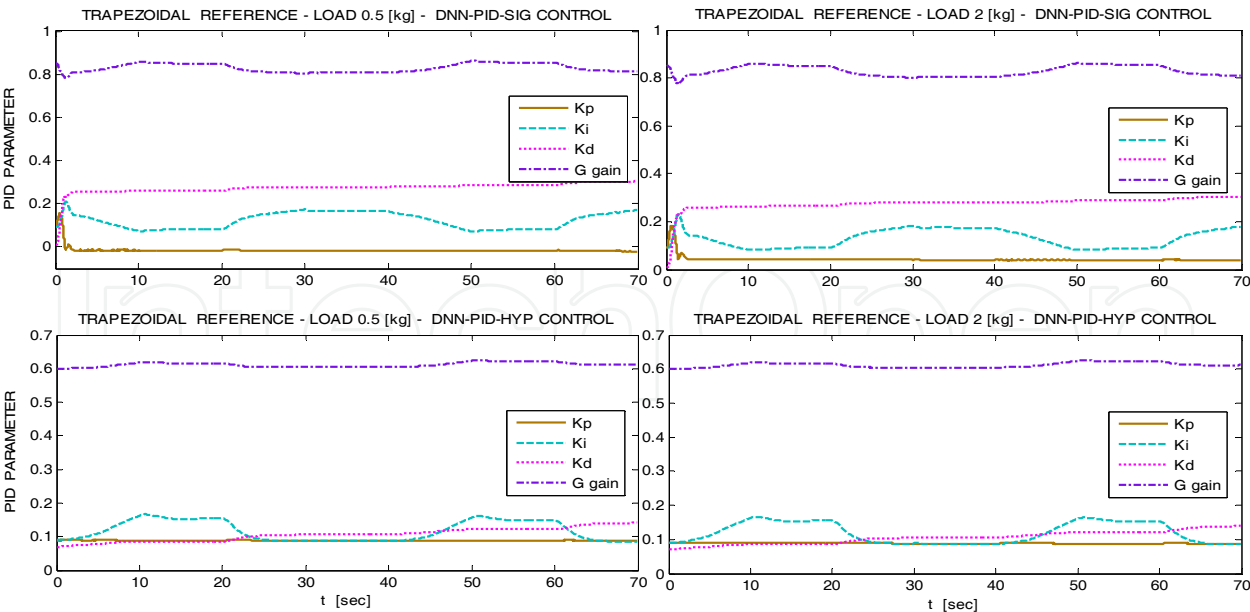
Fig. 9b.The online tuning convergence of proposed DNN-PID controller parameters with triangular reference.

