https://doi.org/10.21289/KSIC.2020.23.2.149

A Study on Track Record and Trajectory Control of Articulated Robot Based on Monitoring Simulator for Smart Factory

Hee-Jin Kim[1], Guen-Han Dong[1], Dong-Ho Kim[1], Gi-Won Jang[1], Sung-Hyun Han[2]

〈Abstract〉

We describe a new approach to implement of trajectory control and track record of articulated manipulator based on monitoring simulator for smart factory. The learning control algorithm was applied in implementation real-time control to provide enhanced motion control performance for robotic manipulators.

The proposed control scheme is simple in structure, fast in computation, and suitable for real-time control. Moreover, this scheme does not require any accurate dynamic modeling, or values of manipulator parameters and payload. Performance of the proposed controller is illustrated by simulation and experimental results for robot manipulator consisting of six joints at the joint space and Cartesian space.by monitoring simulator.

Keywords : Trajectory Control, Track Record, Monitoring simulator, Smart Factory, Real-Time Implementation, Articulated Robot

---

1 Dept.of Mechanical Engineering., Graduate School, Kyungnam University, Changwon, Korea
2 Dept. of Mechanical Engineering., Kyungnam Univ.Korea

1. INTRODUCTION

Current industrial approaches to the design of robot system control systems treat each joint of the robot arm as a simple servomechanism. This approach models the time varying dynamics of a manipulator inadequately because it neglects the motion and configuration of the whole arm mechanism. [2] The changes in the parameters of the controlled system are significant enough to render conventional feedback control strategies ineffective. This basic control system enables a manipulator to perform simple positioning tasks such as in the pick-and-place operation. However, joint controllers are severely limited in precise tracking of fast trajectories and sustaining desirable dynamic performance for variations of payload and parameter uncertainties. In many servo control applications the linear control scheme proves unsatisfactory, therefore, a need for nonlinear techniques is increasing. [1]

2. SIMULATOR DESIGN

Monitoring simulator developed in this study has various functions for a convenient user interface. These functions will require the establishment of a three-dimensional database and the establishment of a graphical algorithm. In particular, this monitoring simulator is designed to operate on a PC. In the case of simulators using graphics, one of the most frequently used functions is the visual position change function, so in this study, the visual position change function is more convenient. [3]

The main functions of the monitoring simulator consist of trajectory planning and path control, data record, state analysis and diagnosis, emulation function, kinetic simulation function, user interface function and three-dimensional modeling function. [4]

(1) Path planning

The offline programming system must emulate exactly the path taken by the manifestator in space. This emulation is necessary for setting up the robot's work and for predicting collisions accurately. Programs developed for this problem provide a route planning approach to select and use. Windows OS supports route planning by configuring virtual classroom boxes using dialog boxes.

(2) kinematic simulation

It is necessary to keep a world that has been simulated in three dimensions valid. It is usually possible by including the regular and reverse mechanics of robots in programs. [5]

In the program that developed the regular and reverse mechanics of robots, the sub-function was written separately by solving the formula.

In terms of dynamic character analysis, if the offline programming system emulates the controller's path planning algorithm well and the actual robot follows the desired path at a very slow rate, the dynamic nature of the simulated behavior in the program can be ignored. However, errors in tracking paths at very high loads or at very high speeds can be important. Accurate dynamic emulation is required to simulate these tracking errors. [6]

Programs developed for this problem were coded as C++ and sub-saturated. In the obtained kinetics equation, the motion equation for the driving motor, a servo system that drives the robot, was also considered and performed through modeling closer to the actual system. [7]

(3) User interface

Offline programming systems should be easy and convenient for users to learn how to use and manage program data. In addition, it is necessary to develop a computer-based user interface method that can replace the robot system's gypsy box. [8]

Much of this problem, which was presented in developing the program, has already been resolved by upgrading to Microsoft's Windows OS. This is because Windows OS basically supports the GUI environment for user convenience. The offline programming system, Windows OS, built-in dialog and mouse functions, and shortcut features made it easier to manage and use programs. Eventually, the box focused on creating some variations of the functions provided on the dialog box and making them as easy to use as possible. [9]

(4) Three-dimensional modeling

It calls for the robot itself, tools and working cells presented on the computer screen to all be modeled into three-dimensional objects. Also, these three-dimensional modeled on-screen objects should have animation features. The program developed for this problem has constructed three-dimensional modeling using OpenGL, a graphic library provided by Silicon Graphics. The use of this library allowed significant speed improvements and eliminated the slightest flickering that existed in the existing GDI (Graphic Device Interface) for Windows during high-density animation. [10]

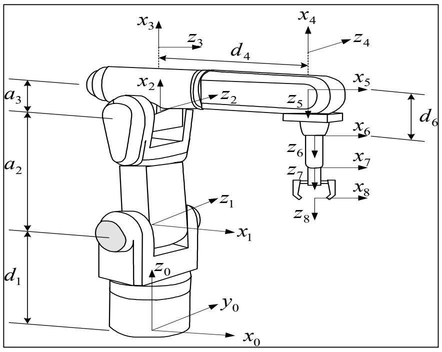


Fig. 1 Link coordinates of a manipulator with six joint

Fig.1 represents link coordinates of the robot. Table I lists values of specification of the robot.

