

**ANVANCES EN LA PROGRAMACIÓN DE ROBOTS INDUSTRIALES
APLICANDO TECNICAS DE GUIADO GESTUAL**

**ADVANCES IN INDUSTRIAL ROBOTS PROGRAMMING APPLYING
GESTUAL GUIDANCE TECHNIQUES**

Ing. Andrés Felipe Padilla Mantilla¹, PhD. César Peña¹, PhD. Gonzalo Moreno²

¹Universidad de Pamplona

Facultad de Ingenierías y Arquitectura,
Grupo de Investigación de Automatización y Control.
Ciudadela Universitaria. Pamplona, Norte de Santander, Colombia.
Tel.: (+577) 5685303, Fax: (+577) 5685303, Ext. 144.
E-mail: {andres.padilla, cesarapc}@unipamplona.edu.co.

²Universidad de Pamplona

Facultad de Ingenierías y Arquitectura,
Grupo de Investigación en Ingeniería Mecánica de la Universidad de Pamplona – GIMUP.
Ciudadela Universitaria. Pamplona, Norte de Santander, Colombia.
Tel.: (+577) 5685303, Fax: (+577) 5685303, Ext. 144.
E-mail: gmoren@unipamplona.edu.co.

Resumen: En este artículo se presenta la visión artificial como aplicación para la generación de trayectorias a través de la captura de movimientos de un operador utilizando una cámara de luz estructurada (Kinect) y aplicando un algoritmo de reconocimiento de las articulaciones del cuerpo se hace una representación tridimensional del esqueleto, tomando como puntos de interés la articulación de la mano izquierda para generar los comandos gestuales y la mano derecha para indicar la trayectoria que debe ejecutar el robot. A través de un entorno de desarrollo donde se incorpora el modelo cinemático del robot se obtienen los resultados de simulación que indican que es posible generar una trayectoria por medio del cuerpo humano utilizando visión artificial.

Palabras clave: Trayectoria, algoritmo, robot, interfaz, cámara de luz estructurada.

Abstract: In this paper, artificial vision is presented as an application for the generation of trajectories through the capture of movements of an operator 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 trajectory that the robot must execute. Simulation results are obtained through a development environment where the kinematic model of the robot is incorporated, indicating that it is possible to generate a trajectory through the human body using artificial vision.

Keywords: Trajectory, algorithm, robot, interface, structured light camera.

1. INTRODUCCIÓN

Los robots industriales han existido desde finales de los años 50 y han ido evolucionando para adaptarse a las necesidades de la industria moderna. La primera generación de robots se trató de complejos sistemas mecánicos que debían ser manipulados por un operario y tenían un simple sistema de control, la segunda generación trajo consigo los robots de aprendizaje, los cuales repiten una secuencia de movimientos ejecutados previamente mediante un código, la tercera generación introduce los dispositivos sensoriales que ayudan al robot a interactuar con el entorno de trabajo, y por último a la cuarta generación pertenecen los robots móviles o inteligentes, acercándonos a la inteligencia artificial (Sánchez Martín, y otros, 2007).

Un robot industrial tiene la capacidad de adaptarse a diversas tareas en una empresa reprogramándose. La programación de un robot se define como el proceso mediante el cual se le indica a este la secuencia de acciones que deberá llevar a cabo durante la realización de determinada tarea (Barrientos, Peñin, Balaguer, & Aracil, 2007)

La programación es, por tanto, la herramienta con la que cuenta el usuario para programar la tarea que el robot debe realizar, donde se relaciona de manera directa las características físicas del robot y las posibilidades de programación (Barrientos, Peñin, Balaguer, & Aracil, 2007). A continuación, se define los dos tipos de programación más usados actualmente.

La primera es la programación por guiado o aprendizaje consiste en manipular la trayectoria o puntos deseados en los que se quiere mover el robot al mismo tiempo que se registran las configuraciones adaptadas, para su posterior reproducción automática (García, Cárdenas, Rendón, & Maya Méndez, 2008).

