

Article

Design and Implementation of a Graphic Simulator for Calculating the Inverse Kinematics of a Redundant Planar Manipulator Robot

Claudio Urrea *  and Daniel Saa

Department of Electrical Engineering, Universidad de Santiago de Chile, Av. Ecuador 3519, Estación Central, Santiago 9170124, Chile; daniel.saa@usach.cl

* Correspondence: claudio.urrea@usach.cl; Tel.: +56-2-27180324

Received: 2 August 2020; Accepted: 24 September 2020; Published: 27 September 2020

Abstract: In this paper, a graphics simulator that allows for characterizing the kinematic and dynamic behavior of redundant planar manipulator robots is presented. This graphics simulator is implemented using the Solidworks software and the SimMechanics Toolbox of MATLAB/Simulink. To calculate the inverse kinematics of this type of robot, an algorithm based on the probabilistic method called Simulated Annealing is proposed. By means of this method, it is possible to obtain, among many possibilities, the best solution for inverse kinematics. Without losing generality, the performance of metaheuristic algorithm is tested in a 6-DoF (Degrees of Freedom) virtual robot. The Cartesian coordinates (x,y) of the end effector of the robot under study can be accessed through a graphic interface, thereby automatically calculating its inverse kinematics, and yielding the solution set with the position adopted by each joint for each coordinate entered. Dynamic equations are obtained from the Lagrange–Euler formulation. To generate the joint trajectories, an interpolation method with a third order polynomial is used. The effectiveness of the developed methodologies is verified through computational simulations of a virtual robot.

Keywords: simulation; redundant manipulator robots; kinematics; dynamics; simulated annealing; solidworks

1. Introduction

Nowadays, the use of manipulator robots is widely extended in the industry because these robots carry out tasks that pose risks to humans, improve the quality of the products manufactured, and increase productivity. Nevertheless, robotics is not limited to the industry, but has also expanded to other fields such as medicine (robotic implants or remote surgical intervention systems), agriculture (sheep shearing or pork cutting), space (planetary surface exploration), and geological and military research, among others [1–3].

Manipulator robots can be classified into serial and parallel based on their structure. The links of a serial manipulator are normally assembled together by means of rotational or prismatic joints that form an open kinematic chain whose structure is similar to a human arm. This configuration allows for assembling a clamp or any other tools that can be moved to any position within the robot's workspace [4].

The industrial robots available in the global markets are not redundant, but they do have the degrees of freedom necessary for performing the tasks for which they were designed [5]. However, if a failure occurs in the actuator of an industrial robot, without a safety backup, its respective joint blocks, and the robot loses the mobility of that joint. This may lead to situations that imply risks to the operators, difficulties for the users, interruption of the mining or industrial process, and economic losses, among other consequences [6]. Therefore, interest has arisen in the study of the behavior of

redundant manipulator robots, owing to a design that gives them more degrees of freedom than necessary for performing a specific task [7].

In general, the kinematic and dynamic study of robotic systems can be conducted in a didactic way, using graphic simulators. The kinematic problem can be solved through the Geometric Method, but, in practice, this method is not widely used because of the complexity of applying it to redundant manipulator robots, which has led to the development of several alternative methods. Kinematics is divided into direct and inverse kinematics (IK) that can be derived based on the standard Denavit-Hartenberg (D-H) model, which describes the geometry of the robot. Besides the almost universally used the D-H parameters [8–11], other methods such Algebra and Lie Groups are also useful to obtain the kinematics of the system [12].

The inverse kinematics chain of redundant manipulator robots is much more difficult to obtain. Its solution is fundamental to deal with problems such as route planning, visually guided movement, grabbing objects, etc. [13]. Currently, the conventional methods for solving inverse kinematics are divided into two categories: analytical (geometric-algebraic) and numerical (iterative algorithms) [14]. Some recent articles show that analytical solutions have only been found for some specific structures of redundant manipulator robots. For instance, Qian et al. [15] present a close loop intuitive solution for a space manipulator robot with 7-DoF and a PYR-P-PYR (Pitch-Yaw-Roll) structured based on the blockage of the first joint P. Teja and Shah [16] present the Inverse Coordinates Method for a special S-RRR (Spherical-Rotary) configuration, which is based on the inversion of the base and the end effector of the manipulator robot. Additionally, Zhao et al. [17] employ analytical, geometric and algebraic methods for a 6R industrial robot, which are combined with the Paden–Kahan subproblem and the matrix theory.

Furthermore, since there are no possible analytical solutions for the inverse kinematics of some robots, some methods are now under discussion, for example, the Iterative Method, which is based on numerical methods [18]. Huang and Nie [19] present a numerical solution, which is achieved through the application of the Sequential Recovery Method to a robot with and RPY (Roll-Pitch-Yaw) configuration based on the modification of the Newton-Raphson Method. Another alternative to obtain the inverse kinematics is using the J^T transpose or the J^{-1} inverse of the Jacobian matrix [20]. The usefulness of the J^T Jacobian is demonstrated by Besset and Taylor [21]. In addition, Farzan and DeSouza [22] show a solution based on estimates of the inverse Jacobian matrix.

Recently, as some traditional methods have become obsolete for some manipulator robots, researchers have started the development or artificial intelligence techniques for calculating inverse kinematics. Among these techniques or meta-heuristic algorithms, artificial neural networks (ANNs), genetic algorithms (GAs), particle swarm optimization (PSO), artificial bee colony (ABC), differential evolution (DE) and simulated annealing (SA) stand out. Köker and Cakar [23] propose a hybrid smart solution for IK, which includes ANN, GA and SA and is applied to a 4-DoF robot. The weak point of this study is that optimization processes based on SA have a prolonged execution time due to the difficulties to select suitable cooling parameters. López et al. [24] present different meta-heuristic algorithms to solve the inverse kinematics of a KUKA Youbot mobile robot with 9-DoF. From simulations and real experimental results, it is demonstrated the superiority of the DE algorithm over other algorithms due to its computational cost, reduced joint movement and minimum error between the end effector and the target point. Additionally, López et al. [24] demonstrate that the hybrid biogeography-based optimization (HBBO) approach generates good results but with high computational costs. In contrast, cuckoo search (CS) and PSO report poor results and are not suggested as methods for obtaining inverse kinematics. The flaw of this study regarding the DE algorithm, is the number of iterations used, which slows down the search for a solution. Dereli and Köker [25] compare the position error and calculation time of two heuristic techniques called PSO and ABC. The simulations show that both algorithms successfully find the angles of the joints of a redundant manipulator robot with 7-DoF. Despite its good performance, the long steps of the search process and the excess of parameters are considered the disadvantages of this method. The algorithms above suffer a common problem: the time

consumed during the iteration process. In this regard, Zhou et al. [26] propose a smart algorithm based on extreme machine learning (ELM) and sequential mutation genetic algorithm (SGA) to improve the efficiency of the Stanford MT-ARM manipulator robot of 6-DoF. Nevertheless, this last study sacrificed the accuracy of the preliminary solution of inverse kinematics.

Currently, several researchers have implemented simulation systems them in robots [24,26,27] because they allow for manipulating and testing the control system of automated robot within a virtual environment. For example, Kereluk and Emani [28] present an integrated design and simulation environment that offers control over the hardware of a reconfigurable manipulator robot with 18-DoF denominated MARS. In this environment, the user can select or design the configuration of the manipulator robot, simulate its operation in a virtual reality environment and operate the physical manipulator robot. The simulation results have proved to be very similar to experimental results. López et al. [24] developed a simulator on MATLAB software to test the performance of different meta-heuristic algorithms for solving the inverse kinematics of manipulator robots. Zhou et al. [26] also used it to analyze, specifically, time efficiency during the training stage of NNs and ELM, as well as for the optimization and computational process.

The contribution of this study is the development (design and implementation) of a graphic simulator using the Solidworks software and the SimMechanics Toolbox from MATLAB/Simulink. This program is commanded from a graphic interface that is created with the MATLAB tool GUIDE. GUIDE allows for characterizing kinematic and dynamic behavior; additionally, tests can be conducted in the control system of a planar manipulator robot with 6-DoF within a virtual environment. An innovative aspect of this work is that the special configuration of the 6-DoF manipulator robot, which includes rotational and prismatic (linear) joints. Most recent research employs robot structures with rotational joints and omit linear ones. Exploring new types of configurations allow for improving the skills of the robot, broadening its workspace, increasing its accuracy, or improving its energy consumption [28]. The SA algorithm will be applied to the manipulator robot without adding any other meta-heuristic algorithm. Additionally, before using SA, an algorithm to help search for the initial solution (end effector initial position) will be implemented to enable the end effector to be as close as possible to the set point and will contribute to a faster convergence of the initial solution.

