IJCSET |September 2012 | Vol 2, Issue 9, 1427-1433 www.ijcset.net | ISSN:2231-0711

Design and Implementation of PSO-PID Controller for MA2000 Robotic Manipulator

Firas Abdullah Thweny Al-Saedi[1], Ali H. Mohammed[2]

[1,2]*Computer Engineering Department,*
Al-Nahrain University,
Baghdad, Iraq
[1]firas_alsaidi@yahoo.com
[2]ali.alyasiri@gmail.com

***Abstract* - In this paper, a complete control system is proposed to control the TQ MA2000 which is a six degree of freedom (DOF) complex structure robotic manipulator arm. The kinematics of the MA2000 manipulator were derived and verified with the aid of Matlab. The implemented control scheme is a networked control system (NCS), where two PCs interacted via a computer network to form the overall control and tuning entities. The control system utilized the proportional-Integral-Derivative (PID) algorithm to control the MA2000 manipulator, where each joint was treated as a separate Single Input Single Output (SISO). Particle Swarm Optimization (PSO) algorithm was used to tune the individual PID controllers of the robotic arm. The tuning process was considered as a nonlinear optimization problem, where a certain fitness function was minimized in order to obtain the desirable transient response for each joint. Finally, a set of tests were conducted against the tuned controllers where satisfactory results were obtained.**

Keywords* - PID, PSO, Robotic Manipulator, *MA2000

I. INTRODUCTION

Robotics has profound cultural roots. Over the course of centuries, human beings have constantly attempted to seek substitutes that would be able to mimic their behavior in the various instances of interaction with the surrounding environment [1]. Robotic manipulation, by definition, implies that parts and tools will be moved around in space by some sort of mechanism. This naturally leads to a need for representing positions and orientations of parts, of tools, and of the mechanism itself. To define and manipulate mathematical quantities that represent position and orientation, coordinate systems must be defined and conventions for representation developed [2]. The mechanical structure of a robot manipulator consists of a sequence of rigid bodies (links) interconnected by means of articulations (joints), a manipulator is characterized by an arm that ensures mobility, a wrist that confers dexterity, and an end-effector that performs the task required of the robot [1].

A robot manipulator should be viewed as more than just a series of mechanical linkages. The mechanical arm is just one component in an overall robotic system which consists of the arm, external power source, end-of-arm tooling, external and internal sensors, computer interface, and control computer. Even the programmed software should be considered as an integral part of the overall system, since the manner in which the robot is programmed and controlled can have a major impact on its performance and subsequent range of applications [3].

In manipulator position control (or trajectory control) the controller determines the inputs to the joint actuators so that the end effector follows the desired trajectory as close as possible. This kind of controller has often been used for general industrial manipulators. Generally, changes in dynamics are due to changes in the manipulator configuration, there is also interaction among the joints [4]. In this approach the manipulator is considered as composed of *n* independent systems, (n joint drives). Each joint is controlled as a Single Input Single Output (SISO) system and the coupling effects among the joints, due to configuration and motion, are considered as disturbances [5].

The PID controller is by far the most commonly used controller strategy in the process control industry [6]. Its widespread use is attributed to its simple structure and robust performance over a wide range of operating conditions. The popularity and widespread use of PID control in the process control industry necessitates a detailed discussion on the fundamental theory that underpins this type of three-term process control.

Particle swarm optimization (PSO) is a stochastic optimization approach, modeled on the social behavior of bird flocks. PSO is a population-based search procedure where the individuals, referred to as particles, are grouped into a swarm. Each particle in the swarm represents a candidate solution to the optimization problem.

The goal of PID controller tuning is to determine parameters that meet closed loop system performance specifications, and the robust performance of the control loop over a wide range of operating conditions should also be ensured. Practically, it is often difficult to simultaneously achieve all of these desirable qualities.

To cope with the dynamical changes the robot manipulators suffer from, many controllers have been proposed by many researchers, some are listed hereafter:

Kim and Jeon [7] presented an Field Programmable Gate Arrays (FPGA) implementation of nonlinear PID controllers for humanoid robot arms. The nonlinear functions as well as the conventional PID control algorithm were implemented by the hardware description language. Trigonometric and exponential functions are designed on an FPGA chip.

