Mechatronics and Automation Technology
J. Xu (Ed.)

doi:10.3233/ATDE241224

Optimal Path-Following Control for Redundant Manipulators: A Multi-Objective Optimization Approach

Ali DEEB[a,1], Ammar AAKEL[b], Vladimir KHOKHLOVSKIY[a], Viacheslav SHKODYREV[a]

[a] *Higher School of Cyberphysical Systems and Control, Institute of Computer Science and Cybersecurity, Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russian Federation*

[b] *ITMO University, Saint Petersburg, Russian Federation*

Abstract. This paper presents an optimal control framework tailored for redundant robotic manipulators, aiming to devise precise joint-space trajectories while minimizing control efforts. The core contribution is the formulation of trajectory planning as a multi-objective optimization problem, tackled through a Genetic Algorithm-based Model Predictive Control strategy, followed by Gradient Descent refinement. Moreover, we develop a strategy to apply the resulted high-level design of the joint-space trajectory into a dynamic time-series control strategy that respects the physical constraints of actuators in motion, speed, acceleration and jerk. Experimental results on a simulation model underscore the framework's efficacy, demonstrating minimal positioning errors without the need for high computational resources for the trajectory design. Moreover, dynamical analysis of the actuators signals for the low-level phase demonstrates the ability of the overall framework to be applied on a real robotic manipulator.

Keywords. Redundant manipulator, inverse kinematics, genetic algorithm, multi-objective optimization, gradient descent.

1. Introduction

In the realm of modern robotics, robotic manipulators stand out as a confluence of mechanical, electrical, and computer engineering disciplines. These tools have become indispensable in diverse sectors, significantly impacting fields such as manufacturing [1], and medicine [2]. Moreover, their role in transforming industrial manufacturing processes is particularly noteworthy, as detailed by Erdős et al. [3].

Redundant robotic manipulators are known for their superior accuracy and precision over their non-redundant counterparts, as Kouabon et al. discuss [4], making them singularities, achieving high dexterity, navigating around obstacles, minimizing torque, and importantly, fault tolerance [5]. Despite these advantages, they pose challenges, including complicated motion planning processes [6] and difficulties in achieving high precision in real-time control, as Ning et al. have observed [7].

[1] Corresponding Author, Ali DEEB, Higher School of Cyberphysical Systems and Control, Institute of Computer Science and Cybersecurity, Peter the Great St. Petersburg Polytechnic University, Saint Petersburg, Russian Federation; Email: dib_a@spbstu.ru.

Inverse kinematics for redundant manipulators involves determining the joint angles required to achieve a desired end effector position and orientation, while also taking into account the redundancy of the system. This problem is considered quite complex in the realm of redundant manipulators where a desired end effector position and orientation can be achieved through a diverse set of joint-space configurations. In the field of inverse kinematics (IK), there has been a variety of research approaches. Pham et al. [8] explored using the bees algorithm for neural network training to solve the IK problem. Particle Swarm Optimization (PSO) has also been widely used, as seen in Huang et al.'s study [9]. Moreover, Adly et al. [10] proposed a Multi-Objective PSO to find the inverse kinematic solution to move between two points considering minimizing two cost functions.

