VOL. 12, NO. 7, APRIL 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

KINEMATIC MODELLING OF A ROBOTIC ARM MANIPULATOR USING MATLAB

Ruthber Rodríguez Serrezuela[1], Adrián Fernando Chávarro Chavarro[2], Miguel Angel Tovar Cardozo[1], Alejandro Leiva Toquica[1] and Luis Fernando Ortiz Martinez[1]
[1]Faculty of Biomedical, Electronic and Mechatronics Engineering, Universidad Antonio Nariño (UAN), Bogota, Colombia
[2]Department of Electronic and Telecommunications, Tecnoparque SENA, Bogota, Colombia
E-Mail: ruthbrodriguez@uan.edu.co

ABSTRACT

In this paper the design and implementation of a kinematic model for a manipulator robot arm type with four degrees of freedom is developed, model robot performance can be checked mathematically using results from coordinate's frames, which set the proposed matrices by Denavit-Hartemberg method to determine the robot joins angle vector. This procedure describes the direct and inverse kinematics. The goal is to determine the final robot´s position and orientation according to the joint angles related to a coordinate system, the final effector position, where joint angles are located. The results were implementedin a MATLAB application that performs fast calculations, it allows the verification of the theory and at the same time becomes as a tool to simplify the analysis and learning for its friendly interface which displays virtually the movements of the robotic arm AL5A.

Keywords: inverse and direct kinematics, robot arm.

1. INTRODUCTION

In recent years, virtual reality has been delivering great contributions in various fields of science and technology. Projects related to the design and implementation of a kinematic model for a robotic arm four or more degrees of freedom have been achieved successfully, implementing analysis and kinematic model of a robotic arm of 5 degrees of freedom giving as an objective modeling of direct and inverse kinematics of a robotic arm from a theoretical and practical experience in robotic systems, automation and control. [1], [2], [3]. Likewise, the development of software for the kinematic analysis of a robotic arm called Lynx 6, which suggests more effective methods to reduce multiple inverse kinematics solutions described. A visual software package called MSG is also developed to test the characteristics of arm movement [4], [5], [6]. In the same way, kinematic analysis for a robot arm based on a prototype with three degrees of freedom is presented. It uses an application that allows the program run on the card, receive data and operate allowing the clamp to be moved to a desired position [7], [8], [9].

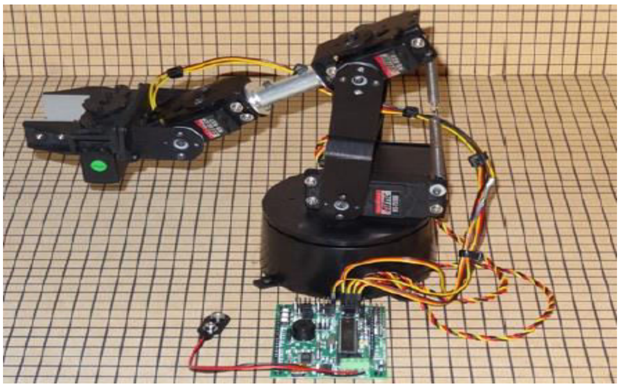

Figure-1. Prototype robotic arm manipulator with four degrees of freedom.

2. MATERIALS AND METHODS

There are different mechanical aspects related to Robotic Arm, to understand the physics of this machine.

Kinematics: Kinematics is the part of physics that is responsible for studying the movement of bodies regardless of their cause, which essentially takes as a reference trajectory in function of time.

Kinematic model: It presents the algorithm of Denavit-Hartenberg as a method by which it is possible to represent the robotic arm using homogeneous transformation matrices.

Homogeneous transformation matrix: The homogeneous transformation matrix established as a 4x4 matrix allows to know the location, position and orientation of an axis system of coordinates$OUVW$ related to the fixed coordinates$OXYZ.$

Direct Kinematics: For using the direct kinematics the vector and matrix algebra is used based on the spatial location of an object in three dimensional space fixed reference. Because the robot is regarded as a kinematic chain consisting of links joined together by means of joints, can create a reference system on the base that describes the location of the links in reference to the system described.

Inverse Kinematics: The inverse kinematics seeks the values that should adopt the robot joint coordinates $\boldsymbol{q} = [\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots . . \boldsymbol{q_n}]$ to position the robot end effector so that its end is oriented in a particular spatial location. To obtain the equations is required a procedure which is entirely dependent on the configuration of the robot. Thus, as you should obtain the values of the joint

ARPN Journal of Engineering and Applied Sciences

variables so that the end effector has a certain position and orientation.

