Design and Implementation an Adaptive Recurrent Neural Networks (ARNN) Controller of the Pneumatic Artificial Muscle (PAM) Manipulator

Kyoung Kwan AHN**, Ho Pham Huy ANH***,

** Corresponding Author
Email: kkahn@ulsan.ac.kr
School of Mechanical and Automotive Engineering, University of Ulsan,
San 29, Muger2dong, Nam-gu, Ulsan, 680-764, Korea
TEL: +82-52-222-1404; FAX: +82-52-259-1680.

*** Faculty of Electronics Engineering, HCM City University of Technology, HCM City, Viet Nam
Email: hphanh@hcmut.edu.vn
Abstract

This paper presents the design, development and implementation of an Adaptive Recurrent Neural Networks (ARNN) controller suitable for real-time manipulator control applications. The unique feature of the ARNN controller is that it has dynamic self-organizing structure, fast learning speed, good generalization and flexibility in learning. The proposed adaptive algorithm focuses on fast and efficient optimization by weighting parameters of inverse Recurrent Neural models used in the ARNN controller. This approach is employed to implement the ARNN controller with a view to controlling the joint angle position of the highly nonlinear pneumatic artificial muscle (PAM) manipulator in real-time. The performance of this novel proposed controller was found to be superior compared with a conventional PID controller. These results can be applied to control other highly nonlinear systems as well.

Keywords: pneumatic artificial muscle (PAM), highly nonlinear PAM manipulator, proposed adaptive recurrent neural networks controller (ARNN), real-time joint angle position control, online learning BP algorithm.
NOMENCLATURE:

\( \delta_j \): Search direction value of \( j^{th} \) neuron of hidden layer \((j=[1 \rightarrow q])\)

\( \delta_i \): Search direction value of \( i^{th} \) neuron of output layer \((i=[1 \rightarrow m])\)

\( \lambda \): Learning rate

\( \theta \): Weighting set \((=W)\)

\( E \): Error to be minimized

\( E_N \): Summed error of batch training mode with \( N \) input-output samples

\( F_i \): Activation function of \( i^{th} \) neuron of the output layer

\( f_j \): Activation function of \( j^{th} \) neuron of the hidden layer

\( k \): \( k^{th} \) iterative step of error value calculation

\( K \): Number of steps used to accumulate the error values

\( m \): Number of neurons of output layer

\( N \): Number of input-output training samples

\( n \): Number of neurons of input layer

\( n_a \): Order of output \( y(z^I) \)

\( n_b \): Order of input \( u(z^I) \)

\( n_k \): Time delay (in this paper, \( n_k = T=1 \))

\( q \): Number of neurons of hidden layer

\( r \): Ratio between predicted error and current error

\( O_j \): \( j^{th} \) output from the hidden layer

\( u_l \): \( l^{th} \) input to the input layer

\( W_{ij} \): Weight from the \( j^{th} \) neuron in the hidden layer to the \( i^{th} \) neuron of the output layer

\( w_{lj} \): Weight from the \( l^{th} \) neuron in the input layer to the \( j^{th} \) neuron of the hidden layer

\( y_i \): \( i^{th} \) output from the output layer

\( \hat{y}_i \): \( i^{th} \) predicted output from the output layer

\( z_l \): \( l^{th} \) output from the input layer (in this paper, \( u_l = z_l \))

\( Z^N \): Training set with \( N \) input-output samples
1. INTRODUCTION.

The development of a compliant manipulator able to undertake monotonous and dangerous tasks has motivated many researchers to develop more sophisticated and intelligent controllers for human-friendly industrial manipulators. Due to uncertainties, it is difficult to obtain an accurate mathematical model for robot manipulators. Thus conventional control methodologies find it difficult or impossible to handle un-modeled dynamics of a robot manipulator. To accommodate system uncertainties and variations, learning methods and adaptive techniques must be incorporated. Most conventional control methods, for example PID controllers, are based on mathematical and statistical procedures for modeling the system and estimating of optimal controller parameters. In practice, the manipulator to be controlled is often highly nonlinear and a mathematical model may be difficult to derive. Consequently conventional techniques will not be able to handle modeling errors and suffer from lack of accuracy and robustness as well.