Yuxin and Peter [8] proposed an approach to establish the global asymptotic stability of the controlled system (robot manipulator) by using Lyapunov direct method and LaSalle's invariance principle. Certain conditions were provided on the regulator gains to ensure global asymptotic stability. The proposed controller does not utilize the modeling information in the control formulation, and thus permits easy implementation. The conducted simulations on a 2 degree of freedom (DOF) robot gave satisfactory results.

Ravari and Taghirad [9] introduced a hybrid fuzzy-PID controller based on learning automata, the goal was the optimal tracking of robot systems including motor dynamics. In the proposed controller, the learning automata is used at the supervisory level for adjustment of the parameters of hybrid fuzzy-PID controller during the system operation. The proposed controller was tested using simulation on PUMA560 manipulator which gave satisfactory results.

Due to the PID algorithm ease of implementation and robustness, many approaches introduced to tune its three parameters in order to enhance the closed loop performance.

Gaing [10] presented a design method for determining the optimal PID controller parameters of an Automatic Voltage Regulator (AVR) system using the PSO algorithm. The author defined a new time-domain performance criterion function in order to assist estimating the performance of the proposed PSO-PID controller.

Wang and Peng [11] presented a way to tune the PID controller gains. The relay feedback approach used first to identify the approximate PID parameters then PSO algorithm is used to refine the acquired parameters. In order to prevent the PSO from falling into a local minima, the search process was divided into two stages by limiting the derivative gain to small range then enlarging it. The simulations were carried out using an arbitrary process where acceptable results were obtained.

Marzoughi and Selamat [12] used PSO algorithm in optimizing the PID controller parameters for the exhaust temperature control of a gas turbine system. The obtained results were compared with classical tuning methods such as Ziegler-Nichols and Cohen-Coon, where the results were more effective and acted well against disturbances.

Dastranj and Moghaddas [13] applied the PSO to tune a PID controller for the inverted pendulum. Fitness function that takes the overshoot term and settling time as arguments was incorporated, which gave a satisfactory results.

In this paper, the PSO algorithm introduced by Kennedy and Eberhart [14] is used to optimize the coefficients of the PID controllers of the MA2000 manipulator joints to overcome the main shortcoming of the PID controller, the lack of efficient tuning method to tune the controller parameters [6].

The range of movement of the major joints (waist, shoulder and elbow) is 270 degrees and 180 degrees on the minor joints. The operational range is depicted in Figure (1).

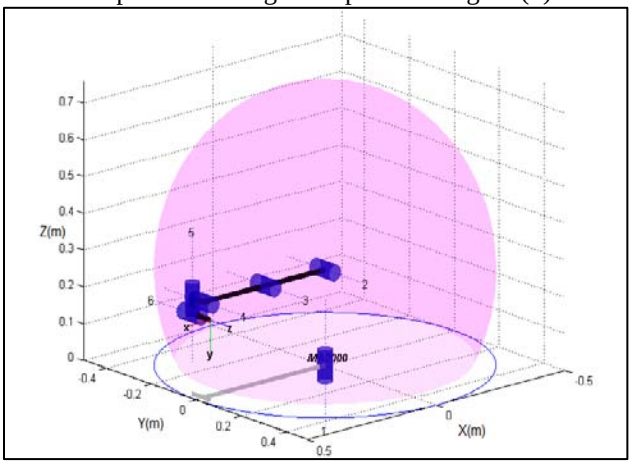

Fig. 1 MA2000 operational range

The main objective of this paper is to develop a tuning methodology for PID controller parameters in controlling the MA2000 robotic manipulator so that the closed-loop system will be able to cope with manipulator configuration changes and external disturbances.

II. PID CONTROL

PID controller is a three-term controller that has a long history in the automatic control field, starting from the beginning of the last century. Owing to its intuitiveness and its relative simplicity, in addition to satisfactory performance which it is able to provide with a wide range of processes, it has become in practice the standard controller in industrial settings. It has been evolving along with the progress of the technology and nowadays it is very often implemented in digital form rather than with pneumatic or electrical components [15].

The basic structure of conventional feedback control system depicted as Figure (2), Where P is the process, C is the controller, F is a feed forward filter, r is the reference signal, $e = r - y$ is the control error, u is the manipulated (control) variable, y is the process (controlled) variable, d is a load disturbance signal and n is a measurement noise signal.

