



Conference Paper

Embedded system generating trajectories of a robot manipulator of five degrees of freedom (D.O.F)

Jorge Luis Aroca Trujillo, Jorge Bernardo Ramírez Zarta,
and Ruthber Rodríguez Serrezuela

Corporación Universitaria del Huila, Neiva, Republica de Colombia

Abstract

The main principal of the interaction of the robot with work environments is to position the coordinate system of the final effector in a desired orientation and position. To do this, algorithms must be used to generate and control the coordinated movements of the robot joints. This article will deal with three movements (MOVEJ, MOVES and MOVEC) implemented in the industrial controller CompactRIO, which can be combined so that the robot can develop any task that is scheduled.

Corresponding Author:

Ruthber Rodríguez Serrezuela
ruthbrodriguez@uan.edu.co

Received: 15 November 2017

Accepted: 5 January 2018

Published: 4 February 2018

Publishing services provided
by Knowledge E

© Jorge Luis Aroca Trujillo
et al. This article is distributed
under the terms of the
[Creative Commons Attribution
License](#), which permits
unrestricted use and
redistribution provided that
the original author and source
are credited.

Selection and Peer-review
under the responsibility of the
ESTEC Conference Committee.

Keywords: MCD, MCI, CompactRIO, Slerp.

1. Introduction

This work presents three types of algorithms that generates trajectories that provides the positions that must be followed by a manipulator robot of Five Degrees of Freedom (5GDL) [Kazemi et al., 2016], to execute a specific task. The first algorithm develops a joint path (MOVEJ), which has no relevance to the path that the end effector makes. The other two will generate Cartesian trajectories that will draw with the end effector the shape of a line or an arc (MOVES and MOVEC).

These algorithms will be implemented in a generic industrial controller (CompactRio, figure 1), that is, a controller different from those designed to operate commercial manipulator robots. It will interact with the user through a computer connected remotely by the Ethernet port. Programming was implemented through LabView, which facilitates visualization, creation and modification of tasks that the user would like the manipulator to replicate. The developed interface will contain the reading of the commands, the mathematical modeling found, the algorithms for generating trajectories, the visualization of variables and the position control (PID) for engines [Deshpande et al., 2012], this will be programmed in the chassis of the controller.

OPEN ACCESS

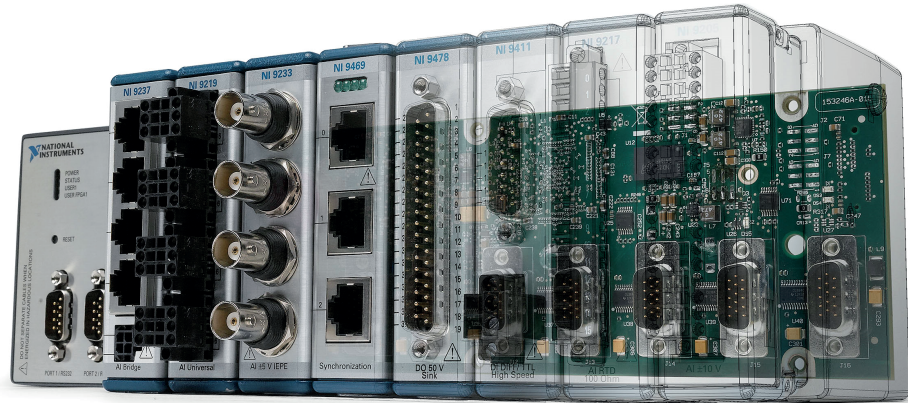


Figure 1: Industrial Controller - NI CompactRIO, [Retrieved on June 8, 2017 from: <http://www.ni.com/compactrio/value-controller/esa/>].

Having programmed the path generator algorithms and the position control system in the CompactRIO controller allows the manipulator to continue the task it is developing when there is a connection failure between the computer and the controller. In this way, there will be an interconnection between the robot manipulator, the industrial controller CompactRIO and the computer designated to modify and visualize the variables present to generate and control the positions and trajectories that the manipulator needs to generate a task.