Furthermore, the trend of modern robotics toward applications requiring closer interaction between the robot and the human operator has recently led researchers to develop a novel actuator which shares some features with natural skeletal muscle. The PAM actuator is now increasingly popular because it provides advantages such as high power/weight ratio, hygiene, ease of preservation and capacity for human compliance which is the most important requirement in the medical and human welfare field. Thus PAM actuators have been regarded during recent years as an interesting alternative to hydraulic and electric actuators. However, the air compressibility and the lack of damping ability of the PAM manipulator create a dynamic disturbance in the pressure response and cause oscillatory motion. Therefore, it is not easy to optimize the performance of transient response with high speed and with respect to various external inertia loads in order to create a human-friendly therapy robot. Numerous intelligent control methods have been devised to solve complicated problems of industrial manipulators in general and of PAM manipulators in particular. Neo et al. (1996) in [1] and Lilly et al. (2003) in [2] improved fuzzy controllers for PAM manipulators. A Kohonen-type neural network for position control of a robot arm is applied in [3]. Forwardly the authors developed a feed forward neural network controller in [4]. Caldwell et al. applied an adaptive controller with error better than ±0.5° [5]. Carbonell et al. successfully applied sliding mode to control a PAM actuator [6]. Balasubramanian (2003) applied fuzzy + PID control to a PAM system [7]. Authors improved fuzzy feed forward control to a PAM system [8]. Ahn, Lee and Yang (2003) developed $H_{\infty}$ control for a 6-DOF manipulator [9]. Gini et al. (2003) in [10] proposed an adaptive controller based on applying a neural network to an artificial hand, which is composed of the PAM. Nil et al. (2006) developed a hybrid fuzzy neural
network to control a 3-DOF robot manipulator [11]. Recently (2006, 08), in [12][13], Ahn and Anh have successfully designed a Neural NARX model and GA-based Fuzzy NARX model of the nonlinear PAM manipulator system. Forwardly, in [14], Ahn et al. have applied magneto-rheological (MR) Brake combining with LVQNN for improving the performance of the 1-link and 2-link PAM manipulator.

Though these control systems were partially successful in producing 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. This 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 of industrial use or human-friendly applications. Assuming that the PAM manipulator is applied in an elbow and wrist rehabilitation robot in the future, which is the final purpose of our study, it is necessary to achieve a rapid response, even if the external inertia load changes severely. At the same time, the external inertia loads can always vary and can not be known exactly. Therefore, it is necessary to propose a new control algorithm, which is applicable to a highly nonlinear PAM system with various loads.

To overcome these drawbacks, the proposed Adaptive Recurrent Neural Networks (ARNN) algorithm in this paper is a newly developed algorithm applied to a PAM manipulator system that has a dynamic self-organizing structure, rapid learning speed, good generalization and flexibility in learning. The ARNN control algorithm was initially implemented in the position control of the CNC machine (Lin et al., 2006) [15]. In this paper, the proposed online ARNN controller applied to the PAM manipulator is first employed to compensate for environmental variations such as payload mass and time-varying parameters during the operation process. By virtue of on-line learning, it is able to learn the PAM manipulator dynamics and make control decisions simultaneously. In effect, it offers an exciting on-line estimation scheme for the plant.

The outline of this paper composes section 1 introducing related works in PAM manipulator control. Section 2 introduces the experimental configuration set up of the PAM manipulator system. Section 3 presents a procedure for designing an adaptive recurrent neural network (ARNN) controller for the PAM manipulator. Section 4 presents and analyses experimental studies and results. Finally, the conclusion belongs to section 5.
2. EXPERIMENTAL CONFIGURATION SET UP.

The prototype PAM manipulator used is a two-axis, closed-loop activated with two antagonistic PAM pairs pneumatically controlled through the 2 proportional valves. Each of the 2-axes provides a different motion and contributes one degree of freedom to the PAM manipulator (see Fig. 1).

In this paper, the first joint of the PAM manipulator is fixed and the proposed ARNN control algorithm is applied to control the joint angle position of the second joint of the PAM manipulator.

