Journal of Engineering and Sustainable Development
Vol. 29, No. 05, *August* 2025
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.2103

An Adaptive Robust PID Controller Design for a Single Motorized Robotic Arm

Ekhlas H. Karam[1], Abeer K. Al-Gburi[1], Yousra A. Mohammed[2*]

[1]Department of Computer Engineering, College of Engineering, Mustansiriyah University, Baghdad, Iraq
[2]Department of Communication Engineering, University of Technology, Baghdad, Iraq

[*]Email: yousra.a.mohammed@uotechnology.edu.iq

Article Info

Received 28/07/2023
Revised 19/04/2025
Accepted 22/08/2025

Abstract

A robotic arm can be defined as a programmable robot manipulator that functions similarly to a human arm. In general, a DC motor precisely moves the robot arm to a desired angular position in accordance with a given input; hence, a suitable control scheme must be used. This study examines three control methodologies for a Single Motorized Robotic Arm (SMRA) with a single degree of freedom (DoF): Robust PID (RPID), Adaptive Robust PID (ARPID), and a newly formulated Modified Adaptive Robust PID (MARPID). The suggested MARPID takes the strong points of RPID and adds a better adaptation law to make the system respond faster and lower the steady-state error. The performance of the RPID, ARPID, and MARPID controllers has been analyzed and compared using MATLAB and Simulink version 2015a. Simulation results of linear and nonlinear input showed that the developed MARAPID outperformed RPID and ARPID compared to the RPID or APID controllers.

Keywords: Adaptive PID, DC motor, Robust PID, Single robot arm, Tuned method.

1. Introduction

Robotic arms are widely used in industry, and they are robotic manipulator that consists of links connected by joints. These manipulators can do either rotational or translational movements. Commonly, the robotic arm is designed to perform many tasks depending on the application, such as assembly, welding, spinning welding, gripping, painting, polishing, transporting materials around a building, grinding, etc. [1]. An autonomous repetitive robot is a robot that can execute one task repeatedly according to preplanned movements and fixed objects.

Start and stop commands are found based on robot position, acceleration, deceleration, distance, and direction. Meanwhile, for more complex actions, such as when the object orientation or position is unknown, machine vision and artificial intelligence are used to help the robot locate the object and control the robotic arm. Various control designs for autonomous robots have been developed to overcome problems such as robot localization, motion planning, and working in different environments under different constraints. Among these designs, robust and adaptive controllers are well considered in relevant literature [2].

Usually, the performance of the robot arm is found according to the specifications of the DC motors used to drive the robot [3]. Being able to control the robot arm motion accurately is a fundamental concern. To obtain an efficient control system operation, it is required to know the exact amount of torque and force (generated by the DC motor) applied at a given time to be able to move the robot arm to a desired location [4]. Due to this, studying DC motors for this purpose has become an active area of research. However, much of this research is focused only on the performance of the DC motor model, ignoring the effect of the arm load. This had resulted in a gap between the simulation findings and the actual performance of the robot arm [3], as illustrated in ref. [5] by W.M. Wan et al., and ref. [6] by Widi A. and et.al.

Hence, not many controls design work is included for DC motor model with load (robot arm) which focuses on reducing the differences between real and simulation results, as in [7] and [8], where artificial neural network is used instead of conventional PID controller (that is used by [9]) for single motorized robot arm. Therefore, in this paper, we suggest a new controller that can be used to control a DC motor model with a load (robot arm).

In this paper, the model of a signal joint horizontal robot arm driven by a DC motor is first built using MATLAB Simulink. To improve the response of the arm, Robust PID (RPID), Adaptive Robust PID (ARPID), and a Modified Adaptive Robust PID (MARPID) are suggested to control this arm. The results obtained were used to compare these methods and identify the most effective controller.

This paper is organized as follows: The model of the SMRA is first presented in Section 2. The procedure for designing the RPID is described in Section three. Section four establishes the details of the suggested MARPID. The simulation results for the overall controlled motorized robot arm with uncertainty and disturbance are presented in Section 5. Finally, conclusions are drawn in section six.

2. Single Motorized Robot Arm Description and Modeling

The robot arm considered in this paper is a single-joint robot arm system, shown in Fig. 1. This system consists of a robot arm, connected horizontally to an actuator using a gear train with a gear ratio, *n* [10]. The dynamic equation of the robot arm is given as [11]:

$$T_m(t) = J\frac{d^2\theta_m}{dt^2} + D\frac{d\theta_m}{dt} \quad (1)$$

Where θ_m is the motor shaft angle position. Meanwhile, J represents all inertia, and D all friction (air friction, bearing friction, etc.) connected to the motor shaft. Taking the Laplace transformation for (1):

$$T_m(s) = Js^2\theta_m(s) + Ds\,\theta_m(s) \quad (2)$$

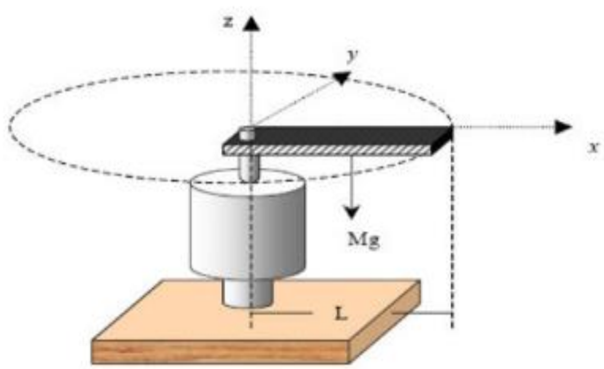


Figure 1. A signal joint robot horizontal arm driven by a DC motor [1].

To drive the single robot arm, a DC motor with armature control and a fixed field is used [11]. Here, the armature voltage control scheme for the DC motor considers varying the voltage applied to the armature without varying the voltage applied to the field [12].