Multiple solutions should be considered, as minimize movements made by the arm to the current position, the concept of allows the nearest solution, to move the links of lower weight, and obstacles are considered important to avoid collisions.

Some generic procedures that can be programmed have been developed so that a computer can, from knowledge of the kinematics of the robot (with its parameters Denavit - Hartenberg) to obtain the n-tuple of joint values that position and orient its end. The disadvantage of these processes is that it is iterative numerical methods, whose convergence speed and even S_ convergence is not always guaranteed. When solving the inverse kinematic problem is much more suitable to find the solution closed. This is to find a mathematical relationship explicit form. $Qk = Fk\,(x, y, z)\ K =$ 1)(D.O.F.) [10].

To this solution, the kinematic problem reverse is solved in real time, so has to follow a certain path. To a solution in iterative time there is no guarantee that the solution develops properly. Otherwise what will happen with direct kinematic problem, quite often inverse kinematics is not unique but there are different joints that guide and position the end of the robot.

Position representation: The representation of the position is determined by various systems that allow locate a point in space. Thus systems Cartesian, cylindrical and spherical coordinates are part of this important method. This article will use the Cartesian coordinate system.

The reference system X, Y, Zallows the representation of a point in space from the position vector$\boldsymbol{P}$, so that the equation describes a column vector of three elements;

$$P = (Px, Py, Pz)^T \tag{1}$$

Likewise, the location of point P can be represented by another coordinate system A, B, C. The new coordinate system, depends on the source location relative to system taken as the main reference X, Y, Z and orientation representing the coordinate axis. This figure shows the

Cartesian system A, B, C whose origin is at the point PsystemX, Y, Z.

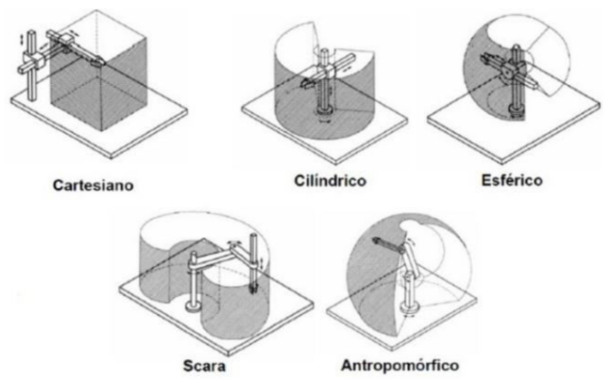

Figure-2. Robot arm achievable space [11].

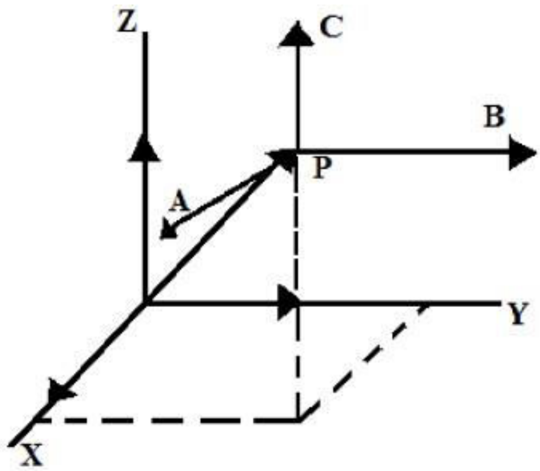

Figure-3. Representation of ABC system with P point in space.

Representation orientation: Point for a spatial location is described by a three-dimensional coordinates. For a solid body it is necessary to know the orientation of it related to a reference system.

It is important to unite the solid body coordinate system A, B, C that be described in space. Therefore, the orientation will take a direction of the unit vectors of the coordinate system attached to the body associated with the axes of the source coordinate $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$.

Rotation matrix: This method solves the problem of orientation, because it allows to use matrix algebra tools to establish relative rotation between main axes of two systems sharing the source.

The rotation matrix is defined as a $3X3$ matrix R which is used as an operator. The equation enables a transformation vector P on a position in three dimensional space. Thus, the coordinates become expressed in a system of coordinate axes rotated A, B, C.

$$P_{UVW} = (P_U, P_V, P_W)^T \tag{2}$$

The orientation is defined by rotating angles to the main axis of fixed reference.

