

Design of Control Interface for a SCARA Manipulator with Subactuated Final Effector

Diseño de la interfaz de control para un manipulador SCARA con efector final subactuado

Julio Cesar **Cordero Muñoz**¹, Humiko Yahaira **Hernández Acosta**²
Raúl **Benítez Álvarez**³, Gerardo Julián **Pérez Mendoza**⁴

¹Universidad Politécnica del Valle de México, MÉXICO

<https://orcid.org/0000-0002-1535-3007> | julio.cordemunoz05@gmail.com

²Universidad Politécnica del Valle de México, MÉXICO

<https://orcid.org/0000-0002-2330-3026> | humikoyah@hotmail.com

³Universidad Politécnica del Valle de México, MÉXICO

<https://orcid.org/0000-0003-1221-5431> | rbentz-mx@hotmail.com

⁴Universidad Politécnica del Valle de México, MÉXICO

<https://orcid.org/0000-0002-5481-2640> | geras493@hotmail.es

Recibido 16-01-2020, aceptado 31-03-2020.

Abstract

The decoupled control of robots eases the generation of trajectories of position, speed and acceleration, as well as the combination of sequences of movement in the joints. The aim of this work is to develop an application for motion control, integrating a virtual model into a prototype SCARA manipulator of 4 DOF. The methodology consists: 1. Definition of the manipulator's workspace; 2. Preparation of the control diagram in LabVIEW; 3. Configuration and communication with Arduino hardware. 4. Development of the interface for simultaneous movements; 5. Configuration of the communication interface with SolidWorks. Developing such integration requires particular hardware characteristics that support simulation and communication environments between the different software mentioned, as well as the acquisition and processing of analog signals used in the control algorithm applied to the model, with Lagrange polynomials and direct and inverse kinematics modelling by the Denavit-Hartenberg method.

Index terms: Arduino, kinematics, virtual Interface, SCARA, dynamics.

Resumen

El control desacoplado de robots facilita la generación de trayectorias de posición, velocidad y aceleración, así como la combinación de secuencias de movimiento en las articulaciones. El objetivo del trabajo es desarrollar una aplicación para el control de movimiento, integrando un modelo virtual en un prototipo de manipulador SCARA de 4 GDL. La metodología consiste en: 1. Definición del espacio de trabajo del manipulador; 2. Elaboración del diagrama de control en LabVIEW; 3. Configuración y comunicación con el hardware Arduino. 4. Elaboración de la interfaz para movimientos simultáneos; 5. Configuración de la interfaz de comunicación con SolidWorks. Desarrollar dicha integración requiere características particulares de hardware que soporte los ambientes de simulación y comunicación entre los diferentes softwares citados, así como, la adquisición y procesamiento de señales analógicas utilizadas en el algoritmo de control aplicado al modelo, con polinomios de Lagrange y ecuaciones de cinemática directa e inversa por el método Denavit-Hartenberg.

Palabras clave: Arduino, cinemática, Interfaz virtual, SCARA, dinámica.

I. INTRODUCTION

Automatic control in the industry has become a priority due to the increasing of production requirements. Therefore, we have seen the need to develop new hardware and software for the automation of processes that allow compliance with both national and international standards, as well as making a faster and more efficient manufacturing of products [1]. This has been satisfied with the development of industrial robots, manipulating both objects and tools in certain tasks. A virtual interface can be added with which to reliably control the movement parameters of both the manipulator and the final effector in real time, and even be able to visualize a 3D model of its real operation before performing the tasks.

Industrial manipulators are closed architecture, so only specialists can make modifications. However, tools such as LabVIEW (Laboratory Virtual Instrument Engineering Workbench) allow the development of control algorithms in restricted virtual environments, which simplify the simulation of the physical model. Thanks to its specialized National Instruments programming environment, this software allows the creation of simple control algorithms through the visual programming of functional control blocks called VI (Virtual Instruments) that are dragged and released [2]. It makes possible to convert test techniques and validation of complex algorithms into simpler and more flexible programming.

From the creation of virtual instruments, we can achieve control structures of both the SCARA manipulator and its final effector, as well as its simulation of the CAD model in real time; all this, thanks to the creation of a virtual interface in real time. The hypothesis was based on the possibility of establishing real-time parameter control in the robot.