Fig. 2 shows the electrical model for this DC motor. The parameters of the considered DC motor are given in Table 1.

In this table, the armature voltage ($E_a(t)$) is the voltage supplied by an amplifier to control the motor.

The back emf (k_b) is proportional to the speed of the motor with a fixed field strength. That is [9]-[11], [13]:

$$v_b(t) = k_b\frac{d\theta_m}{dt} \quad (3)$$

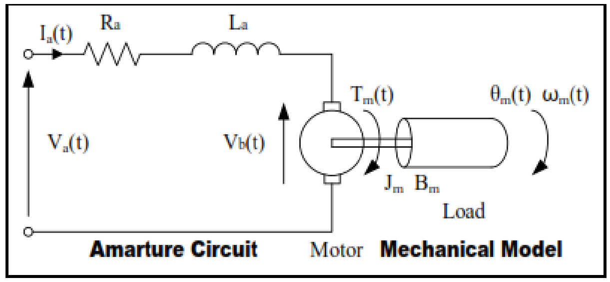


Figure 2. The electrical model of a DC motor [5].

Table 1. The DC motor model parameters.

Symbol	Description	Value	Unit
R_a	armature resistance	1	Ω
K_b	electromotive force constant	1	V-s/rad
I_a	Armature current	-----	A
K_t	Motor torque constant	1	N-m/A
J_m	moment of inertia of the rotor	0.001	Kg- m^2
ω_m	Angular Speed	-----	*rad/s*
V_a	Armature voltage	-----	V
D_m	motor viscous friction constant	0.01	Nms/rad
L_a	electric inductance	0	H

Taking the Laplace transformation for (3), this gives:

$$V_b(s) = K_b s\theta_m(s) \quad (4)$$

The electrical portion of the motor equation is:

$$V_a(s) = R_a I_a(s) + L_a s I_a + V_b(s) \quad (5)$$

This may also be written as;

$$I_a(s) = \frac{V_a(S) - K_b s\theta_m(s)}{L_a s + R_a} \quad (6)$$

And the torque, supplied by the motor, which moves the armature and load, is proportional to the armature current [11]:

$$T_m(s) = K_t I_a(s) \quad (7)$$

Now, solving (2) for the shaft angle, we get

$$\theta_m(s) = \frac{T_m(s)}{Js^2 + Ds} \quad (8)$$

Using (3)-(8), the block diagram for the armature-controlled DC motor can be built in MATLAB Simulink as shown in Fig. 3. Simplifying the block diagram in Fig. 3, the armature-controlled motor transfer function can be obtained as:

$$G(s) = \frac{\theta_L(s)}{E_a(s)} = \frac{K_t n}{s[(Js+D)(L_a s+R_a)+k_b k_t]} \quad (9)$$

This model is a third-order system; however, in the servomotor case, the inductance of the armature L_a can usually be ignored. Thus, the model can be reduced to a second-order system [11].

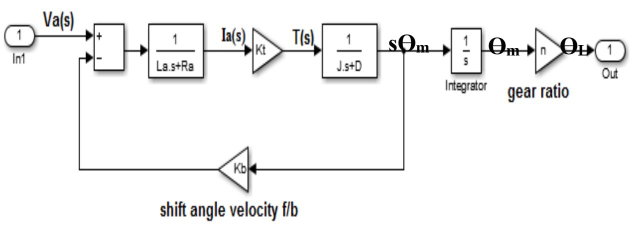


Figure 3. Armature-controlled DC motor block diagram.

Now, suppose the robot arm in Fig. 1 has a length $L = 1\ m$, a viscous damping factor $D = 0.1$, a mass $M = 5$ kg, and consider a system with a gear ratio (n) = 1. The inertia of the rigid arm is defined as [11].

$$J_L = \frac{ML^2}{12} = \frac{5\times 1^2}{12} = 0.4167\ kg\ m^2 \quad (10)$$

The total inertia J and total damping factor D of the robot arm (J_L, D_L), and the motor parameters (J_m, D_m) in Table 1 are [11]:

$$J = J_m + J_L = 0.4177\ kgm^2 \quad (11)$$

$$D = D_m + D_L = 0.01 + 0.1 = 0.11 \quad (12)$$

Substituting the DC motor parameters in Table 1 and the parameters of (11) and (12) into (9), we obtain:

$$G(s) = \frac{\theta_L(s)}{V_a(s)} = \frac{1}{s[(0.4177s + 0.11)(1 + 0 \times s) + 1 \times 1]}$$

$$G(s) = \frac{1}{s(0.4177s+1.11)} = \frac{2.394}{s(s+2.6574)} \quad (13)$$

This equation shows that the motorized arm plant has two roots, one at the origin and the other on the left-hand side of the s-plane ($s = -2.6574$).

3. Robust PID Controller Design

Generally, the control law for an ideal PID controller is [14], [15]:

$$u(s) = G_c(s)e(s) = (k_p + \frac{k_i}{s} + k_d s)\, e(s) \quad (14)$$

Where $e(s)$ is the error signal, in this paper, $e(s)$ will be considered as the difference between the $r(s)$ signal (desired position) and *the* $y(s)$ signal (actual position) of the robot arm. On the other hand, k_p, k_i, and k_d are the proportional, integral, and derivative gains, respectively. In literature, many methods were used to tune the parameters of the PID controller. The controller design approach presented in [16] is considered in this paper. This approach ensures low sensitivity and robust stability for the overall controlled system for a wide range of parameter uncertainties. In this paper, (14) can be rearranged to be:

$$u(s) = G_c(s)e(s) = k_d(\frac{s^2+as+b}{s})\, e(s) \quad (15)$$

where $a = k_p/k_d$ and $b = k_i/k_d$. Hence, the PID controller has one pole (s=0) at the origin and two zeros which can be placed on the left-hand side of the s-plane. Here, the value of k_d and the location of the two zeros are chosen so that the resulting settling time (t_s=4/ $\zeta\,\omega_n$) is ≤ 1 sec and the damping ratio is ($\zeta \geq 1$). For example, if k_d is selected to be equal to 5 and the real zeros are chosen to be (z_1 -8.8730, z_2 = -1.1270) with ζ=1.58, then t_s will be 0.8 and the overall open loop transfer function becomes:

$$G_n(s) = G_c(s)G(s)$$

$$= \frac{5(s+8.873)(s+1.127)}{s}\cdot\frac{2.394}{s(s+2.6574)}$$

$$= \frac{5(s^2+10s+10)}{s}\cdot\frac{2.394}{s(s+2.6574)} \quad (16)$$

The parameters ((k_p,and k_i) can be determined by comparing $G_c(s)$ of (15) with $G_c(s)$ of (16), therefore:

$a = k_p/k_d = 10 \longrightarrow k_p = 50$

$b = k_i/k_d = 10 \longrightarrow k_i = 50$

Plotting the root locus for the system in (16), as shown in Fig. 4, demonstrates that the robot arm model will be stable with the designed robust PID controller for any positive k_d value.

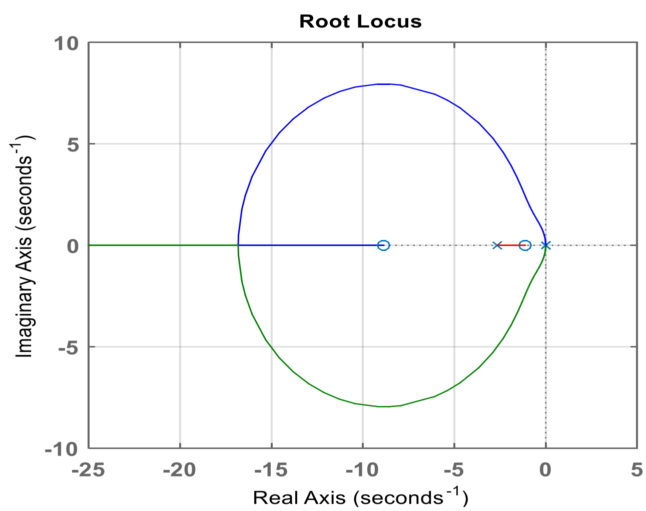


Figure 4. Root locus plot of the open-loop transfer function $G_n(s)$.

4. Modified Adaptive Robust PID (MARPID) Controller Design

The block diagram for the suggested MARPID for a single motorized robot arm (SMRA) is shown in Fig. 5. This controller is designed based on RPID and adaptation laws of APID.

The control law of the APID in the time domain is [17]:

$$u(t) = \widehat{k_p}(t) + \widehat{k_i}(t)\int_0^t e(t)dt + \widehat{k_d}(t)\frac{de(t)}{dt} \quad (17)$$

Where ($\widehat{k_p}(t)$, $\widehat{k_i}(t)$, and $\widehat{k_d}(t)$) are respectively, the proportional, integral, and derivative estimated controller parameters. Using the law of chain and the gradient descent approaches, these parameters are found by the following adaptation laws [18]-[20]:

$$\dot{\widehat{k_p}}(t) = \beta_p s(t) e(t) \quad (18)$$

$$\dot{\widehat{k_i}}(t) = \beta_i s(t) \int_0^t e(t)\, d(t) \quad (19)$$

$$\dot{\widehat{k_d}}(t) = \beta_d s(t) \frac{de(t)}{dt} \quad (20)$$

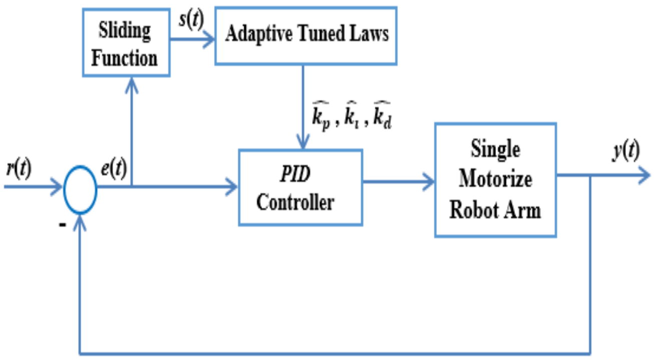


Figure 5. Block diagram of the suggested MARPID for SMRA.

Here, β_p, β_i, and β_d are positive learning values, and $s(t)$ is the sliding function defined as:

$$s(t) = c_1 e(t) + \dot{e}(t) \quad (21)$$

c_1 is a positive constant value. The learning values (β_p, β_i, and β_d) should be selected correctly to achieve optimal system performance. Then, these values should be adjusted manually or using a suitable optimization technique [17].

In this work, the control law for the suggested MARPID depends on RPID (17) and the adaptation laws of the APID parameters. (18)-(20). Then, the control law of the APID (17) is modified to:

$$u(t) = \widehat{k_d}(t)[\hat{a}(t) + \hat{b}(t) \int_0^t e(t)dt + \frac{de(t)}{dt}] \quad (22)$$

The estimated parameters ($\hat{a}(t)$, $\hat{b}(t)$, and $\widehat{k_d}(t)$) are updated according to the following new proposed adaptation laws:

$$\hat{a}(t) = a_0 + u_a(t) \quad (23)$$

$$\hat{b}(t) = b_0 + u_b(t) \quad (24)$$

$$\widehat{k_d}(t) = k_{d0} + u_d(t) \quad (25)$$

Where a_0 =a, $b_0 = b$, $k_{d0} = k_d$, while $u_a(t)$, $u_b(t)$, and $u_d(t)$ are:

$$u_a(t) = \int_0^t \beta_p s(t) e(t)\, dt + u_a(t-1) \quad (26)$$

$$u_b(t) = \int_0^t (\beta_p s(t) \int_0^t e(t)\, d(t))\, dt + u_b(t-1) \quad (27)$$

$$u_d(t) = \int_0^t (\beta_p s(t) \frac{de(t)}{dt})\, dt + u_a(t-1) \quad (28)$$

With the sliding function equal to:

$$s(t) = e(t) + c_2 \int_0^t e(t)dt) \quad (29)$$

And c_2 is a positive constant value.