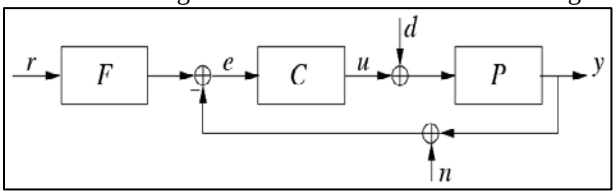

Fig. 2 Conventional feedback system

As the main tools of this structure, PID controller calculates an error value as the difference between a measured process variable and a desired set-point [16].

The PID controller involves three separate parameters: the proportional, the integral, the derivative, denoted as K_p, K_i, and K_d, respectively. Based on current rate of change, these values can be interpreted in terms of time. K_p depends

on the present error, K_i on the accumulation of past errors, and K_d is a prediction of future errors. The weighted sum of these three actions is used to minimize the error by adjusting the process control inputs.

The mathematical formula for calculating the continuous-time PID controller output is [17].

$$u = K_p \left(e + K_i \int_0^t e \, d\tau + K_d \frac{de}{d\tau} \right) \quad (1)$$

The discrete-time implementation for digital systems, computers and microprocessors can be realized by using the following equations [18]:

$$u_p(k) = K_p(e(k)) \quad (2)$$

$$u_i(k) = u_i(k-1) + K_i(e(k))\Delta\tau \quad (3)$$

$$u_d(k) = K_d \frac{e(k) - e(k-1)}{\Delta\tau} \quad (4)$$

$$u(k) = u_p(k) + u_i(k) + u_d(k) \quad (5)$$

Where k is the current sample, $u_p(k)$, $u_i(k)$ and $u_d(k)$ are the proportional part, the integral part and the derivative part respectively. $u(k)$is the final controller output .

The Laplace transfer function of the PID controller $C(s)$ can be shown as:

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (6)$$

Where s is the Laplace variable.

In general, there are three kinds of disturbances acting in the real process control [6]: set-point changes *r*, load disturbance *d*, and measurement noise *n*. These disturbances depicted in Figure (2).

III. PID CONTROLLER TUNING

Controller tuning can be defined as an optimization process that involves a performance criterion related to the form of controller response and to the error between the process variable and the set point [19]. The selection of the PID parameters is obviously the crucial issue in the overall controller design. This operation should be performed in accordance to the control specifications. Usually, the PID parameters are related either to the set-point following or to the load disturbance rejection task, but in some cases both of them are of primary importance. The control effort is also generally of main concern as it is related to the final cost of the product and to the wear and life-span of the actuator. It should be therefore kept at a minimum level. Further the robustness issue has to be taken into account [15].

In the 1940s, Ziegler and Nichols developed two methods for controller tuning based on simple characterization of process dynamics in the time and frequency domains. The time domain method is based on a measurement of part of the open loop unit step response of the process. The step response is measured by applying a unit step input to the process and recording the response. The Ziegler–Nichols methods had a huge impact when they were introduced in the 1940s. The rules were simple to use and gave initial conditions for manual tuning. The ideas were adopted by manufacturers of controllers for routine use. The Ziegler–Nichols tuning rules unfortunately have two severe drawbacks: too little process information is used, and the closed loop systems that are obtained lack robustness [20].

In 1953, Cohen and Coon developed a set of controller tuning recommendations that correct for one deficiency in the Ziegler–Nichols open-loop rules. This deficiency is the sluggish closed-loop response given by the Ziegler–Nichols rules on the relatively rare occasion when process dead time is large relative to the dominant open-loop time constant [19].

Such classical tuning methods are applicable only to linear systems. Many attempts have been made to metaheuristics optimization techniques in controller tuning, especially when the process in nonlinear and varying with time where these techniques proved to be a good tuning choice. The most popular algorithms used in controller tuning are Genetic Algorithms (GA) and PSO.

IV. PARTICLE SWARM OPTIMIZATION

The PSO algorithm is a stochastic population-based search algorithm based on the simulation of the social behavior of birds within a flock. The initial intent of the particle swarm concept was to graphically simulate the graceful and unpredictable choreography of a bird flock, with the aim of discovering patterns that govern the ability of birds to fly synchronously, and to suddenly change direction with a regrouping in an optimal formation. From this initial objective, the concept evolved into a simple and efficient optimization algorithm [21]. A general flowchart of the PSO algorithm is shown in Figure (3).

