MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 4, pp. 946–953 (2024)

The power of metaheuristic algorithms for robotics: singularity & trajectory

Harrade I., Kmich M., Sayyouri M., Chalh Z.

National School of Applied Sciences,
Sidi Mohamed Ben Abdellah-Fez University,
Engineering Systems and Applications Laboratory, Fez, Morocco

(Received 7 January 2024; Revised 16 August 2024; Accepted 2 September 2024)

When calculating the kinematic model of any kind of robot, parallel or planar, the singularity problem frequently crops up. We propose the application of metaheuristic algorithms to identify the needed target to solve this issue and minimize calculus. Simulation results using several metaheuristic algorithms (MA) on the same population have been obtained with reduced computing time (0.50 s). The efficacy of the suggested technique for maximizing the position and trajectory of the joints in a 3-DOF or 3-RRR (with three rotational degrees of freedom) planar parallel manipulator robot is amply illustrated by them. The sine-cosine algorithm (SCA) and certain target points are essentially the basis of the method, which determines the optimal desired path. These outcomes show how well the suggested strategy works for maximizing calculations, positions, and the ideal robot trajectory.

Keywords: *3-RRR planar parallel manipulator robot; 3-DOF Robot; metaheuristic algorithms; avoid singularity; joint position; optimal trajectory; visual servoing.*

2010 MSC: 93C85, 70B15, 65D19, 37N35, 49Jxx **DOI:** 10.23939/mmc2024.04.946

1. Introduction

Between robotics, automation, and computer vision is a field of technology known as visual servoing. It replaces the conventional control methods that have been so crucial to industry, especially robotics. This new technology is based on the integration of visual data extracted from vision sensor images into the robot's control loop, instructing the robot to carry out the required actions [1].

A robot can be defined using dynamic, geometric, or kinematic models [2]. One must comprehend the dynamics and kinematics of the manipulator in issue to design an effective control strategy for a robot. To do this, we must formulate the mathematical formulas that will enable us to define the movements of the manipulator in space.

Based on these calculations, we use complicated multivariable functions, leading us to the singularity problem. Scientists have suggested solutions because this issue can result in a wide variety of setups [3]. In robot control, the singularity issue is frequently encountered, either for a robot plan manipulator (as in the case studied in article [4]), or for a parallel manipulator, which is our case study in this article. Throughout this work, we suggest applying the SCA metaheuristic algorithm as a solution to this issue. This technique minimizes the robot's command graph and lowers computational complexity to avoid singularities while determining the best trajectory [5]. In addition, it is easy to implement, efficient in determining joint angle values, and fast because only joint angle intervals need to be specified. Furthermore, robot control technology based on vision is constantly evolving to incorporate different methods, such as neural network methods [6], using to replace conventional information processing methods for object detection. And it also exists fuzzy logic control [7] for improved system control. The remainder of this article is structured like this: a synopsis of the 3-PPP parallel manipulator robot is given in Section 2. Section 3 outlines the suggested SCA-based methodology. The robot's kinematic model and functional block diagram are shown in Section 4. In Section 5, we present the results of our numerical simulations. Section 6 concludes the paper.

2. 3RRR modeling robot

In this study, we suggest using Figure 1 diagram of a 3RRR parallel robot system.

The system is comprised of movable triangular equilaterals and a mobile terminal unit $(P_1' \, P_2' \, P_3')$, whose three legs and stationary base hold it to the earth. Each chain is made up of two links and three pairs of rotors, as follows: $B_1(C_1') \, B_2(C_2') \, B_3(C_3')$. Subscript between, the kinematic chain is made up of the $(D_h' \, P_h')$ distant link and the $B_h(C_h') \, D_h'$ proximal link, with the rotary and subscript between $B_h(C_h') \, D_h'$ and P_h', with $h = 1, 2, 3$.