5. Simulation Results and Discussion

In this section, simulation trials are performed using MATLAB facilities (version R2015a). The efficiency of the RPID, APID, and MARPID was evaluated by calculating the peak time t_p, maximum peak M_p, settling time ts based on 2% criteria, error at steady state $e_{s.s}$, control signal at steady states us.s, in addition to the Integral Time Absolute Error (ITAE), which is formed as:

$$ITAE = \int_0^\infty t|e(t)|\, dt \quad (29)$$

The RPID was designed based on $k_d = 5$, and two different zero locations are given in Table 2.

Table 2. The controller parameters of the RPID controller.

RPID	z1	z2	k_p	k_i
RPID1	-8.873	-1.127	50	50
RPID2	1	2	-15	10

The RPID2 was simulated to observe the performance of the SMRA system controlled by RPID, APID, and MARPID when the zero locations are chosen to be on the right side. Here, kd =5 and the real zeros are chosen as (z_1 +1, z_2 = +2), therefore the overall open loop transfer function becomes:

$$G_n(s) = \frac{5(s^2-3s+2)}{s} \cdot \frac{2.394}{s(s+2.6574)} \quad (30)$$

By comparing $G_c(s)$ of (15) with $G_c(s)$ of (30), the parameters of k_p and k_i are:

$a = k_p/k_d = -3 \longrightarrow k_p = -15$

$b = k_i/k_d = 2 \longrightarrow k_i = 10$

The parameters β_p, β_i, and β_d for the APID were selected equal to k_p, k_i, *and* k_d values of the RPID, respectively, i.e., ($\beta_p = k_p = 50$, $\beta_i = k_i = 50$, $\beta_d = k_d = 5$) based on RPID1 and ($\beta_p = k_p = -15$, $\beta_i = k_i = 10$, $\beta_d = k_d = 5$) based on RPID2. Meanwhile, for the suggested MARPID, these values were selected as $\beta_p = 1$, $\beta_i = 5$, $\beta_d = 1$. Also, the values chosen for $c_1 = 2$ and $c_2 = 5$.

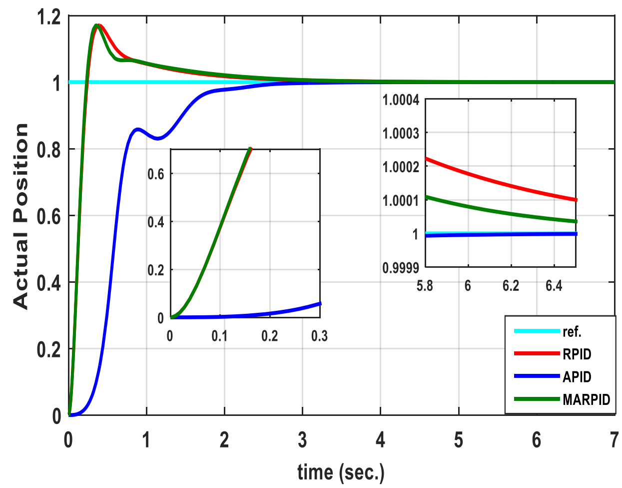
To test the efficiency of these controllers, the following simulation cases were investigated:

5.1. Simulation Result for Linear Position Signal Based on Designed RPID1

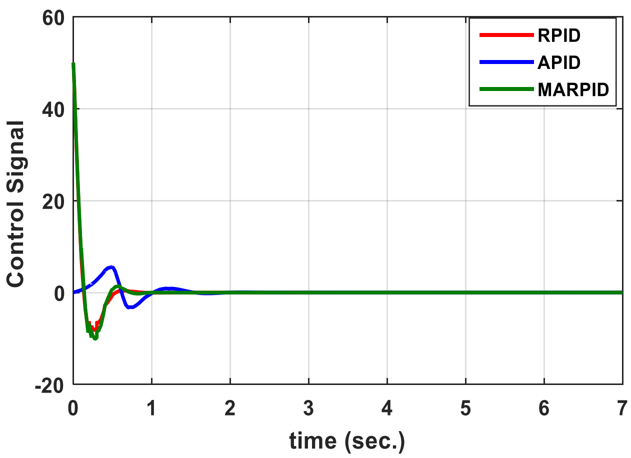
In this case, two linear desired position inputs (continuous unit step signal and mixed step signal) are used to test the performance of the SMRA system controlled by the RPID, APID, and MARPID controllers. The simulation results, which are the actual position signal, control signal, and ITAE for these cases, are shown in Figs. 6 and 7. For the continuous unit step input, the performance evaluation parameters for the SMRA with the suggested controllers are shown in Table 3.

Table 3. The performance evaluation parameters for SMRA with RPID, APID, and MARPID controllers for unit step input.