Let us consider a no redundant joint robotic manipulator in which the n×1 generalized joint torque vector τ(t) is related to the n×1 generalized joint coordinate vector q(t) by the following nonlinear dynamic equation of motion. [11]

$$D(q)\,\ddot{q} + N(q,\dot{q}) + G(q) = \tau(t) \tag{1}$$

where $D(q)$ is the n×n symmetric positive-definite inertia matrix, $N(q,\dot{q})$ is the n×1 coriolis and centrifugal torque vector, and $G(q)$ is the n×1 gravitational loading vector. [12]

Equation (1) describes the manipulator dynamics without any payload. Now, let the n×1 vector X represent the end-effector position and orientation coordinates in a fixed task-related cartesian frame of reference. The cartesian position, velocity, and acceleration vectors of the end-effector are related to the joint variables by

$$\begin{aligned} X(t) &= \Phi(q) \\ X(t) &= J(q)\,\dot{q}(t) \\ X(t) &= \dot{J}(q,\dot{q})\,\dot{q}(t) + J(q)\,\ddot{q}(t) \end{aligned} \tag{2}$$

where Φ(q) is the n×1 vector representing the forward kinematics and J(q) = [$\partial\,\Phi$(q)/∂ q] is the n×n Jacobian matrix of the manipulator. [13]

Let us now consider payload in the manipulator dynamics. Suppose that the manipulator end-effector is firmly grasping a payload represented by the point mass ΔM_p. For the payload to move with acceleration $\ddot{X}(t)$ in the gravity field, the end-effector must apply the n×1 force vector $T(t)$ given by

$$T(t) = \Delta M_p\,[\,\ddot{X}(t) + g\,] \tag{3}$$

where g is the n×1 gravitational acceleration vector.

The end-effector requires the additional joint torque

$$\tau_f(t) = J(q)^T\,T(t) \tag{4}$$

where superscript T denotes transposition. Hence, the total joint torque vector can be obtained by combining equations (1) and (4) as

$$J(q)^T\,T(t) + D(q)\,\ddot{q} + N(q,\dot{q}) + G(q) = \tau(t) \tag{5}$$

Substituting equations (2) and (3) into equation (5) yields

$$\begin{aligned} &\Delta M_p\,J(q)^T[J(q)\,\ddot{q} + \dot{J}(q,\dot{q})\,\dot{q} + g\,] \\ &+ D(q)\ddot{q} + N(q,\dot{q}) + G(q) = \tau(t) \end{aligned} \tag{6}$$

Equation (6) shows explicity the effect of payload mass ΔM_p on the manipulator dynamics. This equation can be written as

$$[D(q)+\Delta M_p J(q)^T J(q)]\,\ddot{q}+[N(q,\dot{q}) +\Delta M_p J(q)^T \dot{J}(q,\dot{q})\,\dot{q}]+[G(q)+\Delta M_p J(q)^T g] \;=\tau(t) \qquad (7)$$

where the modified inertia matrix $[D(q)+\Delta M_p J(q)^T J(q)]$ is symmetric and positive-definite. Equation (7) constitutes a nonlinear mathematical model of the manipulator-plus-payload dynamics. [14]

3. CONTROL ALGORITHM

The manipulator control problem is to develop a control scheme which ensures that the joint angle vector $q(t)$ tracks any desired reference trajectory $q_r(t)$, where $q_r(t)$ is an n×1 vector of arbitrary time functions. It is reasonable to assume that these functions are twice differentiable, that is, desired angular velocity $\dot{q}_r(t)$ and angular acceleration $\ddot{q}_r(t)$ exist and are directly available without requiring further differentiation of $q_r(t)$. It is desirable for the manipulator control system to achieve trajectory tracking irrespective of payload mass ΔM_p. [15]

The controllers designed by the classical linear control scheme are effective in fine motion control of the manipulator in the neighborhood of a nominal operating point P_o. During the gross motion of the manipulator, operating point P_o and consequently the linearized model parameters vary substantially with time. Thus it is essential to adapt the gains of the feed-forward, feedback, and PI controllers to varying operating points and payloads so as to ensure stability and trajectory tracking by the total control laws. The required adaptation laws are developed in this section. [16][18]

Nonlinear dynamic equation (7) can be written as

$$\tau(t) = D^*(\Delta M_p, q, \dot{q})\,\ddot{q}(t) + N^*(\Delta M_p, q, \dot{q})\,\dot{q}(t) + G^*(\Delta M_p, q, \dot{q})\,q(t) \qquad (8)$$

where D^*, N^*, and G^* are n×n matrices whose elements are highly nonlinear functions of ΔM_p, q, and $\dot{q}$.