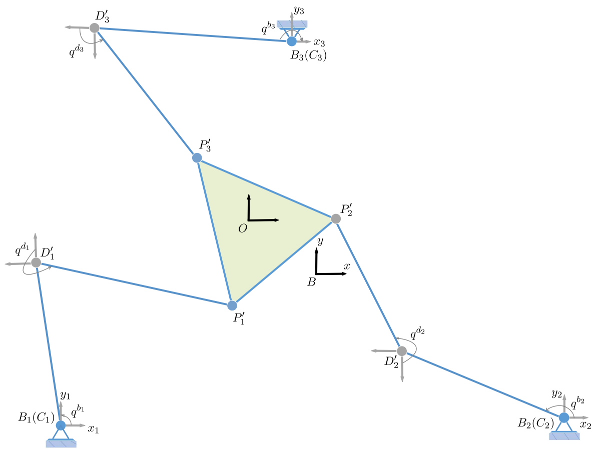


Fig. 1. 3RRR parallele robot.

3. Metaheuristic algorithms for the 3R parallel robot

The metaheuristic algorithm (MA) is a decision-making process that has become ingrained in human nature. Finding the optimal option to raise the performance of the system under examination can be seen as a decision-making challenge. In general, MA evaluates the objective function to achieve the global optimum. MAs come in a variety of forms, from straightforward local searches to intricate global search algorithms [8]. There are several natural characteristics as well. We can mention, for instance; Moth Flame Optimization Algorithm (MFO) [9], Artificial Bee Colony algorithm (ABC) [10], Coronavirus Search Optimizer (CVSO) [11], Ant Colony Optimization (ACO) algorithm [12], and the Harmonic Search (HS) [13]. These are regarded as the most well-liked MAs available right now. Since SCA is more consistent with our research, we will first concentrate on it in the following and then compare it to other MAs.

3.1. The sine cosine algorithm (SCA)

In numerous engineering domains, the SCA has been recognized as one of the most sophisticated and effective techniques for addressing optimization issues in contemporary optimization approaches [14]. The method is based on a trigonometric function model (sine and cosine), allowing for the generation of multiple solutions through multiple iterations. During each iteration, each solution is updated based on the sine or cosine function, according to these two mathematical expressions:

$$H^{(t+1)}(j,k) = H^t(j,k) + \left(v_1 * \sin(v_2) * \text{abs}\left(v_3 * Tgt(k)^t - H^t(j,k)\right)\right), \tag{1}$$

$$H^{(t+1)}(j,k) = H^t(j,k) + \left(v_1 * \cos(v_2) * \text{abs}(v_3 * Tgt(k)^t - H^t(j,k))\right). \tag{2}$$

Featuring, $H^{t+1}(j,K)$ is the current solution's position solution (j) at the dimension (k) in the iteration (t); $Tgt(K)^t$ is the best individual's position in dimension (j) in the iteration (t); abs is the symbol for absolute value; v_1, v_2 and v_3 are three random variables.

In the present case, v_1 establishes the range's magnitude for sin and cos, v_2 specifies the range for either sin or cos, and v_3 is utilized to define the destination and the solution's new location. Here, the SCA makes use of v_4, a random number between 0 and 1, along with the presiding equations (1) and (2) as follows [15]:

$$H^{(t+1)}(j,k) = H^t(j,k) + (v_1 * \sin(v_2)) * \left|\left(v_3 * Tgt(k)^t - H^t(j,k)\right)\right|, \quad v_4 < 0.5, \tag{3}$$

$$H^{(t+1)}(j,k) = H^t(j,k) + (v_1 * \cos(v_2)) * \left|\left(v_3 * Tgt(k)^t - H^t(j,k)\right)\right|, \quad v_4 \geqslant 0.5. \tag{4}$$

To ensure operational and research stability, both sinusoidal and cosinusoidal amplitudes have been changed adaptively at formulas (1) and (2), as the subsequent equation illustrates:

$$v_1 = Z - i * \frac{z}{N}. \tag{5}$$

Assuming that: i is actual iteration, N is maximum number of iterations, z is constant.

