ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA MECÁNICA

ANÁLISIS CINEMÁTICO Y SIMULACIÓN DEL ROBOT PARALELO DE TIPOLOGÍA 3UPE-RPU

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO MECÁNICO

WILLIAM ROLANDO MORALES PALMA
william.morales@epn.edu.ec

DIRECTOR: Ing. Iván Zambrano, M.Sc.
ivan.zambrano@epn.edu.ec

Quito, marzo de 2017

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por William Rolando Morales Palma, bajo mi supervisión.

Ing. Iván Zambrano, M.Sc.

DIRECTOR DE PROYECTO

DECLARACIÓN

Yo, **William Rolando Morales Palma**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

William Morales

ÍNDICE

Introducción 1
Pregunta de investigación 1
Objetivo general 1
Objetivos específicos 2
Alcance 2
1. Marco teórico 3
1.1. Antecedentes 3
1.2. Robot y robot industrial 3
1.3. Cinemática del robot 4
Ángulos y matrices de rotación de Euler 6
Método Denavit & Hartenberg 8
Cinemática directa e inversa 8
2. Análisis cinemático 10
2.1. Tipología del robot 10
2.2. Cadena cinemática del robot 11
Análisis cinemático de la plataforma 11
Análisis cinemático de los brazos UPE 18
Análisis cinemático del brazo RPU 28
2.3. Vinculación de modelos matemáticos del robot 34
3. Cálculo y simulación de resultados 36
3.1. MatLab® 36
3.2. Estructura del programa 36
Variables de entrada 37
Interfaz gráfica 38
Algoritmo de cálculo 40
3.3. Ejemplo de cálculo 43
Ejemplo de cálculo para el caso en reposo 43

Ejemplo de cálculo para el caso en movimiento 45
4. Análisis de resultados, conclusiones y recomendaciones 48
4.1. Análisis de resultados 48
4.2. Conclusiones 48
4.3. Recomendaciones 49
Bibliografía.....50
Anexos 51

RESUMEN

En este estudio se presenta una propuesta para el análisis cinemático del robot paralelo de tipología 3UPE-RPU, que se encuentra en desarrollo como parte del proyecto PIMI-1504 llevado a cabo en la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional (EPN). Este proyecto tiene como objetivo el diseño y construcción de un robot que sea capaz de asistir con los tratamientos de rehabilitación del miembro inferior. El robot tiene una movilidad de 4 grados de libertad, dos movimientos de traslación y dos de rotación, estos grados de libertad son considerados en base a los movimientos de flexo-extensión y rotación interna y externa del miembro inferior. En el análisis cinemático, las ecuaciones desarrolladas se caracterizan por estudiar la cadena cinemática de cada elemento del robot (brazos y plataforma) de forma separada, a través de los métodos de Denavit & Hartenberg y los ángulos de rotación de Euler respectivamente, describiendo las posiciones que adoptan las articulaciones de cada brazo por separado, para luego establecer las relaciones entre cada uno de ellos. A partir de las ecuaciones de posición encontradas, se derivan las expresiones que describen la velocidad y aceleración de cada elemento del robot. Para la validación de resultados se presenta un programa capaz de simular los movimientos del robot y mostrar la relación entre las posiciones de las articulaciones de los brazos del robot y la posición de la plataforma en el espacio; los resultados hallados a través de este programa muestran que existe una diferencia despreciable entre las posiciones de los vértices de la plataforma y los extremos de los brazos del robot, al realizar cualquier movimiento.

Palabras clave: Cinemática, PIMI15-04, Robótica, Simulación

ABSTRACT

This study presents a proposal for the kinematic analysis of the parallel robot of type 3UPE-RPU, which is under development as part of the PIMI-1504 project carried out in the Faculty of Mechanical Engineering of the National Polytechnic School (EPN). This project aims to design and build a robot that is able to assist with the rehabilitation treatments of the lower limb. The robot has a mobility of 4 degrees of freedom, two movements of translation and two of rotation, these degrees of freedom are considered based on the flexo-extension movements and internal and external rotation of the lower limb. In the kinematic analysis, the developed equations are characterized by studying the kinematic chain of each element of the robot (arms and platform) separately, through the Denavit & Hartenberg methods and the Euler rotation angles respectively, describing the positions which adopt the joints of each arm separately, and then establish the relationships between each of them. The expressions describing the velocity and acceleration of each element of the robot are derived from the position equations founded. For the validation of results a program is presented which is able to simulate the movements of the robot and show the relation between the positions of the articulations of the arms of the robot and the position of the platform in the space; the results founded through this program show that there is a negligible difference between the positions of the vertices of the platform and the ends of the arms of the robot, when making any movement.

Keywords: Kinematics, PIMI15-04, Robotics, Simulation

ANÁLISIS CINEMÁTICO Y SIMULACIÓN DEL ROBOT PARALELO DE TIPOLOGÍA 3UPE-RPU

INTRODUCCIÓN

El proyecto PIMI 15-04 llevado a cabo en la Facultad de Ingeniería Mecánica de la EPN, tiene por objetivo el desarrollo de un sistema que permita rehabilitar el miembro inferior del cuerpo humano, a través del empleo de un robot paralelo de tipología 3UPE-RPU, hasta el momento, único en el mundo.

Como avance para el proyecto, se realizó anteriormente un estudio de la cinemática del robot, del cual se pudo obtener un primer modelo matemático capaz de describir las posiciones, pero con algunos problemas de convergencia de resultados para ciertos valores de entrada y cargas de cálculo muy elevadas al efectuar la resolución del sistema de ecuaciones en software matemático, provocando una demora considerable para obtener una respuesta al sistema. Pero las necesidades para avanzar con un correcto estudio de la dinámica implicada en el robot y el posterior control del mismo, dependen también del desarrollo de un modelo matemático capaz de describir las velocidades y aceleraciones asociadas a los movimientos del robot.

En el presente estudio desarrolla un modelo capaz de describir las velocidades y aceleraciones de las articulaciones del robot y se simulará para comprobar gráficamente que el comportamiento del robot sea acorde a las necesidades del proyecto.

Para la realización del análisis cinemático del robot, se propone un análisis por separado de cada una de las cadenas cinemáticas que componen el robot empleando el método desarrollado por Denavit & Hartenberg y las matrices de rotación de Euler.

Pregunta de investigación

¿Se puede conseguir un modelo matemático para el robot de tipología 3UPE-RPU, capaz de describir su movimiento, velocidades y aceleraciones; además de poder ser simulado a través de un software matemático?

Objetivo general

Analizar y simular la cinemática del robot paralelo de tipología 3UPE-RPU

Objetivos específicos

Determinar las matrices de los brazos del robot paralelo a través del método de Denavit & Hartenberg

Determinar las ecuaciones del movimiento de la plataforma a través de las matrices de rotación de Euler.

Determinar el modelo matemático que describa las posiciones, velocidades y aceleraciones asociadas al movimiento del robot.

Simular el comportamiento del robot obtenido como resultado del modelado matemático del mismo, y comprobar gráficamente la coherencia de los resultados.

Alcance

El presente proyecto se limita al análisis de la cinemática presente en el robot paralelo 3UPE-RPU, estableciendo el modelo matemático del movimiento de sus componentes, para luego estudiar las interacciones entre cada uno de ellos. Además, se centra en la creación de un programa capaz de simular el modelo hallado, permitiendo validar los resultados de las ecuaciones establecidas en el análisis cinemático previo.

El programa debe ser capaz de simular un esquema del robot en movimiento y mostrar las relaciones entre las articulaciones del robot y el movimiento de la plataforma del mismo.

1. MARCO TEÓRICO

1.1. Antecedentes

La investigación previa, presentada en el año 2016 titulada “Análisis cinemático directo e inverso del robot paralelo”, realizada por las ingenieras Fernández Yánez, Lorena A. y Sotomayor Reinoso Luisa F., sentaron las bases para el desarrollo de la cinemática del robot paralelo de tipología 3UPE-RPU.

En su estudio se presenta el desarrollo de las cadenas cinemáticas del robot paralelo como un lazo cerrado, en el cual se estudia la convergencia de las posiciones de los vértices de su plataforma móvil a través de las ecuaciones generadas por los brazos UPE e igualadas a las ecuaciones del brazo RPU (brazo central del robot), y la plataforma móvil.

A través de la aplicación de la cinemática inversa, en primer lugar, se obtuvo un sistema de 11 ecuaciones para el cálculo de la posición que toman las articulaciones de cada brazo, respecto a una posición dada de la plataforma móvil del robot. Estos resultados fueron validados matemáticamente a través de la cinemática directa y también por medio una simulación en un ambiente virtual.

De forma complementaria al trabajo, se determinaron las ecuaciones de velocidad y aceleración del robot, a través de la derivación de las ecuaciones de posición previamente encontradas, pero éstas no se llegaron a validar.

1.2. Robot y robot industrial

Históricamente el término robot describe la imagen de una máquina cuya existencia es atribuida a la necesidad de la realización de trabajos productivos, capaces de imitar movimientos y comportamientos de seres vivos.

Es cierto que los avances tecnológicos del último siglo han mejorado la concepción que se tiene en cuanto al término robot, ya que se observa un cambio desde las simples máquinas que en un inicio sólo amplificaban la capacidad muscular del hombre, a máquinas que son también son capaces de procesar información, complementando y hasta sustituyendo al hombre en ciertas tareas.

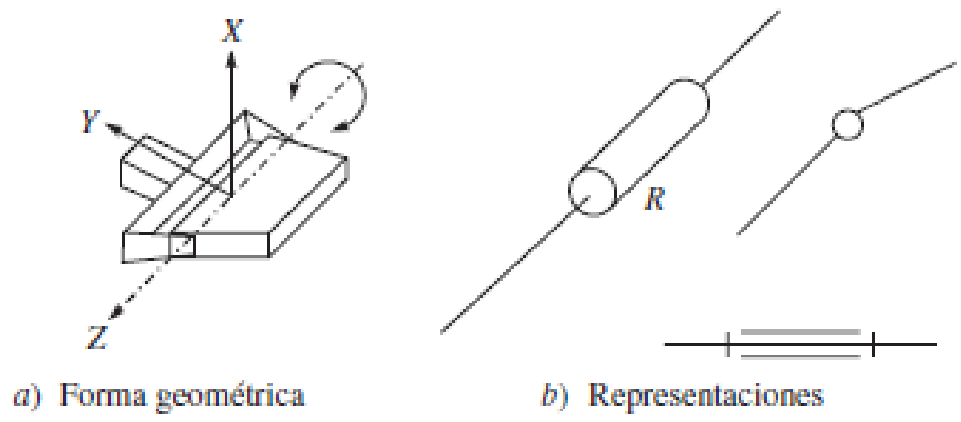
De este punto nace el término de robot industrial, una máquina que implementa tecnologías de control automático, operación remota de manipuladores y computación en tiempo real.

El control automático concibe la idea de un sistema capaz de gobernar un proceso sin la necesidad de la participación de agentes externos; estos sistemas trabajan comparando una señal de respuesta de un proceso con una señal de referencia ya establecida.

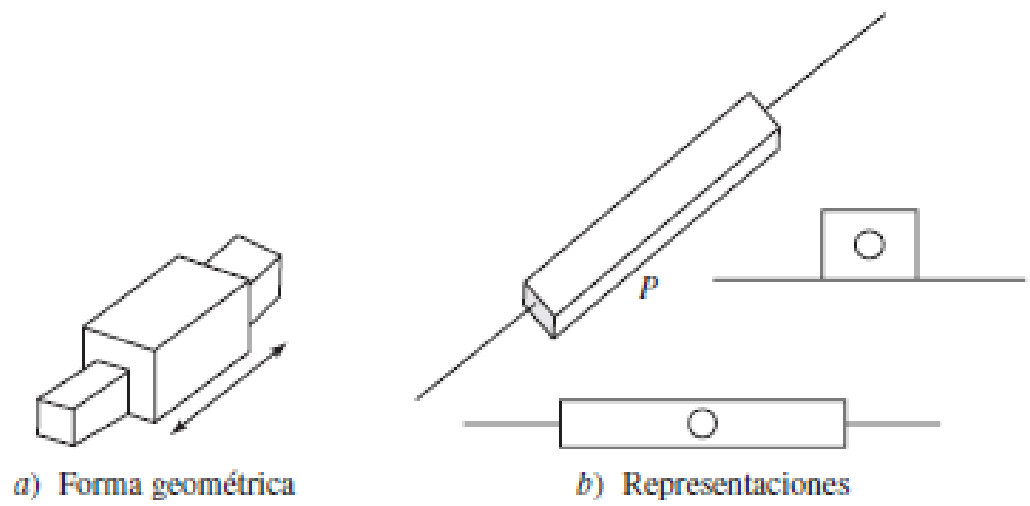
La operación remota de manipuladores ha empleado diversas tecnologías a lo largo de la historia, tales como la neumática e hidráulica, y en años recientes la eléctrica. Esta última tecnología, junto al desarrollo de los microprocesadores, son las que encaminan de forma decisiva al control por computador a convertirse en una poderosa herramienta en el control de robots. (Ollero, 2001)

1.3. Cinemática del robot

Un robot se encuentra conformado por cuerpos individuales llamados “eslabones”, estos eslabones se los considera cuerpos rígidos e interactúan entre sí a través de pares o articulaciones cinemáticas, las mismas que pueden ser de tipo prismática, rotacional, helicoidal, esférica, cilíndrica o planar.

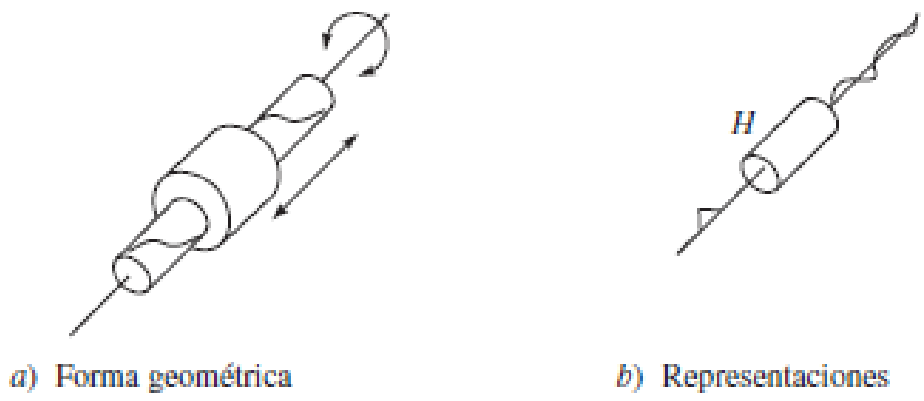


i) Articulación rotacional

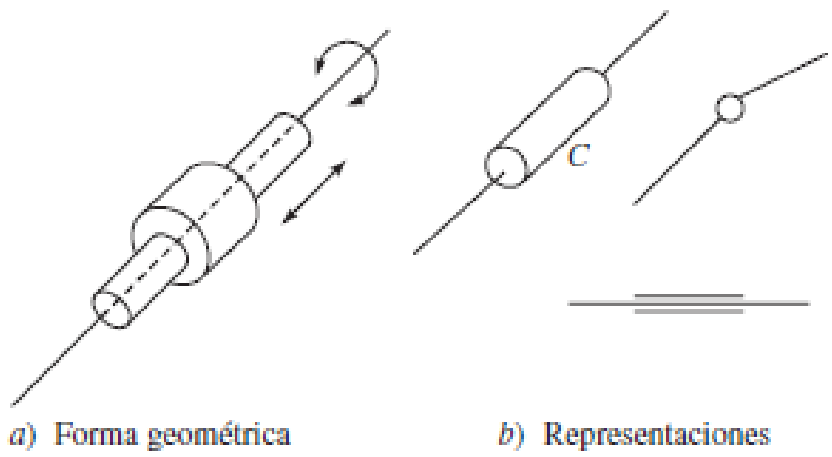


ii) Articulación prismática

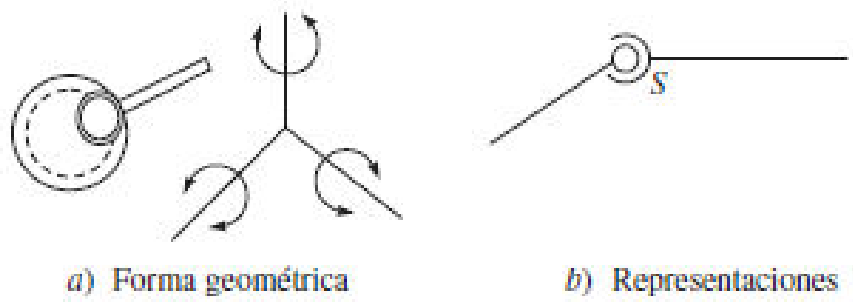
Figura 1. Tipos de articulaciones: rotacional (i) y prismática (ii). (Saha, 2010)



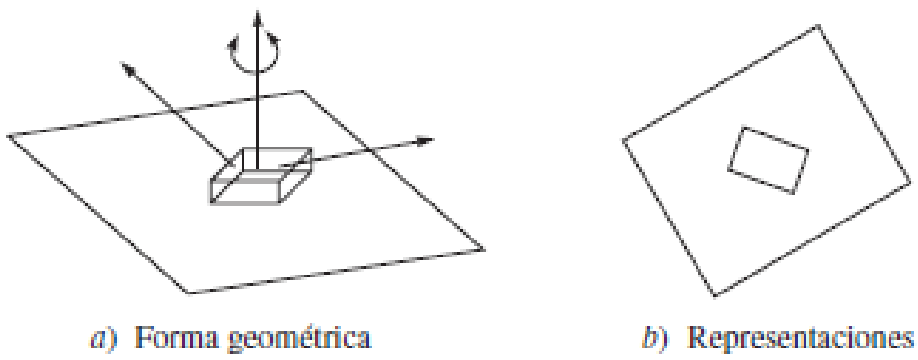
iii) Articulación helicoidal



iv) Articulación cilíndrica



v) Articulación esférica

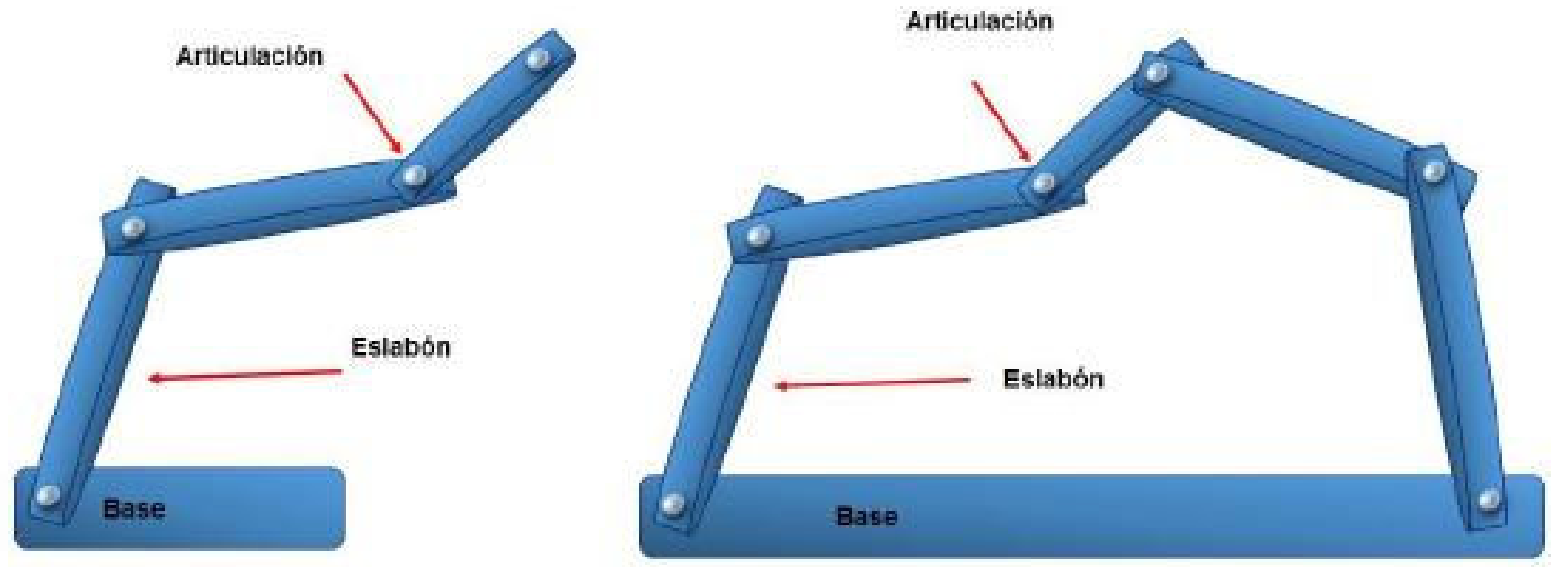


vi) Articulación planar

Figura 2. (continuación) Tipos de articulaciones: helicoidal (iii), cilíndrica (iv), esférica (v) y planar (vi). (Saha, 2010)

Al tener una serie de eslabones conectados se genera una “cadena cinemática”, la cual, dependiendo del número de elementos conectados a cada eslabón, puede ser considerada

como cadena cinemática “abierta”, si todos los eslabones se encuentran conectados a otros dos, exceptuando el eslabón del extremo de la cadena, el cual está conectado sólo a uno. O también considerarse cadena cinemática cerrada, si todos los eslabones se encuentran conectados, de tal forma que el eslabón final de la cadena se conecta con el eslabón inicial, bastidor o base de la cadena.



a) Cadena cinemática abierta b) Cadena cinemática cerrada

Figura 3. Cadenas cinemáticas: abierta (a) y cerrada (b) (Fernández & Sotomayor, 2016)

Para hallar las relaciones matemáticas capaces de describir la posición de cualquier punto perteneciente al eslabón o cuerpo rígido en estudio, se emplean vectores de posición (1.1), los cuales de forma abreviada, definen la posición de un punto en el espacio.(Saha, 2010)

$$[P] = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{1.1}$$

Pero con expresar la posición de un punto en el espacio, el cuerpo al que pertenece no queda completamente definido, siendo imprescindible también la necesidad de definir la orientación del cuerpo en el espacio. Para esto existen métodos como el de los cosenos directores y los ángulos de Euler, los cuales, si bien en esencia cumplen la misma función, cada uno es usado de acuerdo a las necesidades del sistema.

Ángulos y matrices de rotación de Euler

Los ángulos de Euler son un método empleado para definir la orientación de un cuerpo en el espacio, respecto a un sistema de coordenadas dado.

Los ángulos de rotación de Euler son descritos a través de 3 rotaciones mayores, rotación alrededor del eje X, rotación alrededor del eje Y, y rotación alrededor del eje Z, de esta manera, se puede afirmar que cualquier orientación de un cuerpo puede ser descrita a

través de la combinación de estos ángulos. (The MathWorks Inc., n.d.)

Matemáticamente, cada una de las rotaciones de Euler se describen de la siguiente forma:

$$Rot\ X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

Ecuación 1. Matriz de rotación alrededor del eje X (Barrientos, Peñin, Balaguer, & Aracil, 1996)

$$Rot\ Y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

Ecuación 2. Matriz de rotación alrededor del eje Y (Barrientos et al., 1996)

$$Rot\ Z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ecuación 3. Matriz de rotación alrededor del eje Z (Barrientos et al., 1996)

Al realizar las rotaciones sucesivas con los ángulos de rotación de Euler, los movimientos que se realizan al sistema deben determinar correctamente respecto a cuál sistema de coordenadas se lo realiza, es decir, si son rotaciones extrínsecas indican que todas las rotaciones se realizan respecto al sistema de coordenadas inicial; o intrínsecas, lo cual indica que las rotaciones se realizan respecto a los ejes de coordenadas rotados, es decir los nuevos ejes de coordenadas generados. En la siguiente figura se puede apreciar un ejemplo de rotación intrínseca.

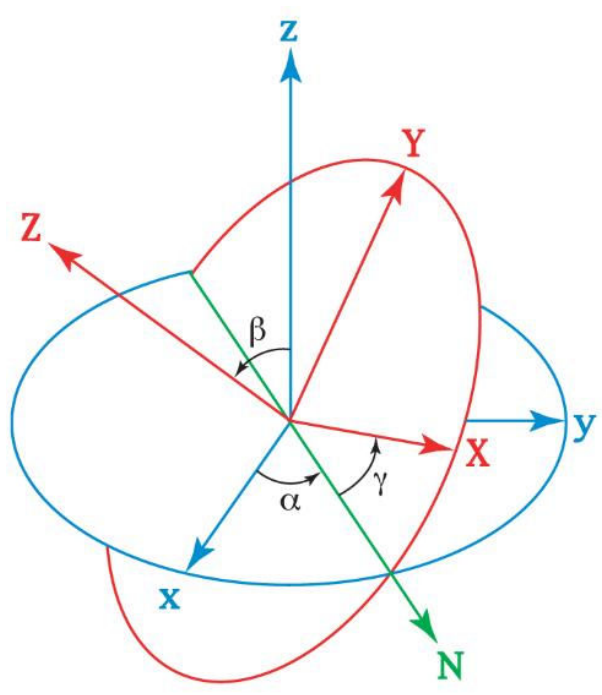


Figura 4. Rotaciones intrínsecas empleando ángulos de Euler (Brits, 2008)

Donde, el sistema de coordenadas inicial (x, y, z), luego de rotar un ángulo α alrededor de z, rota un ángulo β alrededor del nuevo eje N, hasta la posición de Z; en este punto nuevamente se rota un ángulo γ alrededor del nuevo eje Z, hasta la posición final descrita por el sistema (X, Y, Z)

Método Denavit & Hartenberg

Al estudiar la cinemática de un robot, se encuentra con sistemas cada vez más complejos de analizar, para esto Denavit & Hartenberg (D&H) en el año de 1955 proponen el empleo de cuatro parámetros asociados a sistemas de coordenadas móviles, para la descripción del movimiento generado por cualquier cadena cinemática abierta. Estos parámetros corresponden a un desplazamiento (d) a lo largo del eje z, una rotación (θ) alrededor de este mismo eje, un segundo desplazamiento (a) alrededor del eje x y finalmente, una segunda rotación (α) alrededor de este segundo eje.

Existe gran cantidad de literatura que detalla el proceso a seguir para la implementación del método y la evaluación de los parámetros de D&H, por lo que no se abordará este punto en el presente trabajo.