ARPN Journal of Engineering and Applied Sciences

$$P_{xyz} = RP_{uvw} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = R\begin{pmatrix} P_u \\ P_y \\ P_z \end{pmatrix} \quad (3)$$

Rotation matrix operations: Product: The coordinate system has different eventualities in a solid body due to experience more than one rotation about different axes of the source system. To obtain such rotations it is necessary multiply the rotation matrices representing individual rotations, achieving a composite matrix rotation.

The multiplication of matrices is not commutative, so these must be multiplied in the same order in which they spend the rotations of the system in space.

Location in three-dimensional space: The solids in three dimensional space can have movements of rotation and translation. These allow the coordinate system match a solid body translation vector to determine its position and a rotation matrix indicating the orientation linked to a fixed reference system. The relationship of this concept is defined in a 4x4 matrix array called homogeneous transformation (H.T.M.).

Homogeneous transformation matrices: H.T.M.4x4 represents the position and orientation of a moving three-dimensional system related to a fixed three-dimensional system.

A H.T.M. consists of four subarrays of different sizes:
R3x3 submatrix corresponding to a rotation matrix.
P3x1 submatrix corresponding to a translation vector.
F1x3 represents submatrix perspective transformation.
W1x1 submatrix corresponds to a global scale.

Operation with homogeneous transformation matrices: Product Multiplying several MTH have a sequence of positions, which results in a trajectory generation step. This result is called homogeneous transformation matrix composite, which depends on the order in which the products are made.

Inverse transformation: The homogeneous transformation matrix consists of four subarrays, of which both the perspective scaling should be constant, so cannot apply the transpose.

Denavit-Hartenberg representation: The Denavit-Hartenberg representation is a method that applies the properties of MTH to represent the relations of translation and rotation between adjacent elements of a Robot. In 1955 Denavit and Hartenberg proposed an algorithm to establish systematically a coordinate system if linked to each link i of an articulated chain, this allows moving from one link to the next by 4 basic transformations that rely exclusively on the geometrical characteristics of the link [11], [12].

Denavit-Hartenberg algorithm: Zi axis coincides with the hinge axis towards the displacement for a prismatic joint and likewise the direction of the rotational axis for the joint. For the Xi axis perpendicular location takes a common reference Zi and Zi + 1.

Determining the Denavit-Hartenberg Parameters: The Denavit-Hartenberg parameters define the dimensional relationships between consecutive links and joint variables.

Θi	is the variable angle joint for a joint rotation.
di	is the distance of the variable joint for a prismatic joint.
α_{i-1}	is the distance from the axis of joint $i-1$ to i joint axis measured along the perpendicular line common to these axes. When the axes $\alpha_{i-1}$are cut be zero.
α_{i-1}	is the angle between the axis of joint $i-1$ and i joint axis; measured as a rotation axis $i+1$ around the common perpendicular until it coincides with the direction of the axis i.
di	indicates the time of articulation. It is assigned zero value for rotational joints and one for the prismatic joints [13], [14], [15].

Meaning Denavit-Hartenberg representation: D-H parameters provide information about the number and type of joints of a robot. It is thus that disclosed dimensions of the links, the link between consecutive angular joints and the separation between them. Also, determine the joint variables of the Robot.

Transformation matrix D-H: Being determined the representation D-H it results in the successive transformations corresponding to the ith system coordinate with the coordinate system$(i-1)th$. It performs basic operations MTH obtaining a composite $Ai-1$ transformation matrix called D-H. The matrix presented below represents the function of each of the D-H parameters in obtaining the matrix.

$$^{i-1}A^{i} = T_{z,dT_{z,\emptyset}} T_{z,\alpha} T_{x,\alpha} \quad (4)$$

$$= \begin{vmatrix} \cos\emptyset i & -\cos\alpha i\,\mathrm{sen}\emptyset i & \mathrm{sen}\alpha i\,\mathrm{sen}\emptyset i & \alpha i\cos\emptyset i \\ \mathrm{sen}\emptyset i & \cos\alpha i\cos\emptyset i & -\mathrm{sen}\alpha i\cos\emptyset i & \alpha i\,\mathrm{sen}\emptyset i \\ 0 & \mathrm{sen}\alpha i & \cos\alpha i & di \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Interpretation matrix transformation D-H: The matrix^{i-1}Aindicates that A represents a MTH corresponding to the coordinate axis system (i) linked to the joint (i) related to the coordinate axis system $(i-1)$ binding articulation $(i-1)$ .The matrix obtained is determined by a degree of freedom.