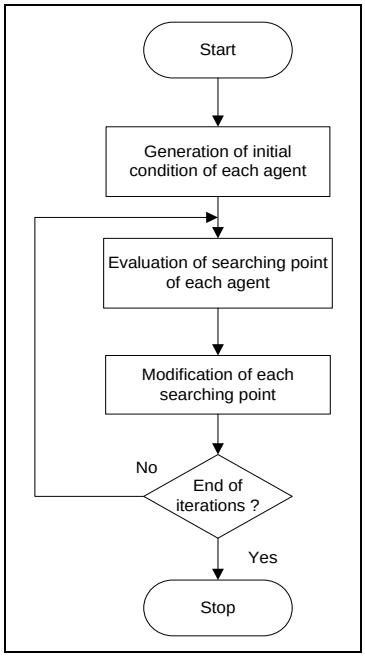

Fig. 3 General PSO flowchart

According to the above background of PSO, Kennedy and Eberhart [14] developed PSO through simulation of bird

flocking in a two-dimensional space. The position of each agent is represented by its x, y axis position and also its velocity is expressed by vx (the velocity of x axis) and vy (the velocity of y axis). Modification of the agent position is realized by the position and velocity information. Bird flocking optimizes a certain objective function. Each agent knows its best value so far ($pbest$) and its x, y position. This information is an analogy of the personal experiences of each agent. Moreover, each agent knows the best value so far in the group ($gbest$) among ($pbests$).

Velocity of each agent can be modified by the following equations:

$$v_i^{k+1} = \chi[v_i^k + c_1 rand_1 \times (pbest_i - s_i^k) + c_2 rand_2 \times (gbest - s_i^k)] \quad (7)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (8)$$

Where

$$\chi = \frac{2k}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \ , \ \varphi = c_1 + c_2 > 4 \ ,$$

v_i^k is velocity of agent *i* at iteration *k*, χ is the constriction factor, c_j is weighting coefficients, rand is random number between 0 and 1, s_i^k is current position of agent *i* at iteration *k*, $pbest_i$ is the local best position of agent *i*, and $gbest$ is the global best position of the group [22].

V. PSO TUNED PID CONTROLLERS

To tune PID controllers with PSO algorithm, The PSO algorithm is mainly utilized to determine three optimal controller parameters: K_p, K_i, and K_d, such that the controlled system could obtain a desired step response output [23]. The function of optimization problem can be viewed as a 3-dimensional space, so that, tuning of PID controller parameters is to search optimization values for K_p, K_i, and K_d [24].

The most crucial step in applying PSO is to choose the best fitness function which is used to evaluate fitness of each particle [25].

The objective function is required to evaluate the best PID controller for the system. An objective function could be created to find a PID controller that gives the smallest overshoot, fastest rise time or quickest settling time [26].

Typical performance criteria have been used to evaluate closed-loop system response include Integral of Squared Error (ISE) index, Integral of time multiplied by Squared Error (ITSE) index, Integral of Absolute Error (IAE) index, and Integral of Time multiplied by Absolute Error (ITAE) index. Each of them has its own characteristic performance. For instance, the ISE index penalizes large errors heavily and small errors lightly. A system designed by this criterion tends to show a rapid decrease in a large initial error. Hence, the response is fast and oscillatory, leading to a system that has poor relative stability, while ITSE places little emphasis on initial errors and heavily penalizes errors occurring late in the transient response to a step input. Therefore a system optimized based on the IAE index penalizes the control error where as if designed using ITAE criterion produces a small overshoot and a well damped oscillation [12].

The above mentioned performance criteria can be calculated by means of the following equations:

$$ISE = \int_0^T e^2(t)\, dt \quad (9)$$

$$IAE = \int_0^T |e(t)|\, dt \quad (10)$$

$$ITAE = \int_0^T t|e(t)|\, dt \quad (11)$$

$$ITSE = \int_0^T te^2(t) dt \quad (12)$$

A typical design of the PSO tuned PID controller is shown in Figure (4).

