Advances in industrial robots programming applying gestural guidance techniques

A F Padilla¹, C A Peña¹, and G G Moreno-Contreras¹

¹ Grupo de Automatización y Control, Universidad de Pamplona, Pamplona, Colombia

E-mail: andres.padilla@unipamplona.edu.co, cesarapc@unipamplona.edu.co

Abstract. In this article, artificial vision is presented as an application for the generation of trajectories through the capture of an operator’s movements using a structured light camera (Kinect) and applying a recognition algorithm to the body's joints, a three-dimensional representation is made of the skeleton, taking as points of interest the articulation of the left hand to generate the gestural commands and the right hand to indicate the path that the robot must execute. Simulation results are obtained through a development environment where the kinematic model of the robot is incorporated, this indicates that it is possible to generate a trajectory through the human body using artificial vision.

1. Introduction

Industrial robots have been around since the late 1950s and have evolved to suit the needs of modern industry. The first generation of robots consisted of complex mechanical systems that had to be manipulated by an operator and had a simple control system, the second generation brought learning robots, which repeat a sequence of movements previously executed by means of a code, the third generation introduces sensory devices that help the robot to interact with the work environment, and finally mobile or intelligent robots belong to the fourth generation, bringing us closer to artificial intelligence [1]. An industrial robot can adapt to various tasks in a company by reprogramming itself.

The programming of a robot is defined as the process by which it is indicated the sequence of actions that it must carry out during the performance of a certain task [2]. Therefore, programming is the tool that the user must program the robot to perform a specific task. The physical characteristics of the robot and the programming possibilities are directly related [2]. The two types of programming mostly used today are defined below. The first is guided or learning programming, which consists of manipulating the desired trajectory or points in which the robot is desired to move at the same time as adapted configurations are recorded for later automatic reproduction [3]. The second is textual programming, which uses a specific programming language to edit orders in the robot’s tasks, which are subsequently executed [4].

In the development of this project, the guided programming method is used, including new technologies such as the structured light camera with the application of artificial vision techniques, to program the trajectory without the need to move the robot out of the operating line. This project is based on a user interface platform in which the trajectory is defined from the movements captured by Kinect under a reference system that simulates the trajectory of the robot, which must later be adjusted and saved for programming. For this application the Raplim robot was used, which is anthropomorphic and has 4 degrees of freedom, this robot was taken as a test model to implement the programming method.
based on artificial vision in order to answer the following question: Is it possible to generate a guided trajectory from references located on the operator's own body through the use of artificial vision?

2. Kinematic of the manipulator robot

Each robot has particular characteristics, so it is necessary to carry out the mathematical calculations to develop the robot’s trajectory generation algorithm which represents the movements of the joints and each of the links respecting the position and orientation of the end effector [5]. To find the relationship between the joint coordinates, the position and orientation of the end effector, the direct and inverse kinematics of the Raplim robot of the Universidad de Pamplona, Colombia, are found [6].

2.1. Direct kinematics

Knowing the morphology and structure of the Raplim robot, the lengths of the links and the types of joints are found, this makes it possible to determine the position and orientation of the robot’s end effector respecting a fixed reference system (robot base). To calculate the direct kinematics, a series of rules known as the Denavit Hartenberg algorithm are used, which takes four basic transformations that allow to relate the reference system of link i with the reference system of link i-1, the product of these four transformations results in the matrix described in Equation (1) [7].

\[
i^{-1}A_i = \begin{bmatrix}
    \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & q_i \cos \theta_i \\
    \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & \alpha_i \sin \theta_i \\
    0 & \sin \alpha_i & \cos \alpha_i & d_i \\
    0 & 0 & 0 & 1 
\end{bmatrix}.
\]

(1)

The Figure 1 show the model of the Raplim robot in consequence the Figure 1(a) shows the simplified model of the Raplim robot and the orientation of the coordinate axes corresponding to each link [8], and Figure 1(b) show the diagram to determine joint coordinate \( q_1 \). According to Figure 1(a), the Denavit Hartenberg parameters for the Raplim robot are obtained as shown in Table 1 [9].