II. METHODOLOGY

The first stage in the control interface design methodology is the mechanical design of the SCARA manipulator and its subsequent study of movement with the SolidWorks Motion complement. The next step is to know the parameters of the potentiometers in each joint according to the angle of position or the distance to subsequently enter these parameters in the control program. Next, the control block diagram in LabVIEW is designed for each one of manipulator joints; for this, the direct and inverse kinematics modelling must be calculated, as well as the dynamics of the manipulator. Subsequently, the electrical circuit must be made for the connection of the Arduino board with the manipulator and the computer that will control the program instructions. Finally, tests of the movement control of the manipulator are performed.

III. MOTION STUDY IN SOLIDWORKS

The SolidWorks model designed has two rotational joints (shoulder and elbow), a prismatic joint (sliding or Z axis) and a gripper with an opening and closing system consisting of two rotating gears, which are considered as a degree of rotational type freedom, because the sensor is coupled to generate a single signal (see Fig. 1).

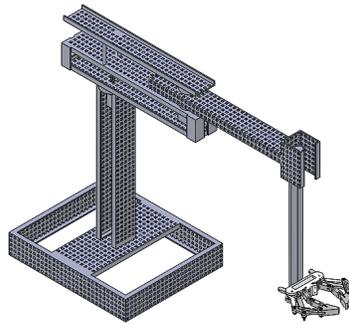


Fig. 1. SCARA manipulator model designed in SolidWorks.

To achieve the motion study, is needed the SolidWorks Motion add-on, which can be used to connect rotary and linear motors to the model, among others. For the manipulator at hand, the movements for the study are those shown in Table 1.

TABLE 1
PARAMETERS FOR STUDY MOVEMENT IN SOLIDWORKS

Parameters for study movement in SolidWorks				
Joint	Type	Range	Time	Animation type
Shoulder	Rotational	0-180°	1 s	Rotatory Motor
Rotatory Motor	Rotational	0-90°	1 s	Rotatory Motor
Rotatory Motor	Prismatic	0-8 in	1 s	Linear Motor
Gripper	Rotational	0-40°	1 s	Rotatory Motor

Link dimensions and robot slide will be necessary to perform the kinematics modelling of the robot, so these dimensions are shown in Table 2:

TABLE 2
SCARA MANIPULATOR LINK DIMENSIONS

Link	Dimension (in)
l_1	15.5"
l_2	10"
l_3	12"
l_4	3"
d_1 max.	8"

A. Direct and inverse kinematics modelling of the SCARA manipulator

Kinematics of a robot is the branch that "studies its movement with respect to a reference system" [3]. For a robot kinematics modelling there are two problems that must be solved: direct kinematics and inverse kinematics. The first involves determining the position and orientation of the final effector of the robot, with respect to a coordinate system that is taken as a reference; the second one resolves the configuration that the robot must have for a known position and orientation of the final effector.

4

One of the most used methods to establish the direct and inverse kinematics modelling of a robot is through Denavit-Hartenberg algorithm, which is a systemic method that describes and represents the spatial geometry of the elements of a kinematic chain, with respect to a fixed reference system. Such method uses a homogeneous transformation matrix to describe the spatial relationship between two immediate rigid elements, whereby the direct kinematic problem can be reduced to a 4x4 homogeneous transformation matrix that relates the spatial location of the final effector of the robot with respect to the coordinate system of its base. The reference systems, distances, lengths and turning angles of the joints of the SCARA manipulator are shown in diagram below (see Fig. 2).

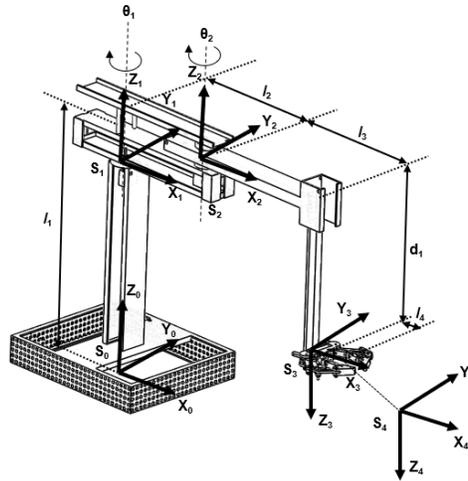


Fig. 2. Modelo matemático.