2. Description of the 6-Dof Redundant Planar Manipulator Robot

This work considers a novel 6-DoF redundant planar manipulator robot with three rotary (R) and three prismatic (P) joints arranged in a RPRPRP configuration. If we draw an analogy with a human arm, this robot is composed of three parts: an arm, a first forearm and a second forearm. Each part is made up of two links and two types of joints (rotary and prismatic).

This redundant planar manipulator robot constitutes an open kinematic chain that originates from the assembling of the rotary joint, specifically between its fixed base and the arm; the arm and the first forearm, and, finally, the first and second forearm. Additionally, the prismatic joints connect the links as follows: first and secondary link, third and fourth link, and fifth and sixth link.

3. Kinematic Analysis

This section is concerned with the development of the mathematical relationships that analytically describe the spatial motion of the redundant planar manipulator robot under study, without considering the causes of the same. The equations relate the position and orientation of the end part of the robot with the coordinates of the robot joints. This is carried out using a reference system arbitrarily chosen, but that is usually placed on the base of the redundant planar manipulator robot.

Forward Kinematics of the 6-Dof Redundant Planar Manipulator Robot

The D-H method is used to characterize the kinematic behavior of the redundant planar manipulator robot. Taking the axes assigned in the geometrical configuration shown in Figure 1, it is possible to obtain the robot joint parameters, which are presented in Table 1.

Table 1. Joint parameters of the 6-DoF redundant planar manipulator robot.

Joint	θ_i	d_i	a_i	α_i
1	$\theta_1^* - 90^\circ$	L_0	0	-90°
2	0°	$d_1^* + L_1 + L_2$	0	90°
3	θ_2^*	0	0	-90°
4	0°	$d_2^* + L_1 + L_2$	0	90°
5	θ_3^*	0	0	-90°
6	0°	$d_3^* + L_1 + L_2$	0	0°

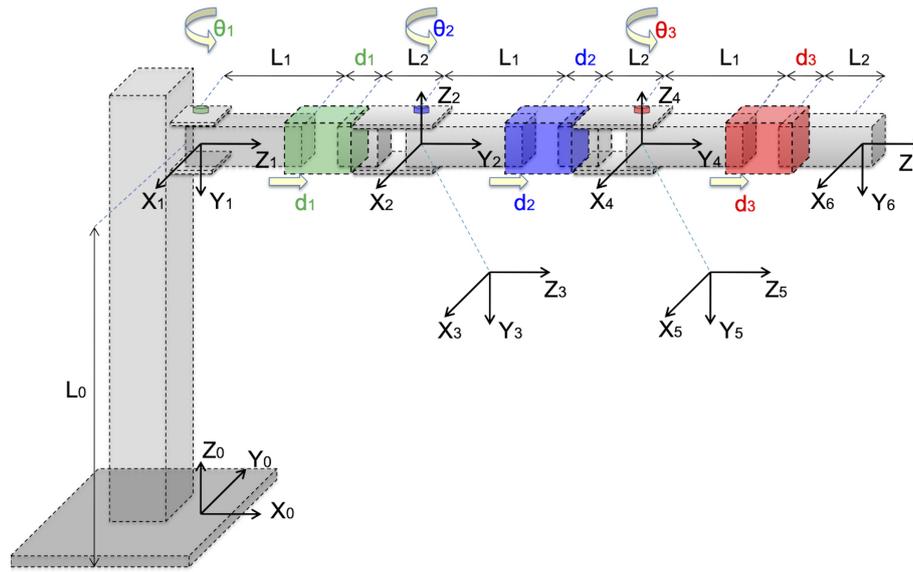


Figure 1. Geometrical configuration of the 6-DoF redundant planar manipulator robot and assignment of the D-H coordinate system (perspective view).

The homogeneous transformation matrix of Equation (1) is completely defined based on joint parameters:

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

where: $i = 1, \dots, 6$.

According to the D-H convention, the product of each transformation matrix that link the coordinate system 0_{i-1} is calculated with the immediately after 0_i , and the 4x4 T matrix is obtained, which represents the pose (position and orientation) of the last link from the base system.

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 \quad (2)$$

Then, from T the following analytical representation of the position of the last link in the base system (coordinates origin) is obtained:

$$x = r_1 \cos(\theta_1) + r_2 \cos(\theta_1 + \theta_2) + r_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (3)$$

$$y = r_1 \sin(\theta_1) + r_2 \sin(\theta_1 + \theta_2) + r_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (4)$$

where: $r_1 = d_1 + L_1 + L_2$; $r_2 = d_2 + L_1 + L_2$; $r_3 = d_3 + L_1 + L_2$

4. Direct Dynamics of the 6-DoF Redundant Planar Manipulator Robot

To obtain the dynamic equations of the 6-DoF redundant planar manipulator robot, the L-E Method is applied to the robot shown in Figure 2. Taking into account the prismatic and rotational motions, the generalized coordinate vector is $\mathbf{q} = [\theta_1 \ d_1 \ \theta_2 \ d_2 \ \theta_3 \ d_3]$.

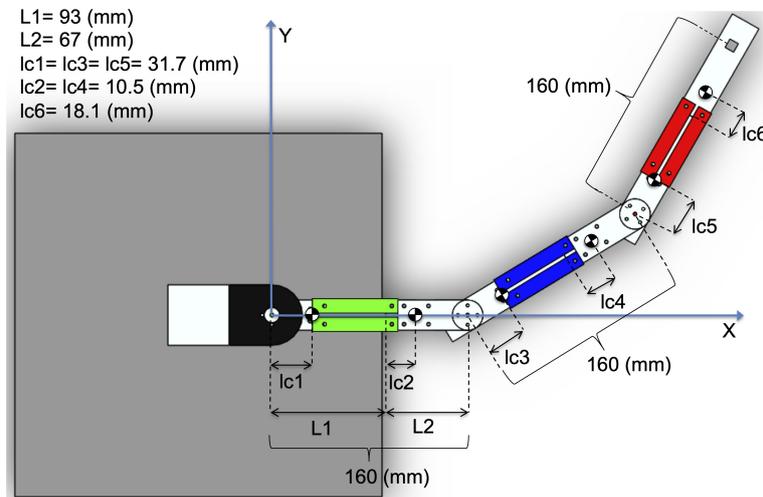


Figure 2. Resting location and position of the mass centers of the 6 links.

where: l_{ci} is the length of the mass center of the link $i, i = 1, 2, 3, 4, 5, 6$.

The dynamic model of the manipulator of 6-DoF, composed of fixed links interconnected by means of joints, can be described in its compact form (matrix), as shown in Equation (5) [29]:

$$\boldsymbol{\tau} - \boldsymbol{\tau}_0 = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{F}(\mathbf{q}) \quad (5)$$

where: $\boldsymbol{\tau}$ is the (6×1) vector of the generalized forces applied by the actuators on the joints; $\boldsymbol{\tau}_0$ is the (6×1) vector of the input unknown disturbances; \mathbf{M} is the (6×6) inertia matrix; \mathbf{C} is the (6×6) centrifugal and Coriolis forces matrix; \mathbf{G} is the (6×1) gravitational forces vector; \mathbf{F} is the (6×1) vector of the frictions, and $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are the vectors of the joint displacement/angle, speed and acceleration joint vectors, respectively.

Since the manipulator works in a horizontal plane, then $\mathbf{G}(\mathbf{q})$ is zero (gravity acts perpendicular to the motion plane). Vectors $\boldsymbol{\tau}_0$ and \mathbf{F} are not considered in the construction of the model. The elements of the matrix $\mathbf{M}(\mathbf{q})$ are presented in Appendix A.

