Kinematics modeling of six degrees of freedom humanoid robot arm using improved damped least squares for visual grasping

Muhammad Ramadhan Hadi Setyawan[1,3], Raden Sanggar Dewanto[2,3], Bayu Sandi Marta[1,3], Eko Henfri Binugroho[2,3], Dadet Pramadihanto[1,3]
[1]Department of Information and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia
[2]Department of Energy Mechanics Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia
[3]Robotics and Intelligent Systems Center, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

Article Info

Article history:

Received Feb 24, 2022
Revised Jul 22, 2022
Accepted Aug 13, 2022

Keywords:

Damped least squares
Humanoid robot
Kinematics
Robotic arm
Visual grasping systems

ABSTRACT

The robotic arm has functioned as an arm in the humanoid robot and is generally used to perform grasping tasks. Accordingly, kinematics modeling both forward and inverse kinematics is required to calculate the end-effector position in the cartesian space before performing grasping activities. This research presents the kinematics modeling of six degrees of freedom (6-DOF) robotic arm of the T-FLoW humanoid robot for the grasping mechanism of visual grasping systems on the robot operating system (ROS) platform and CoppeliaSim. Kinematic singularity is a common problem in the inverse kinematics model of robots, but. However, other problems are mechanical limitations and computational time. The work uses the homogeneous transformation matrix (HTM) based on the Euler system of the robot for the forward kinematics and demonstrates the capability of an improved damped least squares (I-DLS) method for the inverse kinematics. The I-DLS method was obtained by improving the original DLS method with the joint limits and clamping techniques. The I-DLS performs better than the original DLS during the experiments yet increases the calculation iteration by 10.95%, with a maximum error position between the end-effector and target positions in path planning of 0.1 cm.

Corresponding Author:

Dadet Pramadihanto
Robotics and Intelligent Systems Center (RoISC), Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya
PENS Campus, Jalan Raya ITS, Keputih-Sukolilo, Surabaya, Indonesia, 60111
Email: dadet@pens.ac.id

1. INTRODUCTION

In the last few decades, robot manipulators have been increasingly used in the industry. Many approaches to building a robot manipulator include finite element analysis [1]. In a humanoid robot, the manipulator is an arm and usually performs like the human arm to grasp an object. An example of implementing a grasping task using a robotic manipulator or arm is pick-and-place operations of the assembly process in the manufacturing industry [2]. With the rapid development of robotics, the robotics and intelligent systems center (RoISC) is also participating and developing humanoid robots to be helpful to humans, FLoW humanoid robot [3], [4]. The current version of the humanoid robot T-FLoW is installed with a six degrees of freedom (6-DOF) robotic arm that functions as an arm to perform tasks like the grasping task.

The kinematics vectors of the arm robot model are calculated using forward and inverse kinematics. The forward kinematics transform the joint space vectors to the cartesian space vectors whilst the inverse kinematics do the opposite way [5]. The most used method to solve forward kinematics is the homogeneous transformation matrix (HTM) based on the Denavit-Hartenberg (D-H) parameters [6]. The HTM can be defined from the rotation and the translation elements in the D-H parameters or directly from the Euler system of the robot. There are numerous methods to solve the inverse kinematics problem. The numerical method solves the inverse kinematics of the robot manipulator that cannot be solved in closed form using the new inverse kinematics algorithm (NIKA) [7]. The geometric method implements the mathematical solution based on geometric algebra to find the reverse of the forward kinematics [8], [9]. The pseudoinverse of the Jacobian matrix solves the problem of computing the inverse matrix when the matrix is a non-square matrix [10]. The inverse kinematics is based on the weighted least-norm solution that avoids the joint angular position and velocity limits [11]. The kinematics solutions of the robotic arm are immense. They can be obtained from either position-based or velocity-based methods.

The kinematics model is not only the study used to build a robot manipulator. The dynamics model is also an aspect that must be considered to developing a robot manipulator. The Euler-Lagrange formulation is the conventional method for building the dynamic model. An Euler-Lagrange formulation is an approach based on the kinetic and potential energy of the robot [12]. The dynamic model based on the assumed mode method (AMM), a discrete method for continuous systems, uses the linear sum of assumed mode to describe the vibration of the robot body [13]. The other approach to defining the dynamic model is the system identification method. The system identification method creates a mathematical model using input and output data from a real dynamic system [14]. However, there are many approaches to defining the dynamics model of the robot. Therefore, this research will not discuss the dynamics model but only focus on the kinematics model of the robot manipulator.

The kinematics singularities are the common problem in the kinematics model of the robot manipulators or the robotic arm. The kinematics singularities occur when the robot is in the neighborhood of a singular configuration, which causes limitations in the motion [15]. Many research discusses the methods to avoid such singularity, for example, the singular value decomposition (SVD) of the Jacobian matrix [16], the damped least-squares method [17]–[19], the singular robustness algorithm [20], the virtual spring model, and joint-based damping control [21]. All research aims to find the best algorithm for computation, efficiency, and robustness kinematics.

The kinematics singularity is not the only thing to consider in controlling the robotic arm. Such problems that must be considered are a mechanical restriction, actuator safety, and natural motion configuration. To overcome those problems, the joint limits technique is used to limit the angle of each joint to accommodate the mechanical restriction and provide the natural motion configuration [22]. Whilst the clamping technique creates the reachable target when the real target is unreachable or outside the workspace area. This clamping technique will reduce the amount of iteration when calculating the solution since the target will never exceed the workspace area of the robot [23]. The original damped least-squares (DLS) method can solve kinematics singularity yet does not have the safety factor of the actuator and does not consider the unreachable target.

