

Análisis cinemático e instrumentación virtual 3D para el brazo robótico Pegasus II

Kinematic analysis and 3D virtual instrumentation for the Pegasus II robot

Nelson Santiago Giraldo Bustamante^{1*} Gustavo Adolfo Patiño Álvarez¹

Recibido 08 de mayo de 2019, Aceptado 15 de junio de 2019

Received: May 08, 2019 Accepted: June 15, 2019

RESUMEN

En el presente artículo se describe una formulación lógica y matemática para la instrumentación virtual 3D del brazo robótico Pegasus II mediante Matlab, haciendo uso de las técnicas proporcionadas por la cinemática de posición directa, y operaciones respectivas de traslación y rotación. Además de ello, se describe un procedimiento geométrico-vectorial para resolver el problema de la cinemática de posición inversa para el mismo brazo robótico, a fin de que la pinza de este siga una trayectoria en línea recta entre el punto original en que se encuentra, y un punto (x, y, z) dado, realizando aproximaciones pertinentes que faciliten su solución. Se explica por qué este procedimiento es útil para resolver el problema cinemático inverso y la ventaja que representa con respecto a otros procedimientos. Finalmente, se presentan los resultados obtenidos al realizar la instrumentación virtual en tiempo-real de cinemática inversa sobre el brazo robótico Pegasus II, comprobando la eficiencia del procedimiento descrito.

Palabras clave: Brazo robótico, cinemática directa, cinemática inversa, instrumentación virtual, rotación 3D, robot Pegasus II.

ABSTRACT

In this paper, a mathematical and logical formulation for the 3D virtual instrumentation of the Amatrol's Pegasus II robot on Matlab is described, making use of the techniques provided by direct position kinematics, and respective translational and rotational operations. Additionally, a geometricvectorial procedure is described to solve the problem of inverse position kinematics for the same robotic arm, so that the robotic arm follows a straight line trajectory between the original point in which it is located, and a given point (x, y, z), making pertinent approximations that facilitate such solution. It is explained why this procedure is useful to solve the inverse kinematic problem and the advantage it represents concerning other procedures. Finally, the results obtained by the real-time virtual instrumentation of inverse kinematics on the Pegasus II Robot Arm are presented, verifying the efficiency of such described procedure.

Keywords: Robotic arm, direct kinematics, inverse kinematics, virtual instrumentation, 3D rotation, Pegasus II robot.

¹ Universidad de Antioquia. Departamento de Ingeniería electrónica y Telecomunicaciones. Medellín, Colombia.
E-mail: giraldobustamante@gmail.com, adolfo.patino@udea.edu.co

* Autor de correspondencia: giraldobustamante@gmail.com

INTRODUCCIÓN

Durante cientos de años, los pueblos y su gente han tratado de construir mecanismos que imiten partes del cuerpo humano, inicialmente con fines de protección y defensa, y posteriormente con el objetivo de facilitar todas las tareas repetitivas que se suele tener un trabajador en un entorno industrial [1]. Todos estos mecanismos fueron evolucionando a la par con los avances científicos de cada época, llegando así a convertirse en la robótica moderna que hoy se conoce [1].

Existen diversos tipos de sistemas robóticos, entre ellos se encuentran los brazos robóticos. Un brazo robótico puede entenderse como un sistema electromecánico diseñado para realizar funciones y tener características semejantes a las de un brazo humano. Al realizar un movimiento en alguna articulación del sistema se genera un movimiento del extremo final del brazo (el equivalente a su “mano”), posicionándolo en una orientación dada. Es de un gran interés en la gran mayoría de aplicaciones conocer la ubicación espacial de dicho extremo final en un sistema coordenado (x , y , z) de un determinado observador (cinemática directa), o lograr que este extremo final llegue de alguna manera a un punto (x , y , z) deseado (cinemática inversa), a fin de realizar cierto tipo de tareas.

En este trabajo se propone resolver los problemas de la cinemática directa y la cinemática inversa del brazo robótico Pegasus II, fabricado por la empresa Amatrol [2], el cual fue adquirido por el Laboratorio de Automatización y Control de la Universidad de Antioquia (Medellín, Colombia). El problema de la cinemática directa ha sido resuelto a fin de observar gráficamente, mediante una interfaz gráfica, la ubicación del extremo final del brazo robótico, utilizando operaciones elementales de rotación y traslación. Por otro lado, la solución al problema de la cinemática inversa ha sido resuelta en este trabajo a fin de lograr que el extremo final del brazo robótico se mueva entre el punto de origen donde se encuentra, y otro punto (x , y , z) del espacio, siguiendo una trayectoria rectilínea.

Existen diversos métodos para encontrar la cinemática inversa de un sistema robótico, uno de ellos puede ser utilizando algoritmos

genéticos, tal como lo propone Giraldo, Delgado y Castellanos [3]. Cabe destacar que este método para encontrar una solución a la cinemática inversa es muy útil en el sentido de que es capaz de calcularse para un brazo robótico complejo de una manera relativamente precisa, no obstante, la solución requiere un mínimo de 40 iteraciones o generaciones a fin de encontrar dichos valores, pudiendo requerir más iteraciones dependiendo del criterio de parada (precisión). En el caso del brazo robótico Pegasus II el criterio más óptimo encontrado fue una solución analítica geométrica, dado que ésta presenta una precisión mucho mayor sin necesidad de requerir un mayor tiempo de cómputo debido a que las ecuaciones a resolver son siempre las mismas.

Otro método posible para resolver el problema de la cinemática inversa es el método numérico o el método analítico-numérico, el cual mezcla soluciones analíticas con soluciones numéricas, tal como la propuesta por Zalana, Claret y Basanez [4] para el manipulador redundante ABB Yumi. Este tipo de métodos son utilizados para encontrar una solución a la cinemática inversa de sistemas robóticos complejos cuya solución analítica es muy complicada de hallar, o simplemente no existe. De nuevo, en el caso del brazo robótico Pegasus II, el cual no es un sistema robótico tan complejo, es posible encontrar una solución analítica con algunas suposiciones que evitan tener que utilizar soluciones numéricas, las cuales por su naturaleza iterativa no son óptimas en situaciones prácticas con exigencia de respuesta en tiempo real [5].