2. DESCRIPTION OF THE ALGORITHMS OF IMPLEMENTED TRAJECTORIES

Three movements (MOVEJ, MOVES and MOVEC) will be dealt with, which will be implemented in the industrial controller CompactRIO [Alcaraz, 2012], [Parra, 2015], supported in new contributions in the mathematical part [Zarta et al., 2016, 2017], [Serrezuela et al., 2016, 2017], which can be combined so that the robot can carry out a specific task.

2.1. JOINT MOTION: MOVEJ

The use of this movement allows the manipulator to reach an end position, with independent joint movements. This does not ensure a linear trajectory between the initial

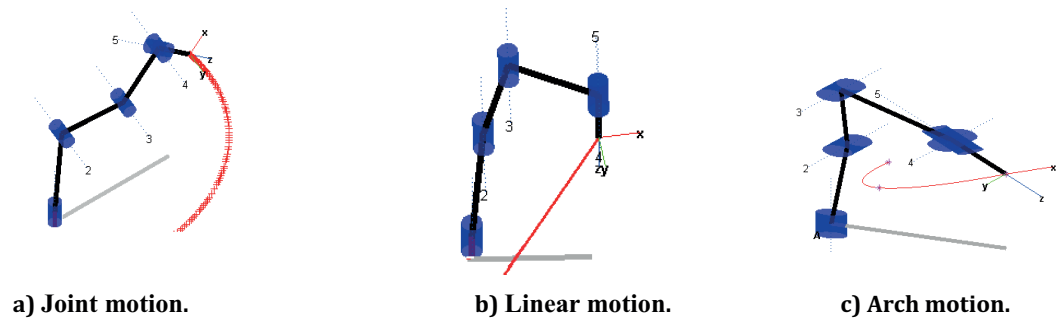


Figure 2: Simulation of joint movement.

and final points, ie, the final effector moves from one point to another by a random path, as shown in Figure 2a.

Algoritmo 1: Algoritmo para el movimiento articular MOVEJ.

```

Entrada: Posición actual:  $[\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]$ ,
Posición deseada:  $[\theta_{1d} \ \theta_{2d} \ \theta_{3d} \ \theta_{4d} \ \theta_{5d}]$ 
Resultado: Posición nueva:  $q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]$ 
1 Mientras  $[\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5] \neq [\theta_{1d} \ \theta_{2d} \ \theta_{3d} \ \theta_{4d} \ \theta_{5d}]$  hacer
2     Si  $\theta_1 \neq \theta_{1d}$  entonces
3          $\theta_1 = \theta_1 \pm \Delta\theta_1$ ;
4     finalizar
5     Si  $\theta_2 \neq \theta_{2d}$  entonces
6          $\theta_2 = \theta_2 \pm \Delta\theta_2$ ;
7     finalizar
8     Si  $\theta_3 \neq \theta_{3d}$  entonces
9          $\theta_3 = \theta_3 \pm \Delta\theta_3$ ;
10    finalizar
11    Si  $\theta_4 \neq \theta_{4d}$  entonces
12         $\theta_4 = \theta_4 \pm \Delta\theta_4$ ;
13    finalizar
14    Si  $\theta_5 \neq \theta_{5d}$  entonces
15         $\theta_5 = \theta_5 \pm \Delta\theta_5$ ;
16    finalizar
17     $q = [q; \theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]$ ;
18 finalizar
19 Nueva posición  $q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5]$ ;

```

The coordinates in the final effector workspace can be obtained through the MCD [Kumar et al., 2015], [Chaudhary et al., 2011], [Constaín et al., 2009], using the joint values executed in the MOVEJ command. Algorithm 1 shows the description of the execution of the MOVES command whose main part of the implementation can be observed in figure 2a.

2.2. LINEAR MOTION: MOVES

If it is desired to arrive at a Cartesian position following a path in a straight line with the final effector, it is made use of the MOVES command. This command requires a

vector with six values, the first three are Cartesian variables (X, Y and Z) of the desired position in the working space of the origin of the coordinate system of the final effector. The following three refers to its orientation, this orientation is entered with the roll (δ), Pitch (β) and Yaw (γ) angles, also known as RPY [Barrientos et al., 1997], [Siciliano et al., 2010] angles.

In this way, a linear motion will be taken as Figure 2b, through the path generated from the current point or from start to end point (point entered with the command).