Taking as reference the steps of the Denavit-Hartenberg method, the parameters for the homogeneous matrix are shown in Table 3. Together, based on the matrix, the equations are obtained to find the position of the robot in the space of work according to the angles formed by the links, as shown in Table 4.

TABLE 3
DENAVIT-HARTENBERG PARAMETERS FOR SCARA MANIPULATOR

Joint	θ	D	a	α
1	0	l_1	0	0
2	θ_1	0	l_2	0
3	θ_2	$-d_1$	l_3	180
4	0	0	l_4	0

TABLE 4
DIRECT KINEMATIC EQUATIONS FOR SCARA MANIPULATOR

Direct kinematic equations	
X position	$p_x = 10\cos(\theta_1) + 15[\cos(\theta_1 + \theta_2)]$
Y position	$p_y = 10\sin(\theta_1) + 15[\sin(\theta_1 + \theta_2)]$
Z position	$p_z = 15.5 - d_1$

Likewise, the inverse kinematics can also be obtained so that the angles are in function of the position. The equations of the above are shown in Table 5:

TABLE 5
INVERSE KINEMATIC EQUATIONS FOR THE SCARA MANIPULATOR. r_x , r_y AND r_z ARE THE POSITIONING POINTS (x, y, z) OF THE FINAL EFFECTOR

Inverse kinematic equations	
Joint 1	$\theta_1 = \tan^{-1}\left(\frac{r_y}{r_x}\right) - \tan^{-1}[15\sin(\theta_2)/(10 + 15 \cos(\theta_2))]$
Joint 2	$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1-M^2}}{M}$, $M = (r_x^2 + r_y^2 - 325)/(300)$
Joint 3	$d_1 = l_1 - r_z$

With the above equations, both the inverse and direct kinematic control diagram can be made in LabVIEW programming, to know the value of the angles from a given point x, y, z and to know the current position of the SCARA manipulator (see Fig. 3).

B. SCARA manipulator dynamics

Dynamics is one that deals with the relationship between the forces that act on a body and the movement that originates from them [3]. Dynamic model relates mathematically: the location of the robot defined by its own articular variables and their derivatives (speed and acceleration); the forces and torques applied at the end of the robot or in the joints; and the dimensional parameters of the robot (length, masses and inertia of the elements). The method to be used to make the dynamic model of the manipulator is that of Newton-Euler and Lagrange.

From the equations of direct kinematics, it is derived with respect to time to obtain the velocity matrix, which is used, in turn, to obtain the equations of kinetic energy, potential energy and, finally, the equation of force-pairs, according to the formula (1).

$$\mathcal{L} = \mathcal{K} - \mathcal{U} = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} \quad (1)$$

The kinetic energy equations of our model for each joint are as follows:

$$k_1 = \frac{1}{2}(m_1 l_2^2 + I_1) \dot{q}_1^2$$

$$k_2 = \frac{1}{2} m_2 (l_2^2 + 2l_2 l_3 c q_2 + l_3^2) \dot{q}_1^2 + \frac{1}{2} I_2] + \frac{1}{2} (m_2 l_3^2 + I_2) \dot{q}_2^2 + m_2 (l_3^2 + l_2 l_3 c q_2) \dot{q}_1 \dot{q}_2$$

$$k_3 = \frac{1}{2} m_3 (l_2^2 + 2l_2 l_3 c q_2 + l_3^2) \dot{q}_1^2 + I_3] + \left(\frac{1}{2} m_3 l_3^2\right) \dot{q}_2^2 + [m_3 (l_3^2 + l_2 l_3 c q_2)] \dot{q}_1 \dot{q}_2 + \left(\frac{1}{2} m_3\right) \dot{q}_3^2$$

$$\mathcal{K} = \frac{1}{2} [(m_1 l_2^2 + I_1) + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2)] \dot{q}_1^2 + (m_2 + m_3) [(l_3^2 + l_2 l_3 c q_2)] \dot{q}_1 \dot{q}_2 + \frac{1}{2} [(m_2 + m_3) l_3^2 + I_2] \dot{q}_2^2 + \left(\frac{1}{2} m_3\right) \dot{q}_3^2$$