In order to cope with changes in operating point, the controller gains are varied with the change of external working condition. [17][18]

This yields the PI control law

$$\tau(t) = [P_A(t)\,\ddot{q}_r(t) + P_B(t)\,\dot{q}_r(t) + P_C(t)\,q_r(t)] + [P_V(t)\,\dot{E}(t) + P_P(t)\,E(t) + P_I(t)] \qquad (9)$$

where $P_A(t)$, $P_B(t)$, $P_C(t)$ are feedforward time-varying PI gains, and $P_P(t)$ and $P_V(t)$ are the feedback PI gains, and $P_I(t)$ is a time-varying control signal corresponding to the nominal operating point term, generated by a feedback controller driven by position tracking error $E(t)$ defined as $q_r(t)-q(t)$. [19]

On applying control law to nonlinear model, the error differential equation can be obtained as

$$D^*\ddot{E}(t)+(N^*+P_V)\dot{E}(t)+(G^*+P_P)\,E(t)$$
$$=P_I(t)+(D^*-P_A)\ddot{q}_r(t)+(N^*-P_B)\dot{q}_r(t)+(G^*-P_C)q_r(t) \qquad (10)$$

Defining the 2n×1 position-velocity error vector $\delta(t)=[E(t),\dot{E}(t)]^T$, equation (10) can be written in the state-space form

$$\dot{\delta}(t)=\begin{pmatrix}0 & I_n\\ Z_1 & Z_2\end{pmatrix}\delta(t)+\begin{pmatrix}0\\ Z_3\end{pmatrix}q_r(t)+\begin{pmatrix}0\\ Z_4\end{pmatrix}\dot{q}_r(t)$$
$$+\begin{pmatrix}0\\ Z_5\end{pmatrix}\ddot{q}_r(t)+\begin{pmatrix}0\\ Z_6\end{pmatrix} \qquad (11)$$

where

$Z_1=-[D^*]^{-1}[G^*+P_P]$, $Z_2=-[D^*]^{-1}[N^*+P_V]$

$Z_3=[D^*]^{-1}[G^*-P_C]$, $Z_4=[D^*]^{-1}[N^*-P_B]$

$Z_5=[D^*]^{-1}[G^*-P_A]$ and $Z_6=-[D^*]^{-1}[P_I]$

Equation (11) constitutes an adjustable system in the model reference PI control frame-work. We shall now define the reference model which embodies the desired performance of the manipulator in terms of the tracking error E(t). [20]

The desired performance is that each joint tracking error $E_i(t)=q_n(t)-q_i(t)$ be decoupled from the others and satisfiy a second-order homogeneous differential equation of the form

$$\ddot{E}_i(t)+2\xi_i\varpi_i\dot{E}_i(t)+\varpi_i^2E_i(t)=0 \quad (i=1,\ \cdots,\ n) \qquad (12)$$

where ξ_i and ϖ_i are the damping ratio and the undamped natural frequency. [21][22]

The desired performance of the control system is embodied in the definition of the stable reference model equation (12) as following vector equation (13).

$$\dot{\delta}_\gamma(t)=\begin{pmatrix}0 & I_n\\ -S_1 & -S_2\end{pmatrix}\delta_\gamma(t) \qquad (13)$$

where $S_1=diag(\varpi_i^2)$ and $S_2=diag(2\xi_i\varpi_i)$ are constant n×n diagonal matrices, $\delta_\gamma(t)=[E_\gamma(t),\dot{E}_\gamma(t)]^T$ is the 2n×1 vector of desired position and velocity errors, and the subscript 'γ' denotes the reference model.

Because reference model is stable, equation (13) has Lyapunov function's solution R defined as following equation

$$RS+S^TR=-H \qquad (14)$$

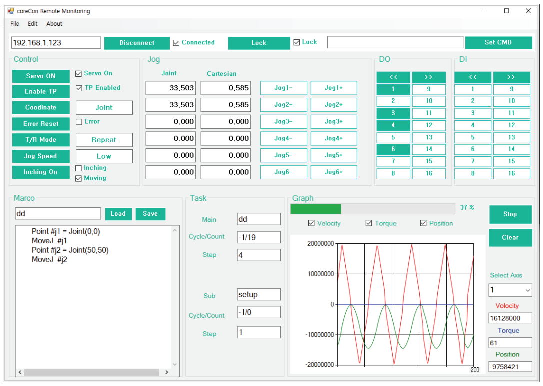
where H is symmetric positive definite matrix. R is symmetric positive definite matrix. We shall now state the adaptation laws which ensure that, for any reference trajectory $q_r(t)$, the state of the adjustable system, $\delta(t)=[E(t),\dot{E}(t)]^T$ approaches $\delta_\gamma(t)=0$ asymptotically. The controller adaptation laws will be derived using the direct Lyapunov method-based model reference PI control technique. [23][24][25]