The search strategy used by SCA to reduce error during search iterations and identify the ideal resolution is displayed in Figure 2.

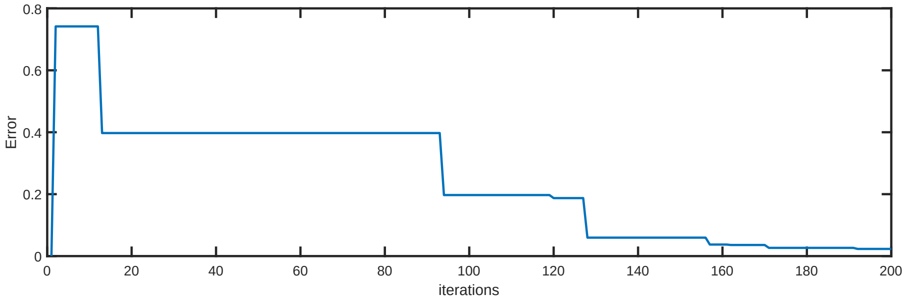


Fig. 2. Algorithm β for minimizing errors over search iterations.

As the literature shows, more iterations mean fewer errors. In our case, we obtain optimal results with SCA after 150 iterations. The suggested pseudo-code for operating a 3RRR robot with SCA can be summed up as follows by Algorithm 1.

Algorithm 1 The robot is controlled by the SCA in pseudo-code.

Inputs: Target coordinates of selected points (M_0: $W(x_w, y_w)$, $X(x_X, y_X)$, $Y(x_Y, y_Y)$, $Z(x_Z, y_Z)$), in addition to the 3RRR robot's homogeneous matrix.
Output: Articulation angle values (q_{d1}, q_{d2}, q_{d3}), and the estimated primitives (M: $\hat{W}(x_W, y_W)$, $\hat{X}(x_X, y_X)$, $\hat{Y}(x_Y, y_Y)$, $\hat{Z}(x_Z, y_Z)$).
Initialization of SCA
$\boldsymbol{A} = 300$ (Number of search staff)
$\boldsymbol{I} = 200$ (Maximum iteration number)
$\boldsymbol{D} = 3$ (Problem dimension): Our goal is to find the 3RRR robot's joint angle values (q_{d1}, q_{d2}, q_{d3}), of the 3R robot.
Min $= \left[-\pi, \frac{-\pi}{2}, \frac{-\pi}{2}\right]$ (Lowest possible values for (q_{d1}, q_{d2}, q_{d3}))
Ub $= \left[\pi, \frac{\pi}{2}, \frac{\pi}{2}\right]$ (Highest possible values for (q_{d1}, q_{d2}, q_{d3}))
Set up an initial MA set of search agents.
Do
  Determine, the quantity of research agents for each $\hat{W}(x_W, y_W)$, $\hat{X}(x_X, y_X)$, $\hat{Y}(x_Y, y_Y)$, $\hat{Z}(x_Z, y_Z)$.
  Assess the objective function for each research agent as **provided** by S (Eq. (6)).
  Comply with SCA's instructions.
  Update the most effective answer found to date (M_0, M).
  Refresh the estimated values of primitives.
  Modify where (q_{d1}, q_{d2}, q_{d3}) is located.
  While$(i < I)$
End Do
Return the successful solution discovered once the global optimum has been achieved.

The objective function that will be assessed is determined by the subsequent formula:

$$S = \sum_{j=0}^{8} |M_0(j) - M(j)|, \tag{6}$$

with: M_0 is the desired primitive and M is the SCA calculated primitive

4. Robotic control architecture

This section provides a brief overview of robot modeling and control methods, including visual control methods that allow a robot to be controlled in a closed loop by visualizing sensor data [16].