Al emplear el método D&H, las matrices resultantes se conocen como matrices homogéneas y presentan la siguiente estructura:

$$T = \begin{bmatrix} R_{3\times3} & P_{3\times1} \\ f_{1\times3} & w_{1\times1} \end{bmatrix}$$

Ecuación 4. Matriz de transformación homogénea (Barrientos et al., 1996)

Donde, $R_{3\times3}$ corresponde a la matriz de rotación, $P_{3\times1}$ corresponde al vector posición, $f_{1\times3}$ define el vector de perspectiva o distorsión y $w_{1\times1}$ describe el factor de escala.

A través de cada matriz de transformación (T) se describe la posición de un eslabón en el espacio. Para obtener la matriz de transformación de todo el sistema, es necesaria la multiplicación sucesiva y ordenada de cada una de estas matrices. La matriz resultante también obedecerá a la forma de T. (Barrientos et al., 1996)

Cinemática directa e inversa

Al analizar la cinemática de un robot, se encuentra con una problemática al momento de evaluar los parámetros que intervienen en el sistema, ya que, en la mayoría de los casos

se conoce una posición en el espacio a la cual se debe alcanzar, pero no se conoce la posición que las articulaciones del robot deben tomar para lograr que el eslabón final alcance dicha posición.

Por otro lado, se tiene el caso inverso, en el que se encuentran determinadas las posiciones de las articulaciones del robot, pero no se conoce la posición del eslabón final.

Para el estudio de estos casos, existen los análisis cinemáticos directo e inverso, en el primero, se analiza la posición del eslabón final del robot, dadas las posiciones de las articulaciones y la geometría de los eslabones que lo componen. En el segundo caso, se da lo contrario, se calcula las posiciones de las articulaciones del robot, conocida la posición del eslabón final y la geometría de los eslabones.

Es necesario un análisis de los datos presentes en el problema, para determinar cuál será el método correcto a emplearse.

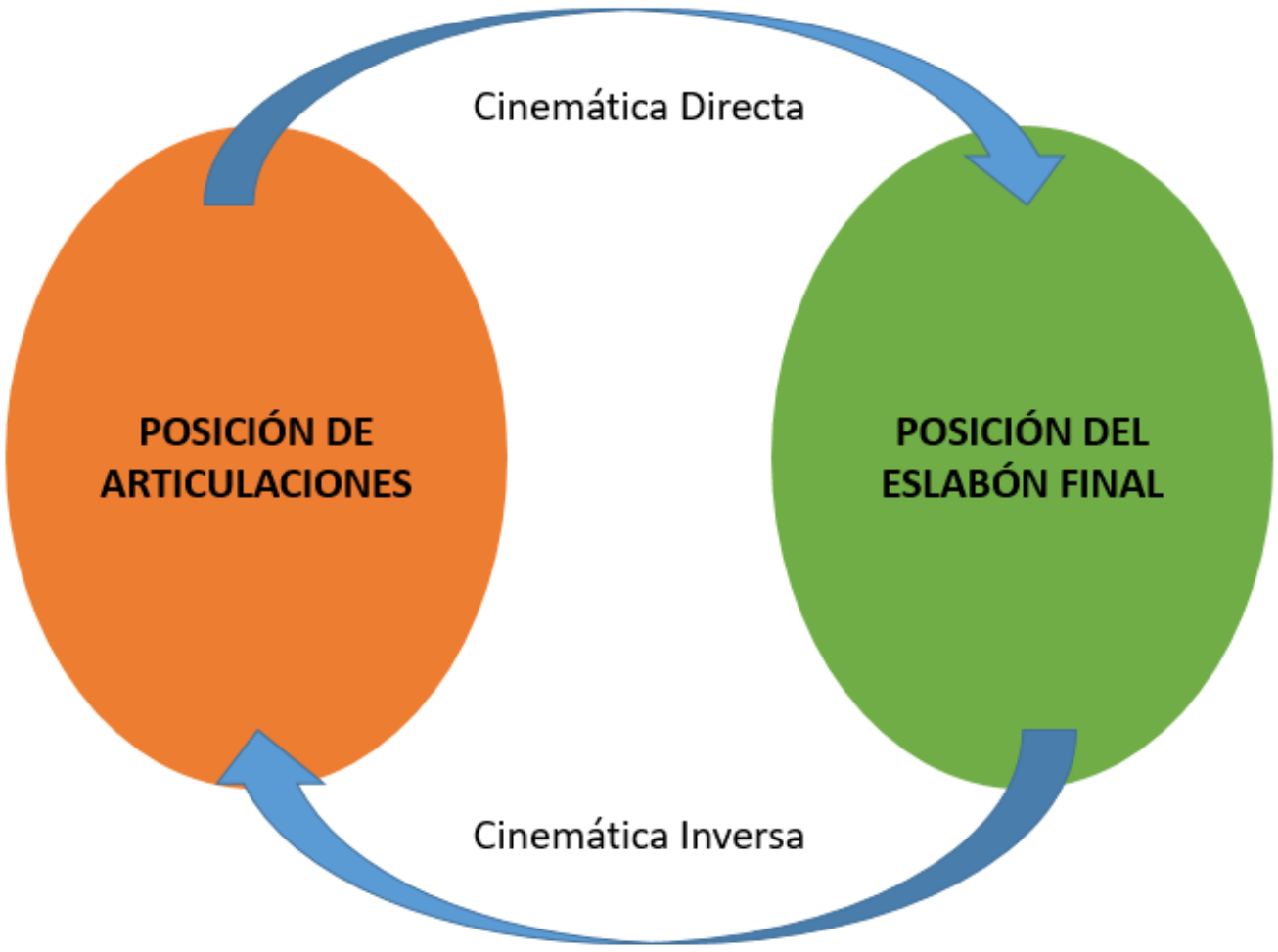


Figura 5. Relación entre cinemática directa e inversa

2. ANÁLISIS CINEMÁTICO

2.1. Tipología del robot

El presente robot corresponde a un modelo paralelo, consta de 4 brazos conectados por sus extremos inferiores a una plataforma base estacionaria de forma triangular y por sus extremos superiores a una plataforma móvil de menor área y geometría similar a la base. Figura 6

Los brazos están divididos en 2 tipos de acuerdo a las articulaciones presentes en cada uno: 3 brazos UPE y un brazo RPU.

Los brazos UPE, cumplen la configuración típica de un brazo polar, constan de una articulación universal en su base, seguida de una articulación prismática y una esférica en su extremo superior, estos brazos se ubican entre los vértices de la base y la plataforma móvil.

El brazo RPU se levanta desde el circuncentro de la base hasta el circuncentro de la plataforma móvil en su extremo superior, se conforma de una articulación rotacional en su base, una articulación prismática y una articulación universal, ubicada justo por debajo de la plataforma.

La plataforma móvil del robot describe 4 grados de libertad al moverse: dos rotacionales y 2 de traslación (Fernández & Sotomayor, 2016)

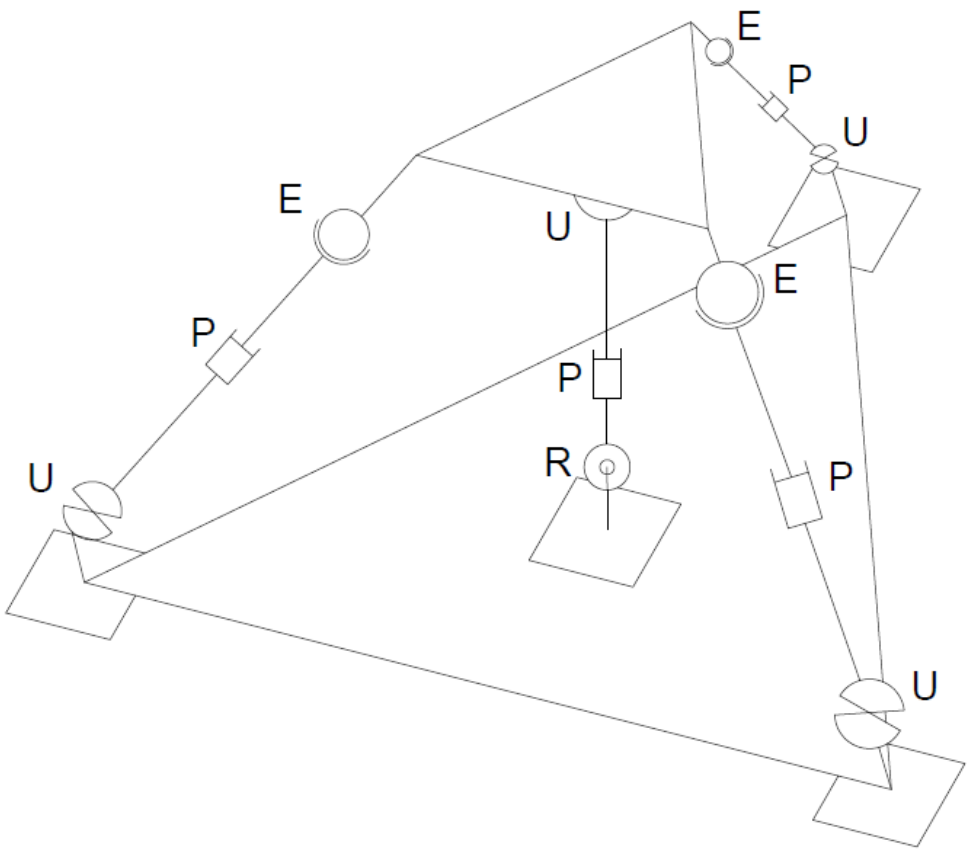


Figura 6. Vista isométrica del robot paralelo 3UPE-RPU

2.2. Cadena cinemática del robot

Para iniciar el análisis de movimiento del robot, se establece el origen del sistema de coordenadas absoluto $(0, X, Y, Z)$ en el circuncentro de la plataforma base.

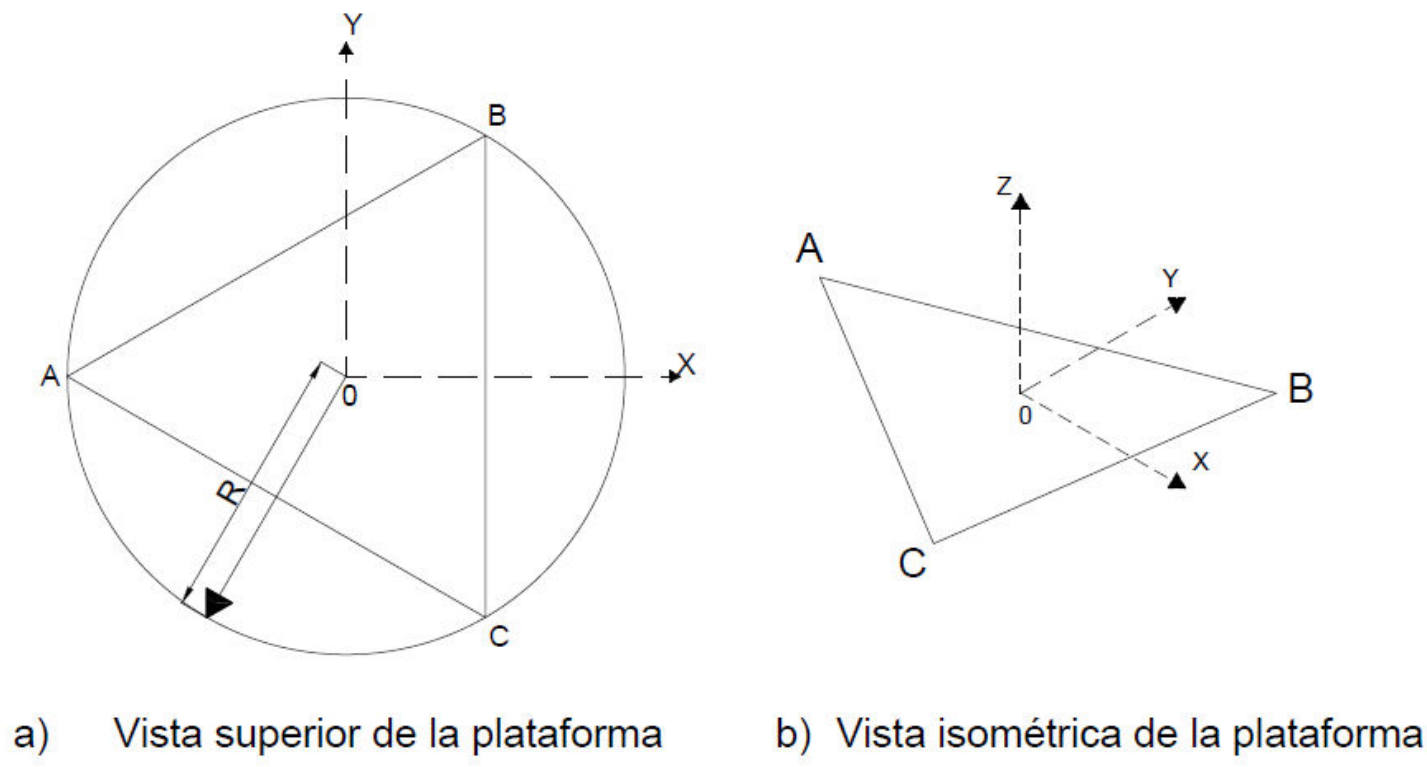


Figura 7. Sistema de coordenadas en la base del robot: vista superior (a), vista isométrica (b)

A partir de este sistema de coordenadas es necesario analizar los movimientos que cada elemento del robot va a ser capaz de realizar. En primer lugar, la plataforma describe movimientos de traslación sobre los ejes X y Z de la base, y rotación alrededor de los ejes Y y Z de sí misma, estos movimientos corresponden a los 4 grados de libertad del robot.

Los brazos UPE, presentan movimientos de forma polar por medio de su articulación universal (U) en la base y su articulación prismática (P) que permite la variación de la longitud total del brazo, la articulación esférica (E) acopla cada brazo a la plataforma por lo cual no se la considera dentro del análisis.

Finalmente, el brazo RPU, describe un movimiento de rotación en su primera articulación (R) alineada con el eje Y de la base, la articulación prismática (P) permite la variación de la longitud total del brazo y la articulación universal (U) permite acoplar este brazo a la plataforma móvil. Todas las articulaciones de este brazo son consideradas en el análisis.

Análisis cinemático de la plataforma

Para analizar el movimiento de la plataforma móvil del robot, se establece un sistema de coordenadas ubicado en el punto donde el brazo central se une a ésta y se lo define como:

$0_m, X_m, Y_m, Z_m$. Al ubicar el sistema de coordenadas en esta posición, también se garantiza la facilidad de cálculo de las ecuaciones a desarrollarse.

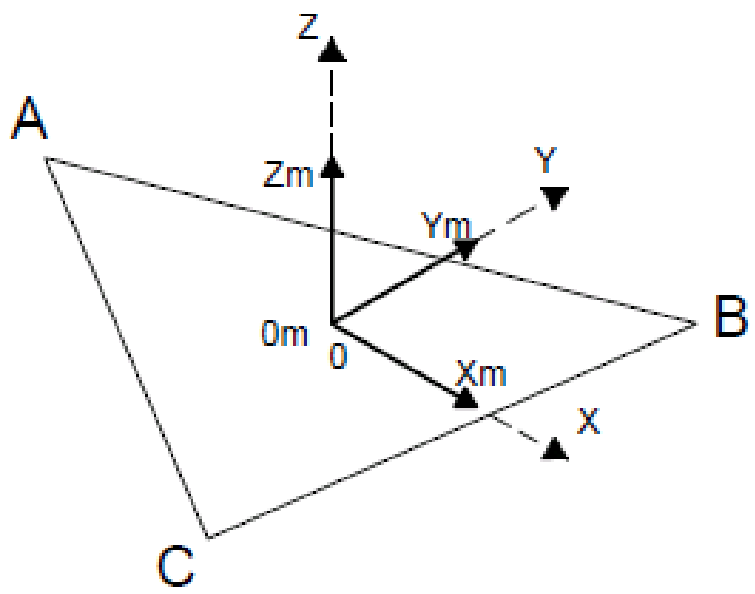


Figura 8. Orientación de sistemas coordenadas absoluto $(0, X, Y, Z)$ y móvil $(0_m, X_m, Y_m, Z_m)$.

Para determinar el desplazamiento de la plataforma, se compara las posiciones del sistema de coordenadas móvil de la plataforma $(0_m, X_m, Y_m, Z_m)$, respecto al sistema de coordenadas absoluto de la base del robot $(0, X, Y, Z)$, así se determina la variación de la posición a lo largo de los ejes X y Z, Figura 9, quedando expresado por la siguiente matriz:

$$[T_{XZ}] = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Donde, x representa el desplazamiento de 0_m en dirección del eje X y z representa el desplazamiento de 0_m en dirección del eje Z.

Cabe resaltar que este análisis es limitado al movimiento de traslación y sólo analiza el punto 0_m, ya que dicho punto es el único perteneciente a la plataforma que siempre se traslada sobre el plano XZ.

En cambio, para estudiar la rotación de la plataforma, es necesario definir vectores que conecten el centro (0_m) con los vértices (A, B, C), sobre los cuales se realiza dicho análisis.

Estos vectores, son definidos por el radio R_m del círculo circunscrito a la plataforma y los ángulos a los que se encuentran rotados cada vector respecto al sistema de coordenadas de la plataforma móvil (X_m, Y_m). Dichos ángulos son representados por la variable (α), de tal forma que los vértices $A, B, C,$ son determinados por los valores 180, 60 y -60, respectivamente, como se indica en la siguiente figura:

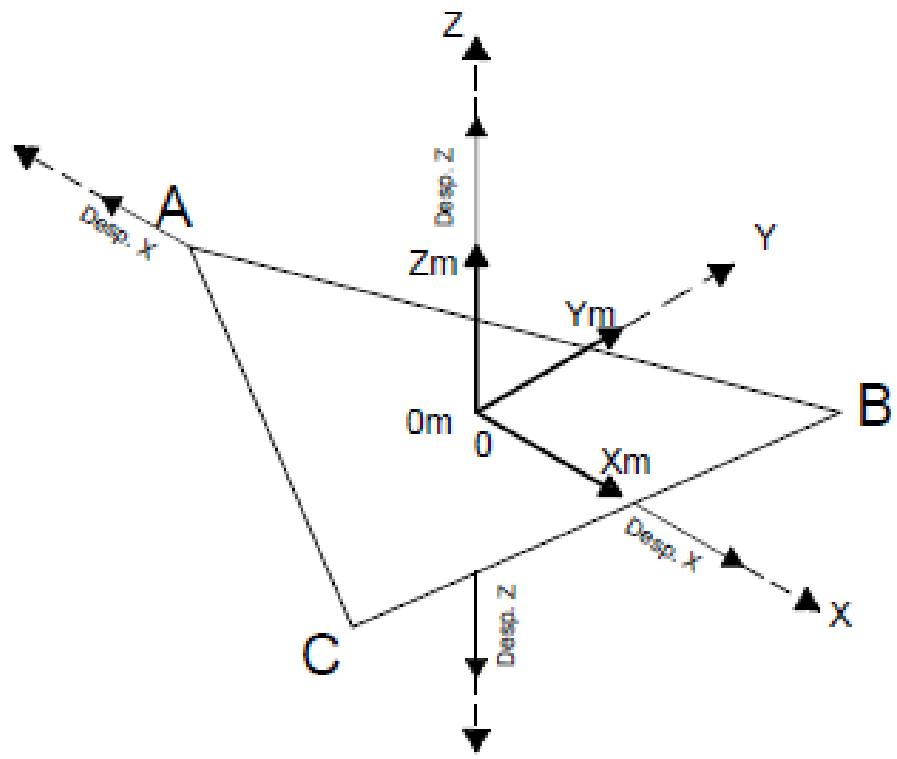


Figura 9. Movimientos de desplazamiento de la plataforma.

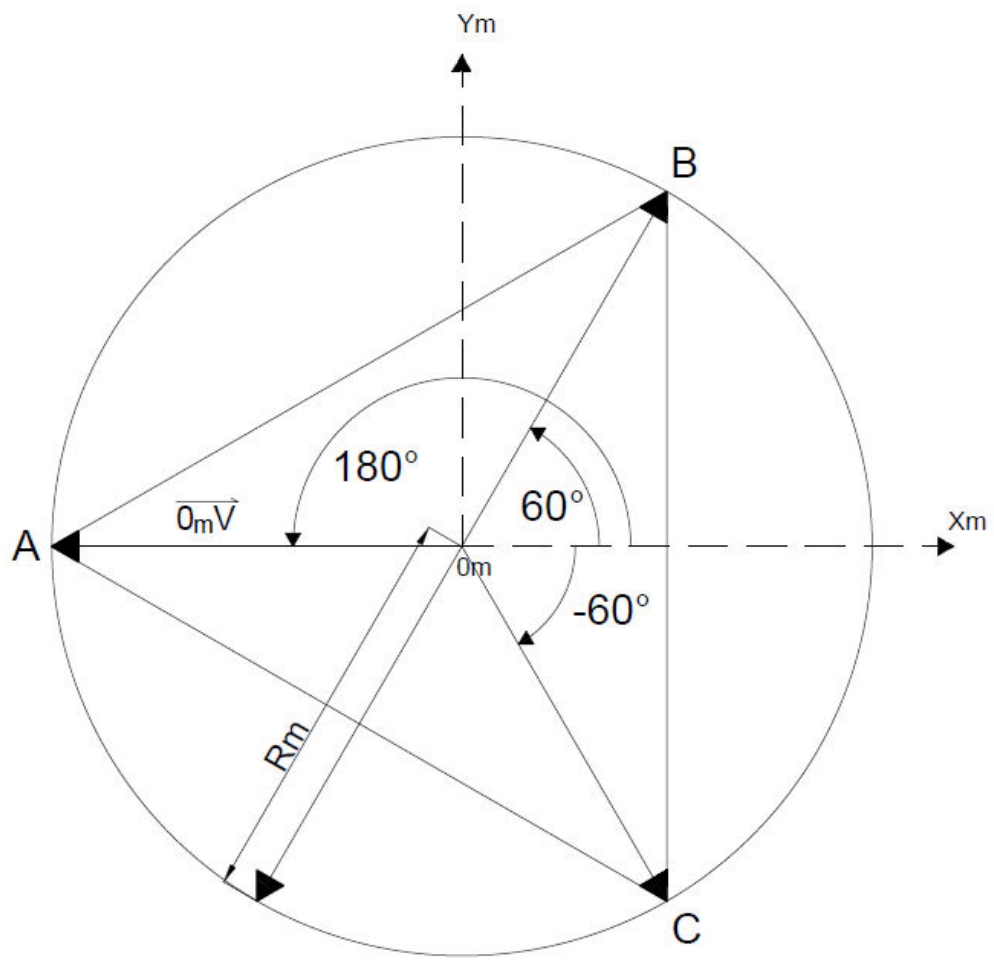


Figura 10. Orientación de vectores $\overrightarrow{0_m A}$, $\overrightarrow{0_m B}$ y $\overrightarrow{0_m C}$ de la plataforma móvil

De esta manera se obtiene los siguientes vectores en el espacio:

$$\overrightarrow{0_m A} = [R_m \cos(180) \quad R_m \sin(180) \quad 0] \tag{2.2}$$

$$\overrightarrow{0_m B} = [R_m \cos(60) \quad R_m \sin(60) \quad 0] \tag{2.3}$$

$$\overrightarrow{0_m C} = [R_m \cos(-60) \quad R_m \sin(-60) \quad 0] \tag{2.4}$$

Definiendo de forma general a cada vector como:

$$V = \{A, B, C\} \wedge \alpha = \{180,60,-60\} \tag{2.5}$$

Para poder trabajar con matrices homogéneas los vectores deben ser ampliados de un espacio 1×3 a 1×4, de tal forma que su valor no se vea alterado, esta condición junto a la definición en (2.5), se obtiene:

$$\overrightarrow{0_m V} = [R_m \cos(\alpha) \quad R_m \sin(\alpha) \quad 0 \quad 1] \tag{2.6}$$

Para describir la rotación de la plataforma se usa sistemas de ejes intrínsecos a los movimientos alrededor de los ejes Y y Z, Figura 11, los cuales están definidos por la matriz $[R_{ZY}]$.

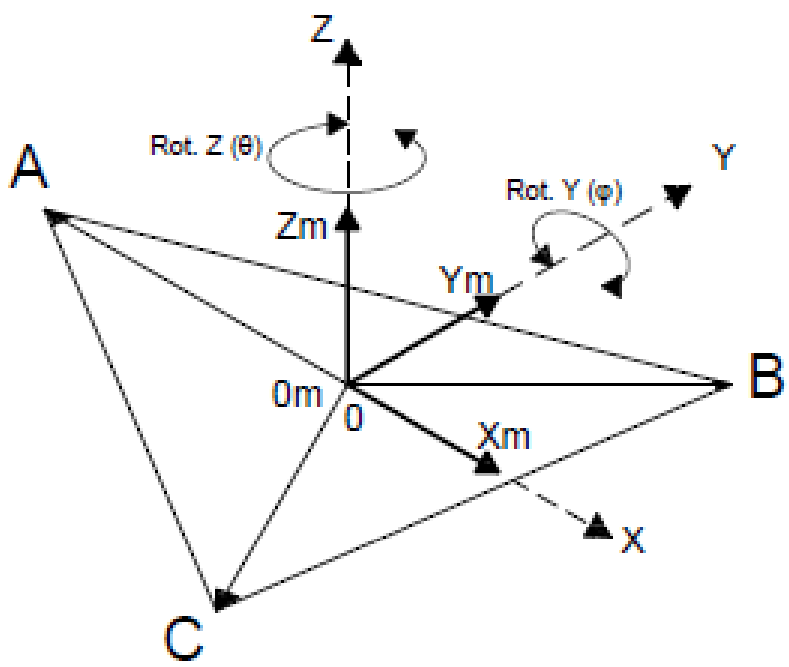


Figura 11. Movimientos de rotación de la plataforma.

$$[R_{ZY}] = \begin{bmatrix} \cos(\phi)\cos(\theta) & -\cos(\phi)\sin(\theta) & \sin(\phi) & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ -\sin(\phi)\cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

Donde, la rotación alrededor del eje Y_m y Z_m quedan definidos por los ángulos phi (ϕ) y theta (θ) respectivamente.

Al multiplicar las matrices $[T_{XZ}]$ y $[R_{ZY}]$, se obtiene:

$$[T_{XZ}][R_{ZY}] = \begin{bmatrix} \cos(\phi)\cos(\theta) & -\cos(\phi)\sin(\theta) & \sin(\phi) & x \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ -\sin(\phi)\cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi) & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

Expresión que describe completamente los movimientos de traslación y rotación de la

plataforma respecto al sistema de coordenadas fijo $(0, X, Y, Z)$.