For this command, we will briefly discuss the two (2) transformations that must be performed. The treatment is done differently, mainly because the space of translations is Euclidean while the space of rotations is not.

Comparing the two movements, the translation is the simplest of the two interpolations, since it is expected that a point $P_o \in \mathbb{R}^3$ and denoted by three (3) variables (x_o , y_o and z_o) defined in the Cartesian space, will be moved through of a translation vector (Δx , Δy and Δz).

The interpolation of orientation and rotation is done by the mathematical SLERP method [Barrera et al., 2004], [Eberly, 2002]. For this, the orientation and rotation must first be represented in four (4) dimensions or quaternions, and then interpolated by the SLERP method. This method is to perform constant velocity motion along a circle on the surface of a hypersphere, if the positions of the trajectories of the final effector are given between the points of P_o and P_f . Is a uniform rotation with angular velocity around a fixed axis of rotation, which guarantees a single route?

Algorithm 2 shows the description of the execution of the MOVES command whose main part of the implementation can be observed in figure 2b.

2.3. ARCH MOTION: MOVEC

Three linearly independent points are indispensable for this movement. The first is the value of X, Y and Z of the Cartesian plane where the final effector (P_o) is initially, the second is the destination point (P_f) and the third belongs to the via point (P_v). The latter, in addition to helping to define the path where the end effector should slide, defines the plane to which it is to be worked, as shown in Figure 2c.

The circumference in the plane is drawn from the perpendicular bisector of P_oP_v and the bisector of P_oP_f where they are cut at a single point, which is the center of the circumference, and passes through P_o , P_v and P_f since the three (3) Are equidistant from it. In the event that the three given points are aligned, the problem results in

Algoritmo 2: Algoritmo para el movimiento articular MOVES.

Entrada: Matrices de transformaciones T_o, T_f
Resultado: Posiciones interpoladas $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]$

```

1  $p = 0 : 1 : (|P_f| - |P_o|)$ ;
2  $N = \text{length}(p)$ ;
3  $t = 0 : (0.1 \frac{N}{N-1}) : N$ ;
4  $P_f = T_f(1 : 3, 4)$ ;
5  $P_o = T_o(1 : 3, 4)$ ;
6  $R_f = T_f(1 : 3, 1 : 3)$ ;
7  $R_o = T_o(1 : 3, 1 : 3)$ ;
8 Para  $i = 1 : 1 : N$  hacer
9      $P_n = P_o + (P_f - P_o) \cdot t(i)$ ;
10     $q_o \leftarrow R_o$ ;
11     $q_1 \leftarrow R_1$ ;
12     $\theta = \arccos(q_o \cdot q_1)$ ;
13     $q_m = q_o \cdot \sin((1 - t(i)) \cdot \Omega) / \sin(\Omega) + q_1 \cdot \sin(t(i) \cdot \Omega) / \sin(\Omega)$ ;
14     $R_n \leftarrow q_m$ ;
15     $T_n = [R_n \ P_n; 0 \ 0 \ 0 \ 1]$ ;
16     $q = [MCI(T_n)] \rightsquigarrow$  Modelo Cinemático Inverso
17 finalizar
18 Posiciones interpoladas  $q = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]$ ;
```

Algoritmo 3: Algoritmo para el movimiento articular MOVEC.