4. EXPERIMENTS AND RESULTS

4.1 3D Simulation and experiments

This section represents the simulation and experiment results of the position and velocity control of a six-link robotic manipulator by the proposed PI control algorithm, as shown in Fig. 4, and discusses the advantages of using joint controller based-on EtherCAT communication for motion control of a robot. The controller scheme developed in this paper will be applied to the control of a robot with six axes

☐ Monitoring Simulator



192.168.1.123 ☑ Connected
Lock ☑ Lock

☑ Servo On
☑ TP Enabled
Joint
☐ Error
Repeat
Low
☐ Inching
☑ Moving

Joint	Cartesian
33.503	0.585
33.503	0.585
0.000	0.000
0.000	0.000
0.000	0.000
0.000	0.000

Marco: dd

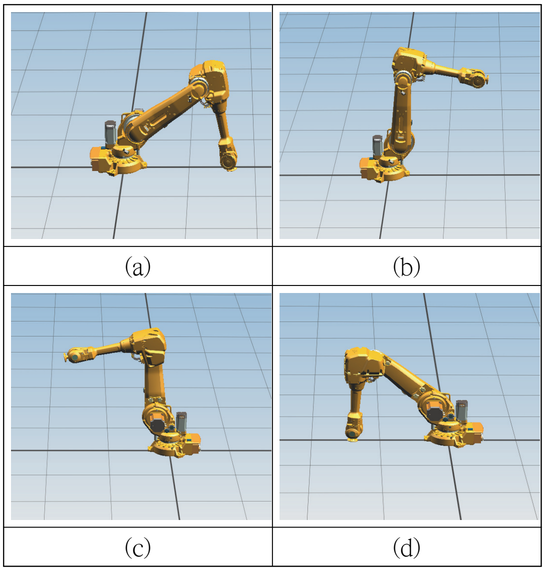
```
Point #j1 = Joint(0,0)
MoveJ  #j1
Point #j2 = Joint(50,50)
MoveJ  #j2
```

Main: dd
Cycle/Count: -1/19
Step: 4
Sub: setup
Cycle/Count: -1/0
Step: 1

37 %
☑ Velocity ☑ Torque ☑ Position

Select Axis: 1
Volocity: 16128000
Torque: 61
Position: -9758421

Fig.2 The Scene of Monitoring Simulator Start Menu



(a) (b)
(c) (d)

Fig.3 The Operating Scene of robot manipulator with six joints by monitoring simulator

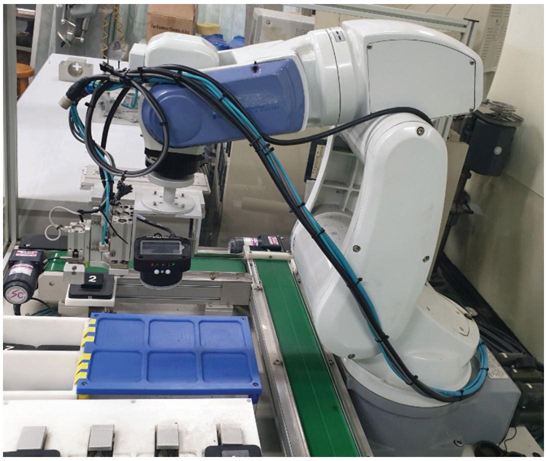


Fig. 4 This scene of experiments for data records

Consider the robot with the end-effector grasping a payload of mass ΔM_p.

In all simulations the load is assumed to be unknown, the PI control algorithm given in equation (10).

It is assumed that $\omega_1 = \omega_2 = 10 rad/\sec$, $\xi_1 = \xi_2 = 1$, and $S_1 = 80I$, $S_2 = 25I$ in the reference model. The sampling time is set as 0.05 sec. Simulations are performed to evaluate the position and velocity control of each joint under the condition of payload variation, inertia parameter uncertainty, and reference trajectory variation. Control performance for the reference trajectory variation is tested for four different position reference trajectories C and velocity reference trajectories D for each joint

The performance of monitoring system is evaluated in track record and tracking errors of the position, velocity, and torque for the six joints.

From the Fig.5 to Fig.10 show the data record results of position trajectory control for each joins. From the Fig.11 to Fig.16 show the data record results of velocity trajectory control for each joins. From the Fig.17 to Fig.22 show the data record results of torque control for each joins with one kg payload and parameter uncertainties. As can be seen from these results, the PI controller represents extremely good performance with very small tracking error.