Arm matrix: For a robot with n degrees of freedom, the equation below shows the possible matrices D n-H multiplying sequentially for $^{0}T_{n}$. The end effector

VOL. 12, NO. 7, APRIL 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

MTH referenced to the source system of the robot base is defined by:

$$ {}^{0}A_1 *^{1} A_2 *^{2} A_3 *^{3} A_4 * \ldots *^{n-1} A_n =^{0} A_n =^{0} T_n $$

${}^{0}T_n$ matrix represents the end of the powertrain, thus comprises all operations performed on the robotic arm to perform a coordinate system from the end to the source. That is why it is called matrix or matrix Tn arm as shown below.

Parameters (Denavit - Hartenberg) (DH): The axes of the coordinates for the robot, allow obtaining the Denavit-Hartenberg parameters of.

Table-1. D-H parameters for AL5Arobot arm.

Join	Θ	D	A	α
1	q1	L1	0	90°
2	q2	0	L2	0
3	q3	0	L3	0
4	q4	0	L4	90°

Conventionstable:

- **θi (length):** Represents the offset distance between Xy-1 and Zi axes along the axis Xi.
- **αi (rotation angle):** The angle from the axis Zi Zi-1 axis about the axis Xi.
- **di (Offset distance):** The distance from the origin of the joint i-1 to the axis along the axis xi Zi-1
- **θi (rotation angle):** The angle between the axes X1 and X about axis Zi-1.

Transformation matrices (D-H): Parameters (D-H) seen in the result of the axes of coordinates AL5A arm frame, the values are replaced in the transformation matrix for each link.

Link 1:
Basismatrix:

$$ A1 = \begin{pmatrix} c1 & 0 & s1 & 0 \\ s1 & 0 & -c1 & 0 \\ 0 & 1 & 0 & L1 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Link 2:
Shouldermatrix:

$$ A2 = \begin{pmatrix} c2 & 0 & -s2 & L2c2 \\ s2 & 0 & c2 & L2s2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Link 3:
Elbowmatrix:

$$ A3 = \begin{pmatrix} c3 & -s3 & 0 & L3c3 \\ s3 & c3 & 0 & L3s3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Link 4:
Wristmatrix:

$$ A2 = \begin{pmatrix} c4 & 0 & s4 & L4c4 \\ s4 & 0 & -c4 & L4s4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Direct kinematic problem solution: Calculating the position and orientation of the end effector with given angle of each of the joints is called the direct kinematics analysis. The direct kinematics equations are generated from transformation matrices.

The solution of the direct kinematics of the arm is multiplying transformation matrices.

Direct Kinematics for the home position of the arm AL5A:

Parameters (D-H) in position Home: Taking the values of the initial position the following values are obtained:

Table-2. D-H parameters for AL5Arobot armin home position.

Join	θ	D	A	α
1	0	6,8	0	90°
2	0	0	9,3	0
3	0	0	10,9	0
4	0	0	8,5	90°

Transformation matrices (D-H) to the home position of the robot arm: Parameters (D-H) seen in the result of the coordinate axes of the arm in the initial position AL5A Table-2, the values are replaced in the transformation matrix for each link.

Inverse Kinematics: A method for solving the inverse kinematics is the method of Pieper, which states that once positioned at a point in space wrist in which the three axes intersect, the movement of the joints of the same around their axes do not alter the spatial position of the cutting point, or what is the same wrist. Therefore I can solve the inverse kinematic problem only for the position of the point of intersection of the axes of the robot wrist, ie, for the first three joints are those positioned in space at the wrist, since the corresponding guidance occurs the final position will not change [16], [17], [18]. For the first 3 joints could be used, the algebraic or geometric solution method.