In the context of redundant manipulators, the task of motion planning becomes quite complex due to the additional degrees of freedom. The challenge, therefore, is to find the optimal path that utilizes the extra degrees of freedom to improve performance, while ensuring that the end-effector reaches the desired goal position without colliding with obstacles [11]. Other additional criteria include Torque minimization and kinetic energy minimization [12], a time-optimal motion planning [13], and a trajectory optimization algorithm in joint space depending on cubic polynomial interpolation and quintic polynomial interpolation [14]. Further researches [15] discussed a Recurrent Neural Network (RNN) model for the double-index scheme, which sets the minimum kinetic energy and minimum joint-angle offset as optimization indices. The application of Evolutionary Computation (EC) to robotics constitutes the field of Evolutionary Robotics (ER). EC has been used in robotics in solving different tasks like navigation [16] and predator-pray tasks [17]. Generally speaking, the control challenge can be re-framed as a regression task to deduce the governing control strategy, allowing for the construction of machine-learned controllers via iterative and stochastic methodologies [18]. For manipulators, EML (namely Neuroevolution) was used for solving tasks related to industrial robots where manipulating an object with suitable effectors [19, 20]. In a more concrete context, Momani et al. [21] suggested using Genetic Algorithms (GAs) to solve the inverse kinematics problem of robot manipulators, having it formulated as an optimization problem based on the concept of minimizing the accumulative path deviation in the absence of any obstacles in the workspace. Moreover, [22] suggested using a neural-network committee machine (NNCM) to solve the inverse kinematics of a 6-DOF redundant robotic manipulator. Recently, [23] developed a strategy to solve the inverse kinematics (IK) problem of free-floating space robot (FFSR) using Evolutionary Algorithms and Gradient Descent. Controlling manipulators usually rely on the principle of Model Predictive Control (MPC), to guide the control decision. MPC strategies were vastly discussed in the field of redundant manipulators control, for example Jin et al. [24] used MPC to solve the problem of positioning the end effector of a redundant manipulator with constraints on joints speed and acceleration. Nicolis et al. [25] used sliding mode control with MPC for this task. Moreover, [26] used Recurrent Neural Network (RNN) to achieve trajectory tracking for redundant manipulators, benefiting from the principle of Receding Horizon Control.

Conversely, while our focus is on solving inverse kinematics to position the end effector, our goal is to optimize the joint-space trajectory over a finite horizon of trajectory points at each step, ensuring minimal accumulated cost. To address this, a genetic algorithm is employed to solve the inverse kinematics problem using Multi-Objective Optimization (MOO), over a horizon as considered in previous work [27, 28].

With the methodology applied to a 7-DOF manipulator called Franka Emika, the goal is to achieve these trajectories with minimal cost defined by three objective functions.

The contributions of this study are summarized as follows:

- Introduction of an Optimal Control framework designed to compute the joint space trajectory for redundant manipulators, focusing on minimizing the overall cost of movements and speed.
- Formulation of the optimal control challenge as a Multi-Objective Optimization (MOO) problem, enabling a comprehensive approach to trajectory planning with policy tradeoffs.
- Development of a solution methodology for the MOO problem through the implementation of a Genetic Algorithm-based Model Predictive Control (GA-MPC) strategy, specifically through Non-dominated Sorting Genetic Algorithm (NSGAII).
- Introduction of a refinement stage employing Gradient Descent to further optimize the solution, enhancing accuracy and reducing positioning errors.

The remainder of this paper is structured as follows: Section 2 details the problem formulation, delving into the optimal control problem and the development of the framework via the proposed strategies. Section 3 presents the experiments conducted and the results obtained. Finally, Section 4 provides the concluding remarks.

2. Problem Formulation

We consider a 3D trajectory T for the manipulator's end-effector, aiming to minimize a cost functional J:

$$J(\boldsymbol{U}) = \sum_{n=1}^{M}(E(\Theta_n)^T Q E(\Theta_n) + U_n^T R U_n) \tag{1}$$

where $\boldsymbol{U} = (U_1, \dots, U_M) \in \mathbb{R}^N \times \mathbb{R}^M$ represents the sequence of control inputs guiding the joint angles Θ_n, $Q \in \mathbb{R}^3 \times \mathbb{R}^3$ and $R \in \mathbb{R}^N \times \mathbb{R}^N$ are diagonal matrices, and $E(\Theta_n) = F(\Theta_N) - P_n \in \mathbb{R}^3$ denotes the Euclidean error between the forward kinematics $\mathrm{F}(\Theta_\mathrm{n})$ and the desired position P_n. The joint angles evolve according to:

$$\Theta_{n+1} = \Theta_n + U_n, \tag{2}$$

for $n \in \mathbb{N}_{M-1}$. The objective is to minimize both the tracking error and the control effort (cost).