![Fig. 1. Photograph of the experimental 2-joint PAM manipulator.](image)

![Fig. 2: Block diagram for working principle of the 2nd joint of the 2-axes PAM manipulator.](image)

The experimental system is illustrated in Fig. 2. An 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 a PC computer. First, the initial control voltage value \( U_0(t) = 5\,[V] \) is sent to the proportional valve to inflate the artificial muscles with air pressure at \( P_0 \) (initial pressure) to set the joint’s initial status. Second, by changing the input of the D/A converter, we can 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.

Table 1 presents the configuration of the hardware set-up installed from Fig.1 and Fig.2 to control the second joint of the PAM manipulator using the proposed ARNN control algorithm.

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

Table 1: Lists of experimental hardware.

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 changes the digital signal from the PC to an analog voltage $u(t)$. The rotating torque is generated by the pneumatic pressure difference supplied from an air-compressor between the antagonistic artificial muscles. Consequently, the second joint of the PAM manipulator will be rotated. The joint angle, $\theta$ [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 a joint angle value $y(t)$. The external inertia load can be changed from 0.5[kg] to 10[kg], which is a 2000 (%) change with respect to the minimum inertia load condition. The experiments are conducted under a pressure of 4[bar] and all control software is coded in MATLAB-SIMULINK using C-mex S-function.

3. Design of the Adaptive Recurrent Neural Networks (ARNN) Controller

3.1 Recurrent Neural ARNN Model and Back Propagation (BP) Learning Algorithm

The Inverse ARNN-1(2) model used in this paper is a combination between the Multi-Layer Perceptron Neural Networks (MLPNN) structure and the Auto-Regressive with eXogenous input (ARX) model. Due to this combination, the Inverse ARNN-1(2) model possesses both a powerful universal approximating feature from an MLPNN structure and a strong predictive feature from the ARX model.

A fully connected 3-layer feed-forward MLP-network with $n$ inputs, $q$ hidden units (also called “nodes” or “neurons”), and $m$ output units is shown in Fig. 4.

Fig. 4 Structure of feed-forward MLPNN
In Fig. 4, \( w_{10}, \ldots, w_{q0} \) and \( W_{10}, \ldots, W_{m0} \) are weighting values of bias neurons of the input layer and hidden layer respectively.

The general form of the discrete ARX model in domain \( z \) is

\[
y(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_a} z^{-n_a}} \tag{1}
\]

in which \( n_a \) and \( n_b \) are the order of output \( y(z^{-1}) \) and input \( u(z^{-1}) \) respectively.

The class of MLPNN-networks considered in this paper is furthermore confined to those having only one hidden layer and using sigmoid activation functions:

\[
\hat{y}_i(w, W) = F_i \left( \sum_{j=1}^{q} W_{ij}O_j(w) + W_{i0} \right) = F_i \left( \sum_{j=1}^{q} W_{ij}f_j \left( \sum_{l=1}^{n} w_{jl}z_l + w_{j0} \right) + W_{i0} \right) \tag{2}
\]

The prediction error approach, which is the strategy applied here, is based on the introduction of a measure of closeness in terms of a mean sum of square error (MSSE) criterion:

\[
E_N(\theta, Z^N) = \frac{1}{2N} \sum_{i=1}^{N} \left[ y(t) - \hat{y}(t|\theta) \right]^T \left[ y(t) - \hat{y}(t|\theta) \right] \tag{3}
\]

Based on the conventional error Back-Propagation (BP) training algorithms, the weighting value is calculated as follows:

\[
W(k+1) = W(k) - \lambda \frac{\partial E(W(k))}{\partial W(k)} \tag{4}
\]

with \( k \) as the \( k^{th} \) iterative step of the calculation and \( \lambda \) as the learning rate.

