



Manual para el manejo del software

Universidad Politécnica de Guanajuato
Departamento de Robótica

Contenido

1. Fundamentos de Robótica	3
1.1 Transformaciones	3
1.1.1 Rotaciones	4
1.1.2 Parametrización de rotaciones	8
1.1.3 Ángulo/Eje	12
1.1.4 Matriz DH (Denavit – Hartenberg)	14
1.2 Cinemática directa	17
1.3 Cinemática inversa	21
1.3.1 Cinemática inversa por el método de Newton – Raphson	22
1.3.2 Cinemática inversa por la función de Scipy	25
1.4 Cinemática diferencial	28

Tabla de ilustraciones

Ilustración 1. Ventana principal	3
Ilustración 2. Ventana de transformaciones	4
Ilustración 3. Ventana de rotaciones	5
Ilustración 4. Ventana emergente de rotaciones	6
Ilustración 5. Ejemplo de la ventana de rotaciones (1)	7
Ilustración 6. Ejemplo de la ventana de rotaciones (2)	7
Ilustración 7. Mensaje emergente de la ventana de parametrización de rotaciones	9
Ilustración 8. Ventana de parametrización de rotaciones	10
Ilustración 9. Ejemplo 1 de la ventana de parametrización	11
Ilustración 10. Ejemplo 2 de la ventana de parametrización de rotaciones	12
Ilustración 11. Ventana de ángulo/eje	13
Ilustración 12. Mensaje emergente de la ventana de ángulo/eje	13
Ilustración 13. Ejemplo de la ventana ángulo/eje	14
Ilustración 14. Venta de la matriz DH	15
Ilustración 15. Mensaje emergente de la ventana de la matriz DH	16
Ilustración 16. Ejemplo de la ventana de la matriz DH	17
Ilustración 17. Ventana de Cinemática directa	18
Ilustración 18. Ejemplo de la ventana de cinemática directa	19
Ilustración 19. Diagrama cinemático (1)	20
Ilustración 20. Diagrama cinemático (2)	21
Ilustración 21. Ventana de cinemática inversa	22
Ilustración 22. Ventana de cinemática inversa por el método de Newton - Raphson	23
Ilustración 23. Ejemplo de la ventana de cinemática inversa por el método de Newton - Raphson	25
Ilustración 24. Ventana de cinemática inversa por la función root	26
Ilustración 25. Ejemplo de la venta de cinemática inversa por medio de la función root	27
Ilustración 26. Ventana de cinemática diferencial	28
Ilustración 27. Ejemplo de la ventana de cinemática diferencial	30

1. Fundamentos de Robótica

Esta es la ventana principal de la interfaz gráfica que le ayudara a entender mejores conocimientos de la carrera de ingeniería robótica, la cual se basa en los conocimientos a nivel matemático, análisis de los manipuladores/robots y velocidades angulares de las articulaciones del manipulador/robot. Cada botón contiene un tema generalizado a tratar, el orden que lleva es orden de cómo se deben de ver los temas con los alumnos de la carrera.

Los botones contarán con los siguientes nombres:

- Transformations
- Forward Kinematics
- Inverse Kinematics
- Differential Kinematics

Para visualizar la ventana véase la *Ilustración 1. Ventana principal*.

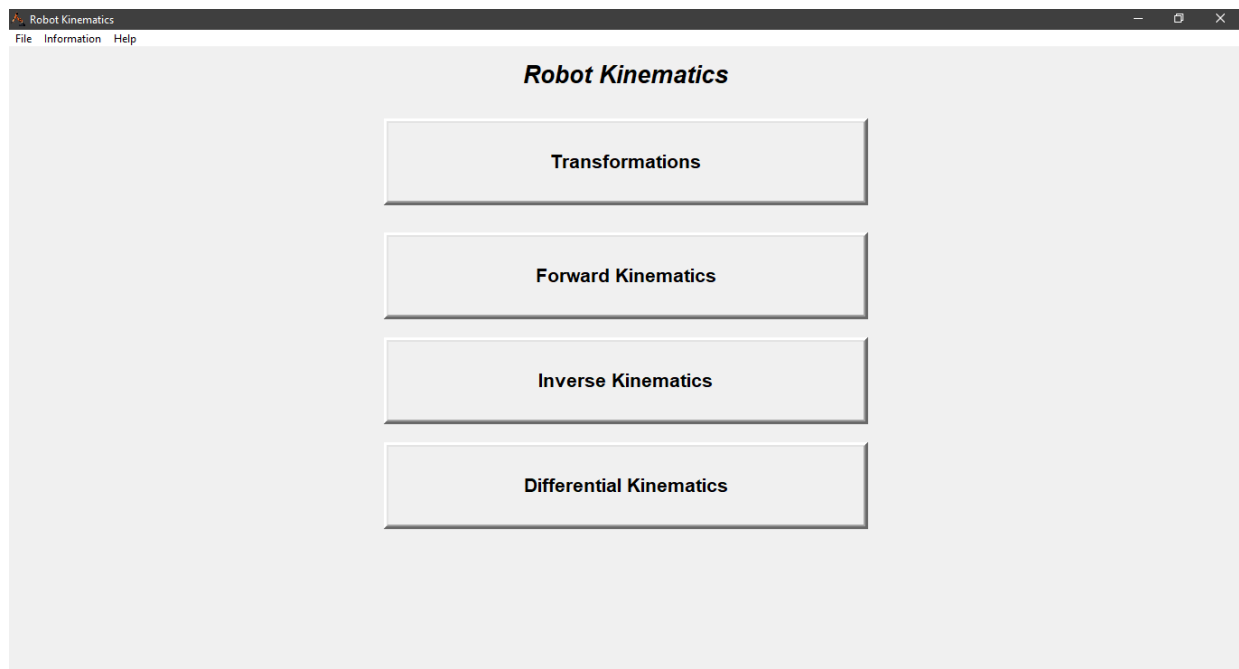


Ilustración 1. Ventana principal

1.1 Transformaciones

En esta ventana se encontrará con varios botones, los cuales le ofrecen varias tareas de manera independientes, en esas ventanas se diseñaron para que tuvieran lo necesario para que obtuviera datos o se visualizaran ciertos casos, ya dependerá al botón quede clic.