The centrifugal and Coriolis forces matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ can be obtained through the Christoffel coefficients $C_{ijk}(\mathbf{q})$ defined in Equation (6) [30]:

$$C_{ijk}(\mathbf{q}) = \frac{1}{2} \left[\frac{\partial M_{kj}(\mathbf{q})}{\partial q_i} + \frac{\partial M_{ki}(\mathbf{q})}{\partial q_j} - \frac{\partial M_{ij}(\mathbf{q})}{\partial q_k} \right] \quad (6)$$

where: $M_{ij}(\mathbf{q})$ denotes the ij -th element of the inertia matrix $\mathbf{M}(\mathbf{q})$. Thus, the kj -th element of the $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ matrix can be obtained as:

$$\mathbf{c}_{kj}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C_{1jk}(\mathbf{q}) \\ C_{2jk}(\mathbf{q}) \\ \vdots \\ C_{njk}(\mathbf{q}) \end{bmatrix}^T \dot{\mathbf{q}} \quad (7)$$

Then, the resulting centrifugal and Coriolis forces vector is: $\mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$

Using Equations (6) and (7), the centrifugal and Coriolis forces vector $\mathbf{c}(q, \dot{q})$ shown in Equation (8) is obtained. The components of this vector are presented in Appendix A.

$$\mathbf{c}(q, \dot{q}) = [C_{i1}]; i = 1, \dots, 6 \tag{8}$$

5. Simulated Annealing Algorithm

These days, several methods exist to solve the inverse kinematics of the manipulator robots, but, unfortunately, some of them have long calculation times and only provide one restricted solution [15,31]. In this section, a new method is presented for calculating the inverse kinematics of (standard and redundant) manipulator robots of any type and configuration, based on the probabilistic technique denominated Simulated Annealing (SA).

5.1. Description of the Annealing Process

The physical process called annealing is a technique that consists in heating and then slowly cooling a material to alter its physical and chemical properties. The heat allows atoms to increase their energy, thus making them capable of moving from their initial positions (from a local minimum value of energy to a global one). This occurs because a slow cooling gives atoms higher probabilities of recrystallizing into configurations with lower energy than the initial one. Based on this idea, the SA algorithm was created, which allows conducting a meta-heuristic search for global optimization problems [23].

5.2. Application of the SA Algorithm to the 6-DoF Redundant Planar Manipulator Robot

The SA basic algorithm starts with an iteration number C , an initial state vector x_i of six components (initial solution), that represents the initial inverse kinematics of the robot, and maintains a calculation of a parameter T , commonly called temperature, which is initially high and then is progressively reduced to a value T_f close to zero during the algorithm execution [23]. The configuration of the state and temperature is typically a solution for the function to be optimized. This paper considers an initial temperature $T = 90$, $C = 450$ iterations, and the cooling factor $\alpha = 0.94$.

The solution is perturbed in each iteration of the algorithm to produce a new neighbor solution x_j from its neighborhood [32]. The first heuristic perturbation of the initial state x_i occurs according to Equation (9).

$$x_j = x_i(1 + n_1c_1 - n_2) \tag{9}$$

where: n_1 and n_2 are the environment variables and c_1 is a random number evenly distributed over the (0,1) interval.

Subsequently, it is verified that the value of each component of the x_j vector calculated in Equation (9), which represents the joint position (current inverse kinematics), is within the mechanical limits of the redundant planar manipulator robot. Otherwise, a new calculation of the values for x_j is made using Equation (10).

$$x_j = x_i(1 + n_3c_2) \tag{10}$$

where: n_3 is an environment variable and c_2 is a random scale evenly distributed. This second disturbance of x_i , governed by Equation (10) does not stop until each value of the joint variables of x_j is within the allowed limits.

In addition, the quality of both solutions is assessed using a D target function, which is shown in Equation (11). Then, a new state x_j could be eventually selected from the two solutions, as long as x_j is better quality than x_i .

$$D = d(x_j) - d(x_i) \tag{11}$$

where: $d(x_i)$ is the performance of the initial state, and $d(x_j)$ the performance of the current state, as shown in Figure 3. The calculation of each performance corresponds to the root mean square function shown in Equation (12):

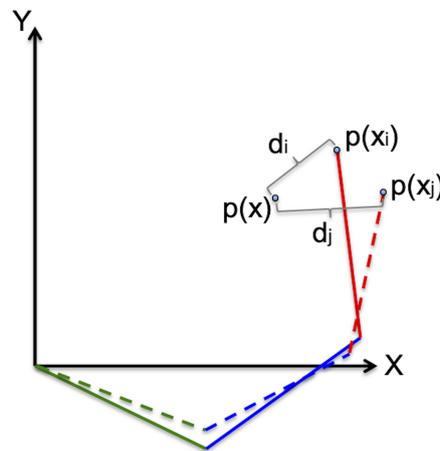


Figure 3. Cost function associated with the x_i and x_j vectors.

$$d(x_k) = \frac{\| p(x) - p(x_k) \|}{2} = \frac{d_k}{2}; k = i, j \tag{12}$$

$$x_k = [q_{1k}, q_{2k}, \dots, q_{nk}]; k = i, j \tag{13}$$

where: d_k is a cost function, $p(x_i)$ is the initial position of the end effector due to the initial joint configuration x_i , $p(x_j)$ is the current position of the redundant planar manipulator robot, x_j is the current state vector of the joints position, $p(x)$ is the set point, and x the target vector that contains the joint variables (inverse kinematics).

To assess the quality of the new solution x_j , one acceptance criterion is assessed in each execution step of the SA algorithm. This criterion is associated with a change of energy ΔE (in this case, $\Delta E = D$), where: if $D \leq 0$, the shift is immediately accepted, while if $D > 0$, the possibility of accepting the disturbing solution depends on the temperature and the solution is finally accepted, but with a probability of [33]:

$$K < \exp\left(-\frac{D}{T}\right) \tag{14}$$

where: K is a random value $\in (0,1)$.

Once the shift is approved, the initial state vector x_i is updated to x_j ($x_i = x_j$) and the algorithm described iterates 449 times (as a maximum) based on Equation (9) until finding a better x_j . Once iterations end, the starting temperature T is reduced following Equation (15), and the procedure is repeated to search for a better x_j until the temperature T is lower than the minimum temperature established, T_f ($T < T_f$) [34]. The latter is the stop criterion of the SA algorithm.

$$T = \alpha T \tag{15}$$

where: α is a cooling factor, which determines the speed at which the system cools down [35].

Finally, considering the real dimensions of the redundant planar manipulator robot, it is established that the inverse kinematics solution be acceptable when the function fitness d_j or the position error between $p(x)$ and $p(x_j)$ is below 1 mm. In that case, x_j is accepted as the final inverse kinematic solution x , knowing that the maximum radius measured from the base to the end of the redundant manipulator robot is 600 mm.

Based on the difference between D performances and the minimization of the cost function, the set of inverse kinematic solutions for manipulator robots is found, as well as whether the current inverse

kinematic solution generates values for the positions of the links), which allows manipulator robots to reach specified positions. The flowchart in Figure 4 summarizes the above described.

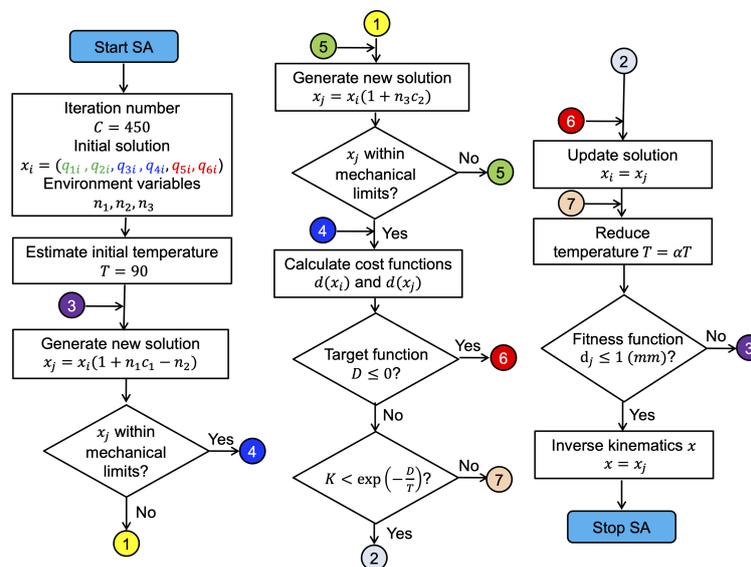


Figure 4. SA algorithm flowchart.

6. Graphic Simulator

This section addresses the development of the redundant planar manipulator robot graphic simulator. Such a simulator is created using the MATLAB/Simulink (including its toolbox SimMechanics™) and Solidworks software, which were chosen due to their 3D modelling and simulation of mechanical system features.