Ecuaciones de posición

La matriz anterior (2.8), define el movimiento de la plataforma, a través de la traslación del punto 0_m, y la rotación de los vectores que parten del mismo hacia cada vértice $(\overrightarrow{0_m V})$. Para definir las ecuaciones de posición de cada vértice de la plataforma, es necesario realizar el producto entre la matriz de movimiento de la plataforma (2.8) y el traspuesto del vector $\overrightarrow{0_m V}$:

$$[PV] = \begin{bmatrix} \cos(\phi)\cos(\theta) & -\cos(\phi)\sin(\theta) & \sin(\phi) & x \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ -\sin(\phi)\cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi) & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_m\cos(\alpha) \\ R_m\sin(\alpha) \\ 0 \\ 1 \end{bmatrix} \tag{2.9}$$

Donde, $[PV]$ es el vector posición de los vértices y es regido por la condición en (2.5):

$$[PV] = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} \tag{2.10}$$

Obteniendo como resultado:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \\ 1 \end{bmatrix} = \begin{bmatrix} R_m\cos(\alpha)\cos(\phi)\cos(\theta) - R_m\sin(\alpha)\cos(\phi)\sin(\theta) + x \\ R_m\cos(\alpha)\sin(\theta) + R_m\sin(\alpha)\cos(\theta) \\ -R_m\cos(\alpha)\sin(\phi)\cos(\theta) + R_m\sin(\alpha)\sin(\phi)\sin(\theta) + z \\ 1 \end{bmatrix} \tag{2.11}$$

Expresión que permite obtener las ecuaciones generales de posición de cualquier vértice (V) de la plataforma en el espacio.

$$V_x = R_m\cos(\alpha)\cos(\phi)\cos(\theta) - R_m\sin(\alpha)\cos(\phi)\sin(\theta) + x \tag{2.12}$$

$$V_y = R_m\cos(\alpha)\sin(\theta) + R_m\sin(\alpha)\cos(\theta) \tag{2.13}$$

$$V_z = -R_m\cos(\alpha)\sin(\phi)\cos(\theta) + R_m\sin(\alpha)\sin(\phi)\sin(\theta) + z \tag{2.14}$$

Ecuaciones de velocidad

Para hallar las ecuaciones de velocidad, se debe derivar las ecuaciones descritas anteriormente (2.12), (2.13) y (2.14) para cada una de las variables que representan los grados de libertad del robot (x, z, θ, ϕ).

En este caso, las derivadas de las ecuaciones de posición son de la forma:

$$\frac{dVx}{dt} = \frac{\partial Vx}{\partial \phi}\frac{\partial \phi}{\partial t} + \frac{\partial Vx}{\partial \theta}\frac{\partial \theta}{\partial t} + \frac{\partial Vx}{\partial x}\frac{\partial x}{\partial t} \tag{2.15}$$

$$\frac{dVy}{dt} = \frac{\partial Vy}{\partial \theta}\frac{\partial \theta}{\partial t} \tag{2.16}$$

$$\frac{dVz}{dt} = \frac{\partial Vz}{\partial \phi}\frac{\partial \phi}{\partial t} + \frac{\partial Vz}{\partial \theta}\frac{\partial \theta}{\partial t} + \frac{\partial Vz}{\partial z}\frac{\partial z}{\partial t} \tag{2.17}$$

Aplicando las derivadas a su correspondiente ecuación de posición, se tiene como resultado las siguientes expresiones:

$$\begin{aligned}\frac{dVx}{dt} = &-[R_m \cos(\alpha)\sin(\phi)\cos(\theta) - R_m \sin(\alpha)\sin(\phi)\sin(\theta)]\dot{\phi} \\ &- [R_m \cos(\alpha)\cos(\phi)\sin(\theta) + R_m \sin(\alpha)\cos(\phi)\cos(\theta)]\dot{\theta} + \dot{x}\end{aligned} \tag{2.18}$$

$$\frac{dVy}{dt} = [R_m \cos(\alpha)\cos(\theta) - R_m \sin(\alpha)\sin(\theta)]\dot{\theta} \tag{2.19}$$

$$\begin{aligned}\frac{dVz}{dt} = &[-R_m \cos(\alpha)\cos(\phi)\cos(\theta) + R_m \sin(\alpha)\cos(\phi)\sin(\theta)]\dot{\phi} \\ &+ [R_m \cos(\alpha)\sin(\phi)\sin(\theta) + R_m \sin(\alpha)\sin(\phi)\cos(\theta)]\dot{\theta} + \dot{z}\end{aligned} \tag{2.20}$$

Que pueden ser expresadas de forma matricial:

$$[vV] = \begin{bmatrix} \frac{dVx}{dt} \\ \frac{dVy}{dt} \\ \frac{dVz}{dt} \end{bmatrix} = [J_P]\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{x} \\ \dot{z} \end{bmatrix} \tag{2.21}$$

Donde, $[vV]$ es el vector de componentes de la velocidad de cada vértice y $[J_P]$ corresponde a la matriz Jacobiana de la plataforma.

Al ser $[J_P]$ muy extensa, se realiza un desarrollo por columnas:

$$[J_P]_{n\times 1} = \begin{bmatrix} -[R_m \cos(\alpha)\sin(\phi)\cos(\theta) - R_m \sin(\alpha)\sin(\phi)\sin(\theta)] \\ 0 \\ [-R_m \cos(\alpha)\cos(\phi)\cos(\theta) + R_m \sin(\alpha)\cos(\phi)\sin(\theta)] \end{bmatrix}$$

$$[J_P]_{n\times 2} = \begin{bmatrix} -[R_m \cos(\alpha)\cos(\phi)\sin(\theta) + R_m \sin(\alpha)\cos(\phi)\cos(\theta)] \\ [R_m \cos(\alpha)\cos(\theta) - R_m \sin(\alpha)\sin(\theta)] \\ [R_m \cos(\alpha)\sin(\phi)\sin(\theta) + R_m \sin(\alpha)\sin(\phi)\cos(\theta)] \end{bmatrix}$$

$$[J_P]_{n\times 3} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; [J_P]_{n\times 4} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Cabe recordar que para obtener la velocidad de cada vértice, se necesita reemplazar el valor de α correspondiente a cada uno, tal como se muestra en la Figura 10.

Ecuaciones de aceleración

Para el cálculo de las ecuaciones de aceleración se procede a derivar las ecuaciones de posición y velocidad halladas anteriormente, la forma de la derivada para estas ecuaciones es la siguiente:

$$\frac{d^2Vx}{dt^2} = \frac{\partial Vx}{\partial \phi}\frac{\partial^2 \phi}{\partial t^2} + \frac{\partial Vx}{\partial \theta}\frac{\partial^2 \theta}{\partial t^2} + \frac{\partial Vx}{\partial x}\frac{\partial^2 x}{\partial t^2} + \frac{\partial^2 Vx}{\partial t \partial \phi}\frac{\partial \phi}{\partial t} + \frac{\partial^2 Vx}{\partial t \partial \theta}\frac{\partial \theta}{\partial t} + \frac{\partial^2 Vx}{\partial t \partial x}\frac{\partial x}{\partial t} \tag{2.22}$$

$$\frac{d^2Vy}{dt^2} = \frac{\partial Vy}{\partial \theta}\frac{\partial^2 \theta}{\partial t^2} + \frac{\partial^2 Vy}{\partial t \partial \theta}\frac{\partial \theta}{\partial t} \tag{2.23}$$

$$\frac{d^2Vz}{dt^2} = \frac{\partial Vz}{\partial \phi}\frac{\partial^2 \phi}{\partial t^2} + \frac{\partial Vz}{\partial \theta}\frac{\partial^2 \theta}{\partial t^2} + \frac{\partial Vz}{\partial z}\frac{\partial^2 z}{\partial t^2} + \frac{\partial^2 Vz}{\partial t \partial \phi}\frac{\partial \phi}{\partial t} + \frac{\partial^2 Vz}{\partial t \partial \theta}\frac{\partial \theta}{\partial t} + \frac{\partial^2 Vz}{\partial t \partial z}\frac{\partial z}{\partial t} \tag{2.24}$$

Al calcular la derivada y expresarla de forma matricial se obtiene:

$$[aV] = \begin{bmatrix} \frac{d^2Vx}{dt^2} \\ \frac{d^2Vy}{dt^2} \\ \frac{d^2Vz}{dt^2} \end{bmatrix} = [MaP] \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{x} \\ \ddot{z} \\ 1 \end{bmatrix} \tag{2.25}$$

Donde, $[aV]$ es el vector de componentes de la aceleración de cada vértice y $[MaP]$ corresponde a la matriz de aceleración de la plataforma.

Al ser $[MaP]$ muy extensa, se realiza un desarrollo por columnas:

$$[MaP]_{n\times 1} = \begin{bmatrix} -[R_m \cos(\alpha)\sin(\phi)\cos(\theta) - R_m \sin(\alpha)\sin(\phi)\sin(\theta)] \\ 0 \\ -[R_m \cos(\alpha)\cos(\phi)\cos(\theta) - R_m \sin(\alpha)\cos(\phi)\sin(\theta)] \end{bmatrix}$$

$$[MaP]_{n\times 2} = \begin{bmatrix} [R_m \cos(\alpha)\cos(\phi)\sin(\theta) + R_m \sin(\alpha)\cos(\phi)\cos(\theta)] \\ [R_m \cos(\alpha)\cos(\theta) - R_m \sin(\alpha)\sin(\theta)] \\ [R_m \cos(\alpha)\sin(\phi)\sin(\theta) + R_m \sin(\alpha)\sin(\phi)\cos(\theta)] \end{bmatrix}$$

$$[MaP]_{n\times 3} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; [MaP]_{n\times 4} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$[MaP]_{n\times 5} = \begin{bmatrix} [MaP]_{(1,5)} \\ -[R_m \cos(\alpha)\sin(\theta) + R_m \sin(\alpha)\cos(\theta)]\dot{\theta}^2 \\ [MaP]_{(3,5)} \end{bmatrix}$$

Los términos $[MaP]_{(1,5)}$ y $[MaP]_{(3,5)}$, por su extensión, son escritos por separado:

$$[MaP]_{(1,5)} = 2[R_m \cos(\alpha)\sin(\phi)\sin(\theta) + R_m \sin(\alpha)\sin(\phi)\cos(\theta)]\dot{\phi}\dot{\theta}$$
$$- [R_m \cos(\alpha)\cos(\phi)\cos(\theta) - R_m \sin(\alpha)\cos(\phi)\sin(\theta)]\dot{\phi}^2$$
$$- [R_m \cos(\alpha)\cos(\phi)\cos(\theta) - R_m \sin(\alpha)\cos(\phi)\sin(\theta)]\dot{\theta}^2$$

$$[MaP]_{(3,5)} = 2[R_m \cos(\alpha)\cos(\phi)\sin(\theta) + R_m \sin(\alpha)\cos(\phi)\cos(\theta)]\dot{\phi}\dot{\theta}$$
$$+ [R_m \cos(\alpha)\sin(\phi)\cos(\theta) - R_m \sin(\alpha)\sin(\phi)\sin(\theta)]\dot{\phi}^2$$
$$+ [R_m \cos(\alpha)\sin(\phi)\cos(\theta) - R_m \sin(\alpha)\sin(\phi)\sin(\theta)]\dot{\theta}^2$$

Cabe recalcar que los términos en la columna $[MaP]_{n\times 5}$, representan a las aceleraciones normales presentes en el movimiento de la plataforma.

Análisis cinemático de los brazos UPE

El análisis cinemático de los brazos UPE del robot paralelo se realiza con el método de Denavit & Hartenberg. Como se conoce que los brazos UPE se encuentran ubicados en cada vértice de la plataforma base, en el análisis solamente se considera a uno de ellos, en este caso se procede con el brazo ubicado en el vértice A de la base. En la Figura 12, se observa la representación esquemática del brazo UPE, constando de dos articulaciones rotacionales las cuales equivalen a la articulación universal, seguido de la articulación prismática. La articulación esférica del brazo UPE es excluida del esquema, ya que ésta no es analizada.

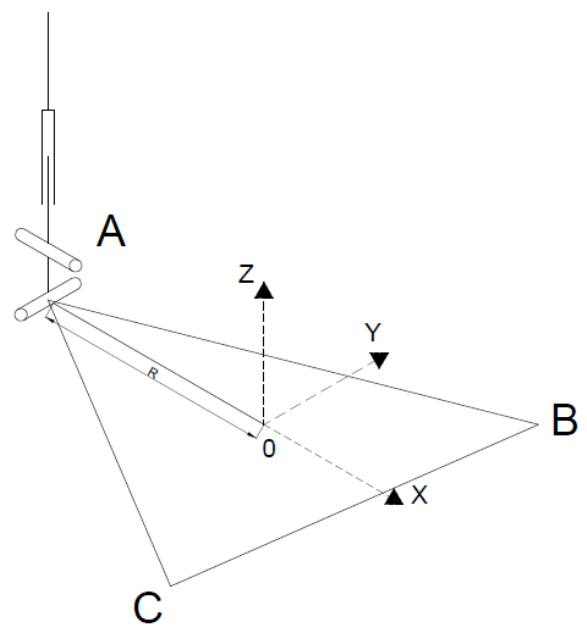


Figura 12. Esquema del brazo UPE ubicado en el vértice A

Una consideración adicional a tomarse en cuenta al momento de analizar la articulación universal, es que, se desprecia la distancia existente entre los ejes de rotación en esta articulación, permitiendo analizar los movimientos como si sus se intersecasen entre sí.

Análisis Denavit & Hartenberg

Iniciando con el método D&H, a partir del punto 0 del sistema de coordenadas absoluto $(0, X, Y, Z)$, se traslada un valor de R, correspondiente al radio del círculo circunscrito a la base, en la dirección negativa del eje X. Es en este punto donde se ubica la base del brazo y la primera articulación rotacional del mismo, a esta posición se la representa a través del sistema X_1, Y_1, Z_1.

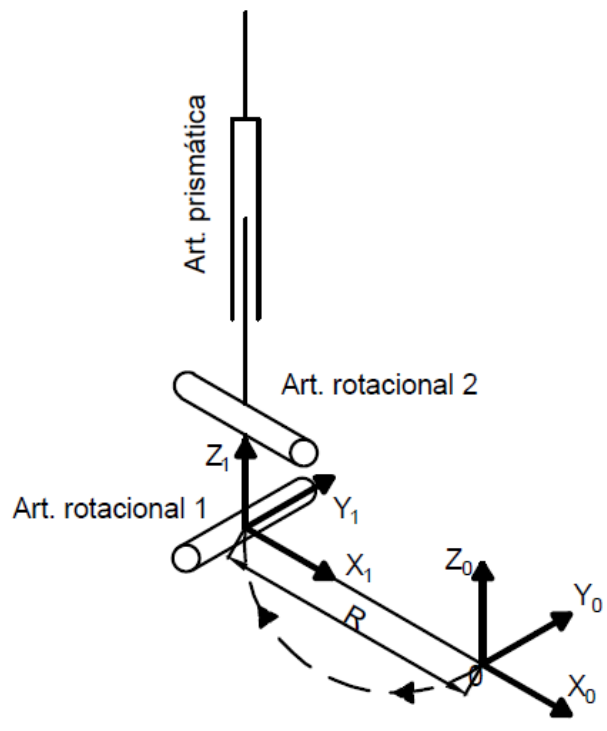


Figura 13. Método D&H brazo A, primer movimiento

Para describir el movimiento de la primera articulación rotacional, el anterior sistema (X_1, Y_1, Z_1) se rota $+90°$ alrededor del eje X_1, obteniendo el sistema X_2, Y_2, Z_2. En este sistema de referencia se analiza el giro de la articulación rotacional 1 mediante el ángulo θ_1, ángulo que describe la rotación del sistema X_2, Y_2, Z_2 alrededor del eje Z_2. Figura 14

Para analizar la segunda articulación rotatoria se rota el sistema X_2, Y_2, Z_2 un ángulo de $+90°$ alrededor de Z_2, generando X_3, Y_3, Z_3. Figura 15

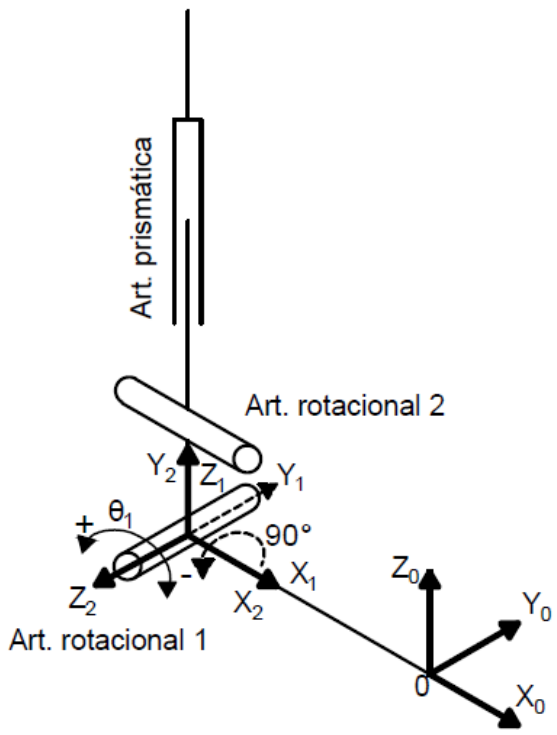


Figura 14. Método D&H brazo A, segundo movimiento y análisis de la primera articulación rotacional

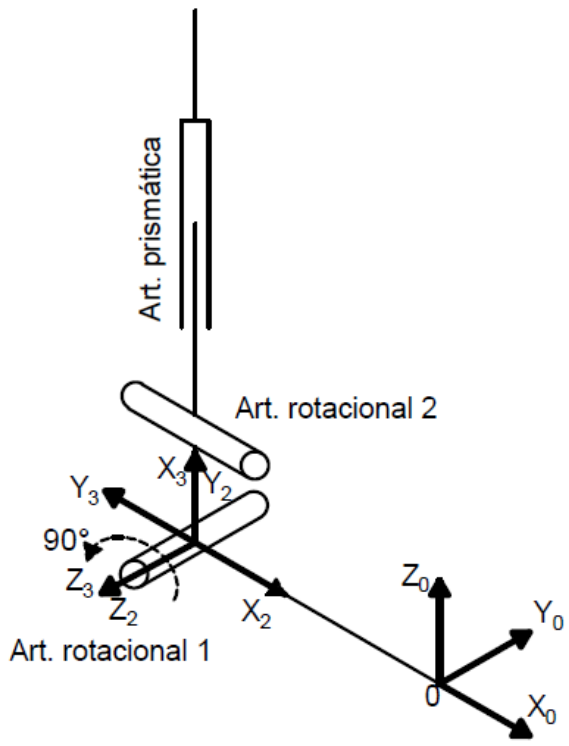


Figura 15. Método D&H brazo A, tercer movimiento

En este punto, el sistema de coordenadas X_3, Y_3, Z_3, se lo ubica al nivel de la segunda articulación rotacional, ya que ambas articulaciones corresponden a una descomposición de la articulación universal del brazo.

El sistema X_3, Y_3, Z_3 se rota $+90°$ alrededor de X_3, generando X_4, Y_4, Z_4. Figura 16. Mediante el ángulo θ_2 se analiza el movimiento de rotación de la segunda articulación rotacional, este ángulo describe la rotación del sistema X_4, Y_4, Z_4 alrededor del eje Z_4.

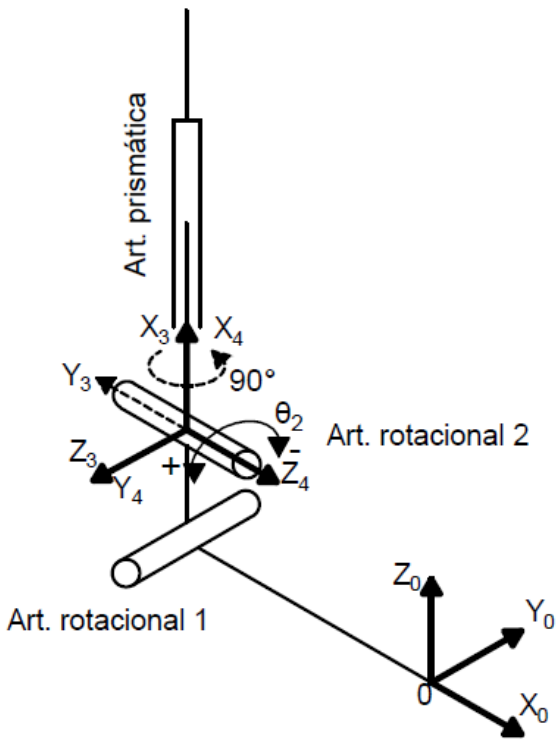


Figura 16. Método D&H brazo A, cuarto movimiento y análisis de la segunda articulación rotacional

Finalmente, el sistema X_4, Y_4, Z_4 es trasladado sobre la articulación prismática del brazo una distancia a_3 a lo largo del eje X_4, generando el sistema X_5, Y_5, Z_5, ubicado al extremo superior del brazo. El valor de a_3 corresponde a la longitud del brazo al variar el desplazamiento de la articulación prismática.

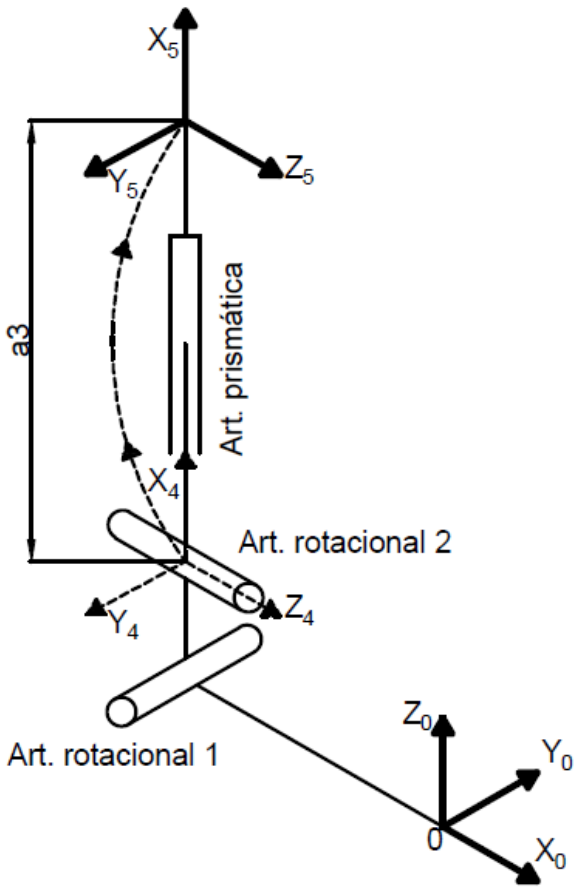


Figura 17. Método D&H brazo A, análisis de articulación prismática

Como resultado del análisis por el método D&H, el movimiento y sentido de movimiento de las articulaciones del brazo A son descritos por las variables θ_1, θ_2 y a_3 como se muestra

en la siguiente figura:

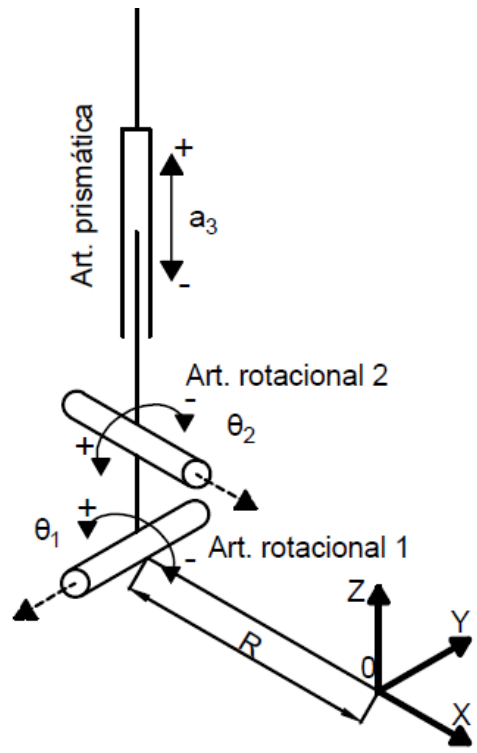


Figura 18. Brazo A, movimiento y sentido de los movimientos de las articulaciones universal y prismática

El resumen de los parámetros D&H se presenta en la siguiente tabla:

Tabla 1. Resumen de parámetros D&H del brazo A

d	θ	a	α
$-$	$-$	$-R_m$	$90°$
$-$	$\theta_1 + 90°$	$-$	$90°$
$-$	θ_2	a_3	$-$

Para convertir los resultados obtenidos para el brazo A, a resultados aplicables a todos los brazos UPE en forma general, se debe tener en cuenta la localización de los brazos en cada vértice respecto al centro del sistema de coordenadas absoluto $(0, X, Y, Z)$, de esta manera, se puede considerar a los brazos como la rotación de uno de ellos alrededor del eje Z. Figura 19.

De esta forma, el ángulo de rotación necesario para definir los brazos restantes debe ser medido desde el brazo A, como se muestra en la Figura 20.

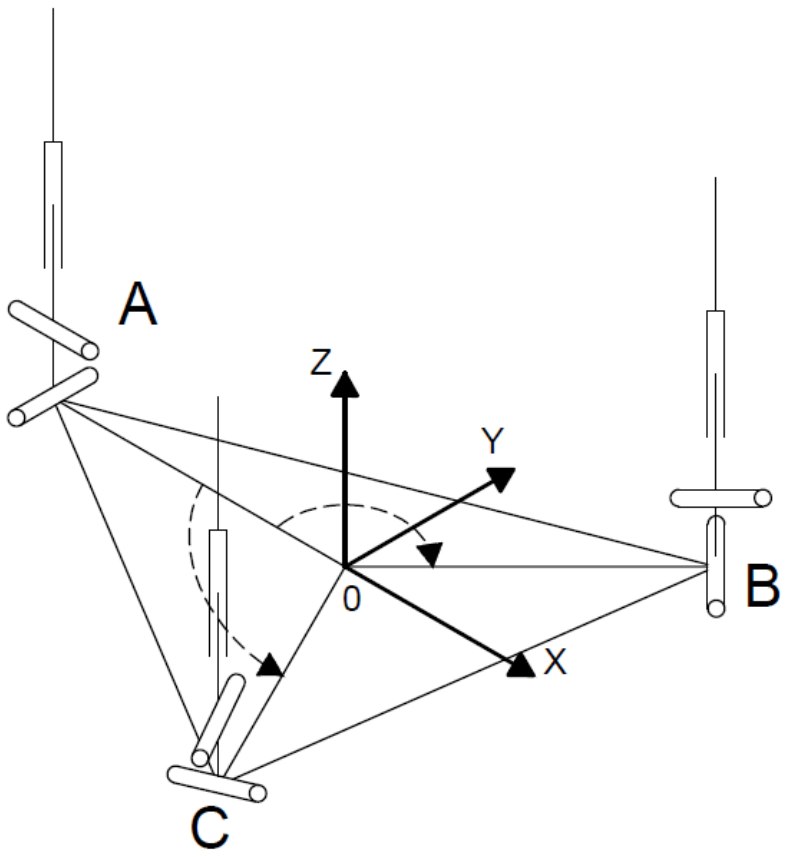


Figura 19. Rotación del brazo A alrededor del eje Z hacia los vértices B y C.