We developed a kinematics model for the 6-DOF robotic arm of the T-FLoW humanoid robot with a visual grasping system. The forward kinematics model using the HTM was obtained from the Euler system of the arm robot. This research improves the original DLS method with a simple joint limits technique using the scale back law and the clamping technique, the improved damped least-squares (I-DLS) method for the inverse kinematics model.

The main contributions of this research can be summarized as: i) the system of the kinematics model on the robot operating system (ROS) platform and CoppeliaSim simulator; ii) the improvement of the original DLS by combining with the joint limits technique to create a safety factor for the joint actuator and the clamping technique to reduce the amount of computation iteration; iii) the improvement of the original DLS by combining with the clamping technique reduces the amount of iteration of calculation because of the unreachable target; and iv) the movement mechanism of the grasping activities.

2. FORWARD KINEMATICS

The forward kinematics model transforms the joint space into the cartesian space. This model provides the position of the end-effector of the robot in the world coordinate. As shown in Figure 1, The robot in this research consists of six revolute joints built by a double spherical wrist configuration. The mechanical structure and the joint coordinate system of the robot, as shown in Figure 1(a), with the subscript number i $(i = 1, \dots, 7)$, θ_i values represent the angle values of joint-1 to joint-6 and a_n represents the length of link-1 to link-6 with 0 offsets of all links, and will be used to create the kinematics model and describe the kinematic relationship between the position of the end-effector and the joint angle of the robot. Figure 1(b) shows the position of each joint in the robot model.

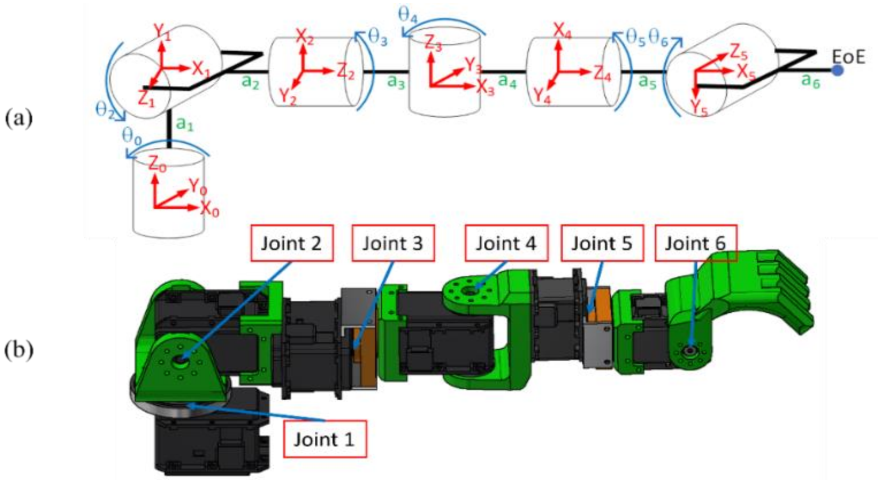


Figure 1. 6-DOF robot arm of T-FLoW humanoid robot (a) structure of the robot and (b) joint position

Forward kinematics solution can be solved using the HTM [24]. Combining the 3×3 rotation matrix (R_n^0) and the 3×1 translation or displacement matrix (d_n^0), the homogeneous transformation matrix (H_n^0) is obtained as (1):

$$H_n^0 = \begin{bmatrix} & R_n^0 & & d_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where n is defined as the number of joints. Using the Euler system of the robot based on the robot coordinate system in Figure 1(a), the homogeneous transformation matrices of all joints from the first joint to the sixth joint are:

$$H_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & sin(\theta_1) & 0 \\ sin(\theta_1) & 0 & -cos(\theta_1) & 0 \\ 0 & 1 & 0 & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_2^1 = \begin{bmatrix} -sin(\theta_2) & 0 & cos(\theta_2) & a_2cos(\theta_2) \\ cos(\theta_2) & 0 & \sin(\theta_2) & a_2\sin(\theta_2) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H_3^2 = \begin{bmatrix} 0 & sin(\theta_3) & cos(\theta_3) & 0 \\ 0 & -cos(\theta_3) & sin(\theta_3) & 0 \\ 1 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_4^3 = \begin{bmatrix} 0 & sin(\theta_4) & cos(\theta_4) & a_4cos(\theta_4) \\ 0 & -cos(\theta_4) & sin(\theta_4) & a_4sin(\theta_4) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$H_5^4 = \begin{bmatrix} 0 & sin(\theta_5) & cos(\theta_5) & 0 \\ 0 & -cos(\theta_5) & sin(\theta_5) & 0 \\ 1 & 0 & 0 & a_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; H_6^5 = \begin{bmatrix} 0 & -cos(\theta_6) & sin(\theta_6) & a_6cos(\theta_6) \\ 0 & -sin(\theta_6) & -cos(\theta_6) & a_6sin(\theta_6) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where sin is indicated sine function and cos for cosine function.

The end-effector robot position can be obtained from the total HTM by multiplying each HTM of the joints, as shown in (3).

$$H_6^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 H_6^5 \tag{3}$$

The (3) can be simplified to get the vector position $p \in \mathbb{R}^3$ of the end-effector in the cartesian space, as explained in (4).

$$H_6^0 = \begin{bmatrix} {H_6^0}_{11} & {H_6^0}_{12} & {H_6^0}_{13} & {H_6^0}_{14} \\ {H_6^0}_{21} & {H_6^0}_{22} & {H_6^0}_{23} & {H_6^0}_{24} \\ {H_6^0}_{31} & {H_6^0}_{32} & {H_6^0}_{33} & {H_6^0}_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} P_{Xeoe} \\ P_{Yeoe} \\ P_{Zeoe} \end{bmatrix} = \begin{bmatrix} H^0_{6\,14} \\ H^0_{6\,24} \\ H^0_{6\,34} \end{bmatrix} \tag{5}$$

3. INVERSE KINEMATICS

Since the solution of inverse kinematics is infinite, many approach algorithms are used to solve inverse kinematics problems, such as the geometric and numeric methods based on the structure and coordinate system of the robot. This research uses the DLS method combined with joint limits and a clamping technique called the I-DLS. This method is used to find the joint angles of the 6-DOF robotics arm of the T-FLoW humanoid robot.