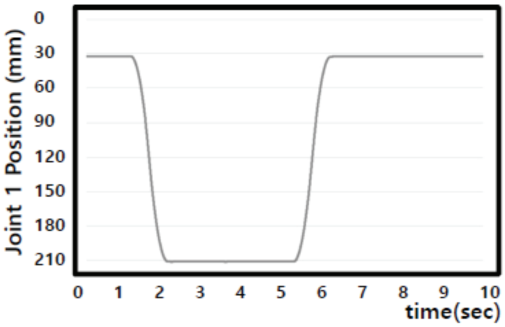


Fig. 5 The data record results of position trajectory control for joint 1

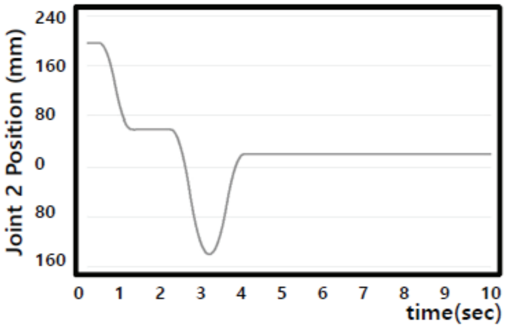


Fig. 6 The data record results of position trajectory control for joint 2

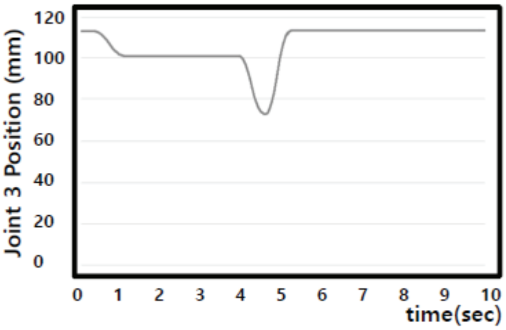


Fig. 7 The data record results of position trajectory control for joint 3

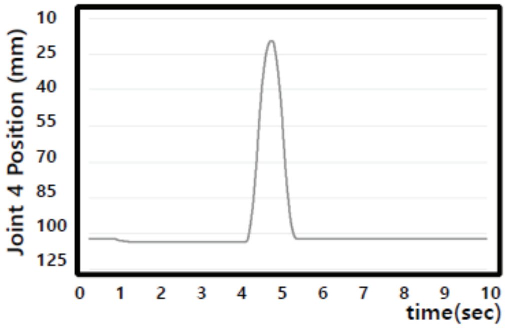


Fig. 8 The data record results of position trajectory control for joint 4

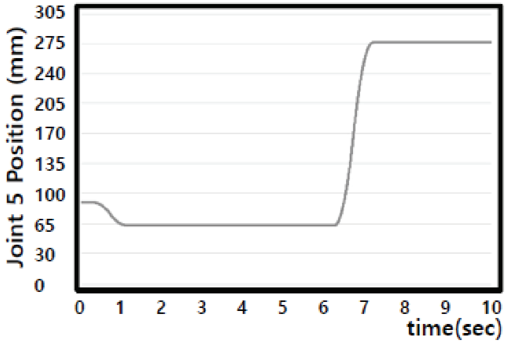


Fig. 9 The data record results of position trajectory control for joint 5

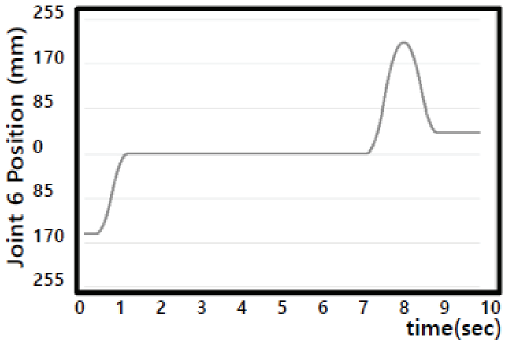


Fig. 10 The data record results of position trajectory control for joint 6

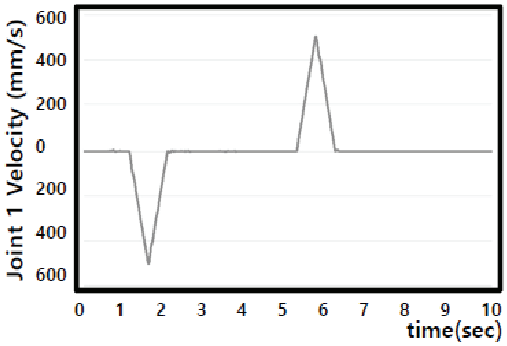


Fig. 11 The data record results of velocity trajectory control for joint 1

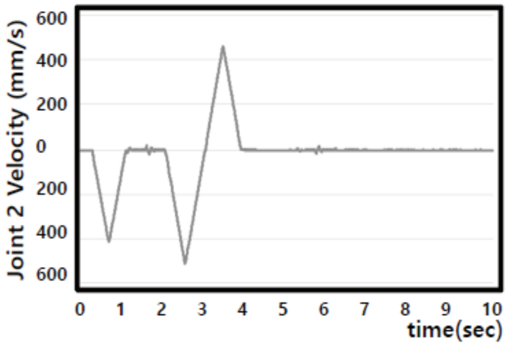


Fig. 12 The data record results of velocity trajectory control for joint 2

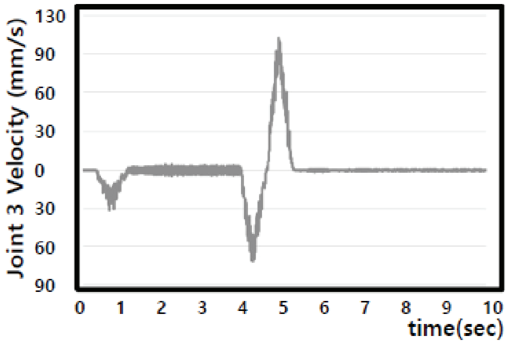


Fig. 13 The data record results of velocity trajectory control for joint 3

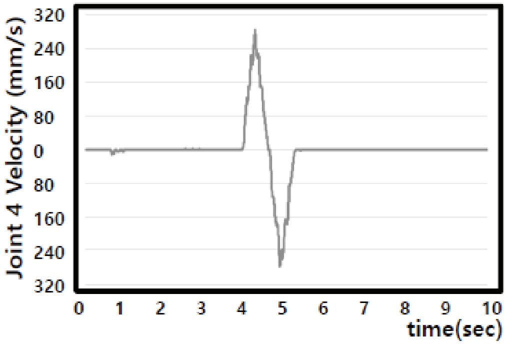


Fig. 14 The data record results of velocity trajectory control for joint 4

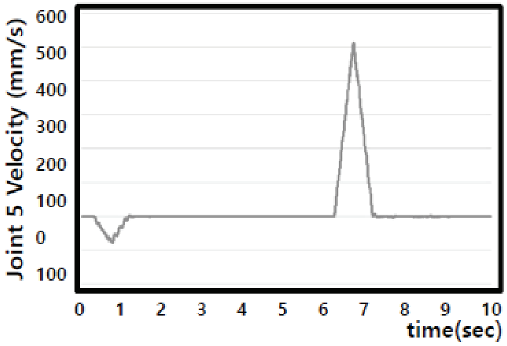


Fig. 15 The data record results of velocity trajectory control for joint 5

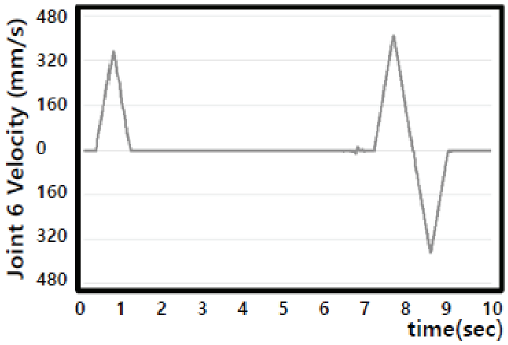


Fig. 16 The data record results of velocity trajectory control for joint 6

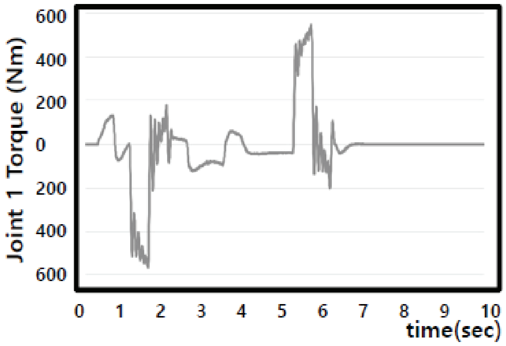


Fig. 17 The data record results of torque control for joint 1

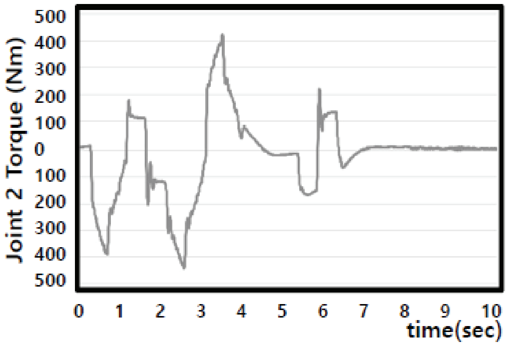


Fig. 18 The data record results of torque control for joint 2

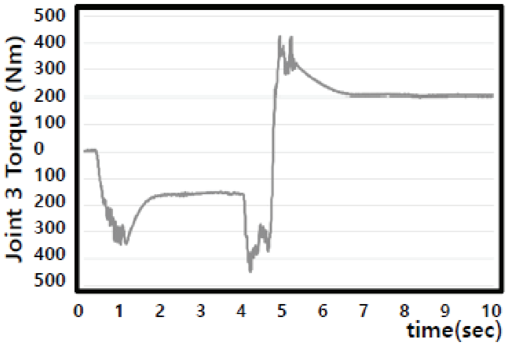


Fig. 19 The data record results of torque control for joint 3