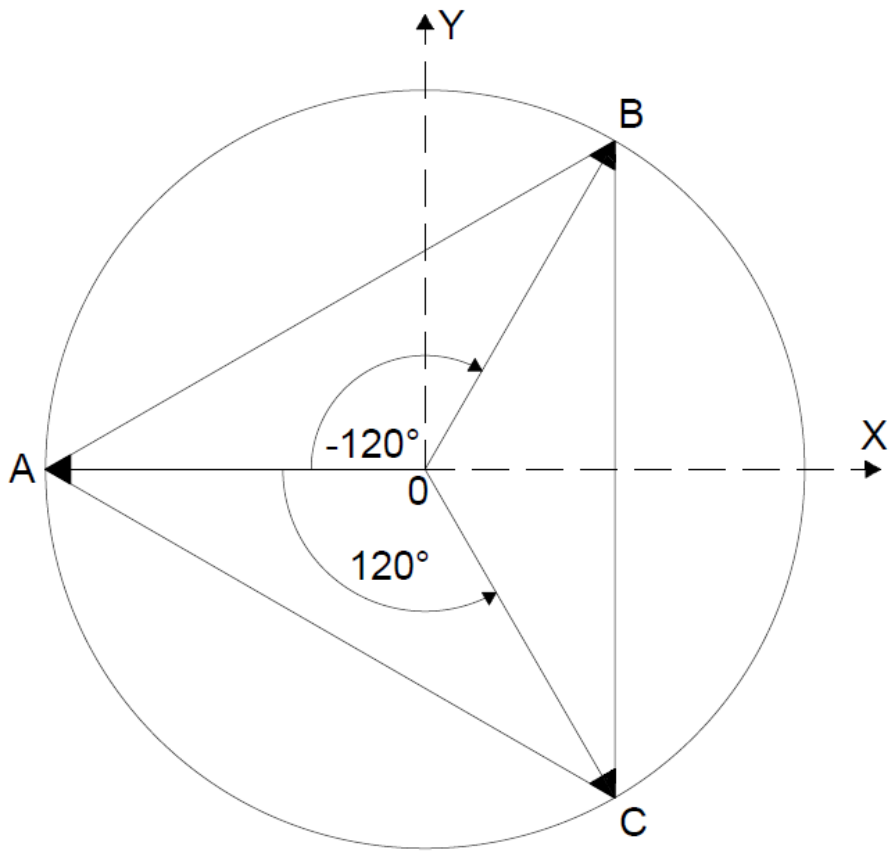


Figura 20. Orientación de los brazos B y C respecto al brazo A

Los ángulos descritos en la figura anterior son representados por la variable β, de esta manera al referirse a los brazos A, B o C; β toma el valor de $0°, -120°$ o $120°$, según corresponda.

Al implementar el ángulo β al análisis D&H del brazo A, éste ángulo se convierte en un paso previo al análisis de dicho brazo, por lo cual los parámetros resultantes son:

Tabla 2. Resumen de parámetros D&H para los brazos UPE

d	θ	a	α
$-$	β	$-R_m$	$90°$
$-$	$\theta_1 + 90°$	$-$	$90°$
$-$	θ_2	a_3	$-$

Ecuaciones de posición

A partir de los resultados de la tabla resumen de parámetros D&H para los brazos UPE, se inicia la determinación de las ecuaciones de posición del extremo de dichos brazos. Estos resultados son expresados en forma de matriz, la cual toma el nombre de matriz homogénea y es usada para determinar las ecuaciones que se requieren.

$$[U] = \begin{bmatrix} U_x \\ U_y \\ U_z \\ 1 \end{bmatrix} = [M_UPE_{4\times4}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.26}$$

Donde, $[U]$ corresponde a las componentes x, y, z del vector posición del brazo UPE, $[M_UPE_{4\times4}]$ es la matriz homogénea determinada a partir de los parámetros de la Tabla 2 y que se encuentra multiplicada por el vector posición del origen de coordenadas.

Cómo se observa en la ecuación anterior, al multiplicar la matriz homogénea por el vector posición del origen, las primeras tres columnas de la matriz son anuladas, conservándose únicamente la cuarta columna, lo que da como resultado:

$$\begin{bmatrix} U_x \\ U_y \\ U_z \\ 1 \end{bmatrix} = \begin{bmatrix} -R\cos(\beta) + B_1\sin(\beta)\sin(B_3) - B_1\cos(\beta)\sin(B_2)\cos(B_3) \\ -R\sin(\beta) - B_1\cos(\beta)\sin(B_3) - B_1\sin(\beta)\sin(B_2)\cos(B_3) \\ B_1\cos(B_2)\cos(B_3) \\ 1 \end{bmatrix} \tag{2.27}$$

Qué expresada de otra manera, se tiene:

$$U_x = -R\cos(\beta) + B_1\sin(\beta)\sin(B_3) - B_1\cos(\beta)\sin(B_2)\cos(B_3) \tag{2.28}$$

$$U_y = -R\sin(\beta) - B_1\cos(\beta)\sin(B_3) - B_1\sin(\beta)\sin(B_2)\cos(B_3) \tag{2.29}$$

$$U_z = B_1\cos(B_2)\cos(B_3) \tag{2.30}$$

Donde, R es el radio del circulo circunscrito a la base, β es el ángulo de rotación para cada brazo y B_1, B_2 y B_3 corresponden a las variables de las articulaciones del brazo y equivalen a los valores de θ_1, θ_2 y a_3, hallados por el método D&H.

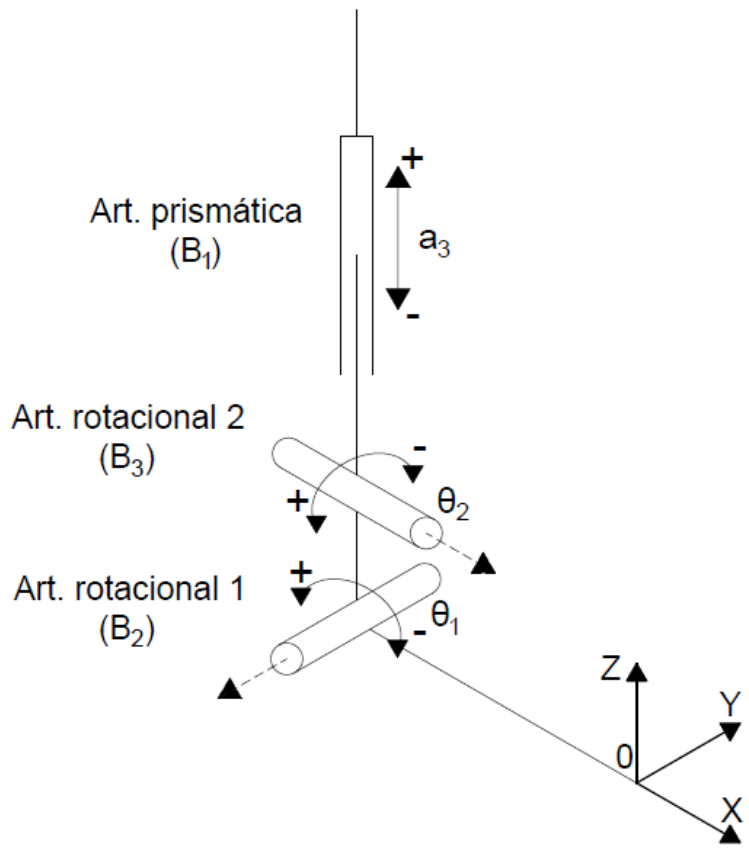


Figura 21. Equivalencia entre variables de ecuaciones y parámetros obtenidos por el método D&H en el brazo UPE

Ecuaciones de velocidad

Las ecuaciones de velocidad del brazo UPE se desarrollan en base a las ecuaciones de posición del brazo (2.28), (2.29), (2.30); para esto, la forma de las derivadas es la siguiente:

$$\frac{dU_x}{dt} = \frac{\partial U_x}{\partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial U_x}{\partial B_2}\frac{\partial B_2}{\partial t} + \frac{\partial U_x}{\partial B_3}\frac{\partial B_3}{\partial t} \tag{2.31}$$

$$\frac{dU_y}{dt} = \frac{\partial U_y}{\partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial U_y}{\partial B_2}\frac{\partial B_2}{\partial t} + \frac{\partial U_y}{\partial B_3}\frac{\partial B_3}{\partial t} \tag{2.32}$$

$$\frac{dU_z}{dt} = \frac{\partial U_z}{\partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial U_z}{\partial B_2}\frac{\partial B_2}{\partial t} + \frac{\partial U_z}{\partial B_3}\frac{\partial B_3}{\partial t} \tag{2.33}$$

Al derivar las ecuaciones de posición del brazo UPE con su respectiva derivada, se obtiene como resultado las siguientes ecuaciones:

$$\begin{aligned}\frac{dU_x}{dt} = {} & [\sin(\beta)\sin(B_3) - \cos(\beta)\sin(B_2)\cos(B_3)]\dot{B}_1 \\ & - B_1\cos(\beta)\cos(B_2)\cos(B_3)\,\dot{B}_2 + [B_1\sin(\beta)\cos(B_3) \\ & + B_1\cos(\beta)\sin(B_2)\sin(B_3)]\dot{B}_3\end{aligned} \tag{2.34}$$

$$\begin{aligned}\frac{dU_y}{dt} = {} & [-\cos(\beta)\sin(B_3) - \sin(\beta)\sin(B_2)\cos(B_3)]\dot{B}_1 \\ & - B_1\sin(\beta)\cos(B_2)\cos(B_3)\,\dot{B}_2 + [-B_1\cos(\beta)\cos(B_3) \\ & + B_1\sin(\beta)\sin(B_2)\sin(B_3)]\dot{B}_3\end{aligned} \tag{2.35}$$

$$\frac{dU_z}{dt} = \cos(B_2)\cos(B_3)\,\dot{B}_1 - B_1\sin(B_2)\cos(B_3)\,\dot{B}_2 - B_1\cos(B_2)\sin(B_3)\,\dot{B}_3 \tag{2.36}$$

Qué al ser expresadas de forma matricial se obtiene:

$$[vU] = \begin{bmatrix} \frac{dU_x}{dt} \\ \frac{dU_y}{dt} \\ \frac{dU_z}{dt} \end{bmatrix} = [J_{UPE}] \begin{bmatrix} \dot{B}_1 \\ \dot{B}_2 \\ \dot{B}_3 \end{bmatrix} \tag{2.37}$$

Donde, $[vU]$ corresponde al vector de componentes de la velocidad de un brazo UPE dado, $[J_{UPE}]$ representa a la matriz Jacobiana del brazo UPE.

Al ser $[J_{UPE}]$ muy extensa, se presenta su desarrollo por columnas:

$$[J_{UPE}]_{n\times 1} = \begin{bmatrix} \sin(\beta)\sin(B_3) - \cos(\beta)\sin(B_2)\cos(B_3) \\ -\cos(\beta)\sin(B_3) - \sin(\beta)\sin(B_2)\cos(B_3) \\ \cos(B_2)\cos(B_3) \end{bmatrix}$$

$$[J_{UPE}]_{n\times 2} = \begin{bmatrix} -B_1\cos(\beta)\cos(B_2)\cos(B_3) \\ -B_1\sin(\beta)\cos(B_2)\cos(B_3) \\ -B_1\sin(B_2)\cos(B_3) \end{bmatrix}$$

$$[J_{UPE}]_{n\times 3} = \begin{bmatrix} B_1\sin(\beta)\cos(B_3) + B_1\cos(\beta)\sin(B_2)\sin(B_3) \\ -B_1\cos(\beta)\cos(B_3) + B_1\sin(\beta)\sin(B_2)\sin(B_3) \\ -B_1\cos(B_2)\sin(B_3) \end{bmatrix}$$

Ecuaciones de aceleración

Para determinar las ecuaciones de aceleración, se procede a derivar las ecuaciones de posición y velocidad del brazo UPE de acuerdo a la forma:

$$\begin{aligned} \frac{d^2U_x}{dt^2} = {} & \frac{\partial U_x}{\partial B_1}\frac{\partial^2 B_1}{\partial t^2} + \frac{\partial U_x}{\partial B_2}\frac{\partial^2 B_2}{\partial t^2} + \frac{\partial U_x}{\partial B_3}\frac{\partial^2 B_3}{\partial t^2} + \frac{\partial^2 U_x}{\partial t\partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial^2 U_x}{\partial t\partial B_2}\frac{\partial B_2}{\partial t} \\ & + \frac{\partial^2 U_x}{\partial t\partial B_3}\frac{\partial B_3}{\partial t} \end{aligned} \tag{2.38}$$

$$\begin{aligned} \frac{d^2U_y}{dt^2} = {} & \frac{\partial U_y}{\partial B_1}\frac{\partial^2 B_1}{\partial t^2} + \frac{\partial U_y}{\partial B_2}\frac{\partial^2 B_2}{\partial t^2} + \frac{\partial U_y}{\partial B_3}\frac{\partial^2 B_3}{\partial t^2} + \frac{\partial^2 U_y}{\partial t\partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial^2 U_y}{\partial t\partial B_2}\frac{\partial B_2}{\partial t} \\ & + \frac{\partial^2 U_y}{\partial t\partial B_3}\frac{\partial B_3}{\partial t} \end{aligned} \tag{2.39}$$

$$\frac{d^2U_z}{dt^2} = \frac{\partial U_z}{\partial B_1}\frac{\partial^2 B_1}{\partial t^2} + \frac{\partial U_z}{\partial B_2}\frac{\partial^2 B_2}{\partial t^2} + \frac{\partial U_z}{\partial B_3}\frac{\partial^2 B_3}{\partial t^2} + \frac{\partial^2 U_z}{\partial t \partial B_1}\frac{\partial B_1}{\partial t} + \frac{\partial^2 U_z}{\partial t \partial B_2}\frac{\partial B_2}{\partial t} + \frac{\partial^2 U_z}{\partial t \partial B_3}\frac{\partial B_3}{\partial t} \tag{2.40}$$

Al calcular la derivada respectiva y expresarlas de forma matricial, se obtiene:

$$[aU] = \begin{bmatrix} \frac{d^2U_x}{dt^2} \\ \frac{d^2U_y}{dt^2} \\ \frac{d^2U_z}{dt^2} \\ 1 \end{bmatrix} = [MaUPE]\begin{bmatrix} \ddot{B}_1 \\ \ddot{B}_2 \\ \ddot{B}_3 \\ 1 \end{bmatrix} \tag{2.41}$$

Donde, $[aU]$ es el vector de componentes de la aceleración de cada vértice y $[MaUPE]$ corresponde a la matriz de aceleración de la plataforma.

Al ser $[MaUPE]$ muy extensa, se realiza un desarrollo por columnas:

$$[MaUPE]_{n\times 1} = \begin{bmatrix} \sin(\beta)\sin(B_3) - \cos(\beta)\sin(B_2)\cos(B_3) \\ -\cos(\beta)\sin(B_3) - \sin(\beta)\sin(B_2)\cos(B_3) \\ \cos(B_2)\cos(B_3) \\ 1 \end{bmatrix}$$

$$[MaUPE]_{n\times 2} = \begin{bmatrix} -B_1\cos(\beta)\cos(B_2)\cos(B_3) \\ -B_1\sin(\beta)\cos(B_2)\cos(B_3) \\ -B_1\sin(B_2)\cos(B_3) \\ 1 \end{bmatrix}$$

$$[MaUPE]_{n\times 3} = \begin{bmatrix} B_1\sin(\beta)\cos(B_3) + B_1\cos(\beta)\sin(B_2)\sin(B_3) \\ -B_1\cos(\beta)\cos(B_3) + B_1\sin(\beta)\sin(B_2)\sin(B_3) \\ -B_1\cos(B_2)\sin(B_3) \\ 1 \end{bmatrix}$$

$$[MaUPE]_{n\times 4} = \begin{bmatrix} [MaUPE]_{(1,4)} \\ [MaUPE]_{(2,4)} \\ [MaUPE]_{(3,4)} \\ 1 \end{bmatrix}$$

Los términos correspondientes a la cuarta columna de la matriz $[MaUPE]$ al ser muy extensos, se los presenta por separado:

$$\begin{aligned} [MaUPE]_{(1,4)} = & -2\cos(\beta)\cos(B_2)\cos(B_3)\,\dot{B}_1\dot{B}_2 \\ & + 2[\sin(\beta)\cos(B_3) + \cos(\beta)\sin(B_2)\sin(B_3)]\dot{B}_1\dot{B}_3 \\ & + 2B_1\cos(\beta)\cos(B_2)\sin(B_3)\,\dot{B}_2\dot{B}_3 + B_1\cos(\beta)\sin(B_2)\cos(B_3)\,\dot{B}_2^2 \\ & - [B_1\sin(\beta)\sin(B_3) - B_1\cos(\beta)\sin(B_2)\cos(B_3)]\dot{B}_3^2 \end{aligned}$$

$$[MaUPE]_{(2,4)} = -2\sin(\beta)\cos(B_2)\cos(B_3)\,\dot{B}_1\dot{B}_2$$
$$- 2[\cos(\beta)\cos(B_3) - \sin(\beta)\sin(B_2)\sin(B_3)]\dot{B}_1\dot{B}_3$$
$$+ 2B_1\sin(\beta)\cos(B_2)\sin(B_3)\,\dot{B}_2\dot{B}_3 + B_1\sin(\beta)\sin(B_2)\cos(B_3)\,\dot{B}_2^2$$
$$+ [B_1\cos(\beta)\sin(B_3) + B_1\sin(\beta)\sin(B_2)\cos(B_3)]\dot{B}_3^2$$

$$[MaUPE]_{(3,4)} = -2\ \sin(B_2)\cos(B_3)\,\dot{B}_1\dot{B}_2 - 2\cos(B_2)\sin(B_3)\,\dot{B}_1\dot{B}_3$$
$$+ 2B_1\sin(B_2)\sin(B_3)\,\dot{B}_2\dot{B}_3 - B_1\cos(B_2)\cos(B_3)\,\dot{B}_2^2 - B_1\cos(B_2)\cos(B_3)\,\dot{B}_3^2$$

La columna $[MaUPE]_{n\times 4}$ compila las aceleraciones normales y de Coreolis presentes en el movimiento del brazo.

Análisis cinemático del brazo RPU

Para determinar las ecuaciones de posición del brazo RPU del robot, de la misma manera que en el caso de los brazos UPE, se desarrolla el análisis cinemático a través del método D&H. Para este caso la articulación universal del brazo también se la desarrolla como dos articulaciones rotacionales separadas, consecutivas e intersecantes entre sí a un ángulo de 90º. A este brazo se lo denomina con la letra D.

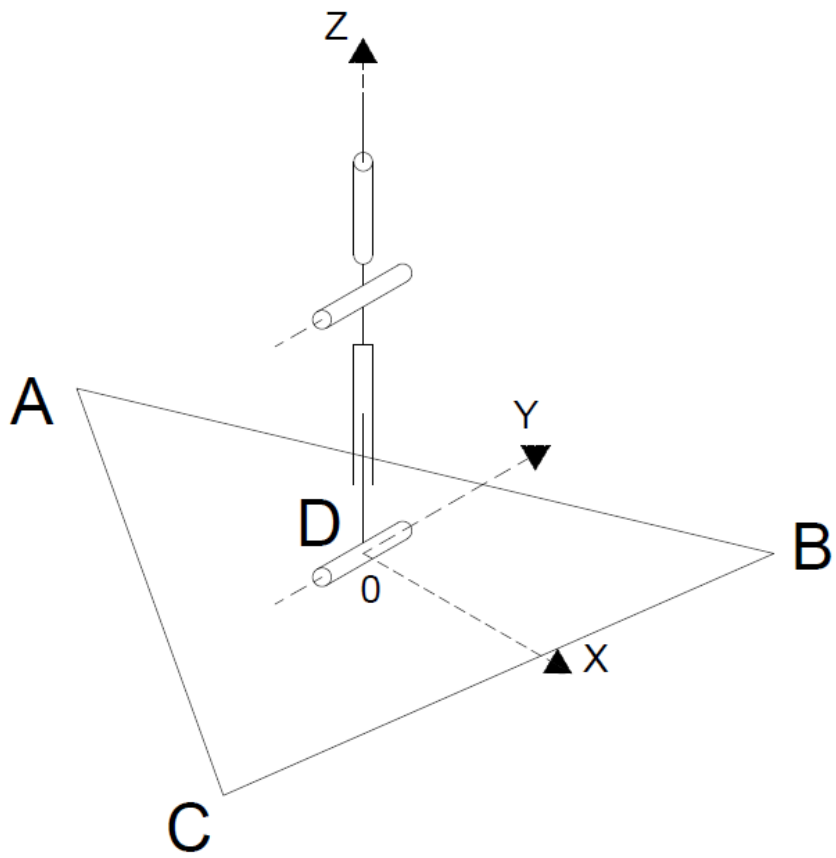


Figura 22. Esquema del brazo central del robot

Análisis Denavit & Hartenberg

El análisis parte del punto 0 con el sistema de coordenadas absoluto, a este se sistema se rota $+90°$ alrededor del eje Z, generando el sistema X_1, Y_1, Z_1. Con esta orientación se

analiza el ángulo α_1 que corresponde a la rotación del sistema X_1, Y_1, Z_1 alrededor del eje X_1, este valor también corresponde a la rotación de la primera articulación del brazo.

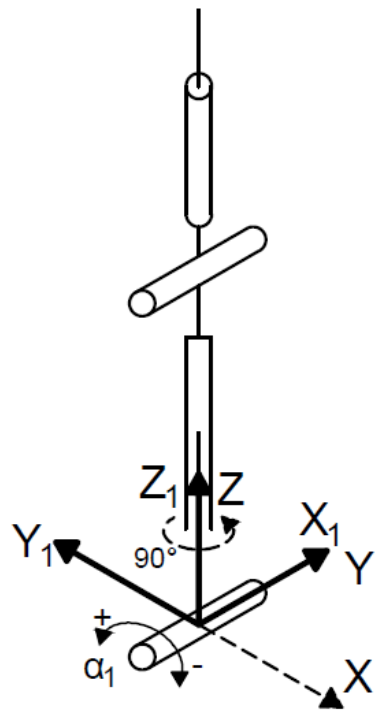


Figura 23. Método D&H brazo D, primer movimiento y análisis de la primera articulación rotacional

Desde este punto se traslada a lo largo del eje Z_1 una distancia a_2, la cual corresponde a la variación de la posición de la articulación prismática. Este movimiento genera el sistema X_2, Y_2, Z_2 ubicado sobre la articulación universal, representado en este caso como una segunda articulación rotacional.

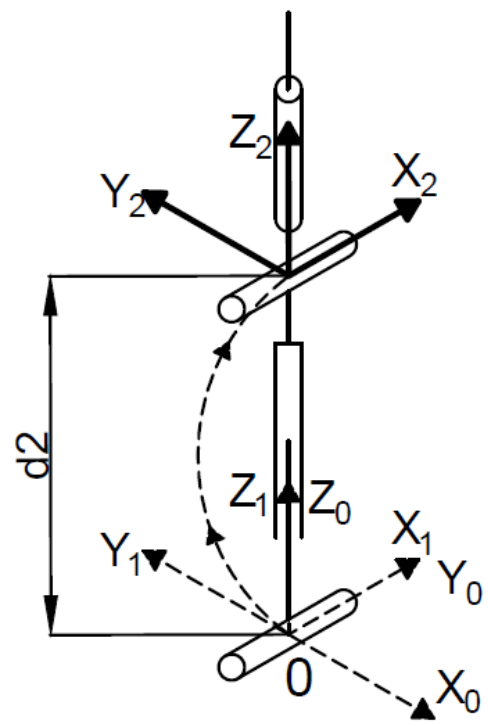


Figura 24. Método D&H brazo D, análisis de la articulación prismática

Con el sistema de coordenadas X_2, Y_2, Z_2 en esta ubicación se analiza el primer movimiento de rotación de la articulación universal, la variable asociada a este movimiento es α_2 y

representa el giro del sistema X_2, Y_2, Z_2 sobre el eje X_2, a la vez que describe la rotación de la articulación en este punto.

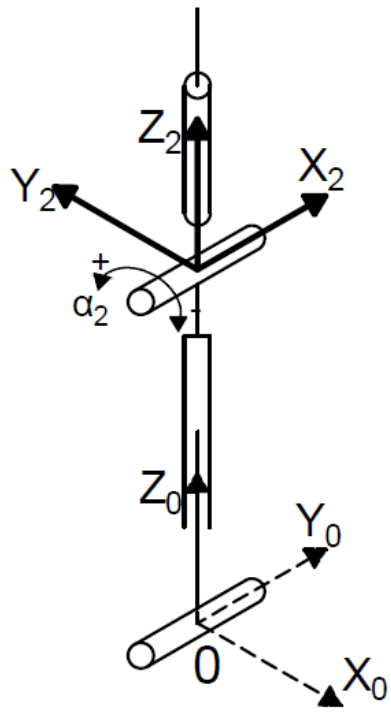


Figura 25. Método D&H brazo D, análisis de la segunda articulación rotacional

Finalmente la segunda rotación correspondiente a la articulación universal se la analiza a través el mismo sistema de coordenadas (X_2, Y_2, Z_2). El valor asociado a este movimiento está representado por el ángulo θ_3 y representa el giro del sistema X_2, Y_2, Z_2 sobre el eje Z_2 y la rotación de la articulación.