Este artículo se ha estructurado de la siguiente manera: en la sección siguiente se presenta una breve descripción teórica del problema de la cinemática y de sus ecuaciones. A continuación, se presenta una breve descripción del trabajo realizado en el brazo robótico y algunas consideraciones importantes sobre la arquitectura de control del mismo. Luego, se describe todo el procedimiento y la forma de las ecuaciones que se obtuvieron para la implementación de los controladores cinemáticos, además de mostrar la lógica de implementación de cada algoritmo. En la siguiente sección se presentan los resultados experimentales obtenidos mediante la implementación de la solución propuesta en este artículo; y finalmente se presentan las conclusiones del artículo.

CINEMÁTICA DE UN SISTEMA ROBÓTICO

La cinemática de un manipulador robótico realiza el estudio de su movimiento en función del tiempo (posiciones, velocidades y aceleraciones), sin considerar las fuerzas que originan dicho movimiento [5].

Cinemática directa

Si se conoce el valor de la rotación de cada articulación respecto a su punto cero y las dimensiones de los eslabones, es posible mediante composición de transformaciones básicas de traslación y rotación determinar la posición y orientación del extremo del robot con respecto a un sistema de referencia fijo (Tal como la base del robot). La solución a este problema es lo que se conoce como cinemática directa [1].

Las operaciones de traslación y rotación se pueden definir en base a la multiplicación de matrices con un vector columna $(x, y, z, 1)$, tales matrices son:

$$Tras = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$Rotx = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$Roty = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$Rotz = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4)$$

La matriz (1) realiza una traslación desde el punto (x, y, z) original en el que se encuentra el vector a un nuevo punto $(x+a, y+b, z+c)$. La matriz (2) realiza

una rotación α con respecto al eje X; (3) realiza una rotación β con respecto al eje Y; (4) realiza una rotación γ con respecto al eje Z. La deducción de dichas matrices puede encontrarse en la referencia [5].

Cinemática Inversa

Ahora, supóngase que el extremo del robot tiene que realizar una trayectoria entre 2 puntos A y B, donde A es el punto donde este se encuentra (punto inicial). El problema ahora consiste en determinar qué valores deben tomar los parámetros de cada una de las articulaciones del robot para que el extremo del mismo siga la trayectoria deseada. Este problema es conocido como cinemática de posición inversa [1].

DESCRIPCIÓN DEL BRAZO ROBÓTICO PEGASUS II

El brazo robótico Pegasus II es un sistema de escala semi-industrial, es decir, es un sistema más pequeño y menos complejo que el utilizado en la industria, dado que ha sido pensado para actividades menos demandantes. Dicho sistema puede utilizarse para representar cómo es el funcionamiento general de los brazos articulados, y cuáles son sus principales aplicaciones. Este sistema incluye un brazo *servo* robot articulado de 5 ejes, donde todas sus articulaciones son de tipo rotacional, una pinza, control, cables de conexión y 6 motores DC, cada uno con un *encoder* incremental para su correcto funcionamiento [6].

Cada uno de los motores DC del brazo robótico cuentan con una etapa de potencia y su controlador respectivo. La etapa de potencia consiste en un conjunto de puentes H y opto acopladores los cuales se utilizan para controlar la velocidad de los motores a través de señales PWM (*Pulse Width Modulation*). Los sistemas de control de potencia son sistemas realimentados (a través de las señales de los *encoders*) los cuales tienen como objetivo controlar la posición angular de cada articulación del sistema. La posición del sistema se ingresa en posición absoluta, es decir, un sistema de coordenadas fue definido para cada articulación y a partir de allí se ingresa una orden por software de la posición a la que se quiere llegar respecto al “cero” definido anteriormente.

El controlador del sistema de potencia realiza entonces una rampa de encendido, luego mantiene una velocidad crucero y posteriormente realiza una rampa de apagado en cada motor, dependiendo

del punto angular en el que se encuentra. Dicho comportamiento se realiza mediante un par de controladores tipo Proporcional (Tipo P) los cuales se activan dependiendo del punto de la trayectoria donde se encuentre el eje, la velocidad cruce es una constante, si el valor de ésta es superada por la acción de control de alguno de los controladores se activa una condición lógica mediante la cual la acción de control toma el valor de la velocidad cruce. Algunos controladores poseen ecuaciones de compensación del torque, el cual es producido al momento de estar en cierta posición α . Esta compensación es requerida dado que el torque producido en ciertas posiciones afecta el comportamiento trapezoidal del sistema. Todo el controlador se encuentra implementado en una placa de desarrollo Arduino Due [7], de modo que las órdenes para cada motor se envían a través del puerto serial del mismo.

Por otro lado, a través del puerto USB se envían datos desde Matlab [8] al brazo robótico Pegasus II, con la información de control de posición necesaria de acuerdo al modelado de los controladores cinemáticos de posición directa e inversa implementados en dicha plataforma matricial.

ARQUITECTURA DE CONTROL DEL PEGASUS II

La arquitectura de control del brazo robótico Pegasus II siguió la forma mostrada en la Figura 1.

En donde se pueden apreciar dos grandes sistemas: Una interfaz computacional, la cual contiene una interfaz gráfica y el control cinemático, en la interfaz gráfica se encuentra toda la instrumentación (sliders, botones, dibujo del brazo, bloques de texto) necesaria para el funcionamiento del brazo robótico, mientras que en la parte del control cinemático se encuentran todas las ecuaciones y la lógica necesaria para encontrar el valor de los distintos ángulos de giro de cada motor dependiendo de la orden ingresada por la interfaz gráfica. Esta etapa se encarga de actualizar la forma del dibujo del robot en la interfaz y también de abrir un puerto USB y de enviar los ángulos obtenidos por allí al conversor USB a UART-TTL. Dicho conversor tiene como objetivo transformar la información entre los protocolos para que pueda llegar a la tarjeta de control de forma correcta.