4.1. Robot modeling

Understanding the kinematics and dynamics of the robot under consideration is crucial for the development of effective control structures for robots. Usually, this requires describing the different mathematical equations that can be used to describe the robot's motion through space [17]. The existing robotic approaches define the robot's movement on the space plane by using transformation models between operating space (which defines the position of the end organs) and joint space (which defines the robot's configuration). There are various models, notably:

- Geometry models (inverse or direct); representing the body position as a function of the robot configuration, or the opposite [18, 19].
- Kinematic models (direct or inverse); to represent the speed of the terminal element as a function of the joint speed, or the opposite [20].
- The dynamic model determines the robot's equations for motion, which establishes the connection between the torque or force exerted by the actuator and the position, velocity, and acceleration of the joint [21].

4.2. Control robot testing methods

Once the robot model is perfectly defined, various control strategies can be applied. Nevertheless, it is necessary to adapt the control because this theoretical requirement can never be entirely achieved in practice due to the several disturbance variables operating on the robot manipulator and the uncertainty in the model [16]. Robots are controlled by a multitude of technologies. The mechanical architecture of the robot affects the chosen control method. The fundamental component of a manipulator robot is a complex mechanical construction with joint axis inertia that depends on its configuration, speed, and acceleration in addition to the force at play [22]. The most commonly used controllers for robot manipulators in industrial applications are traditional or basic control [23], adaptive control [24], and dynamic control [25, 26].

4.3. Kinematic control

In this paper, we control our 3RRR robot with a kinematic servo. Figure 3 presents the block diagram that follows for this purpose.

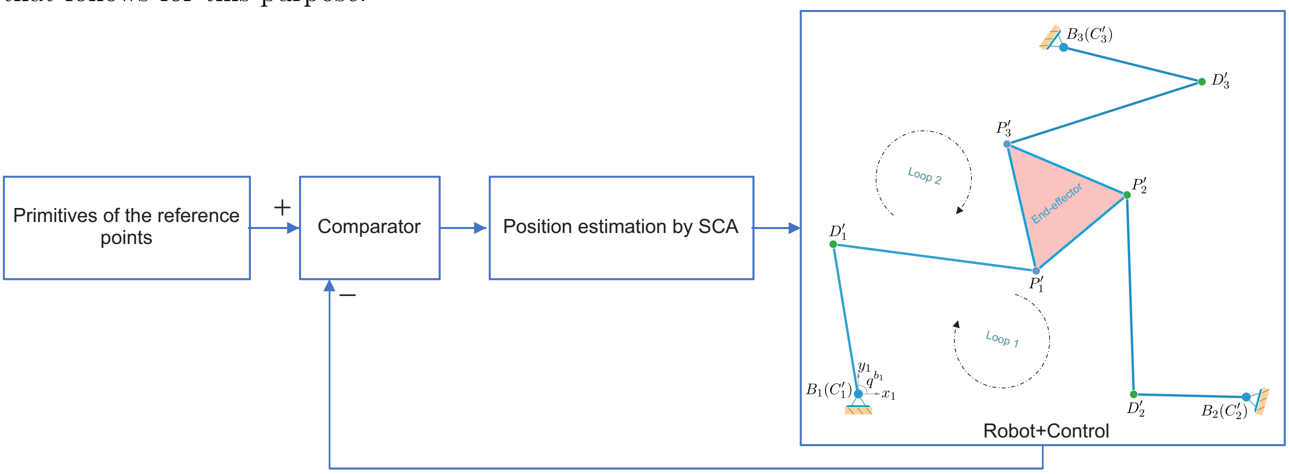


Fig. 3. Diagram for parallel robot control.

When the acquired target converges towards the intended target, the optimal control [27] is matched to the coordinates of the chosen points. Note that the degree of robot controllability used in this article corresponds to the translation speed around the x-axis and the rotation speed along the y-axis.