Figure 9c presents the refined shape of control voltage *U* applied to the joint of PAM manipulator, which is generated by the proposed online tuning DNN-PID controller as to assure the performance and the accuracy of the PAM manipulator response. This figure proves that PID controller generates an oscillatory and unstable control voltage in case of Load 2[kg]. On the contrary, proposed online tuning DNN-PID controller continues to robustly control with refined control voltage as to keep PAM manipulator response stable and accurate tracking.
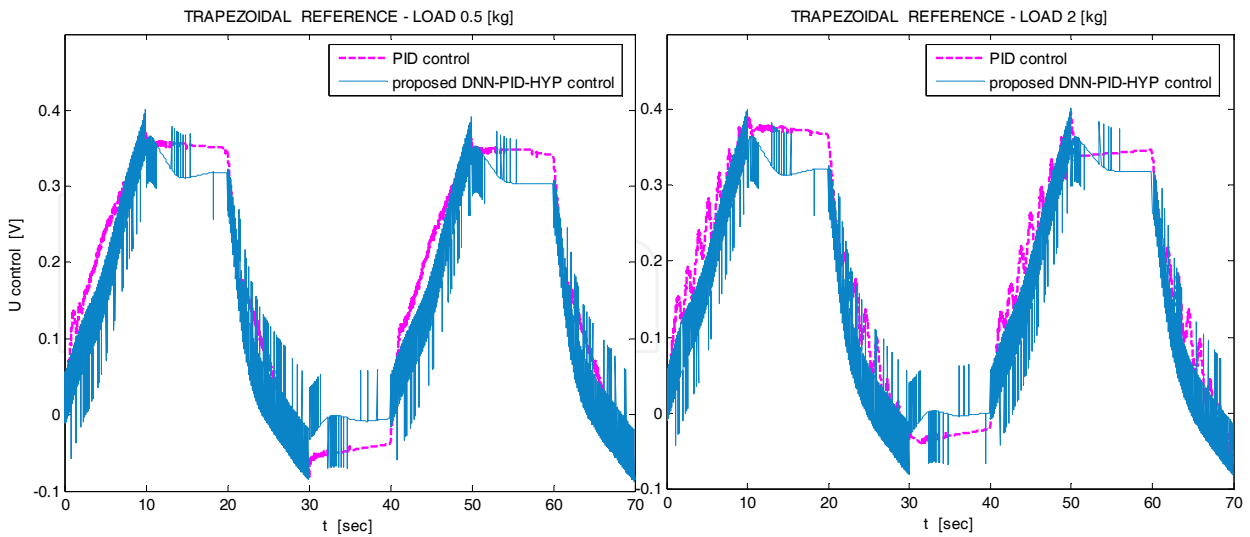
Fig. 9c.The voltage control applied to both joints of the 2-axes PAM robot arm with triangular reference.

Next, the experiments were carried out to verify the effectiveness of the proposed DNN-PID controller using neural network with sinusoidal reference 0.05[Hz]. Fig.10a shows the experimental results in comparison between the conventional PID controller and the two

proposed DNN-PID-SIG and DNN-PID-HYP controllers in 2 cases of Load 0.5[kg] and Load 2[kg] respectively. The online tuning of each control parameter (*G*, *Kp*, *Ki* and *Kd*) in 2 cases of Load 0.5[kg] and Load 2[kg] was shown in Fig. 10b.

These figures show that thanks to the refined online tuning of *G*, *Kp*, *Ki* and *Kd*, the error between desired reference y_{REF} and actual joint angle response *y* of the PAM manipulator continually optimized. Consequently, the minimized error decreases excellently in the range $\pm$ 1[deg] with proposed DNN-PID-HYP and in the range $\pm$ 1.5[deg] with proposed DNN-PID-SIG in case of Load 0.5[kg]. The same good result is also obtained with proposed DNN-PID-SIG and DNN-PID-HYP in case of Load 2[kg]. These results are really superior in comparison with the passive and unchanged error of conventional PID controller ($\pm$ 3[deg] in case of Load 0.5[kg] and up to $\pm$ 4[deg] in case of Load 2[kg]). Furthermore, in case of Load 2[kg], Figure 10a shows that PID controller caused the PAM manipulator response oscillatory and unstable. Otherwise, proposed online tuning DNN-PID controller continues to keep robust control as to maintain PAM manipulator response stable and accurate tracking.

In comparison between proposed DNN-PID-SIG and DNN-PID-HYP, proposed DNN-PID-HYP obtains the excellent robustness and accuracy in comparison with proposed DNN-PID-SIG and thus the proposed DNN-PID-HYP controller is considered to possess the best performance. Furthermore, in initial stage, proposed DNN-PID-SIG possesses again significant overshoot which may cause unstable to PAM manipulator in its initial operation.