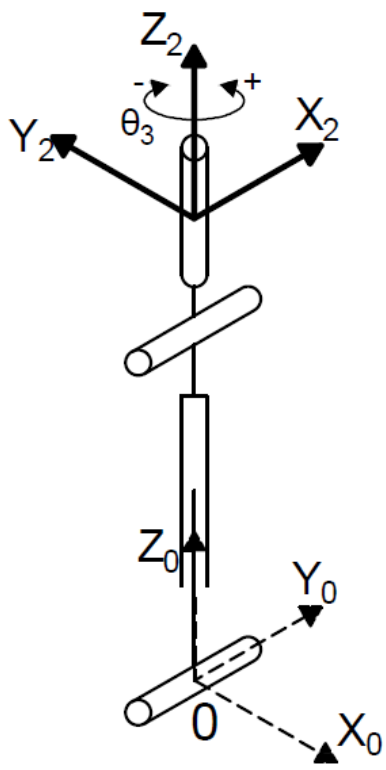


Figura 26. Método D&H brazo D, análisis de la tercera articulación rotacional

El resumen de los parámetros D&H se presenta en la siguiente tabla:

Tabla 3. Resumen de parámetros D&H del brazo D

d	θ	a	α
$-$	$90°$	$-$	α_1
d_2	$-$	$-$	α_2
$-$	θ_3	$-$	$-$

El movimiento y sentido de movimiento de las articulaciones del brazo D descritos por las variables α_1, α_2, d_2 y θ_3 se muestra en la siguiente figura:

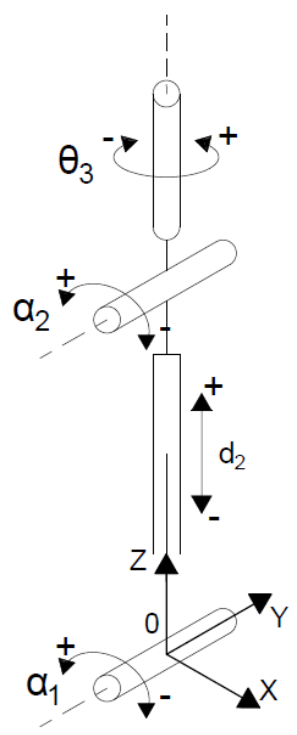


Figura 27. Brazo D, movimientos y sentido de movimientos de sus articulaciones

Ecuaciones de posición

A partir de los resultados en la tabla anterior (Tabla 3), se determina las ecuaciones de posición del extremo del brazo D. Para esto se expresan dichos resultados en forma de matriz homogénea y a partir de ésta se determina las ecuaciones que se requieren.

$$[D] = \begin{bmatrix} D_x \\ D_y \\ D_z \\ 1 \end{bmatrix} = [M_D_{4\times4}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.42}$$

Donde, $[D]$ corresponde a las componentes x, y, z del vector posición del brazo D, $[M_D_{4\times4}]$ es la matriz homogénea determinada a partir de los parámetros de la Tabla 3, ésta a su vez se encuentra multiplicada por el vector posición del origen de coordenadas.

Como resultado de la ecuación anterior se obtiene:

$$D_x = D_1 \sin(D_2) \tag{2.43}$$

$$D_y = 0 \tag{2.44}$$

$$D_z = D_1 \cos(D_2) \tag{2.45}$$

Donde, D_1 y D_2 corresponden a las variables de las articulaciones del brazo y equivalen a los valores de α_1 y d_2, hallados por el método D&H, como se muestra en la siguiente figura:

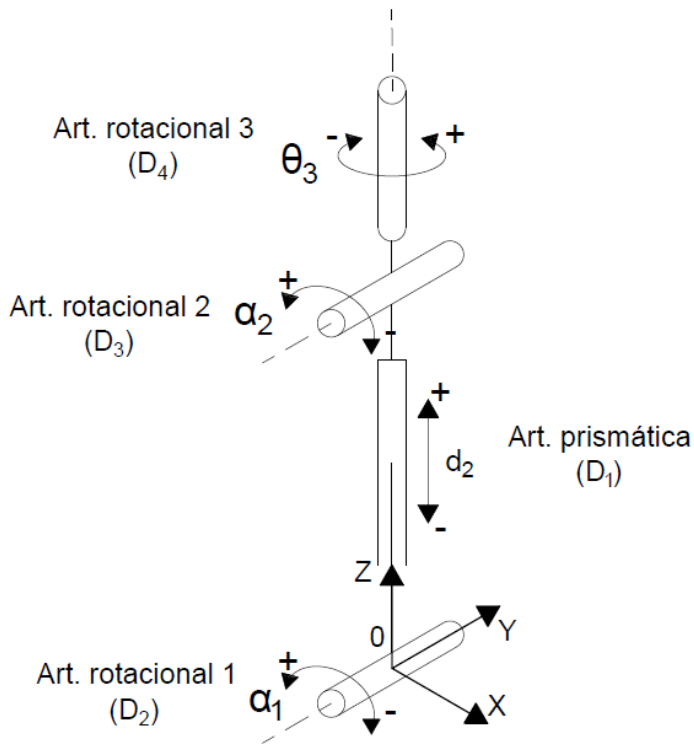


Figura 28. Equivalencia entre variables de ecuaciones y parámetros obtenidos por el método D&H en el brazo RPU

Como se puede observar, a diferencia de los brazos UPE, en el brazo RPU las ecuaciones de cinemática directa desarrolladas por el método D&H son insuficientes para evaluar a todas las articulaciones de dicho brazo, ya que sólo son necesarias 2 de las 4 articulaciones para definir la posición del extremo del brazo D.

Para analizar completamente el brazo RPU, se emplea ecuaciones de cinemática directa junto a relaciones geométricas, de forma que los 4 grados de libertad del robot (x, z, θ, ϕ) también pasan a formar parte de las ecuaciones; definiendo las ecuaciones de movimiento del brazo de la siguiente manera:

$$x = D_1 \sin(D_2) \tag{2.46}$$

$$z = D_1 \cos(D_2) \tag{2.47}$$

$$\phi = D_2 + D_3 \tag{2.48}$$

$$\theta = D_4 \tag{2.49}$$

Esta inclusión de los valores x, z, θ, ϕ, es posible gracias a que el extremo superior del

brazo D está directamente conectado al centro de la plataforma móvil (0_m).

Ecuaciones de velocidad

Con las ecuaciones de posición para el brazo D definidas, se procede a derivar a cada una de ellas para las variables involucradas en éstas.

$$\frac{dx}{dt} = \frac{\partial x}{\partial D_1}\frac{\partial D_1}{\partial t} + \frac{\partial x}{\partial D_2}\frac{\partial D_2}{\partial t} \tag{2.50}$$

$$\frac{dz}{dt} = \frac{\partial z}{\partial D_1}\frac{\partial D_1}{\partial t} + \frac{\partial z}{\partial D_2}\frac{\partial D_2}{\partial t} \tag{2.51}$$

$$\frac{d\phi}{dt} = \frac{\partial \phi}{\partial D_2}\frac{\partial D_2}{\partial t} + \frac{\partial \phi}{\partial D_3}\frac{\partial D_3}{\partial t} \tag{2.52}$$

$$\frac{d\theta}{dt} = \frac{\partial \theta}{\partial D_4}\frac{\partial D_4}{\partial t} \tag{2.53}$$

Al escribir las ecuaciones resultantes en forma matricial, se obtiene:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dz}{dt} \\ \frac{d\phi}{dt} \\ \frac{d\theta}{dt} \end{bmatrix} = \begin{bmatrix} \sin(D_2) & D_1\cos(D_2) & 0 & 0 \\ \cos(D_2) & -D_1\sin(D_2) & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{D}_1 \\ \dot{D}_2 \\ \dot{D}_3 \\ \dot{D}_4 \end{bmatrix} \tag{2.54}$$

Ecuación matricial que relaciona la velocidad de las articulaciones del brazo D con los valores de velocidad de la plataforma móvil.

Ecuaciones de aceleración

Para definir las ecuaciones de aceleración se procede de la misma a calcular las derivadas de las ecuaciones de velocidad y posición del brazo D, de acuerdo a las siguientes formas:

$$\frac{d^2x}{dt^2} = \frac{\partial x}{\partial D_1}\frac{\partial^2 D_1}{\partial t^2} + \frac{\partial x}{\partial D_2}\frac{\partial^2 D_2}{\partial t^2} + \frac{\partial^2 x}{\partial t \partial D_1}\frac{\partial D_1}{\partial t} + \frac{\partial^2 x}{\partial t \partial D_2}\frac{\partial D_2}{\partial t} \tag{2.55}$$

$$\frac{d^2z}{dt^2} = \frac{\partial z}{\partial D_1}\frac{\partial^2 D_1}{\partial t^2} + \frac{\partial z}{\partial D_2}\frac{\partial^2 D_2}{\partial t^2} + \frac{\partial^2 z}{\partial t \partial D_1}\frac{\partial D_1}{\partial t} + \frac{\partial^2 z}{\partial t \partial D_2}\frac{\partial D_2}{\partial t} \tag{2.56}$$

$$\frac{d^2\phi}{dt^2} = \frac{\partial \phi}{\partial D_2}\frac{\partial^2 D_2}{\partial t^2} + \frac{\partial \phi}{\partial D_3}\frac{\partial^2 D_3}{\partial t^2} + \frac{\partial^2 \phi}{\partial t \partial D_2}\frac{\partial D_2}{\partial t} + \frac{\partial^2 \phi}{\partial t \partial D_3}\frac{\partial D_3}{\partial t} \tag{2.57}$$

$$\frac{d^2\theta}{dt^2} = \frac{\partial \theta}{\partial D_4}\frac{\partial^2 D_4}{\partial t^2} + \frac{\partial^2 \theta}{\partial t \partial D_4}\frac{\partial D_4}{\partial t} \tag{2.58}$$

Al escribir las ecuaciones resultantes en forma matricial, se obtiene:

$$\begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2z}{dt^2} \\ \frac{d^2\phi}{dt^2} \\ \frac{d^2\theta}{dt^2} \\ 1 \end{bmatrix} = [MaD] \begin{bmatrix} \ddot{D}_1 \\ \ddot{D}_2 \\ \ddot{D}_3 \\ \ddot{D}_4 \\ 1 \end{bmatrix} \tag{2.59}$$

Donde, $[MaD]$ corresponde a la matriz de aceleración del brazo D desarrollada a continuación:

$$[MaD] = \begin{bmatrix} \sin(D_2) & D_1\cos(D_2) & 0 & 0 & 2\cos(D_2)\,\dot{D}_1\dot{D}_2 - D_1\sin(D_2)\,\dot{D}_2^2 \\ \cos(D_2) & -D_1\sin(D_2) & 0 & 0 & -2\sin(D_2)\,\dot{D}_1\dot{D}_2 - D_1\cos(D_2)\,\dot{D}_2^2 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

La ecuación matricial presentada (2.59) relaciona la aceleración de las articulaciones del brazo D con los parámetros de aceleración de la plataforma móvil.

2.3. Vinculación de modelos matemáticos del robot

Se conoce que, para encontrar la posición, velocidad y aceleración de las articulaciones de los brazos que describen el movimiento de la plataforma del robot paralelo, a través del empleo de cinemática inversa el cálculo se convierte en un proceso más directo y sencillo (Patel & George, 2012; Rastegarpanah, Saadat, & Borboni, 2016), para proceder por este métdos, se debe unificar los modelos matemáticos de cada elemento del robot y definir las relaciones entre ellos.

El modelo matemático de la plataforma, parte de las variables de entrada correspondientes a los grados de libertad del robot (x, z, θ, ϕ), calculando las posiciones de los vértices de ésta. Así se obtiene resultados intermedios, los cuales son utilizados como entradas para el siguiente paso de cálculo.

Las ecuaciones de posición de cada brazo calculan los valores correspondientes a las posiciones de las articulaciones a través de resolución algebraica, con los valores de posición del extremo del brazo hallados en el paso anterior.

En el caso del cálculo de las velocidades y aceleraciones del robot, los valores de entrada para estas magnitudes deben ser proporcionados de acuerdo a los requerimientos de

movimiento de la plataforma.

Con estos valores ingresados, se aprovecha la estructura matricial de las ecuaciones para calcular más rápidamente los valores correspondientes a las articulaciones, este cálculo puede ser ejecutado en cualquier software para resolución de problemas matemáticos.

El algoritmo de cálculo es resumido en el siguiente diagrama:

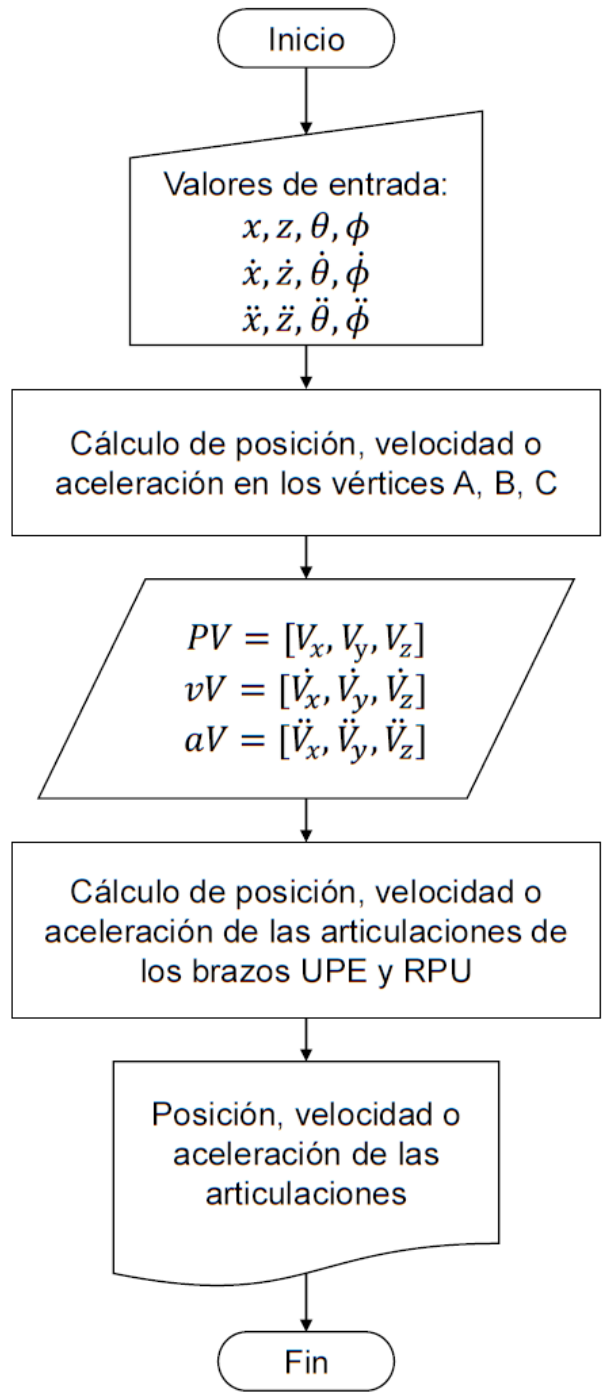


Figura 29. Algoritmo de cálculo para modelo matemático del robot paralelo

3. CÁLCULO Y SIMULACIÓN DE RESULTADOS

Como se explicó en el capítulo anterior, la resolución de las ecuaciones puede ser realizada en un software matemático a través de operaciones entre vectores y matrices, por lo cual, para el cálculo de las ecuaciones de este modelo se seleccionó el software MatLab®.

Este software es el encargado de hallar una solución válida al sistema de ecuaciones del robot y validar estos resultados, como método de validación se emplea una simulación desarrollada en el mismo entorno de MatLab®.

La simulación debe presentar un esquema tridimensional de la posición del robot, reproducir movimientos y trazar gráficas que relacionen la posición, velocidad y aceleración de las articulaciones con el movimiento realizado por la plataforma.

3.1. MatLab®

El software MatLab®, desarrollado por MathWorks Inc., es una plataforma de cálculo y simulación a través de métodos numéricos y matemática simbólica.

Este programa trabaja con un sistema de archivos llamados script, en los cuales se almacena el código de programación donde se encuentran las ecuaciones y además se detalla las instrucciones para la resolución de las mismas.

Entre sus ventajas se encuentra la capacidad de trabajo con estructuras matemáticas como vectores y matrices, permite el uso de condiciones lógicas y graficación, características necesarias para la realización de este proyecto.

3.2. Estructura del programa

Para cumplir con los requerimientos de simulación, el programa debe ser capaz de calcular los valores de posición del robot de forma rápida y precisa, y a la vez calcular los valores de velocidad y aceleración. El proceso de cálculo a su vez presenta dos casos: en reposo, donde solo se calcula la posición de las articulaciones del robot; y el caso en movimiento, donde se realiza tanto los cálculos de posición, como los de velocidad y aceleración.

En el caso en movimiento, se emplea los resultados de las ecuaciones de posición para resolver las ecuaciones de velocidad y aceleración. En esta sección se debe resaltar que los cálculos se realizan para un movimiento por lo que se requiere el ingreso de condiciones (posición, velocidad y aceleración de la plataforma) iniciales y finales del movimiento.

De esta forma, se entiende que existen 2 procesos principales dentro de la resolución de las ecuaciones: primero, la resolución del sistema de ecuaciones de posición de forma algebraica a través de métodos numéricos y la segunda, la resolución de las ecuaciones matriciales para hallar los valores de velocidad y aceleración.

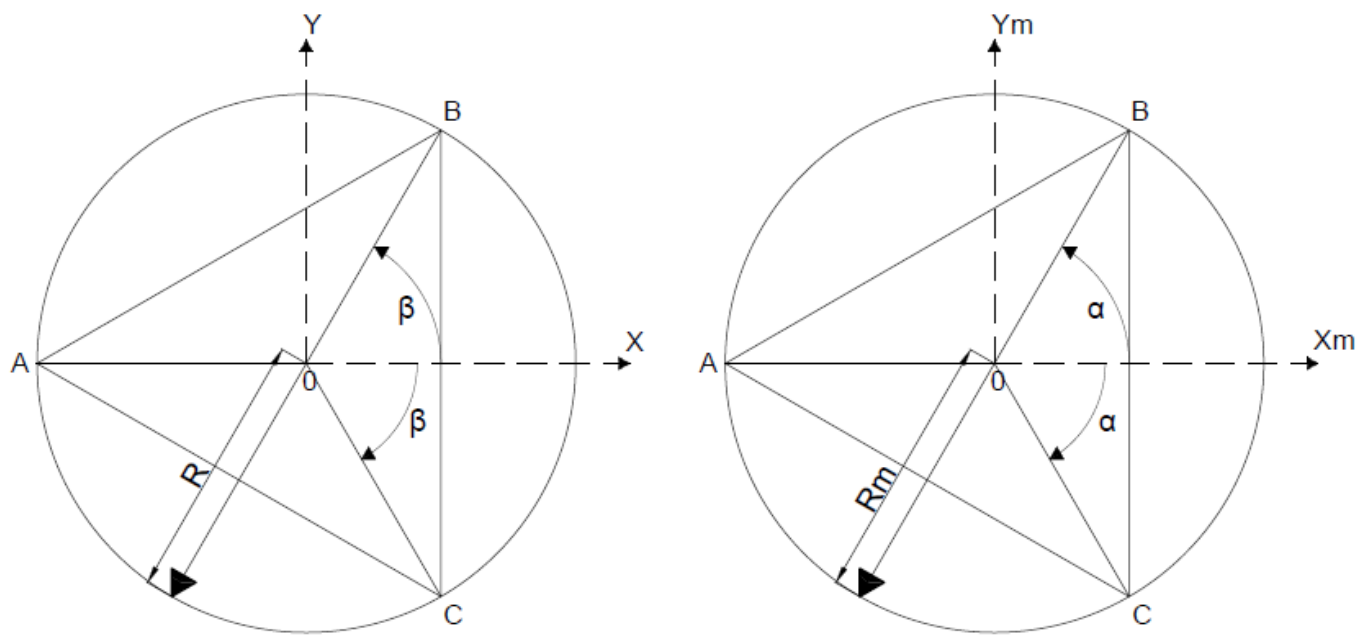
Para la simulación del caso en reposo, se presenta un esquema elaborado a través de líneas y puntos, este esquema muestra la posición de los brazos, el perímetro de la base y el perímetro de la plataforma móvil del robot, permitiendo una visualización sencilla de los resultados obtenidos de las ecuaciones de posición.

Para la simulación del caso en movimiento, se muestra un esquema animado del robot formado por líneas y puntos, a esta simulación se la complementa a través de gráficas que muestran la relación entre los valores de posición, velocidad y aceleración de las articulaciones y la posición de la plataforma.

Variables de entrada

La diferencia entre la resolución de las condiciones estáticas y dinámicas de la plataforma, se da en la cantidad de datos que se necesita ingresar. Para el primer caso se ingresa únicamente los valores x, z, θ, ϕ para una posición única; mientras que para el segundo caso estos valores se convierten en condiciones iniciales y condiciones finales, delimitando la posición inicial y final del movimiento. Para el segundo caso también es necesario incluir las condiciones inicial y final de la velocidad y aceleración de la plataforma.

Otras variables presentes pueden ser clasificadas como constantes y corresponden a las dimensiones de la base y la plataforma móvil (R, β, R_m, α).



b) Variables de la base c) Variables de la plataforma

Figura 30. Variables de entrada correspondientes a la base (a) y la plataforma móvil (b)

Los valores de estas variables, fueron determinados en el trabajo realizado por las ingenieras Fernández Y., Lorena A. y Sotomayor R., Luisa F.; en base a las proporciones geométricas del miembro inferior. Estas variables no son editables desde la ventana de la interfaz gráfica y se las presenta como variables de “solo lectura”.

Adicionalmente, existen dos variables internas que controla la apreciación de cálculo, a éstas sólo se puede acceder desde el script, intervienen en el cálculo del caso en movimiento y permiten controlar cuanto será el paso mínimo de desplazamiento (lineal o angular) que tendrá la plataforma entre cada iteración de cálculo.

Interfaz gráfica

El programa cuenta con una interfaz gráfica (Figura 31) para facilidad de trabajo, aquí se ingresa los datos requeridos según el caso (en reposo o en movimiento) en las secciones de posición, velocidad y aceleración.

Muestra los botones de cálculo (Calcular) y reinicio de variables (Reiniciar), los cuales se encuentran activos por defecto. Los botones “Simular” y “Gráficas” se encuentran desactivados por defecto y sólo se activan cuando el cálculo ya ha concluido.

El botón “Simular”, al estar el caso “Reposo” seleccionado, muestra un esquema tridimensional de la posición del robot paralelo según las condiciones ingresadas; para el caso en movimiento, muestra una animación tridimensional del robot.

El botón “Gráficas”, se activa únicamente al estar el caso en movimiento seleccionado, al presionarlo éste muestra un menú donde se selecciona el sentido de movimiento simulado (Figura 32), seguido por otro menú en el que se especifica la magnitud (posición, velocidad o aceleración). Figura 33.

La ventana de la interfaz gráfica también posee un espacio incorporado en la zona inferior derecha de la ventana, el cual está dedicado a graficación, es en éste espacio donde se dibuja el esquema y la animación generada al seleccionar el botón “Simular”.

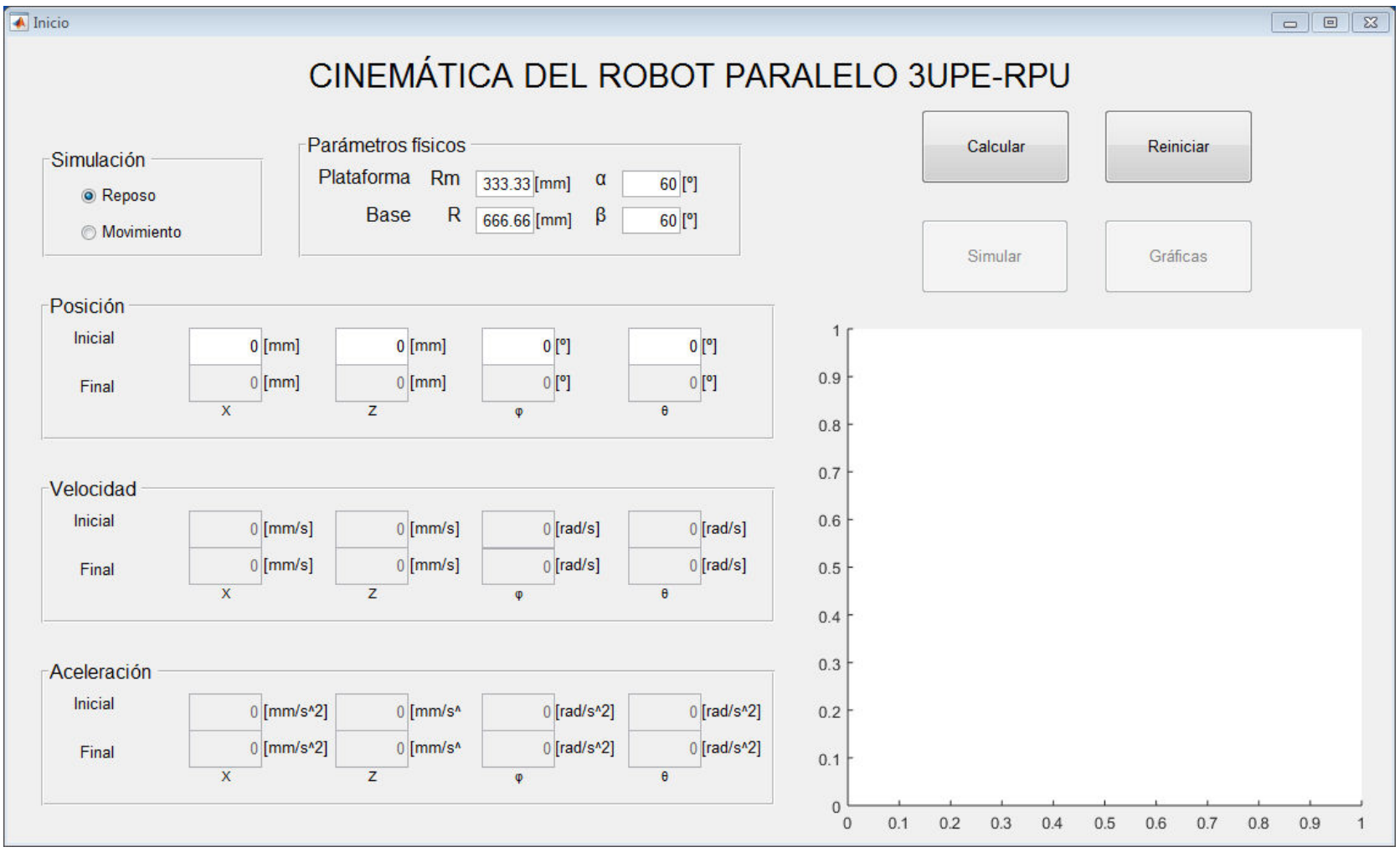


Figura 31. Interfaz gráfica (GUI) del programa.

