OPTIMAL IMPEDANCE CONTROL OF A 2R PLANAR ROBOT MANIPULATOR

Abdullah ERDEMİR [1]

[1] *MPG Machinery Production Group Inc. Co., Konya, Türkiye.*

[1] *ORCID ID: https://orcid.org/0000-0002-7267-3111*

Mete KALYONCU [2]

[2] *Prof. Dr., Konya Technical University, Department of Mechanical Engineering, Konya, Türkiye.*

[2] *ORCID ID: https://orcid.org/0000-0002-2214-7631*

ABSTRACT

In this study, optimum impedance control of a 2R planar robot manipulator was performed. Industrial applications of robot manipulators are generally related to manipulation tasks such as painting that only require arm position control. However, there are other robotic tasks such as pushing, polishing, cleaning, and grinding that require interaction between the robot manipulator and a contact surface or environment. This fact makes it imperative to control the interaction between the robot and the environment. The impedance controller aims to control the dynamic relationship between the robot and the environment. The force applied by the robot to the environment depends on the position of the robot manipulator endpoint and the corresponding impedance. The impedance controller forces the robot to follow the desired reference or target impedance. Force task for force/position hybrid controller in literature; It is divided into two subspaces, the force control subspace, and the position control subspace. Then, two independent controllers are designed for each subspace. Conversely, the impedance controller proposed in this study does not attempt to explicitly control the force. Instead, it tries to control the relationship between the force and the position of the end effector in contact with the environment. It is also possible to plan a virtual trajectory such that a desired force profile is obtained when the environment has a rigid structure with known properties. The Bees Algorithm was used to optimize the proposed impedance controller and a numerical application was made to evaluate its performance. As a result of the optimization, the objective function was reduced by 57%. The obtained results are presented numerically and graphically. Thanks to the proposed impedance controller, the robot manipulator endpoint precisely follows both the desired force profile and the desired position.

Keywords: 2R planar robot manipulator, optimum impedance control, controller design, The Bees Algorithm.

INTRODUCTION

Industrial applications of robot manipulators are generally related to manipulation tasks such as painting that only require arm position control [1-3]. The painting process can be done by spraying from a distance without the need for contact. Therefore, there is usually no need for force control. However, there are other robotic tasks such as pushing, polishing, cleaning, and grinding that require interaction between the robot manipulator and a contact surface or environment [4, 5]. Force based control techniques are well suited to this type of operation. One of these control techniques is impedance control. The impedance controller aims to control the dynamic relationship between the robot and the environment [6-8]. The force applied by the robot to the environment depends on the position of the robot manipulator endpoint and the corresponding impedance. The impedance controller forces the robot to follow the desired reference or target impedance.

Force task for force/position hybrid controller in literature; It is divided into two subspaces, the force control subspace, and the position control subspace [9, 10]. Then, two independent controllers are designed for each subspace. Conversely, the impedance controller proposed in this study does not attempt to explicitly control the force. Instead, it tries to control the relationship between the force and

the position of the end effector in contact with the environment. It is also possible to plan a virtual trajectory such that a desired force profile is obtained when the environment has a rigid structure with known properties.

The parameters of the impedance controller need to be optimized. Optimization algorithms are divided into two as global optimization algorithms and local search algorithms [11].

Global optimization algorithms such as Genetic Algorithm [12, 13], Particle Swarm Optimization [14] and The Bees Algorithm [15, 16] focus on searching across the entire space. Local search algorithms such as Hooke-Jeeves [17] and Newton Raphson [18] focus on searching in local areas. The Bees Algorithm was used to optimize the proposed impedance controller and a numerical application was made to evaluate its performance [19]. The obtained results are presented numerically and graphically. Thanks to the proposed impedance controller, the robot manipulator endpoint precisely follows both the desired force profile and the desired position.

MATHEMATICAL MODEL

The robot has two degrees of freedom and consists of revolute joints. The system consisting of 2 linkages with masses m_1 and m_2 is shown in Figure 1.