La segunda es la programación textual, la cual utiliza un lenguaje de programación específico para editar ordenes en las tareas del robot que posteriormente son ejecutadas (Cantillo Maldonado, Gualdron Guerrero, & Ortiz SAndoval, 2018).

En el desarrollo de este proyecto se utiliza el método de programación por guiado, incluyendo nuevas tecnologías como la cámara de luz estructurada con la aplicación de técnicas de visión artificial, para programar la trayectoria sin necesidad de trasladar fuera de la línea de operativa al robot.

Este proyecto se basa en una plataforma de interfaz de usuario en la que se define la trayectoria a partir de los movimientos capturados por un *Kinect* bajo un sistema de referencia que simula la trayectoria del robot, que posteriormente debe ser ajustada y guardada para su programación.

Para esta aplicación se utilizó el robot *Raplim*, el cual es de tipo antropomórfico y cuenta con 4 grados de libertad, este robot se tomó como modelo de prueba para implementar el método de programación basado en visión artificial con el fin de responder a la pregunta ¿es posible generar una trayectoria de guiado a partir de referencias ubicadas en el propio cuerpo del operador mediante el uso de visión artificial?.

Teniendo en cuenta todos estos aspectos, en el capítulo 2 se describe el modelo cinemático del robot, con el fin de conocer su estructura mecánica y la relación entre los movimientos de sus articulaciones y el efector final, posteriormente en el capítulo 3 se define el algoritmo de programación con el que se genera la trayectoria, por último, en los capítulos 4 y 5 se mencionan los resultados y conclusiones respectivamente.

2. CINEMÁTICA DEL ROBOT MANIPULADOR

Cada robot tiene características particulares, por lo que es necesario efectuar los cálculos matemáticos con los que se desarrolla el algoritmo de generación de trayectorias del robot, que represente los movimientos articulares de cada eslabón con respecto a la posición y orientación del efector final (Rodríguez, Botero, & Quintero, 2007).

Para encontrar la relación entre las coordenadas articulares, la posición y orientación del efector final, se resuelve la cinemática directa e inversa del robot *Raplim* de la Universidad de Pamplona (Torres, y otros, 2013).

2.1 Cinemática directa

Conociendo la morfología y estructura del robot *Raplim* se obtienen las longitudes de los eslabones y los tipos de articulaciones con los cuales es posible determinar la posición y orientación del efector final del robot, con respecto a un sistema de referencia fijo (base del robot).

Para hacer el cálculo de la cinemática directa se usa una serie de reglas conocidas como el algoritmo de *Denavit Hartenberg*, que toma cuatro transformaciones básicas que permiten relacionar el sistema de referencia del eslabón i con el sistema de referencia del eslabón $i-1$, el producto de estas cuatro transformaciones da como resultado la matriz que se describe en la ecuación 1 (Ramírez Arias & Rubiano Fonseca, 2012).

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & \alpha_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

En la Figura 1 se detalla el modelo simplificado del robot *Raplim* y la orientación de los ejes coordenados correspondientes a cada eslabón (Mejía Bugallo, Peña Cortes, & Riaño Jaimes, 2016).

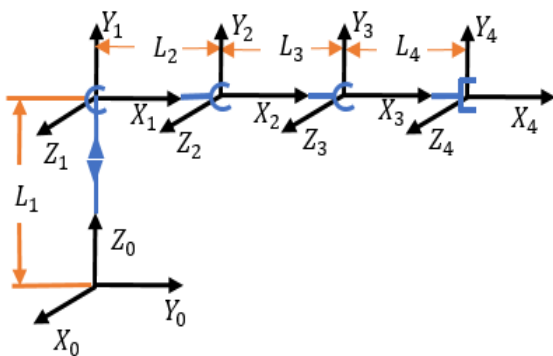


Figura 1. Modelo simplificado del robot *Raplim*.

De acuerdo a la Figura 1 se obtienen los parámetros de *Denavit Hartenberg* para el robot *Raplim* como se muestra en la Tabla 1 (Duque Suárez, Ferreira Pacheco, & Puentes Velásquez, 2020).