VI. THE PROPOSED CONTROL SYSTEM

The MA2000 Manipulator was considered as 6 independent joints (6 control loops) which form 6 SISO systems. Each of these SISO systems is controlled independently by a PID controller.

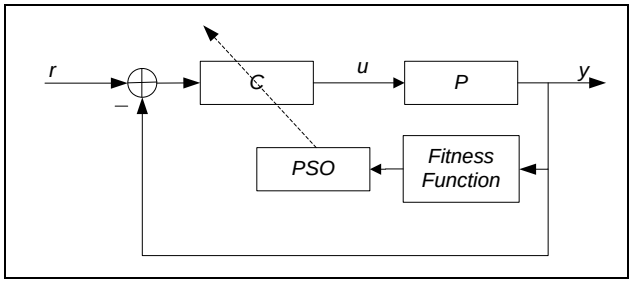

Fig. 4 PSO tuned PID controller

The proposed control system consists of the following elements:

1. MA2000 server machine: This machine is connected to the MA2000 robot arm through 2 interface cards. It contains the server application which exchanges control data with the client application via a Transmission Control Protocol/Internet Protocol (TCP/IP) connection and transfers actuation commands and feedback signals to and from the MA2000 manipulator. As a whole, the MA2000 server machine acts as a peripheral interface for the MA2000 manipulator.

2. MA2000 client machine: The MA2000 client machine contains the client application that is responsible for controlling the MA2000 manipulator joints. The application also contains a tuning module that is used to

tune the joints PID controllers. Data exchange takes place between the client and server applications to form the closed control loop.

3. Interface boards: The control system uses 2 interface boards, the Advantech PCL-711b multifunction card and a custom built interface board. The interface boards provide a physical connection between the server machine and the MA2000 arm, in addition to providing a number of facilities such as data buffering, data de-multiplexing, protection and emergency braking.

The proposed system is depicted in Figure (5).

In order to control the operation of the MA2000 manipulator, 6 independent PID controllers were used. Each controller interacted with a single joint of the MA2000 manipulator. The control process starts by sensing the angular position of each joint, then the PID controller computes the actuator command to correct the positional error of that joint.

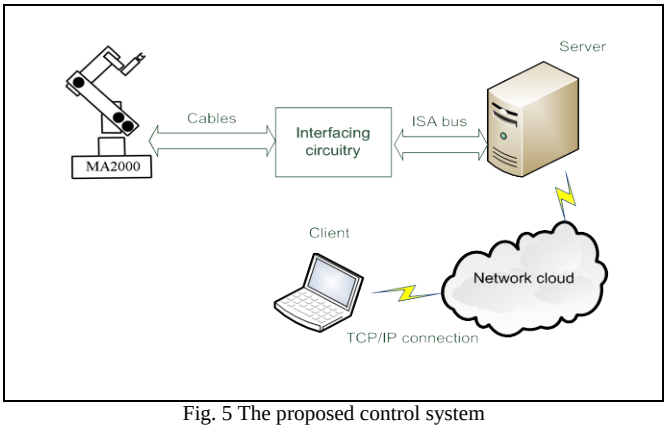

Fig. 5 The proposed control system

The PSO algorithm has been utilized as a tuning method for the proposed controller where each one of the 6 controllers was tuned separately. The tuning phase was done in a serial manner (one joint at a time) to prevent unexpected behavior of the arm and to avoid equipment damage and hazardous operation. A block diagram showing a PSO-PID controller for a single joint is shown in Figure (6).

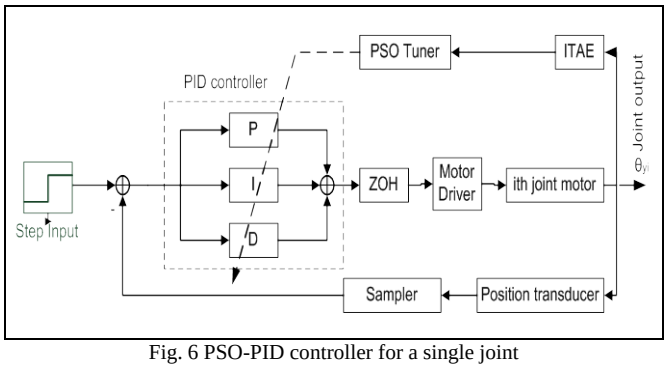

Fig. 6 PSO-PID controller for a single joint

VII. Results

1. Tuning Results

The ITAE performance index was chosen to evaluate the control performance of the joints loops because it penalizes large overshoot and minimizes long settling times, where these two characteristics are of vital importance in obtaining a good and satisfactory control response. The same configuration was adopted for all joint tuners and it is listed in Table (1).