Using the calculation presented in the work of Ahn and Anh [12], the weights \( W_{ij} \) and \( w_{jl} \) of weighting vector \( \theta \) of the ARNN-1(2) are then updated to be

\[
W_{ij}(k+1) = W_{ij}(k) + \Delta W_{ij}(k+1) \\
\Delta W_{ij}(k+1) = \lambda \delta_i O_j \\
\delta_i = \hat{y}_i(1 - \hat{y}_i)(y_i - \hat{y}_i) \tag{5}
\]

where \( \delta_i \) is the search direction value of the \( i^{th} \) neuron of the output layer (\( i = [1 \rightarrow m] \)); \( O_j \) is the output value of the \( j^{th} \) neuron of the hidden layer (\( j = [1 \rightarrow q] \)); \( y_i \) and \( \hat{y}_i \) are real output and predicted output of the \( i^{th} \) neuron of the output layer (\( i = [1 \rightarrow m] \)), and

\[
w_{jl}(k+1) = w_{jl}(k) + \Delta w_{jl}(k+1) \\
\Delta w_{jl}(k+1) = \lambda \delta_j u_i \\
\delta_j = O_j (1 - O_j) \sum_{i=1}^{m} \delta_i W_{ij} \tag{6}
\]
in which $\delta_j$ is the search direction value of the $j^{th}$ neuron of the hidden layer ($j=[1 \rightarrow q]$); $O_j$ is the output value of the $j^{th}$ neuron of the hidden layer ($j=[1 \rightarrow q]$); $u_l$ is the input of the $l^{th}$ neuron of input layer ($l=[1 \rightarrow n]$).

3.2. Adaptive Recurrent Neural Control Scheme

The proposed adaptive recurrent neural ARNN control scheme is depicted in the following presentation (see Fig. 5).

The basic idea of this approach is to learn the PAM manipulator’s inverse characteristics and to use the inverse dynamic model to generate the required control signal $U_{ARNN}$. There are two inverse Adaptive Recurrent Neural Network models (Inverse ARNN-1 and Inverse ARNN-2) used in this control system. These two inverse models have the same MLPNN structure. The Inverse ARNN-2 model is employed for online updating the weights of the PAM manipulator inverse dynamic model whereas the Inverse ARNN-1 model is used to generate an appropriate voltage control signal. The updated weighting values of the Inverse ARNN-1 model are obtained from the weights of the Inverse ARNN-2 model. The proposed control algorithm is based on the equation:

$$U = U_{PID} + U_{ARNN1}$$

where $U$ is the required control voltage, $U_{PID}$ is the compensating control voltage value generated by the PID controller and $U_{ARNN1}$ is the control voltage value generated by the Inverse ARNN-1. The dynamic inverse PAM manipulator model is obtained by the Inverse ARNN-2 model with its weighting
values are online updated by using the error Back Propagation (BP) learning algorithm described in subsection 3.1. In online learning mode, the weights of the Inverse ARNN-2 model are online trained during real-time control of the PAM manipulator’s operation. In this way, the Inverse ARNN-1 model is an intrinsic duplicate copy of the Inverse ARNN-2 model with its weights being online adjusted by using the control signal error \(\text{error}_U(k)=[U(k)-U_{\text{ARNN}2}(k)]\). Meanwhile, the output error \(\text{error}_Y(k)=[y_r(k)-y(k)]\) between the reference trajectory \(y_r(k)\) and the joint angle output \(y(k)\) will be the input of the feedback PID controller. This output error, through PID controller, creates the regulating control value \(U_{\text{PID}}\). The combination of \(U_{\text{ARNN1}}\) and \(U_{\text{PID}}\) in the proposed ARNN control scheme is to improve the accuracy and stabilize the PAM system at the starting time. As a result the parallel-connected feedback PID controller also produces for faster and more accurate tracking performance.

The essence of the weighting training of the proposed ARNN control is to train the Inverse ARNN-2 model so that the mean of the sum squared error (MSSE) \(E=(\text{error}_U)^2/2\) is minimized. The control signal error is \(\text{error}_U(k)=[U(k)-U_{\text{ARNN}2}(k)]\) where \(U\) represents the actual voltage control and \(U_{\text{ARNN2}}\) represents the predictive voltage control value. From these two relations, we have:

\[
E = \frac{1}{2}(U - U_{\text{ARNN}2})^2 \equiv \frac{1}{2}(U_{\text{PID}})^2
\]  

(8)

In order to achieve a fast weight adjustment, the Back-Propagation (BP) method is used to minimize \(E\). The weight is adjusted according to

\[
\Delta W = W(\text{new}) - W(\text{old}) = -\lambda \frac{\partial E}{\partial W}
\]  

(9)

where \(\lambda > 0\) is the learning rate. The learning rate value is chosen by the trial and error method.