Tabla 1. Parámetros D - H del robot.

Articulación	θ	d	A	α
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

Los valores de la Tabla 1 se reemplazan dentro de la matriz de *Denavit Hartenberg*, es decir, debe haber una matriz por cada articulación. A partir del producto de las cuatro matrices se obtiene la matriz que indica la orientación y posición del efector final con respecto al sistema de referencia de la base del

robot como se observa en la ecuación 2 (Zaplama, Arnau Claret, & Basanez, 2018).

$${}^0A_4 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2 Cinemática inversa

La cinemática inversa como su nombre lo indica, busca encontrar la forma que debe adoptar el robot a partir de la posición y orientación del efector final, es decir, lo contrario a la cinemática directa.

Para obtener los valores de las coordenadas articulares del robot a partir de la posición y orientación del efector final con respecto al sistema de referencia de su base, se considera resolver por el método geométrico los primeros 3 grados de libertad con los que cuenta el robot *Raplim* (Zaplama, Arnau Claret, & Basanez, 2018), y el último grado se resuelve por el método vectorial. Esta estrategia se usa debido a que el método geométrico funciona solo cuando el robot tiene pocos grados de libertad.

Partiendo de lo anterior se comienza a resolver la cinemática inversa de los primeros 3 grados de libertad del robot *Raplim*. En la Figura 2 se puede observar la vista superior del modelo simplificado para una posición del efector final $P = [P_x; P_y; P_z]$, con la cual se puede determinar la primera coordenada articular q_1 (Ramírez Arias & Rubiano Fonseca, 2012).

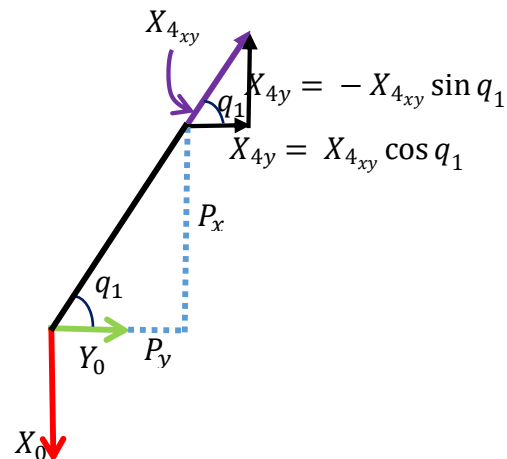


Figura 2. Diagrama para determinar coordenada articular q_1 .

Por consiguiente, con las ecuaciones 3 - 5 se obtiene el valor de la articulación q_1 .

$$\sin q_1 = -P_x \quad (3)$$

$$\cos q_1 = P_y \quad (4)$$

$$q_1 = \tan^{-1} \left(\frac{-P_x}{P_y} \right) \quad (5)$$

Para calcular q_2 y q_3 se ubica un punto p_m (punto de la muñeca) que se puede observar en la Figura 3 con el que se realizan los cálculos geométricos.

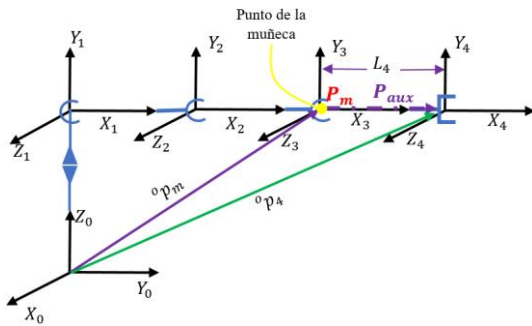


Figura 3. Punto de la muñeca.

Donde

P_m = Punto de la muñeca

P_{aux} = Vector auxiliar

A partir de la Figura 3 se puede determinar el vector P_m mediante la suma de vectores, como se desarrolla en las ecuaciones 6 - 8.

$$P = P_m + P_{aux} \quad (6)$$

$$P_{aux} = X_4 * L_4 \quad (7)$$

$$P_m = P - (X_4 * L_4) \quad (8)$$

Con la ecuación 8 se pueden determinar las componentes de P_m , para lo cual se necesita el valor del término X_4 , el cual se determina según lo indicado en la Figura 2 y Figura 4.