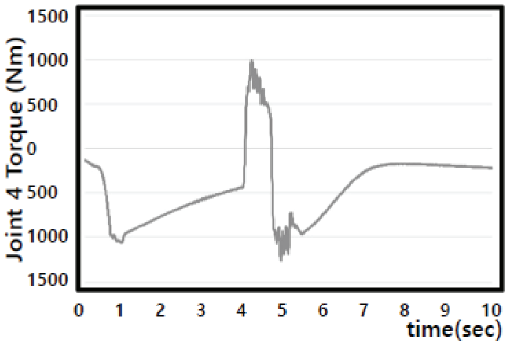


Fig. 20 The data record results of torque control for joint 4

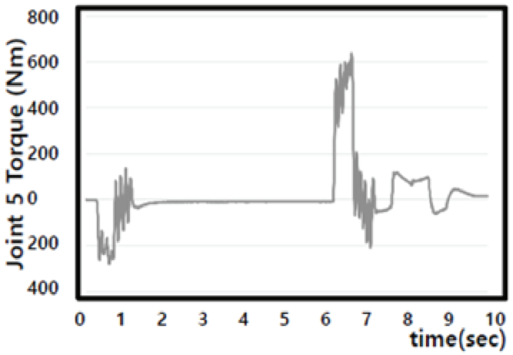


Fig. 21 The data record results of torque control for joint 5

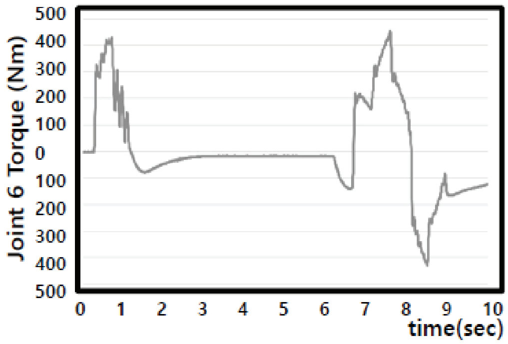


Fig. 22 The data record results of torque control for joint 6

The proposed PI controller represents good performance in the position and velocity at each joint for payload variation, inertia parameter uncertainty, and the change of reference trajectory. These simulation results illustrate that this controller is very robust and suitable to real-time control due to its fast adaptation and simple structure.

5. CONCLUSIONS

In this study, we proposed a new technology of trajectory control and track record of articulated manipulator with six joints based on monitoring simulator for smart factory.

From the simulation and experimental results the proposed monitoring simulator is good performance under the payload variation, inertia parameter uncertainty, and change of reference trajectory. This method has been found to be suitable real-time monitoring and data records of robot system. Another attractive feature of this monitoring simulator is that to verify the reliability of real time track record data of position, velocity, and torque of each joints.

In this method, it neither requires a complex mathematical model of the robotic dynamics nor any knowledge of link arm parameters and payload.

The performance of monitoring system is evaluated in track record and tracking errors of the position, velocity, and torque for the six joints.

Acknowledgments

This study was supported by Industrial Technology Innovation Project (Project Number: 20005020).

References