3.1. Damped least-squares

The general method to solve the singularity is the DLS, which has not been able to be solved by the pseudoinverse method. For the robot manipulator or arm that has joints connected by links, the angle values of the joint can be written as vector column $\theta = (\theta_1, \theta_2, \dots, \theta_n)^T$, n is the number of joints. The end-effector position in the cartesian space can be written as a vector column $\vec{s} = (s_1, s_2, \dots, s_m)^T$ and the target positions as a vector column $\vec{t} = (t_1, t_2, \dots, t_m)^T$, with m is the number of values from $\mathbb{R}^3$. We use forward kinematics to find the $\vec{s}$ in (5), $\vec{s} = \left(P_x, P_y, P_z\right)^T$.

The basic algorithm of inverse kinematics is to calculate the vector joint to reach the target position and can be expressed as (6).

$$t = s(\theta). \tag{6}$$

The (6) was solved using the iterated local search (ILS) algorithm based on the $m \times n$ Jacobian matrix J which is defined as (7):

$$J(\theta) = \left(\frac{\partial s}{\partial \theta}\right) \tag{7}$$

and computes the value of $\Delta\theta$ by using the current values of $\theta, \vec{s}$ and $\vec{t}$ to obtain the change of end-effector position $\vec{e}$, as explained in (8).

$$\vec{e} = \vec{t} - \vec{s} = J\Delta\theta. \tag{8}$$

Furthermore, the algorithm updates the values of joint angle θ by adding up the $\Delta\theta$ value of each iteration, which can be expressed as (8)

$$\theta = \theta + \Delta\theta. \tag{9}$$

The pseudoinverse method [25] calculates the pseudoinverse of the Jacobian matrix J to compute the addition of joint angles, as explained in (10).

$$\Delta\theta = J^{+}\vec{e}. \tag{10}$$

However, if the size of Jacobian matrix J is not full-row rank or $rank(J) < m$, the pseudoinverse method is unstable near the singularity and becomes unconditioned.

In [18], [19] propose an inverse kinematics solution using the damped least-squares method to overcome the singularity problem. The damped least-squares computing the value of $\Delta\theta$ by minimizes the quantity of (11):

$$||J\Delta\theta - \vec{e}||^2 + \lambda^2||\Delta\theta||^2 \tag{11}$$

where $\lambda \in \mathbb{R} > 0$ is a non-zero damping factor. The DLS computes the $\Delta\theta$ values by minimizing tracking errors and joint velocities but compromising the feasibility and the accuracy of the solution. This method accords to the solution of (12).

$$(J^T J + \lambda^2 I)\Delta\theta = J^T \vec{e} \tag{12}$$

Thus, the DLS solution gives numerically stable to select $\Delta\theta$ values, as explained in (13).

$$\Delta\theta = J^T (JJ^T + \lambda^2 I)^{-1} \vec{e} \tag{13}$$

According to the laws identified in [17]–[19], [26], selecting the damping factor λ can be dynamically using different methods. In this research, the damping factor λ dynamically selected as (14):

$$\lambda = \begin{cases} \lambda_0 1\left(-\frac{\omega}{\omega_0}\right), & \text{if } \omega < \omega_0 \\ 0, & \text{if } \omega \geq \omega_0 \end{cases}, \tag{14}$$

where the λ_0 value is the value defined by the user to obtain the solution around the singularity. The ω_0 value is the definition of the size of the singular region, and the ω value is the scalar objective function of the joint variable that can be calculated by (15).

$$\omega = \sqrt{det(JJ^T)} \tag{15}$$

3.2. Improved damped least-squares

The I-DLS affects the original DLS in (13). The joint limits technique limits the $\Delta\theta$ values to a certain range based on the user determination. Also, the I-DLS creates the alternative target position as input of inverse kinematics to obtain the change of end effector position $\vec{e}$ with the clamping technique.

3.2.1. Joint limits

The joint limits technique is used to limit the angle of each joint, which is useful so that the actuator does not reach an excessive angle to avoid damaging the actuator or the link and can be used to achieve a natural motion configuration. The actuator of each joint has the minimum value $\theta_{i\min}$ and the maximum value $\theta_{i\max}$, thus the angle from the calculation result of inverse kinematics θ_i should not exceed those values. We are using the scale back law to limit the value of joint angle θ_i as shown in (16):

$$\theta_i = \begin{cases} \theta_i \times \frac{\theta_{i\max}}{\max(\Delta\theta)}, & \text{if} \max(\Delta\theta) > \theta_{i\max} \\ \theta_i \times \frac{\theta_{i\min}}{\min(\Delta\theta)}, & \text{if} \min(\Delta\theta) < \theta_{i\min} \\ \theta_i, & otherwise \end{cases} \tag{16}$$

where i is the joint number. The values of $\theta_{i\min}$ and $\theta_{i\max}$ can be specified according to the requirement of the user or can refer to the human arm mechanism to limit the angle. The minimum and maximum values of $\Delta\theta$ are used to limit the inverse kinematics angle θ_i based on the $\theta_{i\min}$ and $\theta_{i\max}$.

