ISSN: 2831 - 6789

Research Article

Journal of Robotics and Automation Research

Symbolic and Numerical Mathematical Modeling of the Jacobian of the Yaskawa Motoman - Gp7 Robot using Matlab©

Igor Manoel Garbim Cruz [1]*, Kennedy Robert Sotana Bertoncelli [2], Tiago de Aguiar Souza[3], Rogério Adas Pereira Vitalli [4]

Department of Robotic Engineering Advanced Robotics Institute (IAR) Santo André, Brazil

***Corresponding author**
Igor Manoel Garbim Cruz, Department of Robotic Engineering Advanced Robotics Institute (IAR) Santo André, Brazil

Submitted: 01 Aug 2022; **Accepted**: 08 Aug 2022; **Published**: 16 Aug 2022

Citation: *Cruz, I. M. G., de Aguiar SOUZA, T., Bertoncelli, K. R. S., Vitalli, R. A. P. (2022). Symbolic and numerical mathematical modeling of the Jacobian of the YASKAWA MOTOMAN - GP7 robot using MATLAB©. J Robot Auto Res 3(2), 227-234.*

Abstract

Robotic manipulators have besides static positioning problems, but besides static we have in the robot its speeds and forces applied in any kind of movement. one of the show important activities of robot motion control is the definition of its movements in the workspace. to perform these moves its joints need I 'm moving and to define these positions in the spaces we need to calculate the direct kinematics and the opposite we call the inverse kinematics. The main objective of this paper is to present a study on velocity analysis, known as the Jacobian matrix. The methodology employed in this exploratory scientific research will ok developed from experimental tests, bibliographic references and case studies applied at the Advanced robotics Institute (IAR). the robot under study is the YASKAWA-MOTOMAN-GP7 present at the robotics Laboratory of the IAR. The work brings as contribution the implementation of the Denavit-Hartenberg notation and the validation of the Jacobian matrix of the robot. The results are the determination of the equations of the inverse kinematics using the method based on the inverse Jacobian matrix. The code was implemented in MATLAB© software capable of proving the developed mathematical model, which is applicable in industrial robots.

Keywords: Computational Simulation; Denavit-Hartenberg; Inverse Kinematics; Jacobian.

Introduction

The new technological pattern of industries directs them towards a new production paradigm, leveraging productivity, increasing profitability, performing costs and process quality. the history of industrial automation is characterized by periods of quick technological change, the result of this has been the development of automata capable of performing the show complex tasks with the highest possible level of sophistication and precision. It was exactly in this context that industrial robotic manipulators emerged [1]. The Denavit-Hartenberg (D-H) notation for describing a serial link the geometry of the mechanism is the fundamental tool of the roboticist. given such a description of a manipulator, we can make use of algorithms and techniques to find kinematic, Jacobian, dynamical, motion planning and simulation solutions, for example [1,2].

The manipulator robots are considered an indispensable part in modern factories, due to their capacity to perform the show variety types of tasks of high level of complexity and danger, in an efficient and reliable way, with an unquestionable cost

benefit relation. Aiming at the study and the different practical applications, in the projects of manipulator robots two lines of research can ok addressed, the kinematics and dynamics. Kinematics deals with the movement without considering the static forces that cause this movement, the study of " accelerations, velocities and inertias " are part of dynamics. in the literature, the kinematics of manipulator robots is approached by means of two models, that of direct and inverse kinematics _ The direct kinematics consists in finding the orientation and position of the end effector from the vector of joint angles and the geometric parameters of the model. According to the matrix method is proposed for modeling and stock solving problems that use the direct kinematics technique [3].

The inverse kinematics is considered an effective technique for controlling a robotic arm, consisting in finding the vector of joint angles from the orientation and position of the terminal effector. It presents great challenges due to the nonlinearity of the equations and multiple solutions for manipulators with many degrees of freedom [4,5]. The development of a numerical algorithm to find the angular positions for a work defined with respect to its terminal element in Cartesian space, contains the solution of the inverse kinematic problem through the recursive numerical method that uses the kinematic model and the inverse Jacobian matrix of the manipulator. The Gauss Elimination method was used for the inversion of the Jacobian matrix [6,7].

Literature Revision

Direct Kinematics

Kinematics in robotics is a form of representation referring to the geometric explanation of the structure of a robot. From the geometric equation, it is possible to obtain the connection between the concept of spatial geometry of joints and the theory of end-effector coordinates to figure the position of an object [6]. The purpose of studying kinematics is to determine the relative position of a frame in relation to its original coordinates. Forward kinematics is transforming the joint variable to the end-effector position. Also, inverse kinematics is to transform the position of the final effector to the joint variable. Denavit-Hartenberg parameters (also called D-H parameters) are the four parameters related to a particular convention for delimiting frames of reference to links in a spatial chain [8].

Denavit-Hartenberg (D-H) Notation

Denavit and Hartenberg proposed a systematic notation for assigning an orthonormal coordinate system according to the "right-hand rule", one for each link in an open kinematic chain of links. Once these coordinate systems fixed to the link are assigned, transformations between adjacent coordinate systems can be represented by a homogeneous coordinate transformation matrix [8].

