Inverse Kinematics and an Optimized Path Planning for a 6-Degree-of-Freedom Robot Manipulator

Takumu Okazaki[1], Akira Terui[1] and Masahiko Mikawa[2]
[1] Institute of Pure and Applied Sciences
[2] Institute of Library, Information and Media Science
University of Tsukuba
Tsukuba, Japan, 305-8577
terui@math.tsukuba.ac.jp and mikawa@slis.tsukuba.ac.jp

1 Introduction

Motion planning for 6-degree-of-freedom (6-DOF) robot manipulators requires efficient inverse kinematics (IK) computation and appropriate selection of joint configurations subject to kinematic constraints. In practical tasks, both the position and orientation of the end-effector must be specified, making motion planning for 6-DOF manipulators more challenging than for lower-dimensional cases.

A central difficulty lies in solving the IK problem efficiently. Classical algebraic methods for 6-DOF manipulators often lead to large polynomial systems, and elimination may become computationally intractable. A well-known approach due to Pieper [7] exploits the case where the rotational axes of three consecutive joints intersect at a single point, allowing decomposition into position and orientation subproblems. However, this structural assumption is restrictive and cannot be applied to a wider class of manipulators.

To address this limitation, in our work [5], we extend Pieper's framework to manipulators where only *two* consecutive joint axes intersect, and apply this formulation to the 6-DOF manipulator myCobot 280. Moreover, by using the Comprehensive Gröbner System (CGS) [9], we treat joint variables (such as the position and orientation of the end-effector) as parameters to reduce repeated Gröbner basis computations and obtain IK solutions efficiently.

Next, we tackle the trajectory planning problem. The trajectory planning problem is an extension of the IK problem in which the input is expanded from a single position to an entire trajectory on the given path. In trajectory planning, multiple IK solutions may exist at each via-point, requiring the selection of a sequence of joint configurations that ensures smooth motion; this is known as the path optimization problem. Previous CGS-based studies [8] focused on simplified settings, such as 3-DOF formulations or position-only constraints, and did not fully address 6-DOF planning with orientation.

In our work [5, 6], we extend this framework to trajectory planning and path optimization under fixed end-effector orientation. We derive the feasible region, generate candidate joint trajectories, and formulate path optimization as a graph search problem. By applying Dijkstra's algorithm with multiple criteria, we obtain a unified algebraic framework for IK and trajectory optimization in 6-DOF manipulators.

2 Inverse Kinematics for myCobot 280

This section describes the inverse kinematic (IK) problem for the 6-degree-of-freedom (6-DOF) robot manipulator myCobot 280, focusing on efficient and systematic computation of joint configurations corresponding to a given end-effector position and orientation. The manipulator consists of six revolute joints arranged in series, and its kinematic behavior is modeled using the Denavit-Hartenberg (D–H) convention [1]. This representation allows the transformation from joint space to configuration space to be expressed as a sequence of transformation matrices, which are functions of the joint angles.

In the D–H convention, the joints are numbered sequentially from the root as Joint $1, 2, \ldots, 6$, and a coordinate system Σ_i is defined for each joint. The rotation axis of Joint i is defined as the z-axis of Σ_i, denoted as ${}^{i}z$. Let ${}^{i-1}l_i$ be defined as the common normal to the ${}^{i-1}z$-axis and the ${}^{i}z$-axis. The origin $\mathcal{O}_i$ of Σ_i is defined as the intersection of the ${}^{i}z$-axis and ${}^{i-1}l_i$, and d_i is the distance from $\mathcal{O}_i$ to ${}^{i}l_{i+1}$. Let θ_i represent the angle of rotation of Joint i about the ${}^{i}z$-axis.

The IK problem is formulated as a system of polynomial equations derived from geometric constraints between the joint parameters and the desired position and orientation of the end-effector. Especially, the trigonometric functions of the joint angles are expressed in variables such as $s_i = \sin\theta_i$ and $c_i = \cos\theta_i$, leading to a system of polynomial equations in these variables. Direct algebraic approaches to such systems often result in high computational complexity, making them impractical for repeated use, particularly in trajectory planning. To address this issue, the proposed method exploits the structural property that the rotational axes of two consecutive joints intersect. By introducing the intersection point of these axes as ${}^{t}(x, y, z)$, the IK problem is decomposed into a more tractable system of polynomial equations involving trigonometric functions of the joint angles.