Temas o nombres de cada botón en la ventana de transformaciones:

- Rotations
- Parametrization of rotations
- Axis/Angle
- Matrix DH (Denavit – Hartenberg)

La ventana principal de transformaciones puede verla en la ***Ilustración 2. Ventana de transformaciones.***

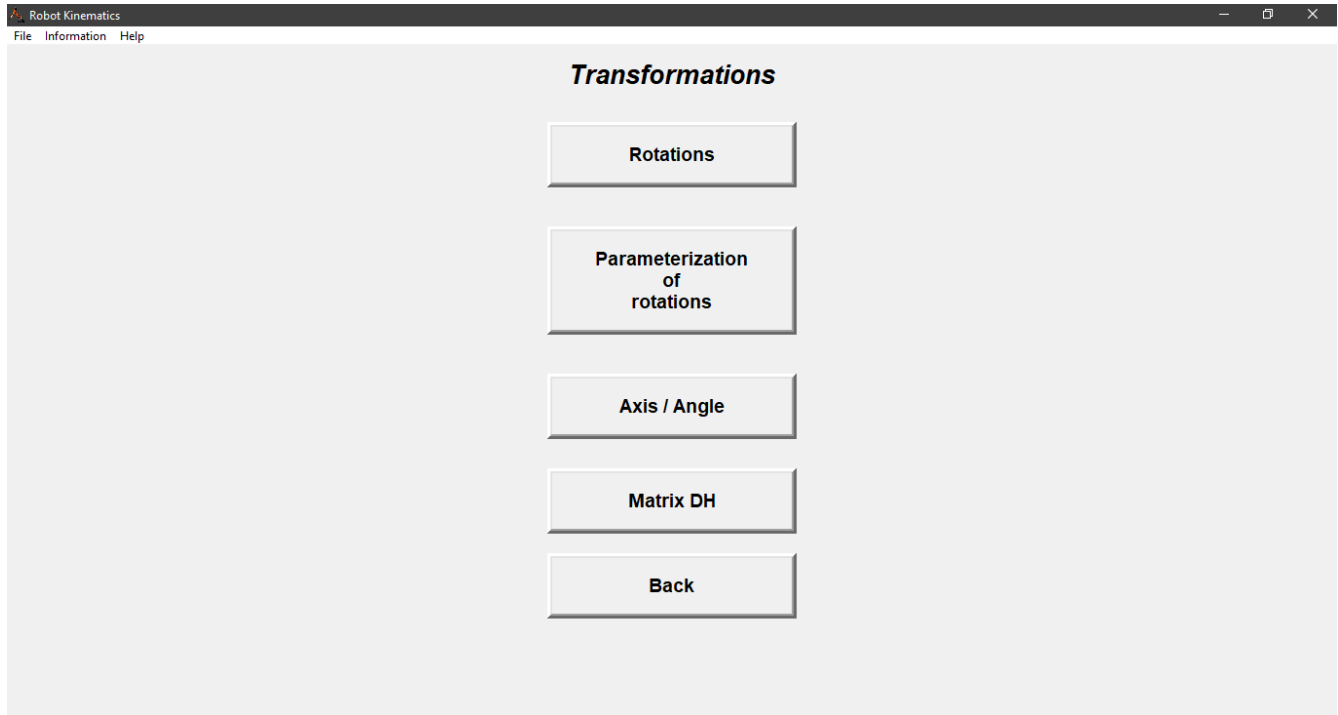


Ilustración 2. Ventana de transformaciones

1.1.1 Rotaciones

En esta ventana tendrá a su disposición la visualización de qué forma es la que giran los ejes de un sistema de referencia, ya que en ocasiones se puede llegar a confundir de qué manera quedaría nuestros ejes de referencia.

Instrucciones:

- Colar el valor del ángulo que se desea ingresar en el "eje x".
- Colar el valor del ángulo que se desea ingresar en el "eje y".
- Colar el valor del ángulo que se desea ingresar en el "eje z".

La ventana de rotaciones se puede ver en la ***Ilustración 3. Ventana de rotaciones.***

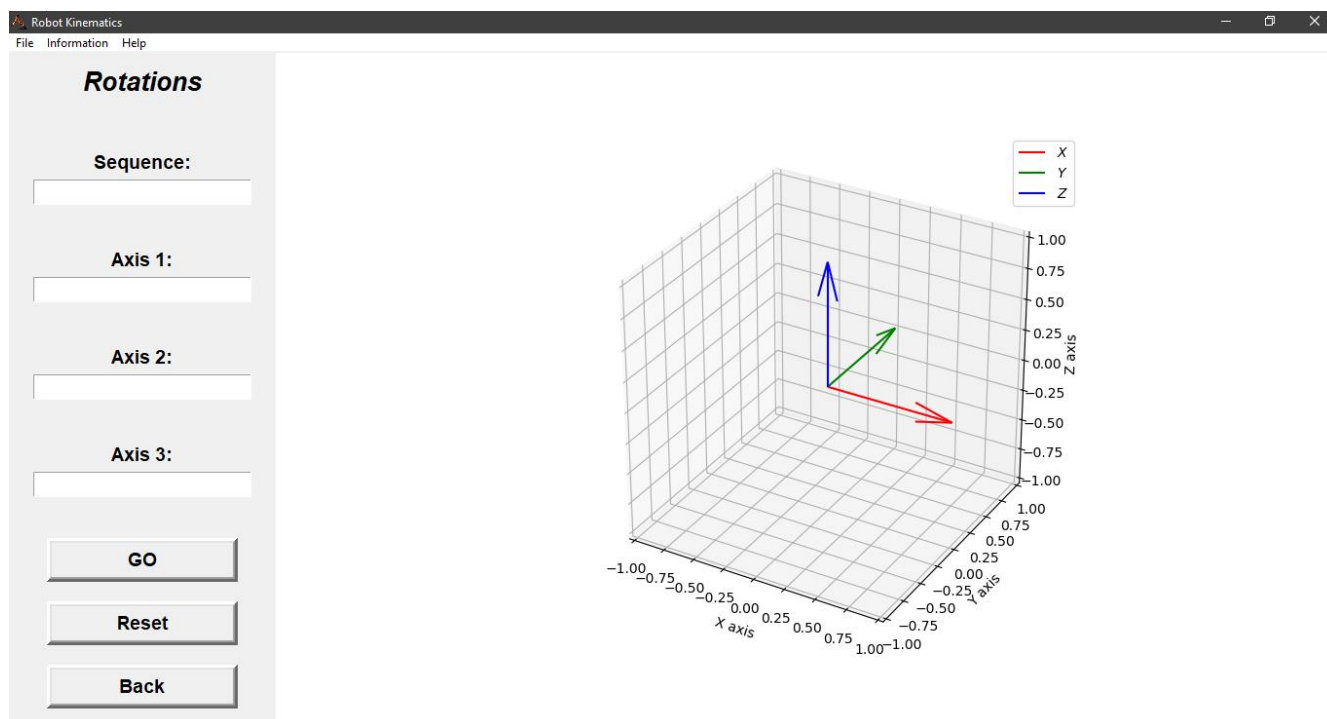


Ilustración 3. Ventana de rotaciones

Como se puede ver, tenemos nuestro sistema de referencia en el origen, una vez que se utilice podremos notar que vamos a tener dos diferentes sistemas de referencia, pero no es así, porque realmente lo que hace esta ventana es mostrar como quedo nuestro sistema rotado a comparación a como estaba originalmente. La ventana de rotaciones es muy didáctica y fácil de entender desde un principio, en la caja de texto "sequence" como su nombre lo dice vamos a ingresar las secuencias como queremos que rote nuestro sistema de referencia. Las demás cajas de texto se estarán ingresando los valores de los ángulos, como se puede observar se tiene el nombre de los ejes enumerados, esto se realizó con el fin de generalizar para que no importará la secuencia y cada caja de texto le pertenece a cada uno de los ejes que se le ponga en la secuencia.

Ejemplo: La secuencia "ZXZ", el eje 1 le pertenecerá a Z, el eje 2 le pertenecerá a X y el eje 3 le pertenecerá a Z.