According to in the original D-H representation, the joint axis is associated with the z axis and each matrix is represented by the product of four basic transformations involving rotations and translations, as we can see in Equation (1) [9].

$$^{k-1}T_k = R_{z,\theta} * Transl_{z,d} * Transl_{x,a} * R_{x,a} \qquad (1)$$

Figure 1 shows the D-H parameters with a graphical representation [1].

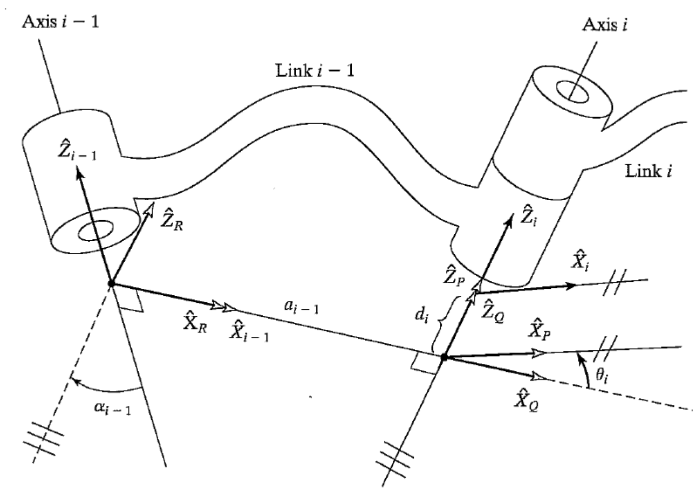

Figure 1: Representation of D-H parameters [1].

Table 1 shows that any homogeneous transformation matrix can be described from four basic transformations.

Table 1: Basic Denavit-Hartenberg Transformations [5].

SYMBOL	MEANING
Rx,θ	Rotation around the z axis by an angle θ
Tz,d	Translation along the z axis for a distance d
Tx,α	Translation along the x axis for a distance a
Rx,α	Rotation around the x axis by an angle α

To obtain the homogeneous transformation matrix (2) that goes from the base to the end of the end effector, we have to multiply all the transformation matrices obtained by the D-H algorithm, that is, the resulting matrix can be considered as the solution to the problem of direct kinematics (3).

$$^{i-1}_{\ \ i}T = \begin{bmatrix} C\theta_i & -S\theta_i . C\alpha_i & S\theta_i . S\alpha_i & \alpha_i . C\theta_i \\ S\theta_i & C\theta_i . C\alpha_i & -C\theta_i . S\alpha_i & \alpha_i . S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$^0_6T = {}^0_1T . {}^1_2T . {}^2_3T . {}^3_4T . {}^4_5T . {}^5_6T \qquad (3)$$

In the case of composite rotations around two or more axes, the rotation component (orientation) of the transformation matrix will be more complex than that of the simple rotations indicated above. The geometric, homogeneous transformation matrix has, in addition to rotations and translation, four more terms (the last line). Figure 2 shows the components of the 3D transformation matrix.

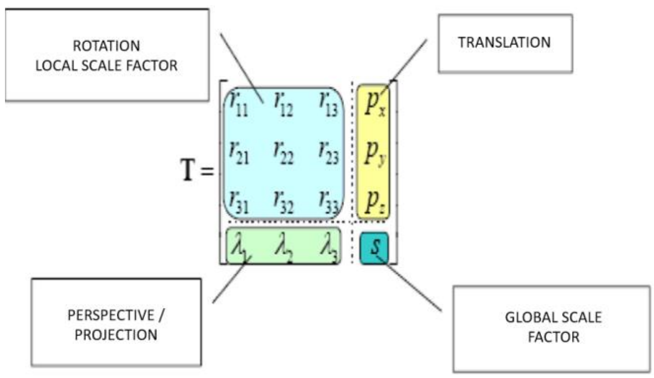

Figure 2: Components of the 3D transformation matrix [8].

Inverse kinematics

Inverse kinematics is considered an effective technique for controlling a robotic arm, consisting in finding the vector of joint angles, based on the orientation and position of the end-effector. It presents great challenges due to nonlinearities of coupling of the equations of motion for manipulators with many degrees of freedom [10,11]. Traditional methods can be used to calculate the inverse kinematics of robotic manipulators, such as geometric, numerical-iterative and algebraic ones, considered laborious if the geometric structure of the manipulator is very complex

[5]. Solving problems dealing with the direct kinematics of manipulators is considered simple, if compared to the inverse way [11]. In inverse kinematics the angles between the joints can be determined by the parameters of the links and the position of the effector in the working volume. Figure 3 presents a diagram that illustrates the relationships between forward and inverse kinematics.