Entrada: P_f, P_v
Resultado: Posiciones interpoladas $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]$

```

1 Vectores a y b  $\mapsto a = (P_f - P_o) \leftrightarrow b = (P_v - P_o)$ ;
2  $x = a / \text{norm}(a) \mapsto$  valor en x asociado al sistema a  $O_1$ ;
3  $z = \text{cross}(a, b)$ ;
4  $z = z / \text{norm}(z) \mapsto$  Vector normal al plano ó z asociado al sistema a  $O_1$ ;
5  $y = \text{cross}(z, x) \mapsto$  valor en y asociado al sistema a  $O_1$ ;
6  $\Theta = \arccos(\text{dot}(a, b) / (\text{norm}(a) * \text{norm}(b))) \mapsto$  Ángulo entre los vectores a y b;
7  $x_{vp} = \text{norm}(b) * \cos(\Theta)$ ;
8  $y_{vp} = \text{norm}(b) * \sin(\Theta)$ ;
9  $X_{fp} = \text{norm}(a)$ ;
10  $m = -x_{vp} / y_{vp}$ ;
11  $x_{cp} = P_{fp} / 2 \mapsto$  x del centro de la circunferencia;
12  $y_{cp} = (y_{vp} / 2) + m(x_{cp} - (x_{vp} / 2)) \mapsto$  y del centro de la circunferencia;
13  $r = \text{norm}([X_{fp} \ 0] - [x_{cp} \ y_{cp}])$ ;
14  $\alpha_1 = -\text{atan2}(y_{cp}, x_{cp}) \Rightarrow \alpha_2 = \pi - \alpha_1$ ;
15  $\sigma = \text{linspace}(\alpha_2, \alpha_1, N)$ ;
16  $y_s = r * \sin(\sigma) + y_{cp} \mapsto$  x interpolado de la circunferencia;
17  $x_s = r * \cos(\sigma) + x_{cp} \mapsto$  y interpolado de la circunferencia;
18  $T = [x \ y \ z \ P_o; 0 \ 0 \ 0 \ 1]$ ;
19 Para  $i = 1 : N$  hacer
20      $P = T * [x_s(i) \ y_s(i) \ 0 \ 1] r$ ;
21      $c = P(1 : 3) - P_o$ ;
22      $x_n = c / \text{norm}(c)$ ;
23      $y_n = \text{cross}(z, x_n)$ ;
24      $Tc = [x_n \ y_n \ z \ P(1 : 3); 0 \ 0 \ 0 \ 1]$ ;
25      $q = [MCI(Tc)] \rightsquigarrow$  Modelo Cinemático Inverso
26 finalizar
27 Posiciones interpoladas  $q = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]$ ;
```

a circumference that tends to infinity. The operations used to implement the MOVEC command are defined in algorithm 3.

3. GENERAL SCHEME OF THE IMPLEMENTED INTERFACE

To integrate the mechanical system of the robot and the industrial controller Com-pactRIO, to the LabView environment, the programming and interconnection scheme of Figure 3 was developed, which is programmed in the industrial controller. This way, it will be ensured that if the communication between the computer and the controller is lost, the manipulator continues to operate with only the industrial controller, until it finishes with the programmed routine.