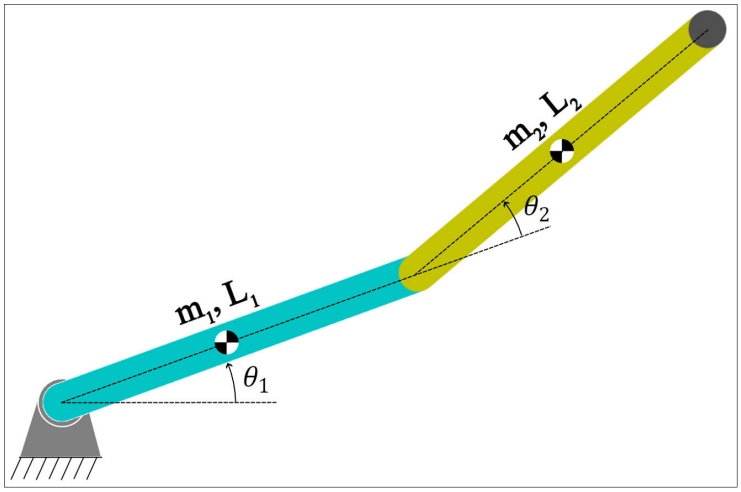

Figure 1. 2 dof robotic system

In order to derive the mathematical model of the system, the position of the centers of mass with respect to the joint angles must first be obtained. The center of mass position of the linkages with masses m_1 and m_2 according to the joint angles is shown in Equation (1).

$$\begin{aligned} \bar{x}_1 &= \frac{L_1}{2}\cos\theta_1 \\ \bar{y}_1 &= \frac{L_1}{2}\cos\theta_2 \\ \bar{x}_2 &= L_1\cos\theta_1 + \frac{L_2}{2}\cos(\theta_1+\theta_2) \\ \bar{y}_2 &= L_1\sin\theta_1 + \frac{L_2}{2}\sin(\theta_1+\theta_2) \end{aligned} \tag{1}$$

The mass moments of inertia of the rotation of the linkages around the center of mass are taken as $I = \frac{1}{12}mL^2$. The kinetic energy T of the system is given in Equation (2). The potential energy V is given in Equation (3).

$$T = \frac{1}{2}m_1(\dot{\bar{x}}_1^2 + \dot{\bar{y}}_1^2) + \frac{1}{2}I_1\dot{\theta}_1^2 + \frac{1}{2}m_2(\dot{\bar{x}}_2^2 + \dot{\bar{y}}_2^2) + \frac{1}{2}I_2\left(\dot{\theta}_1+\dot{\theta}_2\right)^2 \tag{2}$$

$$V = m_1 g\bar{y}_1 + m_2 g\bar{y}_2 \tag{3}$$

Euler – Lagrangian equation of motion is used to obtain the equation of motion of the system. The Lagrangian expression for this is given in Equation (4).

$$L = T - V \tag{4}$$

The Euler – Lagrangian equation of motion applied to obtain the torque in the joints is given in Equation (5).

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} \tag{5}$$

The expansion of torques τ_1 and τ_2 is given in Equation (6).

$$\begin{aligned}\tau_1 = &\left(\frac{1}{3}m_1L_1^2 + m_2\left(L_1^2 + \frac{1}{3}L_2^2 + L_1L_2\cos(\theta_2)\right)\right)\ddot{\theta}_1 + m_2L_2\left(L_1\cos(\theta_2) + \frac{2}{3}L_2\right)\ddot{\theta}_2 \\ &- m_2L_1L_2\sin(\theta_2)\left(\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}\dot{\theta}_2^2\right) + \frac{1}{2}(m_1 + 2m_2)gL_1\cos(\theta_1) \\ &+ \frac{1}{2}m_2\, gL_2\cos(\theta_1 + \theta_2) \\ \tau_2 = &\frac{1}{2}m_2L_2\left(L_1\cos(\theta_2) + \frac{2}{3}L_2\right)\ddot{\theta}_1 + \frac{1}{2}m_2L_1L_2\sin(\theta_2)\,\dot{\theta}_1^2 + \frac{1}{3}m_2L_2^2\dot{\theta}_2^2 \\ &+ \frac{1}{2}m_2gL_2\cos(\theta_1 + \theta_2)\end{aligned} \tag{6}$$

IMPLEMENTATION OF IMPEDANCE CONTROL

The impedance control schematic is shown in Figure 2. Impedance control requires forward kinematics and inverse dynamics equations of the point where the robot is desired to interact, using the joint angles. The distance between the position obtained with forward kinematics and the desired position is converted into interaction force by multiplying the spring and by multiplying the velocity with the damping coefficients. It is then multiplied by the transpose of the Jacobian matrix and converted into torques at the joints. These torques are added to the torques found by inverse dynamics to obtain a final torque. This torque is sent to the robot and control is provided.