Using the chain rule, we have

\[
\frac{\partial E}{\partial W} = \frac{\partial E}{\partial U_{\text{ARNN}2}} \frac{\partial U_{\text{ARNN}2}}{\partial W}
\]  

(10)

It follows from (8) and (10), we have

\[
\frac{\partial E}{\partial U_{\text{ARNN}2}} = -(U - U_{\text{ARNN}2})
\]  

(11)
From equations (9), (10) and (11), we can return to the calculation described in subsection 3.1 to online update the weights of the Inverse ARNN-2 model by using equations (5) and (6).

### 3.3. Simulation results

To evaluate the performance of the proposed ARNN controller, a series of simulation results were performed. These studies were carried out to evaluate the performance of the ARNN controller. They were carried out in three cases. The first case is to evaluate the sinusoidal trajectory tracking capability of the controller without external disturbances. The second evaluation is to test the robustness of the ARNN controller by injecting a disturbance in the circular motion plane. This disturbance is injected by hooking a spring on the PAM manipulator end-effector and pulling it in the motion plane. The third test is to compare the performance of the controller with a conventional Proportional Integral Derivative (PID) controller applied on the same PAM manipulator.

#### a) Without Disturbances.

Fig. 6 depicts sinusoidal trajectory tracking in the absence of external disturbances at a sampling time of 10[ms]. At \( t=0s \), reference trajectories were fed to the ARNN control system. It can be observed that the error of Joint \( T_1 \) increased to a value maximum of \( 20^\circ \) at \( t=0s \). This is partly due to the difference between its initial position and the set point. The main reason for these observations is that the ARNN learning algorithm is still learning the inverse model of the robot manipulator. Therefore, ARNN-2 may not have sufficient number of neurons to compensate for the control signal at the initial stage. When \( t>0.8s \), the ARNN algorithm has learned successfully the inverse model of the PAM manipulator, and fine tune ARNN-2 accordingly. Consequently, the compensated torque \( \tau_{FNN2} \) successfully reduces the trajectory tracking error to a low level of \( \pm 1.958^\circ \) for Joint 2.
Fig. 6: Sinusoidal Trajectory tracking and Angle error for Joint 2.

b) With Disturbances.

Fig. 7 shows the responses when a high external disturbance is injected into the motion plane for a sampling time of 10[ms]. The results obtained show that even when there are external disturbances, the ARNN controller is able to reduce the trajectory tracking error to a low level by modifying quite flexibly value of U control output. Initially, from t=0s up to t=0,8s, the tracking error is high because of the PAM manipulator’s initial position and due to ARNN-1 still in learning process. Consequently, by virtue of the ARNN online learning and weight training, tracking errors for 2\textsuperscript{nd} joint is reduced to a low level very quickly. The compensated torque $U_{\text{ARNN2}}$ successfully reduces the trajectory tracking error to a low level of ±1.98° for Joint T\textsubscript{2} in case sinusoidal tracking and steady state error of ±0.13° for Joint 2 in case ladder trajectory tracking. Fig. 8 shows that there is a significant transient angle error up to 10° at times appearing ladder step (t=5s, t=10s and t=15s). This is because the total output torque from the ARNN controller is saturated corresponding U control value equal 1[V] determined by operator.
Fig. 7: Trajectory tracking and Angle error for Joint 2.

Fig. 8: Ladder Trajectory tracking and Angle error for Joint 2.
c) Without ARNN controller

Fig. 9 demonstrates ladder trajectory tracking capabilities of the control system without the A-RNN controller at a sampling time of 10 ms. The test starts off without the ARNN controller by disabling Inverse ARNN-1 and Inverse ARNN-2 as well, i.e. having only the PID controller controlling the PAM manipulator. As a result, tracking errors of Joint $T_1$ and Joint $T_2$ have a maximum error value of $10^0$ at transient state and of $1,45^0$ at steady state.