5. Simulation and outcomes

Section presents the results of a 3RRR parallel robot simulation, based on SCA for calculating the joint angle. First, we use SCA to simulate the robot's position. After that, we contrasted the SCA with alternative MAs.

5.1. Using SCA to calculate articulation

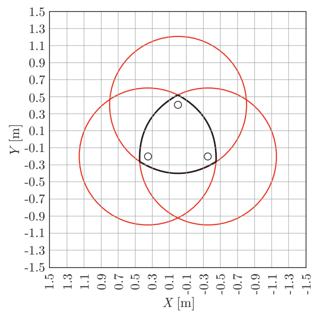
This section describes how to use the calculation inputs of SCA to determine the best locations to set up the robot's articulation angles and provide evidence of how the proposed Algorithm β is robust.

We propose the algorithms CVSA [11], ABC [10], MFO [9], and HS [12] to efficiently analyze and interpret the SCA results. We tested SCA with these algorithms using an equal number of populations and iterations. Consequently, the computation time varies depending on the kind of algorithm. Additionally, we computed the relative value error between the values required and those determined by the following equation:

$$R = \sum_{j=0}^{8} \left| M_0(j) - M(j) \right| / M_0(j). \tag{7}$$

The results of combining the proposed SCA algorithm with the other algorithms, to analyze the performance of the proposed algorithm by comparing the approximation error of the desired position and velocity with the other algorithms, are presented in Table 1 below.

Table 1. Comparison between five MAs.

Algorithm for calculating the target	Errors (R)	Time of calculation (seconds)	Initial position
SCA	0	0.50	
CVSO	0	0.52	
ABC	0.20	60.20	
MFO	0	2.52	
HS	0	1.02	

The results obtained show that the 3RRR Robot Behavior control loop is achieved by using the SCA as well as certain algorithms (CVSO, ABC, ACO, and MFO). This illustrates how the proposed approach can control the trajectory by optimizing the position of the joints, and also how the SCA can solve a singularity problem. In addition, Table 1 shows the speed of ACS compared with other algorithms (such as "ABC"), with errors reduced to almost zero. Running the simulation gives us the robot animation shown in Figure 4.

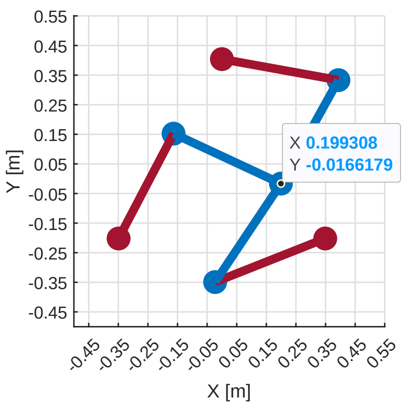


Fig. 4. Robot animation in the initial position.

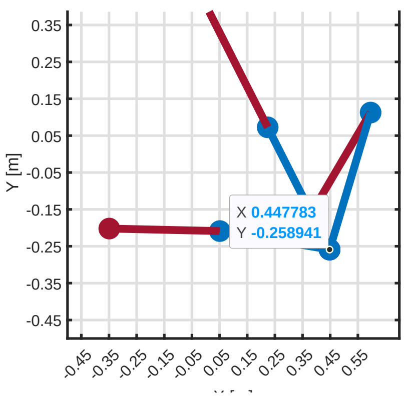


Fig. 5. Robot animation in the target position.

The results show that the joints converge in the direction of the needed target. When using MAs, we obtain a more rapid convergence of the 3 active joints compared to classical methods; in terms of calculation time, as shown in Table 1 (for example, the calculation time for the SCA is 0.50 s).

The convergence of the active arms as a function of position, velocity, and acceleration is depicted in the figures below; this demonstrates the efficient application of the control techniques. The efficiency of the suggested approach for calculating joints is amply illustrated by these outcomes.