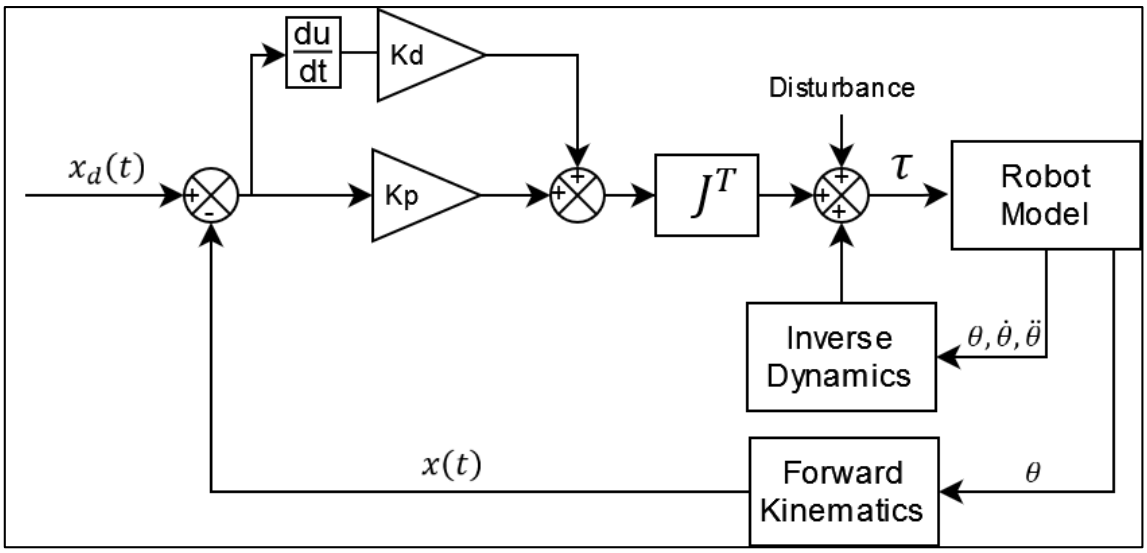

Figure 2. Impedance control

The PID-based interaction force is given in Equation (7).

$$F_{int} = k\begin{bmatrix} x_{2d} - x_2 \\ y_{2d} - y_2 \end{bmatrix} + b\begin{bmatrix} \dot{x}_{2d} - \dot{x}_2 \\ \dot{y}_{2d} - \dot{y}_2 \end{bmatrix} + i\int \begin{bmatrix} x_{2d} - x_2 \\ y_{2d} - y_2 \end{bmatrix} dt \quad (7)$$

The spring coefficient of the impedance controller is defined as *k*, the damping coefficient *b*, and the integral gain *i*. x_{2d} and y_{2d} are the targeted location of the robot's endpoint. x_2 and y_2 are the actual position of the robot's endpoint. The position of the end point of the robot is given in Equation (8).

$$\begin{aligned} x_2 &= L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ y_2 &= L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{aligned} \quad (8)$$

In order for the interaction force in Equation (7) to be converted into torques, it is necessary to multiply it by the Jacobian matrix of the robot endpoint. The Jacobian matrix of the system is given in Equation (9).

$$J = \begin{bmatrix} \frac{dx_2}{d\theta_1} & \frac{dx_2}{d\theta_2} \\ \frac{dy_2}{d\theta_1} & \frac{dy_2}{d\theta_2} \end{bmatrix} \quad (9)$$

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

The impedance torques obtained by multiplying the interaction forces with the transpose of the Jacobian matrix are given in Equation (10).

$$\begin{aligned} \tau_e &= J^T F_{int} \\ \tau_e &= \begin{bmatrix} -F_x(L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2)) + F_y(L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2)) \\ -F_x L_2 \sin(\theta_1 + \theta_2) + F_y L_2 \cos(\theta_1 + \theta_2) \end{bmatrix} \end{aligned} \quad (10)$$

In Equation (10), the x component of the F_{int} force is F_x and the y component is F_y. The final torques are the sum of dynamic torques in Equation (6) and the impedance torques in Equation (10).

NUMERICAL APPLICATION

$L_1 = L_2 = 1$ m and $m_1 = m_2 = 5$ kg. Figure 3 shows the path followed by the scenario. The BC plane is 1 m away from the first joint of the robot and is at a 65° inclination with the ground. The endpoint of the robot first moves from point A to point B. It then proceeds from point B to point C by applying a force of 50 N to the BC plane.