Table 1 Implemented tuning parameters

Swarm Size	60
Number of Iterations	50
Cognitive Acceleration Factor	2.05
Social Acceleration Factor	2.05

The iteration period, which is the time required to move from the tune start position to the tune end position and settle for 2 or 3 seconds, differs depending on the physical characteristics of the joint. Each joint was assigned a different iteration period. These periods were found suitable based on experimental results. The tuning process of each joint took a different time to converge.

The tuning results of the major MA2000 manipulator joints are summarized in Table (2). Table (3) shows the best acquired PID gain sets from the tuning process along with their transient response characteristics.

Table 2 Summary of the tuning results

Joint	ITAE	Run Time (hour:minute)	Iteration Period (second)	Best Response Time (minute)
Waist	0.297	4:32	5	173
Shoulder	0.786	4:30	6	130
Elbow	0.401	5:15	5	125

Table 3 Best acquired PID gains

Joint	Best PID Set			Time Domain Properties		
	K_p	K_i	K_d	T_r (second)	T_s (seccond)	O_s
Waist	12.48	0.15	0	1.12	1.4	0
Shoulder	21.81	1.57	0.02	1.9	2.25	0
Elbow	23.16	1.03	0	1.25	1.45	0

Where T_r, T_s and O_s are the rise-time, settling-time and maximum-overshoot respectively.

The waist joint was tuned using the parameters mentioned in Table (1) in addition to setting its iteration period to 5 seconds. The practically acquired error curves from the tuning operation of the waist joint are shown graphically in Figure (7).

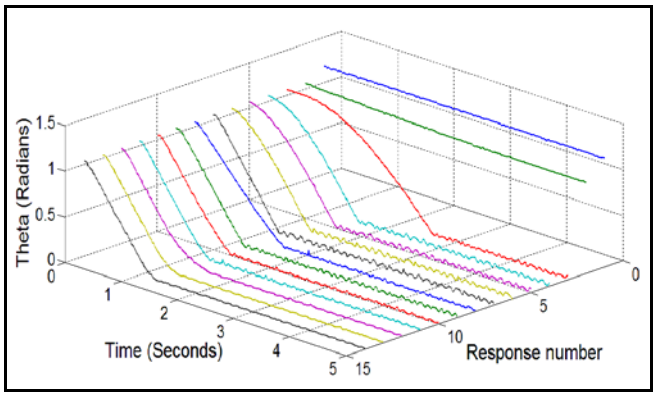

Fig. 7 Cascaded error curves for the best responses of the waist joint showing algorithm convergence

The shoulder joint's tuning iteration period was chosen to be 6 seconds which was found experimentally to suit the shoulder joint. A set of error curves for the best acquired sets during the tuning process are chronically listed in Figure (8).

The elbow joint is the last one of the major joints, which carries at its end, the manipulator wrist. The iteration period was set to 5 seconds. The cascaded error curves for the best acquired joint responses during the tuning process are shown in Figure (9).

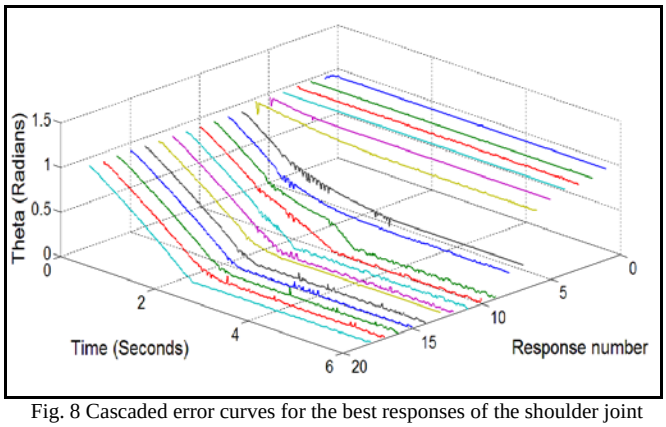

Fig. 8 Cascaded error curves for the best responses of the shoulder joint

1. Movement Test Results

In this test, the end effector is moved from an arbitrarily chosen original position to a new position in the XYZ space as shown in Table 4. The resulting output for the major joints is shown in Figure (10).