The potential energy equations are as shown below:

$$U_1 = m_1 g l_1$$

$$U_2 = m_2 g l_1$$

$$U_3 = m_3 g (l_1 - q_3)$$

$$\mathcal{U} = (m_1 + m_2 + m_3) g l_1 - m_3 g q_3$$

6

Remembering that the Lagrangian results from the difference between kinetic energy and potential energy, we have:

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} [(m_1 l_2^2 + I_1) + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2)] \dot{q}_1^2 + (m_2 + m_3) [(l_3^2 + l_2 l_3 c q_2)] \dot{q}_1 \dot{q}_2 \\ & + \frac{1}{2} [(m_2 + m_3) l_3^2 + I_2] \dot{q}_2^2 + \left(\frac{1}{2} m_3\right) \dot{q}_3^2 - (m_1 + m_2 + m_3) g l_1 + m_3 g q_3 \end{aligned}$$

Deriving partially the Lagrangian with respect to \dot{q}_1 , \dot{q}_2 y \dot{q}_3 , it turns out:

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_1} = [m_1 l_2^2 + I_1 + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2)] \dot{q}_1 + (m_2 + m_3)(l_3^2 + l_2 l_3 c q_2) \dot{q}_2$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_2} = (m_2 + m_3) [(l_3^2 + l_2 l_3 c q_2)] \dot{q}_1 + [(m_2 + m_3) l_3^2 + I_2] \dot{q}_2$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}_3} = m_3 \dot{q}_3$$

Deriving from time the previous equations, the accelerations for each joint are obtained:

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right] = & [m_1 l_2^2 + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2) + I_1] \ddot{q}_1 + (m_2 + m_3)(l_3^2 + l_2 l_3 s q_2) \ddot{q}_2 + \\ & (-2m_2 l_2 l_3 s q_2 - 2m_3 l_2 l_3 s q_2) \dot{q}_1 + (-m_2 l_2 l_3 s q_2 - m_3 l_2 l_3 s q_2) \dot{q}_2 \end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right] = & (m_2 + m_3)(l_3^2 + l_2 l_3 c q_2) \ddot{q}_1 + [(m_2 + m_3)(l_3^2) + I_2] \ddot{q}_2 + (-m_2 l_2 l_3 s q_2 - \\ & m_3 l_2 l_3 s q_2) \dot{q}_1 \dot{q}_2 \end{aligned}$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_3} \right] = m_3 \ddot{q}_3$$

Deriving the previous equations with respect to q_1 , q_2 y q_3 , you get:

$$\frac{\partial \mathcal{L}}{\partial q_1} = 0$$

$$\frac{\partial \mathcal{L}}{\partial q_2} = (-m_2 l_2 l_3 s q_2 - m_3 l_2 l_3 c q_2) \dot{q}_1^2 + [(-m_2 l_2 l_3 s q_2 - m_3 l_2 l_3 s q_2)] \dot{q}_1 \dot{q}_2$$

$$\frac{\partial \mathcal{L}}{\partial q_3} = m_3 \dot{q}_3 + m_3 g$$

The Lagrangian equation states that: $\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right] - \frac{\partial \mathcal{L}}{\partial q_1} = \tau$, so that the following general equation of force-pairs can now be written as follows:

$$\begin{aligned} \tau = & [m_1 l_2^2 + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2) + I_1] \ddot{q}_1 + (m_2 + m_3)(l_3^2 + l_2 l_3 s q_2) \ddot{q}_2 \\ & + (-2m_2 l_2 l_3 s q_2 - 2m_3 l_2 l_3 s q_2) \dot{q}_1 + (-m_2 l_2 l_3 s q_2 - m_3 l_2 l_3 s q_2) \dot{q}_2 \\ & + (m_2 + m_3)(l_3^2 + l_2 l_3 c q_2) \ddot{q}_1 + [(m_2 + m_3)(l_3^2) + I_2] \ddot{q}_2 \\ & + (-m_2 l_2 l_3 s q_2 - m_3 l_2 l_3 s q_2) \dot{q}_1 \dot{q}_2 + m_3 \dot{q}_3 + (m_2 l_2 l_3 s q_2 + m_3 l_2 l_3 c q_2) \dot{q}_1^2 \\ & + [(+m_2 l_2 l_3 s q_2 + m_3 l_2 l_3 s q_2)] \dot{q}_1 \dot{q}_2 - m_3 \dot{q}_3 - m_3 g \end{aligned}$$