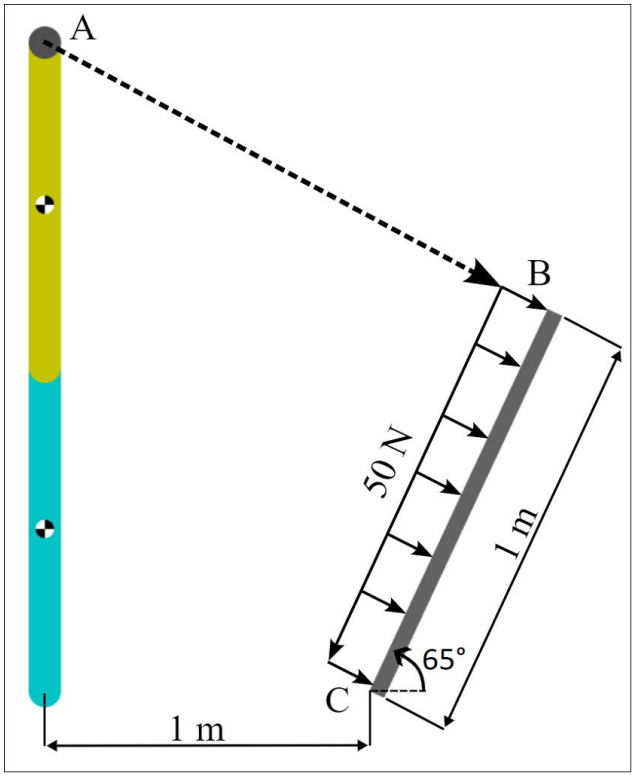

Figure 3. Numeric application view

In order for the robot to apply 50 N perpendicular to the surface while its tip point moves in the BC plane, the 50 N force at a 90° - 65° = 25° inclination from the robot's tip point to the ground is added to the interaction force seen in equation (11).

$$F_{int,modified} = F_{int} + 50\begin{bmatrix}\cos(25°)\\ \sin(25°)\end{bmatrix} \tag{11}$$

The objective function used to optimize the k, b and i coefficients in Equation (7) is defined in Equation (12).

$$f_{obj} = \int_{t_1}^{t_2} (x_{2d} - x_2)^2 + (y_{2d} - y_2)^2 + (F_n - 50N)^2 \tag{12}$$

F_n is the reaction force of the BC plane against the robot endpoint. t_1 is the first contact time of the robot's endpoint to point B. t_2 is the time when the robot endpoint reaches point C.

The objective function is obtained by squaring the distance error and the difference of the surface reaction force from 50 N. By minimizing this objective function, both the distance error will be reduced, and the surface reaction force will be approached to 50 N.

THE BEES ALGORITHM

The algorithm starts with n scout bees being placed randomly in the search space. The fitness of the population is calculated; this means the error of all the scout bees are calculated. The array of the scout bees is reordered from minimum error to maximum error. The best m sites are selected to be search for neighborhood search. The next bees will search those sites within the radius of patch size which is ngh. But more scout bees will be sent to elite sites which are shown as number e. The remaining scout bees are less than the number of elite bees. Each site is reordered from minimum error to maximum error. In addition, the fittest bee is selected for that site. The remaining (n-m) bees are replaced with the new randomly created bees. The fitness of the new population is recalculated, and the loop continues until

the stop condition occurs. The pseudo code for The Bees Algorithm in its simplest form is:

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
4. Select sites for neighborhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitness.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitness.
8. End While

The Bees Algorithm is shown in Figure 4.

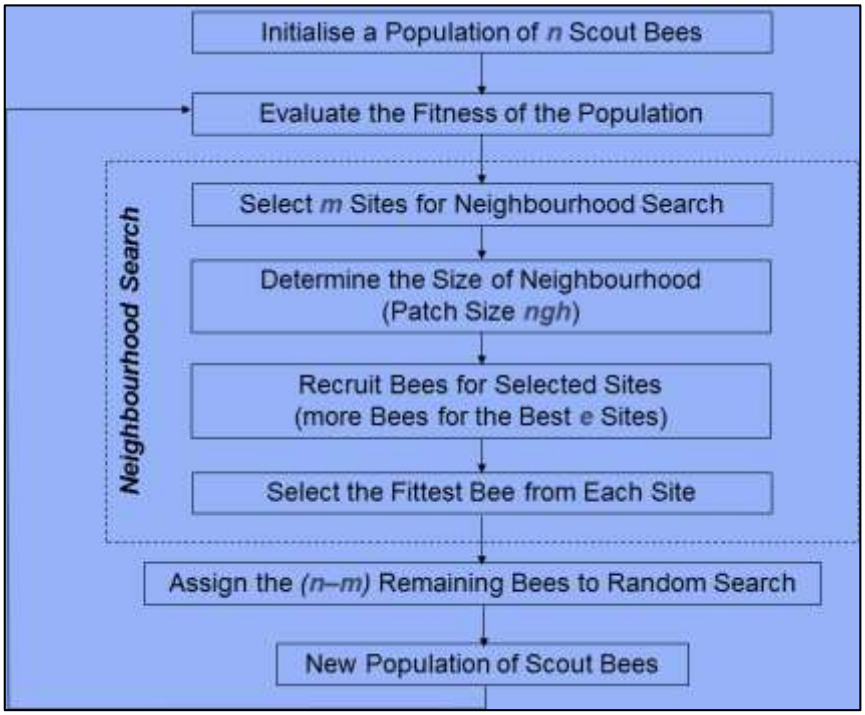

Figure 4. The Bees Algorithm

The parameters of The Bees Algorithm are given in Table 1.

