Journal of Nonlinear and Convex Analysis
Volume 26, Number 7, 2025, 2117–2126

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

KINEMATIC ANALYSIS OF A SEVEN DEGREE OF FREEDOM ROBOTIC ARM BASED ON GENETIC ALGORITHM

ZHILIN WANG, DONGKAI HE, AND DONGHAI SU*

Abstract. A numerical solution method based on genetic algorithm is proposed to address the issue of low accuracy in closed form solutions for the kinematics of a seven degree of freedom robotic arm. Based on the four key parameters of each degree of freedom, an objective function for the precision of mechanical pose in seven degrees of freedom was constructed. The objective function aims to optimize the pose of the end effector of a seven degree of freedom robotic arm, and continuously adjust the error between the actual pose and the expected pose using genetic algorithm. During the optimization process, the selection, crossover, evolution, and other operations of genetic algorithms are repeatedly used. During the experimental process, the population size and genetic generation of the genetic algorithm formed four different configurations. The experimental results show that compared to the closed solution method, the numerical solution method based on genetic algorithm has higher accuracy and can significantly improve the stability of the seven degree of freedom robotic arm.

1. Introduction

In recent years, extreme weather disasters such as typhoons and floods have occurred frequently worldwide, posing new challenges to post disaster rescue work [7]. For the increasingly severe rescue work, manual rescue seems inadequate. On the one hand, many natural disaster sites are inaccessible to humans or cannot be effectively rescued. On the other hand, rescue personnel have insufficient self-protection ability in high-risk environments, are prone to fatigue and unable to continue operations, and have low rescue efficiency [2]. In such a situation, using a mobile robot to carry a multifunctional robotic arm for rescue is a very ideal rescue solution. Multi functional robotic arms are responsible for actual operational tasks and are the core equipment for automated rescue [9].

At present, rescue robotic arms with three to six degrees of freedom are still mainly used domestically and internationally, which has certain limitations [13]. Compared to conventional robotic arms, seven degree of freedom robotic arms have more degrees of freedom. Therefore, the seven degree of freedom robotic arm has advantages such as good flexibility and fault tolerance [5]. Meanwhile, compared to ultra redundant robotic arms with more degrees of freedom, the seven degree of freedom robotic arm has better stiffness and more stable motion. Therefore, the

2020 *Mathematics Subject Classification.* 70B15, 70E60.

Key words and phrases. Seven degree of freedom robotic arm, genetic algorithm, closed form solution, numerical solution.

This study was supported by the National Natural Science Foundation of China (NSFC) with grant number 52275285, Science and Technology Plan Joint Program of Liaoning Province (Natural Science Foundation - Surface Project)(2024-MSLH-179).

*Corresponding author.

seven degree of freedom robotic arm is very suitable for rescue work in large and complex post disaster sites, and can replace manual operations such as demolition, cutting, grabbing, and handling [8]. The structure of post disaster sites is generally complex, and the working space of rescue equipment is limited. In this case, improving the stability and accuracy of the robotic arm's motion is particularly important.

To achieve stability and accuracy during the movement of a robotic arm, it is necessary to conduct in-depth research on the kinematics and dynamics of the robotic arm [6]. The inverse kinematics of a robotic arm refers to the inverse solution of the joint angles of each joint after knowing the position and attitude of a certain segment relative to the base coordinate system. There are many methods for solving the inverse kinematics of robotic arms, mainly divided into two categories: closed form solutions and numerical solutions [3, 1]. The conventional Jacobian pseudoinverse method is prone to sudden changes in joint angular velocity due to the strong singularity caused by redundant joints during its iterative solution process, making it difficult to converge [12]. The redundant angle algorithm based on self motion imposes more restrictions on joints and loses two degrees of freedom [11]. Numerical solutions based on intelligent optimization algorithms require strict control over the solving conditions [4]. The existence of redundant degrees of freedom further increases the complexity and difficulty of inverse kinematics solving. Northardt proposed a virtual control method for multi degree of freedom robots that does not use joints and is entirely based on linkages. This method has Lyapunov stability [10].

