PSO based Adaptive Force Controller for 6 DOF Robot Manipulators

Sutthipong Thunyajarern, Uma Seeboonruang and Somyot Kaitwanidvilai

***Abstract*— Force control in robot arm has been used in many industrial applications especially end-effector contacting with environment. When environment is change, the performance of non-adaptive controller may be decreased. This paper presents adaptive force controller for 6 DOF (degree of freedom) Robot Manipulators that do not require identifying the environment before controlling. Particle swarm optimization (PSO) has been employed to solve this problem. In simulation, the end-effector was moved and touched different environments. The simulation results were compared with typical non-adaptive control. The result shows that when the environment is changed, the performance of non-adaptive force controller decreased. On the other hand, the performance of PSO based adaptive force controller remained the same.**

***Index Terms*— force control, impedance control, environment modelling, particle swarm optimization**

I. Introduction

Force controller in robot manipulators has been adopted in a number of industries, for example, automotive industries, semiconductor and electronic component assembly lines. Force controlling methods can be categorized into direct and indirect force controls. This research applied the indirect force control system. This control scheme composed of two control loops; inner and outer loops. Inner loop controller is for position control of end-effector, while outer loop control is for force control. The outer loop control sends coordinate information to the inner loop control in order to calculate the position and control the end-effector.

In this paper, the indirect force control, which is impedance control, controls the force contact between end-effector and environment. The impedance control can regulate force contact with environment according to some specified mechanical impedance variables such as mass, damping coefficient, and spring coefficient. When the environment is changed, these parameters of impedance controller will be set to new values to generate the desired response. The method of particle swarm optimization (PSO) has been employed in order to search for the new set of parameters for this impedance controller.

Manuscript received July 7, 2017; revised July 22, 2017. This work is supported by National Science and Technology Development Agency and Thai Steel Cable Public Company Limited.

S. Thunyajarern is with Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand (e-mail: joandblood@gmail.com).

U. Seeboonruang is with Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand (e-mail: uma.se@kmitl.ac.th).

S. Kaitwanidvilai is with Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand (phone: 662-329-8300; fax: 662-329-8302; e-mail: drsomyotk@gmail.com).

At present, there are various research on force control in robot arm that apply the impedance control. Initially, 2 DOF force control operated 2 DOF robot arm and environment. Typically, this control scheme employed indirect force control, which used impedance control to control force contact between end effector and environment [1]. The study on the comparison between the impedance control and force-free control showed that both of them can stop end effector when in contact with environment but they may have different behaviors [2]. The optimization algorithm, particle swarm optimization (PSO), has been employed to solve many problems, such as tuning parameters of regression support vectors for predicting pinch force signal. The applications with PSO always perform with better accuracy than typical application [3]. Another application of PSO is for tuning parameters of PID controller to control robot arm as PID controller in MIMO plant is difficult to tune manually [4]. When environment is changed, PSO has been successfully applied in order to adjust parameters of impedance controller [5]. In other algorithm, artificial neural network (ANN) together with neuro-fuzzy is used to control force in contact with environment. When end- effector contacting with environment, force contact overshoots force signal, the neuro-fuzzy algorithm applies reference force through transfer function so that the reference force increases smoothly [6]. Moreover, this technique is the force control method, which combines the adaptive fuzzy controller and PID controller. This latest advancement applies PSO for adjusting the PID parameters for active suspension system [7].

The problem of the force control is mainly caused by the decrease of performance when the environment changes. Although there are many adaptation algorithms proposed to solve this problem; however, as shown in the previous works, the adaptive algorithm with the local minima avoidance for the force control has not been proposed in the 6DOF robot manipulators. The proposed technique enhances the ability of the force control system so that the force at the end-effector can be adapted to the environment changes which is more realistic on the force control system.

This research paper presents the PSO based adaptive force control for 6 DOF. The system of force control, position control and PSO will be proposed. The expected result is the better performance of the proposed force control over the typical impedance control.