[1] W.S. Lee, M.S. Kim, H.Y. Bae, Y.K. Jung, Y.H June, G.S. Shin, Sung-Hyun Han, "A Study on Stable Motion Control of Humanoid Robot with 24 Joints Based on Voice Command," The Korean Society of Industry Convergence, Vo.21, no.1, pp.17-27, 20180

[2] M.S. Kim, M.H. Choi, H.Y. Bae, O.D. Im, J.S. Kang, Sung-Hyun Han, "A Study on Optimal Working Path Control of Seven Axes Vertical Type Robot with Translation Joint for Triming Working Automation in Forming Process," The Korean Society of Industry Convergence, Vo.21, no.2, pp.53-62, 2018

[3] H.S. Sim, M.S. Kim, M.H. Choi, H.Y. Bae,mH.J. Kim, D.B. Kim, Sung-Hyun Han, "A Study On Intelligent Robot Control Based On Voice Recognition For Smart FA," The Korean Society of Industry Convergence, Vo.21, no.3, pp.87-93, 2018

[4] H.S. Sim, H.Y. Bae, D.B. Kim, Sung-Hyun Han, "A Study on Flexible Control and Design of Robot Hand Fingers with Eight Axes for Smart Factory," The Korean Society of Industry Convergence, Vo.21, no.4, pp.183-189, 2018

[5] H.Y .Bae, H.J .Kim, J.I. Paeng, H.S. Sim, Sung-Hyun Han, Y.T. Baek, "A Study on Shape Recognition Technology of Die Casting and Forging Parts Based on Robot Vision for Inspection Process Automation in Limit Environment," The Korean Society of Industry Convergence, Vo.21, no.6, pp.369-378, 2018

[6] D.B. Kim, H.J. Kim, Y.T. Baek, H.Y, Bae, Sung-Hyun Han, "A Study on Obstacle Avoidance and Autonomous Travelling of Mobile Robot in Manufacturing Precess for Smart Factory," The Korean Society of Industry Convergence, Vo.21, no.6, pp.379-388, 2018

[7] D.B. Kim, H.Y. Bae, S.H. Kim, O.D. Im, Y.T. Back, Sung-Hyun Han, "A Study on Intelligent Control of Mobile Robot for Human-Robot Cooperative Operation in Manufacturing Process," The Korean Society of Industry Convergence, Vo.22, no.2, pp.137-146, 2019