The seven degree of freedom robotic arm not only has redundant degrees of freedom, but also requires drive devices to be configured for each joint. So, the kinematic and dynamic analysis of a seven degree of freedom robotic arm is more complex. In order to solve the calculation problems of kinematics, we propose a genetic algorithm based inverse kinematics numerical solution method and compare it with the closed form solution method.

2. Proposed method

2.1. **Structure and parameter configuration of a seven degree of freedom robotic arm.** Multi degree of freedom robotic arm is a typical form of industrial robot, which is generally composed of multiple rods hinged together structurally. In this article, the D-H structure of a seven degree of freedom robotic arm is shown in Figure 1.

Table 1. Parameter configuration of seven degree of freedom robotic arm

i	$\alpha_{i-1}/^\circ$	α_{i-1}/mm	$\theta_i/^\circ$	d_i/mm	θ_i/mm
1	0	0	0	175	(-180,180)
2	-90	270	-90	0	(-97.5,-45.1)
3	0	870	90	0	(29.3,88.6)
4	0	1787	45	0	(-7.9,70.3)
5	0	1020	45	0	(-14.8,83.8)
6	90	210	-90	130	(58.5,138.4)
7	90	0	0	880	(-180,180)

From Figure 1, it can be seen that the key parameters included in the seven degree of freedom robotic arm are: link torsion angle α_{i-1}, link length a_{i-1}, joint angle θ_i, and link offset d_i. According to the needs of disaster environment rescue tasks and design requirements, these parameters are configured in each degree of freedom, as shown in Table 1.

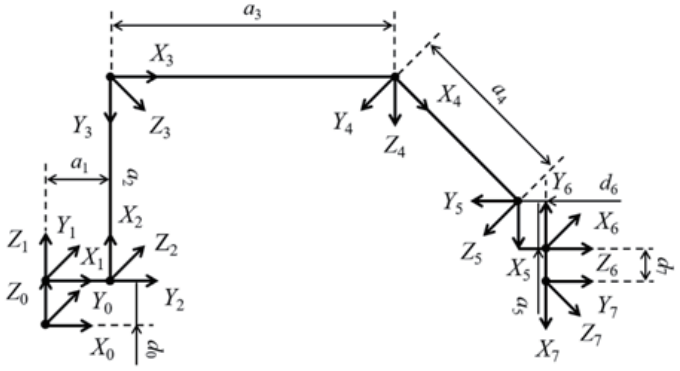

FIGURE 1. D-H structure of a seven degree of freedom robotic arm

At this point, the transformation matrix of the seven degree of freedom robotic arm end relative to the base surface system can be shown in Equation (2.1):

$$
{}^0_7\boldsymbol{T} = {}^0_1\boldsymbol{T}{}^1_2\boldsymbol{T}{}^2_3\boldsymbol{T}{}^3_4\boldsymbol{T}{}^4_5\boldsymbol{T}{}^5_6\boldsymbol{T}{}^6_7\boldsymbol{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.1}
$$

Based on the abstract structure in Figure 1 and the parameter configuration in Table 1, a kinematic simulation model of a seven degree of freedom robotic arm is established in the Matlab Robotics Toolbox environment, as shown in Figure 2.

2.2. **Kinematic calculation based on genetic algorithm.** Genetic algorithm is an intelligent optimization method based on the ideas of natural selection and genetic variation in the evolution of biological populations. The population in genetic algorithm corresponds to a set of solutions to the actual problem, while the individuals in the population correspond to a single solution to the problem. The chromosome of an individual corresponds to a specific format encoding of a single solution, and the fitness function value is used to determine the quality and probability of selecting a single solution.

Similar to global search methods such as gradient descent, genetic algorithms perform optimization through iterations. The fitness function is set based on the pose accuracy of the end effector of the robotic arm and the energy consumption of the robotic arm system.