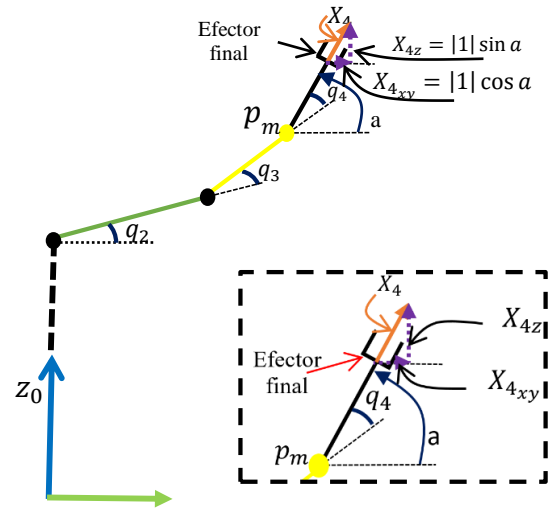


Figura 4. Diagrama para determinar las componentes de X_4 .

En la Figura 4 se detalla el ángulo (a) que determina los grados de giro del efector final con respecto al sistema de referencia de la base del robot. Este valor lo define el operador del robot y junto con las coordenadas cartesianas del efector final son los datos necesarios para calcular la cinemática inversa. De donde se obtienen las componentes de X_4 , (Ec. 9).

$$\begin{bmatrix} X_{4x} \\ X_{4y} \\ X_{4z} \end{bmatrix} = \begin{bmatrix} [\cos(a)] * [\sin(q_1)] \\ [\cos(a)] * [\cos(q_1)] \\ [\sin(a)] * 1 \end{bmatrix} \quad (9)$$

Con estos valores se determina el punto de la muñeca y se calcula q_2 .

Seguidamente en la Figura 5 muestra la vista superior del punto de la muñeca de la Figura 3, de donde se obtiene la ecuación 10 para determinar la distancia (m) necesaria para calcular las últimas dos coordenadas articulares.

$$m = \sqrt{(p_{mx})^2 + (p_{my})^2} \quad (10)$$

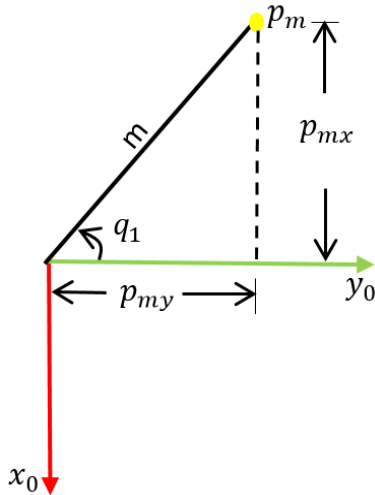


Figura 5. Primera articulación.

Por último, las ecuaciones para determinar las coordenadas articulares q_2 y q_3 se apoyan en la Figura 6 que ilustra la vista lateral del robot.

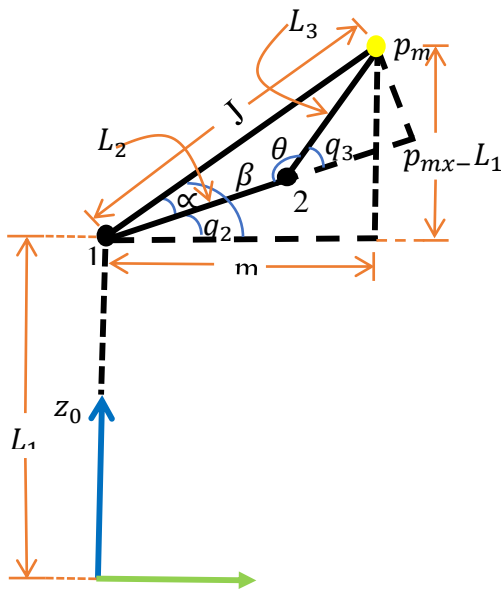


Figura 6. Vista lateral del robot.