El otro gran sistema es la tarjeta de control, la cual contiene un sistema de control de potencia y una etapa o sistema de potencia. Dicho sistema de control de potencia consta de 6 controladores tipo P (Proporcional) para cada uno de los 6 motores del brazo robótico en el cual se controla la variable posición angular en red cerrada. Esta etapa se encarga de recibir valores angulares para cada motor provenientes del conversor USB-UART-TTL y de acuerdo a esta información generar una acción de control determinada y enviarla al sistema de potencia. La realimentación del sistema de control se realiza

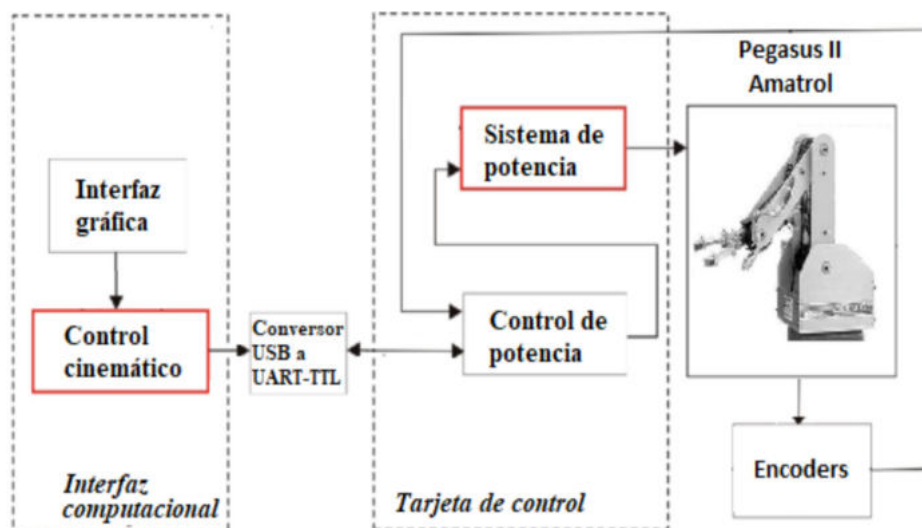


Figura 1. Arquitectura de control implementada para el brazo robótico Pegasus II.

a través del sistema de encoders. El sistema de encoders se compone de los sensores internos que trae el brazo robótico y su acondicionamiento, a través de los cuales se puede obtener la información del punto real donde se encuentra cada motor del brazo robótico. Finalmente, el sistema de potencia consta de un conjunto de opto acopladores, puentes H y protecciones de sobrecorriente, el cual recibe las señales de acción de control generadas por el Arduino Due y en base a éstas inyecta la potencia necesaria a cada motor.

CINEMATICA DEL PEGASUS II

El modelado de la cinemática del brazo robótico Pegasus II se dividió en 2 etapas: El modelo de su cinemática directa, el cual se desarrolló con el objetivo de generar una simulación del comportamiento de dicho brazo robótico, esto es, mostrar gráficamente la posición final de la pinza, por lo cual, siempre debe conservarse el sistema original de coordenadas; y el modelo de la cinemática inversa, el cual se desarrolló con el objetivo de que la pinza del brazo robótico se mueva a un punto (x, y, z) del espacio de trabajo, a través de una trayectoria, en el caso de este artículo será una línea recta. También se adicionó una trayectoria circular para comprobar el correcto funcionamiento de la cinemática inversa ante otro tipo de curvas.

Cinemática directa e instrumentación

Utilizando el software Matlab se generó un sólido a base de cilindros el cual se intentó aproximar lo más que se pudiera a la forma que tiene el brazo

robótico Pegasus II, a fin de tener una simulación lo más realista posible. La forma completa de la instrumentación se puede apreciar en la Figura 2.

Se puede observar en la instrumentación generada dos grandes bloques: Uno de ellos contiene sliders, información de cómo se encuentran los ángulos de rotación, el respectivo eje de cada una y botones, este bloque corresponde a la cinemática directa, ya que los sliders permiten modificar los valores angulares del brazo y el botón pinza permite abrir o cerrar la pinza. El otro gran bloque contiene cuadros de texto, información de la posición actual del sistema y de la curva que se va a recorrer, este bloque corresponde a la cinemática inversa, ya que los bloques de texto permiten ingresar el punto final de la curva que se desea seguir.

Ahora, para poder encontrar las ecuaciones de la cinemática directa, a cada sólido n (donde n va de 1 a 5) se le asignó una posición inicial P_n , de manera que a dicha posición es a la que se le aplican las operaciones de rotación y traslación para mover el sistema.

Se definirá la base como el sólido $n = 1$, al eslabon1 (el que sale de la base) como $n = 2$, al eslabon2 (el que sale del eslabon1) como $n = 3$, al eslabon3 (el que sale del eslabon2) como $n = 4$ y a la pinza como $n = 5$.

Con el sólido generado es posible observar que todos los eslabones giran con respecto al eje X, mientras que la base y la pinza rotan con respecto al eje Z.

Control cinemático para Pegasus II

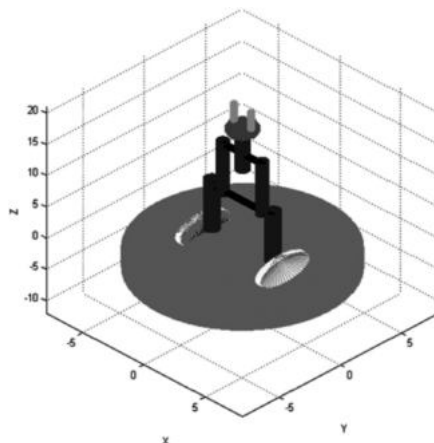


Figura 2. Interfaz de instrumentación generada para el brazo robótico.

Para llevar al sistema completo a un punto cualquiera deseado, se siguieron las siguientes reglas:

- Trasladar al origen de coordenadas a cada eslabón y rotarlo la cantidad deseada.
- Devolverlo al punto donde se encontraba antes de ser trasladado (al punto inicial).
- Aplicar todas las operaciones del sólido $n-1$.

Ecuaciones y modelado físico

Se define A_n^{n-1} como la matriz de transformación que se necesita para rotar el sólido n con respecto al $n-1$, siendo el punto $n-1=0$ el sistema original de coordenadas, es decir, el sistema respecto al cual se deben aplicar todas las operaciones. Usando las reglas definidas anteriormente, puede entenderse que para lograr rotar el sólido n con respecto al origen de coordenadas debe realizarse la operación $A_n^0 * P_n = A_{n-1}^0 * A_n^{n-1} * P_n$, en donde A_{n-1}^0 corresponde a todas las operaciones del sólido $n-1$. Se puede entonces definir en forma general:

$$A_n^{n-1} = \text{Tras}_{\text{inicial } n} * \text{Rotk}(\text{ángulo } n) * \text{Tras}_{\text{origen } n} \quad (5)$$

Debe tenerse en cuenta que para $n = 1$ y $n = 2$ no se realizan traslaciones ya que estos sólidos tienen su eje de rotación en el origen de coordenadas. Para $n = 1$ y $n = 5$ la matriz de rotación es Rot_z y para el resto de sólidos la matriz de rotación es Rot_x . De esta manera, las ecuaciones para cada sólido serán:

$$\text{base}_{[4,x1]} = A_1^0 * P_1 \quad (6)$$

$$\text{es1}_{[4,x1]} = A_1^0 * A_2^1 * P_2 = A_2^0 * P_2 \quad (7)$$

$$\text{es2}_{[4,x1]} = A_2^0 * A_3^2 * P_3 = A_3^0 * P_3 \quad (8)$$

$$\text{es3}_{[4,x1]} = A_3^0 * A_4^3 * P_4 = A_4^0 * P_4 \quad (9)$$

$$\text{pinza}_{[4,x1]} = A_4^0 * A_5^4 * P_5 = A_5^0 * P_5 \quad (10)$$

Mediante estas ecuaciones se describe entonces el movimiento completo del sólido en la instrumentación virtual definida para el brazo robótico Pegasus II.

