cigre Turkey

TEPES
TURKISH JOURNAL OF ELECTRICAL POWER AND ENERGY SYSTEMS

TEPES, Vol. 4, Issue. 2, 57-62, 2024 **DOI:** 10.5152/tepes.2024.24005

RESEARCH ARTICLE

Overshoot Avoiding Control for Rigid Robots

Figen Özen

Department of Electrical and Electronics Engineering, Haliç University, İstanbul, Türkiye

Cite this article as: F. Özen, "Overshoot avoiding control for rigid robots," *Turk J Electr Power Energy Syst.*, 2024; 4(2), 57-62.

ABSTRACT

Control of robots to ensure safe operation has been a current research topic ever since the manipulators were utilized in industrial applications. Humans and robots work together collaboratively more often these days. Providing a safer operation has become an issue that requires more attention. In this paper, the application of a control algorithm that combines a proportional plus derivative controller with an overshoot avoidance trajectory tracking mechanism is introduced. The operation of the algorithm is illustrated on a two-link planar manipulator. The performance is shown both by means of graphs of end effector trajectories and mean absolute error metric. The results indicate that depending on the gain of the proportional plus derivative controller, desired trajectories can be followed very closely and the error becomes negligibly small.

Index Terms—Overshoot, proportional plus derivative control, robot control, rigid robot

I. INTRODUCTION

A robot is a highly nonlinear and complex mechanism. To control a robot, many different approaches can be used.

In [1], sliding mode and backstepping controllers are applied in a simulation study for two-link manipulators. It is concluded that a modified sliding mode controller has a more robust performance, whereas the backstepping algorithm responds faster. In [2], a proportional–integral–derivative controller (PID) controller is used to balance and stabilize a two-link manipulator, keeping it at the desired position. In [3], decoupling control for robot manipulators is introduced to secure a linearized and stable system. In [4], a model predictive controller is used in the case of a space robot. The obstacle avoidance problem is also dealt with. In [5], a collaborative arm is controlled with an added filter for the reference and in the presence of constraints, so as to work with a human safely. In [6], decoupling control is applied to a two-link manipulator, considering the elasticity of one joint.

In [7], a fire-fighting robot is controlled using a PID controller in case of deviation from the desired trajectory. In [8], a neural network is used to estimate the parameters of a robot, which is then controlled using a Lyapunov function.

In [9], a sliding mode controller is used to control a single-link flexible manipulator. In [10], impedance shaping controller is used to avoid force overshoots. In [11], a rigid manipulator is controlled using a fractional sliding mode controller. In [12], a PID controller is combined with an observer to control a mobile robot in the case of kinematic disturbances. In [13], position and orientation tracking algorithms are combined to create a cascaded controller for a mobile robot. In [14], a model reference controller is used to control a mobile platform under steering constraint. In [15], sliding mode controllers are used under uncertain parameters and external disturbance conditions. Also, in [16], a sliding mode controller is used for uncertain environmental conditions. In [17], plus derivative (PD) control is applied to a collaborative robot.

Trajectory tracking is one of the main concerns for robots in industrial applications. Avoidance of overshoot is an essential part of the control since it may cause collision with other robot(s), equipment, or human(s) in the work environment. Even though there are some studies for the solution of this problem, more work and new solutions are needed.

The paper is organized as follows: In Section II, the method is presented. In Section III, the experimental work is summarized, and the

Corresponding author: Figen Özen, figenozen@halic.edu.tr

Received: February 21, 2024
Revision Requested: March 4, 2024
Last Revision Received: March 6, 2024
Accepted: March 23, 2024
Publication Date: May 8, 2024

results are discussed. In Section IV, conclusions are drawn, limitations are listed, and future directions are shown.

II. METHODS

The Overshoot Avoiding Control Algorithm (OACA) consists of two switching controllers. They are basic PD and Constant Error Acceleration (CEA) algorithms.