Como aclaración, si se repite un eje en diferente posición eso no significa que deban ser el mismo ángulo, son dos ángulos completamente diferentes que van a dar una determinada rotación al mismo eje. Las únicas secuencias que se pueden hacer son aquellas secuencias que se apeguen a la realidad, en otras palabras, son las secuencias de ejes de los "ángulos de Euler y los ángulos de Tait-Bryan", si quiere más información sobre estas secuencias puede revisar la siguiente URL: https://en.wikipedia.org/wiki/Euler_angles.

Botones:

- **GO:** En este botón prácticamente se encarga de ejecutar nuestro programa y mostrarnos en la gráfica 3D como quedaría rotado nuestro sistema de referencia respecto a cómo estaba en su sistema de origen. También cuando demos clic a este botón nos va a mostrar

un mensaje emergente preguntándonos que, si lo ángulos los estamos ingresando en grados, si quiere ver como se muestra el mensaje emergente vea la ***Ilustración 4. Ventana emergente de rotaciones***. Para saber lo que hacen cada una de estas opciones de la ventana emergente lea los siguientes puntos.

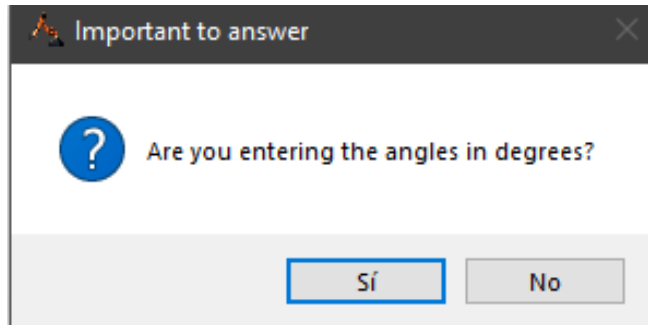


Ilustración 4. Ventana emergente de rotaciones

- *Sí*: Esta opción nos da la posibilidad de ingresar TODOS los ángulos en grados.
- *No*: Esta opción nos permite ingresar todos los ángulos en forma de radianes.
- **Reset**: En este botón vamos a tener el funcionamiento de borrar nuestras cajas de texto y regresar nuestro sistema de referencia a su estado original.
- **Back**: En este botón vamos a tener la posibilidad de regresar a nuestra ventana principal en donde este caso sería la ventana de transformaciones.

Ejemplo: La secuencia que se usó para este ejemplo fue "ZYZ", los ángulos se pueden observar en las ***Ilustración 5. Ejemplo de la ventana de rotaciones (1)*** e ¡Error! No se encuentra el origen de la referencia..

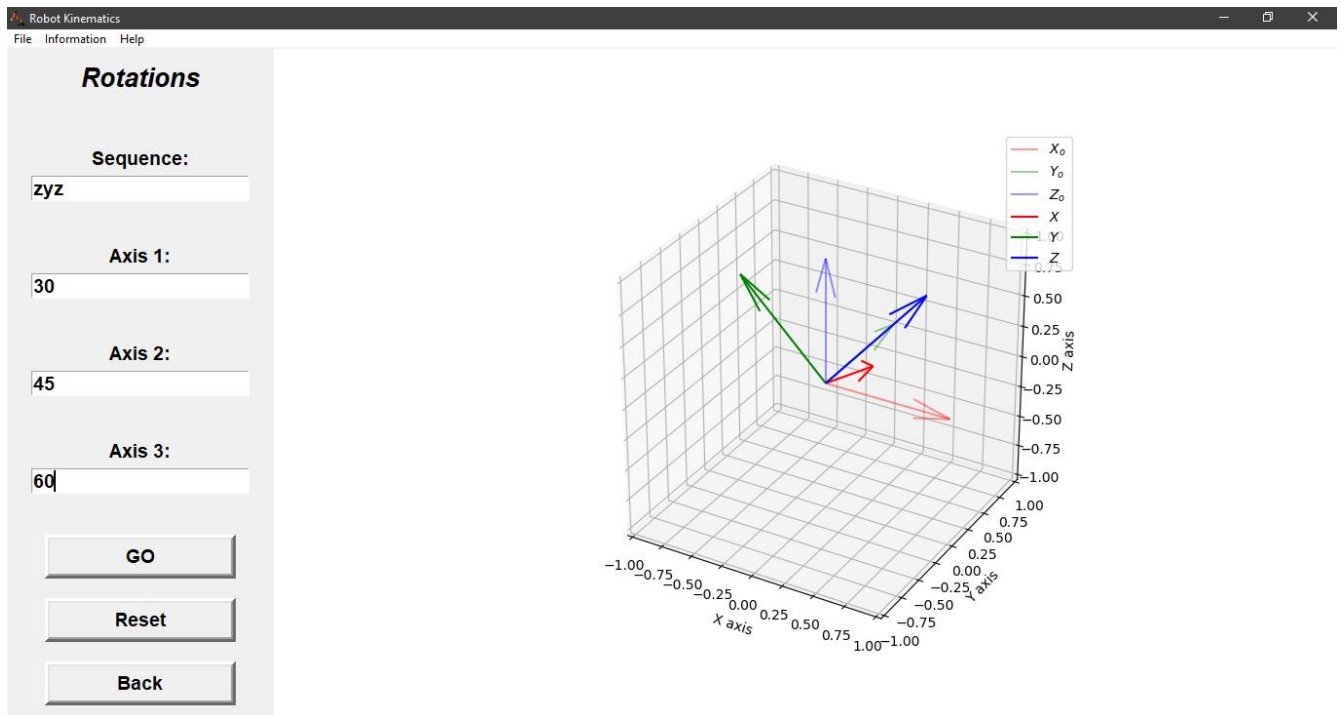


Ilustración 5. Ejemplo de la ventana de rotaciones (1)