La matriz final que permite encontrar la solución a la cinemática directa del sistema es A_4^0 , ya que la rotación en z de la pinza no tiene efecto sobre su posición final.

Adicionalmente, se encontraron los parámetros de Denavit-Hartenberg, los cuales proporcionan otra posible solución al problema cinemático directo. Dichos parámetros se dedujeron a partir de la imagen 3 y se muestran en la Tabla 1.

Tabla 1. Parámetros de Denavit-Hartenberg para el brazo Pegasus II.

Tabla de parámetros DH				
i	θ_i	d_i	a_i	α_i
1	θ_1	L1	0	90°
2	θ_2	0	L2	0
3	θ_3	0	L3	0
4	θ_4	0	L4	0

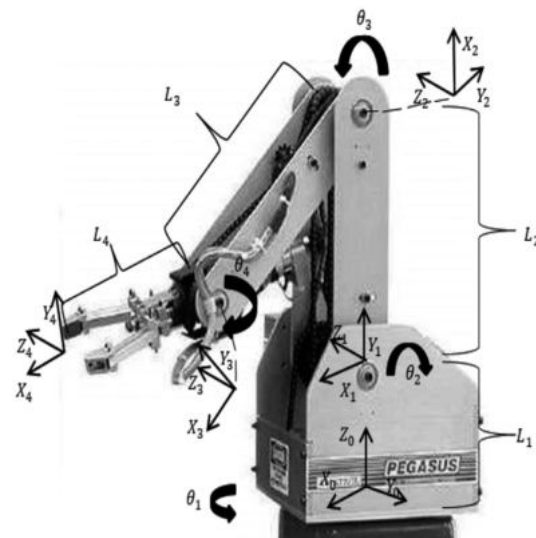


Figura 3. Sistemas coordenados, ángulos y longitudes utilizados para encontrar los parámetros DH.

Implementación del algoritmo de cinemática

El algoritmo para la implementación de la instrumentación virtual mediante la cinemática directa presentada anteriormente, sigue el diagrama de flujo mostrado en la Figura 4.

En el cual pueden observarse varias etapas: Una etapa de generar sólidos, en la cual se generan todos los sólidos en el punto cero y se guarda su posición inicial para luego aplicarle las operaciones de rotación y traslación necesarias a esta, también se inicializan las variables y se generan los sliders

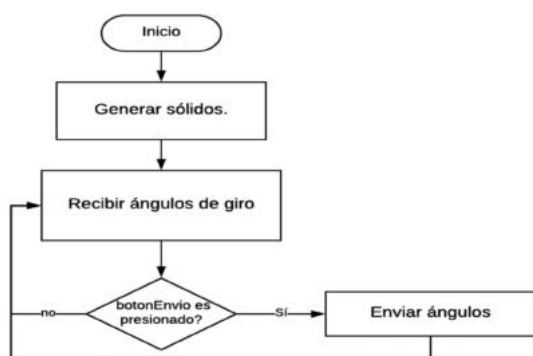


Figura 4. Diagrama de flujo que ilustra la forma de implementar la instrumentación virtual del brazo robótico Pegasus II.

y botones necesarios en la interfaz gráfica. Luego aparece una etapa de recibir ángulos de giro, en dicha etapa se espera el ingreso de los ángulos de giro de cada uno de los motores, se aplican las operaciones de traslación y rotación para cada eje dependiendo de los ángulos ingresados y en base a esto, se actualiza la interfaz gráfica mostrando la nueva posición del brazo robótico. Todo esto se realiza mientras el botón de envío no ha sido presionado. Una vez el botón de envío es presionado, se llega a la etapa de enviar ángulos, la cual se encarga de colocar un identificador a cada ángulo de cada eje y enviarlos a través del puerto USB hacia el brazo robótico real. Una vez culminada esta etapa se vuelve al bloque recibir ángulos de giro y se repite el proceso.

Cinemática inversa

Para resolver el problema de la cinemática inversa del brazo robótico, se utilizaron métodos geométricos y vectoriales en conjunto a fin de encontrar ecuaciones que relacionen los ángulos a girar de cada articulación con relación punto (x, y, z) al que se quiere llegar. También se encontraron las ecuaciones de las trayectorias deseadas (línea recta y círculo).

Ecuaciones y modelado físico

Considere un sistema coordenado de un observador, el cual tendrá como origen la base del Pegasus II, como se muestra en la Figura 5. Para poder lograr que el brazo recorra una curva dada, primero deben encontrarse ecuaciones que expresen los ángulos que debe girar cada articulación como una función del punto (x, y, z) al que se desea llegar.

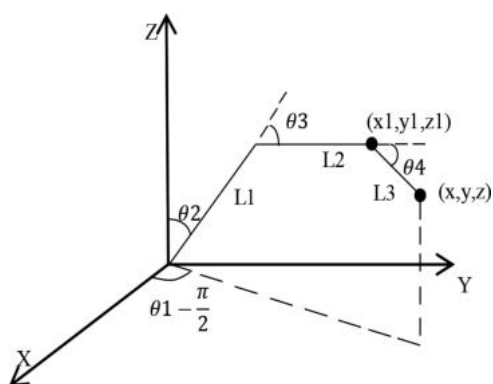


Figura 5. Sistema coordenado y posición final (x, y, z) para la pinza del brazo robótico.

Los ángulos de giro de cada eje del sistema están representados por las letras θ . Los valores $L1$, $L2$ y $L3$ corresponden a las longitudes físicas de los eslabones del brazo robótico. El ángulo $\theta1 - \frac{\pi}{2}$ tiene esta forma debido a la manera en la que se generó el brazo en Matlab, ya que una rotación entre 0 y $\frac{\pi}{2}$ de $\theta1$ posiciona al sistema en el cuarto cuadrante, no en el primero como es de esperarse.