3.2.2. Clamping the distance of the target position

When the target position is unreachable or beyond the workspace of the robot, the original DLS fails to calculate the joint angle, and the error position will be enormous. Clamping the target position reduces the position error when the target is unreachable by moving the target closer to the end-effector position and does not exceed the workspace area. Using the technique that refers to [23], this method works by clamping the length of $\vec{e}$ in the (8), can be written as (17):

$$e = ClampMag(t - s, D_{max}) \tag{17}$$

where the $ClampMag(t - s, D_{max})$ is the function to find the new target position e based on the input target t, as explained in (18).

$$ClampMag(w, d) = \begin{cases} w, & \text{if } \|w\| \leq d \\ d\frac{w}{\|w\|}, & otherwise \end{cases} \tag{18}$$

The $\|w\|$ is representing the Euclidean norm of w and the D_{max} value is an upper bound on how far the end-effector moves in a single update step.

4. RESULTS AND DISCUSSION

4.1. Experimental platform

The experimental platform of this research, as shown in Figure 2, aims to study the kinematics model of the 6-DOF robotic arm T-FLoW humanoid robot. The simulation environment, including the full body of the robot and the object target, was built in the CoppeliaSim simulator V4.2.0 [27] using Vortex physical engine [28]. The kinematics model was built using C++ based on the robot operating system Foxy Fitzroy (ROS 2) [29]. The motion path was generated using the path planner pen motion planning library (OMPL) [30] in the CoppeliaSim. Two ROS nodes regarding the kinematics model and the CoppeliaSim communicate through the two ROS topics. The stereo camera model in the CoppeliaSim node is used to obtain the position of the object in the world coordinate. The blob detection method gets the position of the object relative to the stereo camera perspective and transforms it into the world coordinate perspective. The path planner will create the motion path of the robot between the initial end-effector position and the target position. The target position of each step in the motion path will be sequentially sent to the kinematics node through the target pos topic. The kinematics node will calculate the angle of all joints using the kinematics model and send it to the set angle topic, then will be forwarded to the CoppeliaSim node. Each joint of the arm robot in the simulation will move based on the set angle topic. All joints in the simulation are used as the actuator that rotates like servo motors, using an internal proportional-integral-derivative (PID) controller in all joints to maintain the angle position. The experiment was conducted using a computer with Ubuntu 20.04 LTS, AMD Ryzen 5 3550H, and 16 GB DDR4 to run the simulation.

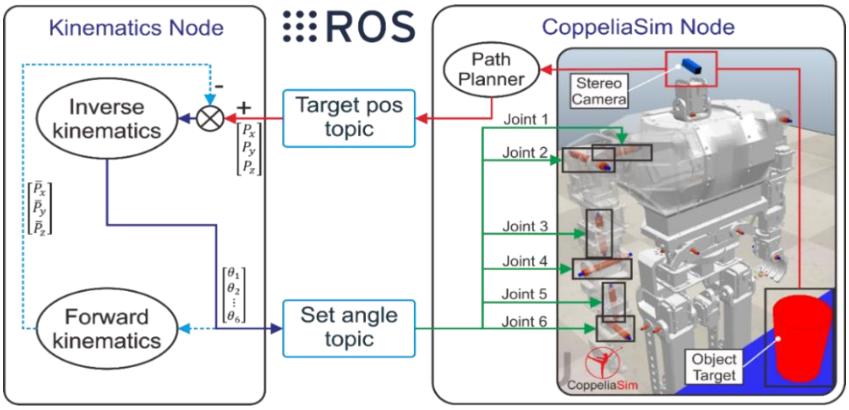


Figure 2. Experimental platform

4.2. Analysis of kinematics model

The kinematics model analysis will be divided into two sections, the forward kinematics analysis and the inverse kinematics analysis. The forward kinematics analysis provides the comparison result of the proposed forward kinematics with the general robotic toolbox. Furthermore, the inverse kinematics analysis provides the results of the proposed inverse kinematics in several experiments.

4.2.1. Forward kinematics analysis

Since the inverse kinematics are based on the DLS method that uses the iterated local search algorithm, the forward kinematics is needed to calculate the current end-effector position of the robot $\vec{s}$ that is used feedback for inverse kinematics. The input of forward kinematics is the joint angle from the inverse kinematics calculation. The result of forward kinematics $\vec{s}$ is used to obtain the error position $\vec{e}$ as explained in (8) by subtracting that with the target position of the end-effector. As shown in Table 1, several experiments were performed to validate the success of the forward kinematics model by comparing the results with the robotic toolbox by Peter Corke in MATLAB [31]. With several randomized set angles as input, the proposed forward kinematics model successfully calculates the end-effector position in the cartesian space (x, y, and z) without error.