Ensuring the pose accuracy of the end effector of the robotic arm is a prerequisite for achieving the expected planned motion of the robotic arm. Therefore, the fitness function should reflect the error between the actual pose and the expected pose of

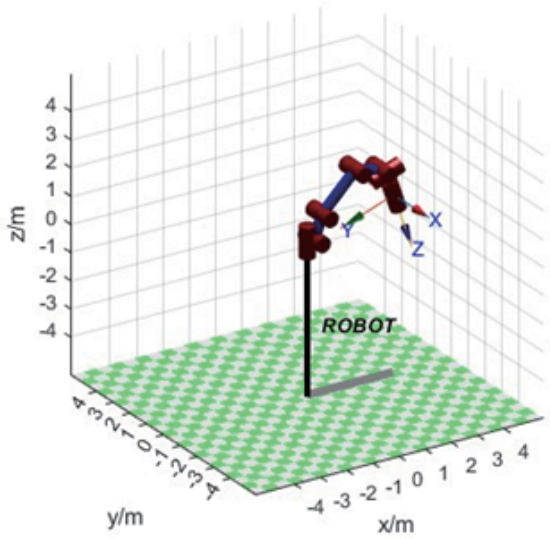

FIGURE 2. Kinematic simulation model of a seven degree of freedom robotic arm

the robotic arm in genetic algorithm, and select the inverse kinematics solution with the smallest error and highest accuracy.

The pose accuracy objective function of the robotic arm is shown in Equation (2.2):

$$P(T) = \|\mathrm{Tran}_{\mathrm{actl}} - \mathrm{Tran}_{\mathrm{exp}}\| = \left\| \begin{array}{cc} \Delta Rot & \Delta Pos \\ O & 1 \end{array} \right\|. \tag{2.2}$$

Here, $\mathrm{Tran}_{\mathrm{actl}}$ is the pose matrix of the actual pose of the end effector of the robotic arm; $\mathrm{Tran}_{\mathrm{exp}}$ is the pose matrix of the expected pose at the end of the robotic arm; ΔRot is the error between the actual posture and the expected posture of the end effector of the robotic arm; ΔPos is the error between the actual position and the expected position of the end effector of the robotic arm.

Simplify the pose accuracy objective function to the accumulation of pose error and position error, as shown in Equation (2.3)–(2.7):

$$F(T) = \|\Delta Rot\| + \|\Delta \,\mathrm{Pos}\,\|, \tag{2.3}$$

$$\|\Delta \text{ Rot } \| = \lambda_R(|\ \text{ Roll } - \alpha| + |\ \text{Pitch } - \beta| + |\ \text{Yaw } - \gamma\ |), \tag{2.4}$$

$$\lambda_R = \frac{7}{\sum_{i=1}^{7} (\theta_{i_{\max}} - \theta_{i_{\min}})}. \tag{2.5}$$

In the Equation,Roll,Pitch and Yaw represent the roll Angle, pitch Angle and yaw Angle corresponding to the actual attitude of the end of the robotics arm. α, β and γ represent the expected roll angle, pitch angle and yaw angle at the end of the robotic arm. λ_R represents the normalization coefficient of the end attitude error

of the robotics arm. θ_{imax} and θ_{imax} represent the maximum and minimum values of each of the seven joint angles of the robotic arm.

$$\|\Delta Pos\| = \lambda_P \sqrt{(x_{\text{actl}} - x_{\text{exp}})^2 + (y_{\text{actl}} - y_{\text{exp}})^2 + (z_{\text{actl}} - z_{\text{exp}})^2}, \tag{2.6}$$

$$\lambda_P = \frac{1}{Dis_{\max} - Dis_{\min}}. \tag{2.7}$$

In the Equation, x_{actl} , y_{actl} , and z_{actl} represent the actual x-axis, y-axis, and z-axis coordinates of the end effector of the robot arm, respectively; $x_{\text{exp}}, y_{\text{exp}}$ and z_{exp} represent the expected x-axis, y-axis, and z-axis coordinates of the end effector of the robot arm, respectively; λ_P is the normalized coefficient of the end effector position error; $Dis_{\max}$ and $Dis_{\min}$ represent the maximum and minimum distances between the end effector of the robot arm and the origin of the relative reference coordinate system, respectively.

