TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 2, April 2020, pp. 1021~1029
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i2.14749

Joint control of a robotic arm using particle swarm optimization based H_2/H_∞ robust control on arduino

Petrus Sutyasadi, Martinus Bagus Wicaksono
Politeknik Mekatronika Sanata Dharma, Indonesia

Article Info

Article history:

Received Aug 7, 2019
Revised Dec 30, 2019
Accepted Feb 15, 2020

Keywords:

Arduino
H_2/H_∞ robust control
Robotic arm

ABSTRACT

This paper proposes a small structure of robust controller to control robotic arm's joints where exist some uncertainties and unmodelled dynamics. Robotic arm is widely used now in the era of Industry 4.0. Nevertheless, the cost for an industry to migrate from a conventional automatic machine to industrial robot still very high. This become a significant challenge to middle or small size industry. Development of a low cost industrial robotic arm can be one of good solutions for them. However, a low-cost manipulator can bring more uncertainties. There might be exist more unmodelled dynamic in a low-cost system. A good controller to overcome such uncertainties and unmodelled dynamics is robust controller. A low-cost robotic arm might use small or medium size embedded controller such as Arduino. Therefore, the control algorithm should be a small order of controller. The synthesized controller was tested using MATLAB and then implemented on the real hardware to control a robotic manipulator. Both the simulation and the experiment showed that the proposed controller performed satisfactory results. It can control the joint position to the desired position even in the presence of uncertainties such as unmodelled dynamics and variation of loads or manipulator poses.

Corresponding Author:

Petrus Sutyasadi,
Politeknik Mekatronika Sanata Dharma,
Paingan, Maguwoharjo, Depok, Sleman, Yogyakarta, Indonesia.
Email: peter@pmsd.ac.id

1. INTRODUCTION

During decades, many researchers have been trying to control non-linear system using linear techniques. Gain scheduling and various of H∞ approaches were used [1-4]. Robotic arm is one example of a non-linear system. Several researches tried to control it using linear approach [5-8]. All of these approaches resulted very high order of controllers that are difficult to be implemented in a small embedded system [9]. Another research synthesized a conventional high order H∞ controller and tried to reduce the order using control reduction method in Matlab [10]. The conventional high order H∞ controller results were 10th to 13th order of controllers. The reduction method resulted around 4th to 7th order of controllers which are still too high for a small embedded system like ATMEGA328 in Arduino board.

This paper proposes a low order H_2/H_∞ robust controller that is possible to be programmed on a small embedded system like Arduino board with ATMEGA328 microcontroller. The controller synthesis is explained in the paper and tested to control a low-cost small industrial or educational robotic arm joint. A low-cost robotic arm may have lower price but consequently, it may have some lower specification such as resolution, power, repeatability, and so on compared to the expensive commercial one.

Out of the specifications that actually can be chosen based on the components that are used, stability is an important parameter that cannot be bargain. Even for the low-cost robotic arm, guaranty of robust stability and robust performance is a must. H_∞ under bound uncertainty is a controller that guarantee robust stability [11-14]. $H\infty$ controller works as an optimal controller that does not provide the best performance of the system but provides the optimal performance in the range of uncertainties [15]. To guarantee robust performance as well, H_2/H_∞ mixed sensitivity is used. $H\infty$ controller synthesis always results a very high order of controller that is not easy to implement in practical embedded system moreover in a small controller like Arduino. Arduino is a small low-cost controller that now is very popular among practitioners. The proposed controller is a low order controller which has predefined structure that can guarantee robust stability and robust performance. The structure was specified before the controller synthesis, and the order of the controller was also set to meet the specification of the controller hardware.