4.2.2. Inverse kinematics analysis

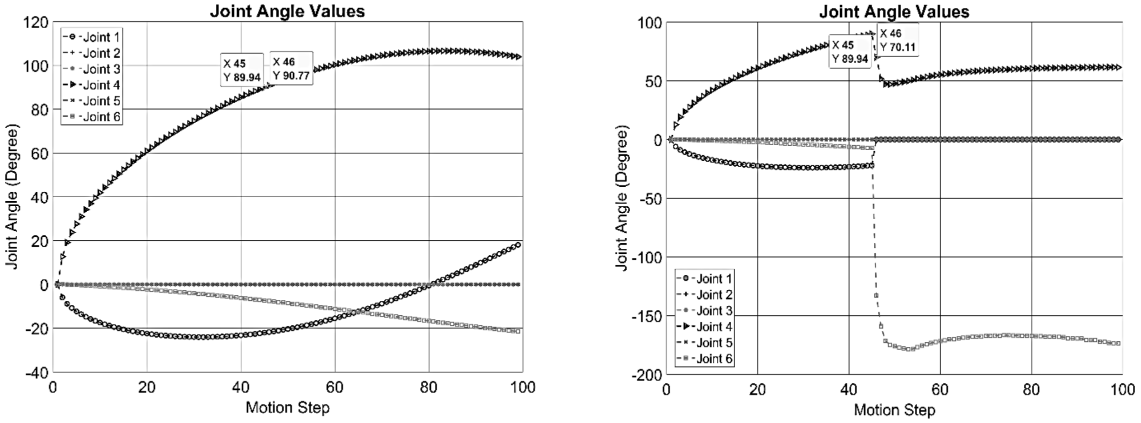
To analyze the inverse kinematics model, we compare the inverse kinematics based on the original DLS with the proposed inverse kinematics using the I-DLS. The simulations were conducted to get a comparative analysis by performing a motion to reach a target position. The first experiment examines the capability of the joint limits method. As shown in Figure 3, an example experiment aims to test the joint limits technique, performed by limiting the joint-4, the minimum angle limit θ_{min} to -90° and the maximum angle limit θ_{max} to 90°. Figure 3(a) shows the angle values without joint limits, which after the 45th step, the joint-4 angle value exceeds 90°. Whereas in Figure 3(b), after the 45th step, the joint-4 angle values with joint limits do not exceed 90° and decrease. However, the robot cannot reach the target position because of the workspace area reduction.

Further experiments are performed to validate the capability of the joint limits technique. As shown in Table 2, The experiments to test the joint limits by setting the minimum angle θ_{min} and maximum angle θ_{max} based on the mechanical design of the robot used in joint 2, joint 4, and joint 6. The red-colored value is the joint angle without the joint limits using the original DLS. The blue-colored value is the joint angle with the joint limits using the proposed DLS or improved DLS. The proposed DLS success performs joint limitation based on the minimum θ_{min} and the maximum θ_{max} angle.

The following experiment was performed to analyze the clamps method to reduce the calculation iteration of the DLS solution. The clamps method works by creating an alternative target position from the actual target position. The experiment works by setting the target position out of the workspace area, as shown in Figure 4(a).

Table 1. The experiment of the forward kinematics model

No	Set Angles (°)	FK Result (x, y, z)	Robotic Toolbox (x, y, z)	Error (x, y, z)
1	{85, 55, 22, 23, 71, 16}	{-4.49, 15.72, 27.88}	{-4.49, 15.72, 27.88}	{0, 0, 0}
2	{34, 60, 86, 48, 27, 80}	{1.21, -0.31, 25.97}	{1.21, -0.31, 25.97}	{0, 0, 0}
3	{78, 65, 77, 46, 58, 21}	{-1.42, 0.86, 30}	{-1.42, 0.86, 30}	{0, 0, 0}
4	{46, -54, -36, -50, 18, -24}	{22.77, 7.56, -16.55}	{22.77, 7.56, -16.55}	{0, 0, 0}
5	{-43, 3, 34, -47, 86, 14}	{13.48, -26.86, -4.41}	{13.48, -26.86, -4.41}	{0, 0, 0}



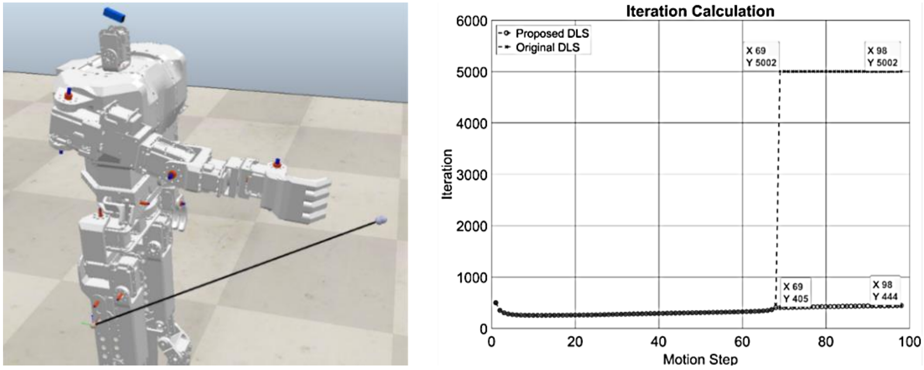
(a) (b)

Figure 3. Joint angle values comparative (a) without joint limits and (b) with joint limits

Table 2. Joint limits experimental table

No	Joint angle (°)	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
	θ_{min}	-179	0	-179	-90	-179	0
	θ_{max}	180	180	180	90	180	150
1	Target Position (x, y, z)			(12.7, 17.7, 0)			
	Original DLS	6.66	0	0	90.49	0	26.21
	Proposed I-DLS	0	0	0	87	0	36.59
2	Target Position (x, y, z)			(32.5, 0, -2.8)			
	Original DLS	0	-4.92	0	0	0	0
	Proposed I-DLS	0	0	0	0	0	0
3	Target Position (x, y, z)			(28.2, 0, -1.6)			
	Original DLS	30.41	-3.63	0.21	-56.34	0.06	-21.5
	Proposed I-DLS	31.38	0	5.83	-63.34	0.19	0

Using path planning as shown in Figures 4(a) and 4(b) shows the calculation of iterations of each step in 98 total steps. Since the DLS method uses an iterated local search algorithm, the maximum computational iteration is limited to 5,000. Comparing the original DLS with the proposed DLS with clamps, when the target position is unreachable in the 69th step, the original DLS fails to find the solution and reaches infinite iteration if the maximum iteration is not defined. On the contrary, though the target position is unreachable, the proposed DLS with clamps successfully finds the solution but increases the iteration by 10.95% between the original target and the clamping target position.



