JST: Smart Systems and Devices
Volume 35, Issue 3, September 2025, 053-060

Kinematics and Dynamics Analysis of a 7-Degree-of-Freedom Collaborative Robot with Joint Limitation Avoidance

Phuong Thao Thai*, Hoang Nguyen Quang, Phong Dinh Van
School of Mechanical Engineering, Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: thao.thaiphuong@hust.edu.vn*

Abstract

These days, collaborative robots (Cobot) are well-known for their adaptability and several uses. With a 7-degree-of- freedom cobot model, the work emphasises more the flexibility as a benefit to overcome the joint constraints usually faced in robotic systems. Applying the Jacobian matrix approach with its null-space helps one to consider the kinematics and dynamics issue in order to prevent joint limitation. Whereas orientation is stated using Roll-Pitch-Yaw angles, the inverse kinematics problem involves parameterising the robot's end-effector by its position in Cartesian coordinates. By means of the null space of the Jacobian matrix, one can escape joint constraints in robotic mobility. Emphasising great accuracy in computations with two path planning: rectilinear and curved path, this research offers a thorough study of the trajectory tracking control dynamics. Different modules and simulation results of various challenges have been numerically implemented in MATLAB-Simulink and the ROS environment to show the efficiency of the suggested method after analysis of the computations.

Keywords: Collaborative robot, dynamics, inverse kinematics, joint limit avoidance

1. Introduction

Designed with capabilities as human hand, collaborative robots—also known as cobots—can be employed extensively with mass production. Robotic manufacturing is evident daily in many different contexts nowadays. Cobots are well-known for their autonomous functioning, safety elements, and rapid deployment adaptability. Unlike other industrial robots, cobots have the main benefits in terms of lightweight, small size and their capacity to perform several tasks without changes [1].

Some cobot arms include 7 degrees of freedom (DOF) with a human arm structure to improve dexterity during operation. This structure lets us obtain analytical solutions of the inverse kinematic issue by means of position and orientation decoupling [2] without offsets at shoulder and wrist. But since it's hard to distribute three axes intersecting at one place, this structure questions the mechanical design. Design and construction of a manipulator with offsets at shoulder and wrist is more simple conversely. Still, the structure with offsets causes inverse kinematics problems to become more challenging. For several studies, inverse kinematics problem for redundant robots has been a fascinating subject [3 - 5]. Analytical approaches using position-orientation decoupling, numerical techniques including the Newton-Raphson method, Jacobian transpose methods [3, 5], pseudoinverse matrix methods, and optimization-based constrained solutions are among the several well-known approaches to solve inverse kinematics. Especially, the analytical position-orientation decoupling technique is relevant just for some configurations [6, 7].

Jacobian based approach offers various benefits using redundant robot. Directly the joint velocities and Cartesian velocities are solved using inverse kinematics with velocity level. Moreover, by use of the damped least squares pseudoinverse and the null space of the Jacobian matrix [8 - 12], obstruction, joint limit limitations, and singularity can be essentially avoided.

This paper introduces a fully integrated and simulation-validated solution for a redundant 7-DOF collaborative robot with non-ideal geometry, where joint limit avoidance is achieved via Jacobian null-space projection. During task performance, this approach ensures the safe operational ranges for arm's movements and joint angles. To show the success of the suggested approach, numerical simulations are carried out with high accuracy. While the differential equations of motion are calculated using the Lagrange method to enable trajectory tracking control based on inverse dynamics, hence testing the accuracy of the method, the end-effector orientation is defined in these simulations using Roll-Pitch-Yaw angles. MATLAB generates the simulation results; they are then evaluated on the Robot Operating System (ROS) platform, therefore guaranteeing practical applicability and robustness.

ISSN 2734-9373
https://doi.org/10.51316/jst.184.ssad.2025.35.3.7
Received: Mar 24, 2025; revised: Jun 20, 2025;
accepted: Jun 25, 2025

2. Kinematics Problem

2.1. Forward Kinematics

The 7-DOF collaborative robot model is shown in Fig. 1. The forward kinematics solution is derived using the Denavit-Hartenberg (DH) method. The coordinate frames are assigned to the robot links following the DH convention, and the corresponding DH parameters are illustrated in Fig. 1 and listed in Table 1, the upper and lower limits of each joint of the cobot are listed in Table 2. Here, q_i $(i = 1,2,\dots,7)$ represent the joint variables.