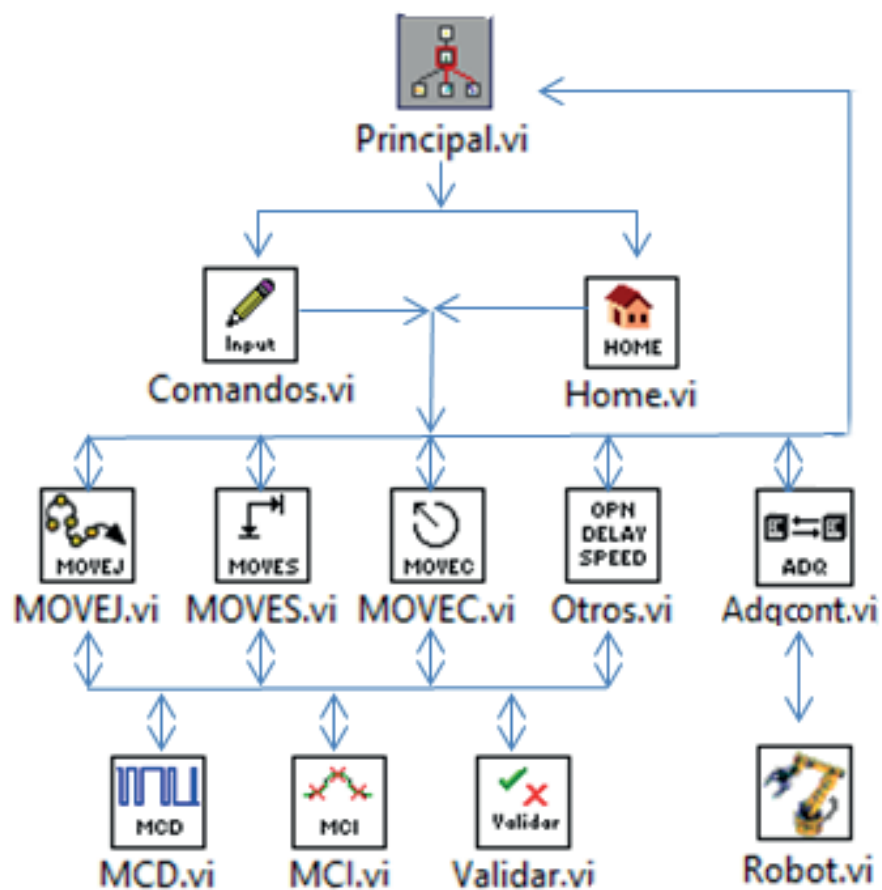


Figure 3: SubVI scheme for robot arm control.

The main VI of the system is completely designed and intended to interact with the user. To do this, it receives and displays the articulator positions of the manipulator, together with the cartesian positions of the final effector. In addition, it sends the commands that are to be executed, to the two blocks responsible for interpreting them (**Comandos.vi** and **Home.vi**).

The VI **Home**, is only responsible for providing the articular positions to bring the manipulator to the initial configuration (**Home**). On the other hand, the VI **Commands**, organizes and sends the other orders to the VIs in charge of executing each of the

algorithms (MOVEJ, MOVES and MOVEC), the desired speed with which to execute each robot movement (SPEED), the Time delay between commands (DELAY) and the opening and closing of the fingers of the clamp (OPEN AND CLOSE).

As the names indicate, the MOVEJ, MOVES and MOVEC VIs, with the help of the MCD, MCI and Validate blocks, are in charge of creating the algorithms generating articular, linear and circular trajectories, respectively. These three first blocks give the point to point the route that must be made to comply with the command, the acquisition block and the control of variables (Adpcont.vi). The VI other, is intended to interpret the commands SPEED, DELAY, OPEN and CLOSE, to later execute them in the manipulator.

4. RESULTS OBTAINED

An analysis of the results of the kinematic model is performed, based on the MOVEJ, MOVES and MOVEC commands. To do this, we start from a known position and generate a path with the different commands implemented. Subsequently a comparison of the data of the movement made by the manipulator and those obtained in the simulation is made.

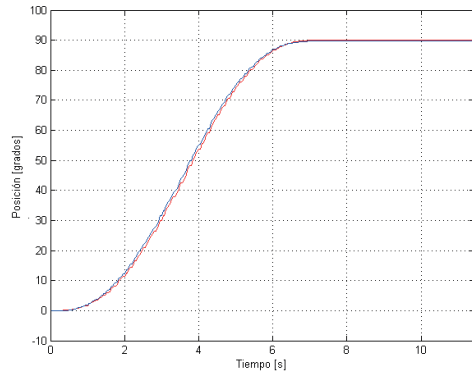
In Figure 4, these joint movements generated by the robot are depicted, where the red stroke represents the expected values and the blue stroke the articular path made by the manipulator.

Figure 4 shows the tendency of the blue line to the red line, which corresponds to the actual movement of the joints following the expected movement. The result shows some position variations, which can be improved by changing the PID controller constants [Castillo et al., 2003].

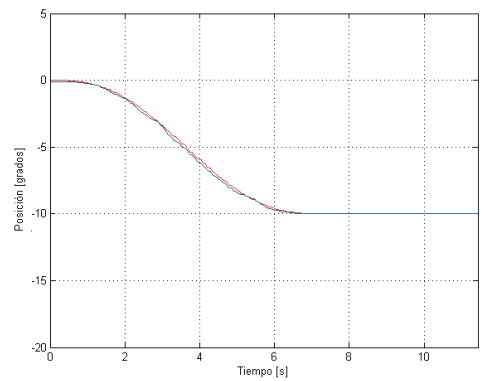
The Cartesian movement of the final effector originated by the coordinated movements of the articulations is represented in the X, Y and Z coordinates shown in Figure 5. There can be found differences between the paths made by the final effector and the desired one. These differences do not represent a significant position error for the final effector tool used in the test (gripper with 32.5 mm radius).