(a) (b)

Figure 4. Path planning: (a) clamping experiment by setting target position unreachable and (b) iteration comparison

4.3. Reach movement in the simulation

The last experiment is to perform a motion to reach the target object in the simulation. We use a cup as a target object and obtain the coordinate position using a stereo camera attached to the head of the robot. The OMPL path planner creates a motion path with obstacle avoidance between the initial end-effector and target positions detected by a stereo camera, as shown in Figure 5. The OMPL path planning makes 98 movement steps where the kinematics node in ROS2 will calculate the value of all joints based on the target position for each step. The comparative data between the end-effector position and the target position can be seen in Figure 6.

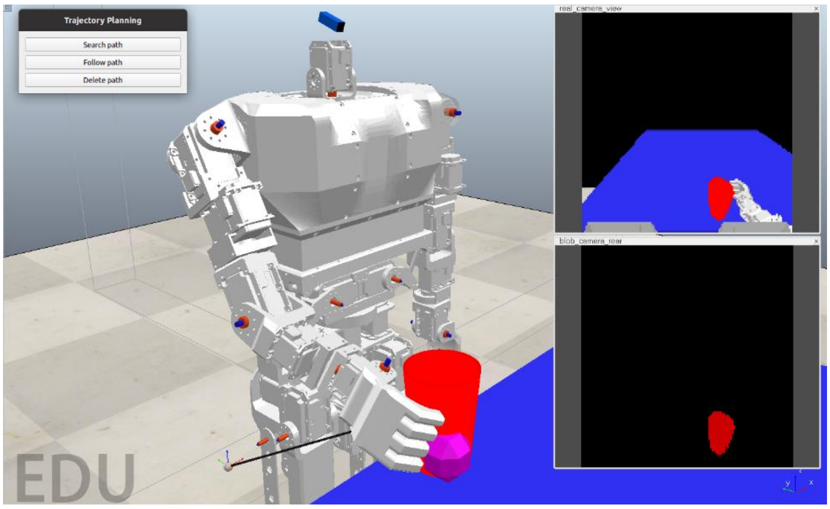
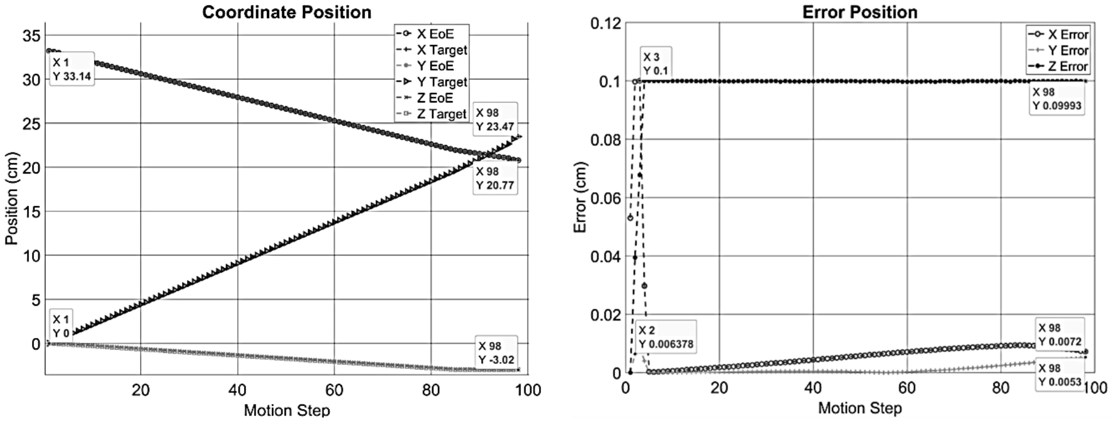


Figure 5. Reach target experiment in simulation

Figure 6(a) shows the comparative data between the end-effector position calculated from the kinematics node and the target position. Figure 6(b) is the error between the end-effector and the target positions on each axis. The obtained maximum error is 0.1 cm, and the average error is 0.0081, 0.0013, and 0.0979 cm on x, y, and z, respectively. The additional data to validate the experiment of reaching the unreachable target by comparing the original DLS and proposed I-DLS can be shown in Table 3. Even though both methods could not reach the target because it was unreachable, the proposed method could maintain a lower mean error to minimize the calculation time. As a result, the average error position improvement percentage between the original DLS and I-DLS on x, y, and z are 11.860%, 15.789%, and 9.800%, respectively.



(a) (b)

Figure 6. Reach target experiment data (a) comparative data position and (b) error data position

Table 3. Reach target experiment

No	Method	Target Position (cm)			Average Error Position in cm (%)		
		x	y	z	x	y	z
1	DLS	20.34	35.24	-2.81	1.980 (9.734)	4.996 (14.178)	0.029 (1.028)
	I-DLS				0.067 (0.329)	0.087 (0.246)	0.009 (0.338)
2	DLS	20.2	32.8	14.83	2.358 (11.673)	5.427 (16.545)	2.304 (15.539)
	I-DLS				0.072 (0.356)	0.081 (0.248)	0.044 (0.297)
3	DLS	19.77	37.73	-6.16	2.705 (13.681)	5.806 (15.388)	0.731 (11.872)
	I-DLS				0.068 (0.344)	0.078 (0.207)	0.017 (0.283)
4	DLS	19.54	37.81	10.12	2.682 (13.725	6.796 (17.973)	1.204 (11.902)
	I-DLS				0.067 (0.343)	0.087 (0.229)	0.023 (0.225)