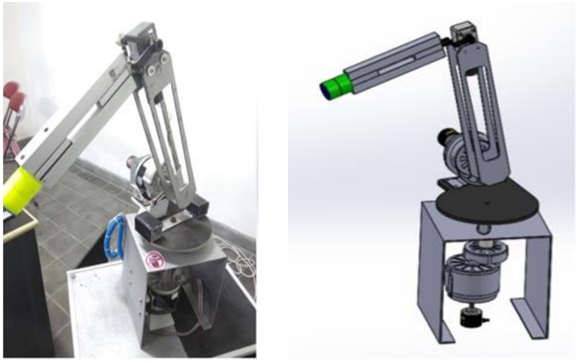
The chosen structure in this research is proportional integral derivatif (PID) structure. The reason is because PID controller has been widely used in variety application [16]. Other than that, PID is good enough for linear position control and used a lot in robotic control. Even the structure is just in the form of PID controller or some people may ask why the so-called $H\infty$ controller now just a PID controller, it has been optimized using robust controller cost functions. It no longer gives the best response system in terms of rise time, settling time, and overshoot, but it provides stability and guarantees the performance even in the presence of uncertainties or disturbances. Nonetheless, the controller will perform satisfactorily if the uncertainties and disturbances remain in the same set range when the controller was designed. PID controller needs to be re-tuned everytime there are changes in the system or environtment. Whereas robust control gain is applicable in all cases [17]. The reason in choosing H_∞ controller is because its capability in handling disturbance cancelation and robust stabilization of uncertain system. However, the drawback is in its transient response behavior [18]. Another robust controller with simple structure is sliding mode controller. But this controller has a chattering effect when the sliding surface is reached [19].

Comparisons of proposed controller and regular PID controller is presented in this paper. The proposed controller synthesis and the comparison to PID controller were tested on the base join of the robot. Load variations were applied during the comparison to see the controller performance under uncertainties. Novelty that is proposed in this paper is a small order with simple structure of H_2/H_∞ robust controller. Therefore, the controller is possible to be implemented in a small microcontroller like Arduino.

2. RESEARCH METHOD

2.1. Hardware design of the robotic arm

The robotic arm is an articulated 3DoF manipulator. The detail of the design is shown in Figure 1. The robot manipulator is made of steel and mostly aluminium. Detail specification is provided in Table 1. For a low-cost robotic arm manipulator design, brushed dc motor is used here instead of brushless dc motor. Brushed dc motor is less expensive but brushed dc motor actually has several advantages such as higher efficiency and reliability, have longer lifespan, and faster torque response [20]. The low-level position control is implemented in Arduino. Two Arduino UNOs are needed to control all the three manipulator's joints. One UNO is used for inverse kinematic and control the base motor. The other one is used to control the hip and knee motors. The two UNOs talk each other using I2C protocol. Wiring diagram of the electric components is shown in Figure 2.



(a) (b)

Figure 1. (a) Robot prototype, (b) Mechanical design of 3DoF articulated robotic arm

Table 1. Hardware specification of the robotic arm

Specification	Value
Max vertical stretch	1005 mm
Max horizontal stretch	641 mm
Base Motor Power & Speed	250W/270 rpm
Shoulder Motor Power & Speed	250W/270 rpm
Knee Motor Power & Speed	250W/31 rpm
Degree of Freedom	3

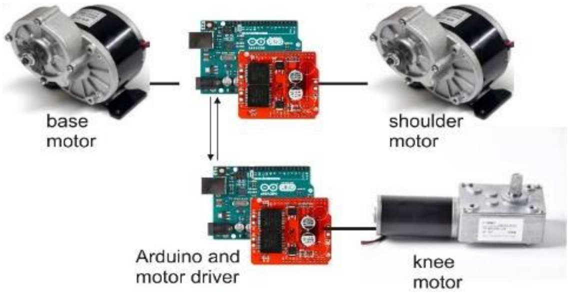


Figure 2. Controller connection of the robotic arm

2.2. Kinematic of the robotic arm

The kinematic coordinate system of the robot is shown in Figure 3. Based on the reference coordinate system, the Denavit-Hartenberg parameters are provided in Table 2. Inverse kinematic of the robot manipulator was derived using this Denavit-Hartenberg parameters. Based on the calculation of inverse kinematic of an articulated manipulator in [21], the joint angles are derived from the known end point position (x, y, z).

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \tag{1}$$

$$\theta_2 = -\tan^{-1}\frac{a_1 - (x\cos\theta_1 + y\sin\theta_1)}{-z} + \tan^{-1}\left(\frac{A}{\pm\sqrt{r^2 - A^2}}\right) \tag{2}$$

$$\theta_3 = \tan^{-1}\left(\frac{z - a_2\sin\theta_2}{x\cos\theta_1 + y\sin\theta_1 - a_2\cos\theta_2 - a_1) - \theta_2}\right) \tag{3}$$

where:

$$A = \frac{2a_1(x\cos\theta_1 + y\sin\theta_1) + a_3^2 - a_2^{22}}{2a_2} + \frac{-a_1^2 - z^2 - (x\cos\theta_1 + y\sin\theta_1)^2}{2a_2} \tag{4}$$

$$r = \pm\sqrt{-z^2 + (a_1 - (x\cos\theta_1 + y\sin\theta_1))^2} \tag{5}$$

which $a_{i,}$ $\alpha_{i,}$ $d_{i,}$ *and* θ_i are the Denavit Hartenberg parameter shown in the Table 2 and Figure 3.