Para calcular q_3 se inicia hallando la distancia entre 1 y P_m , como se muestra en la ecuación 11.

$$J = \sqrt{(m)^2 + (p_{mz} - L_1)^2} \quad (11)$$

Teniendo en cuenta que el ángulo θ de la Figura 6 es $180 - q_3$ se aplica la ley de coseno para despejar J (Ec.12):

$$J^2 = (L_2)^2 + (L_3)^2 - 2(L_2 L_3) \cos(180 - q_3) \quad (12)$$

Reorganizando la ecuación 12 se despeja $\cos q_3$. (Ec. 13)

$$\cos q_3 = \frac{J^2 - L_2^2 - L_3^2}{2 * L_2 * L_3} \quad (13)$$

Utilizando la identidad trigonométrica \tan^{-1} se despeja q_3 tal como se muestra en las ecuaciones 14 y 15.

$$q_3 = \tan^{-1} \left(\frac{\sin q_3}{\cos q_3} \right) \quad (14)$$

$$q_3 = \tan^{-1} \left(\frac{\sqrt{1 - \left(\frac{J^2 - L_2^2 - L_3^2}{2 * L_2 * L_3} \right)^2}}{\frac{J^2 - L_2^2 - L_3^2}{2 * L_2 * L_3}} \right) \quad (15)$$

Con el valor de q_3 se procede a calcular la coordenada articular q_2 tomando en cuenta la relación de las ecuaciones 16 - 18.

$$q_2 = \beta - \alpha \quad (16)$$

$$\beta = \tan^{-1} \left(\frac{p_{mz} - L_1}{m} \right) \quad (17)$$

$$\alpha = \tan^{-1} \left(\frac{L_3 * \sin q_3}{L_2 + (L_3 * \cos q_3)} \right) \quad (18)$$

3. VISIÓN ARTIFICIAL

Las nuevas tecnologías como la visión artificial proporcionan innovación en la programación de robots. El uso de esta tecnología encaja en el tipo de programación por guiado, descrita con anterioridad, con la que capturamos movimientos a través de una cámara de luz estructurada y con ayuda de algunos algoritmos se logra reproducir una trayectoria en el robot.

3.1 Captura de movimientos con Kinect

Con ayuda de una cámara de luz estructurada (Kinect) se logra identificar las partes del cuerpo del operador con los que se elaboran comandos y movimientos que luego puedan ser reproducidos

con ayuda de algoritmos de programación en un robot (Sierra, Medina, Rodríguez, Pacheco, & Barrios, 2019).

Lo que el Kinect captura es una serie de puntos específicos que corresponden a cada parte del cuerpo y con los que se logra asignar un comando a acción cuando se registre el movimiento de un punto específico del cuerpo (Gutierrez Cáceres, Pardo Beainy, Sosa Quintero, & Jiménez López, 2014).

Como se entiende en el control por guiado, es necesario que un operador le muestre al robot el camino que debe recorrer manipulándolo de forma directa como en la programación por guiado pasivo. De acuerdo a lo anterior se propone homologar la metodología con ayuda de la visión artificial para capturar los movimientos del operador y programarlos, sin que haya un contacto directo con el robot.

Para capturar los movimientos se utiliza el sensor *Kinect* que es una cámara de luz estructurada que permite adquirir la representación tridimensional del cuerpo del operador (Duran López, Hernández Amaya, & Peña Cortes, 2012). La Figura 7 describe la captura e interpretación de los movimientos del operador a través del sensor Kinect.

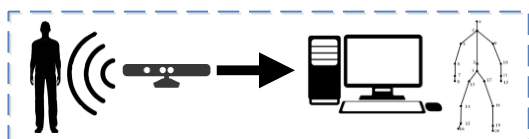


Figura 7. Captura de los movimientos.