ARPN Journal of Engineering and Applied Sciences

Geometric solution: The use of Cartesian mode, allows the user to specify the desired target position of the clamp in Cartesian space (x, y, z), where $z$the height and angle of the clamp is taken to ground remains constant. This constant allows the user to move objects without changing the orientation of the object. Furthermore, either by keeping the gripper in a fixed position or keeping the wrist fixed relative to the arm rest, the inverse kinematics equations can be solved in closed form.
The θ_1, θ_2, $\theta_3$and $\theta_4$angles correspond to the rotation of the shoulder, arm, forearm and wrist respectively. It solves part of joint angles arm, $\theta 1{:}\ 4$ given the required position (x, y, z) and land which are inserted by the user. In the figure, we see clearly that$\boldsymbol{\theta}1 = Atan2(y/x)$and specified radial distance from the base l are related to x, y through:

$$l = \sqrt{x_l^2 + y_l^2} \tag{7}$$

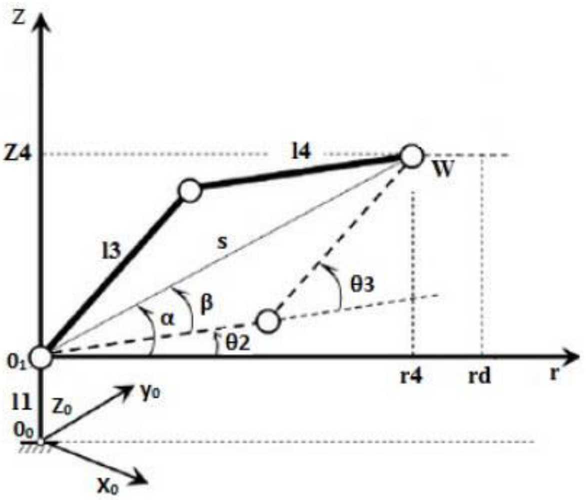

Figure-4. Plan view of the robot.

Pieper solution
From the base to the wrist

$$\mathrm{P}_3^0 = \mathrm{P}_{\mathrm{wrist}}^0(\mathrm{X}_3^0, \mathrm{Y}_3^0, \mathrm{Z}_3^0) \tag{5}$$

From the base to the end

$$\mathrm{P}_3^0 = \mathrm{P}_{\mathrm{exteme}}^0(\mathrm{X}_{\mathrm{extreme}}^0, \mathrm{Y}_{\mathrm{extreme}}^0, \mathrm{Z}_{\mathrm{extreme}}^0)$$

Is fulfilled:

$$\mathrm{P}_3^0 = \mathrm{P}_4^0 - d_4.\mathrm{z4} \tag{6}$$

And considering that the coordinates of $Z4$ respect to the base system is part of the desired orientation for the end of the robot, the position of the wrist is determined.

$$\mathrm{X}_3^0 = \mathrm{X}_{\mathrm{wrist}}^0 = \mathrm{X}_4^0 - d_4.\mathrm{r13}$$

$$Y_3^0 = \mathrm{Y}_{\mathrm{wrist}}^0 = \mathrm{Y}_4^0 - d_4.\mathrm{r23}$$

$$\mathrm{Z}_3^0 = \mathrm{Z}_{\mathrm{wrist}}^0 = \mathrm{Z}_4^0 - d_4.\mathrm{r33}$$

Values that will be taken to solve the inverse kinematics problem for the first three robot joints.

Finally, it remains to determine the values of the last three joints that will guide the end of the robot to the inverse kinematic problem is completely solved. Of course the inverse kinematic problem solved for the first three joints is also known submatrix $((Rot)_3^0)$, since it will be the only function of these first three joints and they are already determined. So:

$$(Rot)_{extreme}^0 = (Rot)_3^0.(Rot)_{extreme}^3$$

$$(Rot)_{extreme}^3 = (Rot)_4^3 =$$

$$((Rot)_0^3)^{-1}.(Rot)_{extreme}^0 =$$

$$((Rot)_3^0)^T.(Rot)_{extreme}^0 \tag{8}$$

Another possible solution is the algebraic

Algebraic solution: The resulting equation can also be obtained by this method. The amplification procedure is similar to the four articulation θ4:

$$(T_{01} * T_{12} * T_{23})^{-1} G = T_{34}$$

$$\begin{pmatrix} * & * & c_{123}r_{13} + c_{23}s_1r_{23} + s_{23}r_{33} & * \\ * & * & -s_{23}c_1r_{13} - s_{123}r_{23} + c_{23}r_{33} & * \\ * & * & * & * \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} (c_4 - 1) & -(c_4 - 1) & -s_4 & -s_4 + a_4 \\ s_4 & -s_4 & c_4 & c_4 \\ * & * & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$s_4 = -(c_{123}r_{13} + c_{23}s_1r_{23} + s_{23}r_{33})$$

$$c_4 = -(+s_{23}c_1r_{13} + s_{123}r_{23} - c_{23}r_{33})$$

As a result:
$\theta_4 = atan2(s_4, c_4)$

3. RESULTS

Work environment: The importance of the environment regarding the functional capacity of the machine determines that once posed an application seeking to resolve the problem proposed adapting or creating a working environment for the robot to provide, to the extent possible, the design of the solution final. Given a working environment must decide the degree of autonomy and "intelligence" that will be necessary to equip the robot to carry out the proposed application.