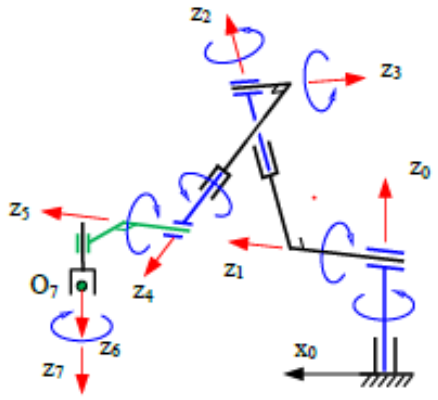

Fig. 1. Model of 7-DOF cobot

Table 1. D-H table of 7-DOF cobot

link i	θ (rad)	d (m)	a (m)	α (rad)
1	q_1	d_1	0	$\pi/2$
2	q_2	d_2	0	$\pi/2$
3	q_3	d_3	0	$\pi/2$
4	q_4	d_4	0	$\pi/2$
5	q_5	d_5	0	$\pi/2$
6	q_6	d_6	a_6	$\pi/2$
7	q_7	d_7	0	0

The homogeneous transformation matrix $\boldsymbol{A}_i^{i-1}(q_i)$ for each joint is derived from the DH parameters in Table 1 in the following form:

$$\mathbf{A}_i^{i-1}(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Table 2. The upper and lower limits of each joint of 7-DOF cobot

Joint	Lower limit (°)	Upper limit (°)	Range (°)
1	175	−175	350
2	355	5	350
3	175	−175	350
4	355	5	350
5	175	−175	350
6	355	5	350
7	175	−175	350

Position and orientation of link k is determined as following:

$$\mathbf{T}_k^0(\mathbf{q}) = \mathbf{A}_1^0(q_1)\mathbf{A}_2^1(q_2)\dots\mathbf{A}_k^{k-1}(q_k) = \begin{bmatrix} \mathbf{R}_k^0(\mathbf{q}) & \mathbf{r}_{O_k}^{(0)}(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix}, k = 1,2,\dots,7 \quad (2)$$

Positions of coordinate frame origins of each links are obtained through results of forward kinematics: $\mathbf{r}_{O_1}^{(0)}(\mathbf{q}) = [0,0,d_1]^T$,

$$\mathbf{r}_{O_2}^{(0)}(\mathbf{q}) = [d_2\sin q_1, -d_2\cos q_1, d_1]^T,$$

$$\mathbf{r}_{O_3}^{(0)}(\mathbf{q}) = [d_2\sin q_1 + d_3\cos q_1\sin q_2, -d_2\cos q_1 + d_3\sin q_1\sin q_2, d_1 - d_3\cos q_2]^T \quad (3)$$

Due to the offsets in the shoulder and wrist, no two coordinate frame origins coincide. The expressions for the position and orientation of the end-effector are highly complex in terms of the joint variables. As a result, the position-orientation decoupling method cannot be applied to the inverse kinematics problem with this configuration.

2.2. Differential Kinematics and Jacobi Matrix

The end-point velocity and the angular velocity of the end-effector link are linearly related to the joint velocities $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_7]$.

$$\mathbf{v}_E^{(0)} = \mathbf{J}_T(\mathbf{q})\dot{\mathbf{q}} \quad (4)$$

$$\boldsymbol{\omega}^{(0)} = \mathbf{J}_R(\mathbf{q})\dot{\mathbf{q}} \quad (5)$$

In these formulation, the rotational and translational Jacobian matrices are obtained as following:

$$\mathbf{J}_T(\mathbf{q}) = [\tilde{\mathbf{z}}_0^{(0)}(\mathbf{r}_E - \mathbf{r}_{O_0}), \tilde{\mathbf{z}}_1^{(0)}(\mathbf{r}_E - \mathbf{r}_{O_1}), \dots, \tilde{\mathbf{z}}_6^{(0)}(\mathbf{r}_E - \mathbf{r}_{O_6})] \quad (6)$$

$$\mathbf{J}_R(\mathbf{q}) = [\mathbf{k}, \mathbf{R}_1^0\mathbf{k}, \dots, \mathbf{R}_{i-1}^0\mathbf{k}, \dots, \mathbf{R}_6^0\mathbf{k}], \quad (7)$$

where $\mathbf{k} = [0,0,1]^T$

2.3. Position and Orientation of End-Effector

Let $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ be the position and orientation vector of end-effector, in which:

$\boldsymbol{\eta}_1 = [x_E \quad y_E \quad z_E]^T$ is vector of position of end-effector in fixed coordinate system;

$\boldsymbol{\eta}_2 = [\psi \quad \theta \quad \phi]^T$ is vector of Roll-Pitch-Yaw angles ($Z-Y-X$ Euler angles).