We introduce an iterative control strategy that minimizes the following cost over a horizon H:

$$J_n(\Theta_n, \boldsymbol{U_{n,H}}) = \sum_{i=n+1}^{n+H}(E(\Theta_i)^T Q E(\Theta_i) + U_{i-1}^T R U_{i-1}), \tag{3}$$

where $\boldsymbol{U_{n,H}} = (U_n, \dots, U_{n+H-1})$. We have by definition: $\Theta_i = \Theta_n + \sum_{k=n}^{i-1} U_k$ for $i \in n+1, \dots, n+H$, therefore for small deviations $\epsilon_i = \Theta_i - \Theta_n$, we approximate $F(\Theta_n + \epsilon_i)$ using a Taylor expansion around Θ_n:

$$F(\Theta_n + \epsilon_i) \approx F(\Theta_n) + \mathfrak{J}(\Theta_n)\epsilon_i, \tag{4}$$

where $\mathfrak{J}(\Theta_n)$ is the Jacobian matrix of F at Θ_n. Neglecting the higher-order terms due to small ϵ_i, the error becomes:

$$E(\Theta_i) \approx F(\Theta) - P_i + J(\Theta_n)\epsilon_i. \tag{5}$$

However, assuming ϵ_i is small and $F(\Theta_n)$ varies slowly, we simplify:

$$E(\Theta_i) \approx F(\Theta_n) - P_i. \tag{6}$$

Following up, we derive four objective functions:

1. The Euclidean error component: $G_{1,n} = \sum_{i=n+1}^{n+H} \gamma^{i-n-1} \left|\left|F(\Theta_n) - P_i\right|\right|^2$, where $\gamma \in (0,1]$ is a damping factor.

2. The control effort component: $G_{2,n} = \left|\left|U_n\right|\right|_2^2$.

3. A time-minimization component: $G_{3,n} = \left|\left|U_n\right|\right|_\infty = \max_{1\le i\le N} \left|u_{n,i}\right|$, where $u_{n,i}$ is the i^{th} element of U_n.

4. Additionally, to ensure the end-effector maintains the desired orientation (perpendicular to a plane), we define an orientation error component: $G_{4,n} = \left|\boldsymbol{v}^T R_{0,7}(\Theta_n)\boldsymbol{v} - r_{3,3}^d\right|$, where $\boldsymbol{v} = [0,0,1]^T$ is the desired orientation vector, $R_{0,7}(\Theta_n)$ is the rotation matrix from base to end-effector frame, and $r_{3,3}^d$ is the desired element of the rotation matrix.

Our Multi-Objective Optimization problem minimizes $G_{1,n}, G_{2,n}, G_{3,n}$, and $G_{4,n}$ simultaneously. We employ the NSGA-II algorithm to approximate the Pareto front and select a solution that balances these objectives. After the Genetic Algorithm search, we refine the solution using Gradient Descent to minimize $G_{1,n}$. The GD update rule is $\Theta_{n+1} = \Theta_n - \eta\nabla_\Theta G_{1,n}(\Theta_n)$, where $\eta > 0$ is the learning rate, and $\eta\nabla_\Theta G_{1,n}(\Theta_n)$ is the gradient of $G_{1,n}$ with respect to Θ_n.