II. Overview of this System

Firstly, the overview of PSO based adaptive force control in robot arm will be presented (Fig. 1). From block diagram, xyz is the desired position of end effector in Cartesian coordinate system.

Inverse kinematics equations are applied to determine the joint parameters that provide information on a desired position for each of the robot's end effectors. Position control is controller for controlling the joint of robot arm. Environment means the material contacting to end effector. “f ref” is force reference signal. Model reference is the transfer function. The PSO scheme is applied here for tuning parameters of the impedance control.

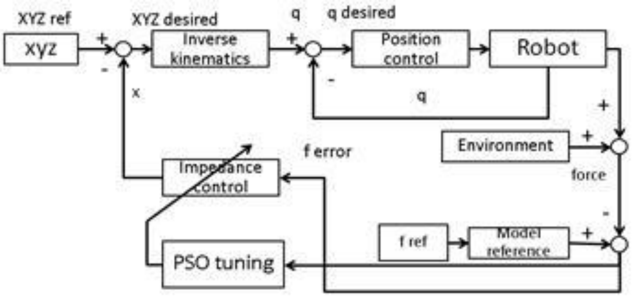

Fig. 1. Block diagram of PSO based adaptive force control in 6 DOF robot manipulators.

A. Forward Kinematics

Forward kinematics equations refer to those equations of a robot to compute the position of the end effector from some specified values for the joint parameters. The kinematics equations for the series chain of a robot are obtained using a rigid transformation [Z] to characterize the relative movement allowed at each joint and separate rigid transformation [X] to define the dimensions of each link. The result is a sequence of rigid transformations alternating joint and link transformations from the base of the chain to its end link, which is equated to the specified position for the end link.

$$[T] = [Z_1][X_1][Z_2][X_2][Z_3][X_3] \ldots [Z_{n-1}][X_{n-1}] \quad (1)$$

Where [T] is the transformation locating the end-link, [Z] is the joint matrices, [X] is the link matrices.

Joint matrices can be defined as the multiplication of translation matrix and rotation matrix as shown in the following equation.

$$Z_i = Trans_{Z_i}(d_i) Rot_{Z_i}(\theta_i) \quad (2)$$

Where $Trans_{Z_i}$ is translation matrix of joint, Rot_{Z_i} is rotation matrix of joint. The matrices can be defined as in the following equation.

$$X_i = Trans_{X_i}(a_{i,i+1}) Rot_{X_i}(\alpha_{i,i+1}) \quad (3)$$

Where $Trans_{X_i}$ is translation matrix of link, Rot_{X_i} is rotation matrix of link. Then, the change of link can be defined as the following:

$${}^{i-1}_{i}T = [Z_i][X_i] = Trans_{Z_i}(d_i) Rot_{Z_i}(\theta_i) Trans_{X_i}(a_{i,i+1}) Rot_{X_i}(\alpha_{i,i+1}) \quad (4)$$

Where θ_i, d_i, $a_{i,i+1}$ and $\alpha_{i,i+1}$ are known as the Denavit-Hartenberg parameters.

B. Inverse Kinematics

In robotics, inverse kinematics employs the kinematics equations to determine the joint parameters that provide a desired position of the robot's end effectors. There are many methods of modelling and solving inverse kinematics problems. Most of the flexible methods among these typically rely on iterative optimization to seek out an approximate solution because of the difficulty of inverting the forward kinematics equation and the possibility of an empty solution space. In this paper, the Jacobian inverse technique is applied. From the following equation,

$$de = J d\theta \quad (5)$$

Where e is position of end effector in Cartesian coordinate system, de is distance between positions of the target and the current position of end effector, θ is joint angle, dθ is difference of joint angles between that of the current position and the position of end effector. J is Jacobian matrix. Then, θ may be described as:

$$d\theta = J^{-1} de \quad (6)$$

Pseudo-inverse is applied for inversion from J to J+ and the following equation is obtained:

$$d\theta = J^{+} de \quad (7)$$

Now the new joint angle (θ) can be described as:

$$\theta = \theta + k d\theta \quad (8)$$

Where k is a constant for the changing rate.