Rotation matrix of end-effector as function of $\boldsymbol{\eta}_2$ is given by: $\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$

The angular velocity vector of the end-effector in terms of $Z-Y-X$ Euler angles is computed as::

$$\boldsymbol{\omega}^{(0)} = \mathbf{Q}(\boldsymbol{\eta}_2)\dot{\boldsymbol{\eta}}_2 \tag{8}$$

and $\dot{\boldsymbol{\eta}}_2 = \mathbf{Q}^{-1}(\boldsymbol{\eta}_2)\boldsymbol{\omega}^{(0)}$ (9)

with $\mathbf{Q}(\boldsymbol{\eta}_2) = \begin{bmatrix} 0 & -\sin\psi & \cos\psi\cos\theta \\ 0 & \cos\psi & \sin\psi\cos\theta \\ 1 & 0 & -\sin\theta \end{bmatrix}$,

$$\mathbf{Q}^{-1}(\boldsymbol{\eta}_2) = \begin{bmatrix} \cos\psi\tan\theta & \sin\psi\tan\theta & 1 \\ -\sin\psi & \cos\psi & 0 \\ \cos\psi/\cos\theta & \sin\psi/\cos\theta & 0 \end{bmatrix}$$

To ensure the existence of the inverse matrix $\mathbf{Q}^{-1}(\boldsymbol{\eta}_2)$, Pitch angle θ must satisfy: $\theta \neq \pm\pi/2$..

The velocity vector of the end-effector, including linear and angular velocity, is expressed as:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_T(\mathbf{q}) \\ \mathbf{Q}^{-1}(\boldsymbol{\eta}_2)\mathbf{J}_R(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}(\boldsymbol{\eta}_2, \mathbf{q})\dot{\mathbf{q}} \tag{10}$$

2.4. Inverse Kinematic

The inverse kinematics problem is formulated as follows: Given the position and orientation of the end-effector, represented by $\mathbf{r}_7^{(0)}$ and $\mathbf{R}_7^0$ through six parameters $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$, we need to determine the joint variables $q_k, k = 1,\ldots,7$..

Due to the offsets at the shoulder and wrist, Pieper's method (position-orientation decoupling) cannot be applied to solve the inverse kinematics of the collaborative robot [2]. Additionally, since a 7-DOF collaborative robot is a redundant manipulator, the inverse kinematics problem needs to be solved at the velocity level based on (10).

Assuming the Jacobian matrix is $\mathbf{J} = \mathbf{J}(\boldsymbol{\eta}_2, \mathbf{q})$, a 6×7 matrix with rank 6, and given vectors $\boldsymbol{\eta}$ and $\mathbf{q}$, the inverse kinematics problem consists of 6 equations with 7 unknown joint velocities $\dot{\mathbf{q}}$. Using the weighted pseudoinverse matrix method, we obtain the optimal solution for $\dot{\mathbf{q}}$ as:

$$\dot{\mathbf{q}} = \mathbf{J}_W^+\dot{\boldsymbol{\eta}} + [\mathbf{I} - \mathbf{J}_W^+\mathbf{J}]\mathbf{z}_0 \tag{11}$$

with $\mathbf{J}_W^+ = \mathbf{W}^{-1}\mathbf{J}^T[\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T]^{-1}$ (12)

where $\mathbf{J}_W$ is the weighted pseudoinverse matrix of $\mathbf{J}$; $\mathbf{z}_0 \in \mathbb{R}^7$ is an arbitrary vector, $\mathbf{I} \in \mathbb{R}^{7\times7}$ is the identity matrix, and $\mathbf{W} \in \mathbb{R}^{7\times7}$is a weighting matrix.