To ensure smooth transitions between joint positions, we interpolate the joint-space trajectory using Cubic Hermite splines. For each joint j, the spline between times t_{k-1} and t_k is defined by:

$$\theta_j = h_{00}(\tau)\theta_j^{k-1} + h_{10}(\tau)\dot{\theta}_j^{k-1}T_s^k + h_{01}(\tau)\theta_j^k + h_{11}(\tau)\dot{\theta}_j^k T_s^k, \tag{7}$$

where $\tau = \frac{t-t_{k-1}}{T_s^k} \in [0,1]$, $T_s^k = t_k - t_{k-1}$, θ_j^k and $\dot{\theta}_j^k$ are the angular position and velocity at time t_k, and finally the Hermit basis functions are: $h_{00} = 2\tau^3 - 3\tau^2 + 1$, $h_{10} = \tau^3 - 2\tau^2 + \tau, h_{01} = -2\tau^3 + 3\tau^2, h_{11} = \tau^3 - \tau^2$. We set $\dot{\theta}_k^k = 0$ at all points to assure smoothness. We determine T_s^k by solving:

$$min\, T_s^k \text{ subject to: } \begin{cases} \max_{t\in[t_{k-1},t_k]} \left|\dot{\theta}_j(t)\right| \le \bar{\omega}_j, \\ \max_{t\in[t_{k-1},t_k]} \left|\ddot{\theta}_{j(t)}\right| \le \bar{\alpha}_j, \\ \max_{t\in[t_{k-1},t_k]} \left|\theta_{j(t)}^{(3)}\right| \le \bar{J}_j, \end{cases} \tag{8}$$

for each joint j, where the right-handed terms are the maximum allowed velocity, acceleration and jerk, respectively.

We use a binary search algorithm to find the minimal T_s^k satisfying the constraints. The algorithm initializes T_{low} and T_{high} based on feasible ranges and iteratively narrows down the interval until convergence.

2.1. Basic Description of a Redundant Manipulator

The first step in this problem is to find the forward kinematics of the robot. To obtain the forward kinematics, one can begin by defining the robot's kinematic structure, including joint variables, link lengths, joint angles, and their corresponding transformation matrices using the Denavit-Hartenberg (D-H) parameters. D-H transformation matrix between two consecutive links i and $i + 1$ is given by the form:

$$T_{i,i+1}(\theta) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Subsequently, the transformation matrices for each joint can be derived, and these can be multiplied to obtain the transformation matrix that maps the joint variables to the end-effector position and orientation. Finally, the position and orientation of the end-effector can be extracted from the resulting transformation matrix. This process allows for the formal derivation of the forward kinematics for a robotic system.

$$T_{0,n}(\theta) = \prod_{i=0}^{n} T_{i,i-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{0,n} & P_{0,n} \\ 0 & 1 \end{bmatrix}. \quad (10)$$

Here, $T_{0,n}(\theta)$ represents the transformation matrix from the base to the end-effector of the manipulator. The matrix consists of the rotational part $R_{0,n}(\theta)$, which indicates the orientation of the end-effector, and the transitional part $P_{0,n}(\theta)$, representing the position of the end-effector in Cartesian coordinates. The elements rij are the components of the rotation matrix, and p_x, p_y, p_z are the coordinates of the position vector. The redundant structure of this degree of freedom makes practically unfeasible to develop an analytic solution for the inverse kinematics. The inverse kinematics problem is usually more complex for redundant robots. Traditionally, three models are used to solve the inverse kinematics problem: geometric, algebraic, and iterative models [29]. Each method has some disadvantages for solving the inverse kinematics problem.

3. Experiments and Results

We chose to test our proposed strategies on a 7-DoF redundant manipulator called Franka Emika Panda [30] introduced in 2017, which is a 7-DoF redundant manipulator, characterized by its seven rotating joints and a humanoid arm structure. The D-H parameters for this manipulator are specified as in table 1:

Table 1. D-H Parameters

i	a_i (mm)	θ_i	α_i	d_i (mm)
1	0	θ_1	0	333
2	0	θ_2	$-\pi/2$	0
3	0	θ_3	$\pi/2$	316
4	82.5	θ_4	$\pi/2$	0
5	-82.5	θ_5	$-\pi/2$	384
6	0	θ_6	$\pi/2$	0
7	88	θ_7	$\pi/2$	0
8	0	θ_8	0	107