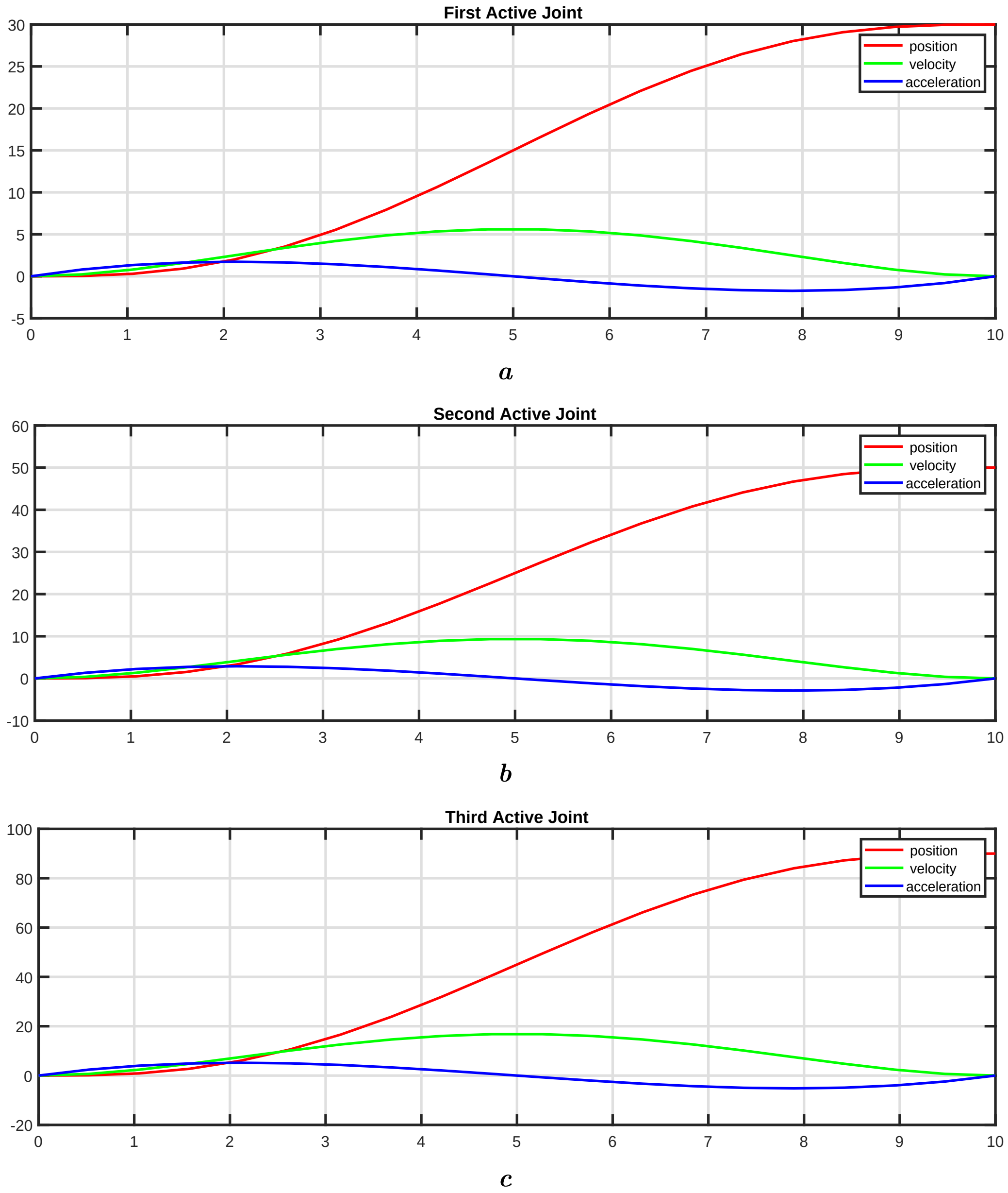


Fig. 6. (***a***) First active joint, (***b***) Second active joint, (***c***) Third active joint.