Simmechanics™ Model

Using the SimMechanics Link utility, the CAD assembly developed in SolidWorks is exported to the toolbox SimMechanics™ of MATLAB/Simulink. Then, the SimMechanics toolbox is used to complement and potentiate the basic simulator previously elaborated in SolidWorks. By means of the configuration of blocks representing the components of the redundant planar manipulator robot it is possible to make the robot perform tasks defined by the user, such as controlling its joints and position of its end, among others.

The configuration of some parameters of the blocks is set considering that motion takes place only in the XY plane, and the rest position of the redundant planar manipulator robot occurs when all the links are placed on the X axis, as seen in Figure 2. In home position, it is observed that when each prismatic joint is retracted, the arm and its two forearms are 160 mm long. However, the maximum extension of 200 mm is reached when each prismatic joint is stretched to their limit values, i.e., when each p-joint is extended up to 40 mm the maximum reach of the manipulator is 600 mm long.

Figure 5 shows the circuit in a block diagram imported from the SolidWorks environment, which allows for the dynamic simulation of the redundant manipulator robot. This scheme represents the basic skeleton of the simulator, and was created based on the robot’s real dimensions.

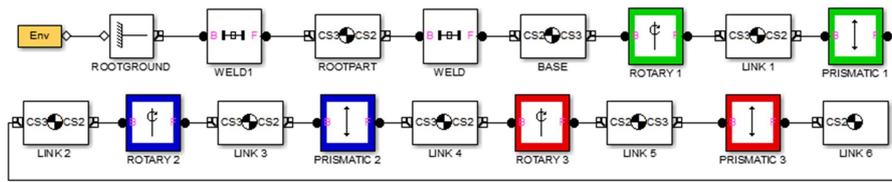


Figure 5. Basic simulation circuit for the 6-DoF redundant planar manipulator robot.

The construction of the manipulator robot considered a structure based on aluminum material. Table 2 presents the main parameters extracted from the blocks modelling each link. These parameters characterize the redundant planar manipulator robot dynamics, with CG as the center of gravity (centroid) measured from the origin of the coordinate system of the respective link.

Table 2. Parameters of the 6-DoF redundant planar manipulator robot links.

Link	Mass (kg)	CG [x; y; z] (mm)
1	0.4822	[31.7; 0; 442]
2	0.6500	[103.5; 0; 443.1]
3	0.4822	[191.7; 0; 442]
4	0.6500	[263.5; 0; 443.1]
5	0.4822	[351.7; 0; 442]
6	0.6453	[431; 0; 0.442]

Among the complementary blocks used in the development of the graphic simulator, those that model actuators and sensors are of great importance, since they generate and record, respectively, all the movements of the redundant planar manipulator robot. In the specific case of actuators, the necessary power signals come from the MATLAB/Simulink environment. Blocks from workspace are used to control each link. These blocks are in charge of reading and storing the diverse joint trajectories programmed by the user.

Figure 6 shows the block diagram and connections necessary for simulating and controlling the motion of the first link. Particularly, control is conducted over the first rotary joint of the redundant planar manipulator robot using a method for interpolation through polynomial. Similar connections and control strategies are used to simulate the movement of the remaining links.

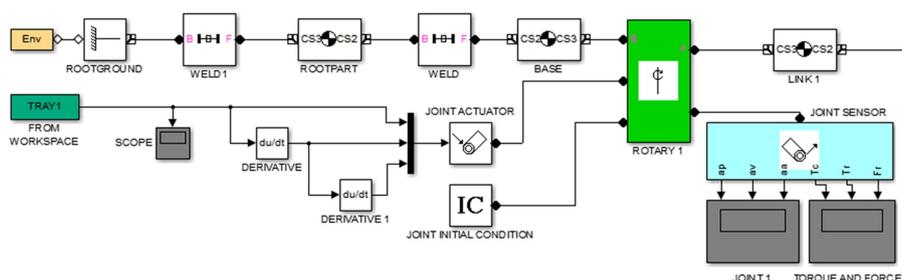


Figure 6. Control loop of the first rotary joint of the 6-DoF redundant planar manipulator robot.

7. Graphic Interface

Below is presented the virtual environment designed to input the parameter values in the blocks that represent the six links that compose the redundant planar manipulator robot. Figure 7 shows, as an example, the configuration window of all the links. Access to this GUIDE in MATLAB is granted through a main window that allows for opening other GUIDE sub-windows. The parameters of the link that can be modified are mass, inertia tensor, position of the center of gravity (CG), and the coordinates of the endpoints of the joints (CS1 and CS2). With this last adjustment, the starting position of each link can be defined and consequently the end effector's. If any box associated with the link is

not configured, the manipulator robot will use the link parameters exhibited by the real manipulator robot in Table 2 by default.

PARAMETER CONFIGURATION

LINK 1				
MASS 1 [Kg]	COORDINATES CG1 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 1 [Kg*m2]
0.4822	[31.7;442;0]	[0;472;0]	[0;467;0]	[2.08 0 0; 0 6.42 0; 0 0 7.46]
LINK 2				
MASS 2 [Kg]	COORDINATES CG2 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 2 [Kg*m2]
LINK 3				
MASS 3 [Kg]	COORDINATES CG3 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 3 [Kg*m2]
LINK 4				
MASS 4 [Kg]	COORDINATES CG4 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 4 [Kg*m2]
LINK 5				
MASS 5 [Kg]	COORDINATES CG5 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 5 [Kg*m2]
LINK 6				
MASS 6 [Kg]	COORDINATES CG6 [mm]	COORDINATES CS1 [mm]	COORDINATES CS2 [mm]	INERTIA TENSOR 6 [Kg*m2]

ADJUST

Figure 7. Configuration of the 6-DoF redundant planar manipulator robot parameters.

An interpolation method based on polynomial equations is used for controlling the joints of the redundant planar manipulator robot. This is achieved using the pchip instruction of the MATLAB, which allows for constructing third-order polynomials that do not cause overshoot during the movement of the joints, thus making it possible to route and move the joints from their initial positions to the desired positions with few oscillations. Figure 8 shows the GUIDE used to program the following joint trajectories.

TRAJECTORY GENERATOR

	ROT.JOINT.1 (from -90° to 90°)	PRISM.JOINT.1 (from 0 to 40mm)	ROT.JOINT.2 (from -90° to 90°)	PRISM.JOINT.2 (from 0 to 40mm)	ROT.JOINT.3 (from -90° to 90°)	PRISM.JOINT.3 (from 0 to 40mm)
	Ang. TI TF TR	Dist. TI TF TR	Ang. TI TF TR	Dist. TI TF TR	Ang. TI TF TR	Dist. TI TF TR
MOVEMENT 1	90 1 2 3	0 1 2 3	-90 1 2 3	40 1 2 3	60 1 2 3	0 1 2 3
MOVEMENT 2	45 4 5 6	10 4 5 6	-45 4 5 6	30 4 5 6	-50 4 5 6	5 4 5 6
MOVEMENT 3	0 7 8 9	20 7 8 9	0 7 8 9	20 7 8 9	80 7 8 9	10 7 8 9
MOVEMENT 4	-45 10 11 12	30 10 11 12	45 10 11 12	10 10 11 12	-90 10 11 12	20 10 11 12
MOVEMENT 5	-90 13 14 15	40 13 14 15	90 13 14 15	0 13 14 15	50 13 14 15	40 13 14 15

CHARGE PLOT EXECUTE STOP

Figure 8. GUIDE for programming trajectories.

This GUIDE allows for entering desired values for the prismatic and rotary joints, indicating also the time of execution (TI), end (TF) and rest (TR) of the movement. The programming limit is five movements per joint. Once the boxes have been filled in using the button “CHARGE”, the values entered are automatically stored in .mat files and in the “from workspace” blocks. Additionally, the trajectories of the programmed joint movements can be visualized through the button “PLOT”, as shown in Figure 9.