The PD control module is very simple and can be easily employed. It has only a few parameters to choose from, and this fact makes it easy to use. Fig. 1 shows the block diagram where a series PD controller is employed to control the plant ($G_p(s)$). The constants K_P and K_D are the design parameters of the controller. The effect of this type of controller on a second-order system depends on the parameter values. If overshoot is not tolerated, these constants should be chosen accordingly. Adding a PD controller to the forward path of a closed-loop system as in Fig. 1 changes the damping ratio and the maximum overshoot values [18].

If the PD parameters are chosen such that there is no overshoot, this may cause a very slow response. For realistic manipulator operation, speed is also an important criterion. Therefore, a trade-off between speed and safety is needed.

The controller has PD characteristics as long as it is guaranteed that there will be no overshoot if the control input calculated this way is applied. On the other hand, if the initial conditions indicate that there will be an overshoot, i.e., (1) will hold:

$$\frac{\dot{e}(t)}{e(t)} < \frac{\alpha}{2} \tag{1}$$

where α is the PD controller gain selected for critical damping, $e(t)$ is the function that shows the difference between the desired and realized outputs, and $e(t)$ is the time derivative of the error function [19, 20]. The success of the controller depends on the selection of the gain α. If it is large, the error function tends to zero faster. On the other hand, a very large gain may lead to exceeding the torque (force) limits.

As an initial step, the gain α is selected, and the required PD controller acceleration to track the target trajectory is calculated. If the resulting acceleration is not attainable, the maximum available acceleration is set as the one to be applied. Another test is needed to determine if the system is in the overshoot region. If the answer is no, a successive test is applied to see if the calculated PD acceleration causes the system to fall into the overshoot region in the next iteration. If both answers are negative, the calculated PD acceleration is applied safely. On the other hand, if the calculated PD acceleration causes the system to fall into the overshoot region in the next iteration, it is called $\ddot{e}_{in}$, and it is divided successively by two until a point is reached where the application of the new acceleration does not cause overshoot. Upon finding this specific acceleration, it will be assigned the name $\ddot{e}_{out}$.

Main Points

- The article provides a simple yet effective overshoot-avoiding algorithm for rigid robots.
- The algorithm ensures safer operation in a work environment.
- The design of the algorithm is independent of the robot. It is general and can be applied to all rigid robots.

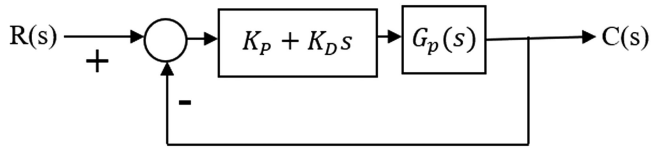

Fig. 1. Feedforward PD controller in the closed-loop system. PD, plus derivative.

Given $\ddot{e}_{in}$ and $\ddot{e}_{out}$, they are averaged to get the acceleration to be applied. After this step, the usual checks on possibility of overshoot are carried out to calculate the new acceleration if the average acceleration is likely to cause an overshoot. Fig. 2 shows the flowchart of the algorithm presented in this paper.

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. Manipulator Equations

To illustrate the operation of the algorithm, a robotic manipulator model is chosen. It is easy to find a manipulator model; on the other hand, the most difficult part in selection is finding a model with the numerical parameters. For simulation purposes of this paper, Sciavicco and Siciliano's two-link planar arm model is employed [21]. Fig. 3 shows the manipulator under consideration.

The equations of motion for this manipulator are given in the reference as:

$$\begin{aligned}&\left(I_{l_1} + m_{l_1} l_1^2 + k_{r_1}^2 I_{m_1} + I_{l_2} + m_{l_2}\left(a_1^2 + l_2^2 + 2a_1 l_2 c_2\right) + I_{m_2} + m_{m_2} a_1^2\right)\ddot{\theta}_1 \\ &+\left(I_{l_2} + m_{l_2}\left(l_2^2 + a_1 l_2 c_2\right) + k_{r_2} I_{m_2}\right)\ddot{\theta}_2 - 2m_{l_2} a_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 \\ &- m_{l_2} a_1 l_2 s_2 \dot{\theta}_2^2 + \left(m_{l_1} l_1 + m_{m_2} a_1 + m_{l_2} a_1\right) g c_1 + m_{l_2} l_2 g c_{12} = \tau_1\end{aligned} \tag{2}$$