2.3. Identification model of the robot manipulator

The system was investigated as a decoupled system. Each motor was loaded with the rest of the manipulator frame to the arm tip included the rest of the dc motor and its gearbox. Since the investigation is firstly focused on the motor base, then the model derived is the equation of the dc motor base with the rest of the manipulator as its load. Equivalent of electrical circuit and mechanical diagram of a DC Motor is shown in Figure 4. The load it self is set to the maximum when the robot is stretched horizontally. The effect of load change can influence the movement of robotic arm manipulator [22]. Nominal transfer function dc motor position control and additional load inertia is obtained as:

$$\frac{\Theta(s)}{V(s)} = \frac{0.331}{1.635\exp-11s^{3} + 1.288\exp-05s^{2} + 0.1096\,s} \tag{6}$$

Table 2. D-H Parameter of the robotic arm

Parameter	Link 1	Link 2	Link 3
a_i	-	a_2	a_3
α_i	$\pi/2$	0	0
d_i	0	0	0
θ_i	θ_1	θ_2	θ_3

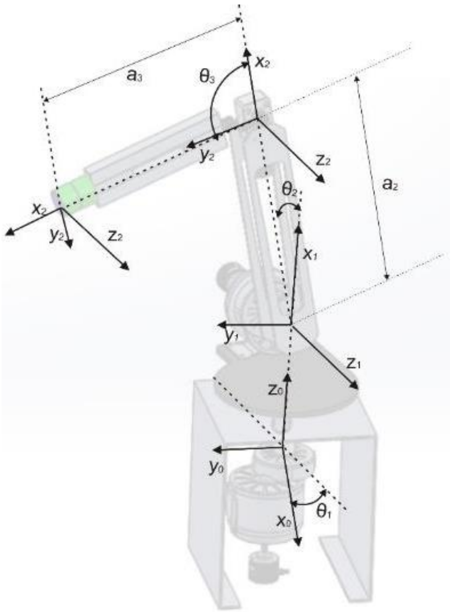


Figure 3. Reference coordinate systems on the robotic arm

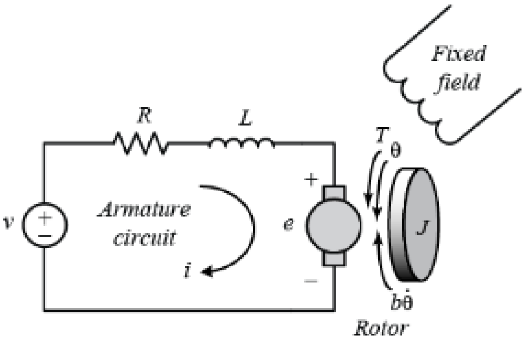


Figure 4. DC motor electrical and mechanical diagram

2.4. 2Structure specified mixed sensitivity H₂/H∞ controller

Robust stability and robust performance against external disturbance will satisfy (7) if such controller K(s) is designed so that the closed loop of the nominal system is asymptotically stable [23].

$$J_{\infty,a} = \left\| W_s(s) S(s) \right\|_{\infty} < 1 \tag{7}$$

Multiplicative perturbation is upper bounded by stable function Wt(s). While the external disturbances is attenuated by astable function Ws(s). The robust stability against system perturbation satisfies the following inequality:

$$J_{\infty,b} = \left\| W_t(s) T(s) \right\|_{\infty} < 1 \tag{8}$$

Ws(s) : sensitivity weight
Wt(s) : complementary sensitivity weight
S(s) : sensitivity function
T(s) : complementary sensitivity or transmissibility function.

The idea of structure specified mixed sensitivity *H₂/H∞* control is to find an admissible structure-specified controller that minimizes the cost function J₂ subjected to both constraints of J∞,a and J∞,b.

$$J_2 = \int_0^{\infty} e^2(t)\,dt = \left\| E(s) \right\|_2^2 \tag{9}$$

Following Skongestad's method [13], after parameters substitution, the sensitivity weight for the base, shoulder, and knee motors is:

$$W_s = \frac{0.5s+1}{s+0.001} \tag{10}$$

The complimentary sensitivity weight (Ws) of the base control system is obtained as:

$$W_{t_h} = \frac{1.26s^2+57.22s+8.79}{s^2+183.5s+3768} \tag{11}$$

The plot of the complimentary sensitivity weight and the system uncertainty singular value is shown in Fig. Figure show that the singular values of the inverse of the weight functions is larger than the sensitivity and the complementary sensitivity singular values which confirms |Ws S|<1 and |Wt T|<1. In this research, a PID controller structure is selected as the structure of the proposed controller.