Internamente el Kinect posee un algoritmo que le permite la identificación de un cuerpo en el espacio y adicional a esto capturar, obtener o mostrar ciertos puntos específicos en el mismo, así como se muestra en la Figura 8 (Rodriguez, Botero, & Quintero, 2007).

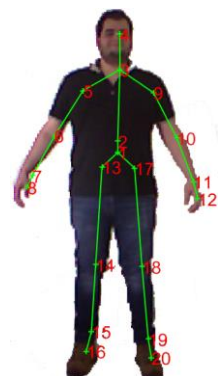


Figura 8. Puntos del cuerpo del operador tomados por el Kinect.

Se especifican los puntos que el sensor *Kinect* reconoce del cuerpo para que a través del algoritmo desarrollado se pueda hacer la representación tridimensional de estos puntos en el entorno de desarrollo, dando forma al esqueleto del operador, tal como se muestra en la Figura 8 (MathWorks, 2020).

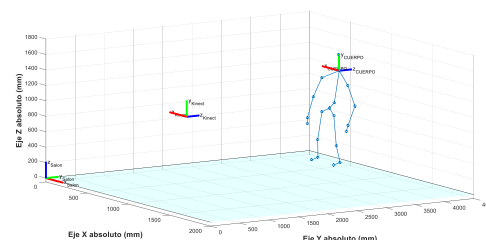


Figura 9. Cuerpo tridimensional visualizado en el entorno de desarrollo.

La Figura 9 muestra el esqueleto del operador desde la vista superior y para ajustar la figura tridimensional es necesario hacer una transformación de los puntos a partir del sistema de referencia del espacio de trabajo donde se encuentra ubicado el robot, debido a que es diferente al sistema de coordenado del Kinect como se muestra en la Figura 10.

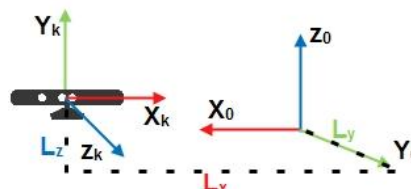


Figura 10. Sistema coordenado del Kinect.

A continuación, se muestra la matriz de transformación homogénea con la que se transforma el sistema coordenado del *Kinect* con respecto al sistema coordenado del salón (Ec. 19).

$${}^0A_K = \begin{bmatrix} -1 & 0 & 0 & L_x \\ 0 & 0 & 1 & L_y \\ 0 & 1 & 0 & L_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

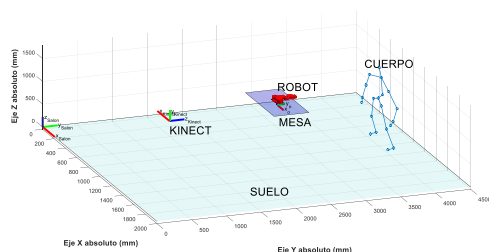


Figura 11. Simulación del espacio en el entorno de desarrollo.

La Figura 11 muestra el ambiente virtual que se programa en el software matemático, donde se puede visualizar la ubicación de los sistemas coordenados del salón, del Kinect y el robot con sus respectivos puntos (Duran López, Hernández Amaya, & Peña Cortes, 2012).

La recolección de los datos a través del *Kinect* de los movimientos de las articulaciones las hace una sola persona, por lo que es necesario programar comandos con ciertas poses del cuerpo para ejecutar acciones que inicien el programa, pausen y finalicen el proceso de recolección de datos, todo sin tener contacto directo con el robot ni el ordenador con él que se simula la trayectoria y se ejecuta el programa. La Figura 12 muestra la pose de inicio y la Figura 13 muestra la pose para finalizar.

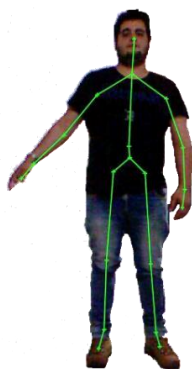


Figura 12. Pose para el comando iniciar.



Figura 13. Pose para el comando finalizar registro.