5. CONCLUSION

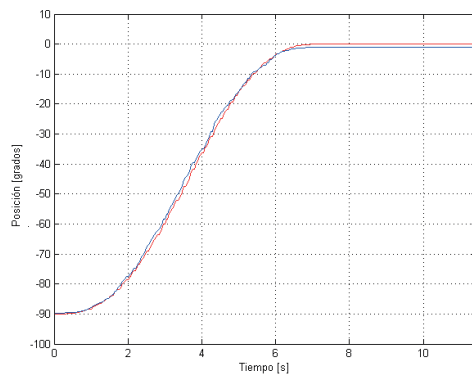
Three path algorithms are implemented, MOVEJ, MOVES and MOVEC, which provide a sequence of points in the joint space to generate the different movements (articular, linear and circular). The interpolations of trajectories in the Cartesian space (MOVES and MOVEC), it is necessary to calculate the MCI to find the articular values ??of each



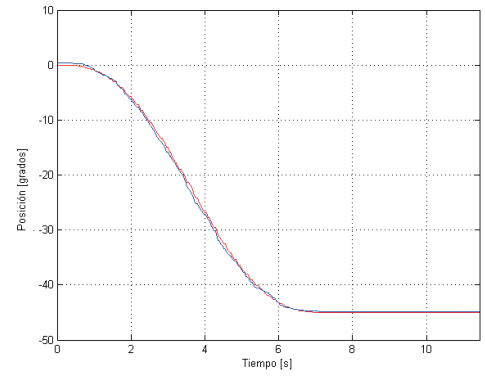
a) Joint motion of θ_1 .



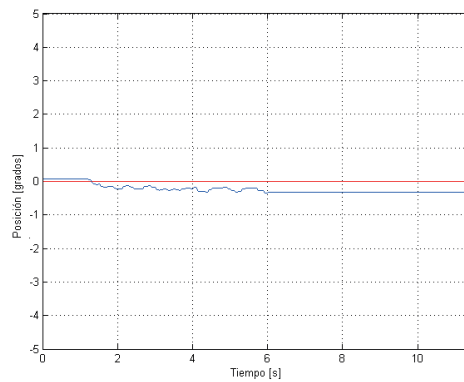
b) Joint motion of θ_2 .



c) Joint motion of θ_3 .



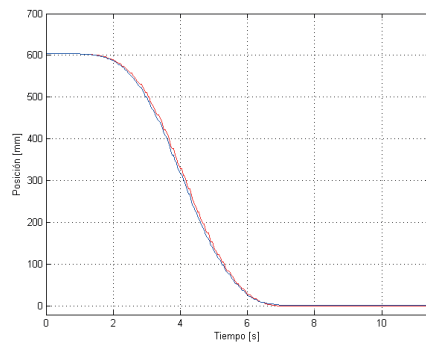
d) Joint motion of θ_4 .



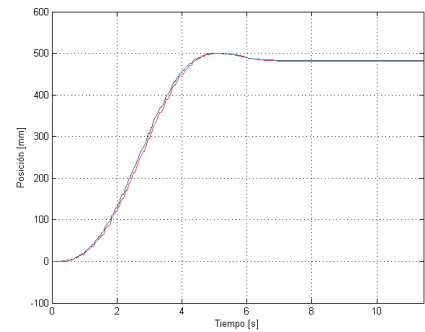
e) Joint motion of θ_5 .

Figure 4: Movimiento articular del manipulador de 5GDL.

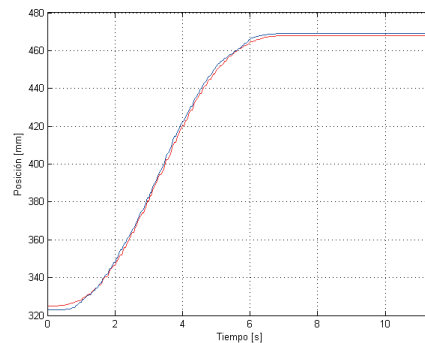
interpolated Cartesian point [Deshpande], which implies that it requires more processing in these commands, compared to MOVE].



a) Movement on the X axis.



b) Movement on the Y axis.



c) Movement on the Z axis.

Figure 5: Cartesian movement of the end effector of the 5GDL manipulator.