When Inverse ARNN-2 was enabled, online weight training will be used to tune the Inverse ARNN-1 and compensate for tracking errors. Subsequently, the PAM manipulator’s joint angle converged to their desired set points quickly once the ARNN-2 was enabled as proved previous results. It is evident from the preceding results that the ARNN control algorithm is able to learn the inverse model quickly and track the reference trajectories accurately. In addition, the ARNN controller is robust in the presence of external disturbances.

Fig. 9: Ladder Trajectory tracking and Angle error for Joint $T_2$ - (PID control only).
4. EXPERIMENT RESULTS.

The performance of the novel proposed ARNN control scheme is verified using joint angle position control of the second Joint of the PAM manipulator. Fig.5 describes the working diagram of this control scheme.

Fig.10 presents the experiment SIMULINK diagram of the proposed ARNN control algorithm which runs in the Real-time Windows Target environment with ARNN-1 and ARNN-2, which are 2 subsystems written in C and compiled in a real-time C-mex file. The PID controller is implemented in parallel with the Inverse ARNN-1 model to regulate the \( errorY(k) \) value and to keep stable the PAM manipulator system during the starting time. Three PID parameters \( K_P \), \( K_I \), \( K_D \) are chosen by the GA-based optimization method and determined to be \( K_P = 0.05; K_I = 0.035 \) and \( K_D = 0.002 \).

In order to extend the range of the investigated pay-load variation, four different conditions of the end-point’s load (0.5[kg], 2[kg], 5[kg] and critical value 10[kg]) were applied. The value 10kg is critical as it represents the dynamic mass of the arm when PAM system is used as a rehabilitation robot. Furthermore, to demonstrate the performance of the proposed ARNN control algorithm, the two different control methods (conventional PID and proposed ARNN control scheme) were applied and compared.
The proposed ARNN control of the PAM manipulator is investigated with fixed configuration as follows. The Inverse PAM manipulator ARNN-1 and ARNN-2 models possess a 3-layer MLPNN structure with 5 neurons in the hidden layer. The sampling-time is 0.01[sec]. The sigmoid is chosen as the activated function of the hidden layer. The linear function is chosen as the activated function of the output layer. The learning rate $\lambda$ of 0.005 is chosen by the trial and error method. The Back Propagation (BP) method is chosen as the fast learning algorithm.

![Graphs showing performance comparison](image)

Fig.11. Output Performance Comparison of proposed ARNN Control (with Trapezoidal Reference)
(a)-Load 0.5 [kg]; (b)-Load 2 [kg]; (c) Load 10 [kg].

Fig.11 presents the performance of the proposed ARNN control algorithm with the trapezoidal reference for three different load cases (0.5[kg], 2[kg] and critical 10[kg]). Since the Inverse ARNN-2 is able to online adaptively auto-update the weighting values, the proposed ARNN controller makes sure the Inverse ARNN-1 model fits well with the varied dynamic nonlinear characteristics of the
second joint of the PAM manipulator over the full range of the load operation. Consequently, the error in the tracking output is very small and limited to $< \pm 0.5$[deg] for a load of 0.5[kg] and $< \pm 2$[deg] for a load of 10[kg]. These results prove that the proposed ARNN control scheme is superior to the PID controller which possesses error varying from $> \pm 2$[deg] for a load of 0.5[kg] and up to $> \pm 5$[deg] for a load of 10[kg]. Furthermore, Fig.11 also shows that for the case of a load 10[kg], the PID controller causes the PAM system to be unstable and oscillatory.

Fig.12 describes the excellent combination of the neural ARNN control scheme and the feedback PID regulator with three different load cases (0.5[kg], 2[kg] and critical 10[kg]). The Inverse ARNN-1 generates $U_{ARNN1}$ which fits well with the change of the reference trajectory and to the varied dynamic characteristics of the PAM system. Meanwhile, the feedback PID regulator of the proposed ARNN control scheme generates the refined regulating value $U_{PID}$ used to ameliorate the accuracy and to stabilize the PAM system. $U_{PID}$ also plays a role in providing the controlled error value ($errorU(k)$) which is minimized by the Inverse ARNN-2 model. This feature helps to update online the weights of the Inverse ARNN-1 during control process. On the contrary the conventional PID controller in Fig.12 generates the control voltage value $U_{control}$ which causes the phase-lag error and thus can not adapt to the varied dynamic characteristics of the PAM system.