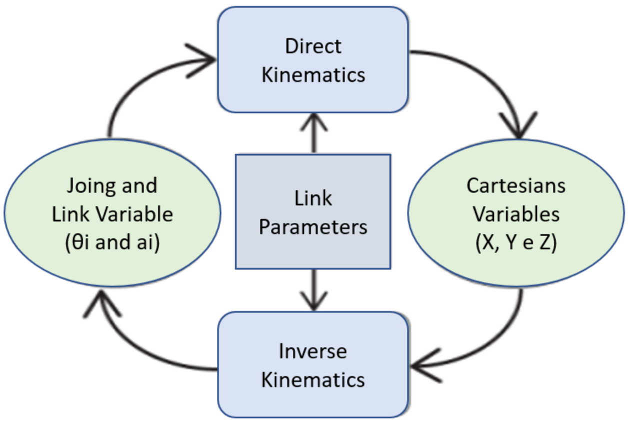

Figure 3: Representation of the relations between the robot kinematics.

Industrial Robotic Manipulators are normally positioned in joint variable space, however the objects to be manipulated are normally expressed in global coordinate systems. In order to control the position and orientation of a terminal element of a robot to reach its object, the inverse kinematics solution is the most important. In other words, we give the position and orientation of the terminal element of a robotic arm with i axes as 0 T i and its joint and link parameters, from there we want to find the joint angles q = (q1, q2, ..., qi) T of the robot so that the terminal element can be positioned as desired. The inverse kinematics problem, in general, can be solved using several methods, including the inverse transformation the "screw" algebra the doubles the interactive and geometric approximations [8,12,13,14,15]. According to Pieper (1968) he presented the kinematics solution for any manipulator of 6 (six) degrees of freedom in which there are revolutes or prismatics for the first 3 (three) joints the joint axes of the remaining 3 (three) joints intersect at a point, presented an inverse transformation technique using the 4 X 4 homogeneous transformation matrices in solving the kinematics solution for the same class of simple manipulators as discussed by Pieper [12]. Although the resulting solution is correct, it suffers from the fact that the solution does not have a clear indication of how to select the most appropriate solution from the many possible solutions for a particular engine configuration. Once the homogeneous transformation matrix has been defined for the robot under study, its inverse matrix is defined, as shown in Figure 4.

$$({}^{i-1}T_i)^{-1} = {}^{i}T_{i-1} = \begin{bmatrix} c\theta_i & s\theta_i & 0 & -a_i \\ -c\alpha_i.s\theta_i & c\theta_i.c\alpha_i & -s\alpha_i & -d_i.s\alpha_i \\ s\alpha_i.s\theta_i & -s\alpha_i.c\theta_i & c\alpha_i & -d_i.c\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 4: Inverse Transformation Matrix for the Homogeneous Transformation Matrix based on D-H Parameters.

Having these global results in hand, we proceed to the analysis of each of the links, according to the observations of the D-H parameters for the system in question. The first step is to multiply the inverse transformation matrix of the link by the homogeneous transformation matrix of the mechanism under study. Then, a new transformation matrix is found by multiplying the inverse transformation matrix of the link by a generic reference transformation matrix, as shown in Figure 5.

$$ {}^{i-1}T_i = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} \text{Yaw} & \text{Pitch} & \text{Roll} & \end{array} \\ \begin{bmatrix} n & s & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} $$

Figure 5: Reference Matrix.

The matrices resulting from the previous steps are compared, element by element, yielding equations to be used to find the Cartesian coordinates of the robotic arm. As the YASKAWA-MOTOMAN-GP7 Robot has 6 degrees of freedom, this sequence is repeated six times, that is, by the number of links in the system.

Jacobian of the YASKAWA-MOTOMAN-GP7

It is possible to correlate the relationships of linear and angular velocities of the effector as a function of the velocities of the joints in the same equation, resulting in equation [4].

$$\begin{bmatrix} V_n \\ W_n \end{bmatrix} = \begin{bmatrix} J_v(q) \\ J_w(q) \end{bmatrix} \dot{q},$$

or, defining the vector $V_n = (V_{n,} W_n)^t$, we have:

$$V_n = J(q)\,\dot{q}. \qquad [4]$$

The matrix J(q) is defined to be the Jacobian matrix of the effector. This matrix relates the linear and angular velocities of the end-effector, expressed in the base coordinate system, with the velocities of the joints, for a given configuration of the manipulator. In summary, column i of the Jacobian Matrix of a manipulator is given by the following equation [5].

$$J_i = \begin{bmatrix} Z_{i-1} * R_{i-1,n} \\ Z_{i-1} \end{bmatrix}, \text{if joint } i \text{ is of revolution;}$$

and

$$J_i = \begin{bmatrix} Z_{i-1} \\ 0 \end{bmatrix}, \text{if joint } i \text{ is translational.} \qquad [5]$$

The dimension of the Jacobian Matrix is m x n, where m is the number of rows, which is equal to the number of degrees of freedom of the robot's working field and n is the number of columns, which is equal to the number of robot joints. For a robot working in space, m will be at most equal to 6 and for a robot working in the plane, m will be at most equal to 3. The 6 degrees of freedom of space correspond to the three positioning degrees of freedom and the three orientations of a rigid body. For the plane there are two positioning degrees of freedom and only one orientation degree of freedom, since in the plane only velocity or angular position is defined around the axis perpendicular to the plane. Thus, it is observed that the number of columns of the Jacobian matrix is not fixed and must be defined by the interest of the problem and mainly, in terms of what the robot is capable of accomplishing.