En la Figura 5 se observa también que al proyectar el punto (x, y, z) en el plano XY , se puede encontrar por leyes trigonométricas una ecuación que relaciona dicho punto con el ángulo $\theta1$. Esta relación se presenta a continuación en la ecuación (11).

$$\tan\left(\theta1 - \frac{\pi}{2}\right) = \frac{y}{x}, \text{ entonces} \quad (11)$$

$$\theta1 = \tan^{-1}\left(-\frac{x}{y}\right)$$

En esta ecuación debe tenerse en cuenta el cuadrante en el que se encuentra el punto $(x, y, 0)$, ya que dependiendo de este se deben realizar ciertas operaciones para encontrar el ángulo de forma correcta. Si se encuentra en el tercer o cuarto cuadrante, el valor coincide con el entregado por la función, pero si se encuentra en alguno de los otros cuadrantes, entonces $\theta1 = \theta1 + \frac{|x|}{x}\pi$.

Debe entenderse que el sistema debe moverse de un punto al otro manteniendo la carga en la pinza lo más quieta posible, esto a fin de no generar daños en dicha carga. Esto implica que debe realizarse

la siguiente aproximación en el eslabón final: La inclinación del segmento L3 con respecto al eje Z debe permanecer constante en toda la trayectoria, así los demás segmentos presenten algún movimiento. Solamente se podrá modificar dicha inclinación una vez el movimiento en ejecución haya concluido. Esta condición es de gran utilidad ya que simplifica significativamente el problema de la cinemática inversa.

Por otro lado, si se traza un vector $\vec{k}$ el cual va desde el origen de coordenadas hasta el punto (x_I, y_I, z_I) , se genera un ángulo $\alpha 1$ entre el eslabón L1 y $\vec{k}$, y otro ángulo $\alpha 2$ entre $\vec{k}$ y el plano XY, por lo tanto:

$$\theta 2 = \frac{\pi}{2} - \alpha 1 - \alpha 2 \quad (12)$$

Aplicando la ley del seno, la ley del coseno y las definiciones básicas de la trigonometría a los triángulos generados por la aparición del vector $\vec{k}$ y escribiendo $|\vec{k}| = k$, se llegan a las siguientes ecuaciones:

$$\begin{aligned} \text{sen}(\alpha 2) &= \frac{z_I}{k} \text{ por lo tanto} \\ \alpha 2 &= \text{sen}^{-1}\left(\frac{z_I}{k}\right) \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{\text{sen}(\alpha 1)}{L2} &= \frac{\text{sen}(\pi - \theta 3)}{k} = \frac{\text{sen}(\theta 3)}{k} \\ \text{de donde} \\ \alpha 1 &= \text{sen}^{-1}\left(\frac{L2 \text{sen}(\theta 3)}{k}\right) \end{aligned} \quad (14)$$

$$\begin{aligned} k^2 &= L2^2 + L1^2 - 2L2L1 \cos(\pi - \theta 3) \\ \text{entonces} \\ \theta 3 &= \cos^{-1}\left(\frac{k^2 - L2^2 - L1^2}{2L2L1}\right) \end{aligned} \quad (15)$$

Ahora, reemplazando (13) y (14) en (12), se llega a:

$$\begin{aligned} \theta 2 &= \frac{\pi}{2} - \text{sen}^{-1}\left(\frac{L2 \text{sen}(\theta 3)}{k}\right) \\ &\quad - \text{sen}^{-1}\left(\frac{z_I}{k}\right) \end{aligned} \quad (16)$$

Obsérvese que (15) y (16) son ecuaciones que relacionan a los ángulos $\theta 2$ y $\theta 3$ con el vector $\vec{k}$ y el punto z_I , pero aún no se relacionan con el punto (x, y, z) .

Ahora supóngase que se traza otro vector $\vec{L}$ el cual comienza en el origen de coordenadas y va hasta el punto (x, y, z) , se puede observar, por álgebra vectorial que $\vec{L} = \vec{L3} + \vec{k}$, donde $\vec{L3}$ es un vector que va desde el punto (x_I, y_I, z_I) hasta el punto (x, y, z) , es decir

$$\begin{aligned} \vec{L3} &= xa \hat{i} + ya \hat{j} + za \hat{k} \\ &= (x - x_I)\hat{i} + (y - y_I)\hat{j} + (z - z_I)\hat{k} \end{aligned} \quad (17)$$

Puede verse fácilmente que dicho vector coincide siempre con el eslabón L3, por lo cual su ángulo con respecto a z debe permanecer constante, al igual que su magnitud.

Ahora, se moverá el vector $\vec{L3}$ hasta el origen de coordenadas y allí se le aplicará una transformación a coordenadas esféricas, con el objetivo de dejarlo representado en términos de variables conocidas, adquiriendo la siguiente forma:

$$\begin{aligned} \vec{L3} &= L3 \text{sen}(\varphi) \cos(\rho) \hat{i} \\ &\quad + L3 \text{sen}(\varphi) \text{sen}(\rho) \hat{j} + L3 \cos(\varphi) \hat{k} \end{aligned} \quad (18)$$

En donde φ es el ángulo que forma $\vec{L3}$ con respecto al eje Z. Al ser φ constante durante todo el movimiento, su valor se puede hallar al comienzo de toda la ejecución, puede encontrarse que:

$$\varphi = \theta 3i + \theta 2i + \theta 4i \quad (19)$$

Donde cada valor de estos ángulos corresponde al valor que tienen inicialmente antes de comenzar a ejecutar el movimiento. Por otra parte, ρ es el ángulo que forma la proyección del vector $\vec{L3}$ en el plano XY con el eje X. Puede observarse que ρ está directamente relacionado con $\theta 1$, por lo cual debe reescribirse en términos del cuadrante en el que se encuentra. Para los 4 cuadrantes, si se evalúa antes de hacer la corrección de $\theta 1$, entonces

$$\rho = \theta 1 + \frac{3\pi}{2} \quad (20).$$

Ahora relacionando (17) y (18), se llegan a las siguientes ecuaciones:

$$\begin{aligned} xa &= x - x_I \\ &= L3 \text{sen}(\varphi) \cos\left(\theta 1 + \frac{3\pi}{2}\right) \end{aligned} \quad (21)$$

$$\begin{aligned} ya &= y - y_1 \\ &= L3(\varphi) \sin\left(\theta_1 + \frac{3\pi}{2}\right) \end{aligned} \quad (22)$$

$$\begin{aligned} za &= z - z_1 \\ &= L3 \cos(\varphi) \text{ y por tanto} \\ z_1 &= z - L3 \cos(\varphi) \end{aligned} \quad (23)$$

Entre las cuales se puede resaltar el valor de z_1 .