$$\begin{aligned}&\left(I_{l_2} + m_{l_2}\left(l_2^2 + a_1 l_2 c_2\right) + k_{r_2} I_{m_2}\right)\ddot{\theta}_1 + \left(I_{l_2} + m_{l_2} l_2^2 + k_{r_2}^2 I_{m_2}\right)\ddot{\theta}_2 \\ &+ m_{l_2} a_1 l_2 s_2 \dot{\theta}_1^2 + m_{l_2} l_2 g c_{12} = \tau_2\end{aligned} \tag{3}$$

where:

τ_1 and τ_2 are torques applied at joints 1 and 2, respectively.

g is the acceleration of gravity.

a_1 and a_2 are the lengths of links 1 and 2.

l_1 and l_2 are the lengths of links 1 and 2 (center of mass).

m_{l_1} and m_{l_2} are the masses of links 1 and 2 (center of mass).

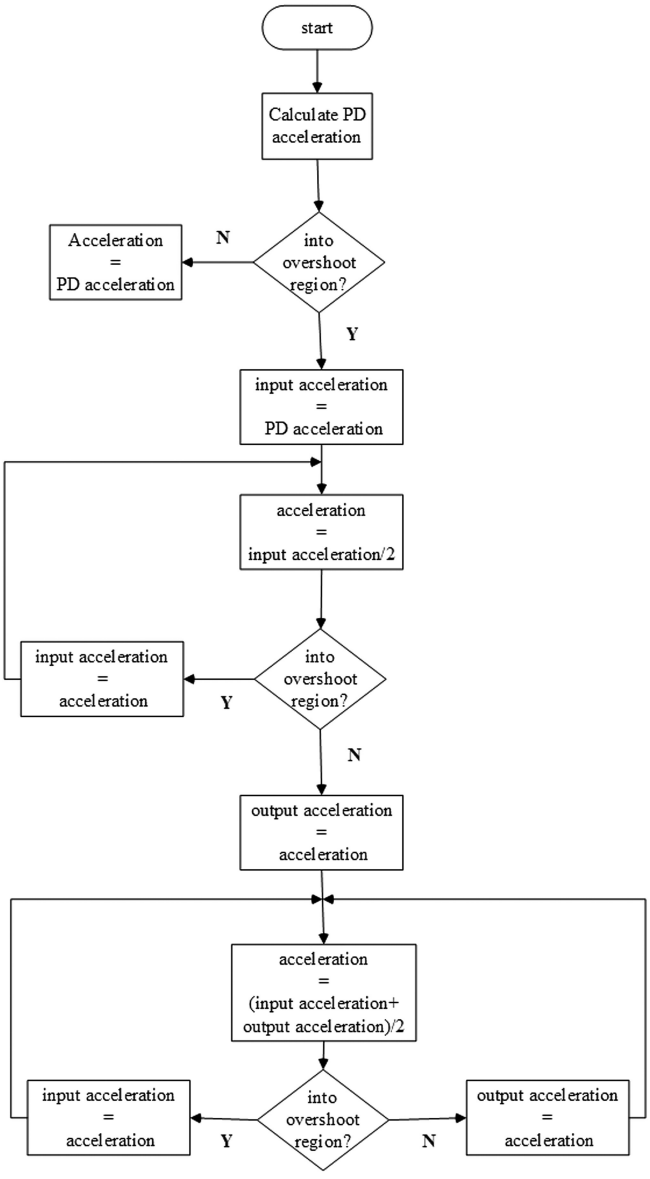

Fig. 2. Overshoot avoiding control algorithm. PD, plus derivative.