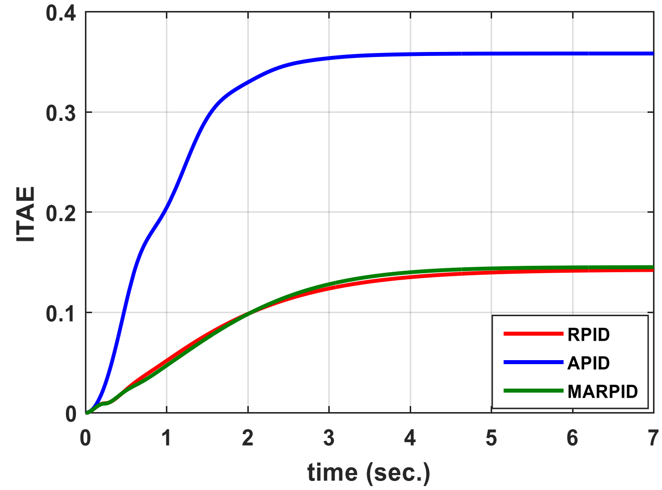
Control method	M_p	t_p	t_s	$e_{s.s}$
RPID	0.1677	0.394	1.9	0.0001
APID	0.0	------	2.1716	0
MARPID	0.1678	0.3467	2.03	0



(a)

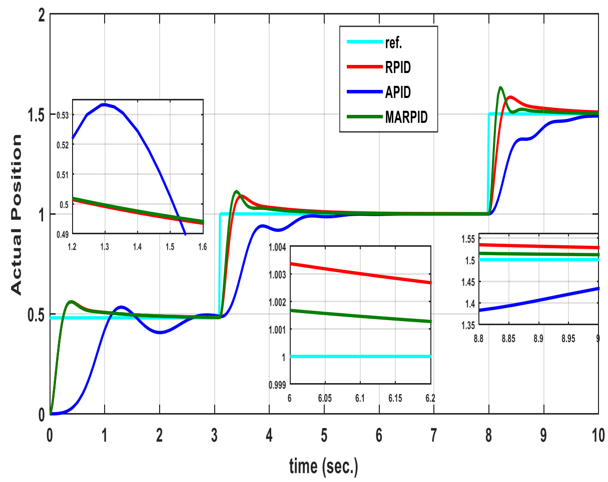


(b)

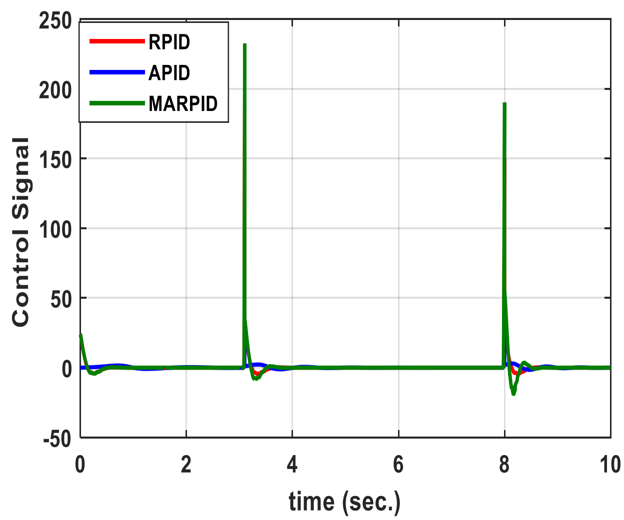


(c)

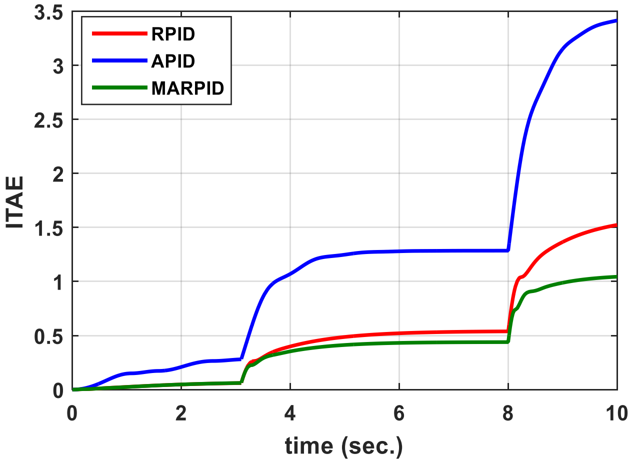
Figure 6. The response of the SMRA system controlled by RPID, APID, and MARPID controllers for a unit step input, where (a) actual position, (b) control signal, and (c) ITAE performance index.



(a)



(b)



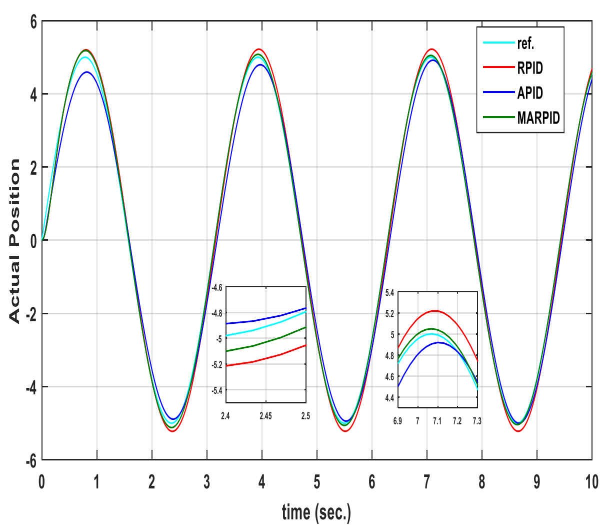
(c)

Figure 7. The response of the SMRA system controlled by RPID, APID, and MARPID controllers for the desired input rate, where (a) actual position, (b) control signal, and (c) ITAE performance index.