5. CONCLUSION

The kinematics model has been successfully built in the ROS2 platform for a 6-DOF robotic arm T-FLoW humanoid robot to perform a reach objects motion for grasping mechanism. We test the forward kinematics to validate its success by comparing it with the robotic toolbox by Peter Corke. We also test the inverse kinematics by comparing the proposed I-DLS combined with joint limit and clamps with the original DLS. The result of the forward kinematics successfully calculated the end-effector position with 0 error. The result of inverse kinematics using the improved DLS is that the joint limits technique can successfully limit the joint angle based on the defined value of the minimum and the maximum angle to avoid mechanical limitations. When the target is unreachable, the original DLS fails to find the solution and obtain infinite iteration. However, the proposed I-DLS with clamps successfully finds the solution despite increasing the iteration by 10.95%. The experiment results to reach an object in the simulation show that the proposed kinematics model successfully follows a motion path with a maximum error is 0.1 cm. The average error of each axis (x, y, and z) is 0.0081, 0.0013, and 0.0979 cm. However, the disadvantage of using the joint limit technique is that it can reduce the workspace area of the robot due to the limited movement because of the angle restrictions. Thus, using the joint limits technique requires additional calculation iterations. Further research is to develop the real robot arms, the dynamics model of the robot, obtain the pose of the hand, and develop the grasping movement algorithms.

ACKNOWLEDGEMENTS

This research was partially funded by Ministry of Education, Culture, Research, and Technology Indonesia under the Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) program, contract number 082/E4.1/AK.04.PT/2021, with the title is "*B-FLoW, bipedal humanoid sebagai platform kolaborasi manusia dan robot dalam tata gerha*". This research also supported by the Robotics and Intelligent Systems Center (RoISC), Politeknik Elektronika Negeri Surabaya.

REFERENCES

[1] G. Shanmugasundar, R. Sivaramakrishnan, S. Meganathan, and S. Balasubramani, “Structural optimization of an five degrees of freedom (T-3R-T) robot manipultor using finite element analysis,” *Materials Today: Proceedings*, vol. 16, pp. 1325–1332, 2019, doi: 10.1016/j.matpr.2019.05.231.

[2] A. Zeng *et al.*, “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching,” *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022, doi: 10.1177/0278364919868017.

[3] R. D. Pristovani, D. Sanggar, and P. Dadet, “Implementation and design of new low-cost foot pressure sensor module using piezoelectric sensor in T-FLoW humanoid robot,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 203–214, Feb. 2019, doi: 10.11591/ijece.v9i1.pp203-214.

[4] W. P. Sari, R. Sanggar Dewanto, and D. Pramadihanto, “Kinematic and dynamic modelling of ‘T-FLoW’ humanoid robot for ascending and descending stair,” in *2019 International Electronics Symposium (IES)*, Sep. 2019, pp. 82–87, doi: 10.1109/ELECSYM.2019.8901633.

[5] R. N. Jazar, *Theory of applied robotics: kinematics, dynamics, and control*. Springer, 2007.

[6] L. Yanto, R. S. Dewanto, D. Pramadihanto, and E. H. Binugroho, “Teen-size humanoid ‘FLoW’ complete analytical kinematics,” *EMITTER International Journal of Engineering Technology*, vol. 5, no. 2, pp. 298–311, 2018, doi: 10.24003/emitter.v5i2.233.

[7] S. Kucuk and Z. Bingul, “Inverse kinematics solutions for industrial robot manipulators with offset wrists,” *Applied Mathematical Modelling*, vol. 38, no. 7–8, pp. 1983–1999, Apr. 2014, doi: 10.1016/j.apm.2013.10.014.

[8] Z. Fu, W. Yang, and Z. Yang, “Solution of inverse kinematics for 6r robot manipulators with offset wrist based on geometric algebra,” *Journal of Mechanisms and Robotics*, vol. 5, no. 3, Aug. 2013, doi: 10.1115/1.4024239.

[9] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, “Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1376–1386, Apr. 2020, doi: 10.11591/ijece.v10i2.pp1376-1386.

[10] O. Hock and J. Šedo, “Forward and inverse kinematics using pseudoinverse and transposition method for robotic arm DOBOT,” in *Kinematics*, InTech, 2017.

[11] J. Wan, H. Wu, R. Ma, and L. Zhang, “A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method,” *Journal of Mechanical Science and Technology*, vol. 32, no. 3, pp. 1367–1378, Mar. 2018, doi: 10.1007/s12206-018-0240-7.

[12] M. Khairudin, Z. Mohamed, A. R. Husain, and M. A. Ahmad, “Dynamic modelling and characterisation of a two-link flexible robot manipulator,” *Journal of Low Frequency Noise, Vibration and Active Control*, vol. 29, no. 3, pp. 207–219, Sep. 2010, doi: 10.1260/0263-0923.29.3.207.

[13] C. Sun, H. Gao, W. He, and Y. Yu, “Fuzzy neural network control of a flexible robotic manipulator using assumed mode method,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5214–5227, Nov. 2018, doi: 10.1109/TNNLS.2017.2743103.

[14] J. J. Jui and M. A. Ahmad, “A hybrid metaheuristic algorithm for identification of continuous-time Hammerstein systems,” *Applied Mathematical Modelling*, vol. 95, pp. 339–360, Jul. 2021, doi: 10.1016/j.apm.2021.01.023.

[15] X. Shi, Y. Guo, X. Chen, Z. Chen, and Z. Yang, “Kinematics and singularity analysis of a 7-DOF redundant manipulator,” *Sensors*, vol. 21, no. 21, Oct. 2021, doi: 10.3390/s21217257.

[16] A. Almarkhi and A. Maciejewski, “Singularity analysis for redundant manipulators of arbitrary kinematic structure,” in *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, 2019, pp. 42–49, doi: 10.5220/0007833100420049.

[17] S. Chiaverini, B. Siciliano, and O. Egeland, “Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator,” *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 123–134, Jun. 1994, doi: 10.1109/87.294335.