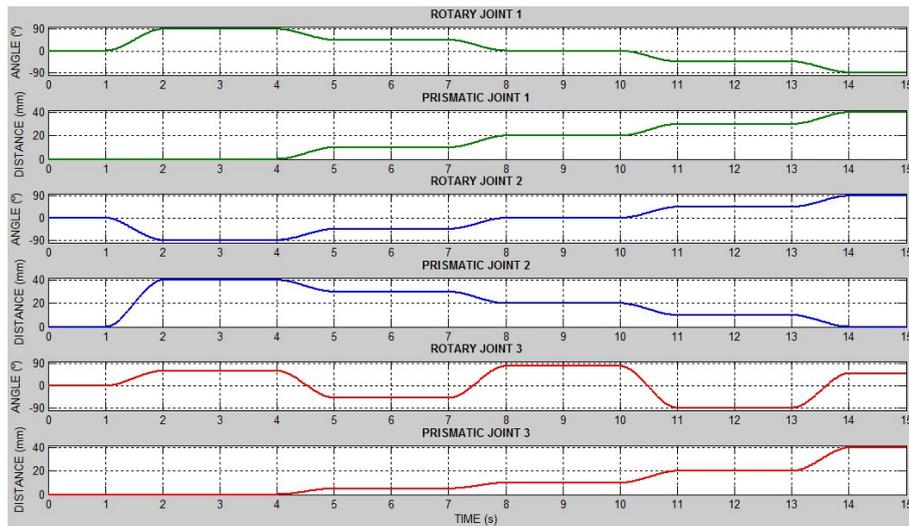


Figure 9. Visualization of the programmed joint trajectories.

The main window of GUIDE also makes it possible to see all the curves resulting from the dynamic performance of the 6-DoF redundant planar manipulator robot, i.e., position, speed, and acceleration of each joint. Specifically, Figure 10 presents the position, speed, and acceleration of the first rotary joint.

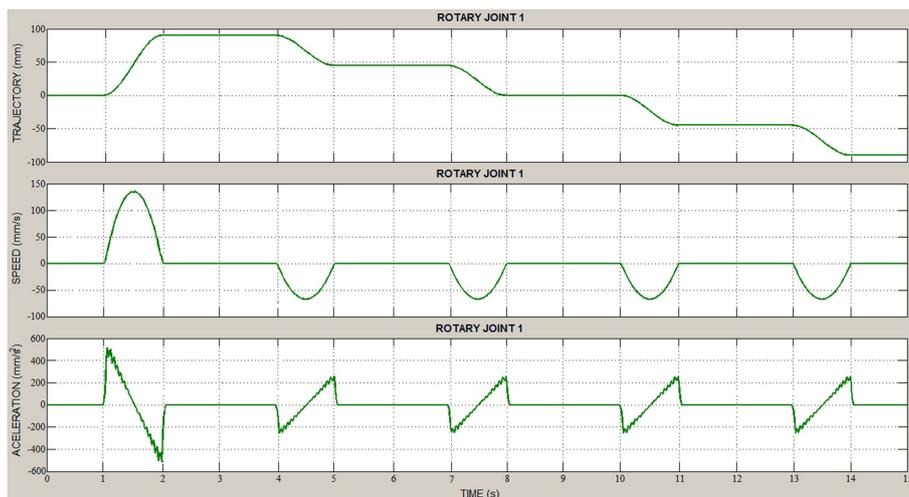


Figure 10. Dynamic performance curves of the first rotary joint. Position, speed, and acceleration.

Figure 11 shows the development of a GUIDE window, which allows for solving the inverse kinematic problem of the redundant planar manipulator robot based on a SA algorithm. Considering the desired location of the end of the redundant planar manipulator robot, it is possible to know the set of values of the joint variables necessary for reaching this location. The joint variables are six and represent the physical location adopted by the three rotary joints and the three prismatic joints. In addition, the position variables are two, because the end of the robot moves only on the XY plane.

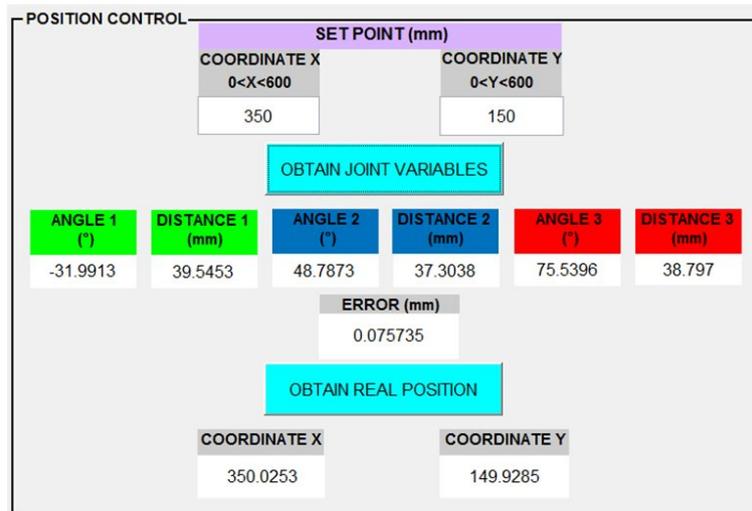


Figure 11. GUIDE window for obtaining the values of the joint variables.

When the coordinates (x, y) are input in the GUIDE window, the workspace allowed to the robot, shown in Figure 12, should be considered, since it depends directly on the mechanical limitations of the same, which are exhibited in Table 3. The base of the redundant planar manipulator robot is located in the $(x, y) = (0, 0)$ coordinate, and its workspace is presented highlighted in blue, with its maximum and minimum scopes being 600 and 160 mm, accordingly, as shown in Figure 12.

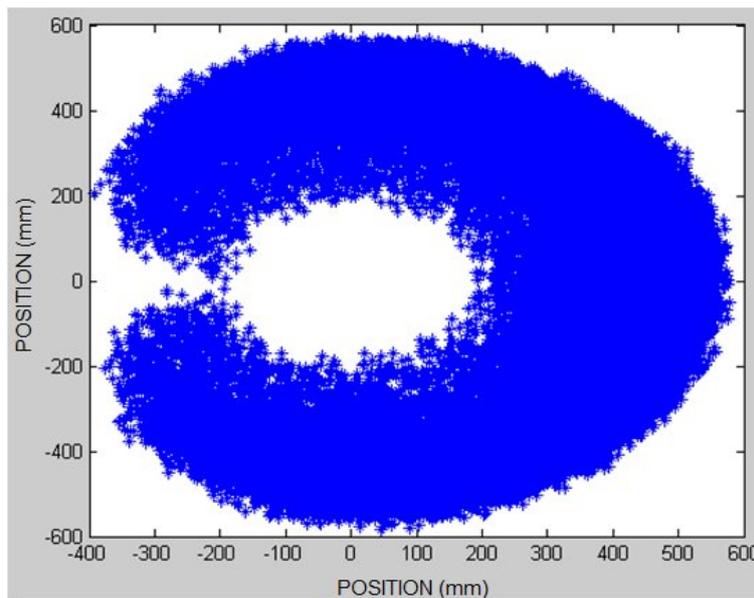


Figure 12. Workspace of the 6-DoF redundant planar manipulator robot.

Table 3. Mechanical limitations of the 6-DoF redundant manipulator planar robot joints.

Joint	Range
Rotary 1	$-90^\circ \leq \theta_1 \leq 90^\circ$
Prismatic 1	$0 \leq d_1 \leq 40$ (mm)
Rotary 2	$-90^\circ \leq \theta_2 \leq 90^\circ$
Prismatic 2	$0 \leq d_2 \leq 40$ (mm)
Rotary 3	$-90^\circ \leq \theta_3 \leq 90^\circ$
Prismatic 3	$0 \leq d_3 \leq 40$ (mm)

By means of the graphic interface, it is possible to enter the coordinates (x, y) of the end effector of the redundant planar manipulator robot; after which, the SA algorithm automatically calculates the inverse kinematics of the same. Figure 11 shows the menu to enter the position. In this example, the coordinate (350,150) yields the solution set with the position adopted by each joint for the entered coordinate. This result corresponds to an inverse kinematics solution selected from multiple options, after 450 iterations of the SA algorithm. Nevertheless, if the coordinates entered are outside its workspace, the SA algorithm automatically adjusts the end of the redundant planar manipulator robot to the closest position within the allowed region, and then delivers the solution set for inverse kinematics.

After continuous simulations for different set points, it is concluded that the SA algorithm requires on average 200 iterations to reach the desired position with an error of 1 mm or less. In addition, with the same number of iterations, the average convergence speed of the algorithm, to make the end effector of the redundant planar manipulator robot move from the initial position to the set point, is 0.5 (s). The scenario changes when the set point is within an unfeasible workspace; in this case, the algorithm needs 2 (s) to take the end effector to an allowed position. The convergence performance of the algorithm strongly depends on the selection of the environmental and cooling parameters defined in Section 6.