[8] J.S. Kang, S.H. Noh, D.B. Kim, H.Y. Bae, S.H. Kim, O.D. Im, Sung-Hyun Han, "A Study on the Real-Tim Path Control of Robot for Transfer Automation of Forging Parts in Manufacturing Process for Smart Factory," The Korean Society of Industry Convergence, Vo.22, no.3, pp.281-292, 2019

[9] D.B. Kim, H.J. Kim, H.Y. Bae, S.H. Kim, O.D. Im, Sung-Hyun Han, J.S. Kang, S.H. Noh, "A Study on Design and Durability Analysis of Vertical Multi-Jointed Robot with Translational Joint to adapt in the High Temperature Environment," The Korean Society of Industry Convergence, Vo.22, no.3, pp.337-351, 2019

[10] S.H. Kim, D.B. Kim, H.J. Kim, O.D. Im, Sung-Hyun Han, "A Study on Real Time Control of Moving Stuff Action Through Iterative Learning for Mobile-Manipulator System," The Korean Society of Industry Convergence, Vo.22, no.4, pp.415-425, 2019

[11] H.J. Kim, S.H. Kim, G.W. Jang, D.B. Kim, G.H. Dong, Sung-Hyun Han, "A Study on Development of Robot Monotoring System Simulator for Smart Factory," The Korean Society of Industry Convergence, Vo.22, no.5, pp.561-573, 2019

[12] H.J. Kim, S.H. Kim, K.W. Jang, Sung-Hyun Han, "A Study on Motion Control and Kinematics Analysis ofArticulated Manipulator Attachment for Excavator," The Korean Society of Industry Convergence, Vo.22, no.6, pp.807-819, 2019

[13] Z. Ma, J. shen, A. Hug, and K. Nakayama, "Automatic optimum Order Assignment in PI Filters," *International conference on signal Processing Applications & Technology,* Boston pp. 629-633, October 1995.

[14] Y.K. Choi, M.J. Chang, and Z. Bien, "*An PI Control Scheme for Robot Manipulators,"* IEEE Trans. Auto. Contr., Vol. 44, No. 4, pp.

1185-1191, 1986.

[15] P. S. Pratama, A. V. Gulakari, Y. D. Setiawan, D. H. Kim, H. K. Kim, and S. B. Kim, "Trajectory tracking and fault detection algorithm for automatic guided vehicle based on multiple positioning modules," International Journal of Control, Automation and Systems, vol. 14, no. 2, pp. 400-410, (2016).

[16] S. Dubowsky, and D.T. DesForges, "The Application of Model Reference Adaptation Control to Robot Manipulators," ASME J. Dyn. Syst., Meas., Contr., Vol. 101, pp. 193-200, 1979.

[17] T.C. Hasi, "PI Control Scheme for Robot Manipulators-A Review," In Proceeding of the 1987 IEEE Conference on Robotics and Automation, San Fransisco, CA, 1986.

[18] D. Koditschek, "Quadratic Lyapunov Functions for Mecanical Systems," Technical Report No. 8703, Yale University, New Haven, CT, 1983.

[19] B. Lavanya, G.S. Gayathri, "Exploration and Deduction of Sensor-Based Human Activity Recognition System of Smart-Phone Data," Computational Intelligence and Computing Research (ICCIC)IEEE International Conference on, pp.1-5, (2017).

[20] Liu, S., An, B., Liu, S., and Guo, Z., "Characteristic Research of Electromagnetic Force for Mixing Suspension Electromagnet Used in Low-Speed Maglev Train," IET Electrical Power Applications, Vol. 9, No. 3, pp.223-228, (2015).

[21] A. Koivo and T. H. Guo, "PI Linear Controller for Robot Manipulators," *IEEE Transactions and Automatic Control*, Vol. AC-28, pp. 162-171, 1983.

[22] R. Ortega and M.W. Spong, "PI Motion Control of Rigid Robots: A Tutorial," *Automatica,* Vol. 25, pp. 877-888, 1989.

[23] P. Tomei, "PI PD Controller for Robot Manipulators," IEEE Trans. Robotics and Automation, Vol.7, No.4, Aug. 1991.

[24] S. Nicosia and P. Tomee, "Model Reference PI Control Algorithm for Imdustrial Robots," *Automatica*, Vol. 20, No. 5, pp. 635-644, 1984.

[25] N. Sadegh and R. Horowitz, "An Exponentially Stable PI Control Law for Robot Manipulators," *IEEE Trans. Robotics and Automation*, Vol. 9, No. 4, Aug. 1990.

[26] P.C.V. Parks, July 1966, "*Lyapunov Redesign of Model Reference PI Control System*," IEEE Trans. Auto. Contr., Vol. AC-11, No. 3, pp. 362-267.

(Manuscript received February 25, 2020;
revised March 30, 2020; accepted April 3, 2020)