Table 1. The Bees Algorithm parameters

n	m	e	nep	nsp	ngh	p_{max}	p_{min}
10	7	4	5	3	0.1	[30,5,5]	[0,0,0]

RESULTS AND DISCUSSION

The result of the optimization made with The Bees Algorithm is given in Table 2. The optimization is completed with 18243 iterations. As a result of the optimization made from a randomly generated starting point, the objective function decreased by 57% compared to the initial condition.

Table 2. The Bees Algorithm parameters

	Initial Conditions	The Bees Algorithm Result	Change
k	25	25.10477514	+0.41%
b	3	2.416703157	-19.44%
i	0	0.541278214	+0.54
f_{obj}	11.09625654	4.771782226	-57%

The convergence graph during the optimization of the objective function is shown in Figure 5.

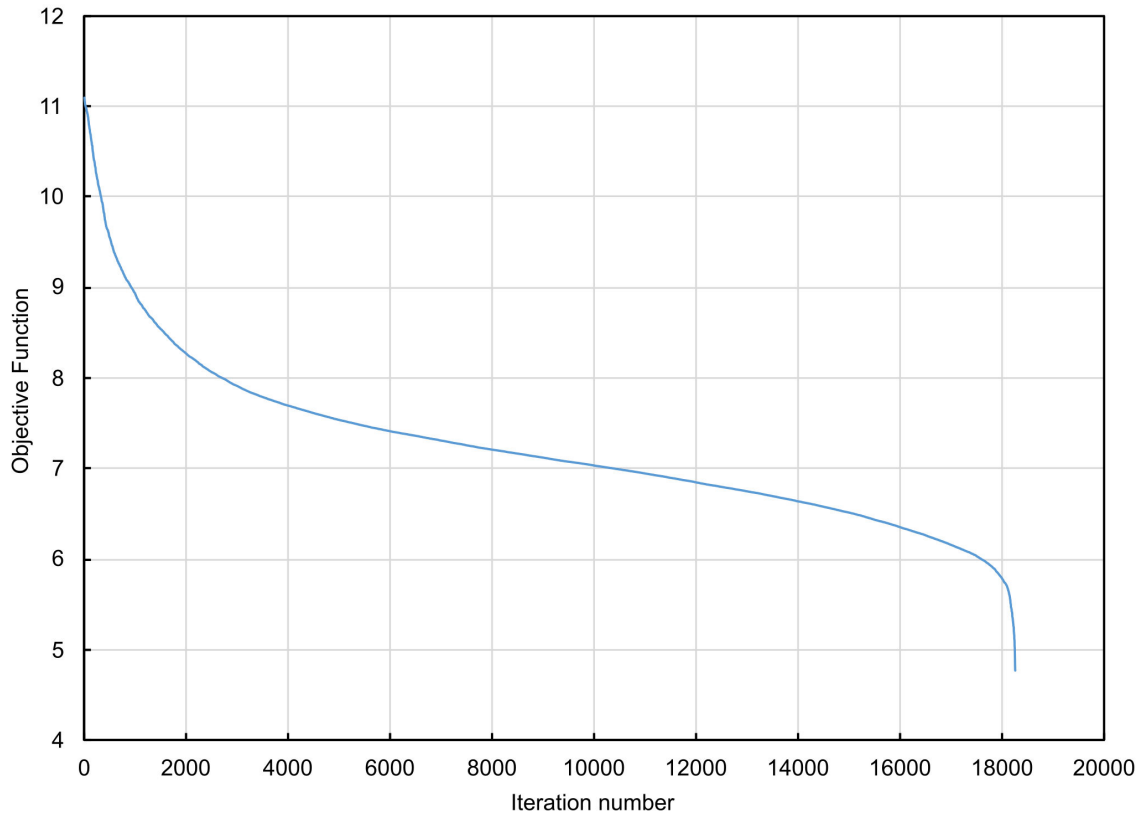

Figure 5. Convergence graph of the objective function.

The positions of the robot during the scenario are shown in Figure 6.