Let ${}^{t}(l_1, l_2, l_3)$, ${}^{t}(m_1, m_2, m_3)$, and ${}^{t}(n_1, n_2, n_3)$ be the unit vectors of the x, y, and z axes of the end-effector orientation, respectively, and let ${}^{t}(p_1, p_2, p_3)$ be the position of the end-effector. Then, the formulation yields a system that determines the sine and cosine values of the joint angles as

$$
\begin{aligned}
&F_1 = 7318s_6 - 10^2(l_1(-p_1 + x) + l_2(-p_2 + y) + l_3(-p_3 + z)),\\
&F_2 = 7318c_6 - 10^2(m_1(-p_1 + x) + m_2(-p_2 + y) + m_3(-p_3 + z)),\\
&F_3 = -n_3s_5 + c_5(l_3c_6 - m_3s_6), \qquad F_4 = s_5^2 + c_5^2 - 1,\\
&F_5 = s_1 + n_1s_5 - c_5(l_1c_6 - m_1s_6), \quad F_6 = c_1 - n_2s_5 + c_5(l_2c_6 - m_2s_6),\\
&F_7 = s_3^2 + c_3^2 - 1,\\
&F_8 = 10^4x^2 + 10^4y^2 + (100z - 13156)^2 - 255799044 - 211968000c_3,\\
&F_9 = 13156 + 11040c_2 + 9600(c_2c_3 - s_2s_3) - 100z,\\
&F_{10} = s_2^2 + c_2^2 - 1, \qquad F_{11} = s_4^2 + c_4^2 - 1,\\
&F_{12} = (c_2c_3 - s_2s_3)c_4 - (s_2c_3 + c_2s_3)s_4 + m_3c_6 + l_3s_6,
\end{aligned} \tag{1}
$$

along with an additional polynomial system

$$
\begin{aligned}
&n_1(p_1 - x) + n_2(p_2 - y) + n_3(p_3 - z) = d_6, \quad (p_1 - x)^2 + (p_2 - y)^2 + (p_3 - z)^2 = d_5^2 + d_6^2,\\
&((n_1n_2x + (1 - n_1^2)y)(p_1 - x) - (n_1n_2y + (1 - n_2^2)x)(p_2 - y) - n_3(n_1y - n_2x)(p_3 - z))^2\\
&\qquad = d_4^2(d_5^2n_3^2 + (n_2(p_1 - x) - n_1(p_2 - y))^2),
\end{aligned} \tag{2}
$$

that characterize the auxiliary intersection point ${}^t(x, y, z)$ in terms of the end-effector position and orientation. These equations are solved using Gröbner basis techniques. Furthermore, instead of recomputing the Gröbner basis for each input configuration, we employ the Comprehensive Gröbner System (CGS), which treats the position and orientation of the end-effector as parameters. By precomputing the CGS, the corresponding Gröbner basis can be efficiently retrieved for each specific input.

The resulting algorithm proceeds as follows. First, the CGS for the Eq. (2) is precomputed. Given a specific position and orientation of the end-effector, the appropriate CGS branch is selected, and the corresponding Gröbner basis is used to determine the auxiliary point. This value is then substituted into Eq. (1) to compute all joint angles. The method demonstrates the practical applicability of computer algebra techniques to real-world robotic systems, as we show in the result of experiments [5].

3 Trajectory Planning and Path Optimization

This section presents a unified framework for trajectory planning and path optimization of myCobot 280, based on the inverse kinematics (IK) solutions described in the above section. The objective is to generate feasible and efficient motion of the end-effector along a prescribed path while satisfying kinematic constraints and optimizing joint movements.