The previous equation can become a matrix equation of force-pairs applied to each link.

$$\begin{aligned} \tau = & \begin{bmatrix} [m_1 l_2^2 + (m_2 + m_3)(l_2^2 + 2l_2 l_3 c q_2 + l_3^2) + I_1] & [(m_2 + m_3)(l_3^2 + l_2 l_3 s q_2)] & 0 \\ [(m_2 + m_3)(l_3^2 + l_2 l_3 c q_2)] & [(m_2 + m_3)(l_3^2) + I_2] & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} \\ & + \begin{bmatrix} -2[(m_2 + m_3)(l_2 l_3 s q_2)] & -(m_2 + m_3) l_2 l_3 s q_2 & 0 \\ -\dot{q}_2 [(m_2 + m_3)(l_2 l_3 s q_2)] & \dot{q}_1 [(m_2 + m_3)(l_2 l_3 s q_2)] & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + g \begin{bmatrix} 0 \\ 0 \\ m_3 \end{bmatrix} \end{aligned}$$

The matrix relates the location of the robot based on the interrelation of all the dimensional parameters, as well as its joint variables.

C. Motor control diagram with LabVIEW-Arduino

The VEX 393 motors used are controlled by the value of the VEX potentiometer range. The range of potentiometers ranges from 0 to 4096, giving a corresponding output voltage of 0 to 5 volts. The functional values were obtained experimentally, which are detailed in Table 6. In addition, the Lagrange polynomial method [4] was necessary to obtain the equivalence between the angle of rotation of the motors 1 and 2 the value of the range of the potentiometers used, since their behavior corresponded to a quadratic equation. The other motors showed a linear behavior in their angle-range potentiometer ratio. The equations below describe the value of the range of potentiometers 1 to 4 depending on the required angle, which is obtained with the inverse kinematic modelling.

$$y(x) = 0.016296296x^2 + 13.7333333x + 500 \quad (2) \quad \text{Joint 1}$$

$$y(x) = 0.0237037x^2 + 11.5111111x + 664 \quad (3) \quad \text{Joint 2}$$

$$y = \frac{3832}{7}x + 232 = 547.428571x + 232 \quad (4) \quad \text{Joint 3}$$

$$y = \frac{-950}{40}x + 1250 = -23.75x + 1250 \quad (5) \quad \text{Gripper}$$

TABLE 6
FUNCTIONAL VALUES OF THE POTENTIOMETERS FOR EACH JOINT

Parameter	Motor of joint 1 (Rot)		
	Origin	Medium	Maximum
Angle of rotation (x)	0°	90°	180°
Potentiometer value (y)	500	1868	3500
Parameter	Motor of joint 2 (Rot)		
	Origin	Medium	Maximum
Angle of rotation (x)	0°	90°	180°
Potentiometer value (y)	664	1892	3504
Parameter	Motor of joint 3 (Pris)		
	Minimum	Maximum	
Distance (in) (x)	0 in	7 in	
Potentiometer value (y)	232	4064	
Parameter	Gripper Motors 4, 5 (Rot)		
	Closing	Opening	
Opening angle (x)	0°	40°	
Potentiometer value (y)	1250	300	

With the above, we can introduce these equations in a LabVIEW block diagram and use the blocks corresponding to the MakerHub plug-in for LabVIEW-Arduino communication, to obtain the reading / writing of signals [5]. The complete control diagram uses the robot's end point (x, y, z) in inches, in order to generate the angles of movement of each joint (see Fig. 3).

D. Communication settings between SolidWorks and LabVIEW

For communication between LabVIEW and SolidWorks it is necessary to install the RealTime and SoftMotion plug-ins in LabVIEW, and Motion in SolidWorks. Subsequent, a series of steps must be followed to link the motion study engines generated in SolidWorks with the LabVIEW project manager. Afterwards, the "Straight Line Move" blocks of the SoftMotion plug-in are used to control the free movement of each motor in SolidWorks, through the LabVIEW interface, which allows real-time movement of both the physical manipulator and the virtual prototype the diagram capable of controlling and recording the movements of the SCARA manipulator is shown in Fig. 3.