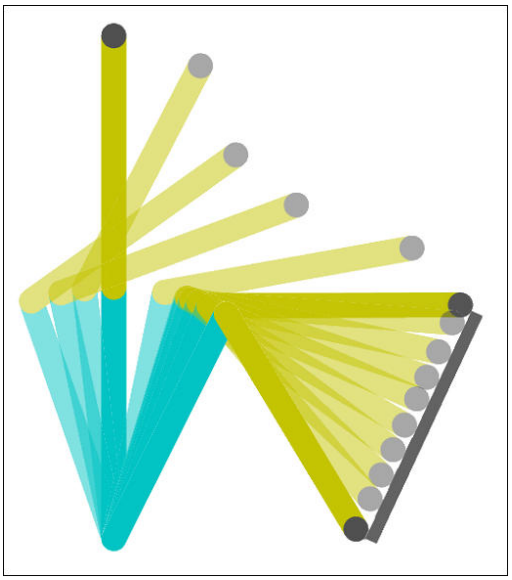

Figure 6. Movement of the robot in space and on the plane

The robot touches the BC plane at the 5th second. According to the scenario 50 N force must be applied on the surface and robot's endpoint must move from point B to C. From 5th to 7th second, robot's end point pushes on the BC plane until the reaction force is 50 N. From point B to point C, robot's endpoint moves while applying 50 N on the plane. The force change is shown in in Figure 7.

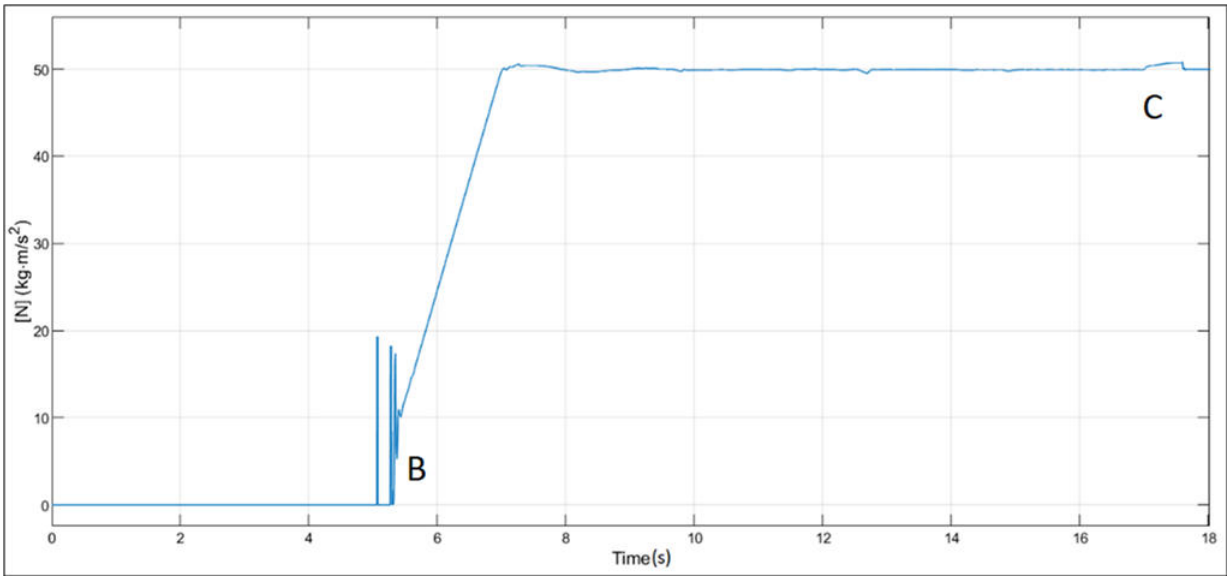

Figure 7. The reaction force of the plane during the movement of the robot

The reaction force of the BC plane to the endpoint of the robot ranges from 49.5 N to 50.8 N. The desired force of 50 N is applied within -1% and +1.6%.

In Figure 8, the desired positions are x_d and y_d, and the actual positions are seen as x_2 and y_2. As seen in the figure, actual positions are very close to desired locations.

The interaction distance at the endpoint of the robot is given in Figure 9.