Figure 10c depicts the refined control voltage *U* applied to the joint of PAM manipulator, which is generated by the proposed online tuning DNN-PID controller as to assure the performance and the accuracy of the PAM manipulator response.
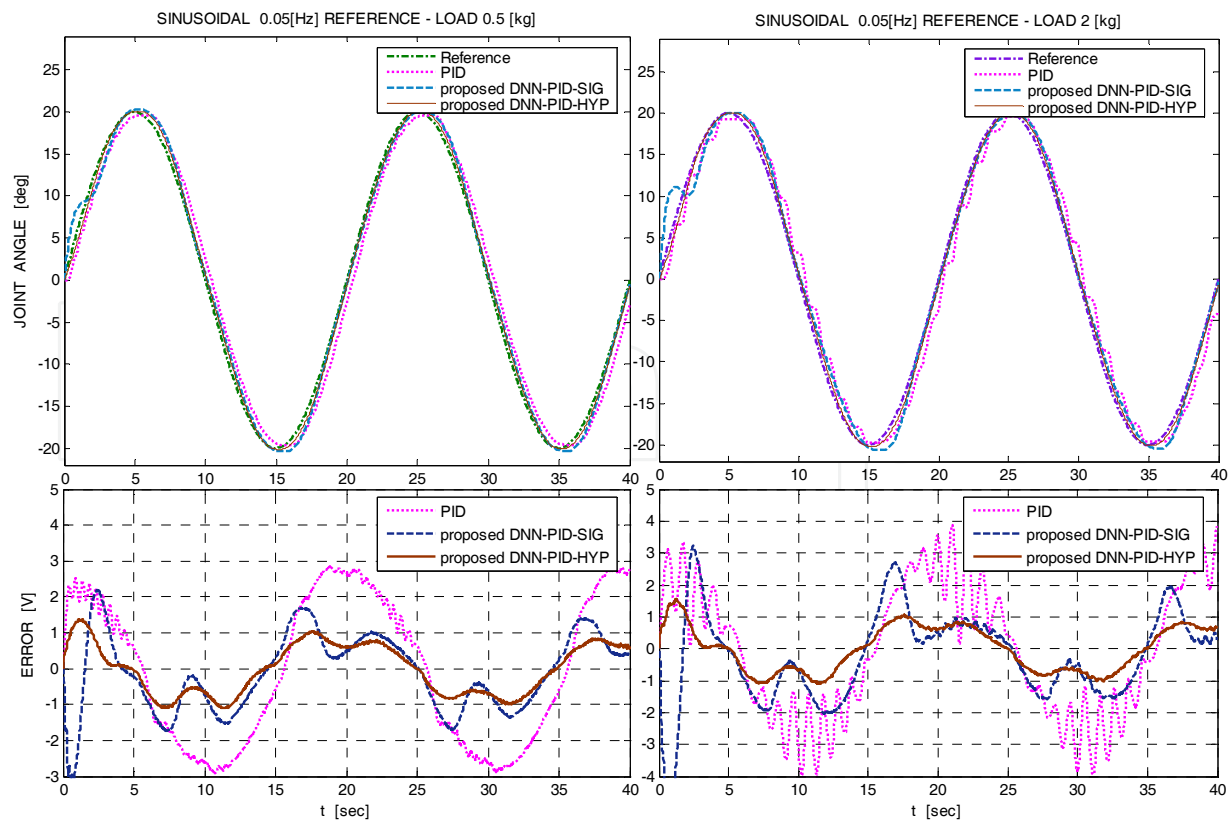