Para hallar el valor de k y con esto dejar todas las ecuaciones referidas al punto (x, y, z) simplemente se aplica la propiedad vectorial $\vec{L} = \vec{L3} + k$, de aquí se llega a la ecuación (24).

$$k = \sqrt{(x - xa)^2 + (y - ya)^2 + z_1^2} \quad (24)$$

Para poder llevar a cabo la cinemática inversa es necesario construir la trayectoria que deberá seguir el sistema. Como la intención es que la trayectoria seguida por la pinza del brazo robótico sea una línea recta o un círculo en el espacio, no se le debe enviar al sistema solo el punto final si no una serie de puntos consecutivos que se encuentren sobre la curva, razón por lo que una ecuación paramétrica facilita su seguimiento. Se considera entonces, para la línea recta, dos puntos en el espacio, tal como se muestra en la Figura 6:

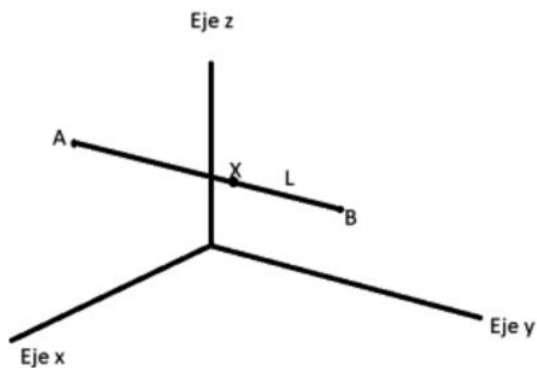


Figura 6. Segmento de recta L en el espacio, el cual tiene la coordenada A como su punto inicial, y la coordenada B como su punto final. X es un punto cualquiera sobre L.

Se definen los vectores $\vec{AB} = \begin{bmatrix} xb - xa \\ yb - ya \\ zb - za \end{bmatrix}$,

$$\vec{AX} = \begin{bmatrix} x - xa \\ y - ya \\ z - za \end{bmatrix} \text{ y el punto } X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Nótese que si el punto X pertenece a la recta, entonces $\vec{AX}$ es múltiplo escalar de $\vec{AB}$. Esto quiere decir que existe un parámetro t tal que $\vec{AX} = t\vec{AB}$. Resolviendo para cada componente del sistema, se puede llegar a las ecuaciones (25), (26) y (27).

$$x = t(xb - xa) + xa \quad (25)$$

$$y = t(yb - ya) + ya \quad (26)$$

$$z = t(zb - za) + za \quad (27)$$

En donde puede observarse que el parámetro t puede tomar valores en el intervalo $0 \leq t \leq 1$.

Para la trayectoria circular se procede de manera similar. Se considera que se desea recorrer una circunferencia de radio $R = 4$, iniciando y finalizando en el punto $(0, 0, 21)$. La proyección de esta curva en el plano XY puede observarse en la Figura 7.

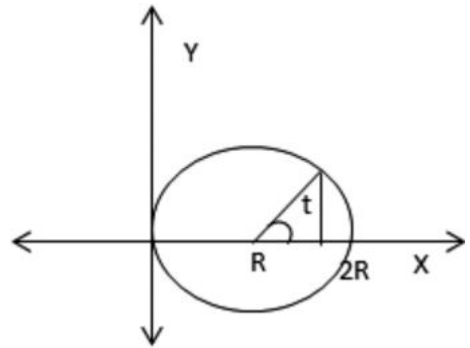


Figura 7. Proyección del círculo de radio R y centro $(R, 0)$ en el plano XY.

A dicho círculo se le puede encontrar una ecuación paramétrica en base a su radio R y el ángulo de giro t . Aplicando la definición del seno y del coseno en el triángulo mostrado en la Figura 7, se llega a las ecuaciones (28), (29) y (30), las cuales corresponden a las ecuaciones paramétricas del círculo en el plano.

$$\begin{aligned} \cos(t) &= \frac{x - R}{R} \text{ de donde} \\ x &= R(1 + \cos(t)) \end{aligned} \quad (28)$$

$$\text{sen}(t) = \frac{y}{R} \text{ por tanto } y = R \text{ sen}(t) \quad (29)$$

$$z = 0 \quad (30)$$

Ahora, se desea rotar el círculo con respecto al eje Y un ángulo de 45° , ya que el punto $(0, 0, 2l)$ es el punto más alto que puede alcanzar el brazo. Para realizar dichas operaciones basta con multiplicar las ecuaciones (28), (29) y (30) por la matriz de la ecuación (3) y luego por la matriz de la ecuación (1). La ecuación del círculo se puede apreciar en la ecuación (31).

$$\begin{pmatrix} x_f \\ y_f \\ z_f \\ 1 \end{pmatrix} = \text{Tras} \begin{pmatrix} 0 \\ 0 \\ 2l \end{pmatrix} * \text{Rot}_y \left(\frac{\pi}{4} \right) * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (31)$$

Donde el resultado de dicha operación genera un vector $(x_f, y_f, z_f, 1)$. Las variables x_f, y_f, z_f corresponden a los puntos de la circunferencia trasladada y rotada.

Implementación del algoritmo de cinemática

El algoritmo para la implementación de la cinemática inversa sigue el diagrama de lógica mostrado en la Figura 8.

En la cual pueden observarse varias etapas: Una etapa de recibir puntos, esta es la encargada de esperar a que el punto final (x, y, z) sea ingresado y permanece allí hasta que el botón de enviar sea presionado. Luego, cuando el botón de enviar es presionado, viene la etapa de inicializar banderas, en esta se realizan todas las inicializaciones de variables que pueden ser de gran importancia como el ángulo φ , una variable *contador* la cual se va a encargar de contar la cantidad de puntos sobre la recta o círculo enviados y la cantidad de puntos m que van a estar sobre la recta o círculo, entre otras variables.