In certain special cases, if the weighting matrix is the identity matrix, we obtain:

$$\mathbf{J}^+ = \mathbf{J}^T[\mathbf{J}.\mathbf{J}^T]^{-1} \tag{13}$$

where $\mathbf{J}^+$ is the pseudoinverse of the Jacobian matrix. Substituting into Equation (11), we get:

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\boldsymbol{\eta}} + [\mathbf{I} - \mathbf{J}^+\mathbf{J}]\mathbf{z}_0 \tag{14}$$

By selecting an appropriate vector $\boldsymbol{z}_0$, the redundancy of the manipulator can be leveraged to avoid kinematic singularities, obstacle, and joint limit. Typically, the vector $\boldsymbol{z}_0$ is chosen as follows:

$$\boldsymbol{z}_0 = \alpha\left(\frac{\partial\phi(q)}{\partial\boldsymbol{q}}\right)^T \tag{15}$$

where α is a constant and greater than 0, and $\phi(\boldsymbol{q})$ is the objective function. Since the solution shifts along the gradient of the objective function, the goal is to locally maximize it while satisfying kinematic constraints. Typical objective functions include following:

- The distance measurement to joint midpoints, defined as:

$$\phi(\boldsymbol{q}) = -\frac{1}{2}\sum_{i=1}^{7} c_i\left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}}\right)^2 \tag{16}$$

where q_{iM}, q_{im} and $\bar{q}_i$ represent the maximum limit, minimum limit, and midpoint of the joint's working range, respectively, with positive weights c_i. Maximizing this function drives the joint variables toward their mid-range values, utilizing redundancy to keep joints away from their limits, thus avoiding joint collision to joint limits.

- The distance measurement to obstacles, defined as:

$$\phi(\boldsymbol{q}) = \min\|\mathbf{p}(\mathbf{q}) - \mathbf{o}\| \tag{17}$$

where $\mathbf{o}$ is position vector of an arbitrary point belonging to the obstacle (e.g: the mass center), $\mathbf{p}(\mathbf{q})$ is the position of the point on the manipulator that approaches an obstacle too closely.

The joint variables $\mathbf{q}(t)$ are computed using:

$$\mathbf{q}(\boldsymbol{t}) = \mathbf{q}(t_0) + \int_{t_0}^{t}\dot{\mathbf{q}}\,dt \tag{18}$$

To reduce accumulated errors from numerical integration, kinematic feedback is applied, and the joint velocity $\dot{\mathbf{q}}$ is computed as:

$$\dot{\mathbf{q}} = \mathbf{J}^+(\dot{\boldsymbol{\eta}} + \mathbf{K}(\boldsymbol{\eta} - \boldsymbol{\eta}(\mathbf{q})) + [\mathbf{I} - \mathbf{J}^+\mathbf{J}]\mathbf{z}_0 \tag{19}$$

The block diagram of the velocity-level inverse kinematics algorithm using the Jacobian method is shown in Fig. 2.

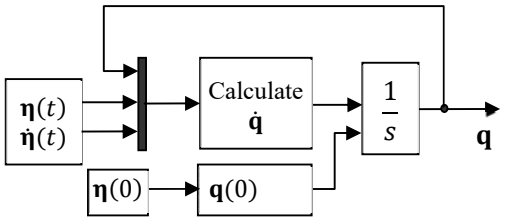

Fig. 2. The block diagram of the velocity-level inverse kinematics algorithm using the Jacobian method

3. Dynamics and Control

3.1. Dynamic Model

The equations of motion are derived using Lagrange's approach. Dynamic parameters are presented in Table 3. The differential equation of motion is formulated as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u} \quad (20)$$

The mass matrix is computed as following:

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^{n}\left[\mathbf{J}_{Ti}^{T} m_i \mathbf{J}_{Ti} + \mathbf{J}_{Ri}^{T}\mathbf{R}_i \mathbf{I}_{Ci}^{i}\mathbf{R}_i^{T}\mathbf{J}_{Ri}\right] \quad (21)$$

The generalized force vector due to gravity is given by:

$$\mathbf{g}(\mathbf{q}) = -\sum_{i=1}^{n} m_i \mathbf{J}_{Ti}^{T}(\mathbf{q})\mathbf{g}^{(0)} \quad (22)$$

The Coriolis matrix is derived from the mass matrix using Christoffel symbols:

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \{C_{ij}(\mathbf{q},\dot{\mathbf{q}}\} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k,$$

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i}\right) \quad (23)$$

3.2. Trajectory Tracking Control

Using the outcomes of the inverse kinematics problem, we formulate the trajectory tracking control problem through inverse dynamics. The problem's input is the predetermined trajectory of the end-effector. The intended joint trajectory is established using a path planning algorithm, followed by the application of inverse kinematics to ascertain the requisite joint angles, then deriving the necessary torques and forces operating on the joints. The objective of the control challenge is to ensure the robot's end-effector moves as intended and that the tracking error approaches zero.

$$\ddot{\tilde{\mathbf{q}}} + \mathbf{K}_d\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}} = \mathbf{0} \quad (24)$$

To achieve this, we select the control function as follows:

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\mathbf{y} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (25)$$

in which $\mathbf{y} = \ddot{\mathbf{q}}_d + \mathbf{K}_d\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}}$, $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$.