Methodological Procedure

The methodology used in this exploratory scientific research will be developed from experimental tests, bibliographic references and a case study applied at the Advanced Institute of Robotics (IAR). The robot under study is the YASKAWA MOTOMAN - GP7, from the IAR Robotics laboratory. This model was chosen because it is the articulated type that is widely used in industries in general. The articulated type of manipulator has a configuration similar to that of a human arm, due to its set of 6 rotational joints, which can be separated into two groups that make up the wrist and those that correspond to the movement of the arm. Figure 6 shows the technical specifications of the robot, with the dimensions of its structure in millimeters.

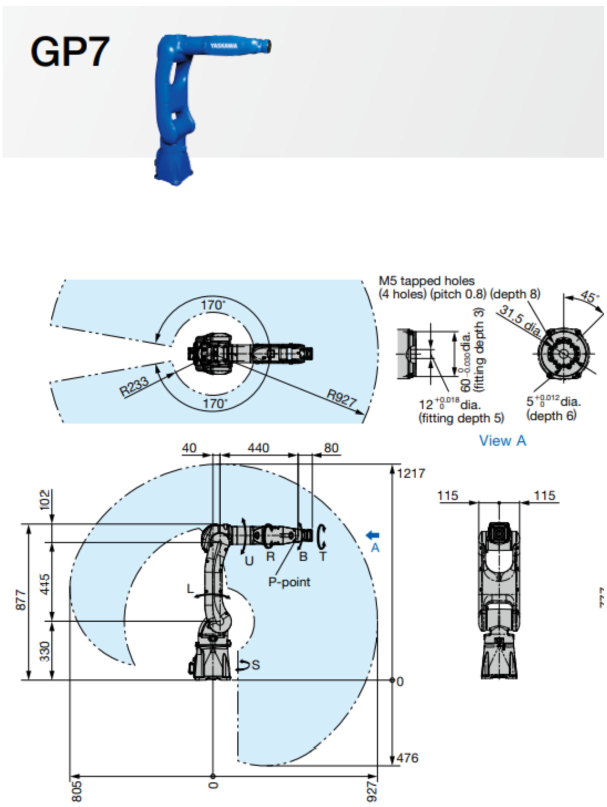

Figure 6: Technical Specifications of the Motoman-GP robot [8].

One of the features that differs this YASKAWA-MOTOMAN-GP7 robot model from the standard is the misalignment of the center of the base with the center of the end of the manipulator. This misalignment can be seen in Figure 6, which clearly shows that the link where the wrist is located 40 mm to the right with respect to the center of the robot base [16].

To identify the respective joints of the YASKAWA-MOTOMAN-GP7 robot, figure 7 shows the joints j1, j2, j3, j4, j5 and j6 in a schematic way.

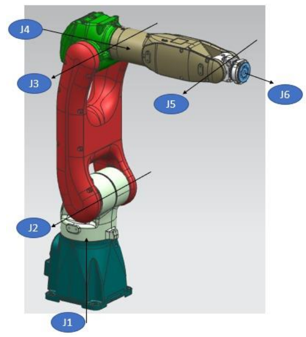

Figure 7: Schematic drawing of the YASKAWA MOTOMAN - GP7 robot with the identification of their respective joints.

The approach adopted to determine the direct kinematics of this robot starts from the preliminary analysis of the possible movements and recognition of the types of links and joints that are components of the system. Then, based on the information acquired, coordinate systems are adopted for the axes to be studied in order to determine the Denavit-Hartenberg parameters of the robotic arm under study. It follows the formation of the transformation matrices for each joint and the consequent composition of the robot's transformation matrix. The inverse kinematics is determined through a specific calculation method, based on the Cartesian coordinates found in the calculation of the forward kinematics. Among the various possibilities, the Jacobian inverse transformation method was chosen [17].

Direct Position Kinematics

To calculate the direct kinematics of position, the literal table is assembled with the parameters of D-H. The D-H parameters require a schematic model of the robot, as well as its characteristics, reference system and joint variables according to convention and Denavit-Hartenberg. Figure 8 shows the model developed for this case study.

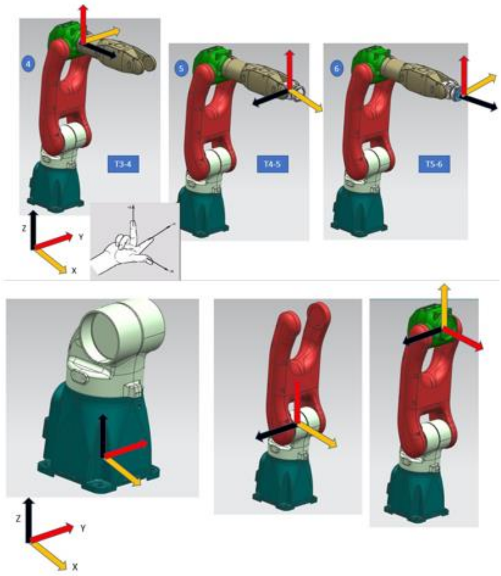

Figure 8 : Schematic Model of the Robot under study.