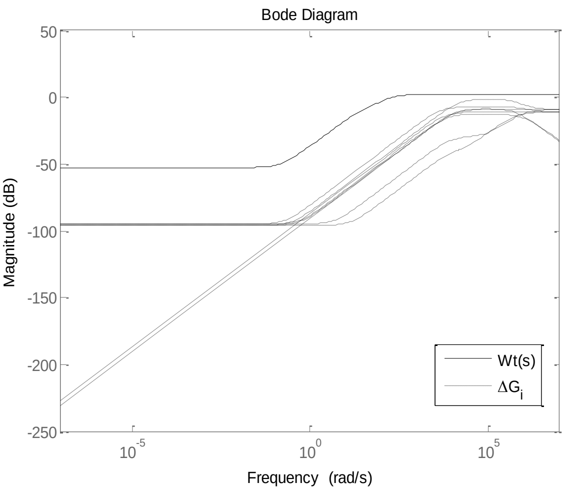


Figure 5. Complimentary sensitivity weight Wt and the system uncertainty of the base joint control

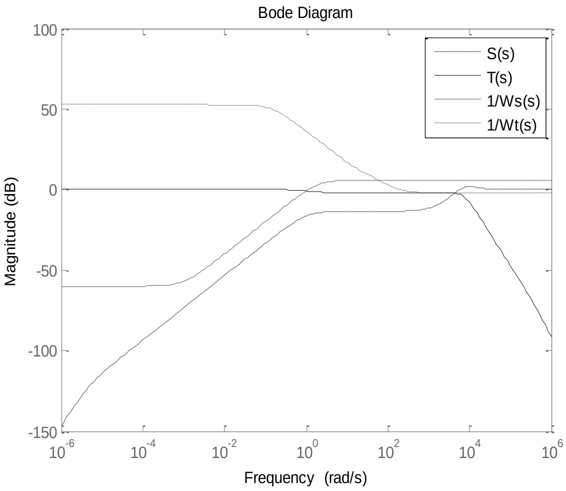


Figure 6. The sensitivity, complementary sensitivity, and their inverse weight singular values of the base joint

$$K(s) = Kp + \frac{Ki}{s} + \frac{Kd.s}{Ns+1} \tag{12}$$

2.5. Particle swarm optimization

Particle Swarm Optimization or PSO is an evolutional method that adopt social behavior of birds to optimize objective function [24]. In this research, PSO is used to minimize the objective function J_2 from (9) subjected to both constraints of $J_{\infty,a}$ and $J_{\infty,b}$ from (7) and (8). As long as the constraints of $J_{\infty,a}$ and $J_{\infty,b}$ are fulfilled, the minimization of J_2 will generate the swarm's positions as the controller constants. Any repetition of the PSO optimization process with any parameter changes, will generate different controller constants combination. As far as all the J_∞ constraints are fulfilled, each combination will form an optimal robust controller that satisfy the robust stability and robust performance against external disturbance [13]. A group of random particles is used to initialized the PSO at the first time. In each iteration, PSO searches for optima by updating generations using best fitness value or the best position of particle *$P_i(k)$* and best particles value or the best global position *$G(k)$*. In every updating process iteration, PSO updates the velocity and the position following (13) and (14).

$$v_i(k+1) = w.v_i(k) + c_1.r_1.\left(P_i(k) - x_i(k)\right) + c_2.r_2.(G(k) - x_i(k)) \tag{13}$$

$$x_i\left(k+1\right)=x_i\left(k\right)+v_i(k) \quad (14)$$

The search space is N-dimensional, so the dimension of position vector of particle i is also N, $x_i=(x_{i1},x_{i2},...,x_{iN})$ The dimension of velocity vector is N, $v_i=(v_{i1},v_{i2},...,v_{iN})$. The objective function evaluates the fitness of particles. Individual best position $p_i=(p_{i1},p_{i2},...,p_{iN})$ is the best previous position of particle i. The w is inertia weight, r1 and r2 are random variables ranged between 0 to 1, c1 and c2 are coefficients of acceleration. The velocity is kept in the range of –vmin to vmax. Global best position is the best individual position of the whole swarm $G=(g_1,g_2,...,g_N)$. The velocity and the position of the particle are updated every iteration. Flow chart of PSO is depicted in Figure 7.