The algorithm diagram for trajectory tracking control based on inverse dynamics is illustrated in Fig. 3.

4. Simulation Results

The inverse kinematics and trajectory tracking control problem based on inverse dynamics is simulated in MATLAB.

The desired trajectory is designed as follows:

$$s(t) = s_i + \frac{(s_f - s_i)}{\pi}\left(\frac{\pi t}{t_f} - \frac{1}{2}\sin\frac{2\pi t}{t_f}\right), 0 \le t \le t_f \quad (26)$$

The orientation of the end-effector, described by Roll-Pitch-Yaw $(R-P-Y)$ angles, is given as follows:

$$\boldsymbol{\eta}_2(t) = \boldsymbol{\eta}_2(0) + \frac{\boldsymbol{\eta}_2(t_f) - \boldsymbol{\eta}_2(0)}{\pi}\left(\frac{\pi t}{t_f} - \frac{1}{2}\sin\frac{2\pi t}{t_f}\right),$$

The trajectory is designed in two following cases:

Case 1: Circular trajectory

The end-effector moves along a circular path with radius $R = 0.133$ m, begins from $A\,(0.3335, -0.4781,\ 0.2654)$ to $B\,(0.3205,\ -0.2131,\ 0.2855)$.

Table 3. Dynamic parameters of 7-dof collaborative robot

	Link 1	Link 2	Link 3	Link 4	Link 5	Link 6	Link 7
Mass [kg]	4.89	4.14	4.12	2.42	2.74	1.74	0.33
Moment of Inertia							
$I_{xi}[\text{kg.m}^2]$	0.012	0.11	0.08	0.04	0.067	0.00188	0.000339
$I_{yi}[\text{kg.m}^2]$	0.01	0.1	0.007	0.04	0.003	0.00188	0.000339
$I_{zi}[\text{kg.m}^2]$	0.01	0.007	0.008	0.003	0.067	0.00213	0.000528

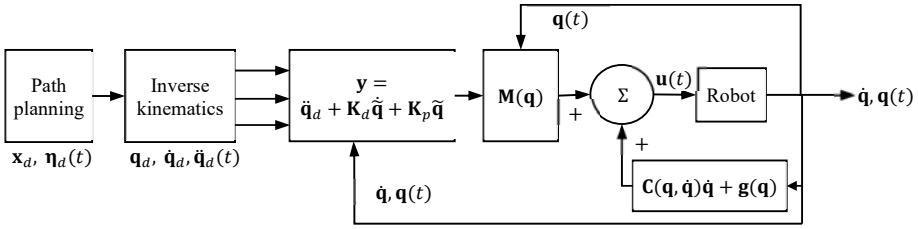

Fig. 3. The algorithm model for trajectory tracking control based on inverse dynamics