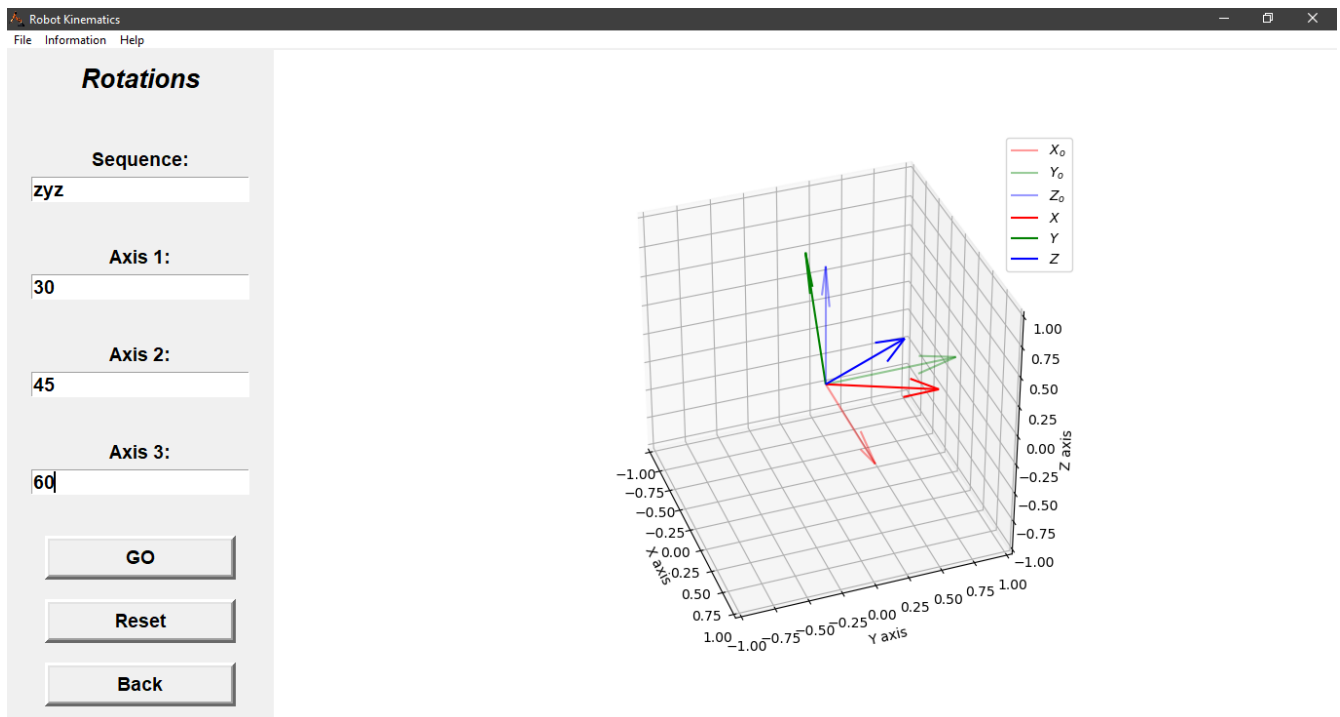


Ilustración 6. Ejemplo de la ventana de rotaciones (2)

1.1.2 Parametrización de rotaciones

En esta ventana será posible trabajar conversiones de matrices de rotación las cuales están restringidas a ser matrices de 3x3. Dicha matriz se debe ingresar en la caja de texto con el nombre de “Matriz de rotación”.

Usted tendrá dos posibles conversiones de ángulos, las cuales son:

- Ángulos de Euler
- Ángulos Roll, Pitch and Yaw (RPY)

Dependiendo del tipo de ángulos que elija, tendrá que revisar los siguientes puntos:

- En los ángulos de Euler tendrá que elegir una posible combinación que nos ofrecen dichos ángulos, esto lo podemos encontrar en la caja de opciones que tenemos enseguida de la caja de texto de “Tipos de ángulos”.
- En los ángulos RPY ya no es necesario elegir una combinación de ángulos, aunque teóricamente sean un tipo de ángulos de Euler, se programó para que hiciera la secuencia convencional de ejes que son: ZYX.

También debemos elegir el número de solución para el ángulo θ , para el cuál siempre dispones de dos posibles soluciones y la ventana nos lo puede mostrar. Si requiere información extra para ver la teoría de estos tipos de ángulos, puede consultar los siguientes URL:

- Ángulos de Euler: https://es.wikipedia.org/wiki/%C3%81ngulos_de_Euler
- Ángulos RPY: https://es.wikipedia.org/wiki/%C3%81ngulos_de_navegaci%C3%B3n

¿Cómo se debe de ingresar la matriz de rotación?

Es muy importante la manera en la que vamos a ingresar la matriz de rotación, sino nos mandará un error la interfaz. Para evitar esto solo se necesita ingresar la matriz de manera de arregló, que sería entre corchetes ([]), lo único que se encerraría entre corchetes serían los renglones de la matriz y cada columna se deberá separar por una coma (.). Esto se hace principalmente para que el programa vaya más rápido y nos devuelva el valor casi inmediatamente después de presionar el botón de ejecución. No sé preocupe si no ha entendido la parte anterior, puede acudir a ver los ejemplos sobre cómo usar esta ventana.

Botones

En los siguientes puntos se encontrarán los nombres de los botones y que funcionalidad tiene cada uno de ellos.

- **GO:** Este botón prácticamente es nuestra ejecución para que nos muestre los resultados del tema relacionado a la ventana. Al presionar este clic y suponiendo que todo lo que ingresamos está bien el programa nos mostrará un mensaje emergente. Que se puede observar la *Ilustración 7. Mensaje emergente de la ventana de parametrización de rotaciones.*

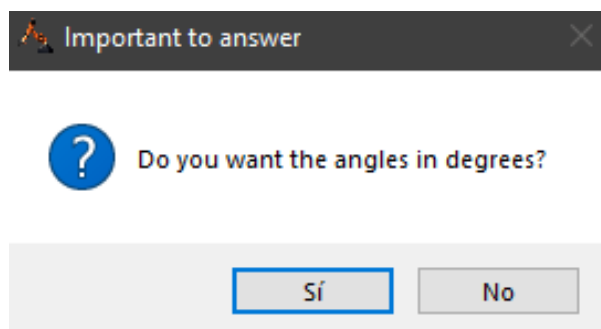


Ilustración 7. Mensaje emergente de la ventana de parametrización de rotaciones

Para saber qué hace cada una de las dos posibles opciones que nos muestra el mensaje emergente lea los siguientes puntos:

- *Sí*: Si se eligió esta opción es porque queremos que el software nos de vuelta los resultados en grados.
- *No*: Si se eligió esta opción es porque queremos que el software nos de vuelta los resultados en radianes.
- **Reset**: Este botón su única tarea es limpiar toda nuestra ventana quita lo escrito en las cajas de texto y los resultados mostrados.
- **Back**: Este botón nos regresa una ventana atrás, la ventaja de cuando nos regresamos y si tenemos un resultado nos lo mantiene mientras el programa siga en ejecución.

Primero empezaremos por explicar cómo agregar una matriz al programa, por cada uno de los ejemplos empezaremos planteando como se ven normalmente las matrices y de cómo se deben de ingresar a la interfaz del software. Se presentarán dos ejemplos con los diferentes tipos de ángulos. Para poder visualizar la ventana original de este apartador vea la ***Ilustración 8. Ventana de parametrización de rotaciones.***

Ilustración 8. Ventana de parametrización de rotaciones

Ejemplos

Ejemplo 1. (Ángulos de Euler)

La matriz que se ingresó para este ejemplo es la siguiente:

$$\begin{bmatrix} 0.0699 & -0.933 & 0.3536 \\ 0.933 & -0.699 & -0.3536 \\ 0.3536 & 0.3536 & 0.866 \end{bmatrix}$$

En la caja de texto que tiene el nombre de "Rotation matrix", se le debe ingresar la anterior matriz en la interfaz es de la siguiente forma:

$$[0.0669, 0.933, 0.3536], [0.933, -0.669, 0.3536], [0.3536, 0.3536, 0.866]$$

Nota: No es importante si se deja o no espacio después de las comas entre cada valor.

Como se puede notar se cumple lo que antes se mencionó de poner los valores por filas y entre corchetes, esto se hace por el motivo de que en esa forma se crean los arreglos en la librería usada, que se llama numpy y cómo podemos ver a simple vista la matriz la estamos escribiendo en forma de arreglo.

En los tipos de ángulos en este caso debemos de desplegar la caja de opciones para poder elegir los ángulos de Euler.

Importante: Para este tipo de ángulos es demasiado importante elegir una secuencia de multiplicación de ángulos, existen 6 diferentes combinaciones de ángulos, si desconoce del tema relacionado le recomendamos que revise los URL que se le recomendaron con anterioridad al principio de este tema.