8. Conclusions

To characterize the kinematic and dynamic behavior of redundant planar manipulator robots, a graphic simulator was designed and implemented, which allows for entering the Cartesian coordinates (x, y) of the end effector of these robots through a graphic interface, with the purpose of automatically obtaining its inverse kinematics. The basic structure of this basic simulator was built using the software Solidworks, which was subsequently potentiated through the SimMechanics tools of MATLAB/Simulink.

An algorithm based on the probabilistic technique called Simulated Annealing was used for the automatic calculation of the optimal solution, among several options, of the inverse kinematics of the 6-DoF redundant planar manipulator robot. By means of the SA algorithm, it was possible to minimize significantly the distance between any initial position and a target position of the robot's end effector, reaching values close to zero. In this way, the enhanced algorithm is able to distinguish and correct joints out of the mechanical limits or, also, out of the workspace. In all tests for different targets, the optimization criteria outlined was satisfied (fitness function below or equal 1 mm) and the inverse kinematic solution of the manipulator robot was obtained in a time shorter than one second. This technique can be easily reproduced in other type of manipulator robot configurations with more (or less) redundancy, given that their main component is the information about the direct kinematics, which is easy to obtain in most cases.

To prevent excessive oscillation in the programmed joint trajectories and to ensure the continuity in speed and acceleration of the robot links, the cubic spline interpolation method was used. The dynamic equations were obtained from the L-E equation. Finally, the effectiveness of the methodologies developed was verified through computational simulations of a virtual robot. The results obtained in this study are currently being implemented in the real redundant robot shown in Figure 13.

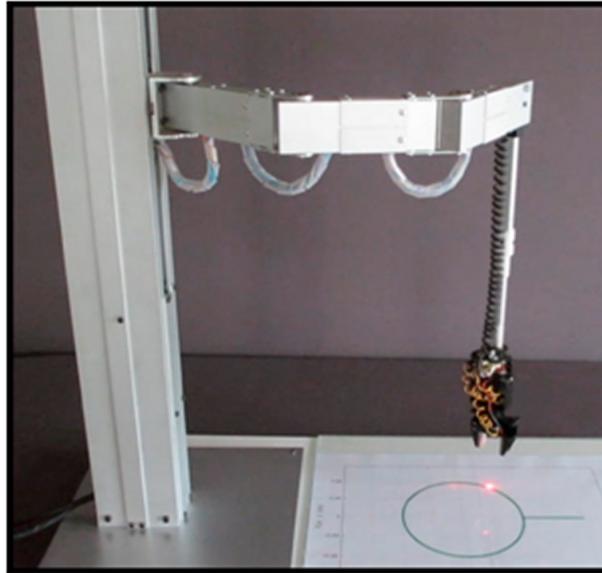


Figure 13. Real redundant planar manipulator robot.

Author Contributions: Conceptualization, C.U. and D.S.; Methodology, C.U. and D.S.; Software, C.U. and D.S.; Validation, C.U. and D.S.; Formal analysis, C.U. and D.S.; Investigation, C.U. and D.S.; Resources, C.U. and D.S.; Data curation, C.U. and D.S.; Writing—original draft preparation, C.U. and D.S.; Writing—review and editing, C.U. and D.S.; Visualization, C.U. and D.S.; Supervision, C.U.; Project administration, C.U.; Funding acquisition, C.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Acknowledgments: This work was supported by the Vicerrectoría de Investigación, Desarrollo e Innovación of the Universidad de Santiago de Chile, Chile.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Elements of the inertia matrix $\mathbf{M}(q)$:

$$M_{11} = I_1 + I_2 + I_3 + m_1 l_{c1}^2 + m_2 \lambda_1^2 + m_3 [l_1^2 + l_{c3}^2 + 2l_{c3} l_1 \cos(\theta_2)] + m_4 [l_1^2 + \lambda_2^2 + 2l_1 \lambda_2 \cos(\theta_2)] + m_5 [l_1^2 + l_2^2 + l_{c5}^2 + \cos(\theta_2) + 2l_2 l_{c5} \cos(\theta_3) + 2l_1 l_{c5} \cos(\theta_2 + \theta_3)] + m_6 [l_1^2 + l_2^2 + l_3^2 + 2l_1 l_2 \cos(\theta_2) + 2l_2 l_3 \cos(\theta_3) + 2l_1 l_3 \cos(\theta_2 + \theta_3)]; \quad (A1)$$

$$M_{12} = m_3 [-l_{c3} \sin(\theta_2)] + m_4 [-\lambda_2 \sin(\theta_2)] + m_5 [-l_2 \sin(\theta_2) - l_{c5} \sin(\theta_2 + \theta_3)] + m_6 [-l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3)]; \quad (A2)$$

$$M_{13} = I_2 + I_3 + m_3 [l_{c3}^2 + l_{c3} l_1 \cos(\theta_2)] + m_4 [\lambda_2^2 + l_1 \lambda_2 \cos(\theta_2)] + m_5 [l_2^2 + l_{c5}^2 + l_1 l_2 \cos(\theta_2) + 2l_2 l_{c5} \cos(\theta_3) + l_1 l_{c5} \cos(\theta_2 + \theta_3)] + m_6 [l_2^2 + l_3^2 + l_1 l_2 \cos(\theta_2) + 2l_2 l_3 \cos(\theta_3) + l_1 l_3 \cos(\theta_2 + \theta_3)]; \quad (A3)$$

$$M_{14} = m_4 [l_1 \sin(\theta_2)] + m_5 [l_1 \sin(\theta_2) - l_{c5} \sin(\theta_3)] + m_6 [l_1 \sin(\theta_2) - l_3 \sin(\theta_3)]; \quad (A4)$$

$$M_{15} = I_3 + m_5 [l_{c5}^2 + l_2 l_{c5} \cos(\theta_3) + l_1 l_{c5} \cos(\theta_2 + \theta_3)] + m_6 [l_3^2 + l_2 l_3 \cos(\theta_3) + l_1 l_3 \cos(\theta_2 + \theta_3)]; \quad (A5)$$

$$M_{16} = m_6 [l_2 \sin(\theta_3) + l_1 \sin(\theta_2 + \theta_3)]; \quad (A6)$$

$$M_{21} = m_3 [-l_{c3} \sin(\theta_2)] + m_4 [-\lambda_2 \sin(\theta_2)] + m_5 [-l_2 \sin(\theta_2) - l_{c5} \sin(\theta_2 + \theta_3)] + m_6 [-l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3)]; \quad (A7)$$

$$M_{22} = m_2 + m_3 + m_4 + m_5 + m_6; \quad (A8)$$

$$M_{23} = m_3[-l_{c3} \sin(\theta_2)] + m_4[-\lambda_2 \sin(\theta_2)] + m_5[-l_2 \sin(\theta_2) - l_{c5} \sin(\theta_2 + \theta_3)] + m_6[-l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3)]; \quad (A9)$$

$$M_{24} = m_4 \cos(\theta_2) + m_5 \cos(\theta_2) + m_6 \cos(\theta_2); \quad (A10)$$

$$M_{25} = m_5[-l_{c5} \sin(\theta_2 + \theta_3)] + m_6[-l_3 \sin(\theta_2 + \theta_3)]; \quad (A11)$$

$$M_{26} = m_6 \cos(\theta_2 + \theta_3); \quad (A12)$$

$$M_{31} = I_2 + I_3 + m_3[l_{c3}^2 + l_{c3}l_1 \cos(\theta_2)] + m_4[\lambda_2^2 + l_1\lambda_2 \cos(\theta_2)] + m_5[l_2^2 + l_{c5}^2 + l_1l_2 \cos(\theta_2) + 2l_2l_{c5} \cos(\theta_3) + l_1l_{c5} \cos(\theta_2 + \theta_3)] + m_6[l_2^2 + l_3^2 + l_1l_2 \cos(\theta_2) + 2l_2l_3 \cos(\theta_3) + l_1l_3 \cos(\theta_2 + \theta_3)]; \quad (A13)$$

$$M_{32} = m_3[-l_{c3} \sin(\theta_2)] + m_4[-\lambda_2 \sin(\theta_2)] + m_5[-l_2 \sin(\theta_2) - l_{c5} \sin(\theta_2 + \theta_3)] + m_6[-l_2 \sin(\theta_2) - l_3 \sin(\theta_2 + \theta_3)]; \quad (A14)$$

$$M_{33} = I_2 + I_3 + m_3l_{c3}^2 + m_4\lambda_2^2 + m_5[l_2^2 + l_{c5}^2 + 2l_2l_{c5} \cos(\theta_3)] + m_6[l_2^2 + l_3^2 + 2l_2l_3 \cos(\theta_3)]; \quad (A15)$$