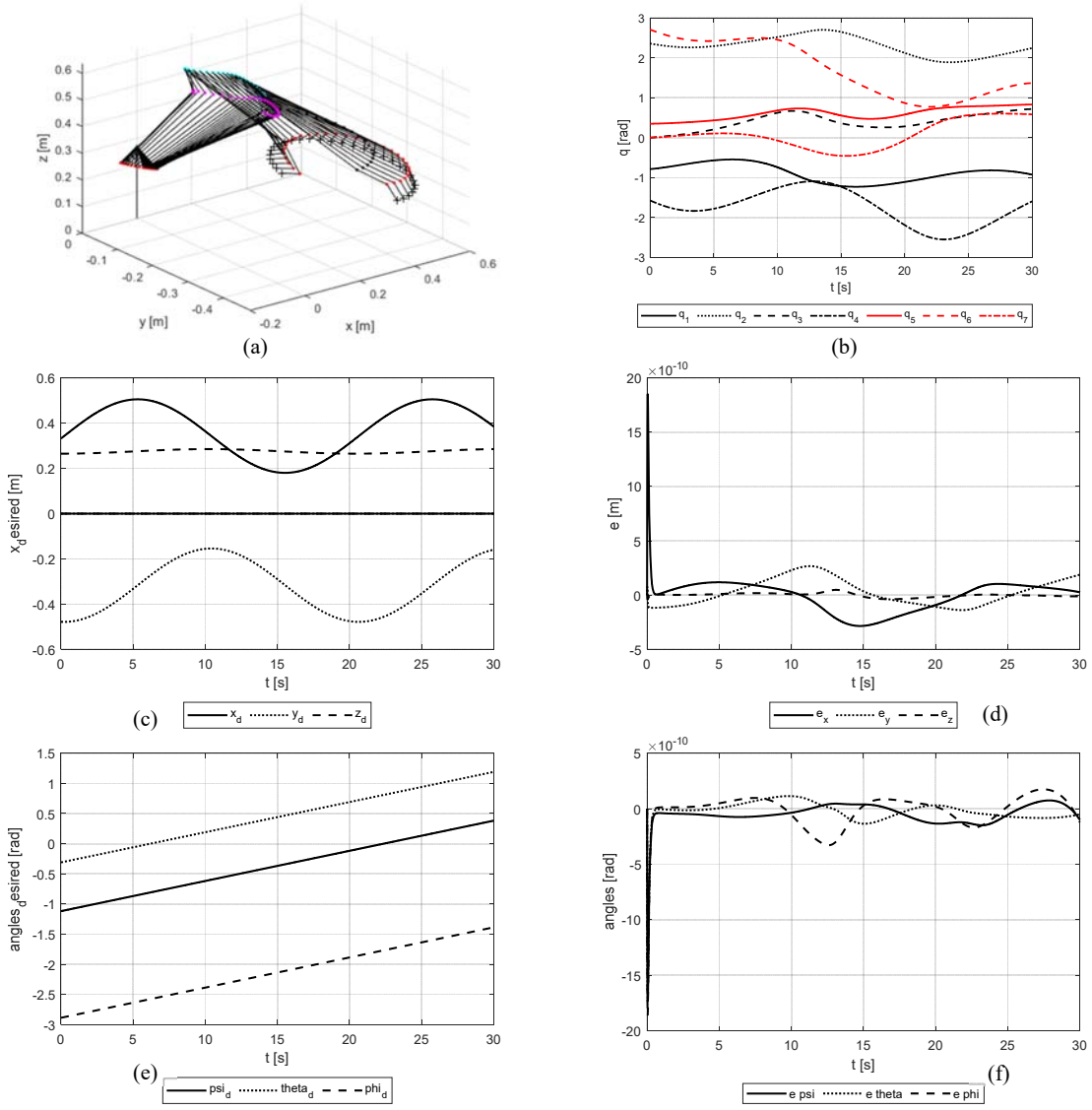

Fig. 4. Simulation of inverse kinematics using joint limit avoidance in circular trajectory;
(a) Motion of cobot model; (b) Joint variables with respect to time;
(c) Desired position of end-effector; (d) Position error of end-effector;
(e) Desired orientation angles of end-effector; (f) Orientation angle error of end-effector