I_{l_1} and I_{l_2} are the moments of inertia of links 1 and 2 (center of mass).

m_{m_1} and m_{m_2} are the masses of rotors 1 and 2.

m_{m_1} and k_{r_2} are the gear reduction ratios of rotors 1 and 2.

I_{m_1} and I_{m_2} are the inertia tensors of the rotors relative to their centers of masses.

θ_1 and θ_2 are the rotation angles of link 1 with respect to the reference and link 2 with respect to link 1, respectively.

The parameters of the planar manipulator model in [21] are repeated here in Table I for convenience.

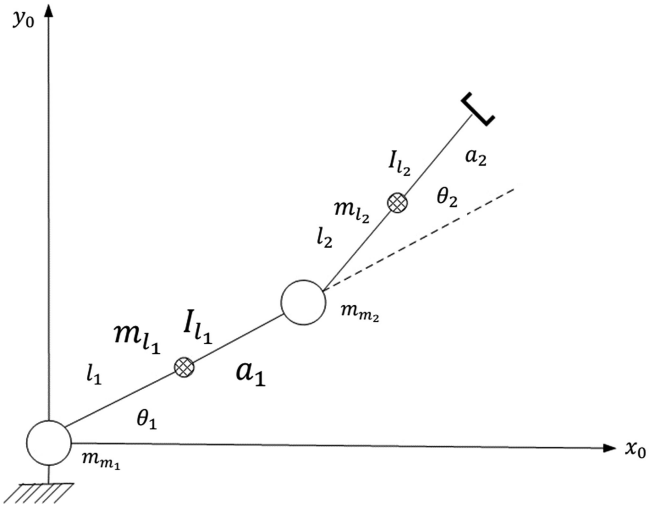

Fig. 3. Two-link planar manipulator.

B. Simulations

The performance of the control algorithm presented here is illustrated using the given parameters of the manipulator model. The results of six experiments are shown below. These experiments are designed on the basis of varying gain, α. The values of α chosen for the experiments are displayed in Table II.

For the following experimental results, the initial errors are held constant:

$$e_0 = 1, \dot{e}_0 = 0.1 \tag{4}$$

The desired trajectory in the x-y plane is given in Fig. 4. The performances in tracking of x and y coordinates are shown in Figs. 5 and 6, respectively. It is common to both performances that the α values of 100 and 400 create very similar results to the desired response; the errors introduced are negligible.

In order to accompany the graphs on tracking, the mean absolute errors in the x and y directions are calculated and listed in Table III. The formula used in the calculations is given by (5):

$$MAE = \frac{\sum_{i=1}^{N} \left| desired\,value_i - realized\,value_i \right|}{N} \tag{5}$$

where N is the number of instants, desired value is a design parameter for the experiment, and the realized value is determined by the control input applied. The formula is exploited both for x and y trajectories. As can be seen from Table III, increasing the α value causes a smaller mean absolute error value. On the other hand, increasing the hyperparameter α beyond a certain value is useless as can be concluded from the same table.

The required torque profiles are shown in Fig. 7 and Fig. 8 for joints 1 and 2, respectively. Since the limits of the applicable torque values are not provided in the example used here, the torque values are assumed to be available and no limits have been applied. In

TABLE I.
PARAMETERS OF TWO-LINK PLANAR MANIPULATOR

Parameters	Value	Unit
a_1, a_2	1	m
l_1, l_2	0.5	m
m_{l_1}, m_{l_2}	50	kg
I_{l_1}, I_{l_2}	10	kg·m^2
k_{r_1}, k_{r_2}	100	
m_{m_1}, m_{m_2}	5	kg
I_{m_1}, I_{m_2}	0.01	kg·m^2

TABLE II.
EXPERIMENT NUMBER AND ITS RELATION WITH THE GAIN

Experiment number	α_i
1	1
2	4
3	10
4	40
5	100
6	400

real applications, there will be limits on maximum available torque values. In this case, the tracking performance is expected to deteriorate. Keeping in mind that very large torque values are not applicable, they should be avoided. This is possible when the very small α values are avoided. This strategy is meaningful also due to the fact that for very small α values the tracking is poor as is obvious from Figs. 5 and 6 and Table III.