[18] Y. Nakamura and H. Hanafusa, “Inverse kinematic solutions with singularity robustness for robot manipulator control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, Sep. 1986, doi: 10.1115/1.3143764.

[19] C. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, Jan. 1986, doi: 10.1109/TSMC.1986.289285.

[20] P. Liu, D. Yu, and R. Li, “Research on singular robustness algorithm of robot inverse kinematics based on dynamic damping coefficient,” in *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, Oct. 2017, pp. 204–209, doi: 10.1109/ITOEC.2017.8122412.

[21] M. Sekiguchi and N. Takesue, “Fast and robust numerical method for inverse kinematics with prioritized multiple targets for redundant robots,” *Advanced Robotics*, vol. 34, no. 16, pp. 1068–1078, Aug. 2020, doi: 10.1080/01691864.2020.1780151.

[22] M. Na, B. Yang, and P. Jia, “Improved damped least squares solution with joint limits, joint weights and comfortable criteria for controlling human-like figures,” in *2008 IEEE Conference on Robotics, Automation and Mechatronics*, Sep. 2008, pp. 1090–1095, doi: 10.1109/RAMECH.2008.4681441.

[23] S. R. Buss and J.-S. Kim, “Selectively damped least squares for inverse kinematics,” *Journal of Graphics Tools*, vol. 10, no. 3, pp. 37–49, Jan. 2005, doi: 10.1080/2151237X.2005.10129202.

[24] S. Wen, Z. Ma, S. Wen, Y. Zhao, and J. Yao, “The study of NAO robot arm based on direct kinematics by using D-H method,” in *2014 UKACC International Conference on Control (CONTROL)*, 2014, pp. 515–518, doi: 10.1109/CONTROL.2014.6915193.

[25] D. Whitney, “Resolved motion rate control of manipulators and human prostheses,” *IEEE Transactions on Man Machine Systems*, vol. 10, no. 2, pp. 47–53, Jun. 1969, doi: 10.1109/TMMS.1969.299896.

[26] O. Egeland, J. R. Sagli, I. Spangelo, and S. Chiaverini, “A damped least-squares solution to redundancy resolution,” in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 945–950, doi: 10.1109/ROBOT.1991.131710.

[27] E. Rohmer, S. P. N. Singh, and M. Freese, “V-REP: A versatile and scalable robot simulation framework,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 1321–1326, doi: 10.1109/IROS.2013.6696520.

[28] Vortex Studio, “Real-time simulation software,” *cm-labs*. https://www.cm-labs.com/vortex-studio/ (accessed Dec. 02, 2021).

[29] Y. Maruyama, S. Kato, and T. Azumi, “Exploring the performance of ROS2,” in *Proceedings of the 13th International Conference on Embedded Software*, Oct. 2016, pp. 1–10, doi: 10.1145/2968478.2968502.

[30] I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robotics and Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012, doi: 10.1109/MRA.2012.2205651.

[31] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011.

BIOGRAPHIES OF AUTHORS

Muhammad Ramadhan Hadi Setyawan received the B.Eng. degree in Computer Engineering from Politeknik Elektronika Negeri Surabaya, Indonesia, in 2021. Currently, he is a Research Assistant at the Robotics and Intelligent Systems Center (RoISC) in the humanoid robot division. His research interests include robotics, kinematics, control, and embedded systems. He can be contacted at email: rmstwn@gmail.com.

Raden Sanggar Dewanto received the B.Eng. and the M.Eng. degree in Electronics Engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 1995 and 2004. He received a Ph.D. degree in Control System Engineering from Newcastle University, United Kingdom, in 2014. He is currently a Lecturer in Mechatronics Engineering Department, Politeknik Elektronika Negeri Surabaya. He is a supervisor member of the Robotics and Intelligent Systems Center (RoISC). His research interests include the robotics system, kinematics and dynamics system, control system, mechanical design and analysis, and the development of Micro-electromechanical systems (MEMS). He can be contacted at email: sanggar@pens.ac.id.

Bayu Sandi Marta received the B.Eng. degree in Computer Engineering from Politeknik Elektronika Negeri Surabaya, Indonesia, in 2011. He received the M.Eng. degree in Electronics Engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 2014. He is currently a Lecturer in Computer Engineering Department, Politeknik Elektronika Negeri Surabaya. He is a supervisor member of the Robotics and Intelligent Systems Center (RoISC). His research interests include robotics, control, embedded systems, and electronics development. He can be contacted at email: bayu@pens.ac.id.

Eko Henfri Binugroho received the B.Eng. degree in Electronics Engineering from Politeknik Elektronika Negeri Surabaya, Indonesia, in 2003. He received his M.Sc. degree in Intelligent Mechanical System, School of Mechanical Engineering, Pusan National University, Korea, in 2009. He is currently a Lecturer in Mechatronics Engineering Department, Politeknik Elektronika Negeri Surabaya. He is a supervisor member of the Robotics and Intelligent Systems Center (RoISC). His research interests include mechatronics, robotics, control, embedded systems, and electronics development. He can be contacted at email: sragen@pens.ac.id.

Dadet Pramadihanto is a professor in robotics and computer vision at the Politeknik Elektronika Negeri Surabaya. He received his B.Eng. degree in Control and Instrumentation Engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 1987. He received his M.Eng. and the Ph.D. degree in Computer vision from Osaka University, Japan, in 1997 and 2003. He is currently a Lecturer in the Computer Engineering Department and the head of the Robotics and Intelligent Systems Center (RoISC). His research areas of interest include the development of humanoid robots, computer vision in robotics, underwater and aerial robots, and cyber-physical systems. He can be contacted at email: dadet@pens.ac.id.