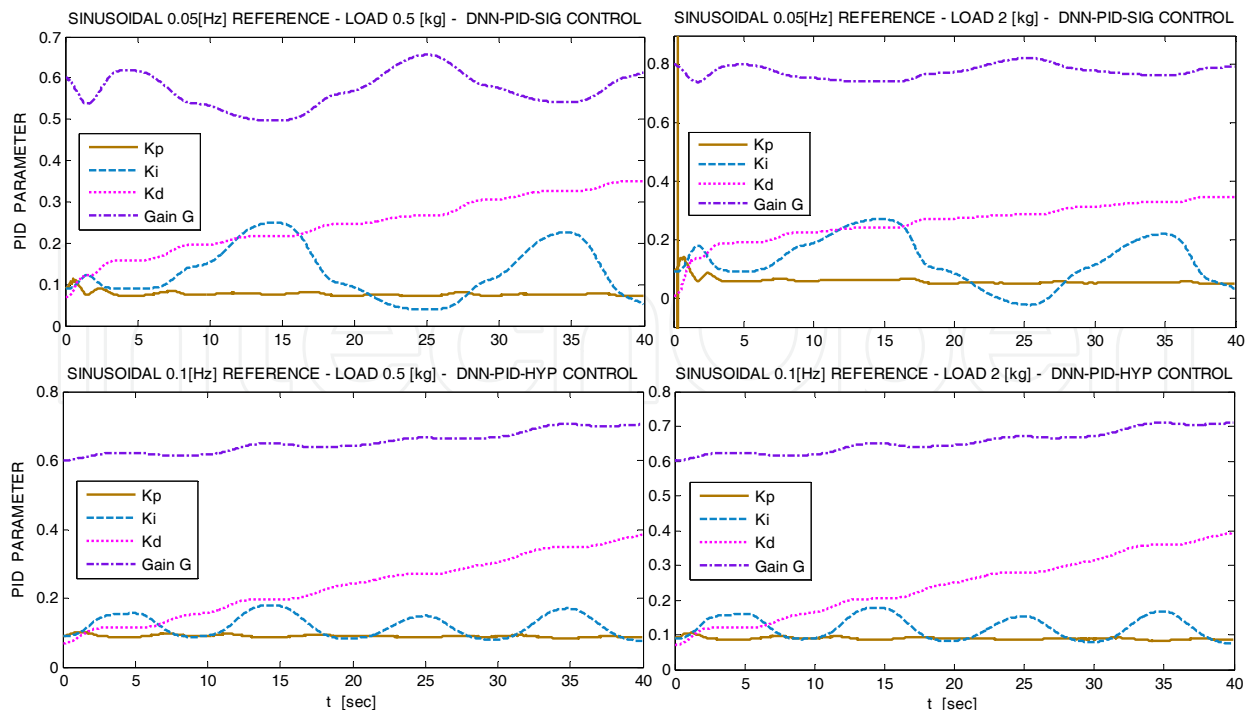Fig. 10a.Sinusoidal response of the PAM robot arm - Load 0.5[kg] and Load 2[kg].

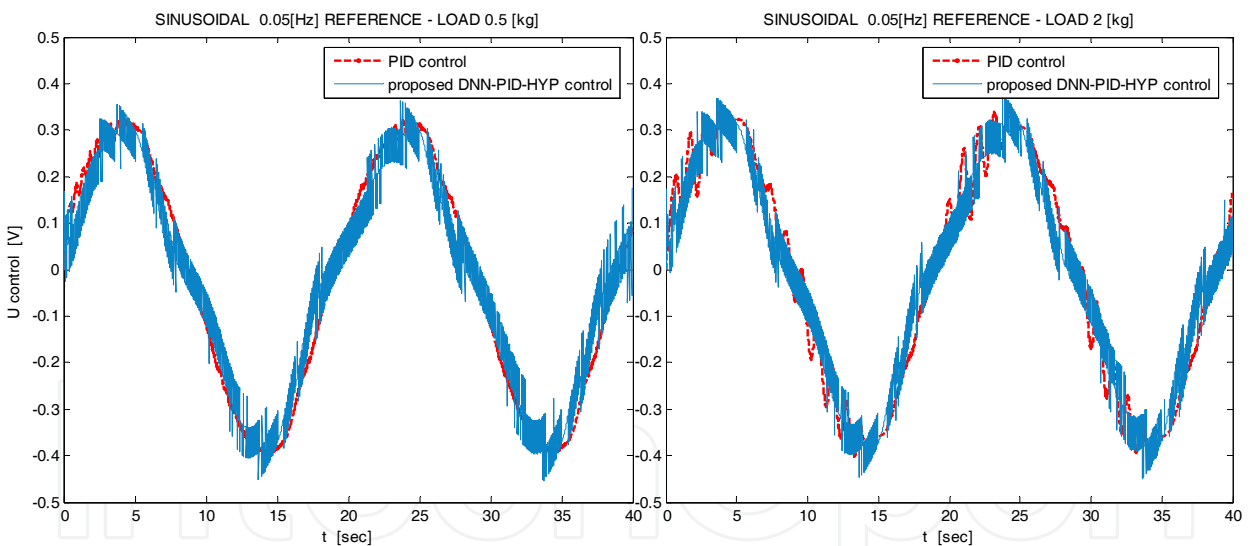Fig. 10b.The online tuning convergence of DNN-PID controller parameters with sinusoidal reference.