During the Jacobian inverse technique, the steps are applied:

While (e is far from the expected e)

{

1. Compute the Jacobian matrix - J
2. Compute the pseudo-inverse of the Jacobian matrix – J^{+}
3. Compute change in joint DOF: $d\theta = J^{+} de$
4. Compute the new location of the joint DOF: $\theta = \theta + k d\theta$
5. Check the new position by forward kinematics: error = expected e - e.

}

C. Dynamic Model

Dynamic model of a robot arm is given by:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau \quad (9)$$

Where $M(q)$ is mass matrix, $C(q,\dot{q})$ is Coriolis and centrifugal forces, $G(q)$ is gravitational force, τ is torque of each joint, q, $\dot{q}$, and $\ddot{q}$ are angle, velocity and acceleration of joint, respectively.

The dynamic equation of an end-effector as the result from external force at the end-effector is given by:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau - J^{T} F_{ext} \quad (10)$$

Where J is end effector Jacobian matrix, F_{ext} is external force in Cartesian coordinate system.

D. Position Control of End Effector

From dynamic model, the control signal can be established as:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) + J^T F_{ext} = u \tag{11}$$

Given the new controller input:

$$\ddot{q} = \ddot{q}_r \tag{12}$$
$$\ddot{q} + K_2\dot{q} + K_1 q = \ddot{q}_r \tag{13}$$

Where K_1 is a multiplier of joint positions, K_2 is a multiplier of joint velocity, $\ddot{q}_r$ is joint accelerate. Then replace $\ddot{q}$ in (11) by $\ddot{q}_r$

The error can be computed by:

$$e = q_d - q \tag{14}$$

Where q_d is expected joint position and q is current joint angle. Then the following equation is obtained:

$$\ddot{e} + K_2\dot{e} + K_1 e = 0 \tag{15}$$

In this paper, K_1 is given as [1000 100 100 10 10 10] and K_2 is given as [50 20 20 1 1 1].

E. Impedance control

Impedance control is the force control between the external force and end-effector according to specified impedance parameters such as mass, spring coefficient and damper coefficient. The main equation of impedance control can be given by:

$$F = m\ddot{x} + b\dot{x} + kx \tag{16}$$

Where m is mass, b is damper coefficient, k is spring coefficient, F is force, and x is displacement.

The impedance parameters such as mass (m), damper coefficient (b), and spring coefficient (k) will be optimized by Particle Swarm Optimization (PSO).

When the end effector initially in contacting with environment, there will be over shoot force signal. In order to solve this problem, transfer function is then applied to the force signal so that the reference force signal would grow smoothly to steady state. The transfer function is in the form of first-order dynamic model (17).

$$G_r(s) = \frac{1}{cs+1} \tag{17}$$

Where c is a time constant.

F. Environment modelling

In this paper, the movement of end effector to contact with environment is in Cartesian coordinate system. The model of environment is a simple linear spring model as (18).

$$f = k_{spring}(x - x_e) \tag{18}$$

Where f is force, k_{spring} is spring constant, x is position of end effector in Cartesian coordinate system, x_e is the static position of the environment.

G. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution regarding to a given measure of quality. PSO's algorithm is similar to the scheme of a group of birds finding food. Birds are compared to particles, which have position (x_i) and velocity (v_i). The best position of each particle is *pbest* and the best position of the swarm is *gbest*.

Main algorithm of Particle Swarm Optimization can be described as following:

1. Locate positions of all particles by random values between upper limit and lower limit.
2. Calculate a fitness value by using the fitness function (23) in order to find a better position of a particle. If the new position of particle is better than before, the position will be remembered as *pbest*.
3. Find the best fitness value of swarm in each iteration, and give that position as *gbest*.
4. Calculate new velocities of particles (20).
5. Update the new positions of particles (19). And then repeat steps #2 to #5 until the number of iteration reaches the maximum value.