The research entailed a series of tests utilizing a mathematical simulation model, which was developed on a computational system equipped with an Intel i5-12 CPU and 32 GB of RAM. The construction of the model class and the formulation of the control strategy were carried out in the Python programming language, employing the Pymoo library for multi-objective optimization [31] and PyTorch for gradient descent algorithms. Specific hyperparameters selected for the experimental procedures included a damping factor (γ) set to 0.9 and a finite horizon of H = 20. In the context of the Genetic Algorithm (GA), the population size (G_z) was determined to be 100, with 40 offspring generated per iteration (G_f), over a span of 600 generations (G_n). The gradient descent was conducted over 103 iterations with a learning rate of 3×10^{-7}. The computational model successfully calculated solutions with an average processing time of 0.1 seconds per step. It is noteworthy that these computations were performed without the aid of a Graphics Processing Unit (GPU), highlighting the model's suitability for execution on less powerful computing systems. Despite the absence of real-time processing requirements for this application, the primary goal was to design a joint-space trajectory for the robotic arm and to store this trajectory as a sequence of joint values. The aim of the experiment was to direct a robotic arm to trace a path defined by an Archimedean spiral on a horizontal plane, with a constant depth (z) coordinate. The spiral's radius was designed to increase linearly with the angular distance from the origin, following the equation

$$r(\theta) = a + b \tag{11}$$

where a and b are constants set to 0 and 2, respectively, and θ varies from 0 to 4π. The path's horizontal (x) and vertical (y) coordinates were derived from the expressions

$$x(\theta) = \alpha r(\theta)\cos(\theta) + x_0, \tag{12}$$
$$y(\theta) = \alpha r(\theta)\sin(\theta) + y_0, \tag{13}$$

where θ is iteratively calculated as the following:

$$\theta_{i+1} = \theta_i + \frac{\Delta s}{a + b\theta_i}. \tag{14}$$

This framework iterates to populate a sequence of θ values that aim to maintain a constant Δs, and subsequently calculates corresponding r, x, and y values for plotting the spiral, with $\alpha, x_0, y_0, \Delta s$ having the values of $10, 0, 300, 0.5$ respectively. The z coordinate was fixed at 200 mm, implying that the robotic arm's movement was confined to a horizontal plane. Additionally, the orientation of the arm's end-effector was kept perpendicular to this plane to maintain a vertical posture throughout the execution of the task.

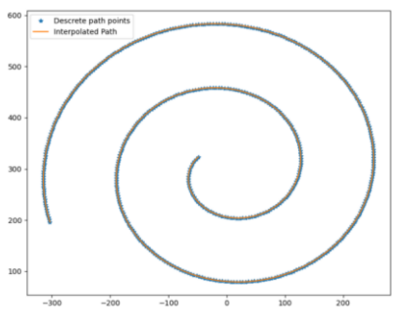

Figure 1. Suggested operational-space trajectory

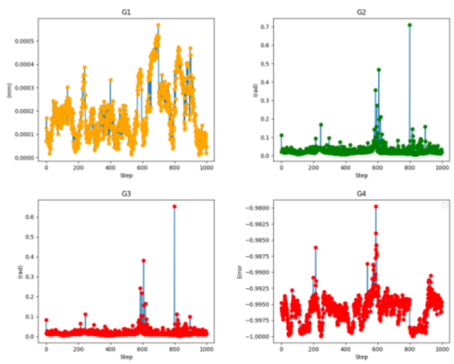

Figure 2. Results of applying our Multi-Objective control strategy

Figure 1 shows the discrete path points of the Archimedean spiral, and how the robot traverses this trajectory with minimal error, and figure 2 displays the outcomes of implementing our multi-objective control approach, highlighting the values of the four proposed objective functions with respect to the path points. A prominent observation is the minimal positioning error, which suggests the proposed solution's viability for practical application. Additionally, the cost metrics, depicted in the two functions, are maintained at a low level throughout.