El comando de inicio tiene 2 condiciones, la primera es que la muñeca izquierda debe alejarse a una distancia mínima del cuerpo que inicia la segunda condición, que es mantener esta posición mínimo durante 5 segundos para iniciar el registro de la trayectoria que indica la muñeca derecha, pero sin interrumpir la primera condición, es decir, la posición de la muñeca izquierda no debe cambiar durante el registro de la trayectoria, porque pausa el registro e inicia nuevamente en la posición en la que se encuentre la muñeca derecha.

Una vez ejecutada la trayectoria a través del movimiento de la muñeca derecha se debe finalizar el registro, esto se hace elevando la muñeca izquierda por encima de la cabeza, es decir, la muñeca izquierda debe moverse hacia la parte superior de la cabeza para finaliza el proceso de registro.

finalizando el programa utilizando el comando respectivo, se genera una matriz que almacena las posiciones de la muñeca derecha (*Trayectorias*) y el tiempo de ejecución. Para definir los puntos fuera y dentro del espacio de trabajo del robot *Raplim*, aplicamos la cinemática inversa y evaluamos las restricciones a cada posición almacenada en la matriz.

4. RESULTADOS

Al capturar los movimientos del operador a través del *Kinect* se genera la trayectoria en el entorno de desarrollo como se observa en la Figura 14, donde se diferencian los puntos azules de los rojos como la posición que está al alcance o no del robot respectivamente.

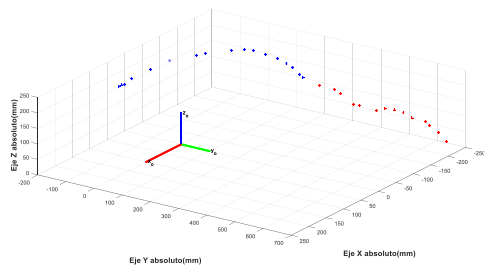


Figura 14. Grafica de los puntos en el espacio de trabajo del robot.

Definidos los puntos que indican la trayectoria se hace la simulación, donde se visualiza que el robot reproduce los movimientos muy bruscos, por lo que se genera un conjunto de puntos intermedios entre los ya establecidos para mejorar este movimiento. Cabe destacar que el conjunto de puntos generados entre cada par de puntos de la trayectoria original se evalúa nuevamente para asegurar que no estén fuera del espacio de trabajo del robot, si es así, el robot se queda en la última posición alcanzable, hasta que encuentre un punto en la trayectoria que este dentro de su espacio de trabajo. La Figura 15 muestra la trayectoria continua que debe ejecutar el robot.

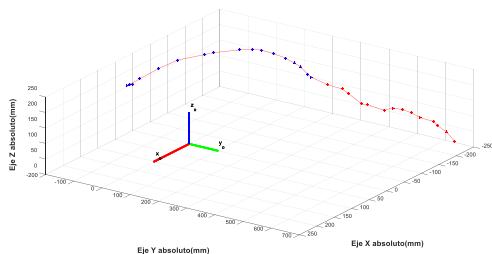


Figura 15. Trayectoria linealizada.

La Figura 16 representa la simulación que demuestra que los movimientos del robot son suaves y no colisiona con el mismo, garantizando que ejecute sin problemas la trayectoria en un entorno real.

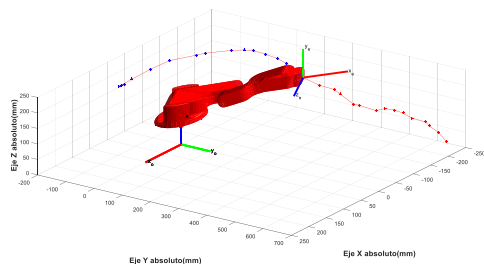


Figura 16. Simulación en entorno virtual de la ejecución de la trayectoria.

5. CONCLUSIONES

El algoritmo de visión artificial permite identificar los puntos significativos ubicados en el cuerpo del operador, con los cuales es posible generar un conjunto de puntos que representen un movimiento, al cual se le aplica un método de control cinemático que genera la trayectoria ejecutable por el robot.