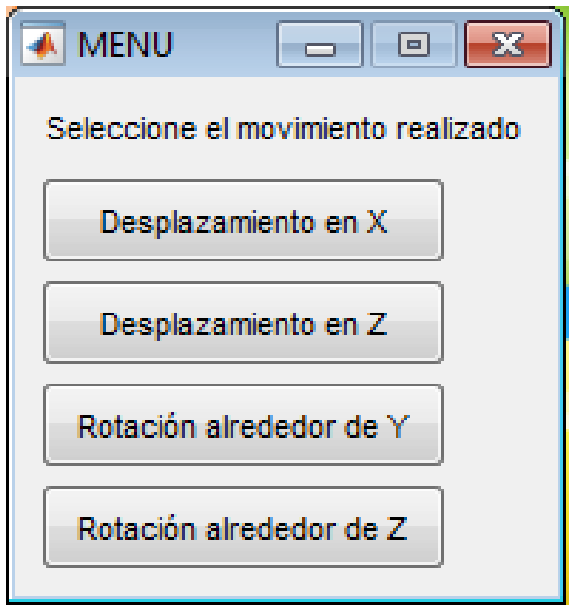


Figura 32. Menú de selección de movimiento

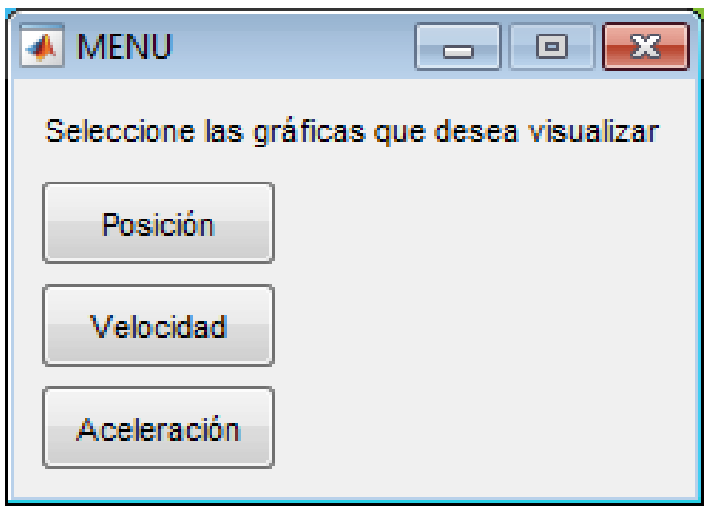


Figura 33. Menú de selección de magnitudes de las gráficas

Algoritmo de cálculo

El programa calcula los resultados del caso en reposo por separado de los del caso en movimiento, dando lugar a dos operaciones completamente diferenciadas. Figura 34

En primer lugar, en el caso en reposo se da lectura a las variables de entrada correspondientes, estas variables son almacenadas y a partir de estás se calculan los vectores que definen la posición de los vértices de la plataforma en el espacio. Con los vértices definidos, se relaciona cada vértice con el extremo del brazo UPE correspondiente para determinar las posiciones de las articulaciones de cada uno.

El brazo central RPU, también se lo calcula de forma paralela, en este caso el centro de la plataforma es relacionado con el extremo del brazo.

Una vez hallados los valores, estos resultados son nuevamente reemplazados en las ecuaciones del brazo correspondientes y la solución de esta sustitución es comparada con las posiciones de los vértices que se definió anteriormente, así se consigue determinar el error relativo entre la posición del final del vértice de la plataforma y el extremo de cada brazo.

Como resultado del procesamiento realizado en el caso en reposo, se obtiene un esquema tridimensional en el que se visualiza las posiciones que adoptan los brazos del robot para posicionar a la plataforma en el espacio de acuerdo a las variables de entrada especificadas.

Para el caso en movimiento, las variables de entrada correspondientes a las condiciones iniciales y finales de posición, velocidad y aceleración son leídas y almacenadas; el almacenamiento de éstas se realiza en forma de vectores cuyo tamaño está en función del número de iteraciones y se determina al dividir el desplazamiento lineal (x, z) y angular (θ, ϕ) de la plataforma, para el valor de la apreciación de 5 [mm] o 1[º] respectivamente; estos valores son comparados entre sí y se elige al mayor de ellos, este número será el que determine tanto el número de elementos de cada vector, como el número de iteraciones a realizarse. Los valores de apreciación mostrados se encuentran definidos por defecto y sólo pueden ser modificados desde el código del programa.

Con el número de iteraciones definido y los vectores correspondientes a posición, velocidad y aceleración listos, se procede al cálculo de la posición de los vértices, continuando con el cálculo de la posición de las articulaciones de los brazos, este proceso se lo realiza n veces, de acuerdo al número de iteraciones determinado.

Los productos de este proceso son los valores de posición de las articulaciones, que son usados para la creación de la animación del robot; y las gráficas que relacionan la posición de las articulaciones del robot, con la posición de la plataforma.

Definidas las posiciones, se procede de la misma manera para la determinación de las velocidades y aceleraciones, primero se realiza el cálculo de la velocidad y aceleración de los vértices, para éstos a su vez ser usados en el cálculo de la velocidad o aceleración de las articulaciones de los brazos, según corresponda. De la misma forma, se obtiene como resultado las gráficas de relación entre la velocidad o aceleración de las articulaciones y el desplazamiento de la plataforma.

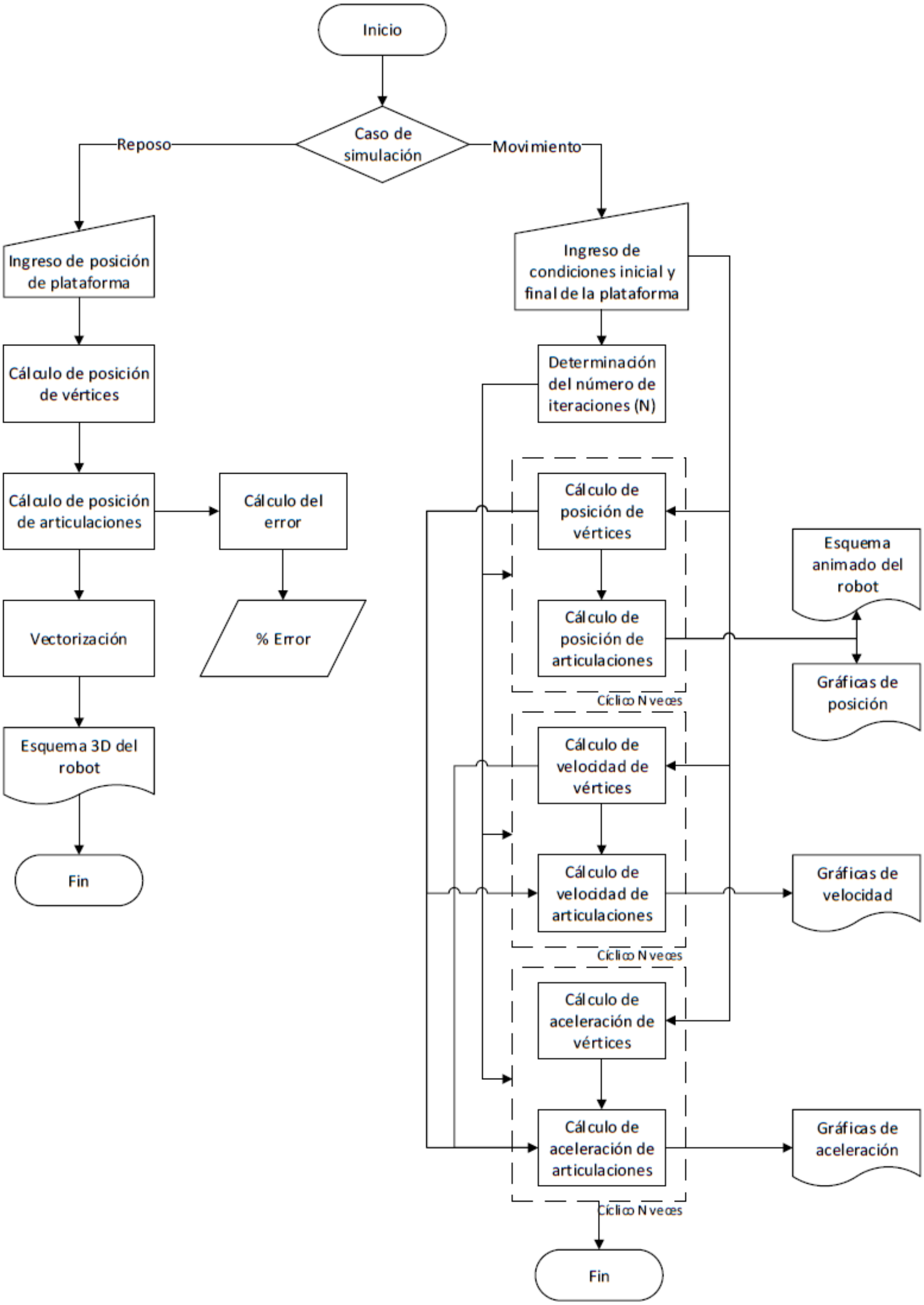


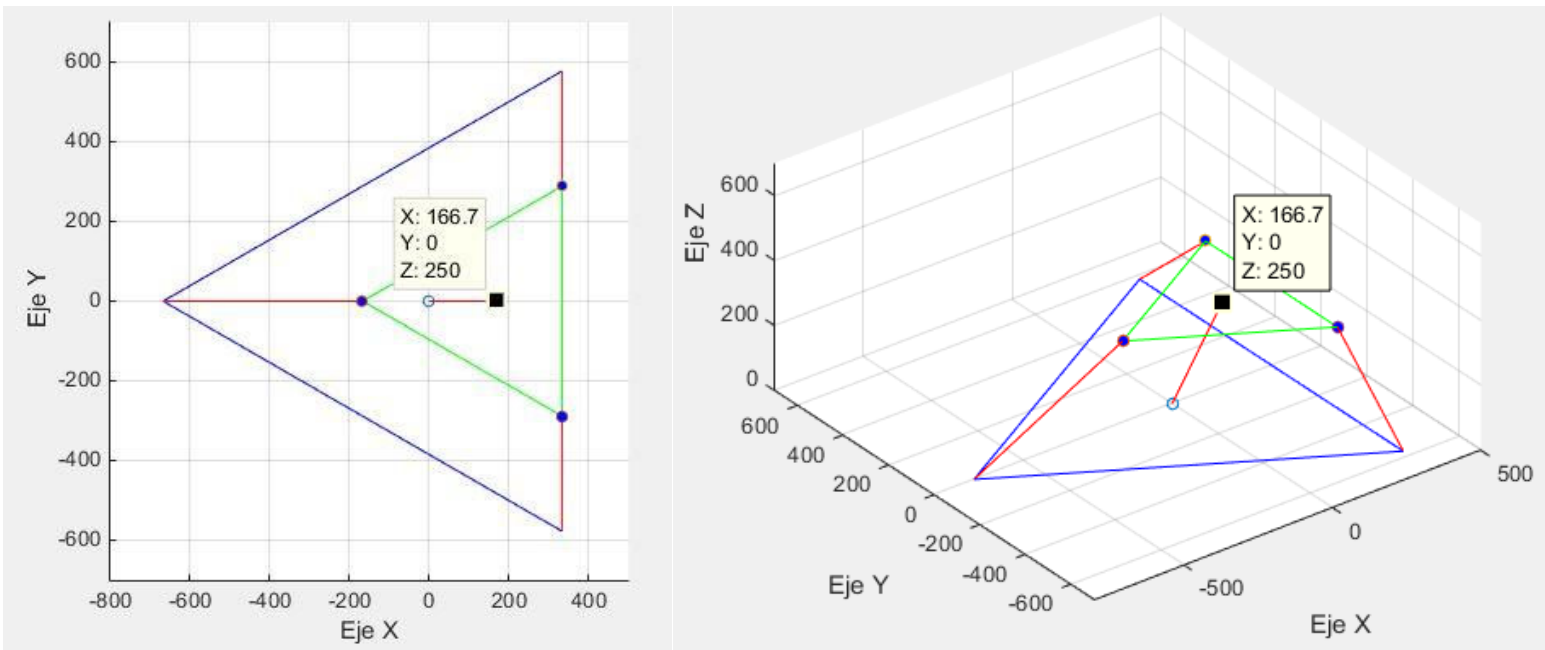
Figura 34. Diagrama de flujo del programa

3.3. Ejemplo de cálculo

Para validar los resultados calculados por el programa, se realiza un ejemplo de cálculo para mostrar el funcionamiento del programa en el caso en reposo y otro para el caso en movimiento.

Ejemplo de cálculo para el caso en reposo

En este ejemplo se considera una posición en la que la plataforma se encuentre ubicada de tal forma que la arista ubicada entre los vértices B y C, se encuentre justo por encima de la arista homóloga de la base, a una altura de 250 [mm] (Figura 35). Esta posición está definida por los siguientes valores:



a) Vista vertical b) Vista isométrica

Figura 35. Vista vertical (a) e isométrica (b) del robot paralelo con una de sus aristas de la plataforma y base alineadas

Tabla 4. Valores de entrada para ejemplo del caso en reposo

Variable	Valor
x	$166.7\ mm$
z	$250\ mm$
ϕ	$0°$
θ	$0°$

Los valores resultantes en este caso, se los presenta a continuación:

Tabla 5. Resultados de posición de las articulaciones de cada brazo del robot

Brazos UPE			Brazo RPU	Articulación
A	B	C	D	
559.0	381.9	381.9	300.5	Prismática [mm]
-63.4	-45.0	-45.0	33.7	1ra rotacional [º]
0	-22.2	22.2	-33.7	2da rotacional [º]
-	-	-	0	3ra rotacional [º]

Los resultados presentados en la tabla anterior obedecen a la numeración y sentidos de movimiento de los brazos UPE y RPU, mostrados en las figuras 21 y 28 respectivamente.

Para validar los resultados presentados en la Tabla 5, a continuación se calcula la diferencia existente entre los valores de posición de cada vértice de la plataforma resultantes de las ecuaciones de cinemática directa, y la posición del extremo de cada brazo que resulta del reemplazo de los valores presentados en la tabla anterior. Esta diferencia se la presenta en la siguiente tabla:

Tabla 6. Diferencia entre la posición del vértice de la plataforma y el extremo del brazo

Vértice	Dirección	Posición del vértice (Pv)	Extremo del brazo (EB)	Diferencia $(d = Pv - EB)$	Error relativo $(e = \lvert d/Pv \rvert * 100\%\ )$
A	X	-666.63	-666.63	0	0
	Y	0	0	0	0
	Z	250.0	250.0	0	0
B	X	-166.635	-166.635	0	0
	Y	288.6722	288.6723	-0.3052 e-4	0.1057 e-4 %
	Z	250.0	250.0	-0.1526 e-4	0.061 e-4 %
C	X	-166.635	-166.635	0	0
	Y	-288.6722	-288.6723	0.3052 e-4	0.1057 e-4 %
	Z	250.0	250	-0.1526 e-4	0.061 e-4 %

Los valores de posición de los vértices, extremo del brazo y diferencia se encuentran

expresados en milímetros $[mm]$.

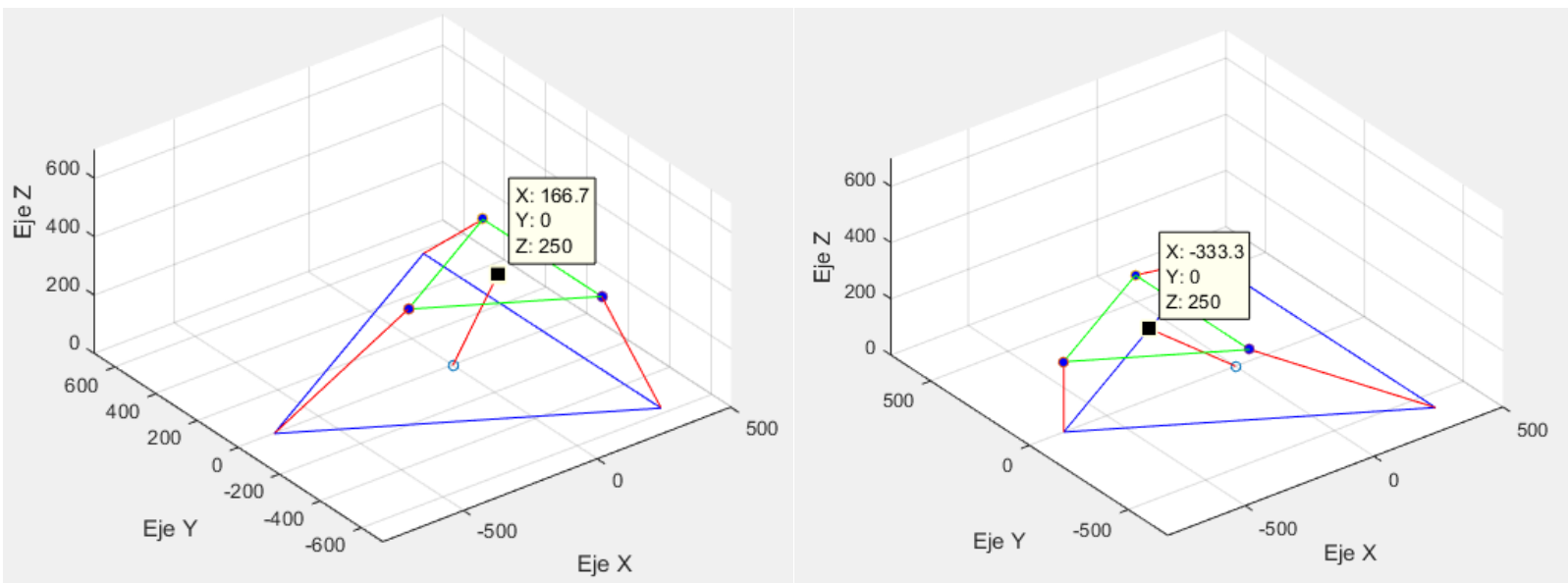
Ejemplo de cálculo para el caso en movimiento

Para este caso se emplea la posición definida en el ejemplo anterior como punto de partida, la plataforma realizará un desplazamiento horizontal, a la misma altura, hasta que su vértice A se sitúe justo por encima del vértice A de la base. Para este caso las condiciones iniciales y finales de movimiento se muestran en la siguiente tabla:

Tabla 7. Valores de entrada para ejemplo del caso en movimiento

Posición				Velocidad				Aceleración			
Condición inicial		Condición final		Condición inicial		Condición final		Condición inicial		Condición final	
Var.	Valor	Var.	Valor	Var.	Valor	Var.	Valor	Var.	Valor	Var.	Valor
x_i	$166.7\ mm$	x_f	$-333\ mm$	$\dot{x}_i$	$10\frac{mm}{s}$	$\dot{x}_f$	$10\frac{mm}{s}$	$\ddot{x}_i$	$0\frac{mm}{s^2}$	$\ddot{x}_f$	$0\frac{mm}{s^2}$
z_i	$250\ mm$	z_f	$250\ mm$	$\dot{z}_i$	$0\frac{mm}{s}$	$\dot{z}_f$	$0\frac{mm}{s}$	$\ddot{z}_i$	$0\frac{mm}{s^2}$	$\ddot{z}_f$	$0\frac{mm}{s^2}$
ϕ_i	$0°$	ϕ_f	$0°$	$\dot{\phi}_i$	$0\frac{rad}{s}$	$\dot{\phi}_f$	$0\frac{rad}{s}$	$\ddot{\phi}_i$	$0\frac{rad}{s^2}$	$\ddot{\phi}_f$	$0\frac{rad}{s^2}$
θ_i	$0°$	θ_f	$0°$	$\dot{\theta}_i$	$0\frac{rad}{s}$	$\dot{\theta}_f$	$0\frac{rad}{s}$	$\ddot{\theta}_i$	$0\frac{rad}{s^2}$	$\ddot{\theta}_f$	$0\frac{rad}{s^2}$

En la Figura 36, se puede observar el esquema del robot en las posiciones inicial y final, definidas para este ejemplo. Otras consideraciones son: que el movimiento tiene velocidad constante, sin aceleración; los grados de libertad restantes (z, θ, ϕ) no presentan movimiento, por lo que los valores de velocidad y aceleración asociados a ellos, se mantienen en cero.



a) Posición inicial de la plataforma b) Posición final de la plataforma

Figura 36. Posición inicial (a) y final (b) del ejemplo de movimiento de la plataforma.

En este caso, los resultados obtenidos son la animación y las gráficas de relación entre el movimiento de la plataforma y las variables de las articulaciones. A continuación, se presenta las gráficas correspondientes a posición, velocidad y aceleración de las articulaciones.