In order for the robot arm to work for longer periods of time, it is essential to reduce its energy consumption. To evaluate the relative energy consumption of the robotic arm during the process of moving from the actual pose to the desired pose at the previous moment, it can be briefly analyzed from two aspects: joint rotation angle and link length.

Based on the joint rotation angle, energy consumption analysis of the system is conducted. The smaller the joint rotation angle, the smaller the joint torque and the lower the system energy consumption. In addition, if the other joints stop moving, the near joint near the base of the robotic arm or the far joint near the end of the robotic arm will rotate by the same angle, because the rotation of the near joint will drive all the connecting rods to move and the load will be greater, so the energy consumption of the near joint system is higher than that of the far joint. To reduce system energy consumption, inverse kinematics solutions with small near joint rotation angles and large far joint rotation angles should be selected.

Based on the analysis of system energy consumption based on the length of the connecting rod, when the joints of the robotic arm rotate a fixed angle, the longer the connecting rod corresponding to each joint, the greater the joint torque that drives the connecting rod movement, and the more energy is consumed by the system. To reduce system energy consumption, inverse kinematics solutions with small rotation angles for long link joints and large rotation angles for short link joints should be selected.

According to the above principles, establish a system energy consumption objective function, as shown in Equation (2.8):

$$F(\varepsilon) = \sum_{i=1}^{7} A_i L_i \left(\theta_i - \theta_{i\text{pre}}\right)^2. \tag{2.8}$$

In the above Equation, θ_i, and θ_{ipre} represent the current Angle of the i-th joint in the robotic arm and the angle of the previous attitude; A_i represents the joint position weighting coefficient. The closer the joint is to the base, the larger the coefficient is, and the farther the joint is from the base, the smaller the coefficient is. L_i represents the weighting coefficient of link length, which increases with the relative increase of link length.

3. Experimental results and analysis

For the solution of inverse kinematics, the closed solution based on dynamic pose has a limited number of derived results compared to the numerical solution based on genetic algorithm, and from the perspective of system energy consumption, the results of the closed solution may not be the optimal solution. Genetic algorithm can not only obtain the optimal solution under actual conditions based on the accuracy of the end pose and the reduction of system energy consumption, but also adjust the direction of population evolution by constructing different fitness functions to obtain the optimal solution that meets different needs, making it more widely applicable. Therefore, the experiment here mainly compares our proposed method with the closed solution method.

Set the joint angle of the robotic arm at the previous moment to:

$$\theta = \left[\begin{array}{ccccccc} 0.2000 & -1.0472 & 1.0472 & 0.7854 & 0.5236 & 1.5708 & 1.0000 \end{array} \right].$$

The expected pose matrix of the end effector of the robotic arm is as follows:

$$^{0}_{7}T = \left[\begin{array}{cccc} 0.0269 & -0.9962 & -0.0829 & 3.2887 \\ -0.9933 & -0.0173 & -0.1146 & -0.0790 \\ 0.1128 & 0.0854 & -0.9899 & -0.9791 \\ 0 & 0 & 0 & 1.0000 \end{array} \right].$$

The expected pitch angle α corresponding to the end pose of the robotic arm is 3.0555 rad , the yaw angle β is -0.1130 rad , and the roll angle γ is -1.5437 rad ; The expected position of the end effector of the robotic arm corresponds to an x-axis coordinate of 3.2887 m , a y-axis coordinate of -0.0790 m , and a z-axis coordinate of -0.9791 m; Based on $\boldsymbol{\theta}$ and $^{0}_{7}\boldsymbol{T}$, the dynamic pose method independently calculates closed solutions and records the computation time of the Matlab program.

The computation time for the closed solution is 0.012s, and the angles of each joint are as follows:

$$\theta_a = \left[\begin{array}{ccccccc} 0 & -0.7854 & 0.6981 & 1.0472 & 0.6981 & 1.2566 & 3.1416 \end{array} \right].$$

Taking the pose accuracy weighting coefficient $\alpha_{\mathrm{a}} = 0.5$, we can obtain $\mathrm{Fit}\left((\boldsymbol{T}_{\mathrm{a}}, \boldsymbol{\theta}_{\mathrm{a}})\right) = \alpha f(\boldsymbol{T}) + (1-\alpha) f\left(\boldsymbol{\theta}_{\mathrm{a}}\right) = 0.2198$.