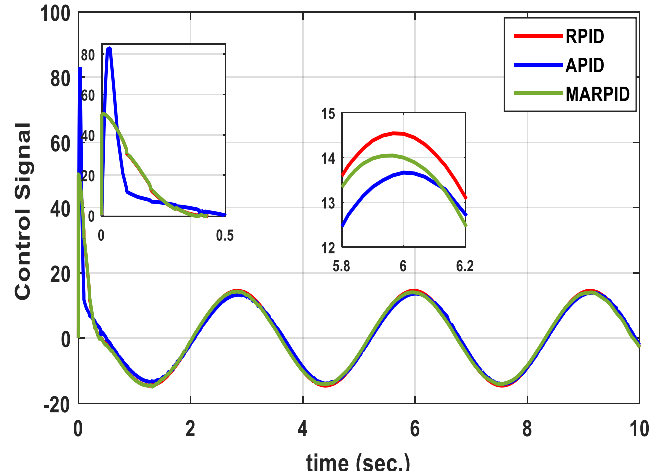
Simulation results show that the suggested controllers RPID, APID, and MARPID successfully made the system follow the desired position quickly (with small t_s) and had nearly zero es.s, slight overshoot (M_p), and an acceptable control signal with minimum IATE, especially when using the MARPID controller.

5.2. Simulation Result for Nonlinear Position Signal based on the designed RPID1

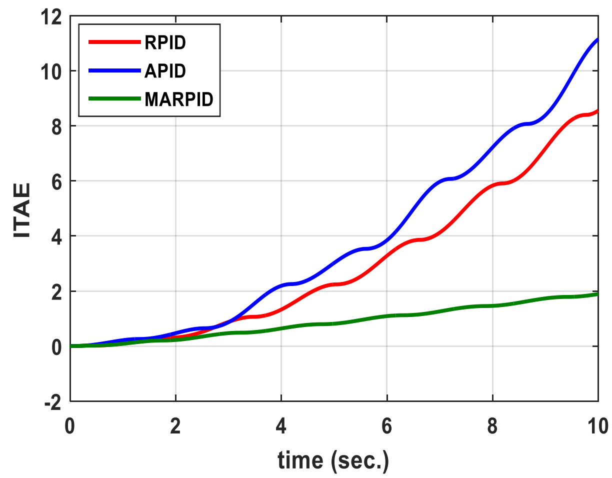
The performance of the SMRA system controlled by RPID, APID, and MARPID controllers is tested using the nonlinear input signal $r(t) = 5\sin(2t)$. The simulation results for this case, as well as the actual position signal, control signal, and ITAE, are shown in Fig. 8. As can be seen in this figure, the performance of the SMRA system with the MARPID controller is much improved over its performance with RPID or APID controllers, where the actual robot position follows the desired input very closely with an appropriate control signal and minimum ITAE.



(a)



(b)

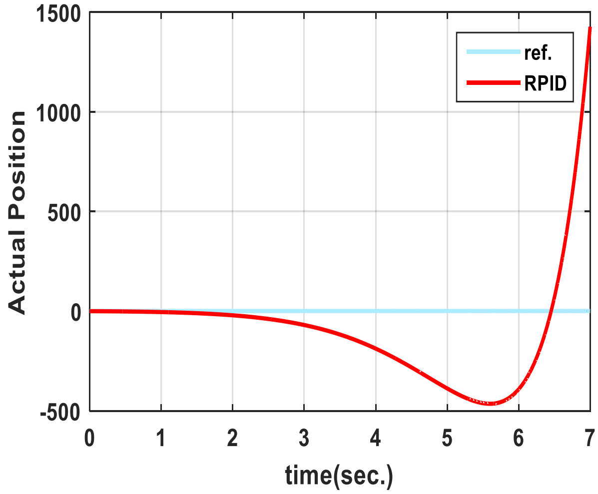


(c)

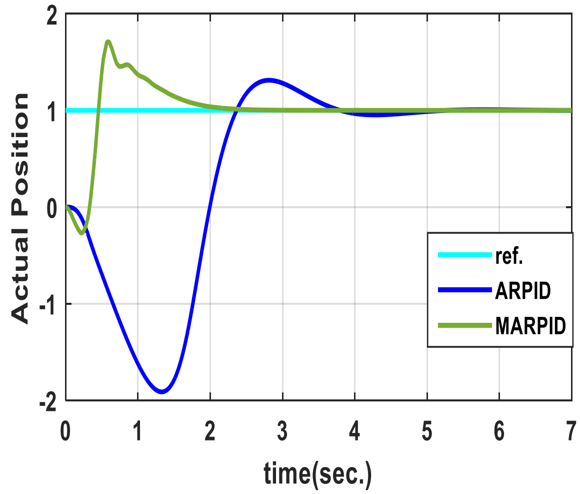
Figure 8. The response of the SMRA system controlled by RPID, APID, and MARPID controllers for a nonlinear desired input, where (a) actual position, (b) control signal, and (c) ITAE performance index.

5.3. Simulation Result for Linear and Nonlinear Position Signal Based on Designed RPID2

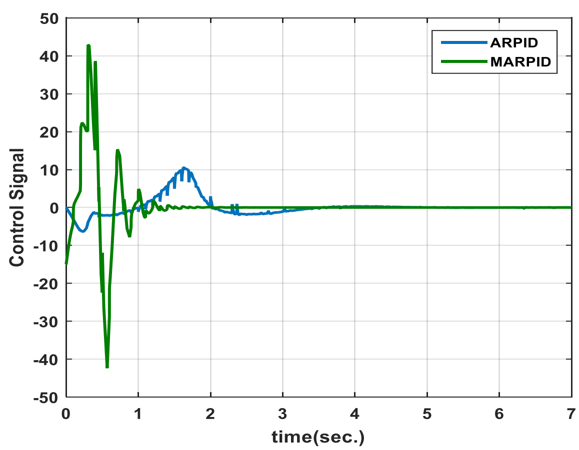
For the continuous unit step and the nonlinear signal $r(t) = 5\sin(2t)$, the simulation results are shown in Fig. 9 and Fig. 10. As can be seen from these figures, the performance of the SMRA system using the MARPID controller has improved a lot compared to its performance using the ARPID controller with an appropriate control signal and minimum ITAE. The response of the SMRA system using the RPID becomes unstable for both input signals.