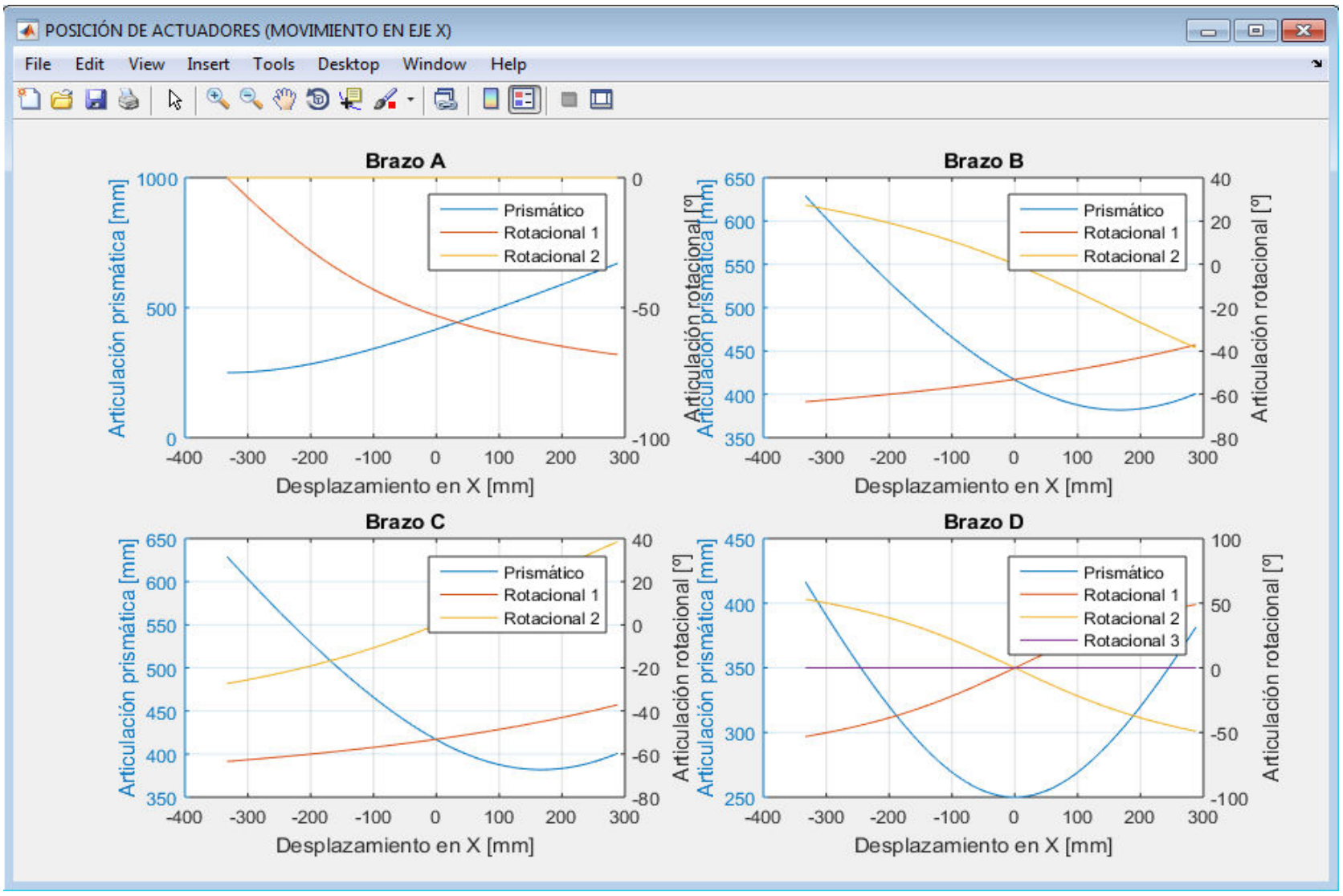


Figura 37. Gráficas de posición de las articulaciones

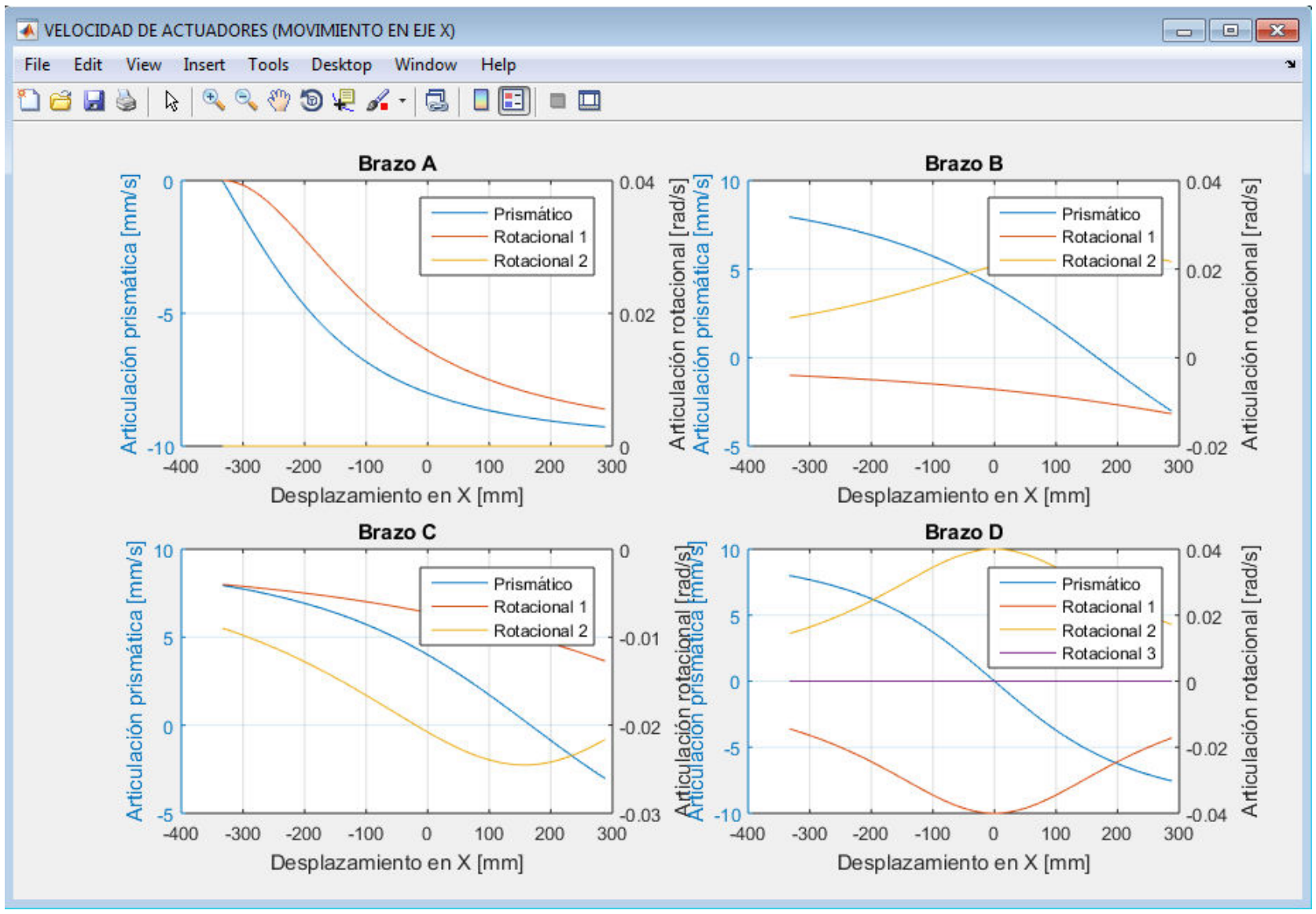


Figura 38. Gráficas de velocidad de las articulaciones

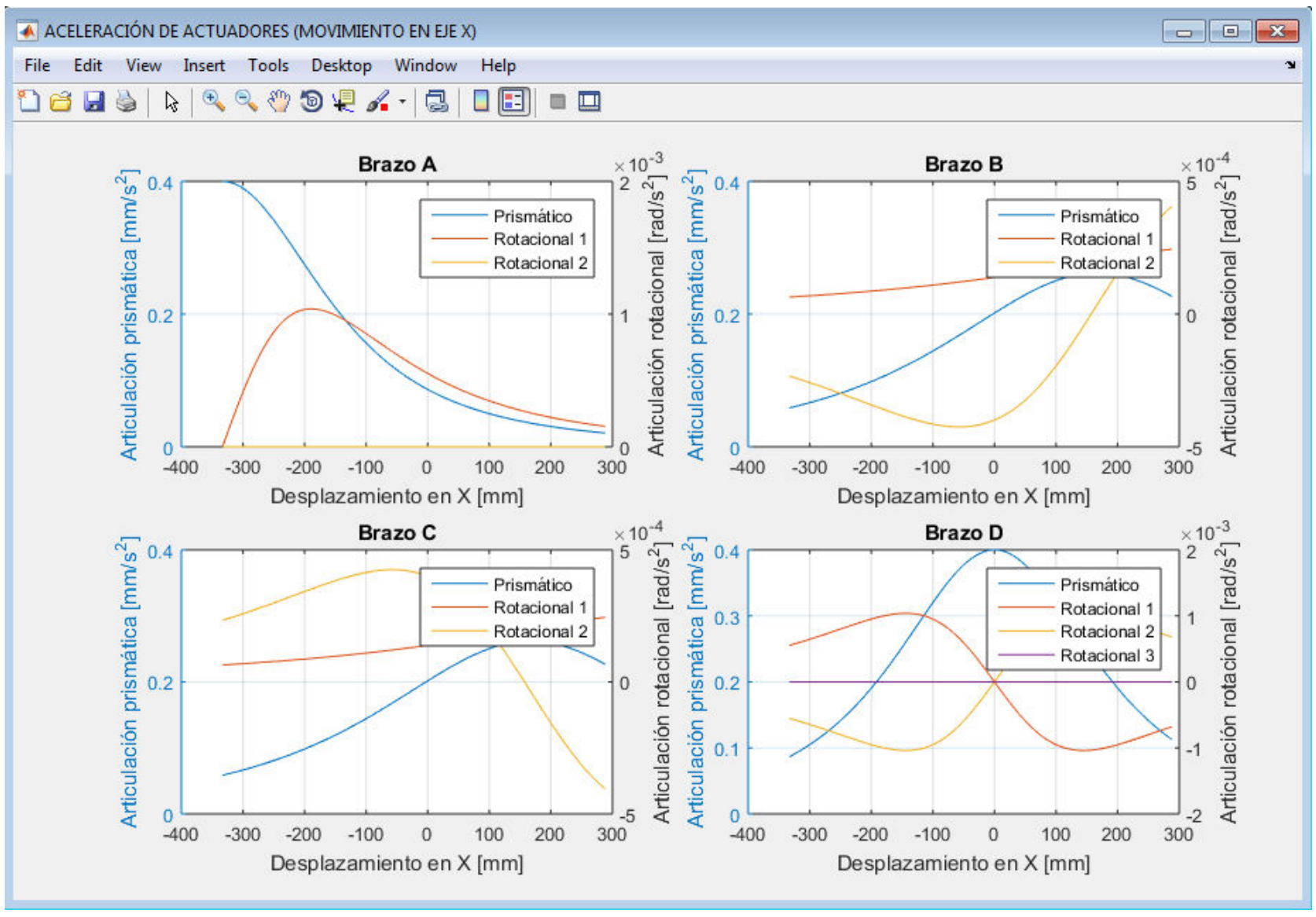


Figura 39. Gráficas de aceleración de las articulaciones

4. ANÁLISIS DE RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

4.1. Análisis de resultados

Tanto para la simulación del caso en reposo como del caso en movimiento, se muestra que las posiciones que toman los brazos son correctas, ya que los extremos de los brazos se sitúan en donde los vértices de la plataforma deben estar ubicados.

Para el segundo caso, el movimiento especificado se realiza con desplazamiento suave y continuo, sin irregularidades, esto puede notarse de mejor manera a través de las gráficas de relación (Figura 37, 38 y 39). En estas gráficas se observa que estas no presentan discontinuidades o puntos de inflexión con cambios pronunciados.

Para un análisis más profundo de los resultados, a través del código del programa se puede acceder a los valores exactos de los parámetros graficados en el esquema en reposo, la animación y las gráficas de relación.

4.2. Conclusiones

- El análisis cinemático presentado en el presente trabajo, satisface las necesidades planteadas al inicio del proyecto, ya que el modelo matemático es capaz de describir el movimiento del robot, determinar la posición que adoptan las articulaciones, a la vez que calculan las velocidades y aceleraciones asociadas al realizar dichos movimientos; estos resultados son calculados y simulados a través de un software matemático.
- El modelo cinemático de los brazos del robot paralelo elaborado a través del método de Denavit & Hartenberg, priorizó una rápida velocidad de cálculo a través de software matemático, junto con un error relativo bastante bajo.
- Al determinar el modelo matemático del movimiento de la plataforma por separado, se asegura que no existan movimientos fuera de los cuatro grados de libertad establecidos.
- Los modelos cinemáticos de los brazos UPE y RPU pueden ser usados para calcular de forma directa la posición de su extremo, dadas las posiciones de sus articulaciones; al igual que de forma inversa para calcular las posiciones de sus articulaciones, dada la posición de su extremo.
- Al estudiar el robot como un solo sistema se observa que cualquier elemento,

excluyendo la base, puede ser tomado como punto de partida para el análisis cinemático, pero al hacer esto se provoca que los elementos restantes pasen a ser dependientes del primero. Esto se traduce a un empleo de modelos cinemáticos directo e indirecto para el elemento principal y sus dependientes, respectivamente.

- Las ecuaciones de posición del robot son validadas al calcular el error existente entre las posiciones de los vértices determinadas por el modelo de la plataforma y la posición del extremo del brazo determinada al reemplazar las posiciones de las articulaciones en su modelo correspondiente. El error obtenido se encuentra por debajo del 0.001%.
- El esquema y la animación presentadas como resultado, son construidas en base a los valores obtenidos del reemplazo de la posición de las articulaciones calculada en las ecuaciones correspondientes a cada brazo, de tal forma que, si se presentase cualquier anomalía en los resultados, tanto el esquema como la animación lo mostrarían fácilmente.

4.3. Recomendaciones

- Se recomienda realizar un análisis adicional, en el que se plantee la ubicación del brazo central del robot entre un punto ubicado en el área de contacto entre talón del pie y la plataforma móvil, y su correspondiente proyección sobre la base del robot.
- Para mejorar los resultados y la animación, se recomienda emplear como datos de entrada los resultados de ecuaciones de trayectoria que describan el movimiento de la plataforma en el espacio.
- En base a los resultados del análisis cinemático del brazo RPU, se recomienda estudiar la factibilidad, ventajas y desventajas de ubicar el control de orientación e inclinación de la plataforma en la articulación universal de este brazo.
- Se recomienda estudiar la capacidad de implementar el modelo matemático hallado en el presente estudio, con software CAD (Computer Aided Design) a través de la extensión de Simulink® de MatLab® para estudiar el fenómeno en movimiento al que están sometidos los cuerpos que conforman el robot.

BIBLIOGRAFÍA

Barrientos, A., Peñin, L. F., Balaguer, C., & Aracil, R. (1996). *Fundamentos de Robótica* (2nd Ed.). Madrid: McGraw-Hill.

Brits, L. (2008). Eulerangles.svg. Retrieved February 2, 2017, from https://en.wikipedia.org/wiki/File:Eulerangles.svg

Fernández, A., & Sotomayor, L. (2016). *Análisis cinemático inverso y directo del robot paralelo*. Quito: EPN.

Ollero, A. (2001). *ROBÓTICA Manipuladores y robots móviles*. BOIXAREU Editores.

Patel, Y. D., & George, P. M. (2012). Parallel Manipulators Applications—A Survey. *Modern Mechanical Engineering*, *2*(3), 57–64. https://doi.org/10.4236/mme.2012.23008

Rastegarpanah, A., Saadat, M., & Borboni, A. (2016). Parallel Robot for Lower Limb Rehabilitation Exercises. *Applied Bionics and Biomechanics*, *2016*, 1–10. https://doi.org/10.1155/2016/8584735

Saha, S. K. (2010). *Introduccion a la Robótica* (1ra Edición). McGraw-Hill.

The MathWorks Inc. (n.d.). Euler Angles. Retrieved February 2, 2017, from https://www.mathworks.com/discovery/euler-angles.html