From the schematic model, the Denavit-Hartenberg numerical table is defined, according to table 2. The units used are millimeters for length and deg for angles.

Table 2: Denavit-Hartenberg Numerical Table.

Link	Joint	T^i_j	θ_i	α_i	a_i	d_i	Range
1	S	0-->1	0	0	40	330	±170°
2	L	1-->2	90	0	445	0	-65° +145°
3	U	2-->3	0	0	40	0	-70° +190°
4	R	3-->4	-90	90	0	440	±190°
5	B	4-->5	90	-90	0	0	±135°
6	T	5-->6	-90	90	0	80	±360°

From the schematic model, the Denavit-Hartenberg literary table is defined, according to table 3.

Table 3 :Denavit-Hartenberg Literary Table.

Link	Joint	T^i_j	θ_i	α_i	a_i	d_i	Range
1	S	0-->1	0	0	a1	d1	±170°
2	L	1-->2	θ1	0	a2	0	-65° +145°
3	U	2-->3	0	0	a3	0	-70° +190°
4	R	3-->4	θ3	α3	0	d4	±190°
5	B	4-->5	θ4	α4	0	0	±135°
6	T	5-->6	θ5	α5	0	d6	±360°

MATLAB©

MATLAB©, short for Matrix Laboratory, is a simple and straight forward language *software used to perform mathematical calculations, which has high computational performance and a wide library of pre-defined mathematical functions*. These features allow programming problems to be solved in a simpler way than in other computer languages [6]. The software used to demonstrate and mathematically prove the Denavit-Hartenberg algorithm was created in a simulation in MATLAB©, R2021a from the company MathWorks, software frequently used by researchers to perform calculations and systems in general [18].

The first step was to define the Denavit-Hartenberg parameters for each link joint, substituting the real values of the angular relationship between one joint to another and the translation (in millimeters) between joints of the YASKAWA-MOTOMAN-GP7 robot. The values were obtained from the manufacturer's datasheet [8]. Then, by multiplying these matrices, we obtain the homogeneous transformation matrix, which provides the mapping of the coordinates from the base to the end of the tool. In MATLAB© each link 'L' is defined from 1 to 6 and then the object YASKAWA-MOTOMAN-GP7 that will represent the robot with the connection of its links in series, as we can see in Table for D-H literary.

Validation

For validation, a test of code calculated in Robot were performed using the robot available at Advance Robotics Institute (IAR) carried out.

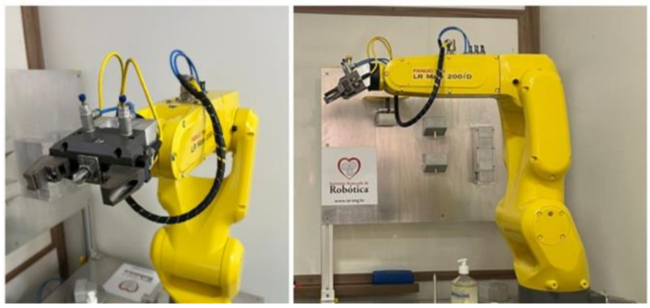

Figure 9: Fanuc LR Mate 200ID.

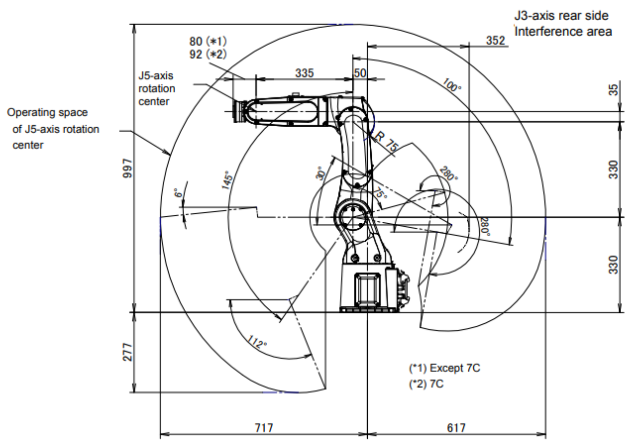

Figure 10: Fanuc LR Mate 200ID Specifications [19].

Results and Discursion

The methodology used to obtain the results regarding the forward kinematics and the inverse kinematics pointed to the initial analysis of the YASKAWA-MOTOMAN-GP7 Robot structure and to the relationship of the Denavit-Hartenberg parameters for each robot axis. Then, these values were related through transformation matrices for the same axes. Finally, these matrices were summed, resulting in a final transformation matrix of the robot, whose positioning at point T refers to the sum of the others (six links).

The representation of D-H results in obtaining a homogeneous 4x4 transformation matrix, representing that transforms the coordinates of the system of link i to the system i-1 of the previous link. With this, it is possible to express the transformation of coordinates from the "i" system to the "i-1" system. Then, the kinematic modeling is ready to be applied in the controller for the accomplishment of experimental tests and simulations of control of the system. By calculating the forward kinematics, it is possible to determine the x and y coordinates, once the joint coordinates are known. The resulting matrix can be considered as the solution of the forward kinematics problem according to figure 11, that is, the product of all transformation matrices obtained by D-H results in the forward kinematics solution.