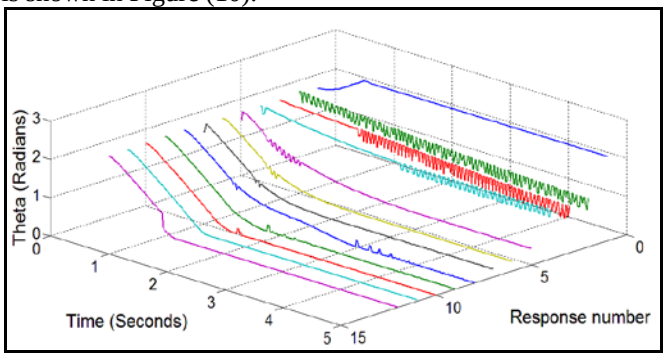

Fig. 9 Cascaded error curves for the best responses of the elbow joint

Table 4 Movement test parameters

Test	Position (mm)			Joint angles (degrees)		
	X	Y	Z	Waist	Shoulder	Elbow
Original position	200	200	200	71	57	58
Moving in XYZ	300	300	300	61	48	108

2. Robustness Test

In the robustness test, the major joints ran concurrently, in order to test the joints' controllers behavior with the existence of the nonlinear effects of the manipulator configuration change, gravity effects and measurement noise. Each of the joints moved from its zero position with a displacement of 1 radian. The resulting output is depicted in Figure (11).

VIII. Conclusions

PID proved to be an effective control algorithm for complex nonlinear systems like robot manipulators, in addition to its simple implementation and computational efficiency.

The PSO has several attractive features that make it an ideal candidate for the tuning of PID controllers, like fast convergence and simple computation.

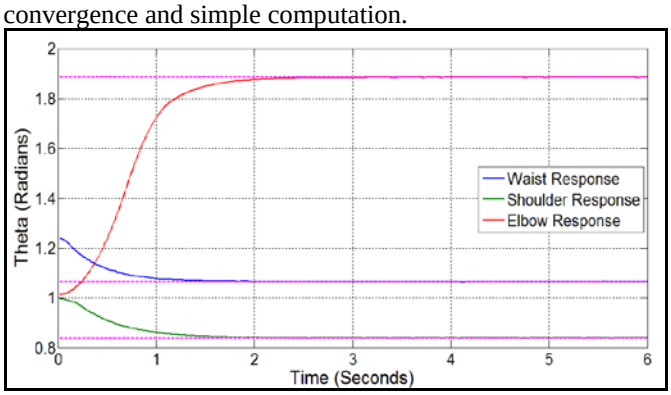

Fig. 10 Movement test

Using online tuning of PID controllers that are applied to nonlinear systems, provides better tuning results than if offline tuning technique is used, since the latter depends on a modeled plant where that plant is one of a set of decoupled models comprising the overall nonlinear system, where some decoupling effects are neglected. On the other hand the online tuning technique uses the real physical plant where the all the nonlinear effects and disturbances are considered.

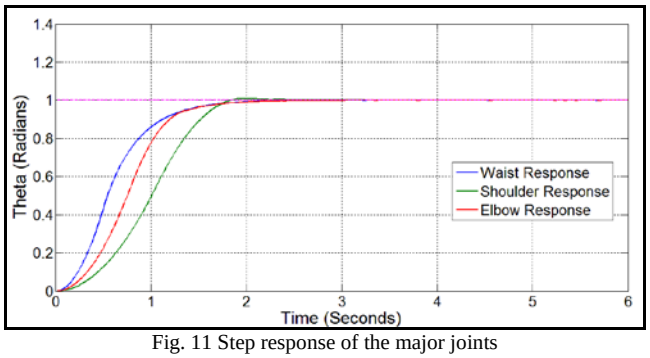

Fig. 11 Step response of the major joints

REFERENCES

[1] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, Robotics: modeling, plannig and control, Springer, 2009.