E. Creation of the interface for movement control in LabVIEW

The LabVIEW front panel consists of writing the desired parameters (manipulator position, gripper opening-closing, PWM for each of the motors, input voltage and number of positions to be recorded) and reading the

parameters of each joint (positions, ranges and current degrees, as well as corresponding degrees to each joint depending on the desired position). In addition, it has buttons to activate / deactivate motor movement, record and stop the program (see Fig. 4).

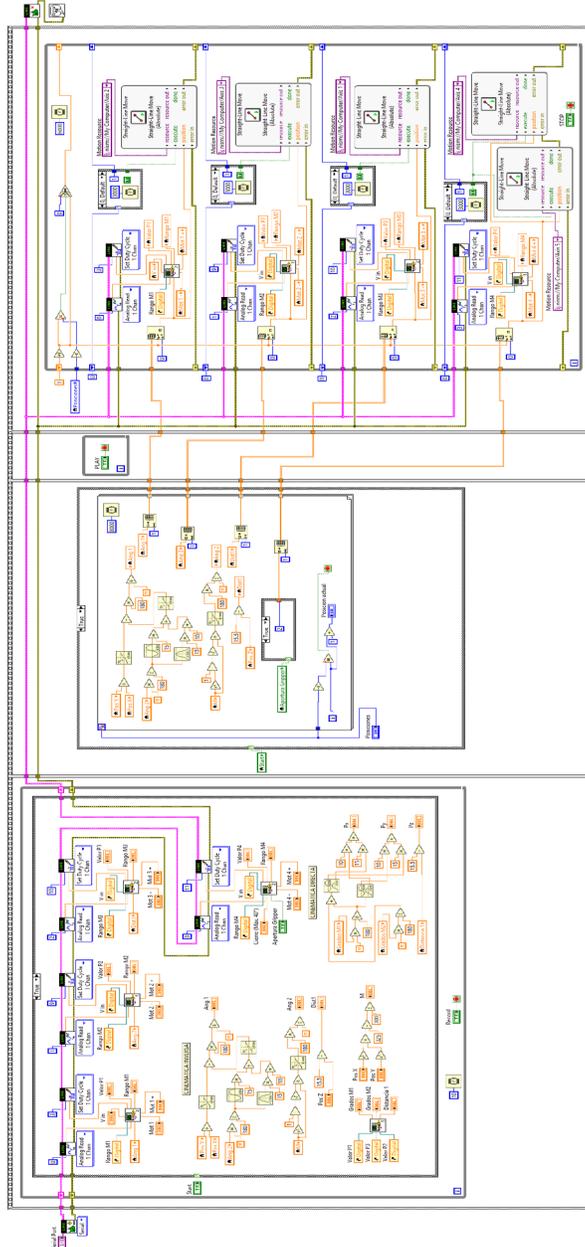


Fig. 3. Block diagram in LabVIEW for control and recording of SCARA manipulator positions.



Fig. 4. Front panel to control and display the SCARA manipulator parameters.

IV. SUMMARY OF RESULTS

A routine for each joint was tested separately, introducing the positioning points (x, y, z) and the opening and closing value of the final effector (gripper). It was theoretically proven that the angular results of inverse kinematics were correct. In addition, the graph of the coordinates (x, y) vs time and the value of the angles (θ_1, θ_2) vs time was made to visualize the change of the latter as values are given in x and y . The Waveform chart block was used for the graphs (see Fig. 5).