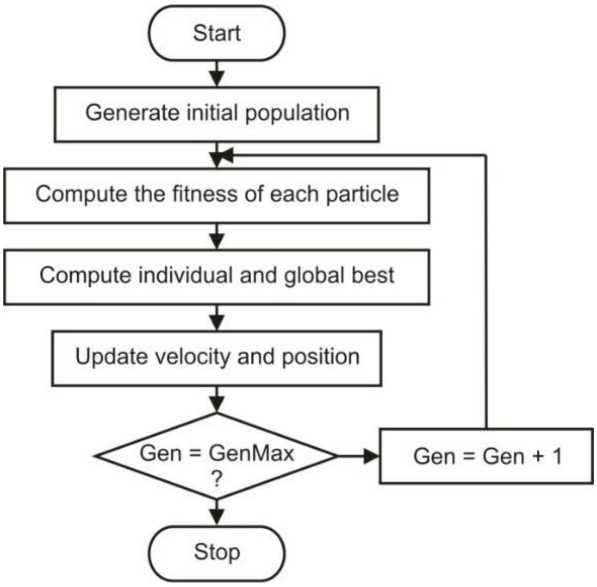

Start
Generate initial population
Compute the fitness of each particle
Compute individual and global best
Update velocity and position
Gen = GenMax ?
Gen = Gen + 1
Stop

Figure 7. Flowchart of particle swarm optimization

The parameters of the PSO are set as follows: swarm size = 20, dimension of the particle is 4 (kp, ki, kd, and td), c1 = c2 = 2, number of maximum generation = 100. The inertia weight is changed from 0.95 to the final weight 0.4, and the velocity is limited at [-vmin,vmax] = [-100,100]. Following the controller structure in (12), the kp, ki, kd, and td are replaced with the results from the PSO. The best particles from the iteration is put on the equation to replace the four PID parameter. This generates one combination of best H_2/ H_∞ robust controller in the form of simple PID structure. One combination result is obtained as:

$$K\left(s\right)=11.554+\frac{0.0000069}{s}+\frac{1,234s}{0.000735s+1} \quad (15)$$

With the $J_{\infty,a}$ = 0.61121, $J_{\infty,b}$ = 0.9963, and J_2 = 0.1503.
The optimization process is not put on the Arduino embedded system due to its large search space and long fitness calculation [25].

3. RESULTS AND ANALYSIS

3.1. Simulation result

The proposed controller showed in (15) is a robust controller in a simple PID structure. Figure 8 shows that the proposed controller can control the robot’s base joint under various uncertainties. The simulation was conducted under uncertainties as shown in Figure 5. In term of stability, all the responses are stable, have good rise time, no overshoot and steady state error which is important in robotic arm control [26]. The result was compared to other three controllers derived using MATLAB toolboxes which are:

- Full order H_∞ robust controller derived using MATLAB command “mixsyn”,
- Structure specified H_∞ robust controller derived using MATLAB command “*hinfstruct*”,
- Auto tuned Proportional Integral Derivative (PID) Controller.

Figure 9 shows response system using full order H_∞ robust controller. A classic synthesis of high order H_∞ robust controller can be derived automatically using MATLAB command “mixsyn”. The result controller using “mixsyn” command is:

$$\frac{1.546\times 10^{5}.s^{5}+1.218\times 10^{4}.s^{4}+1.059\times 10^{16}.s^{3}+1.907\times 10^{2}+3.906\times 10^{19}+4.001\times 10^{11}}{s^{6}+9.741\times 10^{5}.s^{5}+1.708\times 10^{11}.s^{4}+1{,}498\times 10^{16}.s^{3}+2.827\times 10^{2}.s^{2}+5.901\times 10^{19}.s+5.898\times 10^{16}} \tag{16}$$