Fig. 10c.The voltage control applied to the 2nd joint of the 2-axes PAM robot arm with sinusoidal reference.

Finally, the experiments were carried out with critical sinusoidal reference input 0.2[Hz]. Fig.11a shows the experimental results in comparison between the two proposed DNN-PID-SIG and DNN-PID-HYP controllers in 2 cases of Load 0.5[kg] and Load 2[kg] respectively. The online tuning of each control parameter (G, Kp, Ki and Kd) in 2 cases of Load 0.5[kg] and Load 2[kg] was shown in Fig. 11b. It's important to note that PID controller is impossible to

apply with critical sinusoidal reference input 0.2[Hz] because it caused uncontrollable and unstable as well to the operation of PAM manipulator.

These figures show that thanks to the refined online tuning of *G, Kp, Ki* and *Kd,* the error between desired reference y_{REF} and actual joint angle response *y* of the PAM manipulator continually optimized. Consequently, the minimized error decreases spectacularly in the range ± 1[deg] with proposed DNN-PID-HYP in case of Load 2[kg] and in the range ± 1.5[deg] with proposed DNN-PID-SIG in case of Load 0.5[kg]. In critical sinusoidal reference input 0.2[Hz], proposed online tuning DNN-PID controller continues to keep robust control as to maintain PAM manipulator response stable and accurate tracking.
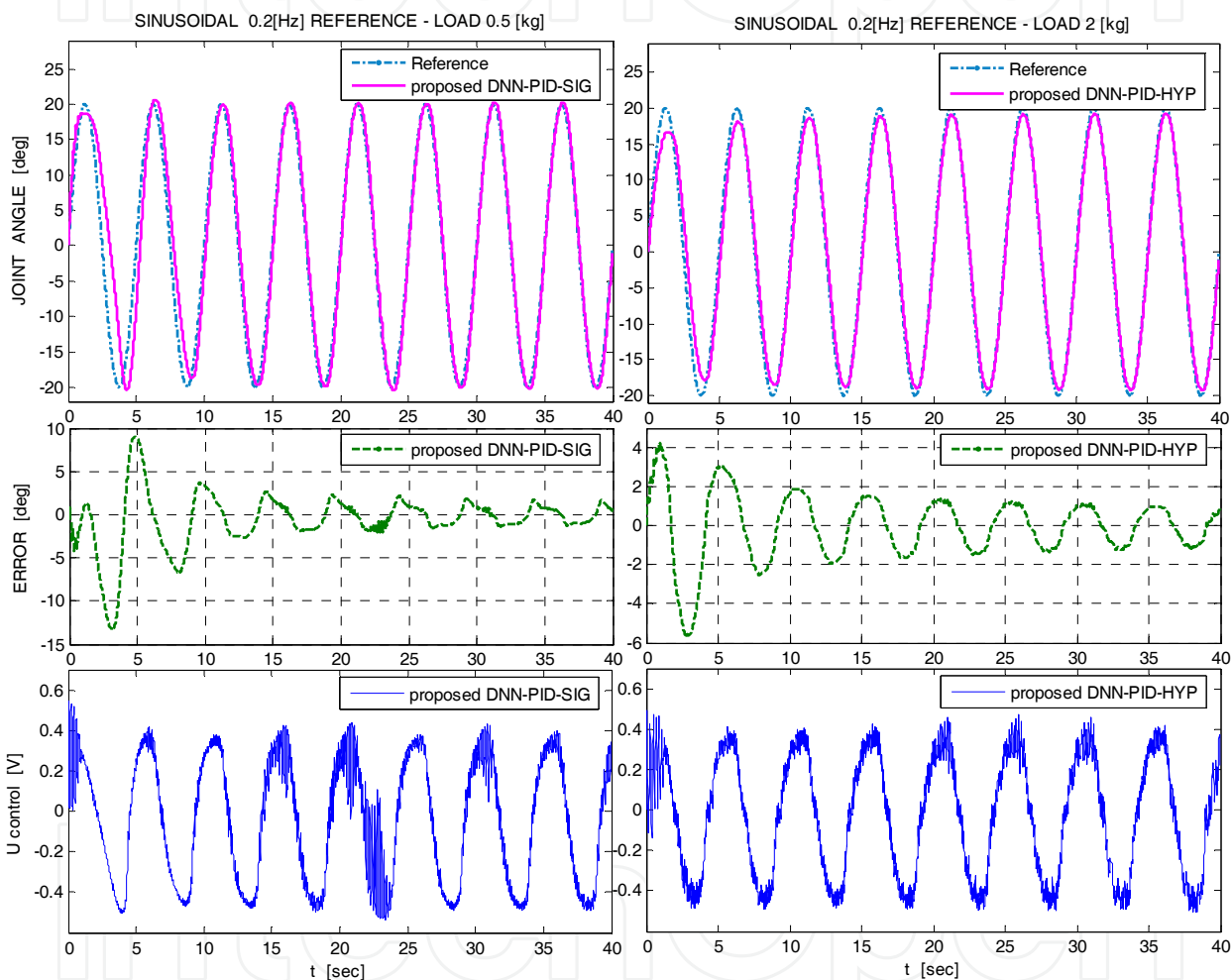