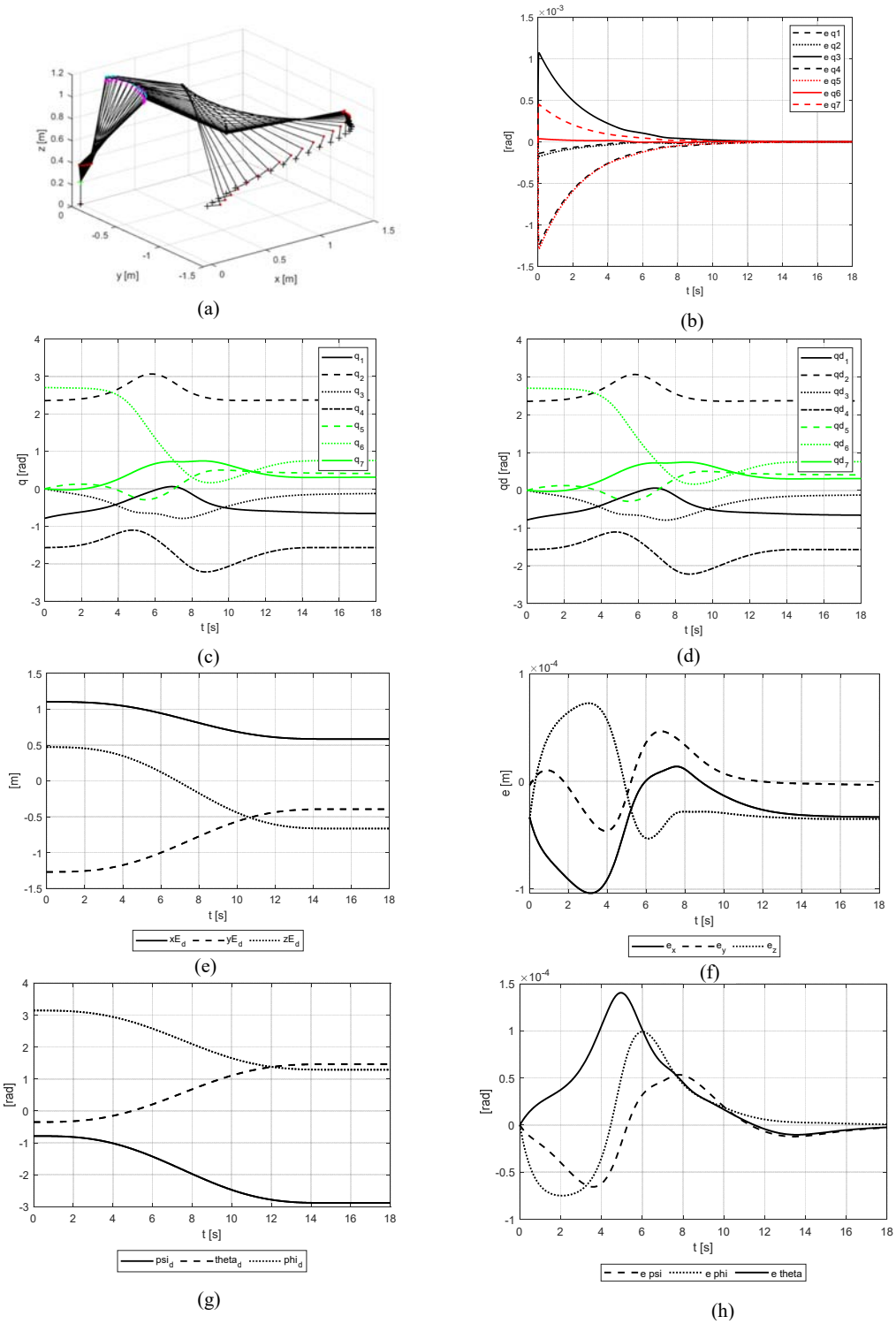

Fig. 5. Simulation of inverse dynamics using joint limit avoidance in rectilinear trajectory;
(a) Motion of cobot model; (b) Error of joint variables with respect to time;
(c) Calculated joint variables with respect to time; (d) Desired joint varialbes with respect to time;
(e) Desired position of end-effector; (f) Position error of end-effector;
(g) Desired orientation angles of end-effector; (h) Orientation angle error of end-effector

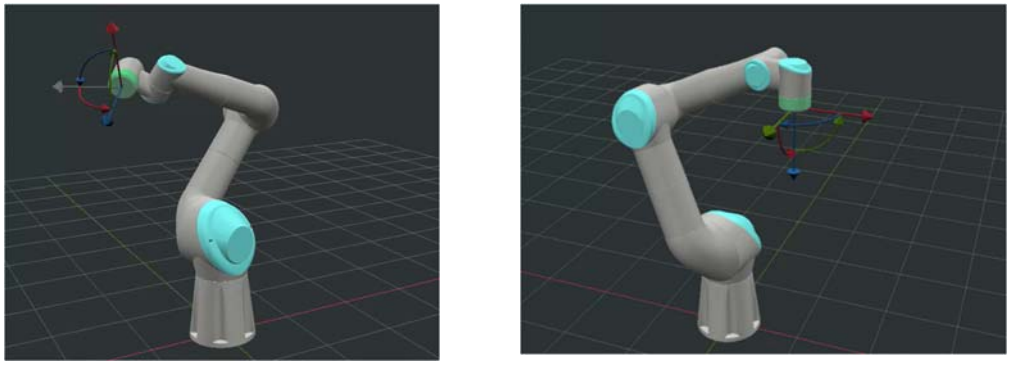

Fig. 6. Visualization of robot in different poses on ROS environment

Fig. 4 shown the inverse kinematics with joint limit avoidance solution. The precision of the suggested approach is demonstrated by the inaccuracy of 10^{-10} m. The robot moves continuously, without any sudden changes. The joint variations are in safe operational areas, so they can lower the danger of running across kinematic singularities and hardware constraints.