La cámara de luz estructurada captura los movimientos con precisión, inclusive aquellos movimientos leves e involuntarios como vibraciones, generando fluctuaciones en la trayectoria al procesar la captura.

Al capturar los datos se genera un gran número de posiciones que debe ejecutar el robot en su trayectoria lo que implica un tiempo considerable a la hora de ejecutar el movimiento, por lo tanto, en trabajos futuros se debe considerar el uso de un algoritmo de optimización que permita simplificar o reducir el número de datos prediciendo la ruta que está indicando el operador.

REFERENCIAS

- Barrientos, A., Peñin, L. F., Balaguer, C., & Aracil, R. (2007). *Fundamentos de robótica*. Madrid: Mc Graw-Hill.
- Cantillo Maldonado, A., Gualdron Guerrero, O., & Ortiz SAndoval, J. (2018). Procesamiento de señales emg en un sistema embeido para el control neuronal de un brazo robótico. *Revista Colombiana de Tecnologías de Avanzada*, 143.
- Duque Suárez, O. M., Ferreira Pacheco, J., & Puentes Velásquez, A. M. (2020). Diseño e implementación de us sistema de control cinemático con OPEN-Harware del manipulador didactico pegasus de marca amatrol ubicado en tecnoacademia Cúcuta. *Revista Colombiana de Tecnologías de Avanzada*, 147.
- Duran López, D. A., Hernández Amaya, J. E., & Peña Cortes, C. A. (2012). Avances en el control de plataformas bípedas usando el sistema kinect. *Revista Colombiana de Tecnologías de Avanzada*, 121-123.
- García, M. A., Cárdenas, A., Rendón, J. M., & Moya Méndez, M. (2008). Una Plataforma de Control Basado en Visión para la Rehabilitación de Robots Manipuladores

- de Tipo Industrial. *Computación y Sistemas*, 412.
- Gutierrez Cáceres, E. A., Pardo Beainy, C. E., Sosa Quintero, L. F., & Jiménez López, F. R. (2014). Robot dibujante controlado mediante el sensor kinect. *Revista Colombiana de Tecnologías de Avanzada*, 45.
- MathWorks. (20 de Julio de 2020). *MathWorks help center*. Obtenido de https://www.mathworks.com/help/support/pkg/kinectforwindowsruntime/ug/plot-skeletons-with-the-kinect-v2.html?s_tid=srchtitle
- Mejía Bugallo, D. A., Peña Cortes, C. A., & Riaño Jaimes, C. (2016). Sistema solar para la operación de un robot agrícola. *Revista Colombiana de Tecnologías de Avanzada*, 37.
- Ramirez Arias, J. L., & Rubiano Fonseca, A. (2012). Modelamiento matemático de la cinemática directa e inversa de un robot manipulador de tres grados de libertad. *ingeniería solidaria*, 47-48.
- Rodriguez, C. F., Botero, J. C., & Quintero, H. (2007). aplicación de la teoría de robots manipuladores a la biomecánica del brazo humano. *Revista Iberoamericana de Automatica e Informatica Industrial RIAI*, 52.
- Sánchez Martín, F. M., Jiménez Schlegl, P., Millán Rodríguez, F., Salvador Bayarri, J., Manllau Font, V., Palou Redorta, J., & Villavicencio Mavrich, H. (2007). Historia de la robótica: de Arquitas de Tarento al Robot da Vinci (Parte II). *SciELO Analytics*, 186-192.
- Sierra, J. E., 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*, 139.
- Torres, C. J., Archila, J. F., Tronco, M. L., Becker, M., Viera Porto, A. J., & Tiberti, A. J. (2013). estudio cinemático de una plataforma robótica para agricultura. *Revista Colombiana de Tecnologías de Avanzada*, 133-134.
- Zaplama, I., Arnau Claret, J., & Basanez, L. (2018). Analisis Cinemático de Robots Manipuladores Redundantes: Aplicación a los Robots Kuka LWR4+ y ABB Yumi. *Revista Iberoamericana de Automático e Informática Industrial RIAI*, 198.