To illustrate the advantage of applying OACA instead of a PD controller, Fig. 9 is helpful. Fig. 9 depicts the mean absolute error (MAE)

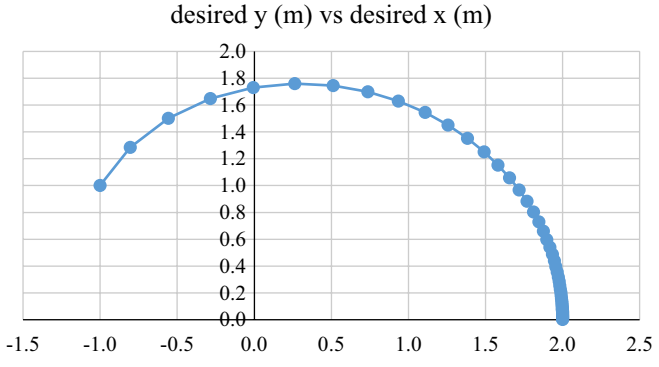

Fig. 4. The desired tip position trajectory.

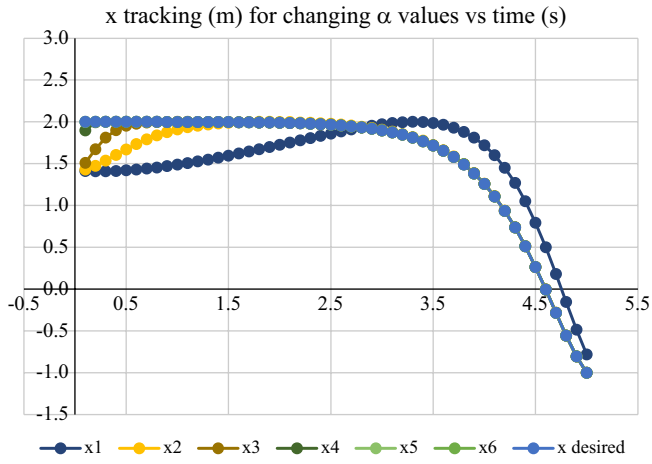

Fig. 5. Trajectory tracking in the x direction for various α values.

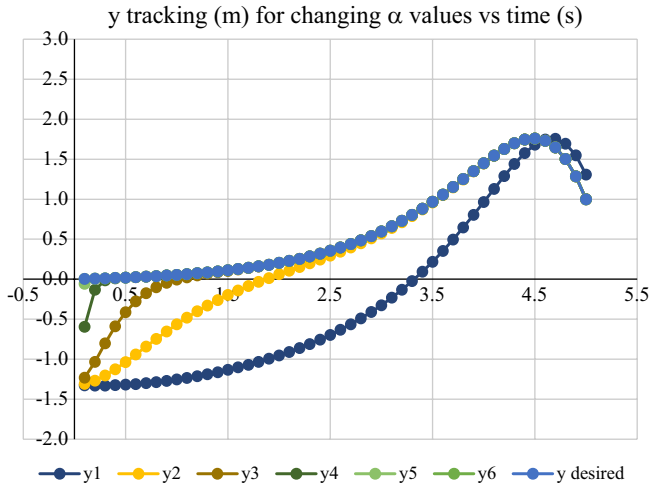

Fig. 6. Trajectory tracking in the y direction for various α values.

values in x and y coordinates for various values of α. As shown in the figure, the MAE values in both coordinates have been decreased by OACA, the proposed controller.

TABLE III.
MEAN ABSOLUTE ERROR VALUES FOR X AND Y COORDINATES IN THE EXPERIMENTS

Experiment number	MAE in x	MAE in y
1	0.359247	0.905532
2	0.070233	0.275848
3	0.023898	0.101000
4	0.002165	0.015413
5	0.000058	0.001293
6	0.000040	0.000065

MAE, mean absolute error.