Combining the LabView programming platform and the CompactRio industrial controller, they provide a complete variable monitoring and control system to design, to create prototype and deploy programs with greater efficiency, speed and power in a good controller.

The use of rigorous mathematical models allows to obtain the algorithms that generate trajectories, which provide articular and Cartesian coordinates for the movement of the manipulator. In this case (Figure 5), the actual coordinates of the final effector show oscillations on the desired coordinates. These results are acceptable as they are within the tolerance of the tool used for testing (gripper with radius of 32.5 mm).

Authorization and Disclaimer

Authors authorize ESTEC to publish the papers in the conference proceedings. Neither ESTEC nor the editors are responsible either for the content or for the implications of what is expressed in the paper.

References

- [1] Alcaraz Salvago, A. (2012). "Desarrollo de algoritmos de control y movimiento de robots", mediante LabVIEW Robotics.
- [2] Barrera, T., Hast, A., & Bengtsson, E. (2004, November). "Incremental spherical linear interpolation". In *The Annual SIGRAD Conference. Special Theme-Environmental Visualization*. (No. 013, pp. 7-10). Linköping University Electronic Press.
- [3] Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (1997). "Fundamentos de robótica" (Vol. 256). McGraw-Hill. Págs. 12, 17.
- [4] Castillo, S. A., & Caberta, R. Ñ. (2003). "Caracterización de un robot manipulador articulado". *Coordinación de Mecatrónica, Tesis de Maestría, CENIDET, México, Junio del*.
- [5] Constaín, A., Torres, K., Arango, J., & Vivas, A. (2009). "Modelado, Identificación Paramétrica y Control del Robot SCORBOT-ER 5 PLUS". In *Presentado en el VIII Congreso de la Asociación Colombiana de Automática, Cartagena, Colombia*.
- [6] Chaudhary, H., & Prasad, R. (2011). "Intelligent inverse kinematic control of scorbot-er v plus robot manipulator".
- [7] Deshpande, V. A., & George, P. M. (2012). "Analytical Solution for Inverse Kinematics of SCORBOT-ER-Vplus Robot". *International Journal of Emerging Technology and Advanced Engineering*.
- [8] Eberly, D. (2002). "Quaternion algebra and calculus". *Magic Software Inc*.
- [9] Kazemi, M. S., & Dominguez, M. J. (2016, October). "Simulation and evaluation of neuro-controllers applied in a SCORBOT. In *Automatica (ICA-ACCA)*", *IEEE International Conference on* (pp. 1-9). IEEE.
- [10] Kumar, R. R., & Chand, P. (2015, February). "Inverse kinematics solution for trajectory tracking using artificial neural networks for SCORBOT ER-4u. In *Automation, Robotics and Applications (ICARA)*". *6th International Conference on* (pp. 364-369). IEEE.
- [11] Parra Andrade, G. E. (2015). "Control del robot Pioneer 3D utilizando una FPGA Rio (Doctoral dissertation, Universidad de las Fuerzas Armadas ESPE". Carrera de Ingeniería en Electrónica, Automatización y Control).

- [12] Rojas, J. H. C., Serrezuela, R. R., López, J. A. Q., Perdomo, K. L. R. (2016). LQR hybrid approach control of a robotic arm two degrees of freedom. *International Journal of Applied Engineering Research*. 11(17), pp. 9221-9228
- [13] 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.
- [14] Serrezuela, R. R., Chavarro, A. F. C., Cardoso, M. Á. T., Toquica, A. L., & Martinez, L. F. O. (2017). Kinematic Modelling of a robotic arm manipulator using MatLab. *ARNP Journal of Engineering and Applied Sciences*. 12(7), pp. 2037-2045
- [15] Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *“Robotics: modelling, planning and control”*. Springer Science & Business Media.
- [16] Zarta, J. R. & Serrezuela, R. R. (2017). Solution of systems of differential equations deformed with K-Exponential Matrix, In Taekyun Kim, *Advanced Mathematics: Theory and applications* (p.p. 189-204), India, Delhi.

Authorization and Disclaimer

Authors authorize ESTEC to publish the paper in the conference proceedings. Neither ESTEC nor the editors are responsible either for the content or for the implications of what is expressed in the paper.