It is a six order of controller. The plant itself is a third order transfer function. Therefore, the controller combines with the plant itself result a nine-order system which is actually very difficult to be implemented on an embedded system controller. However, in this case we just compared it in simulation using MATLAB. The response of the system is quite stable, except the rise time and the settling time is still too long. Figure 10 shows response system using another structure specified H_∞ robust controller but it was derived directly using MATLAB command “*hinfstruct*”. The controller equation is:

$$K(s)=10.1037+\frac{25.5478}{s}+0.1578s \tag{17}$$

The structure chosen is also PID structure. The result is also quite stable. However, it has overshoot. Even small overshoot, but it is quite dangerous for some robotic arm application if we have overshoot. The end effector may have hit the target before it reached the position. Lastly, we generated an auto tune PID controller using MATLAB. The controller equation is:

$$K(s)=7172.4416+\frac{1087598.31}{s}+0.5531\frac{37214.7841}{1+37214.78\times\frac{1}{s}} \tag{18}$$

The controller was tuned in nominal value of the plant. Later on, the controller was tested with the same plant but the same uncertainties were applied on the system. The response system are shown in Figure 11. The response of the system in the nominal plant was very good. However, in some uncertainties, there were some significant overshoots.

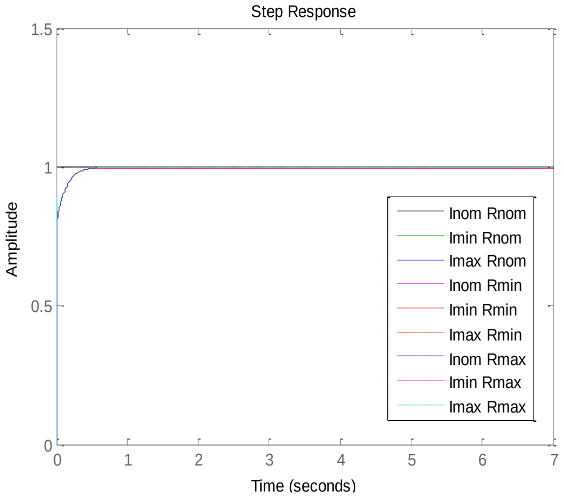


Figure 8. System response using proposed controller

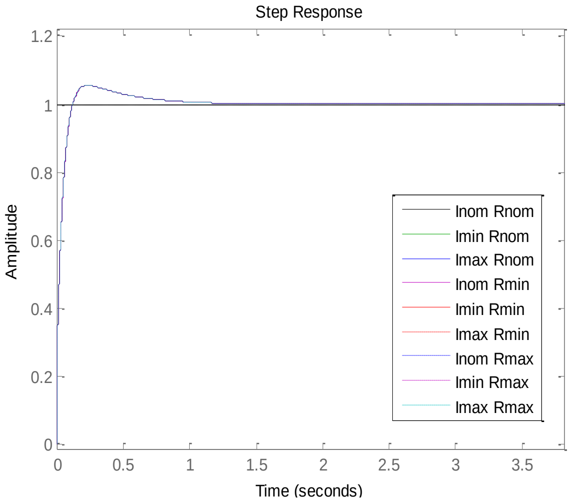


Figure 9. System response using a full order H_∞ robust controller

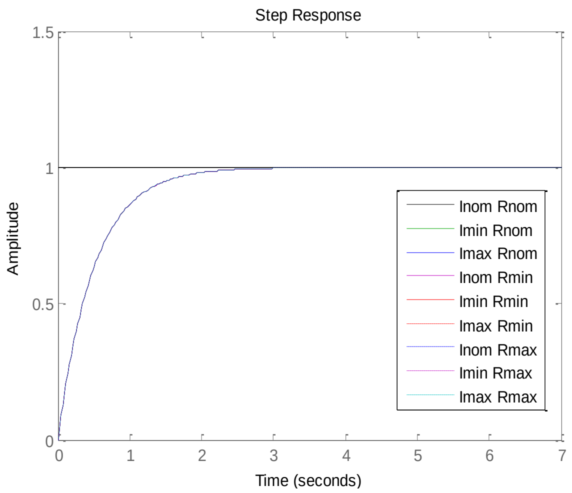


Figure 10. System response using structure specified H_∞ robust controller

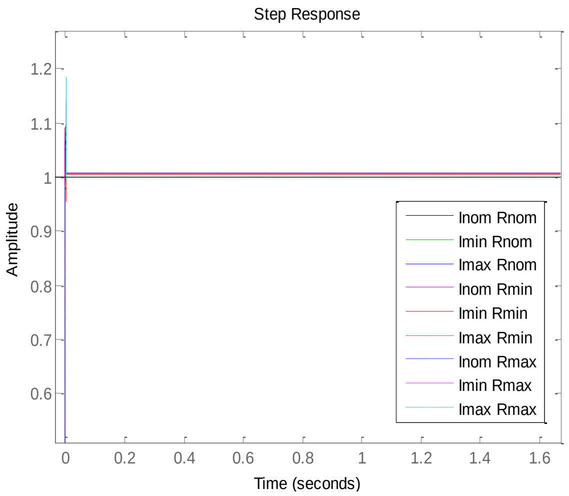


Figure 11. System response using auto tuned PID controller

3.2. Experimental Result

The synthesized controller was programmed into the controller hardware. The hardware response system is shown in Figure 12. There were some steady state errors due to some uncertainties. However, the resolution of the encoder is very small. The maximum error that happened for the maximum load and the arm is stretched horizontally is less than 0.1°. All the arm stretched horizontally is a rare configuration in daily work of the robot. So, in normal works the robot still has good position control repeatability.

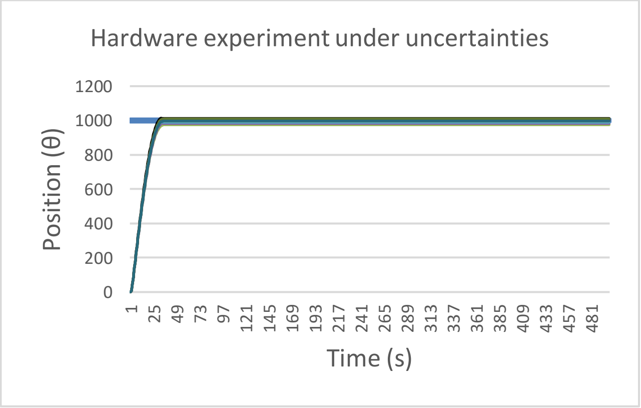


Figure 12. Hardware response system using the proposed controller