VOL. 12, NO. 7, APRIL 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

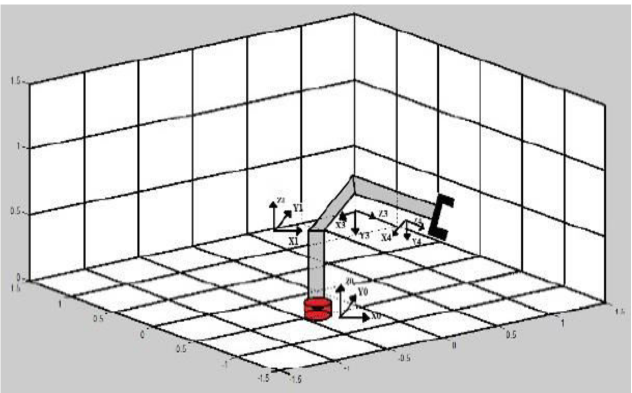

Figure-5. Design work environment.

The working environment that has been chosen for this project consists of three parts, whose dimensions are 60cm x 60cm given by x 59cm. This environment consists of the robot AL5A which from its home position allows easy movement through space.

Botboarduino interface card with MATLAB: The main disadvantages are presented to achieve communication between MATLAB and Arduino are DLLs. These are necessary in MATLAB to manage the USB port. For this reason, the implementation of which MATLAB serial port recognized as COM port and manages to be obtained from a number of communication ports to use is necessary.

The Botboarduino card is programmed via USB serial ports recognizing where a serial to USB converter within itself facilitating communication intervenes. For the implementation of these two programs it was necessary to establish the following steps for proper operation. Load the sketch on arduino to obtain a correct data acquisition.

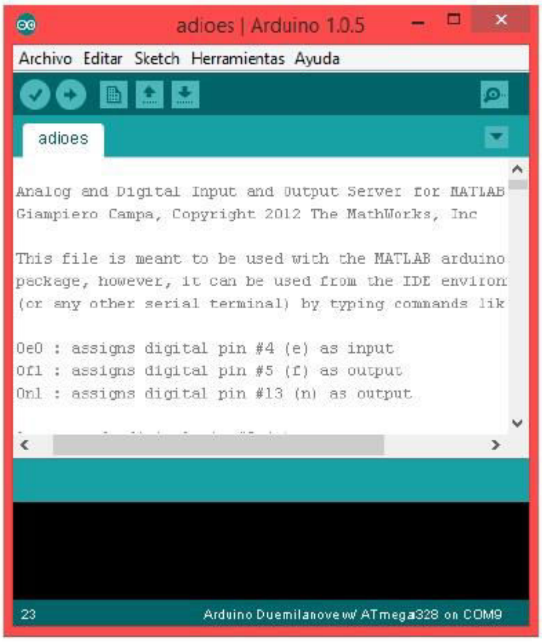

Figure-6. Sketch Arduino-MATLAB.

The procedure to load the skit on the Botboarduino card. Once the file is loaded, MATLAB opens for communication. Redirect the Arduino IO folder in the current folder or current MATLAB folder Select the install file - arduino. m and within the Windows command write installarduino. Finally run the program in MATLAB.

Implementation of the kinematic model in MATLAB: Proposed by Peter Corke, in his book robotics toolbox was used to implement the kinematic model in MATLAB; Control and vision. Which he underwent some adjustments to fit the model AL5A Robot arm?

Robotics toolbox: This toolbox provides many features that are useful for the study and simulation of classic type robotic arm, for example things like kinematics, dynamics and trajectory generation. The toolkit is based on a very general method of representing the kinematics and dynamics of manipulators serial links. These parameters are encapsulated in MATLAB objects - objects robot can be created by the user to any serial-link manipulator and a number of examples are provided to as__ know robots such as Puma arm 560 and Stanford, among others. The toolbox also provides functions for handling and conversion between data types, such as vectors, homogeneous and unit-quaternions transformations that are necessary to represent the position and orientation of 3 dimensions.