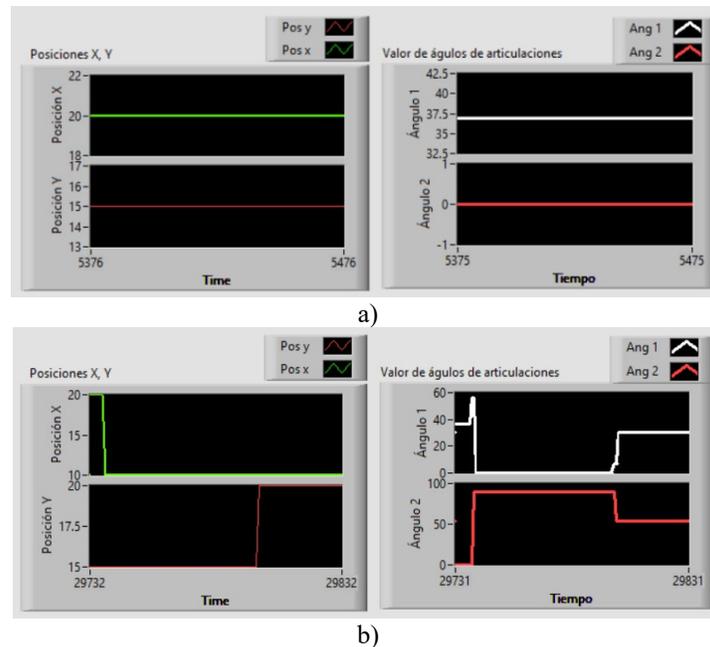


Fig. 5. a) Position (20,15) results in $\theta_1 = 36.86^\circ$, $\theta_2 = 0^\circ$ with inverse kinematic equations;
b) Position (10,20) results in a change to $\theta_1 = 30.4^\circ$, $\theta_2 = 54.3^\circ$ with inverse kinematic equations.

Physically, reading/writing of signals from the potentiometers and to the motors was verified, being able to obtain movement in each articulation of the manipulator. A simple electrical circuit was made with Arduino capable of sending and receiving signals (see Fig. 6).

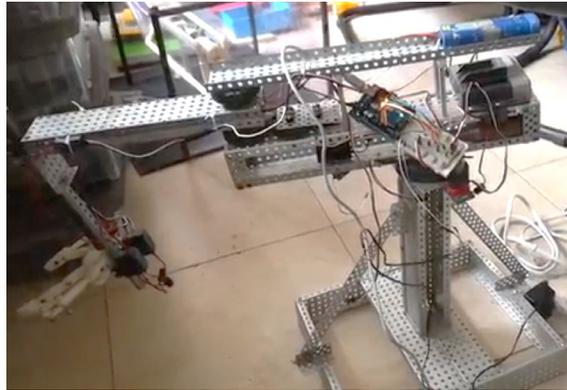


Fig. 6. Signal test in manipulator using a simple circuit made with Arduino.

The interface was able to control the movement of the manipulator and the CAD model in real time, so that each physical movement desired was assigned a virtual physical movement (see Fig. 7).



Fig. 7. Real-time virtual interface of the SCARA manipulator along with its CAD model.

V. CONCLUSIONS

A SCARA manipulator can be controlled with a virtual interface in real time, designing a program based on control blocks in LabVIEW that eases programming and eliminates physical interfaces that can be damaged in the medium term by use. A subactuated and adaptive final effector was adapted, which also was controlled with such interface, demonstrating flexibility of working with virtual interfaces. In addition, the above allows to

simulate in real time and off-line the movement of all the articulations and links of the manipulator, as could be observed in the start-up of each of the SCARA manipulator's joints.

The interface made in LabVIEW is also especially useful when it is needed to simulate the movement of a model with SolidWorks that has not yet been manufactured. Finally, it was learned that this type of interfaces saves material costs since no physical controls are needed because it works with the commands of the front panel and only, in our case, a simple circuit with which to read and manipulate the movements was needed, the equipment to run the simulation and an external power supply for both motors and potentiometers. This project can be useful for handlers who are programmed using a conventional method and need to optimize their equipment with a low cost.

REFERENCES

- [1] N. Kodaira, "Expected innovation in industrial robots," *Advanced Robotics*, p. 8, 2016.
- [2] B. B. Rhoades, J. P. Sabo, J. M. Conrad, "Enabling a National Instruments DaNI 2.0 robotic development platform for the Robot Operating System," SoutheastCon 2017, Concord, NC, USA, 2017, pp. 1-5, doi: <https://doi.org/10.1109/SECON.2017.7925293>
- [3] A. Barrientos, L. Peñin, C. Balaguer, R. Aracil, *Fundamentos de Robótica*, España: McGraw-Hill, 2007.
- [4] G. Montiel, R. Cantoral, "Una presentación visual del polinomio de lagrange," *Números. Revista de Didáctica de las Matemáticas*, n° 55, pp. 3-22, 2003.
- [5] K. García, Youtube, 4 Oct. 2015 [online], available: <https://www.youtube.com/watch?v=cLdIPzvQAQw>
- [6] M. Hägele, K. Nilsson, J. Pires, R. Bischoff, *Industrial robotics, Springer handbook of robotics*, Springer, Cham, 2016.