Si el valor del contador es inferior a m entonces se pasa a la etapa calcular punto siguiente. En la etapa calcular punto siguiente se encuentran las ecuaciones paramétricas de la recta en el espacio o la ecuación del círculo trasladado y con éstas se

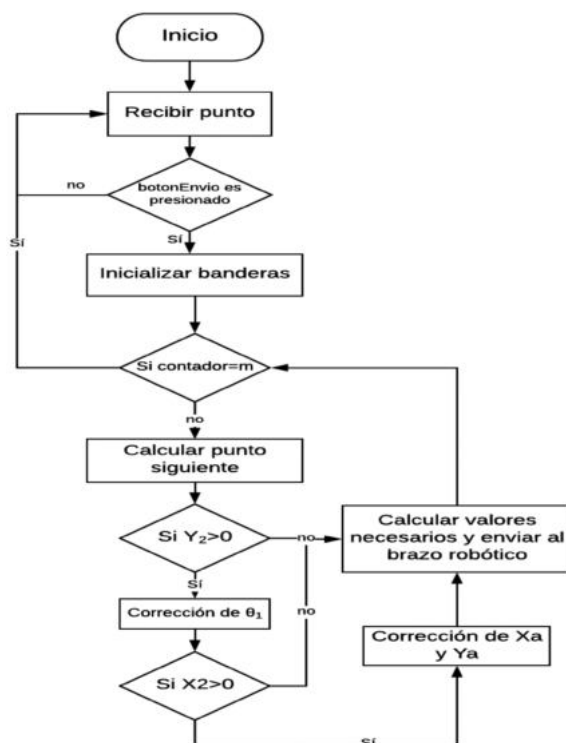


Figura 8. Diagrama de flujo que ilustra la manera como fue implementada la cinemática inversa del brazo robótico Pegasus II.

calcula el punto (x, y, z) siguiente sobre la curva dependiendo del valor de m y *contador*. También se calculan los valores de x_a , y_a , z_a y θ_1 .

Si el punto final se encuentra en el primer o segundo cuadrante se pasa al estado corrección de θ_1 , en donde se realiza la corrección indicada en la subsección anterior.

Luego, si la condición anterior es correcta y el punto final se encuentra en el primer cuadrante se pasa al estado corrección de x_a y y_a donde se aplican correcciones a dichas variables debido a la dependencia de éstas con θ_1 .

Más adelante, se encuentra el estado calcular valores necesarios y enviar al brazo robótico, en donde se calculan los valores de k , θ_2 , θ_3 y θ_4 . Se envían todos los ángulos al brazo robótico, se actualiza el valor de la posición inicial, la interfaz gráfica y también se incrementa *contador*. El estado siguiente está definido por la condición *si contador = m*.

RESULTADOS Y ANÁLISIS

En la Figura 9 (lado izquierdo) puede observarse la instrumentación del brazo robótico Pegasus II en cuatro puntos diferentes de un recorrido rectilíneo en el espacio tridimensional, el cual parte del punto $(0, 0, 21)$ y llega hasta el punto $(0.1, 5, 17)$. La línea delgada ilustra la trayectoria que la pinza del brazo debe seguir. Puede observarse que el brazo recorre la recta correctamente manteniendo la inclinación del extremo final con respecto al eje Z, tal y como se ha indicado en este artículo.

En la Figura 9 (Lado derecho) puede observarse cuatro puntos de un recorrido a través de una curva circular. Dicha trayectoria se realizó para comprobar el correcto funcionamiento de la cinemática inversa ante una curva diferente a la recta. Puede observarse que el brazo recorre la circunferencia correctamente manteniendo la inclinación del extremo final con respecto al eje Z, tal y como era de esperarse.

Por otro lado, en la Figura 10 pueden observarse los puntos de inicio y de finalización del recorrido del brazo, ya que la perspectiva dada por Matlab en la Figura 9 no permite ver con claridad los puntos finales del recorrido, debido a que desde el punto de vista utilizado pareciera que dichos puntos no coincidieran.

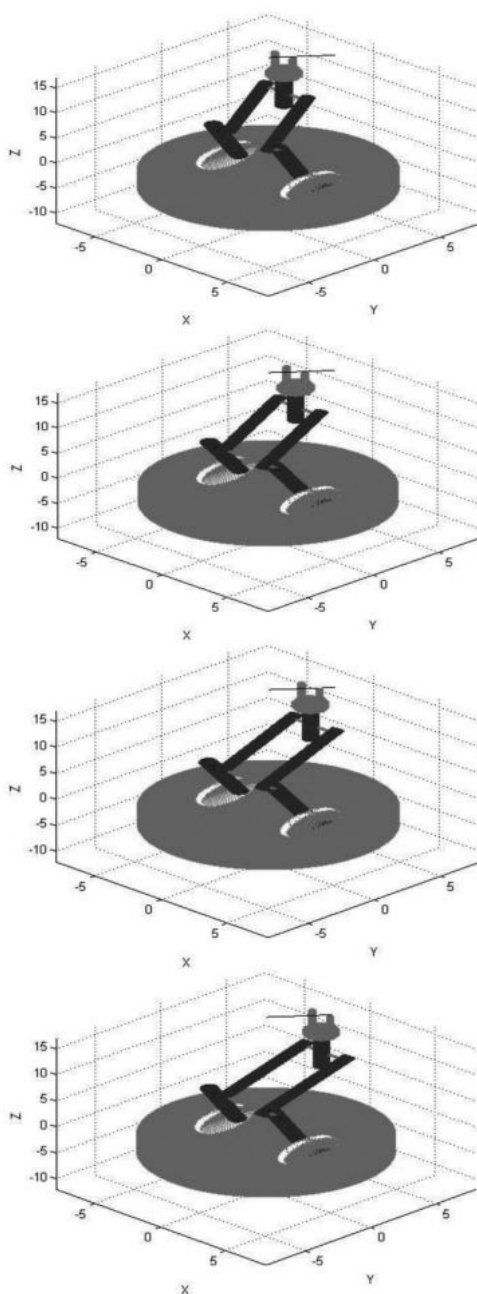


Figura 9. Instrumentación del brazo robótico en 4 puntos diferentes del recorrido de una recta (Lado Izquierdo) y en 4 puntos diferentes de un recorrido circular inclinado (Lado Derecho).

En la Figura 11 puede observarse cuatro puntos de un recorrido a través de una curva circular. Dicha trayectoria se realizó para comprobar el correcto

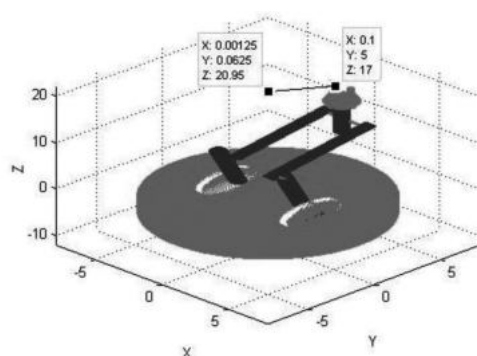


Figura 10. Simulación del brazo robótico en su punto final, se pueden apreciar los extremos de los segmentos de recta a través de la cual el brazo se mueve.

funcionamiento de la cinemática inversa ante una curva diferente a la recta. Puede observarse que el brazo recorre la circunferencia correctamente manteniendo la inclinación del extremo final con respecto al eje Z, tal y como era de esperarse.