^0_6T	T1 = 0 -1 0 0 1 0 0 40 0 0 1 330 0 0 0 1	T2 = 0 -1 0 0 1 0 0 445 0 0 1 0 0 0 0 1	T3 = 1 0 0 40 0 1 0 0 0 0 1 0 0 0 0 1
	T4 = 0 0 -1 0 -1 0 0 0 0 1 0 440 0 0 0 1	T5 = 0 0 -1 0 1 0 0 0 0 -1 0 0 0 0 0 1	T6 = 0 0 -1 0 -1 0 0 0 0 1 0 80 0 0 0 1
TR	1 0 0 -485 0 -1 0 -40 0 0 -1 770 0 0 0 1		

Figure 11: Result of the transformation matrices obtained by D-H.

Figure 12 shows the result of the mathematical calculations performed with the MATLAB © software for the values established by the robot under study.

```
>> T1=  [cos(pi/2)       -sin(pi/2)*cos(0)    sin(pi/2)*sin(0)         40*cos(pi/2)
    sin(pi/2)    cos(pi/2)*cos(0)      -cos(pi/2)*sin(0)    40*sin(pi/2)
    0        sin(0)            cos(0)           330
    0        0            0             1]

T1 =

    0.0000   -1.0000         0    0.0000
    1.0000    0.0000         0   40.0000
         0         0    1.0000  330.0000
         0         0         0    1.0000

>> T2=  [cos(pi/2)       -sin(pi/2)*cos(0)    sin(pi/2)*sin(0)         445*cos(pi/2)
    sin(pi/2)    cos(pi/2)*cos(0)      -cos(pi/2)*sin(0)    445*sin(pi/2)
    0        sin(0)            cos(0)           0
    0        0            0             1]

T2 =

    0.0000   -1.0000         0    0.0000
    1.0000    0.0000         0  445.0000
         0         0    1.0000         0
         0         0         0    1.0000

>> T3 = [cos(0)      -sin(0)*cos(0)       sin(0)*sin(0)        40*cos(0)
    sin(0)       cos(0)*cos(0)         -cos(0)*sin(0)       40*sin(0)
    0        sin(0)            cos(0)           0
    0        0            0             1]

T3 =

     1     0     0    40
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> T4 = [cos(-pi/2)      -sin(-pi/2)*cos(pi/2)        sin(-pi/2)*sin(pi/2)         0*cos(-pi/2)
    sin(-pi/2)        cos(-pi/2)*cos(pi/2)          -cos(-pi/2)*sin(pi/2)        0*sin(-pi/2)
    0            sin(pi/2)            cos(pi/2)            440
    0            0                0               1]

T4 =

    0.0000    0.0000   -1.0000         0
   -1.0000    0.0000   -0.0000         0
         0    1.0000    0.0000  440.0000
         0         0         0    1.0000

>> T5=  [cos(pi/2)       -sin(pi/2)*cos(-pi/2)        sin(pi/2)*sin(-pi/2)         0*cos(pi/2)
    sin(pi/2)         cos(pi/2)*cos(-pi/2)          -cos(pi/2)*sin(-pi/2)        0*sin(pi/2)
    0            sin(-pi/2)           cos(-pi/2)           0
    0            0                0               1]

T5 =

    0.0000   -0.0000   -1.0000         0
    1.0000    0.0000    0.0000         0
         0   -1.0000    0.0000         0
         0         0         0    1.0000

>> T6 = [cos(-pi/2)      -sin(-pi/2)*cos(pi/2)        sin(-pi/2)*sin(pi/2)         0*cos(-pi/2)
    sin(-pi/2)        cos(-pi/2)*cos(pi/2)          -cos(-pi/2)*sin(pi/2)        0*sin(-pi/2)
    0            sin(pi/2)            cos(pi/2)            80
    0            0                0               1]

T6 =

    0.0000    0.0000   -1.0000         0
   -1.0000    0.0000   -0.0000         0
         0    1.0000    0.0000   80.0000
         0         0         0    1.0000
```

Figure 12: Resolution numeric of Mathematical Calculations.

```
>> TR=T1*T2*T3*T4*T5*T6

TR =

    1.0000   -0.0000    0.0000 -485.0000
    0.0000   -1.0000   -0.0000  -40.0000
    0.0000    0.0000   -1.0000  770.0000
         0         0         0    1.0000
```

Figure 13: Resolution numeric of Mathematical Calculations.