Para poder ver el resultado de este problema revise la *Ilustración 9. Ejemplo 1 de la ventana de parametrización*.

Ilustración 9. Ejemplo 1 de la ventana de parametrización

Ejemplo 2. (Ángulos Roll, Pitch and Yaw)

La matriz que se ingresó para este ejemplo es la siguiente:

$$\begin{bmatrix} 0.0699 & -0.933 & 0.3536 \\ 0.933 & -0.699 & -0.3536 \\ 0.3536 & 0.3536 & 0.866 \end{bmatrix}$$

En la caja de texto que tiene el nombre de "Rotation matrix", se le debe ingresar la anterior matriz en la interfaz es de la siguiente forma:

$$[0.0669, 0.933, 0.3536], [0.933, -0.669, 0.3536], [0.3536, 0.3536, 0.866]$$

Nota: No es importante si se deja o no espacio después de las comas entre cada valor.

Como se puede notar se cumple lo que antes se mencionó de poner los valores por filas y entre corchetes, esto se hace por el motivo de que en esa forma se crean los arreglos en la librería usada, que se llama numpy y cómo podemos ver a simple vista la matriz la estamos escribiendo en forma de arreglo.

Para este tipo de ángulos no es necesario elegir una opción de posibles combinaciones por el motivo de que ya cuenta con una secuencia definida y por lo que nosotros no podemos cambiarla, dicha secuencia está definida por la siguiente multiplicación de ejes XYZ.

Para ver el resultado de este problema revise la *Ilustración 10. Ejemplo 2 de la ventana de parametrización de rotaciones*.

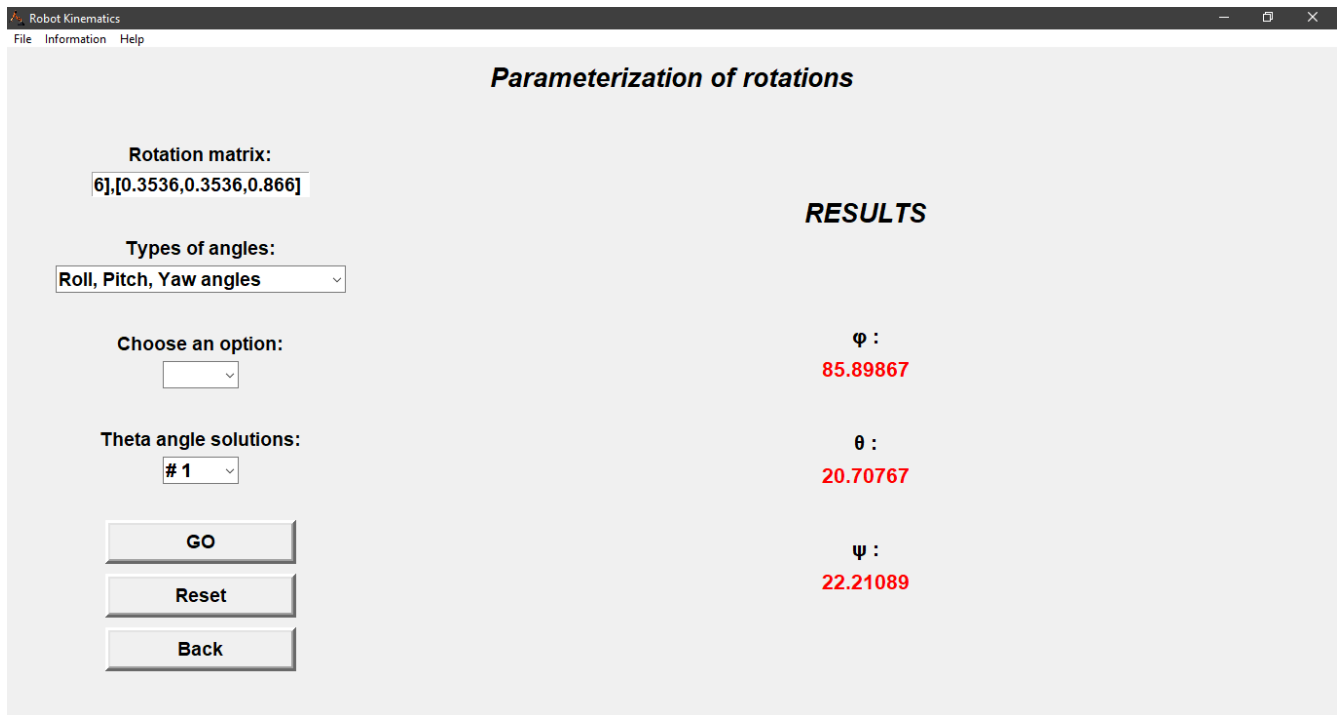


Ilustración 10. Ejemplo 2 de la ventana de parametrización de rotaciones

1.1.3 Ángulo/Eje

En esta ventana vamos a tener la funcionalidad que a partir de una matriz de rotación de 3x3 esto siempre lo va a requerir ya que es uno de los requisitos que pide el método por el que se está realizando. Si quiere conocer más sobre el tema, como funciona y cuáles son las ecuaciones algebraicas con las que se resuelve puede consultar el capítulo 2.5.3 *Axis/angle representation* del libro *Robot Modeling and Control* - Mark W. Spong, Seth Hutchinson, M. Vidyasagar.

Esta ventana nos va a devolver un ángulo y vector de dirección/orientación de dicha matriz de rotación que se le está ingresando.

Si quiere ver la ventana de ángulo/eje véase la *Ilustración 11. Ventana de ángulo/eje*. Como se comentó anteriormente lo que nos devuelve la ventana es un ángulo y un vector de dirección que es la representación de nuestro eje.