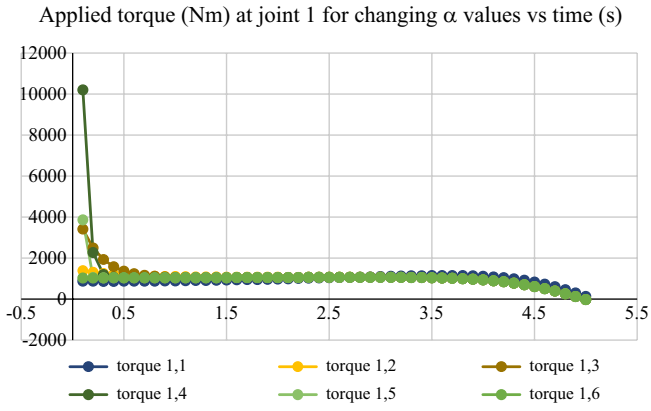

Fig. 7. Torque profiles at joint 1 for various α values.

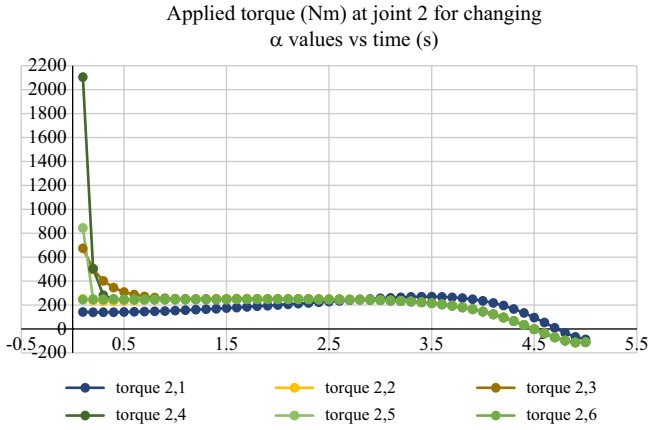

Fig. 8. Torque profiles at joint 2 for various α values.

In order to test the OACA controller using a discontinuous trajectory, the one shown in Fig. 10 is used. Due to the overshoot-avoiding nature of the controller, the trajectory is followed smoothly. The initial error is also quickly compensated. The result shown in Fig. 10 is for $\alpha = 40$.

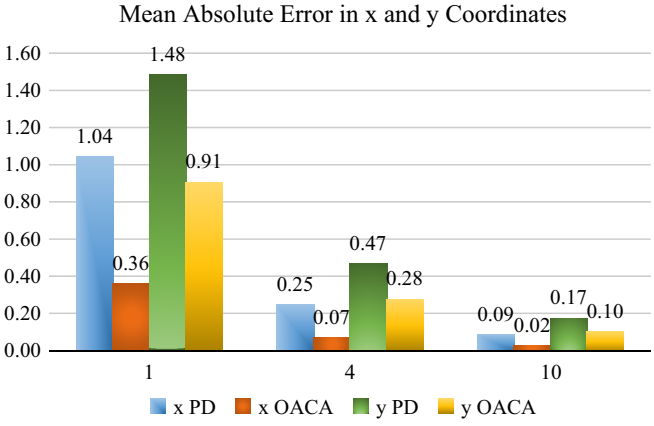

Fig. 9. Comparison of mean absolute error values in x and y coordinates for PD and OACA. OACA, overshoot avoiding control algorithm; PD, plus derivative.