The same good results are shown in Fig.13 and Fig.14 which presents the performance of the proposed ARNN control scheme for the triangular reference trajectory. Due to the BP online learning algorithm, the Inverse ARNN-2 model is able to update its weights, which ensures the Inverse ARNN-1 model fits well with the varied dynamic nonlinear characteristics of the second joint of the PAM manipulator during its operation. Consequently, the error in the tracking output is very well within the
limit $\leq \pm 1[\text{deg}]$ for a load of $0.5[\text{kg}]$ and only up to $\pm 2[\text{deg}]$ for a load of $10[\text{kg}]$. These results prove that the proposed ARNN control scheme is much better than the PID controller which causes the error to vary from $>\pm 1.5[\text{deg}]$ for a load of $0.5[\text{kg}]$ and up to $>\pm 2[\text{deg}]$ for a load of $2[\text{kg}]$. For a load of $10[\text{kg}]$, the PID controller can not maintain the operation of the PAM manipulator system and this causes breakdown of the PAM system’s operation.

Fig.13 describes the excellent combination of the ARNN control scheme and the feedback PID regulator for three different load cases ($0.5[\text{kg}]$, $2[\text{kg}]$ and $10[\text{kg}]$). The Inverse ARNN-1 generates the control voltage value $U_{\text{ARNN1}}$ which fits well with the change in the triangular reference trajectory and with the varied dynamic characteristics of the PAM system. Meanwhile, the feedback PID regulator of the proposed ARNN control scheme generates the small regulating value $U_{\text{PID}}$ used to ameliorate the accuracy and to stabilize the PAM system. In addition, the variant shapes of $U_{\text{control}}= U+U_0 = (U_{\text{PID}} + \text{Load } 0.5[\text{kg}])$.}

Fig.14 describes the excellent combination of the ARNN control scheme and the feedback PID regulator for three different load cases ($0.5[\text{kg}]$, $2[\text{kg}]$ and $10[\text{kg}]$). The Inverse ARNN-1 generates the control voltage value $U_{\text{ARNN1}}$ which fits well with the change in the triangular reference trajectory and with the varied dynamic characteristics of the PAM system. Meanwhile, the feedback PID regulator of the proposed ARNN control scheme generates the small regulating value $U_{\text{PID}}$ used to ameliorate the accuracy and to stabilize the PAM system. In addition, the variant shapes of $U_{\text{control}}= U+U_0 = (U_{\text{PID}} + \text{Load } 0.5[\text{kg}])$.}

Fig.13.Performance Comparison of proposed PAM Manipulator ARNN Control with Ramp reference
(a)-Load 0.5 [kg]; (b)-Load 2 [kg]; (c) critical Load 10 [kg].
$U_{\text{ARNN}} + 5/V$ presented in Fig. 14 demonstrate the performance of the proposed ARNN control corresponding to the three load cases (0.5[kg], 2[kg] and 10[kg]).

Fig.14.Performance Comparison of proposed PAM Manipulator ARNN Control with Ramp reference (a)-Load 0.5 [kg]; (b)-Load 2 [kg]; (c) Load 10 [kg].

Fig.15.Performance Comparison of proposed ARNN Control with Sinusoidal Reference (a)-Load 0.5 [kg]; (b)-Load 2 [kg].
Fig. 15 presents the performance of the proposed ARNN control algorithm with the sinusoidal reference trajectory for two different load cases (0.5[kg] and 2[kg]). The tracking output error is kept minimized to be $<\pm 1.5[\text{deg}]$ for a load of 0.5[kg] and only up to $<\pm 2[\text{deg}]$ for a load of 2[kg]. These results are better than the results of the PID controller which causes the error to vary from $>\pm 2[\text{deg}]$ for a load of 0.5[kg] and up to $>\pm 4[\text{deg}]$ for a load of 2[kg] respectively. Furthermore, the shape of the weighting value presented in Fig. 15 once more demonstrates the performance of the proposed ARNN control algorithm which ensures to online update the weights of the Inverse ARNN-1 model in order to adapt to the variation of the load and the reference trajectory.