$$M_{34} = m_5[-l_{c5} \sin(\theta_3)] + m_6[-l_3 \sin(\theta_3)]; \quad (A16)$$

$$M_{35} = I_3 + m_5[l_{c5}^2 + l_2l_{c5} \cos(\theta_3)] + m_6[l_3^2 + l_2l_3 \cos(\theta_3)]; \quad (A17)$$

$$M_{36} = m_6l_2 \sin(\theta_3); \quad (A18)$$

$$M_{41} = m_4[l_1 \sin(\theta_2)] + m_5[l_1 \sin(\theta_2) - l_{c5} \sin(\theta_3)] + m_6[l_1 \sin(\theta_2) - l_3 \sin(\theta_3)]; \quad (A19)$$

$$M_{42} = m_4 \cos(\theta_2) + m_5 \cos(\theta_2) + m_6 \cos(\theta_2); \quad (A20)$$

$$M_{43} = m_5[-l_{c5} \sin(\theta_3)] + m_6[-l_3 \sin(\theta_3)]; \quad (A21)$$

$$M_{44} = m_4 + m_5 + m_6; \quad (A22)$$

$$M_{45} = m_5[-l_{c5} \sin(\theta_3)] + m_6[-l_3 \sin(\theta_3)]; \quad (A23)$$

$$M_{46} = m_6 \cos(\theta_3); \quad (A24)$$

$$M_{51} = I_3 + m_5[l_{c5}^2 + l_2l_{c5} \cos(\theta_3) + l_1l_{c5} \cos(\theta_2 + \theta_3)] + m_6[l_3^2 + l_2l_3 \cos(\theta_3) + l_1l_3 \cos(\theta_2 + \theta_3)]; \quad (A25)$$

$$M_{52} = m_5[-l_{c5} \sin(\theta_2 + \theta_3)] + m_6[-l_3 \sin(\theta_2 + \theta_3)]; \quad (A26)$$

$$M_{53} = I_3 + m_5[l_{c5}^2 + l_2l_{c5} \cos(\theta_3)] + m_6[l_3^2 + l_2l_3 \cos(\theta_3)]; \quad (A27)$$

$$M_{54} = m_5[-l_{c5} \sin(\theta_3)] + m_6[-l_3 \sin(\theta_3)]; \quad (A28)$$

$$M_{55} = I_3 + m_5l_{c5}^2 + m_6l_3^2; \quad (A29)$$

$$M_{56} = 0; \quad (A30)$$

$$M_{61} = m_6[l_2 \sin(\theta_3) + l_1 \sin(\theta_2 + \theta_3)]; \quad (A31)$$

$$M_{62} = m_6 \cos(\theta_2 + \theta_3); \quad (A32)$$

$$M_{63} = m_6l_2 \sin(\theta_3); \quad (A33)$$

$$M_{64} = m_6 \cos(\theta_3); \quad (A34)$$

$$M_{65} = 0; \quad (A35)$$

$$M_{66} = m_6; \quad (A36)$$

Components of the centrifugal and Coriolis forces vector $\mathbf{c}(q, \dot{q})$:

$$\begin{aligned} C_{11} = & 2\dot{d}_1\dot{\theta}_1[m_2\lambda_1 + (m_3 + m_4 + m_5 + m_6)l_1 + (m_3l_{c3} + m_4\lambda_2 + m_5l_2 + m_6l_2) \cos(\theta_2) \\ & + (m_5l_{c5} + m_6l_3) \cos(\theta_2 + \theta_3)] + 2\dot{d}_2(\dot{\theta}_1 + \dot{\theta}_2)[m_4\lambda_2 + (m_5 + m_6)l_2 + (m_4 + m_5 \\ & + m_6)l_1 \cos(\theta_2) + (m_5l_{c5} + m_6l_3) \cos(\theta_3)] + 2\dot{d}_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)[m_6l_3 + m_6l_2 \cos(\theta_3) \\ & + m_6l_1 \cos(\theta_2 + \theta_3)] - \dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2)[(m_3l_{c3} + m_4\lambda_2 + m_5l_2 + m_6l_2)l_1 \sin(\theta_2) + (m_5l_{c5} \\ & + m_6l_3)l_1 \sin(\theta_2 + \theta_3)] - \dot{\theta}_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)[(m_5l_{c5} + m_6l_3)l_2 \sin(\theta_3) + (m_5l_{c5} \\ & + m_6l_3)l_1 \sin(\theta_2 + \theta_3)]; \end{aligned} \quad (A37)$$

$$\begin{aligned} C_{21} = & -(\dot{\theta}_1)^2[m_2\lambda_1 + (m_3 + m_4 + m_5 + m_6)l_1] - (\dot{\theta}_1 + \dot{\theta}_2)^2(m_3l_{c3} + m_4\lambda_2 + m_5l_2 + m_6l_2) \\ & \cos(\theta_2) - (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2(m_5l_{c5} + m_6l_3) \cos(\theta_2 + \theta_3) - 2\dot{d}_2(\dot{\theta}_1 + \dot{\theta}_2)(m_4 + m_5 + m_6) \\ & \sin(\theta_2) - 2\dot{d}_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)m_6 \sin(\theta_2 + \theta_3); \end{aligned} \quad (A38)$$

$$\begin{aligned} C_{31} = & 2\dot{d}_1\dot{\theta}_1[(m_3l_{c3} + m_4\lambda_2 + m_5l_2) \cos(\theta_2) + (m_5l_{c5} + m_6l_3) \cos(\theta_2 + \theta_3)] + 2\dot{d}_2(\dot{\theta}_1 + \dot{\theta}_2) \\ & [m_3\lambda_2 + (m_5 + m_6)l_2 + (m_5l_{c5} + m_6l_3) \cos(\theta_3)] + 2\dot{d}_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)[m_6l_3 \\ & + m_6l_2 \cos(\theta_3)] + (\dot{\theta}_1)^2[(m_3l_{c3} + m_4\lambda_2 + m_5l_2 + m_6l_2)l_1 \sin(\theta_2) + (m_5l_{c5} \\ & + m_6l_3) \sin(\theta_2 + \theta_3)] - \dot{\theta}_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)[(m_5l_{c5} + m_6l_3)l_2 \sin(\theta_3)]; \end{aligned} \quad (A39)$$

$$\begin{aligned} C_{41} = & -(\dot{\theta}_1)^2(m_4 + m_5 + m_6)l_1 \cos(\theta_2) - (\dot{\theta}_1 + \dot{\theta}_2)^2(m_4\lambda_2 + m_5l_2 + m_6l_2) - (\dot{\theta}_1 + \dot{\theta}_2 \\ & + \dot{\theta}_3)^2(m_5l_{c5} + m_6l_3) \cos(\theta_3) + 2\dot{d}_1\dot{\theta}_1(m_4 + m_5 + m_6) \sin(\theta_2) - 2\dot{d}_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ & m_6 \sin(\theta_3); \end{aligned} \quad (A40)$$

$$\begin{aligned} C_{51} = & 2\dot{d}_1\dot{\theta}_1(m_5l_{c5} + m_6l_3) \cos(\theta_2 + \theta_3) + 2\dot{d}_2(\dot{\theta}_1 + \dot{\theta}_2)(m_5l_{c5} + m_6l_3) \cos(\theta_3) + 2\dot{d}_3(\dot{\theta}_1 \\ & + \dot{\theta}_3)m_6l_3 + (\dot{\theta}_1)^2(m_5l_{c5} + m_6l_3)l_1 \sin(\theta_2 + \theta_3) + (\dot{\theta}_1 + \dot{\theta}_2)^2(m_5l_{c5} + m_6l_3)l_2 \sin(\theta_3); \end{aligned} \quad (A41)$$

$$\begin{aligned} C_{61} = & -(\dot{\theta}_1)^2m_6l_1 \cos(\theta_2 + \theta_3) - (\dot{\theta}_1 + \dot{\theta}_2)^2m_6l_2 \cos(\theta_3) - (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2m_6l_3 \\ & + 2\dot{d}_1\dot{\theta}_1m_6 \sin(\theta_2 + \theta_3) + 2\dot{d}_2(\dot{\theta}_1 + \dot{\theta}_2)m_6 \sin(\theta_3) \end{aligned} \quad (A42)$$

with: $\lambda_1 = L_1 + d_1 + l_{c2}$; $l_1 = L_1 + d_1 + L_2$; $\lambda_2 = L_1 + d_2 + l_{c4}$; $l_2 = L_1 + d_2 + L_2$; $l_3 = L_1 + d_3 + l_{c6}$, where m_i : mass of the link i (kg), $i=1,2,3,4,5,6$; l_1 : length of the arm (mm); l_2 : length of the first forearm (mm); l_3 : length of the second forearm (mm); I_1 : moment of inertia of the first link ($kg \cdot m^2$); I_2 : moment of inertia of the third link ($kg \cdot m^2$); I_3 : moment of inertia of the fifth link ($kg \cdot m^2$).