Fig. 11a Sinusoidal 0.2[Hz] response of the PAM manipulator - Load 0.5[kg] and Load 2[kg].

In comparison between proposed DNN-PID-SIG and DNN-PID-HYP, in this case of critical sinusoidal reference input 0.2[Hz], proposed DNN-PID-HYP once more obtains the excellent robustness and accuracy in comparison with proposed DNN-PID-SIG and thus the proposed DNN-PID-HYP controller is considered to possess the best performance between them. Furthermore, in initial stage, proposed DNN-PID-SIG possesses again significant overshoot which may cause unstable to PAM manipulator in its initial operation.
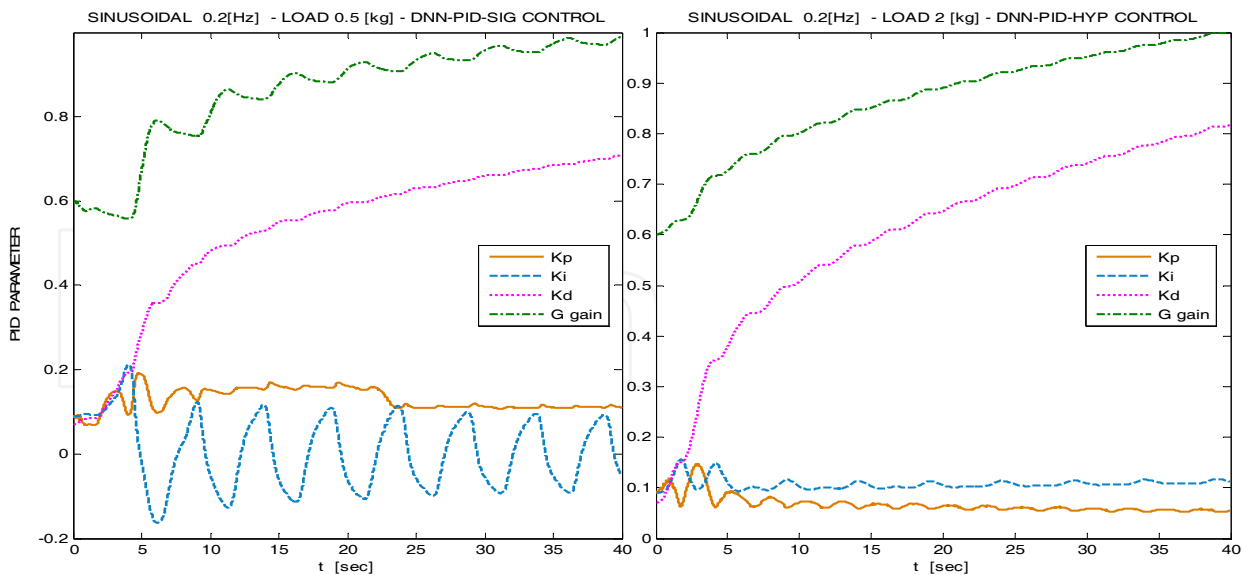
Fig. 11b. The online tuning convergence of proposed DNN-PID controller parameters in case of sinusoidal reference.