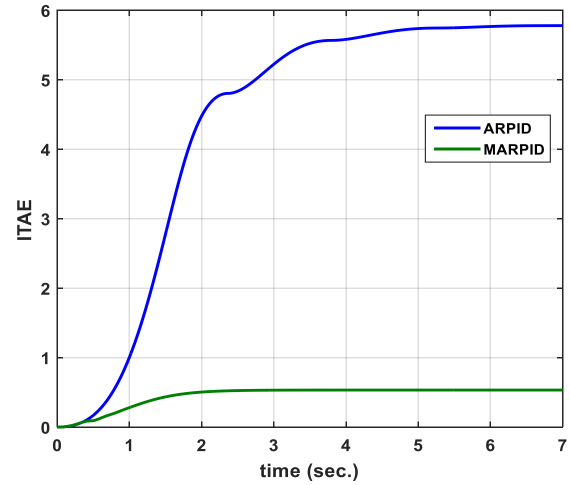
(a)



(b)

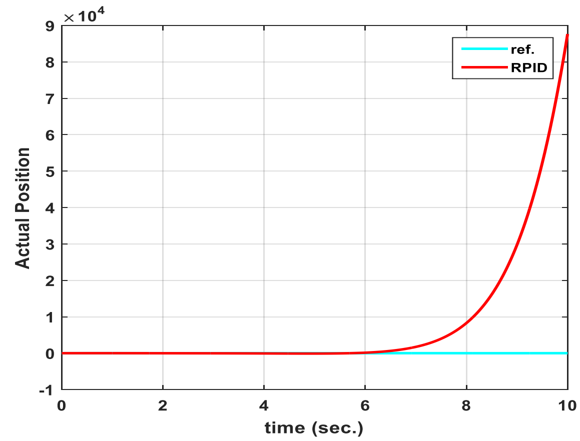


(c)

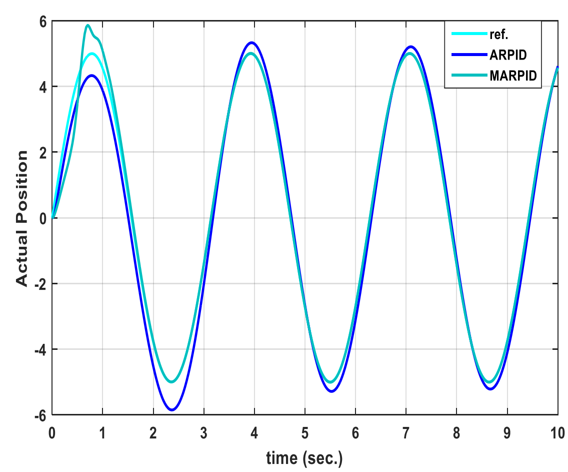


(d)

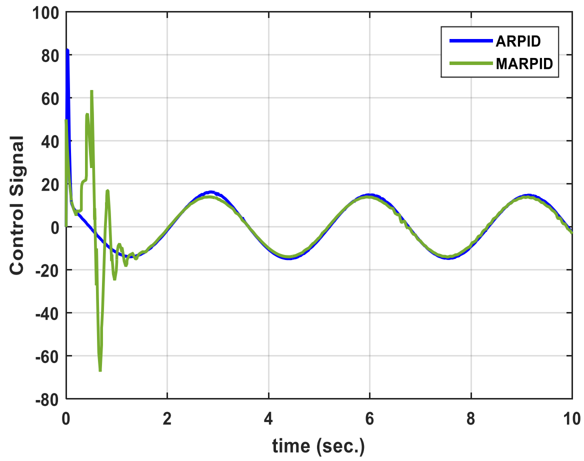
Figure 9. The response of the SMRA system controlled by RPID, APID, and MARPID controllers for linear desired input, where (**a**) actual position with RPID, (**b**) actual position with ARPID and MARPID, (**c**) control signal, and (**d**) ITAE performance index.



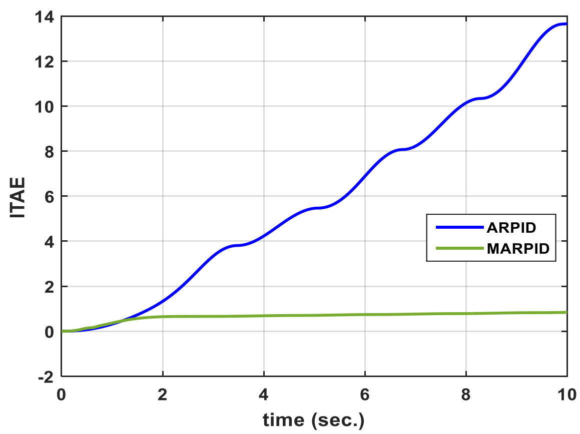
(a)



(b)



(c)



(d)

Figure 10. The response of the SMRA system controlled by RPID, APID, and MARPID controllers for a nonlinear desired input, where (**a**) actual position with RPID, (**b**) actual position with ARPID and MARPID, (**c**) control signal, and (**d**) ITAE performance index.

6. Conclusions

In this article, first, an adaptive PID controller has been developed for a single motorized robot arm system. The performance for this controller has been evaluated using the following measures: peak time, maximum peak value, settling time, and error at steady state for the controlled system, as shown in Table 3.

To improve the performance of the APID controller, a Modified Adaptive Robust PID controller was proposed. This controller was built depending on the RPID equation and the adaptation laws of the APID parameters.

The simulation results showed that the proposed MARPID controller had more effectively controlled the SMRA system compared to the RPID or APID controllers, as given in Table 3, especially in the linear position input case. For the nonlinear position input, the MARPID controller had a better response, since the actual robot position closely followed the desired input with an appropriate control signal and minimum ITAE.

The results in this paper can be applied to build robust controllers for other types of robotic systems, including nonlinear models for these systems.