[2] J. J. Craig, Introduction to Robotics: mechanics and control, Pearson Education, 2005.

[3] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Modeling and Control, JOHN WILEY & SONS, 2005.

[4] T. Yoshikawa, Foundations of robotics : analysis and control, MIT Press, 1990.

[5] F. Cupertino, V. Giordano, D. Naso, L. Salvatore and B. Turchiano, "Genetic Design of Decentralized Controllers for 5-dof Robotic Manipulator," in *International Symposium on Computational Intelligence in Robotics and Automation*, Kobe, 2003.

[6] K. Astrom and T. Hagglund, PID Controllers, 2nd Edition, Instrument Society of America, 1995.

[7] J.-S. Kim, H.-W. Jeon and S. Jung, "Hardware Implementation of Nonlinear PID Controller with FPGA Based on Floating Point Operation for 6-DOF Manipulator Robot Arm," in *International Conference on Control, Automation and Systems*, Seoul, 2007.

[8] Y. Su, P. C. Müller and C. Zheng, "A Global Asymptotic Stable Output Feedback PID Regulator for Robot Manipulators," in *IEEE International Conference on Robotics and Automation*, Roma, 2007.

[9] N. Ravari and H. Taghirad, "A novel hybrid Fuzzy-PID controller for tracking control of robot manipulators," in *IEEE International Conference on Robotics and Biomimetics*, Bangkok, 2008.

[10] Z.-L. Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE TRANSACTIONS ON ENERGY CONVERSION*, vol. 19, no. 2, pp. 384-391, 2004.

[11] Y.-B. WANG, X. PENG and B.-Z. WEI, " A New Particle Swarm Optimization Based Auto-Tuning of PID Controller," in *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kunming, 2008.

[12] A. Marzoughi, H. Selamat and F. Marzoughi, " Application of Particle Swarm Optimization Approach to Improve PID," in *Proceedings of 2010 IEEE Student Conference on Research and Development*, Putrajaya, 2010.

[13] M. R. Dastranj, M. Moghaddas, S. S. Afghu and M. Rouhani, "PID Control of Inverted Pendulum Using Particle Swarm Optimization (PSO)Algorithm," in *IEEE*, 2011.

[14] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE Proceedings of the 4th International Conference on Neural Networks*, 1995.

[15] A. Visioli, Practical PID Control, Springer, 2006.

[16] P.-L. Chen, M.-C. Yang and T.-Y. Sun, "PSO-based on-line tuning PID controller for setpoint changes and load disturbance," in *IEEE Congress of Evolutionary Computation (CEC)*, 2011.

[17] J. Ledin, Embedded Control Systems in C/C++: An Introduction for Software Developers Using MATLAB, CMP Books, 2004.

[18] S. W. Sung, J. Lee and I.-B. Lee, Process Identification and PID Control, John Wiley & Sons, 2009.

[19] W. Y. Svrcek, D. P. Mahoney and B. R. Young, A Real-Time Approach to Process Control,2nd edition, John Wiley & Sons, 2006.

[20] K. J. A. ström and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, Princeton University Press, 2008.

[21] A. P. Engelbrecht, Computational Intelligence: An Introduction, 2nd ed., John Wiley & Sons, 2007.

[22] K. Y. Lee and M. A. El-Sharkawi, Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems, John Wiley & Sons, 2008.

[23] M. Rahimian and K. Raahemifar, "Optimal PID Controller Design for AVR System using Particle Swarm Optimization Algorithm," in *Proceedings of the 24th IEEE Canadian Conference on Electrical and Computer Engineering*, Niagara Falls, 2011.

[24] L. Xu-zhou, Y. Fei and W. You-bo, " PSO Algorithm Based Online Self-Tuning of PID Controller," *IEEE International Conference on Computational Intelligence and Security*, pp. 128-132, 2007.

[25] Z. Bingul and O. Karahan, "Tuning of Fractional PID Controllers Using PSO Algorithm for Robot Trajectory Control," in *Proceedings of the IEEE International Conference on Mechatronics*, Istanbul, 2011.

[26] B. Nagaraj and N. Murugananth, "A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO," *International Conference on Communication, Control and Computing Technologies*, pp. 305-313, 2010.