Case 2. Rectilinear trajectory

The end-effector moves along a rectilinear path from $A\,(0.3534, -0.4814,\ 0.2818)$ to $B\,(0.2586,\ -0.3318,\ 0.2256)$.

The example of trajectory control with joint limit avoidance is depicted Fig. 5. The error in this calculation is 10^{-4} m, which indicates that both in position and orientation the suggested computing techniques are efficient. For pragmatic uses, the smooth and continuous trajectories of all links can be advantageous. Well used during robot motion is the joint limit avoidance.

Robot motion in ROS environment has been implemented using MATLAB computation data (Fig. 6). Both position and orientation angles in various robot paths planning (circular and rectilinear path) are shown in Fig. 6a and Fig. 6b in different poses of robot. These figures show that the numerical computations are concised and ready for use practically.

5. Conclusion

This work addresses the inverse kinematic and dynamic problems of a 7-degree-of-freedom collaborative robot, focussing on the avoidance of joint limits. Inverse kinematic control has been implemented with great accuracy in two distinct trajectory path planning methods: curved and rectilinear paths. The null space of the Jacobian matrix has been utilised for joint limit avoidance in both kinematic and dynamic problems. The computational efficiency has been demonstrated by simulation results in MATLAB and ROS implementation. Future research aims to enhance the computational efficiency and task specificity for robotic applications.

Acknowledgments

This work was supported in part by the National Program: Support for research, development, and technology application of industry 4.0 (KC-4.0/19-25), under the grant for the project: Research, design and manufacture of Cobot applied in industry and some other fields with human-robot interaction (code: KC-4.0-35/19-35).

References

[1] Financial Times, March of the cobots: the technology lowering the barrier to automation, Financial Times, May 27, 2025. [Online]. Available: https://www.ft.com/content/78c1d4e9-ad30-47f5-ab7b-390df5bc1f10

[2] J. J Craig, Introduction to Robotics: Mechanics and Control, 3rd ed., Pearson/Prentice Hall, 2005.

[3] J. Wang, Y. Li, and X. Zhao, Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm, International Journal of Advanced Robotic Systems, vol. 7, no. 4, Jan. 2010. https://doi.org/10.5772/10495

[4] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control, Springer Science & Business Media, 2009. https://doi.org/10.1007/978-1-84628-642-1

[5] M. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, John Wiley & Sons, New York, 2006.

[6] X. Tian, Q. Xu, Q. Zhan, An analytical inverse kinematics solution with joint limits avoidance of 7-DOF anthropomorphic manipulators without offset, Journal of the Franklin Institute, Dec. 2020.

[7] Y. Wang, P. Artemiadis, Closed-form inverse kinematic solution for anthropomorphic motion in redundant robot arms, Advances in Robotics & Automation, Jan. 2013. https://doi.org/10.4172/2168-9695.1000110

[8] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, K. Kosuge, Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution, IEEE Transactions on Robotics, vol. 24, no. 5, pp. 1131-1142, Oct. 2008. https://doi.org/10.1109/TRO.2008.2003266

[9] A. Reiter, A. Müller, H. Gattringer, On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators, IEEE Transactions on Industrial Informatics, vol. 14, iss. 4, pp. 1681–1690, Apr. 2018. https://doi.org/10.1109/TII.2018.2792002

[10] D. Tolani, A. Goswami, N. I. Badler, Real-time inverse kinematics techniques for anthropomorphic limbs, Graphical Models, vol. 62, iss. 5, pp. 353–388, Sep. 2000. https://doi.org/10.1006/gmod.2000.0528

[11] Z. Mu, H. Yuan, W. Xu, T. Liu, B. Liang, A segmented geometry method for kinematics and configuration planning of spatial hyper-redundant manipulators, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 5, pp. 1746-1756, May 2020. https://doi.org/10.1109/TSMC.2017.2784828

[12] L. Jiang, X. Huo, Y. Liu, H. Liu, An integrated inverse kinematic approach for the 7-DOF humanoid arm with offset wrist, in 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, Mar. 2013, pp. 2737–2742. https://doi.org/10.1109/ROBIO.2013.6739888