References

- Aljalal, M.; Ibrahim, S.; Djemal, R.; Ko, W. Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals. *Intell. Serv. Robot.* **2020**. [[CrossRef](#)]
- Kim, J.; Cauli, N.; Vicente, P.; Damas, B.; Bernardino, A.; Santos-Victor, J.; Cavallo, F. Cleaning tasks knowledge transfer between heterogeneous robots: A deep learning approach. *J. Intell. Robot. Syst.* **2020**, *98*, 191–205. [[CrossRef](#)]
- Wang, Z.; Or, K.; Hirai, S. A dual-mode soft gripper for food packaging. *Robot. Auton. Syst.* **2020**, *125*, 103427. [[CrossRef](#)]
- Iqbal, H.; Khan, M.U.A.; Yi, B.-J. Analysis of duality-based interconnected kinematics of planar serial and parallel manipulators using screw theory. *Intell. Serv. Robot.* **2020**, *13*, 47–62. [[CrossRef](#)]
- Kern, J.; Jamett, M.; Urrea, C.; Torres, H. Development of a neural controller applied in a 5-DoF robot redundant. *IEEE Lat. Am. Trans.* **2014**, *12*, 98–106. [[CrossRef](#)]
- Le, Q.D.; Kang, H.-J. Finite-time fault-tolerant control for a robot manipulator based on synchronous terminal sliding mode control. *Appl. Sci.* **2020**, *10*, 2998. [[CrossRef](#)]
- Cieślak, P.; Simoni, R.; Ridao Rodríguez, P.; Youakim, D. Practical formulation of obstacle avoidance in the task-priority framework for use in robotic inspection and intervention scenarios. *Robot. Auton. Syst.* **2020**, *124*, 103396. [[CrossRef](#)]
- Shanda, W.; Xiao, L.; Qingsheng, L.; Baoling, H. Existence conditions and general solutions of closed-form inverse kinematics for revolute serial robots. *Appl. Sci.* **2019**, *9*, 4365. [[CrossRef](#)]

9. Fan, S.; Xie, X.; Zhou, X. Optimum manipulator path generation based on improved differential evolution constrained optimization algorithm. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419872060. [[CrossRef](#)]
10. Shafei, A.M.; Mirzaeinejad, H. A general formulation for managing trajectory tracking in non-holonomic moving manipulators with rotary-sliding joints. *J. Intell. Robot. Syst.* **2020**. [[CrossRef](#)]
11. Rahmani, M.; Rahman, M.H. Adaptive neural network fast fractional sliding mode control of a 7-DOF exoskeleton robot. *Int. J. Control Autom. Syst.* **2020**, *18*, 124–133. [[CrossRef](#)]
12. Bobrow, J.E.; Martin, B.; Sohl, G.; Wang, E.C.; Park, F.C.; Kim, J. Optimal robot motions for physical criteria. *J. Robot. Syst.* **2001**, *18*, 785–795. [[CrossRef](#)]
13. López-Franco, C.; Hernandez-Barragan, J.; Alanis, A.Y.; Arana-Daniel, N. A soft computing approach for inverse kinematics of robot manipulators. *Eng. Appl. Artif. Intell.* **2018**, *74*, 104–120. [[CrossRef](#)]
14. Li, T.; Zheng, S.; Shu, X.; Wang, C.; Liu, C. Self-recognition grasping operation with a vision-based redundant manipulator system. *Appl. Sci.* **2019**, *9*, 5172. [[CrossRef](#)]
15. Qian, Y.; Yuan, J.; Wan, W. Improved trajectory planning method for space robot-system with collision prediction. *J. Intell. Robot. Syst.* **2020**, *99*, 289–302. [[CrossRef](#)]
16. Teja, H.; Shah, S. Learning inverse kinematic solutions of redundant manipulators using multiple internal models. In Proceedings of the 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), Utown, Singapore, 26–29 June 2016; p. 1371.
17. Zhao, R.; Shi, Z.; Guan, Y.; Shao, Z.; Zhang, Q.; Wang, G. Inverse kinematic solution of 6R robot manipulators based on screw theory and the Paden-Kahan subproblem. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–11. [[CrossRef](#)]
18. Bai, L.; Yang, J.; Chen, X.; Jiang, P.; Liu, F.; Zheng, F.; Sun, Y. Solving the time-varying inverse kinematics problem for the Da Vinci surgical robot. *Appl. Sci.* **2019**, *9*, 546. [[CrossRef](#)]
19. Huang, Q.; Nie, L. A fast convergence efficiency method of inverse kinematics for robot manipulators. *J. Appl. Sci.* **2013**, *13*, 5174–5179. [[CrossRef](#)]
20. Park, S.-O.; Lee, M.C.; Kim, J. Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics. *Int. J. Control Autom. Syst.* **2020**, *18*, 2095–2107. [[CrossRef](#)]
21. Besset, P.; Taylor, C.J. Inverse kinematics for a redundant robotic manipulator used for nuclear decommissioning. In Proceedings of the UKACC International Conference on Control (CONTROL), Loughborough, UK, 9–11 July 2014; pp. 56–61.
22. Farzan, S.; DeSouza, G.N. A parallel evolutionary solution for the inverse kinematics of generic robotic manipulators. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 358–365.
23. Köker, R.; Cakar, T. A neuro-genetic-simulated annealing approach to the inverse kinematics solution of robots: A simulation based study. *Eng. Comput.* **2016**, *32*, 553–565. [[CrossRef](#)]
24. López-Franco, C.; Hernández-Barragán, J.; Alanis, A.Y.; Arana-Daniel, N.; López-Franco, M. Inverse kinematics of mobile manipulators based on differential evolution. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–22. [[CrossRef](#)]
25. Dereli, S.; Köker, R. Simulation based calculation of the inverse kinematics solution of 7-DOF robot manipulator using artificial bee colony algorithm. *SN Appl. Sci.* **2020**, *2*, 27. [[CrossRef](#)]
26. Zhou, Z.; Guo, H.; Wang, Y.; Zhu, Z.; Wu, J.; Liu, X. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–15. [[CrossRef](#)]
27. Song, B.; Wang, Z.; Zou, L.; Xu, L.; Alsaadi, F.E. A new approach to smooth global path planning of mobile robots with kinematic constraints. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 107–119. [[CrossRef](#)]
28. Kereluk, J.A.; Emami, M.R. A new modular, autonomously reconfigurable manipulator platform. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 1–17. [[CrossRef](#)]
29. Truong, L.V.; Huang, S.D.; Yen, V.T.; Cuong, P.V. Adaptive trajectory neural network tracking control for industrial robot manipulators with deadzone robust compensator. *Int. J. Control Autom. Syst.* **2020**, 1–12.
30. Bjerkgeng, M.; Pettersen, K.Y. A new Coriolis matrix factorization. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 4974–4979.
31. Zarrin, A.; Azizi, S.; Aliasghary, M. A novel inverse kinematics scheme for the design and fabrication of a five degree of freedom arm robot. *Int. J. Dyn. Control* **2020**, *8*, 604–614. [[CrossRef](#)]

32. Segota, S.B.; Andelic, N.; Lorencin, I.; Saga, M.; Car, Z. Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1–16.
33. García-Ródenas, R.; López-García, M.L.; Sánchez-Rico, M.T.; López-Gómez, J.A. A bilevel approach to enhance prefixed traffic signal optimization. *Eng. Appl. Artif. Intell.* **2019**, *84*, 51–65. [[CrossRef](#)]
34. Bi, H.; Lu, F.; Duan, S.; Huang, M.; Zhu, J.; Liu, M. Two-level principal–agent model for schedule risk control of IT outsourcing project based on genetic algorithm. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103584. [[CrossRef](#)]
35. Ileri, Y.Y.; Hacibeyoglu, M. Advancing competitive position in healthcare: A hybrid metaheuristic nutrition decision support system. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1385–1398. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).