Conflict of interest

The authors confirm that the publication of this article does not cause any conflict of interest.

Author Contribution Statement

The Author's Contributions to this paper are as follows:

• Author Ekhlas H. Karam & Abeer K. Al-Gburi: proposed the research problem.

• Author Abeer K. Al-Gburi developed the theory and performed the computations.

• Authors Ekhlas H. Karam & Yousra A. Mohammed: verified the analytical methods and supervised the findings of this work.

• All authors discussed the results and contributed to the final manuscript.

References

[1] K. M. Kumar and C. Sambaiah, "Modelling and study of motion control system for motorized robot arm using MATLAB and analysis of the arm by using Ansys," *Int. J. Mech. Ind. Eng.*, vol. 2, no. 1, pp. 142–147, 2012. https://doi.org/10.47893/ijmie.2012.1084

[2] T. Chatchanayuenyong and M. Parnichkun, "Neural network-based time optimal sliding mode control for an autonomous underwater robot," *Mechatronics*, vol. 16, no. 8, pp. 471–478, 2006. https://doi.org/10.1016/j.mechatronics.2006.02.003

[3] L. Bai, "Electric drive system with BLDC motor," in *2011 Int. Conf. Electr. Inf. Control Eng.*, 2011, pp. 359–363. https://doi.org/10.1109/iceice.2011.5777507

[4] C. H. Aung, K. T. Lwin, and Y. M. Myint, "Modeling motion control system for motorized robot arm using MATLAB," *Eng. Technol.*, vol. 1, pp. 142–147, 2008. https://doi.org/10.47893/ijmie.2012.1084

[5] W. S. Wan Mohd, *Linear quadratic Gaussian (LQG) controller design for servo motor*. Thesis, Univ. Tun Hussein Onn, Malaysia, 2013. [Online]. Available: http://eprints.uthm.edu.my/id/eprint/6837

[6] W. Aribowo, S. Supari, and B. Suprianto, "Optimization of PID parameters for controlling DC motor based on the aquila optimizer algorithm," *Int. J. Power Electron. Drive Syst.*, vol. 13, no. 1, pp. 216–222, 2022. https://doi.org/10.11591/ijpeds.v13.i1

[7] X. D. Su, Q. S. Luo, and W. Zhi, "Improved single-neuron PID controller for DC motor applied in robot system," *J. Beijing Inst. Technol. (Engl. Ed.)*, vol. 21, no. 4, pp. 472–478, 2012.

[8] M. Aamir, "On replacing PID controller with ANN controller for DC motor position control," *Int. J. Res. Stud. Comput.*, vol. 2, no. 1, pp. 21–29, 2013. https://doi.org/10.5861/ijrsc.2013.236

[9] F. A. Salem, "Modeling, simulation, controller selection and design of electric motor for mechatronics motion applications, using different control strategies and verification using MATLAB/Simulink," *Eur. Sci. J.*, vol. 9, no. 27, pp. 211–238, 2013.

[10] A. A. Mahfouz, M. K. M., and F. A. Salem, "Modeling, simulation and dynamics analysis issues of electric motor, for mechatronics applications, using different approaches and verification by MATLAB/Simulink," *Int. J. Intell. Syst. Appl.*, vol. 5, no. 5, pp. 39–57, 2013. https://doi.org/10.5815/ijisa.2013.05.06

[11] R. L. Shell and E. L. Hall, *Handbook of industrial automation*. New York: CRC Press, 2000. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Handbook+of+Industrial+Automation#0

[12] D. Xue, C. Zhao, and Y. Q. Chen, "Fractional order PID control of a DC-motor with elastic shaft: A case study," in *Proc. Amer. Control Conf.*, 2006, pp. 3182–3187. https://doi.org/10.1109/acc.2006.1657207

[13] S. Y. Moradi, "Controlling DC motor position, using a PID controller made by a PIC microcontroller," *ZANCO J. Pure Appl. Sci.*, vol. 28, no. 2, pp. 82–89, 2016.

[14] A. Adnan and E. H. Karam, "Optimal improved PID controller with GOA algorithm for single link human leg robot," *J. Eng. Sustain. Dev.*, vol. 26, no. 2, pp. 103–110, 2022. https://doi.org/10.31272/jeasd.26.2.10

[15] E. H. Karam, N. S. Abdul-Jaleel, and R. S. Habeeb, "Design an optimal PID controller with nonlinear function using bacteria foraging optimization for a single flexible link robot manipulator," *J. Eng. Sustain. Dev.*, vol. 21, no. 4, pp. 14–29, 2017.

[16] I. J. Nagrath and M. Gopal, *Control systems engineering*, 3rd ed. New Delhi: New Age Int., 2002.

[17] A. Noordin, M. A. Mohd Basri, and Z. Mohamed, "Adaptive PID control via sliding mode for position tracking of quadrotor MAV: Simulation and real-time experiment evaluation," *Aerospace*, vol. 10, no. 6, p. 512, 2023. https://doi.org/10.3390/aerospace10060512

[18] W.-D. Chang and J.-J. Yan, "Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems," *Chaos Solitons Fractals*, vol. 26, no. 1, pp. 167–175, 2005. https://doi.org/10.1016/j.chaos.2004.12.013

[19] A. Noordin, M. A. Mohd Basri, and Z. Mohamed, "Position and attitude tracking of MAV quadrotor using SMC-based adaptive PID controller," *Drones*, vol. 6, no. 9, p. 263, 2022. https://doi.org/10.3390/drones6090263

[20] A. Noordin, M. A. Mohd Basri, and Z. Mohamed, "Real-time implementation of an adaptive PID controller for the quadrotor MAV embedded flight control system," *Aerospace*, vol. 10, no. 1, p. 59, 2023. https://doi.org/10.3390/aerospace10010059