![Figure 1. Model of the Raplim robot, (a) simplified model of the Raplim robot, (b) diagram to determine joint coordinate \( q_1 \).](image)

| Joint | \( \theta \) | \( d \) | \( A \) | \( \alpha \) |
|-------|-------|-------|-------|-------|
| 1     | \( q_1 + \pi/2 \) | \( L_1 \) | 0     | \( \pi/2 \) |
| 2     | \( q_2 \)     | 0     | \( L_2 \) | 0     |
| 3     | \( q_3 \)     | 0     | \( L_3 \) | 0     |
| 4     | \( q_4 \)     | 0     | \( L_4 \) | 0     |

Table 1. Denavit Hartenberg parameters of the robot.
The values in Table 1 are replaced within the Denavit Hartenberg matrix, there must be one matrix for each joint. From the product of the four matrices \((0A_4 = 0A_1 \; 1A_2 \; 2A_3 \; 3A_4)\), the matrix that indicates the orientation and position of the end effector respecting the reference system of the robot’s base [10].

2.2. **Inverse kinematics**

Inverse kinematics, as its name indicates, seeks to find the shape that the robot should adopt from the position and orientation of the end effector, it is the opposite of direct kinematics. To obtain the values of the robot’s joint coordinates from the position and orientation of the end effector respecting the reference system of its base, it is considered to solve by the geometric method the first 3 degrees of freedom that the Raplim robot has [10], the last degree is solved by the vector method. This strategy is used because the geometric method works only when the robot has few degrees of freedom. Starting from the above, we begin to solve Raplim’s first 3 degrees of freedom inverse kinematics. In Figure 1(b) the top view of the simplified model for an end effector position can be appreciated \(P = [P_x; P_y; P_z]\), with it, the first joint coordinate can be determined \(q_1\) as \(\tan^{-1}(P_y/P_x)\) [7].

In Figure 2(a) the angle “\(a\)” that determines the degrees of rotation of the end effector respecting the reference system of the robot’s base is detailed. This value is defined by the robot operator and together with the Cartesian coordinates of the end effector, these are the necessary data to calculate the inverse kinematics from which the \(X_4\) components are obtained, Equation (2).

\[
\begin{bmatrix}
X_{4x} \\
X_{4y} \\
X_{4z}
\end{bmatrix} =
\begin{bmatrix}
\cos(a) \sin(q_1) \\
\cos(a) \cos(q_1) \\
\sin(a)
\end{bmatrix}.
\]  

With these values, the point of the wrist is determined and \(q_1\) is calculated. Figure 1(b) shows the top view which is used to determine the distance \(m\) as the square of the components \(P_{WX}\) and \(P_{WX}^2\). Finally, the equations to determine the joint coordinates \(q_2\) and \(q_3\) are supported by Figure 2(b) which illustrates the robot’s side view. To calculate \(q_3\) begin with finding \(J\) that is the distance between 1 and \(R_W\). Taking into consideration that the \(\theta\) angle from Figure 2(b) is \(180 - q_3\) the cosine law is applied to solve \(\cos q_3\), Equation (3).
Using the trigonometric identity $\tan^{-1}(a) = \arctan\left(\frac{a}{1}\right)$, $q_3$ is solved as shown in Equation (4).

$$q_3 = \arctan\left(\frac{\sin q_3}{\cos q_3}\right) = \arctan\left(\frac{1 - \cos^2 q_3}{\cos q_3}\right).$$

With the value of $q_3$ proceeding to calculate the joint coordinate $q_2$ is possible, considering the relationship between Equation (5).

$$q_2 = \beta - \alpha = \arctan\left(\frac{P_{mx} - L_1}{m}\right) - \arctan\left(\frac{L_3 \sin q_3}{L_2 + (L_3 \cos q_3)}\right).$$

3. Artificial vision

New technologies such as computer vision provide innovation in robot programming. The use of this technology fits into the type of guided programming, described previously, with it, movements are captured through a structured light camera. With the help of some algorithms it is possible to reproduce a trajectory in the robot.

3.1. Motion capture with the Kinect

With the help of a structured light camera (Kinect), it is possible to identify the parts of the operator's body, which are used to elaborate commands and movements that can then be reproduced with the help of programming algorithms in a robot [11]. What the Kinect captures is a series of specific points that correspond to each part of the body, and with it, it is possible to assign a command to action when the movement of a specific point of the body is registered [12].