(1) The first set of experiments

The population size set is $P = 60$, chromosome length $C = 60$, and the encoding format is binary encoding. When the upper limit of evolutionary algebra $G = 50$, independent numerical solutions are obtained 100 times, and the average computation time of Matlab program is 10.831 s . The evolution process of the optimal solution genetic algorithm is shown in Figure 3:

The angles of each joint are as follows:

$$\theta_{a1} = \left[\begin{array}{ccccccc} 0.0990 & -1.0474 & 1.0916 & 0.9417 & 0.5032 & 1.0948 & -3.0590 \end{array} \right].$$

The pose accuracy objective function $f\left(\boldsymbol{T}_{\mathrm{n1}}\right)$ is 0.0072 , and the system energy consumption objective function $f\left(\theta_{\mathrm{n1}}\right)$ is 0.0968 . Taking the pose accuracy weighting coefficient $\alpha_{\mathrm{n}} = 0.5$, the fitness function $\mathrm{Fit}\left((\boldsymbol{T}_{\mathrm{n1}}, \boldsymbol{\theta}_{\mathrm{n1}})\right) = 0.0832$ can be obtained.

(2) The second set of experiments

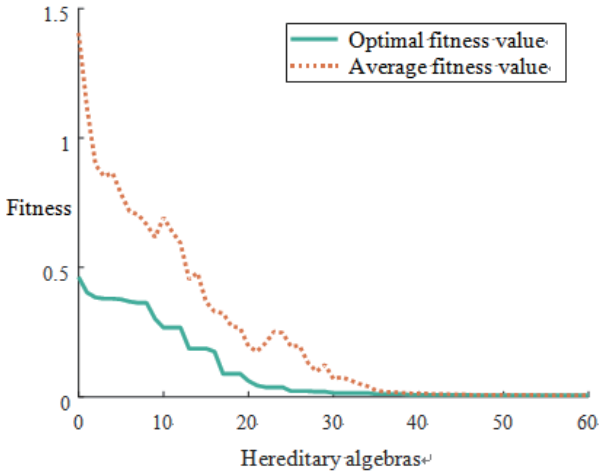

FIGURE 3. Evolution diagram of genetic algorithm with $P = 60$ and $G = 50$

This article sets the population size $P = 100$, chromosome length $C = 60$, and the encoding format to binary encoding. When the upper limit of evolutionary algebra $G = 50$, independent numerical solutions are obtained 100 times, and the average computation time of Matlab program is 16.438s. The evolution process of the optimal solution genetic algorithm is shown in Figure 4:

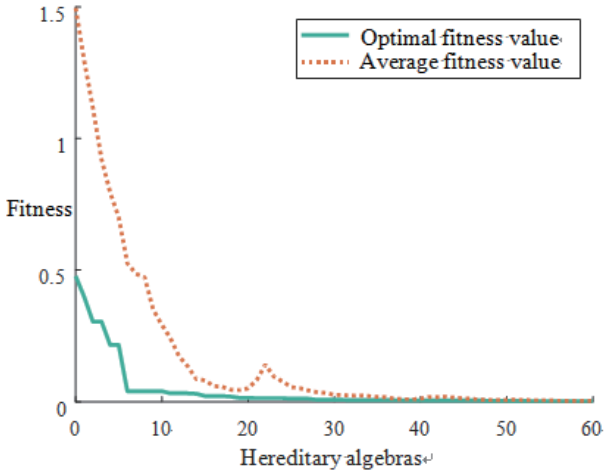

FIGURE 4. Genetic Algorithm Evolution Diagram with $P = 100$ and $G = 50$

The angles of each joint are as follows:

$$\theta_{a2} = \left[\begin{array}{ccccccc} -0.0009 & -0.7651 & 0.6876 & 0.9817 & 0.8079 & 1.4843 & 3.1272 \end{array} \right].$$

Taking the pose accuracy weighting coefficient $\alpha_{\mathrm{n}} = 0.5$, the fitness function $\mathrm{Fit}\left((\boldsymbol{T}_{\mathrm{n3}}, \boldsymbol{\theta}_{\mathrm{n3}})\right) = 0.0641$ can be obtained.