The algorithm successfully identified solutions that achieved a low Euclidean error with the terminal joint's orientation remaining close to the desired perpendicular position, and with minimal transition costs. These performance measures are denoted by the functions (G_1, G_2, G_3, G_4). Specifically, the Euclidean error was maintained below 2×10^{-4}mm, signifying high precision in positioning. The algorithm effectively navigated the solution space to find low-cost paths, taking advantage of the robot's redundancy to move between consecutive points along the trajectory efficiently.

4. Conclusion

This study successfully developed and validated an innovative optimal control framework for redundant robotic manipulators. By formulating the trajectory planning as a MOO problem, the research provides a nuanced approach that balances precision in following a desired path with the practicality of control efforts. The hybrid GA-MPC strategy, complemented by Gradient Descent refinement, offers a robust solution methodology that outperforms traditional methods in terms of accuracy and efficiency. The results of this study pave the way for advanced control strategies in robotic systems, with potential applications in complex, real-world environments where redundancy and precision are critical.

Acknowledgments

The work is sponsored by the Russian Science Foundation, Grant 23-29-0051 of Jan.13, 2023.

References

[1] T. S. Hsia, T. A. Lasky, Z. Guo. Robust independent joint controller design for industrial robot manipulators[J]. IEEE transactions on industrial electronics, 1991, 38 (1): 21–25.

[2] F. Alambeigi, S. Sefati, M. Armand, A convex optimization framework for constrained concurrent motion control of a hybrid redundant surgical system[C]. 2018 Annual American Control Conference (ACC), IEEE, 2018, pp. 1158–1165.

[3] G. Erdős, A. Kovács, J. Váncza. Optimized joint motion planning for redundant industrial robots[J]. CIRP Annals, 2016, 65 (1): 451–454.

[4] A. J. Kouabon, A. Melingui, J. M. Ahanda, O. Lakhal, V. Coelen, M. Kom, R. Merzouki. A learning framework to inverse kinematics of high dof redundant manipulators[J]. Mechanism and Machine Theory, 2020, 153: 103978.

[5] R. G. Roberts, A. A. Maciejewski. A local measure of fault tolerance for kinematically redundant manipulators[J]. IEEE Transactions on Robotics and Automation, 1996, 12 (4): 543–552.

[6] S. Yahya, M. Moghavvemi, H. A. Mohamed, Redundant manipulators kinematics inversion[J]. Scientific Research and Essays, 2011, 6 (26): 5462–5470.

[7] Ning Y., Li T., Du W., Yao C., Zhang Y., Shao J. Inverse kinematics and planning/control codesign method of redundant manipulator for precision operation: Design and experiments[J]. Robotics and Computer-Integrated Manufacturing 80 (2023) 102457.

[8] D. Pham, M. Castellani, A. Fahmy, Learning the inverse kinematics of a robot manipulator using the bees algorithm[C]. 2008 6th IEEE international conference on industrial informatics, IEEE, 2008, pp. 493–498.

[9] Huang H.-C., Chen C.-P., Wang P.-R. Particle swarm optimization for solving the inverse kinematics of 7-dof robotic manipulators[C]. 2012 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, 2012, pp. 3105–3110.

[10] M. A. Adly, S. K. Abd-El-Hafiz, Inverse kinematics using single-and multi-objective particle swarm optimization[C]. 2016 28th International Conference on Microelectronics (ICM), IEEE, 2016, pp. 269–272.

[11] A. A. Maciejewski, C. A. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments[J]. The international journal of robotics research 4 (3) (1985) 109–117.

[12] J. Woolfrey, Lu W., Liu D. A control method for joint torque minimization of redundant manipulators handling large external forces[J]. Journal of Intelligent & Robotic Systems 96 (2019) 3–16.