4. CONCLUSION

The 3DoF articulated robot manipulator has been developed. A low order H_2/ H_∞ robust controller in the form of simple PID structure has been successfully synthesized. The proposed controller was sintesized using Particle Swarm Optimization with integral of squared error cost function in the constraint of robust stability and robust performance functions. Computer simulations and hardware experiments showed that the controller is able to control the robot's base joint motor position even in the presence of uncertainties. The proposed controller also gave better responses compared to a full order H_∞ robust control, structure specified H_∞ robust controller, and an auto tune PID controller in the presence of uncertainties. In the hardware simulation there was a small steady state error maximum around 0.1° that happened at the maximum uncertainty. For certain application this value is still acceptable. Especially for educational robot, it still can be used for teaching and learning equipment. Future work is to synthesis another robust controller to control the rest of the joints. For the hardware, additional two or three degree of freedom manipulator will be added on the tip of the robot. It will control the orientation of the end effector.

ACKNOWLEDGEMENTS

The Authors would like to acknowledge the reserach funding support from Ministry of Research, Technology, and Higher Education of the Republic of Indonesia (Kemen-RISTEKDIKTI)

REFERENCES

[1] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H∞ control of linear parameter-varying systems," *Proc. IEEE American Contr. Conf., (Baltimore, USA)*, pp. 856–860, 1994.

[2] P. Apkarian and R. Adams, "Advanced gain-scheduling techniques for uncertain systems," *IEEE Trans. Contr. Syst. Techn.*, vol. 6, 1997, pp. 21–32.

[3] G. Angelis, "System Analysis, modelling and control with polytopic linear models - PhD thesis," Technische Universiteit Eindhoven, The Netherlands, 2001.

[4] V. Montagner, R. Oliveira, V. Leite, and P. Peres, "LMI approach for H∞ linear parameter-varying state feedback control," *IEE Proc. Part D, Contr. Theory & Applications*, vol. 152, vol. 2, pp. 195-201, 2005.

[5] T. Namerikawa, M. Fujita, and F. Matsumura, "H∞ control of robot manipulator using a linear parameter varying representation," *Proc. IEEE American Contr. Conf.,* (Albuquerque, USA), 1997.

[6] Z. Kang, Y. Yin, S. Fujii, and T. Chai, "Gain-scheduling H∞ vibration control for scara type robot manipulators via lmis," *Proc. IEEE American Contr. Conf.,* (Albuquerque, USA), 1997.