(3) The third set of experiment

This article sets the population size $P = 60$, chromosome length $C = 60$, and the encoding format is binary encoding. When the upper limit of evolutionary algebra $G = 80$, independent numerical solutions are obtained 100 times, and the average computation time of Matlab program is 16.651 seconds.

The angles of each joint are as follows:

$$\theta_{a3} = \left[\begin{array}{ccccccc} -0.1721 & -1.1114 & 1.3024 & 0.4912 & 0.9722 & 2.1549 & 3.0320 \end{array} \right].$$

The pose accuracy objective function $f(\boldsymbol{T}_{\mathrm{n3}})$ is 0.0035 , and the system energy consumption objective function $f(\boldsymbol{\theta}_{\mathrm{n3}})$ is 0.0424 . Taking the pose accuracy weighting coefficient $\alpha_{\mathrm{n}} = 0.5$, the fitness function $\mathrm{Fit}((\boldsymbol{T}_{\mathrm{n3}}, \boldsymbol{\theta}_{\mathrm{n3}})) = 0.0251$ can be obtained.

(4) The fourth set of experiments

The population size set in this article is $P = 100$, the chromosome length is $C = 60$, the encoding format is binary encoding, and the upper limit of evolutionary algebra is $G = 80$. When independently solving the numerical value 100 times, the average computation time of the Matlab program is 25.968 seconds.

The angles of each joint are as follows:

$$\theta_{a4} = \left[\begin{array}{ccccccc} -0.2280 & -0.8044 & 0.8700 & 0.9533 & 0.6793 & 2.4298 & 2.8666 \end{array} \right].$$

Taking the pose accuracy weighting coefficient $\alpha_{\mathrm{n}} = 0.5$, the fitness function $\mathrm{Fit}((\boldsymbol{T}_{\mathrm{n4}}, \boldsymbol{\theta}_{\mathrm{n4}})) = 0.0194$ can be obtained.

The comparison between the above four sets of experimental data and the closed form solution is shown in Table 2.

TABLE 2. Comparison of Closed Solutions and Proposed Method Numerical Solutions

Index	closed-form solution	Ours $P = 60, G = 50$	Ours $P = 100, G = 50$	Ours $P = 60, G = 80$	Ours $P = 100, G = 80$
Time: t/s	0.012	10.831	16.438	16.651	25.968
Accuracy: $f(\boldsymbol{T})$	0	0.0072	0.0055	0.0035	0.0014
Energy: $f(\boldsymbol{\theta})$	0.3382	0.0968	0.0830	0.0424	0.0334
Fitness: Fit $((\boldsymbol{T} : \boldsymbol{\theta}))$	0.2198	0.0832	0.0641	0.0251	0.0194

According to the results in Table 2, it can be seen that:

Firstly, regarding the average computation time t, the closed solution solving time based on dynamic pose method is much lower than the numerical solution solving time based on genetic algorithm;

Secondly, regarding the pose accuracy objective function $f(T)$, the closed solution is almost zero and smaller than the numerical solutions. The maximum difference between the numerical solution and the closed solution is 0.0072 , and the minimum difference is 0.0014 ;

Thirdly, regarding the energy consumption objective function $f(\theta)$ of the system, the closed solution is greater than the numerical solutions. The maximum difference

between the numerical solution and the closed solution is 0.3048 , and the minimum difference is 0.2414 ;

Fourthly, regarding the fitness function Fit($(\boldsymbol{T}, \boldsymbol{\theta})$), the closed solution is greater than the numerical solutions, and the maximum difference between the numerical solution and the closed solution is 0.2004 , while the minimum difference is 0.1366 .