Figure 14 shows the product of the multiplication of the transformation matrices, that is, the resultant of the product of the matrices, in this case the solution for the robot's direct kinematics.

```
>> syms theta6 alpha6 A6 D6
T6L = [cos(theta6)   -sin(theta6) *cos(alpha6)    -sin(theta6) *sin(alpha6)    A6*cos(theta6)
sin(theta6) cos(theta6)*cos(alpha6) -cos(theta6)*sin(alpha6)     A6*sin(theta6)
0   sin(alpha6) cos(alpha6) D6
0   0   0   1
]

T6L =

[cos(theta6), -cos(alpha6)*sin(theta6), -sin(alpha6)*sin(theta6), A6*cos(theta6)]
[sin(theta6),  cos(alpha6)*cos(theta6), -sin(alpha6)*cos(theta6), A6*sin(theta6)]
[          0,              sin(alpha6),              cos(alpha6),             D6]
[          0,                        0,                        0,              1]
```

Figure 14: Product of the multiplication of the transformation matrices.

Using a mathematic method on MATLAB© was calculated the Jacobian matrix 6x6. The numeric matrix result for the Jacobian is shown in the Figure 15. The Figure 16 shows the resulted found for the Jacobian, one matrix 6x6. The Jacobian matrix represent the 6 axes of the GP7 with the linear velocity and angular velocity, these details are presented in Figure 17

```
J =

   -0.0000         0         0         0         0         0
    0.0000         0         0         0    0.0000   80.0000
 -770.0000  440.0000  440.0000  440.0000         0         0
    0.0000         0         0         0    1.0000         0
   -0.0000         0         0         0   -0.0000   -1.0000
   -1.0000    1.0000    1.0000    1.0000    0.0000    0.0000
```

Figure 15: Jacobian matrix 6x6 resulted from Mathematical Calculations on MATLAB©

$$J = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \\ -770 & 440 & 440 & 440 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{vmatrix}$$

Figure 16: Jacobian matrix 6x6

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	
J =	0	0	0	0	0	0	V = Linear velocity
	0	0	0	0	0	80	
	-770	440	440	440	0	0	
	0	0	0	0	1	0	W= Angular velocity
	0	0	0	0	0	-1	
	1	1	1	1	0	0	

Figure 17: Jacobian description, Axis 1-6, linear velocity and angular velocity.

The Jacobian literal has been resolved and is presented in the attached file, developed in MATLAB©.

Results of Validation

During the validation were checked the D-H and Jacobian for the robot Fanuc us present on item 3.3, the table 4 present the D-H numerical, the results of the Kinematic direct is shown in Figure 18 and Jacobian results from MATLAB©, shown in Figure 19 and Jacobian matrix with Axis and Linear and Angular velocity, shown in Figure 20.

```
DH FANUC
>> T1 = [cos(0)        -sin(0)*cos(0)        sin(0)*sin(0)       50*cos(0)
   sin(0)         cos(0)*cos(0)        -cos(0)*sin(0)      50*sin(0)
   0          sin(0)             cos(0)           330
   0          0              0            1]

T1 =

     1     0     0    50
     0     1     0     0
     0     0     1   330
     0     0     0     1

>>  T2= [cos(pi/2)      -sin(pi/2)*cos(0)      -sin(pi/2)*sin(0)        330*cos(pi/2)
   sin(pi/2)          cos(pi/2)*cos(0)      -cos(pi/2)*sin(0)                  330*sin(pi/2)
   0        sin(0)          cos(0)          0
   0        0           0          1]

T2 =

   0.0000   -1.0000         0    0.0000
   1.0000    0.0000         0  330.0000
        0         0    1.0000         0
        0         0         0    1.0000
>> T3=  [cos(0)      -sin(0)*cos(0)       -sin(0)*sin(0)       35*cos(0)
   sin(0)       cos(0)*cos(0)        -cos(0)*sin(0)       35*sin(0)
   0        sin(0)           cos(0)          145
   0        0            0           1]

T3 =

     1     0     0    35
     0     1     0     0
     0     0     1   145
     0     0     0     1

>> T4 = [cos(-pi/2)      -sin(-pi/2)*cos(pi/2)        sin(-pi/2)*sin(pi/2)        0*cos(-pi/2)
   sin(-pi/2)       cos(-pi/2)*cos(pi/2)         -cos(-pi/2)*sin(pi/2)       0*sin(-pi/2)
   0            sin(pi/2)            cos(pi/2)            190
   0            0                0             1]

T4 =

   0.0000    0.0000   -1.0000         0
  -1.0000    0.0000   -0.0000         0
        0    1.0000    0.0000  190.0000
        0         0         0    1.0000
>> T5 = [cos(pi/2)       -sin(pi/2)*cos(-pi/2)        sin(pi/2)*sin(-pi/2)        0*cos(pi/2)
   sin(pi/2)        cos(pi/2)*cos(-pi/2)         -cos(pi/2)*sin(-pi/2)       0*sin(pi/2)
   0            sin(-pi/2)           cos(-pi/2)           0
   0            0                0             1]

T5 =

   0.0000   -0.0000   -1.0000         0
   1.0000    0.0000    0.0000         0
        0   -1.0000    0.0000         0
        0         0         0    1.0000

>> T6=  [cos(-pi/2)      -sin(-pi/2)*cos(pi/2)        sin(-pi/2)*sin(pi/2)        0*cos(-pi/2)
   sin(-pi/2)       cos(-pi/2)*cos(pi/2)         -cos(-pi/2)*sin(pi/2)       0*sin(-pi/2)
   0            sin(pi/2)            cos(pi/2)            190
   0            0                0             1]

T6 =

   0.0000    0.0000   -1.0000         0
  -1.0000    0.0000   -0.0000         0
        0    1.0000    0.0000  190.0000
        0         0         0    1.0000
```