In summary, novel DNN-PID controller using neural network was investigated in this paper. It has shown that the proposed method had a good control performance for the highly nonlinear system, such as the PAM manipulator. The controller had an adaptive control capability and the control parameters were optimized via the back propagation algorithm. The controller designed by this method does not need any training procedure in advance, but it uses only the input and output of the plant for the adaptation of proposed control parameters and can tune these parameters online iteratively. From the experiments of the position control of the PAM manipulator, it was verified that the proposed control algorithm presented in this paper was online control with simple structure and had better dynamic property, strong robustness and it was suitable for the control of various plants, including linear and nonlinear process, compared to the conventional PID controller. In comparison between 2 proposed DNN-PID-SIG and DNN-PID-HYP control algorithms, based on experiment results, proposed DNN-PID-HYP control obtains the excellent robustness and accuracy in comparison with proposed DNN-PID-SIG and thus the proposed DNN-PID-HYP controller is considered to possess the better performance than the proposed DNN-PID-SIG one.

4. Conclusions

An innovative online tuning gain scheduling neural DNN-PID Controller suitable for real-time human-friendly industrial applications has been designed, developed and implemented for position control the joint angle of the experimental PAM manipulator in this paper. Experiment results show that the proposed online tuning Gain Scheduling DNN-PID controller is able to learn the nonlinear and dynamic characteristics of the PAM manipulator quickly and thus reduce the tracking error to nearly zero in its operation. The performance of the online tuning gain scheduling DNN-PID controller was found to be very good and robust in the presence of external disturbances. Furthermore, with this proposed online tuning DNN-PID control algorithm, gain scheduling value G and PID parameters K_p,

K_i and K_d can be modified in real time and actual trajectories can be monitored as well. This facilitates testing under different input conditions and ensures future applications of the PAM manipulator as a rehabilitation device for stroke patients. It determines confidently that the proposed online tuning Gain Scheduling DNN-PID controller not only proves its superb performance in control the highly nonlinear PAM manipulator but also would be very efficient in control of other real-time industrial and human-friendly applications.

Acknowledgements

5. References

Ahn K.K., Anh H.P.H., 2006. System modeling and identification of the two-link pneumatic artificial muscle (PAM) manipulator optimized with genetic algorithm. In: Proceedings of the 2006 IEEE-ICASE Int. Conf., Busan, Korea, pp. 356–61.

Ahn K.K., Anh H.P.H., 2007b. A new approach of modeling and identification of the pneumatic artificial muscle (PAM) manipulator based on recurrent neural network. In Proc. IMechE, Part I: Journal of Systems and Control Engineering, 2007, 221(I8), 1101-1122.

Ahn K.K., Anh H.P.H., 2009. Identification of the pneumatic artificial muscle manipulators by MGA-based nonlinear NARX fuzzy model. In MECHATRONICS, 2009, Volume 19, Issue 1, pp. 106-133.

Ahn K.K., Thanh T.D.C., 2005. Nonlinear PID control to improve the control performance of PAM manipulators using neural network. In KSME, Int., Jour., 19(1):pp.106~15.