Al ejecutar la simulación con el brazo robótico Pegasus II, éste siguió las trayectorias de una manera aproximada, ya que solamente se realizó control cinemático de posición mas no de velocidad, por lo cual las velocidades de los ejes no eran las precisas para seguir la curva de una manera completamente perfecta. Se observó que el brazo real en algunos puntos presentaba unos huecos parecidos a una escalera debido a la falta de sincronismo entre las velocidades de los motores.

Como trabajo a futuro se puede implementar un control cinemático sobre las velocidades de cada eje con el objetivo de que siga de una manera más precisa cada curva predefinida.

CONCLUSIONES

En el presente artículo, se logra demostrar que para un brazo robótico como el Pegasus II es posible definir una cinemática inversa de manera analítica, con una buena precisión a la hora de recorrer curvas, tomando ciertas consideraciones para su movimiento, evitando así las soluciones numéricas.

En general, para un brazo robótico la solución de la cinemática inversa no siempre puede encontrarse de manera única, para el caso del brazo robótico Pegasus

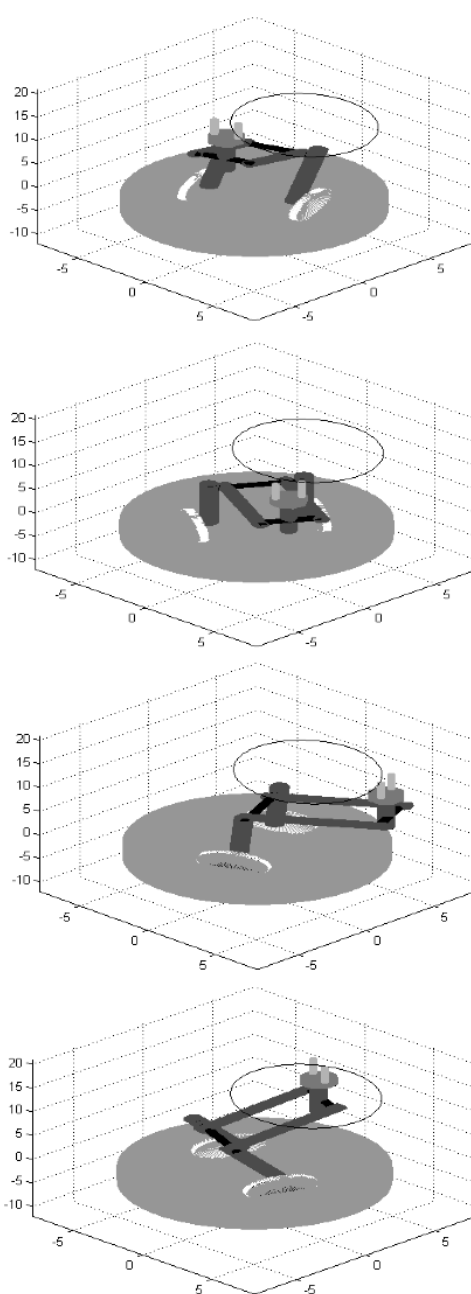


Figura 11. Instrumentación del brazo robótico en 4 puntos diferentes de un recorrido circular inclinado.

II se encontró una posible solución, suponiendo que el ángulo con respecto a Z del eslabón que contiene a la pinza debe permanecer constante, simplificando así el problema, no obstante, despreciando otras posibles soluciones que también pueden ser válidas dependiendo del trabajo que vaya a realizar.

De esta manera se concluye que encontrar la cinemática inversa de un sistema robótico es la base para lograr que se puedan seguir diversos tipos de curvas, lo cual permite que el sistema robótico se pueda utilizar para realizar tareas de una alta complejidad. Así, el trabajo presentado en este artículo le otorga al brazo robótico Pegasus II la libertad necesaria para poder realizar otro tipo de tareas para las cuales no estaba habilitado.

AGRADECIMIENTOS

Este trabajo es financiado por la Universidad de Antioquia y el Fondo de Apoyo Trabajos de Grado de la Facultad de Ingeniería de dicha Universidad.

REFERENCIAS

- [1] F. Torres, J. Pomares, P. Gil, S. Puente and R. Aracil. "Robots y Sistemas Sensoriales". Pearson Educación. Segunda edición, pp. 81-122. Madrid, España. 480 pag. ISBN: 84-205-3574-5. Año 2002.
- [2] Amatrol. "About Us-Amatrol-Learning Systems for Technical Education". Julio 2, 2018. URL: <https://amatrol.com/about-us/>
- [3] L. Giraldo, E. Delgado y G. Castellanos. "Cinemática Inversa de un Brazo Robot Utilizando Algoritmos Genéticos". Revista Avances en Sistemas e Informática. Vol. 3 N° 1, pp. 29-34, Junio 2006. ISSN: 1657-7663. URL: <http://www.redalyc.org/html/1331/133114991005/>
- [4] I. Zaplana, J. Claret y L. Basanez. "Análisis cinemático de robots manipuladores redundantes: Aplicación a los robots Kuka LWR 4+ y ABB Yumi". Revista Iberoamericana de automática e informática industrial. Vol. 15 N° 2, pp. 192-202, Marzo 2018. ISSN: 1697-7912. URL: <https://polipapers.upv.es/index.php/RIAI/article/view/8822>
- [5] A. Botero. "Cinemática de manipuladores robóticos", pp. 12-23. Junio 20, 2005. Julio 2, 2018. URL: http://www.wag.caltech.edu/home/ajaramil/libro_robotica/cinematica.pdf
- [6] Amatrol. "Pegasus II Robotic Programming and Training System". Julio 2, 2018. URL: <https://amatrol.com/coursepage/servo-robotic-training-and-programming-system/>
- [7] Farnell. "Arduino Due", pp. 2-6. Junio 5, 2018. [Online]. URL: <http://www.farnell.com/datasheets/1682211.pdf>
- [8] J. López y J. Hueso. "Introducción a Matlab", pp. 5-6. 2015. Junio 5, 2018. URL: <http://www.upibi.net/pdf/tutoriales/MATLAB-PERDIENDO-EL-MIEDO.pdf>