[7] Z. Kang, T. Chai, K. Oshima, J. Yang, and S. Fujii, "Robust vibration control for scara-type robot manipulators," *Control Engineering Practice*, vol. 5, 1997, pp. 907–917

[8] Z. Yu, H. Chen, and P. Woo, "Gain scheduled LPV H∞ control based on lmi approach for a robotic manipulator," *Journal of Robotic Systems*, vol. 19, pp. 585–593, 2002.

[9] Harouna Souley Ali, Latifa Boutat-Baddas, Yasmina Becis-Aubry, Mohamed Darouach, "H-infinity control of a SCARA robot using polytopic LPV approach," 2006.

[10] Axelsson P., Helmersson A., Norrlof M., "$\mathcal{H}_\infty$-Controller Design Methods Applied to One Joint of a Flexible Industrial Manipulator," 19th IFAC World Congress, Cape Town, South Africa, 2014

[11] Aoi S., "Simple Legged Robots that Reveal Biomechanical and Neuromechanical Functions in Locomotion Dynamics," *Journal of Robotics and Mechatronics*, vol. 26, no. 1, pp. 98-99, 2014.

[12] Fukuoka Y, Katabuchi H, Kimura H., "Dynamic Locomotion of Quadrupeds Tekken 3&4 Using Simple Navigation," *Journal of Robotics and Mechatronics*, vol. 22, no. 1, pp. 36-42, 2010.

[13] Zang ZG, Kimura H, Fukuoka Y., "Self-Stabilizing Dynamics for a Quadruped Robot and Extension Toward Running on Rough Terrain," *Journal of Robotics and Mechatronics*, vol. 19, no. 1, pp. 3-12. 2007.

[14] Mamiya S, Sano S, Uchiyama N., "Foot Structure with Divided Flat Soles and Springs for Legged Robot and Experimental Verification," *Journal of Robotics and Mechatronics*, vol. 28, no. 6, pp. 799-807. 2016.

[15] Sutyasadi P, Parnichkun M., "Gait Tracking Control of Quadruped Robot Using Differential Evolution Based Structure Specified Mixed Sensitivity Robust Control," *Journal of Control Science and Engineering*, vol. 2016, pp. 1-18, 2016.

[16] Lekkala K., Mittal V., "PID controlled 2D precision robot. 2014 International Conference on Control, Instrumentation," *Communication and Computational Technologies*, ICCICCT 2014, pp. 1141-1145, 2014. DOI: 10.1109/ICCICCT.2014.6993133

[17] Khairudin M., Mohamed Z., and Husain A.R., "Dynamic Model and Robust Control of Flexible Link Robot Manipulator," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 9, no. 2, pp. 279-286, August 2011.

[18] Tumari M.Z.M., Subki S.R.A., Aras M.SM., Kasno A.K., Ahmad M.A., Suid M.H., "H-infinity controller with graphical LMI region profile for liquid slosh suppression," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol.17, no.5, pp.2636-2642, October 2019. DOI: 10.12928/TELKOMNIKA.v17i5.11252

[19] Mousmi A., Abbou A., Houm Y.E., "Real-time implementation of a novel hybrid fuzzy sliding mode control of a BLDC motor," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 10, no. 3, pp. 1167-1177, Sep 2019. DOI: 10.11591/ijpeds.v10.i3.1167-1177

[20] Tarmizi Y., Jidin A., Karim K.A., Sutikno T., "A simple constant switching frequency of direct torque control of brushless DC motor," *International Journal of Power Electronics and Drive System (IJPEDS),* vol. 10, no. 1, pp. 10-18, March 2019. DOI: 10.11591/ijpeds.v10.i1.pp10-18

[21] P G Santos, E Garcia, J Estreera, "Quadrupedal Locomotion," *Springer*, 2006.

[22] Pambudi W.S., Alfianto E., Rachman A., Hapsari D.P., "Simulation design of trajectory planning robot manipulator," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 196-205, March 2019. DOI: 10.11591/eei.v8i1.1179

[23] S. Skogestad, I. Postlethwaite, "Multivariable Feedback Control," *New York: John Wiley & Sons*. 2001

[24] J Kennedy, R Ebenhart, "Particle swarm optimization," *IEEE International Conference of Neural Network*, pp. 1942-1948, 1995.

[25] Harun S., Ibrahim M.F., "A genetic algorithm based task scheduling system for logistics service robots," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 1, pp. 206-213, March 2019. DOI: 10.11591/eei.v8i1.1437

[26] Mahamed HWA, "Designing and implementation of PID controller robotic arm," Dissertation, Sudan University of Science and Technology, College of Graduate Studies, 2018.