Balasubramanian K, Rattan K.S., 2003a. Fuzzy logic control of a pneumatic muscle system using a linearizing control scheme. In: Proceedings of Int. Conf., North American Fuzzy Information Processing Society, pp. 432-6.

Chan S.W., Lilly J., Berlin J.E., May 2003. Fuzzy PD+I learning control for a pneumatic muscle. In: Proceedings of IEEE Int. Conf. Fuzzy Systems, St. Louis, MO, pp. 278–83.

Hesselroth T, Sarkar K, Van der Smagt P, Schulten K., 1994. Neural network control of a pneumatic robot arm. IEEE Trans. System Man Cybernetics 24(1): pp.28–38.

Lilly J., Sep. 2003.Adaptive tracking for pneumatic muscle actuators in bicep and tricep configurations. IEEE Trans. Neural Syst. Rehabil. Eng. 11(3):pp.333–9.

Lilly J.H., Chang X., Sep.2003. Tracking control of a pneumatic muscle by an evolutionary fuzzy controller. In IEEE Intell. Automat. Soft Comput., 9(3):pp. 227–44.

Medrano-Cerda G.A., Bowler C.J., Caldwell D.G., Aug. 1995. Adaptive position control of antagonistic pneumatic muscle actuators. In: Proceedings of IEEE Int. Conf. Intelligent Robots and Systems, Pittsburgh, PA, pp. 378–83.

Nelles O., Nonlinear system identification, Springer, 2000.

Repperger D.W., Johnson K.R., Phillips C.A., 1998. VSC position tracking system involving a large scale pneumatic muscle actuator. In: Proceedings of IEEE Conf. Decision Control, Tampa, FL, Dec. pp. 4302–7.

Repperger D.W., Phillips C.A., Krier M., Aug. 1999. Controller design involving gain scheduling for a large scale pneumatic muscle actuator," In: Proceedings of IEEE Conf. Control Applications, Kohala Coast, HI, pp. 285–90.

Reynolds D.B., Repperger D.W., Phillips C.A., Bandry G., 2003. Dynamic characteristics of pneumatic muscle. In IEEE Ann. Biomed. Eng., 31(3):pp.310–7.

Tsagarakis N, Darwin G.C., 2000. Improved modeling and assessment of pneumatic muscle actuators. In: Proceedings of IEEE Int. Conf. Robotics and Automation, San Francisco, CA, pp. 3641–6.
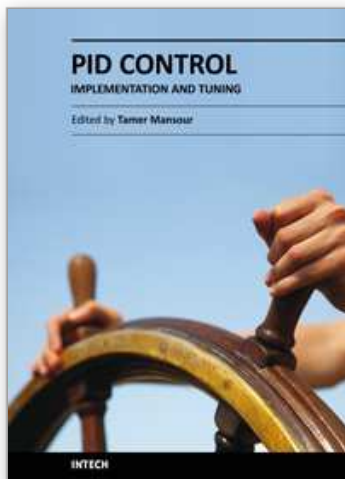
PID Control, Implementation and Tuning

Edited by Dr. Tamer Mansour

The PID controller is considered the most widely used controller. It has numerous applications varying from industrial to home appliances. This book is an outcome of contributions and inspirations from many researchers in the field of PID control. The book consists of two parts; the first is related to the implementation of PID control in various applications whilst the second part concentrates on the tuning of PID control to get best performance. We hope that this book can be a valuable aid for new research in the field of PID control in addition to stimulating the research in the area of PID control toward better utilization in our life.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ho Pham Huy Anh and Nguyen Thanh Nam (2011). A New Approach of the Online Tuning Gain Scheduling Nonlinear PID Controller Using Neural Network, PID Control, Implementation and Tuning, Dr. Tamer Mansour (Ed.), ISBN: 978-953-307-166-4, InTech, Available from: http://www.intechopen.com/books/pid-control-implementation-and-tuning/a-new-approach-of-the-online-tuning-gain-scheduling-nonlinear-pid-controller-using-neural-network

INTECH
open science | open minds