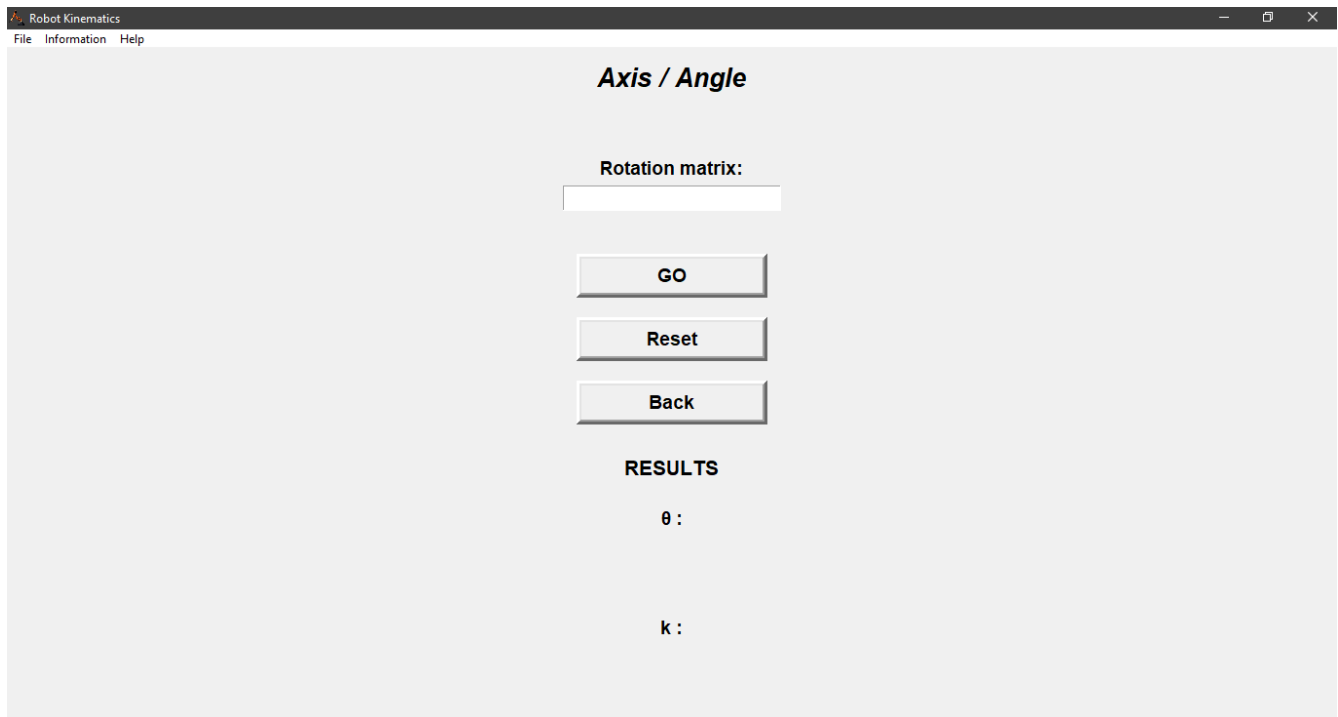


Ilustración 11. Ventana de ángulo/eje

La funcionalidad de los botones es la siguiente:

- **GO:** Tiene la funcionalidad de ejecutar el programa una vez le ingresemos los datos ingresados en la caja de texto. Al dar clic este botón nos mandará un mensaje emergente que se puede observar en la **Ilustración 12. Mensaje emergente de la ventana de ángulo/eje.**

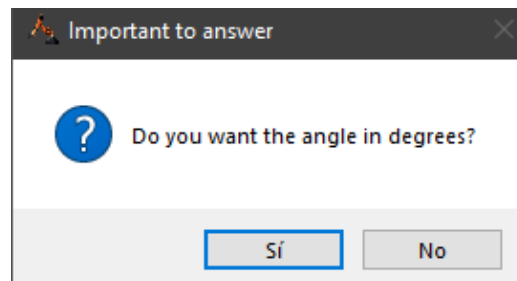


Ilustración 12. Mensaje emergente de la ventana de ángulo/eje

Como podemos ver en la anterior imagen, solo nos da dos opciones las cuales solo debemos de responder que es lo que queremos, si queremos que el ángulo de respuesta lo queremos en grados o en radianes. Si damos clic en la opción “sí” es porque efectivamente queremos el ángulo en grados. Y al contrario elegimos la opción “no” es porque queremos nuestro ángulo en radianes.

- **Reset:** Este botón solamente tiene la funcionalidad de borrar todo que se encuentra en esa ventana como: resultados y la matriz de rotación ingresada.
- **Back:** La funcionalidad que tiene es regresar a la ventana principal, en este caso sería a la ventana de transformaciones.

Para poder entender cómo se debe ingresar la debida matriz de rotación a la caja de texto se le recomienda que vea el ejemplo que se muestra a continuación.

Ejemplo: El ejemplo que vamos a utilizar es el mismo que viene en el libro que se hizo referencia unos cuantos párrafos anteriores, esto se hace con finalidad de comprobar que la ventana tiene una eficiencia correcta.

La matriz de rotación ingresada está dada por los siguientes valores:

$$\begin{bmatrix} 0 & -0.8660254 & 0.5 \\ 0.5 & -0.4330127 & -0.75 \\ 0.8660254 & 0.25 & 0.4330127 \end{bmatrix}$$

Y la matriz se debe ingresar en la caja de texto de la siguiente forma:

[0,-0.8660254,0.5],[0.5,-0.4330127,-0.75],[0.8660254,0.25,0.4330127]

Para ver el resultado que nos regresa la venta véase la *Ilustración 13. Ejemplo de la ventana ángulo/eje.*

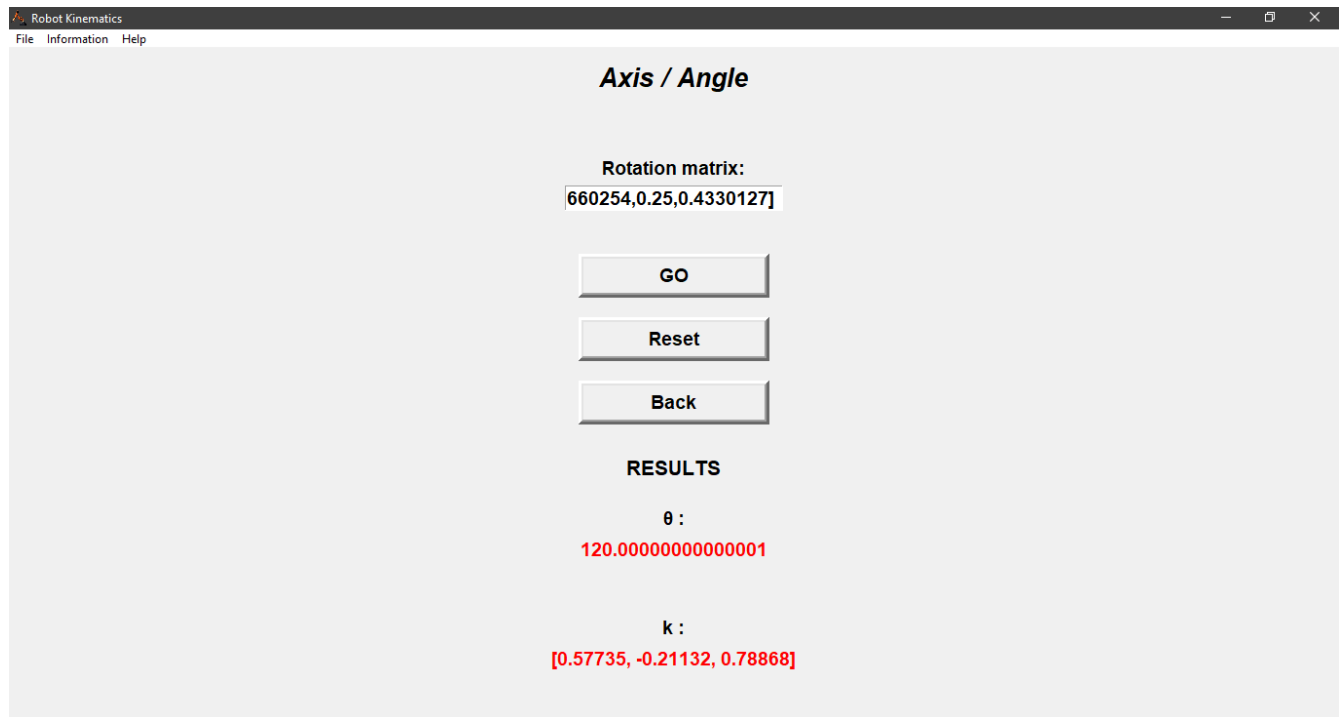


Ilustración 13. Ejemplo de la ventana ángulo/eje

1.1.4 Matriz DH (Denavit – Hartenberg)

Esta ventana tiene la funcionalidad de crear por nosotros la matriz de Denavit – Hartenberg, pero solamente es de un grado de libertad de nuestro manipulador, en otras palabras esta ventana nos ayuda a que nos demos una idea de cómo es la forma correcta de construir nuestra matriz DH y si no queremos entretenernos en crear nuestras matrices, aunque en futuras ventanas que se explicaran en este archivo nos daremos cuenta que podemos hacer cosas más específicas con este tipo de matrices y con más número de matrices.