[13] L Lu., Zhang J., J. Fu Y. H., Han J., Wang H. Time-optimal tool motion planning with tooltip kinematic constraints for robotic machining of sculptured surfaces[J]. Robotics and Computer Integrated Manufacturing, 2020, 65: 101969.

[14] S. Luyang, P. Yichao. Study on trajectory optimization algorithm of industrial robot in joint space[C]. Journal of Physics: Conference Series, Vol. 1676, IOP Publishing, 2020, p. 012206.

[15] He L., Liu M., Yi C., Shi Y., Sun Z., Chen,L. Jin L. Recurrent neural network for kinematic control of redundant robot manipulators[C]. 2020 39th Chinese Control Conference (CCC), IEEE, 2020, pp. 345–3850.

[16] S. Doncieux, N. Bredeche, J.-B. Mouret, A. E. Eiben, Evolutionary robotics: what, why, and where to[C]. Frontiers in Robotics and AI,2015, 2 .

[17] S. Nolfi, J. Bongard, P. Husbands, D. Floreano, Evolutionary robotics, 2015, chapter 76.

[18] C. Cáceres, J. M. Rosario, D. Amaya. Approach of kinematic control for a nonholonomic wheeled robot using artificial neural networks and genetic algorithms[C]. 2017 international conference and workshop on bioinspired intelligence (iwobi), IEEE, 2017, pp. 1–6.

[19] G. Y. Cornejo Maceda, B. R. Noack, Evolutionary machine learning in control. Handbook of Evolutionary Machine Learning, Springer, 2023, pp. 629–656.

[20] Hu Y., Wu X., Geng P., Li Z. Evolution strategies learning with variable impedance control for grasping under uncertainty[J]. IEEE Transactions on Industrial electronics, 2018, 66 (10): 7788–7799.

[21] S. Momani, Z. S. Abo-Hammour, O. M. Alsmadi, Solution of inverse kinematics problem using genetic algorithms[J]. Applied Mathematics & Information Sciences, 2016, 10 (1): 225.

[22] R. Köker, T. Çakar, Y. Sari. A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators[J]. Engineering with Computers, 2014, 30: 641–649.

[23] Zhang H., Tang Y., Zhu Z. Inverse kinematics solver based on evolutionary algorithm and gradient descent for free-floating space robot[C]. International Conference on Intelligent Robotics and Applications, Springer, 2023, pp. 520–532.

[24] Jin L., Zhang F., Liu M., Xu S. D., Finite-time model predictive tracking control of position and orientation for redundant manipulators[J]. IEEE Transactions on Industrial Electronics2022, 70 (6): 6017–6026.

[25] D. Nicolis, F. Allevi, P. Rocco. Operational space model predictive sliding mode control for redundant manipulators[J]. IEEE Transactions on Robotics, 2020, 36 (4): 1348–1355.

[26] Yan J., Jin L., Yuan Z., Liu Z. Rnn for receding horizon control of redundant robot manipulators[J]. IEEE Transactions on Industrial Electronics, 2021, 69 (2): 1608–1619.

[27] Liang H B, et al. An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system[J]. Mechanical Systems and Signal Processing, 2020, 142: 106708.

[28] A. Deeb, V. N. Khokhlovskiy, V. P. Shkodyrev. Model predictive control and genetic algorithms for optimization of continuous stirred tank reactors[C]. Smart Electromechanical Systems: Mathematical and Software Engineering, Springer, 2024, pp. 185–191.

[29] Löw, Tobias, and Sylvain Calinon. Geometric algebra for optimal control with applications in manipulation tasks[J]. IEEE Transactions on Robotics, 2023, 39(5): 3586-3600.

[30] Franka Robotics GmbH, FRANKA RESEARCH 3 - The platform of choice for cutting edge AI & Robotics research Accessed: 2024-03-04.

[31] J. Blank, K. Deb. Pymoo: Multi-objective optimization in python[J]. Ieee Access, 2020, 8: 89497–89509.