As understood in the control by guidance, it is necessary for an operator to show the robot the path that it must travel by manipulating it directly as in passive guidance programming. According to the above, it is proposed to standardize the methodology with the help of artificial vision to capture the operator's movements and program them without having direct contact with the robot. To capture the movements, the Kinect sensor is used, which is a structured light camera that allows to acquire the three-dimensional representation of the operator's body [13]. Internally, the Kinect has an algorithm that allows it to identify a body in space and, in addition to this, capture, obtain or show certain specific points in it, as shown in Figure 3(a) [5].

The points that the Kinect sensor recognizes from the body are specified so that through the algorithm developed, a three-dimensional representation of these points can be made in the development environment, shaping the skeleton of the operator, as shown in Figure 3(a) and Figure 3(b) show the command finish record [14]. To represent the elements of the virtual environment, a transformation of all the coordinated systems is made with respect to the reference system of the robotic cell ($S_0$).

Figure 3(c) illustrates the representation of the base change of the Kinect reference system with respect to the reference system of the robotic cell. Of the points identified by the Kinect of the operator's body shown in Figure 3(a), point 7 (right wrist) is selected to be taken as a reference in the generation of the trajectory. It is necessary to make a change of base of this point with respect to the robot's reference system to obtain the cartesian positions with which the inverse kinematics of the robot can be calculated. The Equation (6) is used to apply the transformation of the point of wrist to the coordinate system of the robot. Where $^0A_R$ represents the location of the robot with respect to system 0 and $^0A_K$ represents the location of the camera system.

$$P_R = ^0A_R^{-1}^0A_K P_K.$$
Figure 4 shows the virtual environment that is programmed in the mathematical software, where the location of the coordinate systems in the room, Kinect, and Raplim can be appreciated in it with their respective points [13].

The data collection of the joint movements is done by a single person through Kinect, so it is necessary to program commands with certain poses of the body to execute actions that start, pause and end the data collection process of the program, all without having direct contact with the robot or the computer in which the trajectory is simulated and the program is executed. Figure 3(a) shows the start pose and Figure 3(b) shows the finish pose.

The start command has 2 conditions, the first is that the left wrist must move a minimum distance from the body that initiates the second condition, which is to maintain this minimum position for 5 seconds to start recording the trajectory indicated by the right wrist, but without interrupting the first condition, that is, the position of the left wrist should not change during the trajectory recording, because it pauses the recording and starts it again in the position where the right wrist is.

Once the trajectory has been executed through the movement of the right wrist, the registration must be completed, this is done by raising the left wrist above the head, that is, the left wrist must move towards the top of the head to end the registration process. For finishing the program using the respective command, a matrix is generated that stores the positions of the right wrist (trajectories) and the execution time. To define the points outside and inside the Raplim’s workspace, we apply inverse kinematics and evaluate the constraints to each position stored in the matrix.
4. Results
By capturing the movements of the operator through Kinect, the trajectory is generated in the development environment, where the blue points are differentiated from the red ones as is the position that is within reach or not of the robot respectively. From the points identified by the vision sensor (Kinect) in the space of the task, the analytical trajectories are defined and then sampled with a resolution of 5 mm as shown in Figure 5. It should be noted that all the points of the trajectory must meet the conditions established in the inverse kinematics the robot to perform the movement, if the path is outside the workspace of the robot, the system identifies and makes the appropriate corrections with prior authorization of the user either to remove the points outside the workspace of the robot or generate a new trajectory. Figure 5 represents the simulation that shows that the robot’s movements are smooth and does not collide with itself, guaranteeing that it executes the trajectory without problems in a real environment.

![Figure 5](image_url)

**Figure 5.** Simulation in virtual environment of the execution of the trajectory.

Once obtained the continuous trajectory, the kinematic control algorithm or cubic interpolator is applied to the articular coordinates of each link. Figure 6 illustrates the angular position and speed of each joint. The results of the kinematic control algorithm are used to perform a new simulation of the path, so that it can be compared with respect to the original path and determine whether the algorithm requires adjustments so that the robot can execute the movement.