ANEXOS

ANEXO 1

Programación en MatLab®

```
%%%%%%%%%%%%%CÓDIGO DE GENERACIÓN AUTOMATICA PARA GUI%%%%%%%%%%%%%

function varargout = Inicio(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Inicio_OpeningFcn, ...
                   'gui_OutputFcn',  @Inicio_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Inicio_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = Inicio_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
global alfa beta R Rm
x0=str2num(get(handles.edit1,'String'));
z0=str2num(get(handles.edit2,'String'));
f0=str2num(get(handles.edit3,'String'));
t0=str2num(get(handles.edit4,'String'));
x1=str2num(get(handles.edit5,'String'));
z1=str2num(get(handles.edit6,'String'));
f1=str2num(get(handles.edit7,'String'));
t1=str2num(get(handles.edit8,'String'));
Vx0=str2num(get(handles.edit9,'String'));
Vz0=str2num(get(handles.edit10,'String'));
Wf0=str2num(get(handles.edit11,'String'));
Wt0=str2num(get(handles.edit12,'String'));
Vx1=str2num(get(handles.edit13,'String'));
Vz1=str2num(get(handles.edit14,'String'));
Wf1=str2num(get(handles.edit15,'String'));
Wt1=str2num(get(handles.edit16,'String'));
Ax0=str2num(get(handles.edit17,'String'));
Az0=str2num(get(handles.edit18,'String'));
af0=str2num(get(handles.edit19,'String'));
at0=str2num(get(handles.edit20,'String'));
Ax1=str2num(get(handles.edit21,'String'));
Az1=str2num(get(handles.edit22,'String'));
af1=str2num(get(handles.edit23,'String'));
at1=str2num(get(handles.edit24,'String'));
Rm=str2num(get(handles.edit25,'String'));
```

```
R=str2num(get(handles.edit27,'String'));
alfa=str2num(get(handles.edit26,'String'));
beta=str2num(get(handles.edit28,'String'));

%%%%%%%%%%%CÓDIGO PARA RESOLUCIÓN DEL SISTEMA DE ECUACIONES%%%%%%%%%%%

if get(handles.radiobutton1,'Value') == 1
    global GPf Gx Gz
    bB=-180+beta;
    bC=180-beta;
    %Posición vértices de plataforma (Pi=[A,B,C])
    Pi=zeros(4,3);
    Pi(:,1)=[-Rm 0 0 1]';
    Pi(:,2)=[Rm*cosd(alfa) Rm*sind(alfa) 0 1]';
    Pi(:,3)=[Rm*cosd(alfa) -Rm*sind(alfa) 0 1]';
    TR=[cosd(f0)*cosd(t0)    -cosd(f0)*sind(t0)    sind(f0)  x0;
        sind(t0)             cosd(t0)             0       0;
        -sind(f0)*cosd(t0)    sind(f0)*sind(t0)     cosd(f0)  z0;
        0                    0                    0       1];
    %Coordenadas finales de plataforma
    Pf=zeros(4,3);
    Pf(:,1)=TR*Pi(:,1);
    Pf(:,2)=TR*Pi(:,2);
    Pf(:,3)=TR*Pi(:,3);
    Pf=single(Pf);
    syms A1 A2 A3 B1 B2 B3 C1 C2 C3 D1 D2
    P=zeros(4,4);
    %Resolución de parámetros de articulaciones para el brazo A
    PAx=-R-(A1)*cos(A3)*sin(A2)==Pf(1,1);
    PAy=-(A1)*sin(A3)==Pf(2,1);
    PAz=(A1)*cos(A3)*cos(A2)==Pf(3,1);
    A=vpasolve([PAx,PAy,PAz],[A1, A2, A3]);
    A1=single(A.A1);
    A2=single(A.A2);
    A2=A2*180/pi;
    A3=single(A.A3);
    A3=A3*180/pi;
    I=find(A1>0);
    if A2(I(1))>-90 && A2(I(1))<0
        if A3(I(1))>-90 && A3(I(1))<90
            I=I(1);
        end
    elseif A2(I(2))>-90 && A2(I(2))<0
        if A3(I(2))>-90 && A3(I(2))<90
            I=I(2);
        end
    end
    P(:,1)=[A1(I) A2(I) A3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo B
    PBx=-R*single(cosd(bB))+B1*(single(sind(bB))*sin(B3)-
single(cosd(bB))*cos(B3)*sin(B2))==Pf(1,2);
    PBy=-R*single(sind(bB))-
B1*(single(cosd(bB))*sin(B3)+cos(B3)*single(sind(bB))*sin(B2))==Pf(2,2);
    PBz=B1*cos(B2)*cos(B3)==Pf(3,2);
    B=vpasolve([PBx,PBy,PBz],[B1, B2, B3]);
    B1=single(B.B1);
    B2=single(B.B2);
    B2=B2*180/pi;
    B3=single(B.B3);
    B3=B3*180/pi;
```

```
    I=find(B1>0);
    if B2(I(1))>-90 && B2(I(1))<0
        if B3(I(1))>-90 && B3(I(1))<90
            I=I(1);
        end
    elseif B2(I(2))>-90 && B2(I(2))<0
        if B3(I(2))>-90 && B3(I(2))<90
            I=I(2);
        end
    end
    P(:,2)=[B1(I) B2(I) B3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo C
    PCx=-R*single(cosd(bC))+C1*(single(sind(bC))*sin(C3)-
single(cosd(bC))*cos(C3)*sin(C2))==Pf(1,3);
    PCy=-R*single(sind(bC))-
C1*(single(cosd(bC))*sin(C3)+cos(C3)*single(sind(bC))*sin(C2))==Pf(2,3);
    PCz=C1*cos(C2)*cos(C3)==Pf(3,3);
    C=vpasolve([PCx,PCy,PCz],[C1, C2, C3]);
    C1=single(C.C1);
    C2=single(C.C2);
    C2=C2*180/pi;
    C3=single(C.C3);
    C3=C3*180/pi;
    I=find(C1>0);
    if C2(I(1))>-90 && C2(I(1))<0
        if C3(I(1))>-90 && C3(I(1))<90
            I=I(1);
        end
    elseif C2(I(2))>-90 && C2(I(2))<0
        if C3(I(2))>-90 && C3(I(2))<90
            I=I(2);
        end
    end
    P(:,3)=[C1(I) C2(I) C3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo D
    PDx=D1*sin(D2)==x0;
    PDz=cos(D2)*D1==z0;
    D=vpasolve([PDx,PDz],[D1, D2]);
    D1=single(D.D1);
    D2=single(D.D2);
    D2=D2*180/pi;
    I=find(D1>0);
    P(:,4)=[D1(I) D2(I) -D2(I)+f0 t0]';
    GPf=zeros(4,3);
    %Brazo A
    GPf(1,1)=-R-(P(1,1))*cosd(P(3,1))*sind(P(2,1));
    GPf(2,1)=-(P(1,1))*sind(P(3,1));
    GPf(3,1)=(P(1,1))*cosd(P(3,1))*cosd(P(2,1));
    %Brazo B
    GPf(1,2)=-R*cosd(bB)+P(1,2)*(sind(bB)*sind(P(3,2))-
cosd(bB)*cosd(P(3,2))*sind(P(2,2)));
    GPf(2,2)=-R*sind(bB)-
P(1,2)*(cosd(bB)*sind(P(3,2))+sind(bB)*cosd(P(3,2))*sind(P(2,2)));
    GPf(3,2)=P(1,2)*cosd(P(2,2))*cosd(P(3,2));
    %Brazo C
    GPf(1,3)=-R*cosd(bC)+P(1,3)*(sind(bC)*sind(P(3,3))-
cosd(bC)*cosd(P(3,3))*sind(P(2,3)));
    GPf(2,3)=-R*sind(bC)-
P(1,3)*(cosd(bC)*sind(P(3,3))+sind(bC)*cosd(P(3,3))*sind(P(2,3)));
    GPf(3,3)=P(1,3)*cosd(P(2,3))*cosd(P(3,3));
```

```
    %Brazo D
    Gx=single(P(1,4)*sind(P(2,4)));
    Gz=single(cosd(P(2,4))*P(1,4));
    GPf=single(GPf);
    READY=1;
    if READY==1
        set(handles.pushbutton3, 'enable', 'on')
    end

elseif get(handles.radiobutton2,'Value') == 1
    global N PA PB PC PD bB bC x z Gf Gt GPA GPB GPC GPD VA VB VC VD AA
AB AC AD
    at=5;                        %Apreciación traslacional [mm]
    ar=1;                        %Apreciación rotacional [°]
    bB=-180+beta;
    bC=180-beta;
    f0=f0*pi/180;
    f1=f1*pi/180;
    t0=t0*pi/180;
    t1=t1*pi/180;
    alfa=alfa*pi/180;
    beta=beta*pi/180;
    bB=bB*pi/180;
    bC=bC*pi/180;
    ar=ar*pi/180;
    %Posición vértices de plataforma (Pi=[A,B,C])
    Pi=zeros(4,3);
    Pi(:,1)=[-Rm 0 0 1]';
    Pi(:,2)=[Rm*cos(alfa) Rm*sin(alfa) 0 1]';
    Pi(:,3)=[Rm*cos(alfa) -Rm*sin(alfa) 0 1]';
    N=[abs(x1-x0)/at abs(z1-z0)/at abs(f1-f0)/ar abs(t1-t0)/ar];
    N=round(max(N));
    x=linspace(x0,x1,N);
    z=linspace(z0,z1,N);
    f=linspace(f0,f1,N);
    t=linspace(t0,t1,N);
    PA=zeros(4,N);
    PB=zeros(4,N);
    PC=zeros(4,N);
    PD=zeros(4,N);
    for i=1:N
    TR=[cos(f(i))*cos(t(i))    -cos(f(i))*sin(t(i))    sin(f(i))   x(i);
        sin(t(i))               cos(t(i))               0           0;
        -sin(f(i))*cos(t(i))    sin(f(i))*sin(t(i))     cos(f(i))   z(i);
        0                       0                       0           1];
    Pf=zeros(4,3);
    Pf(:,1)=TR*Pi(:,1);
    Pf(:,2)=TR*Pi(:,2);
    Pf(:,3)=TR*Pi(:,3);
    Pf=single(Pf);
    syms A1 A2 A3 B1 B2 B3 C1 C2 C3 D1 D2
    %Resolución de parámetros de articulaciones para el brazo A
    PAx=-R-(A1)*cos(A3)*sin(A2)==Pf(1,1);
    PAy=-(A1)*sin(A3)==Pf(2,1);
    PAz=(A1)*cos(A3)*cos(A2)==Pf(3,1);
    A=vpasolve([PAx,PAy,PAz],[A1, A2, A3]);
    A1=single(A.A1);
    A2=single(A.A2);
    A3=single(A.A3);
    I=find(A1>0);
```

```
    if A2(I(1))>(-pi/2) && A2(I(1))<pi/2
        if A3(I(1))>-pi/2 && A3(I(1))<pi/2
            I=I(1);
        end
    elseif A2(I(2))>-pi/2 && A2(I(2))<0
        if A3(I(2))>-pi/2 && A3(I(2))<pi/2
            I=I(2);
        end
    end
    PA(:,i)=[A1(I) A2(I) A3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo B
    PBx=-R*single(cos(bB))+B1*(single(sin(bB))*sin(B3)-
single(cos(bB))*cos(B3)*sin(B2))==Pf(1,2);
    PBy=-R*single(sin(bB))-
B1*(single(cos(bB))*sin(B3)+cos(B3)*single(sin(bB))*sin(B2))==Pf(2,2);
    PBz=B1*cos(B2)*cos(B3)==Pf(3,2);
    B=vpasolve([PBx,PBy,PBz],[B1, B2, B3]);
    B1=single(B.B1);
    B2=single(B.B2);
    B3=single(B.B3);
    I=find(B1>0);
    if B2(I(1))>-pi/2 && B2(I(1))<0
        if B3(I(1))>-pi/2 && B3(I(1))<pi/2
            I=I(1);
        end
    elseif B2(I(2))>-pi/2 && B2(I(2))<0
        if B3(I(2))>-pi/2 && B3(I(2))<pi/2
            I=I(2);
        end
    end
    PB(:,i)=[B1(I) B2(I) B3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo C
    PCx=-R*single(cos(bC))+C1*(single(sin(bC))*sin(C3)-
single(cos(bC))*cos(C3)*sin(C2))==Pf(1,3);
    PCy=-R*single(sin(bC))-
C1*(single(cos(bC))*sin(C3)+cos(C3)*single(sin(bC))*sin(C2))==Pf(2,3);
    PCz=C1*cos(C2)*cos(C3)==Pf(3,3);
    C=vpasolve([PCx,PCy,PCz],[C1, C2, C3]);
    C1=single(C.C1);
    C2=single(C.C2);
    C3=single(C.C3);
    I=find(C1>0);
    if C2(I(1))>-pi/2 && C2(I(1))<0
        if C3(I(1))>-pi/2 && C3(I(1))<pi/2
            I=I(1);
        end
    elseif C2(I(2))>-pi/2 && C2(I(2))<0
        if C3(I(2))>-pi/2 && C3(I(2))<pi/2
            I=I(2);
        end
    end
    PC(:,i)=[C1(I) C2(I) C3(I) 0]';
    %Resolución de parámetros de articulaciones para el brazo D
    PDx=D1*sin(D2)==x(i);
    PDz=cos(D2)*D1==z(i);
    D=vpasolve([PDx,PDz],[D1, D2]);
    D1=single(D.D1);
    D2=single(D.D2);
    I=find(D1>0);
    PD(:,i)=[D1(I) D2(I) -D2(I)+f(i) t(i)]';
```

```
end
GPA=PA;
GPA(2,:)=GPA(2,:)*180/pi;
GPA(3,:)=GPA(3,:)*180/pi;
GPB=PB;
GPB(2,:)=GPB(2,:)*180/pi;
GPB(3,:)=GPB(3,:)*180/pi;
GPC=PC;
GPC(2,:)=GPC(2,:)*180/pi;
GPC(3,:)=GPC(3,:)*180/pi;
GPD=PD;
GPD(2,:)=GPD(2,:)*180/pi;
GPD(3,:)=GPD(3,:)*180/pi;
Gt=t*180/pi;
Gf=f*180/pi;
%Vectores de velocidad
Vx=linspace(Vx0,Vx1,N);
Vz=linspace(Vz0,Vz1,N);
Wf=linspace(Wf0,Wf1,N);
Wt=linspace(Wt0,Wt1,N);
VA=zeros(3,N);
VB=zeros(3,N);
VC=zeros(3,N);
VD=zeros(4,N);
for j=1:N
%Velocidad de brazo A
MV_A=[1 0 Rm*sin(f(j))*cos(t(j))   Rm*cos(f(j))*sin(t(j));
      0 0 0                       -Rm*cos(t(j));
      0 1 Rm*cos(f(j))*cos(t(j))  -Rm*sin(f(j))*sin(t(j))];
MVA=[-cos(PA(3,j))*sin(PA(2,j))  -PA(1,j)*cos(PA(3,j))*cos(PA(2,j))
PA(1,j)*sin(PA(3,j))*sin(PA(2,j));
     -sin(PA(3,j))                0
-PA(1,j)*cos(PA(3,j));
      cos(PA(3,j))*cos(PA(2,j))  -PA(1,j)*sin(PA(2,j))*cos(PA(3,j))
-PA(1,j)*sin(PA(3,j))*cos(PA(2,j))];
VA(:,j)=MVA\(MV_A*[Vx(j) Vz(j) Wf(j) Wt(j)]');
%Velocidad de brazo B
MV_B=[1 0 -Rm*(cos(alfa)*sin(f(j))*cos(t(j))-
sin(alfa)*sin(f(j))*sin(t(j))) -
Rm*(cos(alfa)*cos(f(j))*sin(t(j))+sin(alfa)*cos(f(j))*cos(t(j)));
      0 0 0
Rm*(cos(alfa)*cos(t(j))-sin(alfa)*sin(t(j)));
      0 1 -Rm*(cos(alfa)*cos(f(j))*cos(t(j))-
sin(alfa)*cos(f(j))*sin(t(j)))
Rm*(cos(alfa)*sin(f(j))*sin(t(j))+sin(alfa)*sin(f(j))*cos(t(j)))];
MVB=[sin(bB)*sin(PB(3,j))-cos(bB)*cos(PB(3,j))*sin(PB(2,j)) -
PB(1,j)*cos(bB)*cos(PB(3,j))*cos(PB(2,j))
PB(1,j)*(sin(bB)*cos(PB(3,j))+cos(bB)*sin(PB(3,j))*sin(PB(2,j)));
     -cos(bB)*sin(PB(3,j))-sin(bB)*cos(PB(3,j))*sin(PB(2,j)) -
PB(1,j)*sin(bB)*cos(PB(3,j))*cos(PB(2,j)) -PB(1,j)*(cos(bB)*cos(PB(3,j))-
sin(bB)*sin(PB(3,j))*sin(PB(2,j)));
      cos(PB(3,j))*cos(PB(2,j))                                  -
PB(1,j)*cos(PB(3,j))*sin(PB(2,j))         -
PB(1,j)*sin(PB(3,j))*cos(PB(2,j))];
VB(:,j)=MVB\(MV_B*[Vx(j) Vz(j) Wf(j) Wt(j)]');
%Velocidad de brazo C
MV_C=[1 0 -Rm*(cos(-alfa)*sin(f(j))*cos(t(j))-sin(-
alfa)*sin(f(j))*sin(t(j))) -Rm*(cos(-alfa)*cos(f(j))*sin(t(j))+sin(-
alfa)*cos(f(j))*cos(t(j)));
```

```
          0 0 0
Rm*(cos(-alfa)*cos(t(j))-sin(-alfa)*sin(t(j)));
          0 1 -Rm*(cos(-alfa)*cos(f(j))*cos(t(j))-sin(-
alfa)*cos(f(j))*sin(t(j)))  Rm*(cos(-alfa)*sin(f(j))*sin(t(j))+sin(-
alfa)*sin(f(j))*cos(t(j)))];
    MVC=[sin(bC)*sin(PC(3,j))-cos(bC)*cos(PC(3,j))*sin(PC(2,j)) -
PC(1,j)*cos(bC)*cos(PC(3,j))*cos(PC(2,j))
PC(1,j)*(sin(bC)*cos(PC(3,j))+cos(bC)*sin(PC(3,j))*sin(PC(2,j)));
        -cos(bC)*sin(PC(3,j))-sin(bC)*cos(PC(3,j))*sin(PC(2,j)) -
PC(1,j)*sin(bC)*cos(PC(3,j))*cos(PC(2,j)) -PC(1,j)*(cos(bC)*cos(PC(3,j))-
sin(bC)*sin(PC(3,j))*sin(PC(2,j)));
         cos(PC(3,j))*cos(PC(2,j))                                  -
PC(1,j)*cos(PC(3,j))*sin(PC(2,j))         -
PC(1,j)*sin(PC(3,j))*cos(PC(2,j))];
    VC(:,j)=MVC\(MV_C*[Vx(j) Vz(j) Wf(j) Wt(j)]');
    %Velocidad de brazo D
    MVD=[sin(PD(2,j))  PD(1,j)*cos(PD(2,j)) 0 0;
         cos(PD(2,j)) -PD(1,j)*sin(PD(2,j)) 0 0;
         0             1                    1 0;
         0             0                    0 1];
    VD(:,j)=MVD\[Vx(j) Vz(j) Wf(j) Wt(j)]';
    end
    %Vectores de Aceleración
    Ax=linspace(Ax0,Ax1,N);
    Az=linspace(Az0,Az1,N);
    af=linspace(af0,af1,N);
    at=linspace(at0,at1,N);
    AA=zeros(4,N);
    AB=zeros(4,N);
    AC=zeros(4,N);
    AD=zeros(5,N);
    for k=1:N
    %Aceleración del brazo A
    MA_A=[1 0 Rm*sin(f(k))*cos(t(k))  Rm*cos(f(k))*sin(t(k)) -
2*Rm*sin(f(k))*sin(t(k))*Wf(k)*Wt(k)+Rm*cos(f(k))*cos(t(k))*Wf(k)^2+Rm*co
s(f(k))*cos(t(k))*Wt(k)^2;
          0 0 0                       -Rm*cos(t(k))
Rm*sin(t(k))*Wt(k)^2;
          0 1 Rm*cos(f(k))*cos(t(k)) -Rm*sin(f(k))*sin(t(k)) -
2*Rm*cos(f(k))*sin(t(k))*Wf(k)*Wt(k)-Rm*sin(f(k))*cos(t(k))*Wf(k)^2-
Rm*sin(f(k))*cos(t(k))*Wt(k)^2;
          0 0 0                        0                        1];
    MAA=[-cos(PA(3,k))*sin(PA(2,k))  -PA(1,k)*cos(PA(3,k))*cos(PA(2,k))
PA(1,k)*sin(PA(3,k))*sin(PA(2,k)) -
2*cos(PA(3,k))*cos(PA(2,k))*VA(1,k)*VA(2,k)+2*sin(PA(3,k))*sin(PA(2,k))*V
A(1,k)*VA(3,k)+2*PA(1,k)*sin(PA(3,k))*cos(PA(2,k))*VA(2,k)*VA(3,k)+PA(1,k
)*cos(PA(3,k))*sin(PA(2,k))*VA(2,k)^2+PA(1,k)*cos(PA(3,k))*sin(PA(2,k))*V
A(3,k)^2;
         -sin(PA(3,k))                 0                                -
PA(1,k)*cos(PA(3,k))                -
2*cos(PA(3,k))*VA(1,k)*VA(3,k)+PA(1,k)*sin(PA(3,k))*VA(3,k)^2;
          cos(PA(3,k))*cos(PA(2,k))  -PA(1,k)*sin(PA(2,k))*cos(PA(3,k)) -
PA(1,k)*sin(PA(3,k))*cos(PA(2,k)) -
2*cos(PA(3,k))*sin(PA(2,k))*VA(1,k)*VA(2,k)-
2*sin(PA(3,k))*cos(PA(2,k))*VA(1,k)*VA(3,k)+2*PA(1,k)*sin(PA(3,k))*sin(PA
(2,k))*VA(2,k)*VA(3,k)-PA(1,k)*cos(PA(3,k))*cos(PA(2,k))*VA(2,k)^2-
PA(1,k)*cos(PA(3,k))*cos(PA(2,k))*VA(3,k)^2;
          0                            0
0                                 1];
    AA(:,k)=MAA\(MA_A*[Ax(k) Az(k) af(k) at(k) 1]');
```

```
    %Aceleración del brazo B
    MA_B=[1 0 -Rm*(cos(alfa)*sin(f(k))*cos(t(k))-
sin(alfa)*sin(f(k))*sin(t(k))) -
Rm*(cos(alfa)*cos(f(k))*sin(t(k))+sin(alfa)*cos(f(k))*cos(t(k)))   -
Rm*(cos(alfa)*cos(f(k))*cos(t(k))-sin(alfa)*cos(f(k))*sin(t(k)))*Wf(k)^2-
Rm*(cos(alfa)*cos(f(k))*cos(t(k))-
sin(alfa)*cos(f(k))*sin(t(k)))*Wt(k)^2+2*Rm*(cos(alfa)*sin(f(k))*sin(t(k)
)+sin(alfa)*sin(f(k))*cos(t(k)))*Wf(k)*Wt(k);
          0 0  0
Rm*(cos(alfa)*cos(t(k))-sin(alfa)*sin(t(k)))                          -
Rm*(cos(alfa)*sin(t(k))+sin(alfa)*cos(t(k)))*Wt(k)^2;
          0 1 -Rm*(cos(alfa)*cos(f(k))*cos(t(k))-
sin(alfa)*cos(f(k))*sin(t(k)))
Rm*(cos(alfa)*sin(f(k))*sin(t(k))+sin(alfa)*sin(f(k))*cos(t(k)))
Rm*(cos(alfa)*sin(f(k))*cos(t(k))-
sin(alfa)*sin(f(k))*sin(t(k)))*Wf(k)^2+Rm*(cos(alfa)*sin(f(k))*cos(t(k))-
sin(alfa)*sin(f(k))*sin(t(k)))*Wt(k)^2+2*Rm*(cos(alfa)*cos(f(k))*sin(t(k)
)+sin(alfa)*cos(f(k))*cos(t(k)))*Wf(k)*Wt(k);
          0 0  0
0                                                                    1];
    MAB=[sin(bB)*sin(PB(3,k))-cos(bB)*cos(PB(3,k))*sin(PB(2,k)) -
PB(1,k)*cos(bB)*cos(PB(3,k))*cos(PB(2,k))
PB(1,k)*(sin(bB)*cos(PB(3,k))+cos(bB)*sin(PB(3,k))*sin(PB(2,k)))    -
2*cos(bB)*cos(PB(2,k))*cos(PB(3,k))*VB(1,k)*VB(2,k)+2*(sin(bB)*cos(PB(3,k
))+cos(bB)*sin(PB(2,k))*sin(PB(3,k)))*VB(1,k)*VB(3,k)+2*PB(1,k)*cos(bB)*c
os(PB(2,k))*sin(PB(3,k))*VB(2,k)*VB(3,k)+PB(1,k)*cos(bB)*sin(PB(2,k))*cos
(PB(3,k))*VB(2,k)^2-PB(1,k)*(sin(bB)*sin(PB(3,k))-
cos(bB)*sin(PB(2,k))*cos(PB(3,k)))*VB(3,k)^2;
        -cos(bB)*sin(PB(3,k))-sin(bB)*cos(PB(3,k))*sin(PB(2,k)) -
PB(1,k)*sin(bB)*cos(PB(3,k))*cos(PB(2,k)) -PB(1,k)*(cos(bB)*cos(PB(3,k))-
sin(bB)*sin(PB(3,k))*sin(PB(2,k)))    -2*(cos(bB)*cos(PB(3,k))-
sin(bB)*sin(PB(3,k))*sin(PB(2,k)))*VB(1,k)*VB(3,k)-
2*sin(bB)*cos(PB(3,k))*cos(PB(2,k))*VB(1,k)*VB(2,k)+2*PB(1,k)*sin(bB)*sin
(PB(3,k))*cos(PB(2,k))*VB(2,k)*VB(3,k)+PB(1,k)*sin(bB)*cos(PB(3,k))*sin(P
B(2,k))*VB(2,k)^2+PB(1,k)*(cos(bB)*sin(PB(3,k))+sin(bB)*cos(PB(3,k))*sin(
PB(2,k)))*VB(3,k)^2;
         cos(PB(3,k))*cos(PB(2,k))                                   -
PB(1,k)*sin(PB(2,k))*cos(PB(3,k))         -
PB(1,k)*sin(PB(3,k))*cos(PB(2,k))                                    -
2*cos(PB(3,k))*sin(PB(2,k))*VB(1,k)*VB(2,k)-
2*sin(PB(3,k))*cos(PB(2,k))*VB(1,k)*VB(3,k)+2*PB(1,k)*sin(PB(3,k))*sin(PB
(2,k))*VB(2,k)*VB(3,k)-PB(1,k)*cos(PB(3,k))*cos(PB(2,k))*VB(2,k)^2-
PB(1,k)*cos(PB(3,k))*cos(PB(2,k))*VB(3,k)^2;
         0                                                           0
0                                                                    1];
    AB(:,k)=MAB\(MA_B*[Ax(k) Az(k) af(k) at(k) 1]');
    %Aceleración del brazo C
    MA_C=[1 0 -Rm*(cos(-alfa)*sin(f(k))*cos(t(k))-sin(-
alfa)*sin(f(k))*sin(t(k))) -Rm*(cos(-alfa)*cos(f(k))*sin(t(k))+sin(-
alfa)*cos(f(k))*cos(t(k)))   -Rm*(cos(-alfa)*cos(f(k))*cos(t(k))-sin(-
alfa)*cos(f(k))*sin(t(k)))*Wf(k)^2-Rm*(cos(-alfa)*cos(f(k))*cos(t(k))-
sin(-alfa)*cos(f(k))*sin(t(k)))*Wt(k)^2+2*Rm*(cos(-
alfa)*sin(f(k))*sin(t(k))+sin(-alfa)*sin(f(k))*cos(t(k)))*Wf(k)*Wt(k);
          0 0  0
Rm*(cos(-alfa)*cos(t(k))-sin(-alfa)*sin(t(k)))                        -
Rm*(cos(-alfa)*sin(t(k))+sin(-alfa)*cos(t(k)))*Wt(k)^2;
          0 1 -Rm*(cos(-alfa)*cos(f(k))*cos(t(k))-sin(-
alfa)*cos(f(k))*sin(t(k)))  Rm*(cos(-alfa)*sin(f(k))*sin(t(k))+sin(-
alfa)*sin(f(k))*cos(t(k)))    Rm*(cos(-alfa)*sin(f(k))*cos(t(k))-sin(-
alfa)*sin(f(k))*sin(t(k)))*Wf(k)^2+Rm*(cos(-alfa)*sin(f(k))*cos(t(k))-
```

```
sin(-alfa)*sin(f(k))*sin(t(k)))*Wt(k)^2+2*Rm*(cos(-
alfa)*cos(f(k))*sin(t(k))+sin(-alfa)*cos(f(k))*cos(t(k)))*Wf(k)*Wt(k);
          0 0  0
0                                                                    1];
    MAC=[sin(bC)*sin(PC(3,k))-cos(bC)*cos(PC(3,k))*sin(PC(2,k)) -
PC(1,k)*cos(bC)*cos(PC(3,k))*cos(PC(2,k))
PC(1,k)*(sin(bC)*cos(PC(3,k))+cos(bC)*sin(PC(3,k))*sin(PC(2,k)))    -
2*cos(bC)*cos(PC(2,k))*cos(PC(3,k))*VC(1,k)*VC(2,k)+2*(sin(bC)*cos(PC(3,k
))+cos(bC)*sin(PC(2,k))*sin(PC(3,k)))*VC(1,k)*VC(3,k)+2*PC(1,k)*cos(bC)*c
os(PC(2,k))*sin(PC(3,k))*VC(2,k)*VC(3,k)+PC(1,k)*cos(bC)*sin(PC(2,k))*cos
(PC(3,k))*VC(2,k)^2-PC(1,k)*(sin(bC)*sin(PC(3,k))-
cos(bC)*sin(PC(2,k))*cos(PC(3,k)))*VC(3,k)^2;
        -cos(bC)*sin(PC(3,k))-sin(bC)*cos(PC(3,k))*sin(PC(2,k)) -
PC(1,k)*sin(bC)*cos(PC(3,k))*cos(PC(2,k)) -PC(1,k)*(cos(bC)*cos(PC(3,k))-
sin(bC)*sin(PC(3,k))*sin(PC(2,k)))    -2*(cos(bC)*cos(PC(3,k))-
sin(bC)*sin(PC(3,k))*sin(PC(2,k)))*VC(1,k)*VC(3,k)-
2*sin(bC)*cos(PC(3,k))*cos(PC(2,k))*VC(1,k)*VC(2,k)+2*PC(1,k)*sin(bC)*sin
(PC(3,k))*cos(PC(2,k))*VC(2,k)*VC(3,k)+PC(1,k)*sin(bC)*cos(PC(3,k))*sin(P
C(2,k))*VC(2,k)^2+PC(1,k)*(cos(bC)*sin(PC(3,k))+sin(bC)*cos(PC(3,k))*sin(
PC(2,k)))*VC(3,k)^2;
         cos(PC(3,k))*cos(PC(2,k))                                   -
PC(1,k)*sin(PC(2,k))*cos(PC(3,k))          -
PC(1,k)*sin(PC(3,k))*cos(PC(2,k))                                      -
2*cos(PC(3,k))*sin(PC(2,k))*VC(1,k)*VC(2,k)-
2*sin(PC(3,k))*cos(PC(2,k))*VC(1,k)*VC(3,k)+2*PC(1,k)*sin(PC(3,k))*sin(PC
(2,k))*VC(2,k)*VC(3,k)-PC(1,k)*cos(PC(3,k))*cos(PC(2,k))*VC(2,k)^2-
PC(1,k)*cos(PC(3,k))*cos(PC(2,k))*VC(3,k)^2;
         0                                                        0
0                                                                    1];
    AC(:,k)=MAC\(MA_C*[Ax(k) Az(k) af(k) at(k) 1]');
    %Aceleración del brazo D
    MAD=[sin(PD(2,k))  PD(1,k)*cos(PD(2,k))  0   0
2*cos(PD(2,k))*VD(1,k)*VD(2,k)-PD(1,k)*sin(PD(2,k))*VD(2,k)^2;
         cos(PD(2,k)) -PD(1,k)*sin(PD(2,k))  0   0  -
2*sin(PD(2,k))*VD(1,k)*VD(2,k)-PD(1,k)*cos(PD(2,k))*VD(2,k)^2;
         0             1                     1   0   0;
         0             0                     0   1   0;
         0             0                     0   0   1];
    AD(:,k)=MAD\[Ax(k) Az(k) af(k) at(k) 1]';
    end
    READY=1;

    if READY==1
    set(handles.pushbutton3, 'enable', 'on')
    set(handles.pushbutton4, 'enable', 'on')
    end
end

%%%%%%%%%%CÓDIGO PARA GRAFICACIÓN Y SIMULACIÓN%%%%%%%%%%

function pushbutton2_Callback(hObject, eventdata, handles)
set(handles.edit25,'String','333.33')
set(handles.edit26,'String','60')
set(handles.edit27,'String','666.66')
set(handles.edit28,'String','60')
set(handles.pushbutton1, 'enable', 'on')
set(handles.pushbutton3, 'enable', 'off')
set(handles.pushbutton4, 'enable', 'off')
clearvars -global
clear
```

```
clc

function pushbutton3_Callback(hObject, eventdata, handles)
if get(handles.radiobutton1,'Value') == 1
    global GPf beta Gx Gz R
    xA=[-R GPf(1,1)];
    yA=[0 GPf(2,1)];
    zA=[0 GPf(3,1)];
    xB=[R*cosd(beta) GPf(1,2)];
    yB=[R*sind(beta) GPf(2,2)];
    zB=[0 GPf(3,2)];
    xC=[R*cosd(-beta) GPf(1,3)];
    yC=[R*sind(-beta) GPf(2,3)];
    zC=[0 GPf(3,3)];
    xD=[0 Gx];
    yD=[0 0];
    zD=[0 Gz];
    xP=[xA(1) xB(1) xC(1) xA(1)];
    yP=[yA(1) yB(1) yC(1) yA(1)];
    zP=[zA(1) zB(1) zC(1) zA(1)];
    GPf(:,4)=GPf(:,1);
    axes(handles.axes1);
    plot3(0,0,0,'o','MarkerSize',5)
    hold on
    plot3(xP,yP,zP,'b')
    plot3(GPf(1,:),GPf(2,:),GPf(3,:),'g')
    xlabel('Eje X')
    ylabel('Eje Y')
    zlabel('Eje Z')
    axis equal
    axis([-800 500 -700 700 0 700])
    grid on
    plot3(xA,yA,zA,'r');
    plot3(xB,yB,zB,'r');
    plot3(xC,yC,zC,'r');
    plot3(xD,yD,zD,'r');

plot3(GPf(1,1),GPf(2,1),GPf(3,1),'o','MarkerFaceColor','b','MarkerSize',5
)

plot3(GPf(1,2),GPf(2,2),GPf(3,2),'o','MarkerFaceColor','b','MarkerSize',5
)

plot3(GPf(1,3),GPf(2,3),GPf(3,3),'o','MarkerFaceColor','b','MarkerSize',5
)
    view(3)
    hold off
    set(handles.pushbutton1, 'enable', 'off')
elseif get(handles.radiobutton2,'Value') == 1
    global PA PB PC PD bB bC beta R N
    GPf=zeros(4,3);
    for g=1:N
    %Brazo A
    GPf(1,1)=-R-(PA(1,g))*cos(PA(3,g))*sin(PA(2,g));
    GPf(2,1)=-(PA(1,g))*sin(PA(3,g));
    GPf(3,1)=(PA(1,g))*cos(PA(3,g))*cos(PA(2,g));
    %Brazo B
    GPf(1,2)=-R*cos(bB)+PB(1,g)*(sin(bB)*sin(PB(3,g))-
cos(bB)*cos(PB(3,g))*sin(PB(2,g)));
```

```
    GPf(2,2)=-R*sin(bB)-
PB(1,g)*(cos(bB)*sin(PB(3,g))+sin(bB)*cos(PB(3,g))*sin(PB(2,g)));
    GPf(3,2)=PB(1,g)*cos(PB(2,g))*cos(PB(3,g));
    %Brazo C
    GPf(1,3)=-R*cos(bC)+PC(1,g)*(sin(bC)*sin(PC(3,g))-
cos(bC)*cos(PC(3,g))*sin(PC(2,g)));
    GPf(2,3)=-R*sin(bC)-
PC(1,g)*(cos(bC)*sin(PC(3,g))+sin(bC)*cos(PC(3,g))*sin(PC(2,g)));
    GPf(3,3)=PC(1,g)*cos(PC(2,g))*cos(PC(3,g));
    %Brazo D
    Gx=single(PD(1,g)*sin(PD(2,g)));
    Gz=single(cos(PD(2,g))*PD(1,g));
    GPf=single(GPf);
    %Vectorización del brazo A
    xA=[-R GPf(1,1)];
    yA=[0 GPf(2,1)];
    zA=[0 GPf(3,1)];
    %Vectorización del brazo B
    xB=[R*cos(beta) GPf(1,2)];
    yB=[R*sin(beta) GPf(2,2)];
    zB=[0 GPf(3,2)];
    %Vectorización del brazo C
    xC=[R*cos(-beta) GPf(1,3)];
    yC=[R*sin(-beta) GPf(2,3)];
    zC=[0 GPf(3,3)];
    %Vectorización del brazo D
    xD=[0 Gx];
    yD=[0 0];
    zD=[0 Gz];
    %Vectorización de plataforma fija
    xP=[xA(1) xB(1) xC(1) xA(1)];
    yP=[yA(1) yB(1) yC(1) yA(1)];
    zP=[zA(1) zB(1) zC(1) zA(1)];
    %Vectorización de plataforma movil
    GPf(:,4)=GPf(:,1);
    %Graficación
    axes(handles.axes1);
    plot3(0,0,0,'o','MarkerSize',5)
    hold on
    plot3(xP,yP,zP,'b')
    plot3(GPf(1,:),GPf(2,:),GPf(3,:),'g')
    xlabel('Eje X')
    ylabel('Eje Y')
    zlabel('Eje Z')
    axis equal
    axis([-800 500 -700 700 0 700])
    grid on
    plot3(xA,yA,zA,'r');
    plot3(xB,yB,zB,'r');
    plot3(xC,yC,zC,'r');
    plot3(xD,yD,zD,'r');

plot3(GPf(1,1),GPf(2,1),GPf(3,1),'o','MarkerFaceColor','b','MarkerSize',5
)

plot3(GPf(1,2),GPf(2,2),GPf(3,2),'o','MarkerFaceColor','b','MarkerSize',5
)

plot3(GPf(1,3),GPf(2,3),GPf(3,3),'o','MarkerFaceColor','b','MarkerSize',5
)
```

```
    view(3)
    hold off
    pause(0.01)
    end
    set(handles.pushbutton1, 'enable', 'off')
end

function pushbutton4_Callback(hObject, eventdata, handles)
menu1=menu('Seleccione el movimiento realizado','Desplazamiento en
X','Desplazamiento en Z','Rotación alrededor de Y','Rotación alrededor de
Z');
if menu1==1
    menu2=menu('Seleccione las gráficas que desea
visualizar','Posición','Velocidad','Aceleración');
    if menu2==1
        X_P_plot
    elseif menu2==2
        X_V_plot
    elseif menu2==3
        X_A_plot
    end
elseif menu1==2
    menu3=menu('Seleccione las gráficas que desea
visualizar','Posición','Velocidad','Aceleración');
    if menu3==1
        Z_P_plot
    elseif menu3==2
        Z_V_plot
    elseif menu3==3
        Z_A_plot
    end
elseif menu1==3
    menu4=menu('Seleccione las gráficas que desea
visualizar','Posición','Velocidad','Aceleración');
    if menu4==1
        F_P_plot
    elseif menu4==2
        F_V_plot
    elseif menu4==3
        F_A_plot
    end
elseif menu1==4
    menu5=menu('Seleccione las gráficas que desea
visualizar','Posición','Velocidad','Aceleración');
    if menu5==1
        T_P_plot
    elseif menu5==2
        T_V_plot
    elseif menu5==3
        T_A_plot
    end
end


function radiobutton1_Callback(hObject, eventdata, handles)
set(handles.edit5, 'enable', 'off')
set(handles.edit6, 'enable', 'off')
set(handles.edit7, 'enable', 'off')
set(handles.edit8, 'enable', 'off')
```

```
set(handles.edit9, 'enable', 'off')
set(handles.edit10, 'enable', 'off')
set(handles.edit11, 'enable', 'off')
set(handles.edit12, 'enable', 'off')
set(handles.edit13, 'enable', 'off')
set(handles.edit14, 'enable', 'off')
set(handles.edit15, 'enable', 'off')
set(handles.edit16, 'enable', 'off')
set(handles.edit17, 'enable', 'off')
set(handles.edit18, 'enable', 'off')
set(handles.edit19, 'enable', 'off')
set(handles.edit20, 'enable', 'off')
set(handles.edit21, 'enable', 'off')
set(handles.edit22, 'enable', 'off')
set(handles.edit23, 'enable', 'off')
set(handles.edit24, 'enable', 'off')

function radiobutton2_Callback(hObject, eventdata, handles)
set(handles.edit5, 'enable', 'on')
set(handles.edit6, 'enable', 'on')
set(handles.edit7, 'enable', 'on')
set(handles.edit8, 'enable', 'on')
set(handles.edit9, 'enable', 'on')
set(handles.edit10, 'enable', 'on')
set(handles.edit11, 'enable', 'on')
set(handles.edit12, 'enable', 'on')
set(handles.edit13, 'enable', 'on')
set(handles.edit14, 'enable', 'on')
set(handles.edit15, 'enable', 'on')
set(handles.edit16, 'enable', 'on')
set(handles.edit17, 'enable', 'on')
set(handles.edit18, 'enable', 'on')
set(handles.edit19, 'enable', 'on')
set(handles.edit20, 'enable', 'on')
set(handles.edit21, 'enable', 'on')
set(handles.edit22, 'enable', 'on')
set(handles.edit23, 'enable', 'on')
set(handles.edit24, 'enable', 'on')
```