Esta ventana para poder trabajar nos pide cuatro parámetros de entrada que son los que nos indica la teoría para construir este tipo de matrices. La salida o el resultado que nos va a mostrar nuestra ventana va a ser una matriz de 4x4.

Si se quiere conocer más sobre el tema le recomendamos que revise el siguiente enlace: https://en.wikipedia.org/wiki/Denavit-Hartenberg_parameters. En este enlace encontrará información más detallada de él porque pedimos los cuatro parámetros de entrada y cómo es que funciona. Aunque ya tenga noción del tema creemos que tiene una información muy clara y fácil de entender para cualquier tipo de usuario.

Si quiere observar como luce esta ventana en la interfaz véase la *Ilustración 14. Venta de la matriz DH*.

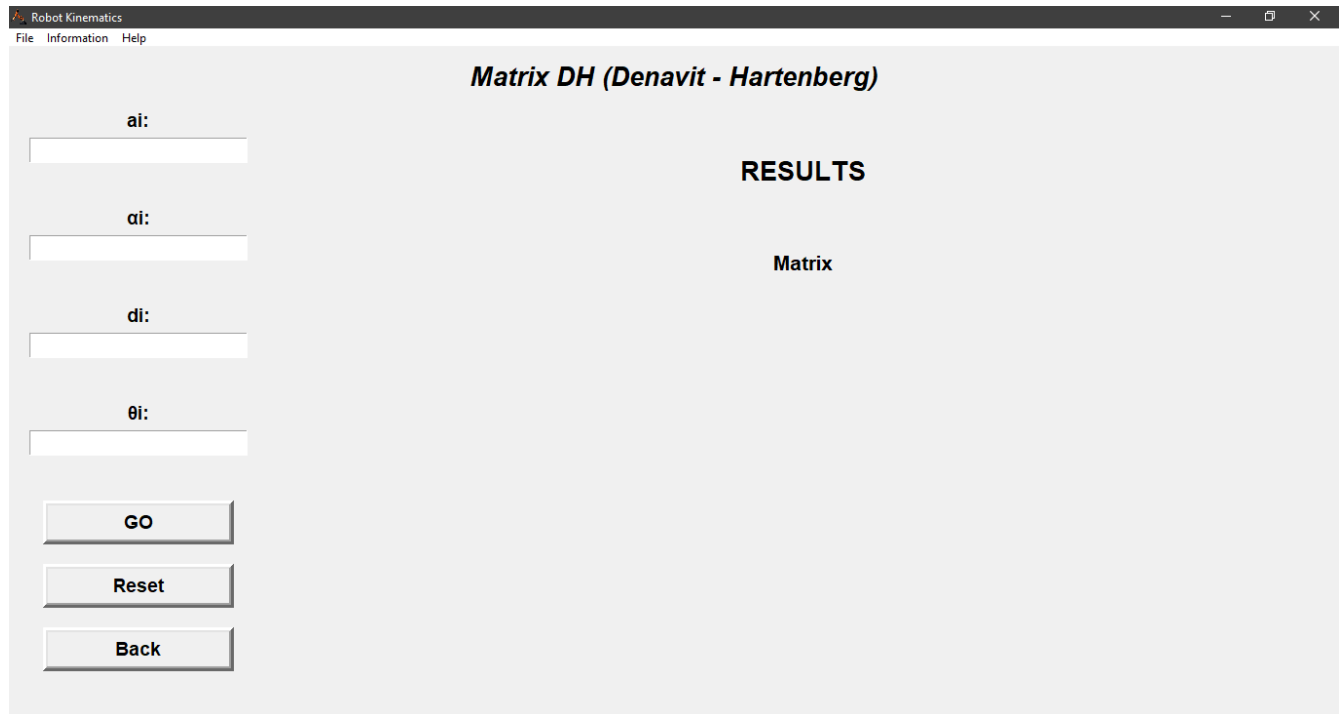


Ilustración 14. Venta de la matriz DH

Como podemos observar en la anterior ilustración que tenemos cuatro cajas de texto que es donde se ingresaran cada uno de nuestros parámetros.

La funcionalidad de los botones que se nos muestra es la siguiente:

- **GO:** Tiene la funcionalidad de ejecutar nuestra ventana con nuestros parámetros ya ingresados y gracias a como se fue programando la interfaz prácticamente tenemos una respuesta inmediata de nuestros resultados. Al dar clic en este botón nos va a mostrar un mensaje emergente que se puede ver en la *Ilustración 15. Mensaje emergente de la ventana de la matriz DH*.

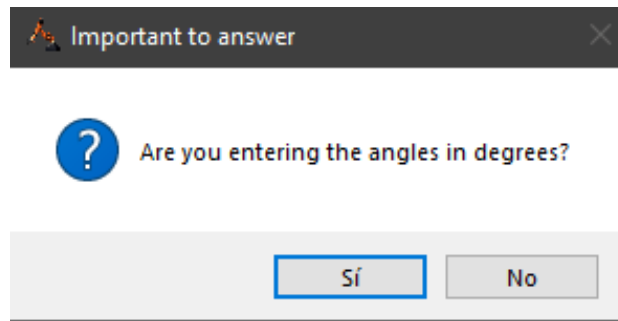


Ilustración 15. Mensaje emergente de la ventana de la matriz DH

Como podemos observar nos pregunta que, si estamos ingresando nuestros ángulos en grados, esto se debe ya que dos de nuestros cuatro parámetros de entrada son ángulos, dicha información se puede leer en el enlace que se hizo referencia anteriormente. Como la pregunta es muy específica dice “ángulos” lo cual nos indica que los dos ángulos deben ser grados o radianes para que nuestro programa funcione de manera correcta. Si damos clic en la opción de “sí” es porque estamos diciendo que ambos ángulos están en grados. Si damos clic en la opción de “no” es porque estamos diciendo que ambos ángulos están en radianes.

- Reset: Este botón tiene la funcionalidad de borrar todo dato que se haya modificado de la ventana original, lo que borra son: cajas de texto y la matriz que nos muestra como resultado.
- Back: Solamente tiene la funcionalidad de regresarnos una venta antes en la que estamos trabajando actualmente, en este caso la ventana a la que nos regresaría sería la ventana de transformaciones.

Para saber cómo ingresar de una manera correcta los valores de entrada, para que la interfaz no nos muestre ningún error al momento de dar clic en “GO”. Revise el ejemplo de cómo ingresar nuestros parámetros de manera correcta.

Ejemplo. Los parámetros que ingresaremos en las cajas de texto van a ser las siguientes:

$$a_i = 20$$

$$\alpha_i = 30^\circ$$

$$d_i = 50$$

$$\theta_i = 60^\circ$$

Solamente que en los ángulos Alpha y theta no se debe ingresar el icono representativo de los ángulos en grados, por esos pequeños detalles se recomienda que se va la **Ilustración 16. Ejemplo de la ventana de la matriz DH**. Que dicha ilustración también tendrá incluida el resultado de la matriz.