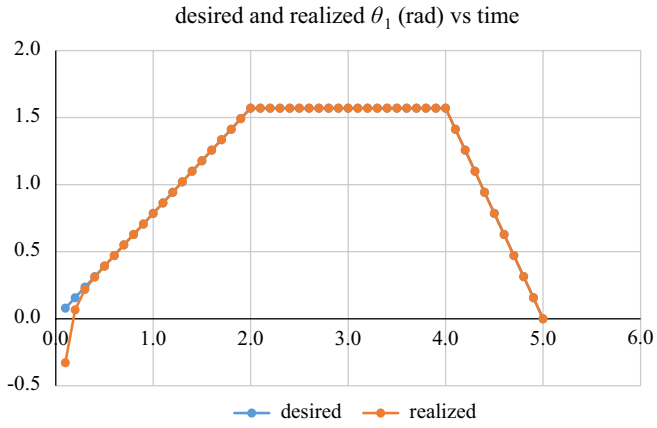

Fig. 10. Tracking a trajectory with discontinuities using OACA. OACA, overshoot avoiding control algorithm.

The performance of the OACA controller can be compared to the industry-standard PID (proportional, integral, and derivative) controller [18]. Fig. 11 shows the tracking of the first angle (θ_1) in Fig. 3. It can be observed that OACA tracking is better than PID. To make a fair comparison, the gain that provides critical damping and the same initial conditions are used.

The simulations in this study are conducted using a 12th Gen Intel Core i7-12650H Processor at 2.3 GHz with 16 GB of RAM and NVIDIA GeForce RTX 3060. The programming language used is Python, and the programming environment is Jupyter Notebook.

IV. CONCLUSION

A control algorithm that combines a PD controller and a Constant Error Acceleration, namely OACA, is presented in this paper. The function of OACA is illustrated using a two-link planar manipulator. Simulations show that the adoption of this algorithm results in

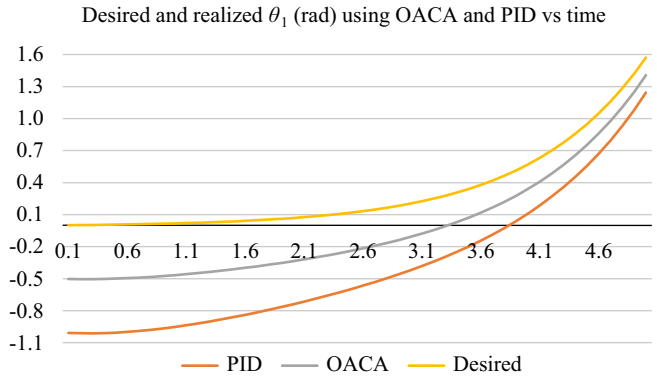

Fig. 11. Comparison of tracking of OACA and PID controllers under similar conditions. OACA, overshoot avoiding control algorithm; PID, proportional integral derivative.

successful trajectory tracking, and the reduction of initial errors is fast. Therefore, it can be applied safely for rigid robots.

The limitations of this work are the assumptions of operation without a payload and the certainties of the given parameters of the manipulator. In future work, these assumptions will be removed, and the OACA will be tested under the conditions of uncertainty and operation with the payload.

Peer-review: Externally peer-reviewed.

Declaration of Interests: The author has no conflicts of interest to declare.

Funding: This study received no funding.