Implemented in MATLAB code: Starts design environment GUI (GUIDE) and allows you to select which type of calculation needs to be made; if direct or inverse kinematics and thus call the function to conduct the calculation kinematics.

VOL. 12, NO. 7, APRIL 2017 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A.0.4. Código diseño de la interfaz gráfica.

```
\leftline{
function varargout = AL5A(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @AL5A_OpeningFcn, ...
                   'gui_OutputFcn',  @AL5A_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});

end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function AL5A_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
movegui('center');


function varargout = AL5A_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
cin_directa; % Ejecuta la rutina para manejo de la cinemática directa

function pushbutton2_Callback(hObject, eventdata, handles)
cin_inversa; % Ejecuta la rutina para manejo de la cinemática inversa
}
```

Figure-7. Code for graphical interface design.

Direct Kinematics: Starts Design Environment graphical user interface that allows you to type or by a slider give values to those desired by the users and thus calculate the points angles X, Y, Z to show animated form the final location of the robot arm AL5A.

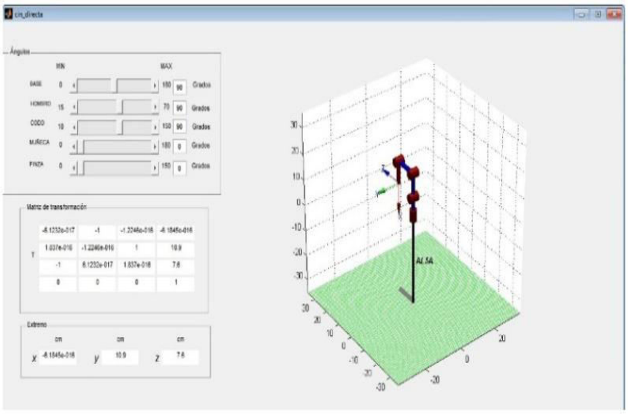

Figure-8. Direct kinematics interface using MATLAB.

Inverse Kinematics: Starts Design Environment graphical user interface that allows you to type or by a slider give values X, Y, Z desired by a user and thus calculate the values of θ angles to plot animated form the final location of the robot arm AL5A.

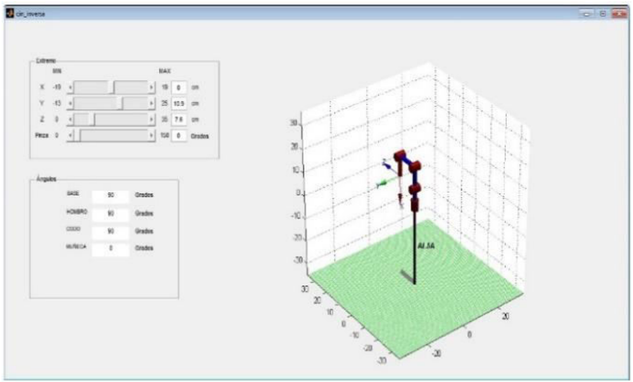

Figure-9. Graphical user interface for inverse kinematics robot arm using Matlab.

4. CONCLUSIONS

Using the Denavit-Hartemberg method (D-H) coordinates of the articulated chain for obtaining the parameters, which were then replaced in the transformation matrix of each link were established for the solution of direct kinematics.

The inverse kinematics is much more complex than the direct kinematics since there is no single analytical solution. Each handler needs a particular method taking into account the structure of the system and restrictions. Various methods such as the method of Pieper, which states that you can solve the inverse kinematic problem only for the position of the point of intersection of the axes of the robot wrist, i.e. for the first 3 joints using the algebraic method or geometric solution. The numerical solution of these equations involves a complexity that leads to a high demand computing by MATLAB, causing delays in the order of 1 to 2 minutes depending on the spatial location of the point to which you want to reach. Because of this, we chose a solution that allowed type iterative find joint angles by trial and error.

To check the direct kinematics and the inverse was developed with a toolbox created by Peter Corke and implemented in MATLAB; Botboarduino using a card like interface, which allowed AL5A, position the robotic arm at points X, Y, Z and mathematically calculate the angles. The result of this project was an experimental educational tool which can be used for the study of robotics, encouraging students to pursue research to improve the application of the kinematic solution in practical leraning.