4. Conclusions

The seven degree of freedom robotic arm has better flexibility and practicality for rescue work in disaster environments. In order to improve the stability and accuracy of the seven degree of freedom robotic arm, a kinematic solution based on genetic algorithm was established. Based on the four key parameters of each degree of freedom, an objective function for the precision of mechanical pose in seven degrees of freedom was constructed. The objective function aims to optimize the pose of the end effector of a seven degree of freedom robotic arm, and continuously adjust the error between the actual pose and the expected pose using genetic algorithm. During the optimization process, the selection, crossover, evolution, and other operations of genetic algorithms are repeatedly used. During the experimental process, the population size and genetic generation of the genetic algorithm formed four different configurations. The experimental results show that the numerical solution method based on genetic algorithm has significantly improved accuracy compared to closed solutions, and can adjust the population size and genetic algebra according to the actual situation, making the seven degree of freedom robotic arm more stable.

References

[1] S. Alaci, F. C. Ciornei and M. C. Romanu, *Solution for the Kinematics of Non-H-D Couplings Applied to RPCR Mechanism.* Axioms **12** (2023): 357.

[2] Z. T. Alali, S. A. Alabady and T. Glade, *Techniques and methods for managing disasters and critical situations*, Natural Hazards **120** (2024), 6943–6989.

[3] M. Bombile and A. Billard, *Bimanual dynamic grabbing and tossing of objects onto a moving target*, Robotics and Autonomous Systems **167** (2023): 104481.

[4] T. Das, L. Gohain and N. M. Kakoty, *Hierarchical approach for fusion of electroencephalography and electromyography for predicting finger movements and kinematics using deep learning*, Neurocomputing **527** (2023), 184–195.

[5] Y. H. Deng and J. Y. Chang, *Human-like posture correction for seven-degree-of-freedom robotic arm*, Journal of Mechanisms and Robotics **14** (2022): 024501.

[6] G. Du and J. Lu, *Based on artificial bee colony algorithm for spatial layout planning of urban public service facilities*, Journal of Nonlinear and Convex Analysis, **23** (2022), 2449–2458.

[7] O. F. Goenen, T. Cavusoglu and E. Kurnaz, *Medical rescue team's experiences in disaster response to individuals with special needs: The case of Turkiye*, Natural Hazards **120** (2024), 1841–1867.

[8] A. Joseph, M. Ismail and O. Edward, *A multi-function two degree of freedom robotic arm for simple grip and shake action*, Journal of Electrical Engineering and Electronic Technology, **12** (2023), 1–5.

[9] J. S. Kubicko, C. M. Verlinden, J. Sarkar, K. G. Sabra, B. V. Nichols, J. S. Martin and A. I. Fagan, *Information content of ship noise on a drifting volumetric array for passive environmental sensing*, IEEE Journal of Oceanic Engineering, **45** (2020), 607–630.

[10] T. Northardt, *Observability criterion guidance for passive towed array sonar tracking*, IEEE Transactions on Aerospace and Electronic Systems **58** (2022), 3578–3585.

[11] M. Sekiguchi and N. Takesue, *Formulation method of robot kinematics considering versatility and mobility of each joint*, Journal of the Robotics Society of Japan, **41** (2023), 175–186.
[12] A. V. S. S. Somasundar and G. Yedukondalu, *Six DOF robot: inverse kinematics solution to path planning for intersecting pipes for welding operation and inverse Jacobian comparison*, International Journal on Interactive Design and Manufacturing (IJIDeM), **18** (2024), 3313–3322.
[13] X. Zhou, Y. Luo and Y. Gao, *Development and application of a mechanical arm-based in situ 3D bioprinting method for the repair of skin wounds*, Discover Applied Sciences, **6** (2024): article number 438.

Manuscript received September 24, 2024
revised January 14, 2025

Zhilin Wang
College of Mechanical Engineering, Shenyang University of Technology, Shenyang, Liaoning Province, 110870, China;
Liaodong University, Dandong, Liaoning Province, 118001, China
E-mail address: 89312074@qq.com

Dongkai He
College of Mechanical & Electrical Engineering, Hohai University, Changzhou 213022, Jiangsu Province, P. R. China
E-mail address: hedongkai_2000@163.com

Donghai Su
College of Mechanical Engineering, Shenyang University of Technology, Shenyang, Liaoning Province, 110870, China
E-mail address: sudonghai2024@126.com