![Figure 6](image_url)

**Figure 6.** Articular coordinates for each link.

5. Conclusion
In contrast to the common types of programming (textual and guided), this system allows the user to generate a trajectory without the need for mathematical calculations, since the operator can indicate it with a part of his own body, reducing the complexity in the use of industrial robots. In addition, some components of the robots could be eliminated, such as force sensors, which are essential for guided programming, reducing manufacturing costs and therefore the cost of acquiring robotic equipment.
The artificial vision algorithm allows identifying the important points located on the operator's body, which make it possible to generate a set of points that represent a movement, to which a kinematic control method is applied so that a trajectory executable by the robot is generated.

The structured light camera captures movements with precision, including those slight and involuntary movements such as vibrations, generating fluctuations in the trajectory when processing the capture. When capturing the data, a large number of positions that the robot must execute on its path are generated, this implies a considerable time when executing movements, therefore, in future instances, the use of an optimization algorithm should be considered, since it will allow to simplify or reduce the number of data predicting the route that the operator is indicating.

References
[1] Sánchez F, Jiménez P, Millán F, Salvador J, Manllau V, Palou J, Villavicencio H 2007 Historia de la robótica: de arquitas de tarento al robot da vinci (parte II) Actas Urológicas Españolas 31(3) 185-196
[2] Barrientos A, Peñín L, Balaguer C, Aracil R 2007 Fundamentos de Robótica, volumen 2 (Madrid: McGraw-Hill)
[3] García M, Cárdenas A, Rendón J, Maya M 2008 Una plataforma de control basado en visión para la rehabilitación de robots manipuladores de tipo industrial Computación y Sistemas 12(4) 409-420
[4] Cantillo A, Gualdron O, Ortiz J 2018 Procesamiento de señales emg en un sistema embeido para el control neuronal de un brazo robótico Revista Colombiana de Tecnologías de Avanzada 2(32) 139-147
[5] Rodríguez F, Botero C, Quintero H 2007 Aplicación de la teoría de robots manipuladores a la biomecánica del brazo humano Revista Iberoamericana de Automática e Informática Industrial RIAI 4(4) 51-57
[6] Torres J, Archila F, Tronco L, Becker M, Viera A, Tiberti A 2013 Estudio cinemático de una plataforma robótica para agricultura Revista Colombiana de Tecnologías de Avanzada 2(22) 131-137
[7] Ramírez J, Rubiano A 2012 Modelamiento matemático de la cinemática directe e inversa de un robot manipulador de tres grados de libertad Ingenieria Solidaria 8(15) 46-52
[8] Mejía D, Peña C, Riaño C 2016 Sistema solar para la operación de un robot agrícola Revista Colombiana de Tecnologías de Avanzada 1(27) 33-39
[9] Duque O, Ferreira J, Puentes A 2020 Diseño e implementación de un sistema de control cinemático con open-hardware del manipulador didáctico pegasus de marca amatrol ubicado en tecnoacademia Cúcuta Revista Colombiana de Tecnologías de Avanzada 1(35) 146-152
[10] Zaplana I, Arnau J, Basanez L 2018 Análisis cinemático de robots manipuladores redundantes: aplicación a los robots Kuka LWR 4+ y ABB Yumi Revista Iberoamericana de Automática e Informática Industrial RIAI 15(2) 192-202
[11] Sietra J, Medina B, Rodríguez L, Pacheco C, Barrios M 2019 Desarrollo e implementación de tecnologías biomédicas para la telerehabilitación funcional Revista Colombiana de Tecnologías de Avanzada 2(34) 137-143
[12] Gutierrez E, Pardo C, Sosa L, Jiménez F 2014 Robot dibujante controlado mediante el sensor kinect Revista Colombiana de Tecnologías de Avanzada 2(24) 43-48
[13] Duran D, Hernández J, Peña C 2012 Avances en el control de plataformas bípedas usando el sistema kinect Revista Colombiana de Tecnologías de Avanzada 2(22) 117-124
[14] MathWorks 2020 Skeleton Viewer for Kinect V2 Skeletal Data (United State of America: MathWorks)