Fig. 16 shows the excellent combination of the proposed ARNN control with the regulating PID controller for two different load cases (0.5[kg] and 2[kg]). The Inverse ARNN-1 model generates the control voltage value $U_{\text{ARNN1}}$ which fits well with the change of reference trajectory and with the varied dynamic characteristics of the PAM system. Meanwhile, the feedback PID regulator of the proposed ARNN control scheme generates the fast adapted value $U_{\text{PID}}$ which is used to regulate the tracking output and also to control the error value. The error is minimized by the Inverse ARNN-2 model which updates the weights of Inverse ARNN-1 during the control process. In addition, the shape of $U_{\text{control}} = U + U_0 = (U_{\text{PID}} + U_{\text{ARNN1}}) + 5[V]$ is presented in Fig. 16 to demonstrate the performance of the proposed ARNN control corresponding to the two load cases (0.5[kg], 2[kg]).
Fig.17 and Fig.18 show the excellent performance of the proposed ARNN control algorithm with the sinusoidal reference corresponding to the three different frequencies (0.05[Hz], 0.1[Hz] and critical 0.2[Hz]).

Due to the BP online fast learning algorithm, the Inverse ARNN-2 model is able to online update the weights, which ensures that the Inverse ARNN-1 model fits well with the rapidly varying dynamic characteristics of the second joint of the PAM manipulator during its operation. Consequently, the tracking output error proves minimal and is limited within the range ±1[deg] for a sinusoidal frequency of 0.05[Hz] and up to ±5[deg] for a critical sinusoidal frequency of 0.2[Hz]. The shape of the weighting value presented in Fig.17 demonstrates the performance of the proposed ARNN control algorithm.

Fig.18 shows that rapid variation in the sinusoidal reference trajectory causes permanent critical error. As a result, the feedback PID regulator of the proposed ARNN control scheme always generates an important rapidly adapted value $U_{PID}$ which is used to compensate for stabilizing the PAM system and to provide the error value $errorU(k)$ which is permanently minimized by the Inverse ARNN-2.
Consequently, the proposed ARNN control algorithm maintains the robustness of the PAM system so that it follows the critical sinusoidal trajectory 0.2[Hz].

Fig.18. Performance Comparison of proposed ARNN Control (with Sinusoidal Reference, Load 0.5 [kg])
(a)- 0.05 [Hz]; (b)- 0.1 [Hz]; (c)- 0.2 [Hz].

Fig.19. Performance Comparison of proposed ARNN Control (with Triangular & Trapezoidal Reference)
(a)- Load 0.5 [kg]; (b)- Load 2 [kg]; (c)- Load 10 [kg].

Fig.19 presents the performance of both of the Inverse ARNN-1 and Inverse ARNN-2 models in online weight-updated operation which is tested in both the trapezoidal and triangular reference trajectories. These results show that $U_{\text{PID}}$ is close to the error value $errorU(k)$ which is online.
minimized by the BP fast learning algorithm. Consequently, the Inverse ARNN-1 model corresponds with the time-varying dynamic characteristics of the nonlinear PAM manipulator and maintains the robustness of the PAM system to follow the desired trajectory corresponding to a critically varying payload.

5. CONCLUSIONS

An Adaptive Recurrent Neural Networks (ARNN) controller suitable for real-time human-friendly industrial applications has been designed, developed and implemented for position control of the joint angle of the experimental PAM manipulator in this paper. Experimental results show that the proposed ARNN controller is able to learn and to update the inverse dynamics of the PAM manipulator and to reduce the tracking error to nearly zero in load-varying operation. The performance of the ARNN controller was found to be very good and robust in the presence of external disturbances. With this proposed ARNN control algorithm, the parameters of the neural ARNN model 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 for the PAM manipulator as a rehabilitation device for stroke patients. It determines confidently that the proposed ARNN controller not only proves its good performance in control the highly nonlinear PAM manipulator but also would be very efficient in controlling other real-time industrial and human-friendly applications.

ACKNOWLEDGEMENTS

This research was supported by Brain Korea 21 (BK21), Republic of Korea.

REFERENCES