REFERENCES

[1] Sciavicco L. & Siciliano B. 1996. Modeling and control of robot manipulators (Vol. 8, No. 0). New York: McGraw-Hill.

[2] Azhmyakov V., Martinez J. C., Poznyak A. & Serrezuela R. R. 2015, July. Optimization of a class of nonlinear switched systems with fixed-levels control inputs. In 2015 American Control Conference (ACC) (pp. 1770-1775). IEEE.

[3] Walker I. D., Dawson D. M., Flash T., Grasso F. W., Hanlon R. T., Hochner B. & Zhang Q. M. 2005, May. Continuum robot arms inspired by cephalopods. In Defense and Security (pp. 303-314). International Society for Optics and Photonics.

[4] Azhmyakov V., Rodriguez Serrezuela R., Rios Gallardo A. M. & Gerardo Vargas W. 2014. An Approximations Based Approach to Optimal Control of Switched Dynamic Systems. Mathematical Problems in Engineering.

[5] Sciavicco L. & Siciliano B. 2012. Modelling and control of robot manipulators. Springer Science & Business Media.

[6] Serrezuela R. R., Chavarro A. F. C., Cardozo M. A. T. & Zarta J. B. R. 2016. An Optimal Control Based Approach to Dynamics Autonomous Vehicle. International Journal of Applied Engineering Research. 11(16): 8841-8847.

[7] Siciliano B., Sciavicco L., Villani L. & Oriolo G. 2010. Robotics: modelling, planning and control. Springer Science & Business Media.

[8] Serrezuela R. R. & Chavarro A. F. C. 2016. Multivariable Control Alternatives for the Prototype Tower Distillation and Evaporation Plant. International Journal of Applied Engineering Research. 11(8): 6039-6043.

[9] Everett L., Driels M. & Mooring B. 1987, March. Kinematic modelling for robot calibration. In Robotics and Automation. Proceedings. 1987 IEEE International Conference on (Vol. 4, pp. 183-189). IEEE.

[10] Asada H., Ma Z. D., & Tokumaru H. 1990. Inverse dynamics of flexible robot arms: modeling and computation for trajectory control. Journal of dynamic systems, Measurement and Control. 112(2): 177-185.

[11] Tarn T. J. A. K., Bejczy A. & Yun X. 1987, March. Design of dynamic control of two cooperating robot arms: Closed chain formulation. In Robotics and Automation. Proceedings. 1987 IEEE International Conference on. 4: 7-13. IEEE.

[12] Serrezuela R. R., Villar O. F., Zarta J. R. & Cuenca Y. H. 2016. The K-Exponential Matrix to solve systems of differential equations deformed. Global Journal of Pure and Applied Mathematics. 12(3): 1921-1945.

[13] Fernando T., Jorge P., Pablo G., Santiago P., & Rafael A. 2002. Robots y sistemas sensoriales.

[14] Sánchez J. A. S. 2002. Avances en robótica y visión por computador (Vol. 38). Univ de Castilla La Mancha.

[15] Corke P. 2011. Robotics, vision and control: fundamental algorithms in MATLAB (Vol. 73). Springer.

[16] Serrezuela R. R., Chavarro A. F., Cardozo M. A. 2017. Audio signals processing with digital filters implementation using MyD.S.P, Journal of engineering and applied sciences. Vol. 12, 1.

[17] Ude A., Atkeson C. G. & Riley M. 2004. Programming full-body movements for humanoid robots by observation. Robotics and autonomous systems. 47(2): 93-108.

[18] Stone H. W. 2012. Kinematic modeling, identification, and control of robotic manipulators (Vol. 29). Springer Science & Business Media.

[19] Mao Z. & Hsia T. C. 1997. Obstacle avoidance inverse kinematics solution of redundant robots by neural networks. Robotica. 15(01): 3-10.

[20] Zhu Z., Li J., Gan Z. & Zhang H. 2005. Kinematic and dynamic modelling for real-time control of Tau parallel robot. Mechanism and machine theory. 40(9): 1051-1067.

[21] Khalil W. & Dombre E. 2004. Modeling, identification and control of robots. Butterworth-Heinemann.

[22] Dupont P. E., Lock J., Itkowitz B. & Butler E. 2010. Design and control of concentric-tube robots. IEEE Transactions on Robotics. 26(2): 209-225.