6. Conclusion

The robot convergence error and computation rate perfectly illustrate the proposed optimization technique, to converge towards the desired target. The suggested computational approach is validated by simulation results when compared to other popular computational methods. In the future, we can focus on the dimensional design of parallel robots, which can also be optimized using the proposed method.

[1] Shi H., Li R., Bai X., Zhang Y., Min L., Wang D., Lu X., Yan Y., Lei Y. A review for control theory and condition monitoring on construction robots. Journal of Field Robotics. **40** (4), 934–954 (2023).

[2] Choudhury R., Singh Y. Planar parallel manipulators: A review on kinematic, dynamic, and control aspects. Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering. (2023).

[3] Liu S., Qiu Z., Zhang X. Singularity and path-planning with the working mode conversion of a 3-DOF 3-RRR planar parallel manipulator. Mechanism and Machine Theory. **107**, 166–182 (2017).

[4] Harrade I., Daoui A., Chalh Z., Sayyouri M. Visual Servoing of a 3R Robot by Metaheuristic Algorithms. Statistics, Optimization & Information Computing. **11** (1), 116–124 (2023).

[5] Hubert J. Parallel manipulators, singularities and static analysis. PhD Thesis, École Nationale Supérieure des Mines de Paris (2010).

[6] Guo J., Nguyen H. T., Liu C., Cheah C. C. Convolutional Neural Network-Based Robot Control for an Eye-in-Hand Camera. IEEE Transactions on Systems, Man, and Cybernetics: Systems. **53** (8), 4764–4775 (2023).

[7] Chotikunnan P., Pititheeraphab Y. Adaptive P Control and Adaptive Fuzzy Logic Controller with Expert System Imple-mentation for Robotic Manipulator Application. Journal of Robotics and Control (JRC). **4** (2), 217–226 (2023).

[8] Turgut O. E., Turgut M. S., Kirtepe E. A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems. Neural Computing and Applications. **35**, 14275–14378 (2023).

[9] Sahoo S. K., Saha A. K., Ezugwu A. E., Agushaka J. O., Abuhaija B., Alsoud A. R., Abualigah L. Moth Flame Optimization: Theory, Modifications, Hybridizations, and Applications. Archives of Computational Methods in Engineering. **30**, 391–426 (2023).

[10] Ganguli C., Shandilya S. K., Nehrey M., Havryliuk M. Adaptive Artificial Bee Colony Algorithm for Nature-Inspired Cyber Defense. Systems. **11** (1), 27 (2023).

[11] Golalipour K., Faraji Davoudkhani I., Nasri S., Naderipour A., Mirjalili S., Abdelaziz A. Y., El-Shahat A. The corona virus search optimizer for solving global and engineering optimization problems. Alexandria Engineering Journal. **78**, 614–642 (2023).

[12] El Asri F., Tajani C., Fakhouri H. Investigation of ant colony optimization with Levy flight technique for a class of stochastic combinatorial optimization problem. Mathematical Modeling and Computing. **10** (4), 1132–1142 (2023).

[13] Brambila-Hernández J. A., García-Morales M. Á., Fraire-Huacuja H. J., Villegas-Huerta E., Becerra-del-Ángel A. Hybrid Harmony Search Optimization Algorithm for Continuous Functions. Mathematical and Computational Applications. **28** (2), 29 (2023).

[14] Abualigah L., Diabat A. Advances in Sine Cosine Algorithm: A comprehensive survey. Artificial Intelligence Review. **54**, 2567–2608 (2021).

[15] Mirjalili S. SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems. **96**, 120–133 (2016).

[16] Alandoli E. A., Lee T. S. A critical review of control techniques for flexible and rigid link manipulators. Robotica. **38** (12), 2239–2265 (2020).

[17] Benameur S., Tadrist S., Mellal M. A., Williams E. J. Basic Concepts of Manipulator Robot Control. Design and Control Advances in Robotics. 1–12 (2023).