Equation for updating position is (19).

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{19}$$

Where x_i is position and v_i is velocity.

As the result, the updated velocity can be computed as the following equation.

$$v_i(t+1) = wv_i(t) + c_1r_1(pbest_i(t) - x_i(t)) + c_2r_2(gbest(t) - x_i(t)) \tag{20}$$

Where w is a coefficient for inertial weight, c_1 and c_2 are acceleration constants, r_1 and r_2 are positive values given by random generation (21).

$$r = \frac{random(0\ to\ 1)}{2} \tag{21}$$

Where *random*(0 to 1) is a random function generating a number range between 0 and 1.

The equation for updating w is given by (22)

$$w = iter\,\frac{w_{max} - w_{min}}{iter_{max}} \tag{22}$$

Where *iter* is number of current iteration, $iter_{max}$ is number of maximum iteration, w_{max} is maximum value of inertial weight, w_{min} is minimum value of inertial weight.

To establish *pbest* and *gbest*, the fitness value is acquired from the fitness function (23).

$$f = \int e^2 dt \tag{23}$$

ISBN: 978-988-14048-4-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Where f is fitness value and e is the feedback force error. PSO scheme can be shown as in the block diagram (Fig. 2).

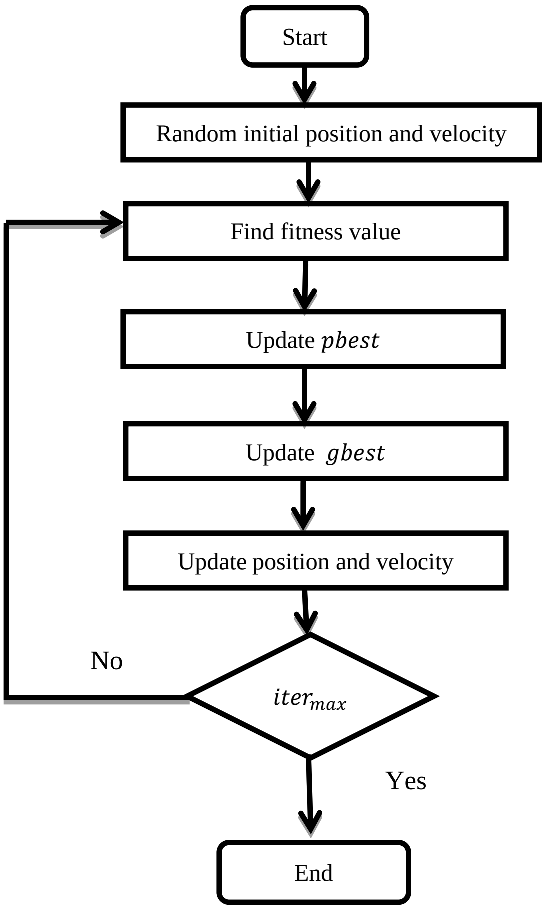

Fig. 2 Scheme of Particle Swarm Optimization (PSO)

III. Simulation and Result

In this simulation, the applied robot arm is puma560 [8]. First, the initial position of end effector is set at y = -0.1501 m, z = 0 m and x = 0.5 m. The ramp function has slope = 0.8. The external force is k_{spring} = 500 N/m at x = 0.6 m, while the force reference is set at 30 N.

The transfer function applied here is expressed as (24)

$$G_r(s) = \frac{1}{0.2s+1} \tag{24}$$

The PSO parameters are as following, $iter_{max}$ = 15, particles = 10, dimension = 3 (m, b, k), $c_1 = c_2 = 2.1$, w_{max}= 3, w_{min}= 0.6, upper limit = 1000, and lower limit =0.1

The result of the PSO based the adaptive force controller for 6 DOF robot manipulators is shown here.