Suppose that the path that the end-effector must follow is given by a series of consecutive line segments. Each segment is further discretized into a series of via-points, where inverse kinematics solutions are computed using a previously established algebraic method. Two trajectory generation strategies are considered. The first assumes uniform motion along the path, where the end-effector moves at a constant speed through equally spaced via-points. The second generates smooth motion by introducing a time-dependent parameterization that ensures continuous velocity and acceleration profiles. This is achieved using a fifth-degree polynomial function satisfying boundary conditions for smooth start and stop behavior [4].

A key requirement in trajectory planning is that all via-points lie within the feasible region of the manipulator, which depends on the fixed orientation of the end-effector. The feasible region is derived from the kinematic constraints, and only trajectories contained within this region are considered valid. Experimental results demonstrate that the proposed trajectory generation methods can produce feasible paths that closely follow the desired trajectory while maintaining the specified orientation.

Next, the path optimization problem is formulated as a directed graph, where nodes correspond to IK solutions at each via-point and edges represent transitions between consecutive configurations. The cost associated with each edge reflects the motion effort between configurations.

To evaluate motion quality, several cost functions are introduced, including the total joint displacement, the maximum joint displacement, and the standard deviation of joint movements. These functions capture different criteria, such as minimizing overall motion, avoiding excessive load on individual joints, and ensuring balanced movement across joints. Additionally, a manipulability measure [10] based on the Jacobian matrix is incorporated to favor configurations that maintain good kinematic performance. By combining these criteria using weighted sums, the framework enables flexible adaptation to different task requirements.

The optimal trajectory is obtained by applying Dijkstra's algorithm [2] to the constructed graph, identifying the sequence of joint configurations that minimizes the selected cost function. The results of experiments [6] show that the proposed method effectively generates smooth and efficient motion, and that the computation cost scales with the number of via-points and candidate solutions. The framework provides a practical and extensible approach to motion planning that integrates algebraic IK solving, trajectory generation, and path optimization in a consistent manner.

References

[1] Denavit, J., Hartenberg, R.: A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. Journal of Applied Mechanics **22**, 215–221 (1955). `https://doi.org/10.1115/1.4011045`

[2] Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959). `https://doi.org/10.1007/BF01386390`

[3] Elephant Robotics Co., Ltd.: myCobot 280 M5 (2023), `https://www.elephantrobotics.com/en/mycobot-280-m5-2023-en`, accessed 2026-06-12.

[4] Lynch, K.M., Park, F.C.: Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press (2017). `https://doi.org/10.1017/9781316661239`

[5] Okazaki, T., Terui, A., Mikawa, M.: Inverse Kinematics for a 6-Degree-of-Freedom Robot Manipulator Using Comprehensive Gröbner Systems. In: Computer Algebra in Scientific Computing, CASC 2025. Lecture Notes in Computer Science 16235, Springer (2025), pp. 336–359. `https://doi.org/10.48550/arXiv.2509.00823`

[6] Okazaki, T., Terui, A., Mikawa, M.: An Effective Trajectory Planning and an Optimized Path Planning for a 6-Degree-of-Freedom Robot Manipulator. In: Computer Algebra in Scientific Computing, CASC 2025. Lecture Notes in Computer Science 16235, Springer (2025), pp. 312–335. `https://doi.org/10.48550/arXiv.2509.00828`

[7] Pieper, D.L.: The Kinematics of Manipulators under Computer Control. Ph.D. Thesis, Stanford University (October 1968), `https://apps.dtic.mil/sti/citations/AD0680036`.

[8] Shirato, Y., Oka, N., Terui, A., Mikawa, M.: An optimized path planning of manipulator with spline curves using real quantifier elimination based on comprehensive gröbner systems. In: Proceedings of the SCSS 2024 Work in Progress Workshop. pp. 105–112. Sun SITE Central Europe, RWTH Aachen University (2024). `https://dblp.org/rec/conf/sycss/ShiratoOTM24.html`, accessed 2026-06-12.

[9] Suzuki, A., Sato, Y.: A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In: Proc. ISSAC 2006. pp. 326–331 (2006). `https://doi.org/10.1145/1145768.1145821`

[10] Yoshikawa, T.: Manipulability of robotic mechanisms. The International Journal of Robotics Research **4**(2), 3–9 (1985). `https://doi.org/10.1177/027836498500400201`