[18] Adagolodjo Y., Renda F., Duriez C. Coupling Numerical Deformable Models in Global and Reduced Coordinates for the Simulation of the Direct and the Inverse Kinematics of Soft Robots. IEEE Robotics and Automation Letters. **6** (2), 3910–3917 (2021).

[19] Sayed A. S., Mohamed N. A., Salem A. A., Ammar H. H. Modeling of Nonlinear 3-RRR Planar Parallel Manipulator: Kinematics and Dynamics Experimental Analysis. International Journal of Mechanical & Mechatronics Engineering. **20**, 175–185 (2020).

[20] Rocha C. R., Tonetto C. P., Dias A. A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. Robotics and Computer-Integrated Manufacturing. **27** (4), 723–728 (2011).

[21] Gambhire S. J., Kishore D. R., Londhe P. S., Pawar S. N. Review of sliding mode based control techniques for control system applications. International Journal of Dynamics and Control. **9**, 363–378 (2021).

[22] Nawress B., Lakhal A. N. G., Braïek N. B. Neural State and Disturbance Observer-based Sliding Mode Control of a Unicycle Robot. 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET). 1–6 (2023).

[23] Chotikunnan P., Chotikunnan R. Dual design PID controller for robotic manipulator application. Journal of Robotics and Control (JRC). **4** (1), 23–34 (2023).

[24] Spong M. W. An historical perspective on the control of robotic manipulators. Annual Review of Control, Robotics, and Autonomous Systems. **5**, 1–31 (2022).

[25] Wang C., Frazelle C. G., Wagner J. R., Walker I. D. Dynamic Control of Multi-Section Three-Dimensional Continuum Manipulators Based on Virtual Discrete-Jointed Robot Models. IEEE/ASME Transactions on Mechatronics. **26** (2), 777–788 (2021).

[26] Mashkov O. A., Chumakevich V. A., Mamchur Y. V., Kosenko V. R. The method of inverse problems of dynamics for the synthesis of a system of stabilization of the movement of a dynamic object on operatively programmable trajectories. Mathematical Modeling and Computing. **7** (1), 29–38 (2020).

[27] Kada D., Kouidere A., Balatif O., Rachik M. Mathematical modeling of the gaming disorder model with media coverage: optimal control approach. Mathematical Modeling and Computing. **10** (1), 245–260 (2023).

Потужність метаевристичних алгоритмів для робототехніки: сингулярність і траєкторія

Харраде І., Кміч М., Сайюрі М., Чалг З.

Національна школа прикладних наук,
Університет Сіді Мохамед Бен Абделла-Фес,
Лабораторія інженерних систем і застосувань, Фес, Марокко

Під час розрахунку кінематичної моделі будь-якого робота, паралельного чи плоского, часто виникає проблема сингулярності. Запропоновано застосовувати метаевристичні алгоритми для визначення необхідної цілі для вирішення цієї проблеми та мінімізації обчислення. Результати моделювання з використанням декількох метаевристичних алгоритмів (MA) на тій самій популяції були отримані зі скороченим часом обчислення (0.50 с). Ефективність запропонованої техніки для максимізації положення та траєкторії шарнірів у плоскопаралельному роботі–маніпуляторі 3-DOF або 3-RRR (з трьома ступенями свободи обертання) достатньо проілюстрована ними. Синус–косинусний алгоритм (SCA) і певні цільові точки по суті є основою методу, який визначає оптимальний бажаний шлях. Ці результати показують, наскільки добре працює запропонована стратегія для максимізації обчислень, позицій та ідеальної траєкторії робота.

Ключові слова: *3-PPP плоскопаралельний робот-маніпулятор; робот з трьому ступенями свободи; метаевристичні алгоритми; уникання сингулярності; положення шарніра; оптимальна траєкторія; візуальне обслуговування.*