Fig. 3 shows the function of the fitness value. The result suggests that the optimized value has been established at the iteration 5th. Fig. 4 shows contact force between end effector and environment. The contact force is increasing smoothly following the model reference through the transfer function.

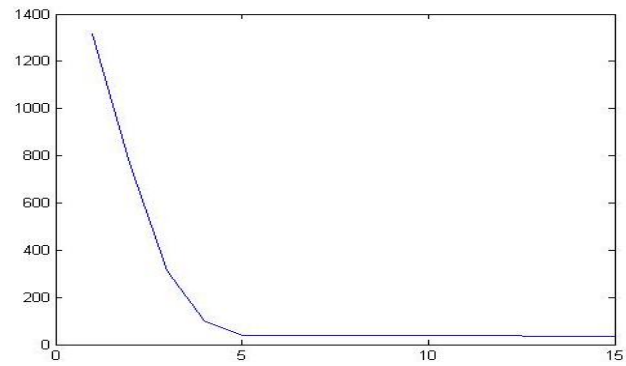

Fig.3. Fitness value function

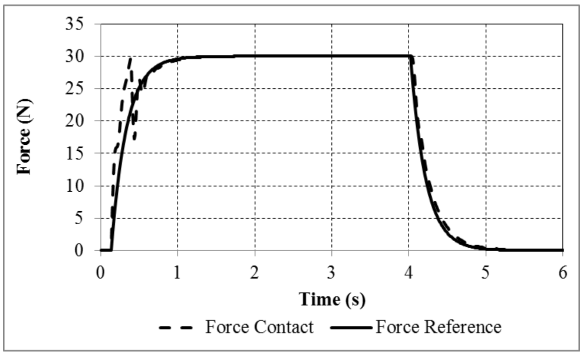

Fig.4. Contact force between end effector and environment after optimized by PSO

Fig. 5 shows the end effector position on x-axis when in contact with environment. The end effector stops at the x reference position in order to maintain the equilibrium of the contact force at the end effector.

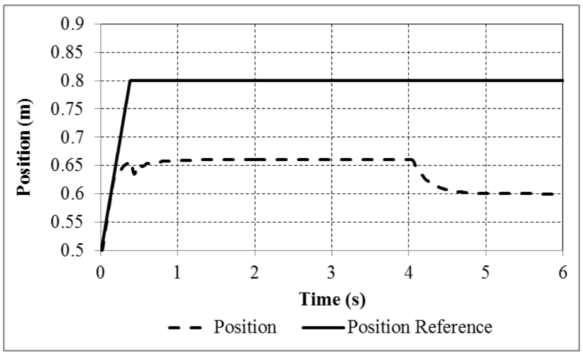

Fig.5. Position of end effector when in contact with environment.

To test the control force efficiency, another simulation is set with k_{spring}= 1000 N/m and the result is shown in Fig. 6.

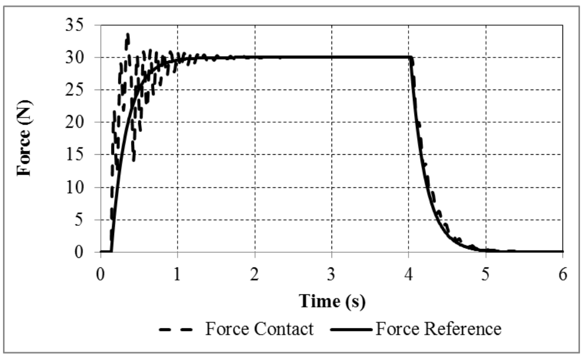

Fig.6. Contact force with k_{spring}= 1000 N/m

The results from the simulation with the traditional impedance control at k_{spring}= 500 N/m and k_{spring}= 1000 N/m are also shown in Fig. 7 and Fig. 8, respectively.