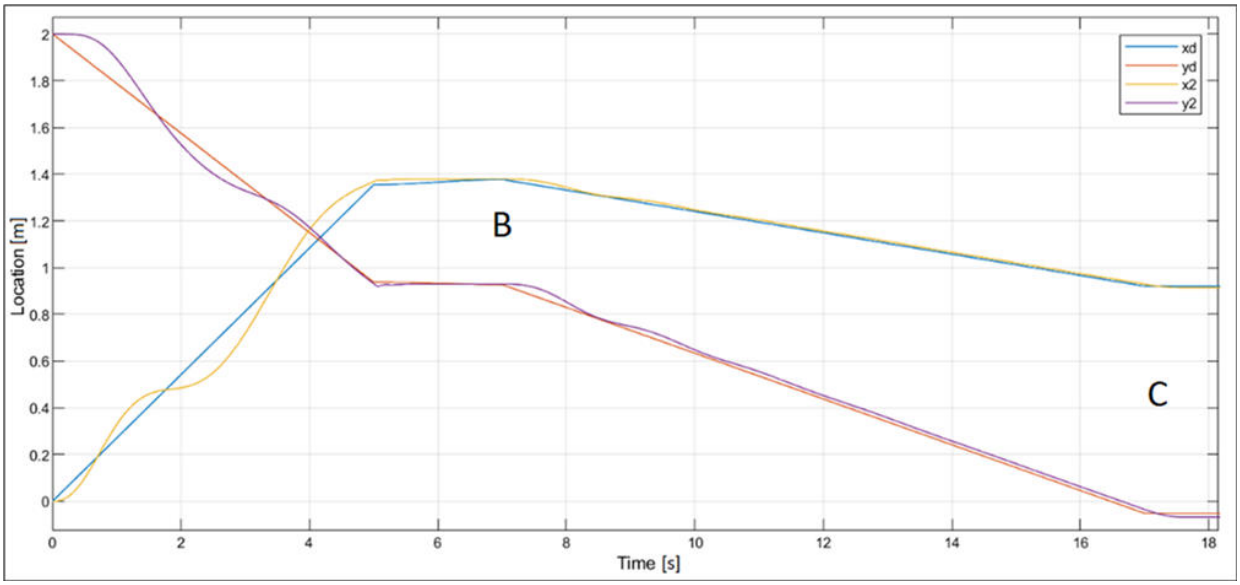

Figure 8. Desired and actual locations of the robot's endpoint.

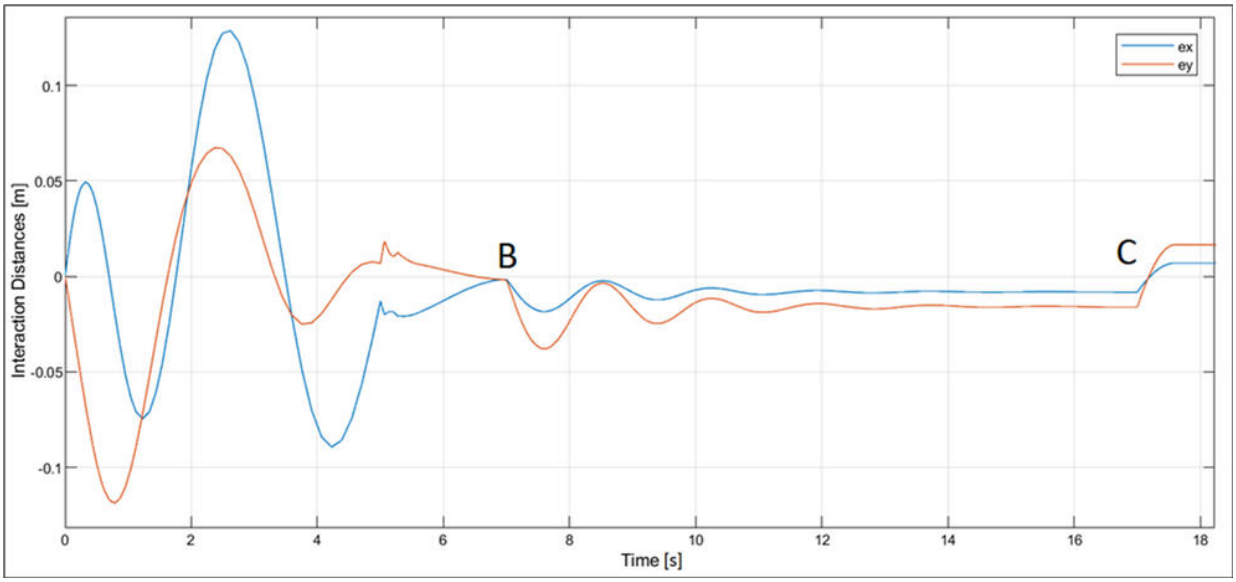

Figure 9. Interaction distances at the robot endpoint

Accordingly, while there are very large interaction distances before the contact with the BC plane, that is, before the 5th second, it is seen that the interaction distances are less when applying 50 N on the BC plane, that is, after the 5th second.

CONCLUSIONS

In this study, the position of the endpoint of a 2 dof robot was controlled using the impedance control technique. In addition, during the movement of the robot's end point on a plane, a change was made on the impedance force to apply a workload such as 50 N. During this movement, the workload is applied with an error between -1% and +1.6%.