Figure 18: Fanuc resolution numeric of kinematics direct.

```
>> %JACOBIANO
>> J=[z0.*(t6-t0) z1.*(t6-t1) z2.*(t6-t2) z3.*(t6-t3) z4.*(t6-t4) z5.*(t6-t5); z0 z1 z2 z3 z4 z5]

J =

    0.0000         0         0         0         0  190.0000
   -0.0000         0         0         0         0         0
 -665.0000  335.0000  335.0000  190.0000         0         0
   -0.0000         0         0         0         0   -1.0000
   -0.0000         0         0         0   -1.0000   -0.0000
   -1.0000    1.0000    1.0000    1.0000    0.0000    0.0000
```

Figure 19: Jacobian resolution numeric of Mathematical Calculations.

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	
J =	0	0	0	0	0	190	V= Linear velocity
	0	0	0	0	0	0	
	-665	335	335	190	0	0	
	0	0	0	0	0	-1	W= Angular velocity
	0	0	0	0	-1	0	
	-1	1	1	1	0	0	

Figure 20: Fanuc Jacobian matrix with axis and velocity's.

Conclusion

To conclude the study, we carried out the development of the equations of the forward and inverse kinematics of the robot YASKAWA MOTOMAN - GP7, equations that are part of the Denavit-Hartenberg theorem "D-H" in order to reach the Jacobian matrix equation of the manipulator in question.With the multiplication of the DH matrices, we obtained the homogeneous transformation matrix, which provides the mapping of the coordinates from the base to the end of the tool. Then we arrive at the Jacobian matrices to perform the calculations of angular and linear velocities. The study shows that through the calculations performed, we obtained a satisfactory performance, similar to the results obtained in real tests carried out in the robotic manipulator of the FANUC brand located in the mobile unit of the IAR. All results are within the oscillation limits, knowing that in both situations the robots were in their "home" position. We can conclude that the analyzes performed can be applied to any six-axis industrial robot. This work presents a contribution to the Advanced Institute of Robotics, as they were competencies and skills acquired in the curricular unit of the discipline of modeling robotic manipulators of the postgraduate course in Robotic Engineering.

References

1. CRAIG, J., J. (2006). Robotics, 3rd ed., Pearson Education, Mexico.
2. Nethery, J. F., & Spong, M. W. (1994). Robotica: A mathematica package for robot analysis. IEEE Robotics & Automation Magazine, 1(1), 13-20.
3. Niku, S. (2010). Introduction to Robotics. New Jersey: John Wiley & Son.
4. Hasan, A. T., Hamouda, A. M. S., Ismail, N., & Al-Assadi, H. M. A. A. (2006). An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot manipulator. Advances in Engineering Software, 37(7), 432-438.
5. Craig, J., J. (2012). Robotics, 3rd ed. São Paulo: Editora Pearson.
6. SÁ, C., E. (1996). Implementation of numerical methods for solving the inverse kinematic problem of robotic manipulators with emphasis on position control, master's thesis, Unicamp.
7. William S.. Dorn, & McCracken, D. D. (1972). Numerical methods with FORTRAN IV case studies. John Wiley & Sons.
8. Hartenberg, R., & Danavit, J. (1964). Kinematic synthesis of linkages. New York: McGraw-Hill.
9. Melek, W. (2010). ME 547: robot manipulators: kinematics, dynamics, and control.
10. Jakimovski, B. (2011). Biologically inspired approaches for locomotion, anomaly detection and reconfiguration for walking robots (Vol. 14). Springer Science & Business Media.
11. Hasan, A. T., Hamouda, A. M. S., Ismail, N., & Al-Assadi, H. M. A. A. (2006). An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot manipulator. Advances in Engineering Software, 37(7), 432-438.
12. Paul, R. P. (1981). Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. Richard Paul.
13. Kohli, D., & Soni, A. H. Synthesis of spherical mechanisms for multiply-separated positions of a rigid body. In Proc. of 1FToMM Int. Symp. on Linkages.
14. Uicker Jr, J. J., Denavit, J., & Hartenberg, R. S. (1964). An iterative method for the displacement analysis of spatial mechanisms.
15. Lee, C. S. G., & Ziegler, M. (1984). Geometric approach in solving inverse kinematics of PUMA robots. IEEE Transactions on Aerospace and Electronic Systems, (6), 695-706.
16. YASKAWA ELETRIC,(2016) CORPORATION – Motoman-GP Series.
17. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). Robot modeling and control. John Wiley & Sons.
18. MATHWORKS, 2022. https://www.mathworks.com/products/matlab/student.html , accessed July 26, 2022.
19. https://www.fanucamerica.com/docs/default-source/fanuc-robot-datasheets-new/datasheet-lrmate-200id.pdf