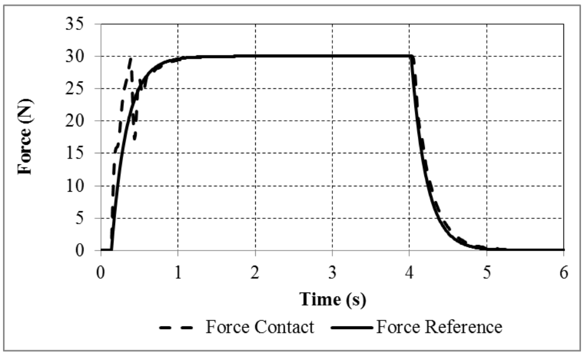

Fig.7. Contact force with impedance control when k_{spring}= 500 N/m

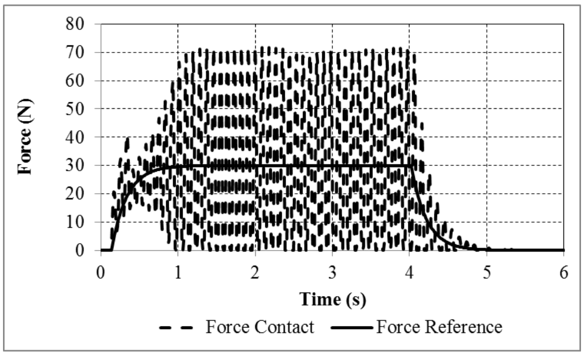

Fig.8. Contact force with impedance control when k_{spring}= 1000 N/m

Fig. 7 and Fig. 8 illustrate the result that when environment is changed, the performance of the contact force with the traditional impedance control decreases.

IV. Conclusion

This paper presents the PSO based adaptive force controller for 6 DOF robot manipulators. When end effector is in contact with environment, contact force increases smoothly following the reference signal, which can reduce the damage of the end effector and environment. Comparing to the non-adaptive force control algorithm when the environment is changed, the proposed technique can be adapted to control the contact force with better performance in terms of low oscillation and overshoot. The simulation results have proven the effectiveness of this proposed system.

Acknowledgment

The project is financially funded by by National Science and Technology Development Agency and Thai Steel Cable Public Company Limited under the project "The development of integrated vision based-robotic arm with wireless communication for assembly production line of automotive window drive system".

References

[1] J. de Gea, F. Kirchner, "Modelling and Simulation of Robot Arm Interaction Forces Using Impedance control ", Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008

[2] A. Pallegedara, Y. Matsuda, N. Egashira, T. Sugi and S. Goto, " Comparison of Interaction Forces of Robot Arm on Force-free Control and Impedance Control by Model Based Simulations", SICE Annual Conference 2012 August 20-23, 2012, Akita University, Akita, Japan

[3] X. Wang, J. Gao, "Application of Particle Swarm Optimization for Tuning the SVR Parameters*", 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM) July 7-11, 2015. Busan, Korea

[4] O. Djaneye-Boundjou, X. Xu, and R. Ordónez, "Automated Particle Swarm Optimization Based PID Tuning for Control of Robotic Arm", 978-1-5090-3441-3/16/$31.00 ©2016 IEEE

[5] S. Kaitwanidvilai and Y. Sarucha, and A. Saenthon, "PSO based Adaptive Force Controller", IMECS 2013, March 2013.

[6] S. Kaitwanidvilai and M. Parnichkun, "Force control in a pneumatic system using hybrid adaptive Neuro-fuzzy model reference control," Mechatronics 2005, Vol. 15, No. 1, pp. 23–41.

[7] J. Chiou, S. Tsai and M. Liu, "A PSO-based adaptive fuzzy PID-controllers", Simulation Modelling Practice and Theory 26 (2012) 49–59

[8] P.I. Corke, "Robotic Toolbox for Matlab release 9 ", release date April 2014

[9] M.W. Spong, Seth Hutchinson, and M. Vidyasagar "Robot Dynamics and Control Second Edition", January 28, 2004