REFERENCES

1. F. Massaoudi, D. Elleuch, and T. Damak, "Robust control for a two dof robot manipulator," *J. Electr. Comput. Eng.*, vol. 2019, pp. 1–11, 2019. [CrossRef]
2. B. Mahboub, and D. Stephen, "A two-link robot manipulator: Simulation and control design," *Int. J. Robot. Eng.*, vol. 5, no. 2, pp. 1-17, 2020. [CrossRef]
3. P. N. Paraskevopoulos, and A. S. Tsirikos, and X. Koutsoukos, "*Nonlinear Decoupling Control For a Robot Manipulator*, 1998.
4. M. Wang, J. Luo, and U. Walter, "A non-linear model predictive controller with obstacle avoidance for a space robot," *Adv. Space Res.*, vol. 57, no. 8, pp. 1737–1746, 2016. [CrossRef]
5. K. Merckaert, B. Convens, C. JuWu, A. Roncone, M.M. Nicotra, and B. Vanderborght, "Real-time motion control of robotic manipulators for safe human–robot coexistence," *Robot Comput Integr Manuf*, vol. 73, pp. 1-16, 2022. [CrossRef]
6. A. De Luca, C. Manes, and F. Nicolo, "A task space decoupling approach to hybrid control of manipulators," *IFAC Proceedings Volumes*, vol. 21, no. 16, 157–162, 1988. [CrossRef]
7. S. Zhang, R. Wang, Y. Tian, J. Yao, and Y. Zhao, "Motion analysis of the fire-fighting robot and trajectory correction strategy," *Simul. Modell. Pract. Theor.*, vol. 125, pp. 1-17, 2023. [CrossRef]
8. M. Adinehvand, E. Asadi, C. Y. Lai, H. Khayyam, and R. Hoseinnezhad, "Design and adaptive control of a kinematically redundant robot with enhanced trajectory tracking for climbing in tight spaces," *Mech. Mach. Theor.*, vol. 177, no. 5, pp. 1-16, 2022. [CrossRef]
9. J. F. Peza-Solis, G. Silva-Navarro, O. A. Garcia-Perez, and L. G. Trujillo-Franco, "Trajectory tracking of a single flexible-link robot using a modal cascaded-type control," *Appl. Math. Modell.*, vol. 104, pp. 531–547, 2022. [CrossRef]
10. L. Roveda, N. Pedrocchi, M. Beschi, and L. Molinati Tosatti, "High-accuracy robotized industrial assembly task control schema with force overshoots avoidance," *Control Eng. Pract.*, vol. 71, pp. 142–153, 2018. [CrossRef]
11. S. Chávez-Vázquez, J. E. Lavín-Delgado, J. F. Gómez-Aguilar, J. R. Razo-Hernández, S. Etemad, and S. Rezapour, "Trajectory tracking of Stanford robot manipulator by fractional-order sliding mode control," *Appl. Math. Modell.*, vol. 120, pp. 436–462, 2023. [CrossRef]
12. R. Miranda-Colorado, "Observer-based proportional integral derivative control for trajectory tracking of wheeled mobile robots with kinematic disturbances," *Appl. Math. Comput.*, vol. 432, no. 5, pp. 1-14, 2022. [CrossRef]
13. Y. Zhou, H. Ríos, M. Mera, A. Polyakov, G. Zheng, and A. Dzul, "Trajectory tracking in unicycle mobile robots: A homogeneity–based control approach," *IFAC PapersOnLine*, vol. 56, no. 2, pp. 54–59, 2023. [CrossRef]
14. T. Ding, Y. Zhang, G. Ma, Z. Cao, X. Zhao, and B. Tao, "Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint," *Mechatronics*, vol. 84, pp. 1-14, 2022. [CrossRef]
15. D. Shi, J. Zhang, Z. Sun, G. Shen, and Y. Xia, "Composite trajectory tracking control for robot manipulator with active disturbance rejection," *Control Eng. Pract.*, vol. 106, pp. 1-8, 2021. [CrossRef]
16. L. Li, W. Cao, H. Yang, and Q. Geng, "Trajectory tracking control for a wheel mobile robot on rough and uneven ground," *Mechatronics*, vol. 83, pp. 1-9, 2022. [CrossRef]
17. K. Merckaert, B. Convens, M. M. Nicotra, and B. Vanderborght, "Real-time constraint-based planning and control of robotic manipulators for safe human–robot collaboration," *Robot. Comput. Integr. Manuf.*, vol. 87, pp. 1-16, 2024. [CrossRef]
18. B. C. Kuo, *Automatic Control Systems*. Prentice-Hall International Inc., 1991.
19. Y. Denizhan, *A Variable Structure Control Algorithm for Robotic Systems and a Learning Scheme for Load Adaptation*. İstanbul, Turkey: Ph.D., Boğaziçi University, 1988.
20. F. Özen, "Variable structure control of a manipulator with four degrees of freedom," *M.Sc.* İstanbul, Turkey: Boğaziçi University, 1989.
21. L. Sciavicco, and B. Siciliano, *Modelling and Control of Robot Manipulators*. London: Springer, 2002.