In addition, the impedance controller parameters are optimized with The Bees Algorithm. As a result of the optimization made with a randomly determined initial condition, an improvement of 57% was achieved and the optimization was completed.

In this study, it has been shown that a force can be applied in a desired direction while the end point of the robot moves on a trajectory using the impedance control technique. In future studies, the change in desired position can be studied instead of changing the interaction force in this study.

REFERENCES

[1] Chen, H., Fuhlbrigge, T., Li, X., 2008, *Automated industrial robot path planning for spray painting process: a review*, Proc. 2008 IEEE International Conference on Automation Science and Engineering, IEEE, pp. 522-527.

[2] Muzan, I. W., Faisal, T., Al-Assadi, H., Iwan, M., 2012, *Implementation of industrial robot for painting applications*, Procedia engineering, 41(pp. 1329-1335.

[3] Asakawa, N., Takeuchi, Y., 1997, *Teachingless spray-painting of sculptured surface by an industrial robot*, Proc. Proceedings of international conference on robotics and automation, IEEE, pp. 1875-1879.

[4] Bicchi, A., Kumar, V., 2000, *Robotic grasping and contact: A review*, Proc. Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065), IEEE, pp. 348-353.

[5] Pliego-Jiménez, J., Arteaga-Pérez, M. A., 2015, *Adaptive position/force control for robot manipulators in contact with a rigid surface with uncertain parameters*, European Journal of Control, 22(pp. 1-12.

[6] Hogan, N., 1984, *Impedance control: An approach to manipulation*, Proc. 1984 American control conference, IEEE, pp. 304-313.

[7] Hogan, N., 1984, *Impedance control of industrial robots*, Robotics and computer-integrated manufacturing, 1(1), pp. 97-113.

[8] Hogan, N., 1987, *Stable execution of contact tasks using impedance control*, Proc. Proceedings. 1987 IEEE International Conference on Robotics and Automation, IEEE, pp. 1047-1054.

[9] Hayati, S., 1986, *Hybrid position/force control of multi-arm cooperating robots*, Proc. Proceedings. 1986 IEEE International Conference on Robotics and Automation, IEEE, pp. 82-89.

[10] Zeng, G., Hemami, A., 1997, *An overview of robot force control*, Robotica, 15(5), pp. 473-482.

[11] Erdemir, A., Kalyoncu, M., 2019, *Optimization of a Multi-Axle Steered Heavy Vehicle Steering Mechanism by using the Bees Algorithm and the Hooke-Jeeves Algorithms Simultaneously*, Proc. The 1st International Symposium On Automotive Science And Technology (ISASTECH 2019), Ankara/Turkey, pp. 613-622.

[12] Lau, T. L., 1999, *Guided genetic algorithm*, pp. v, 166 leaves.

[13] Ortmann, M., Weber, W., 2001, *Multi-criterion optimization of robot trajectories with evolutionary strategies*, Proc. Proceedings of the 2001 Genetic and Evolutionary Computation Conference. Late-Breaking Papers, Morgan Kaufmann San Francisco, CA, pp. 310-316.

[14] Wang, D., Tan, D., Liu, L., 2018, *Particle swarm optimization algorithm: an overview*, Soft computing, 22(2), pp. 387-408.

[15] Pham, D., Kalyoncu, M., 2009, *Optimisation of a fuzzy logic controller for a flexible single-link robot arm using the Bees Algorithm*, Proc. 2009 7th IEEE International Conference on Industrial Informatics, IEEE, pp. 475-480.

[16] Erdemir, A., Kalyoncu, M., 2015, *Bir Ağır Vasıtanın Çok Akslı Direksiyon Mekanizmasının Arı Algoritması Kullanılarak Optimizasyonu,* Uluslararası Katılımlı 17. Makina Teorisi Sempozyumu – (UMTS 2015), 17), pp. 421-426.

[17] Hooke, R., Jeeves, T. A., 1961, *``Direct Search''Solution of Numerical and Statistical Problems*, Journal of the ACM (JACM), 8(2), pp. 212-229.

[18] Ypma, T. J., 1995, *Historical development of the Newton–Raphson method*, SIAM review, 37(4), pp. 531-551.

[19] Erdemir, A., Alver, V., Kalyoncu, M., 2022, *Arı Algoritması Kullanılarak Önden Dümenlemeli Bir Aracın Dümenleme Mekanizmasının Optimizasyonu*, Proc. The 6th International Aegean Conferences on Innovation Technologies & Engineering, İzmir, Türkiye, pp. 50-59.