Robot Kinematics
File Information Help

Matrix DH (Denavit - Hartenberg)

ai:
20

ai:
30

di:
50

θi:
60

GO

Reset

Back

RESULTS

Matrix

$$\begin{bmatrix} 0.5 & -0.75 & 0.4330127 & 10. & \\ 0.8660254 & 0.4330127 & -0.25 & 17.32050808 & \\ 0. & 0.5 & 0.8660254 & 50. & \\ 0. & 0. & 0. & 1. & \end{bmatrix}$$

Ilustración 16. Ejemplo de la ventana de la matriz DH

Nota: Actualmente se está trabajando para que la ventana nos devuelva la matriz alineada y sea más fácil de leer visualmente.

1.2 Cinemática directa

En esta ventana estaremos mirando el segundo tema principal el cual tratará de entregarnos una matriz de 4x4 que dicha matriz es de una determinada de multiplicación de matrices de Denavit – Hartenberg, bueno en otras palabras la matriz resultante representa la matriz DH del extremo de nuestro manipulador que en dicha matriz podemos deducir las coordenadas del extremo de nuestro manipulador. También esta ventana viene acompañada con una representación gráfica de nuestro diagrama de alambre o también conocido como diagrama cinemático, esto se realiza para que el usuario se dé una idea cual va a ser la forma que tendrá su manipulador según los parámetros de DH que el mismo definió. Si se quiere ver como se ve originalmente esta ventana véase la *Ilustración 17. Ventana de Cinemática directa*.

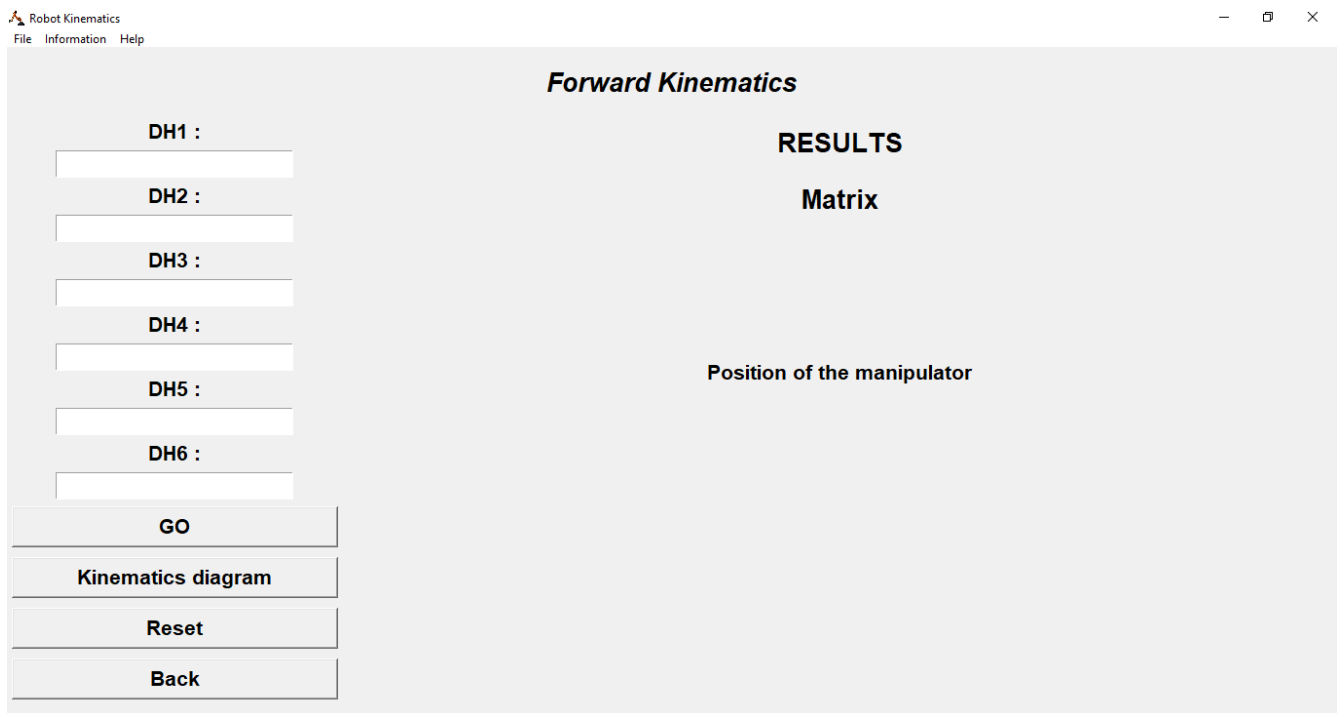


Ilustración 17. Ventana de Cinemática directa

Como se puede ver en la ilustración anterior tenemos seis cajas de texto, esto se diseñó principalmente pensando en los manipuladores industriales o robots industrial cuyos robots industriales en su mayoría son de seis grados de libertad, aunque se pueden utilizar menos grados de libertad si es que se requiere, no nos debe de marcar si ingresamos nuestro valores correctos en las cajas de texto, las cajas de texto están enumeradas por DH1, DH2, ..., DH6, a las cajas de texto se le debe de ingresar los parámetros de DH según el grado de libertad en el que vayamos. La programación que tiene esta ventana ya hace todos los cálculos para que al momento de que nos muestre la matriz resultante del extremo de nuestro manipulador nos la entregue de manera casi de inmediato. También en los resultados nos muestra la posición correcta del manipulador.

La funcionalidad de los botones que se muestran en la venta de cinemática directa es:

- **GO:** Tiene la funcionalidad de ejecutar los valores de cada una de las cajas de texto que le estamos ingresando la cual nos devolverá la matriz del extremo de nuestro manipulador que es lo que nos interesa. En el caso de esta ventana al dar clic en este botón no nos mostrará un mensaje emergente ya que cuando ingresemos nuestros parámetros podremos elegir si son grados o radianes lo cual se verá mejor explicado en el ejemplo.
- **Kinematics diagram:** Este botón tiene la funcionalidad de que una vez que ya hallamos dado clic en GO se aguardará ahí el diagrama cinemático del manipulador analizado a partir de los parámetros ingresados en las entradas disponibles en la ventana. La forma que nos va a mostrar el diagrama cinemático será en un espacio de trabajo en 3D para tener una mejor perspectiva de la posible posición del manipulador.
- **Reset:** Este botón tiene la funcionalidad de borrar todas las modificaciones de la ventana original como los datos escritos por el usuario en las cajas de texto y los resultados que entrega la ventana.

- Back: Este botón tiene la funcionalidad de regresar una ventana anterior a la cual estamos trabajando actualmente en este caso regresaremos a la ventana principal.

Ejemplo. En este caso usaremos de referencia un manipulador de seis grados de libertad haciendo una representación a un robot de la marca Fanuc. Los parámetros de muestra son los siguientes:

$$DH1 = 75, 90, 330, 33.6900675259798$$

$$DH2 = 300, 0, 0, 57.7632635578044$$

$$DH3 = 75, -90, 0, -48.5005956216640$$

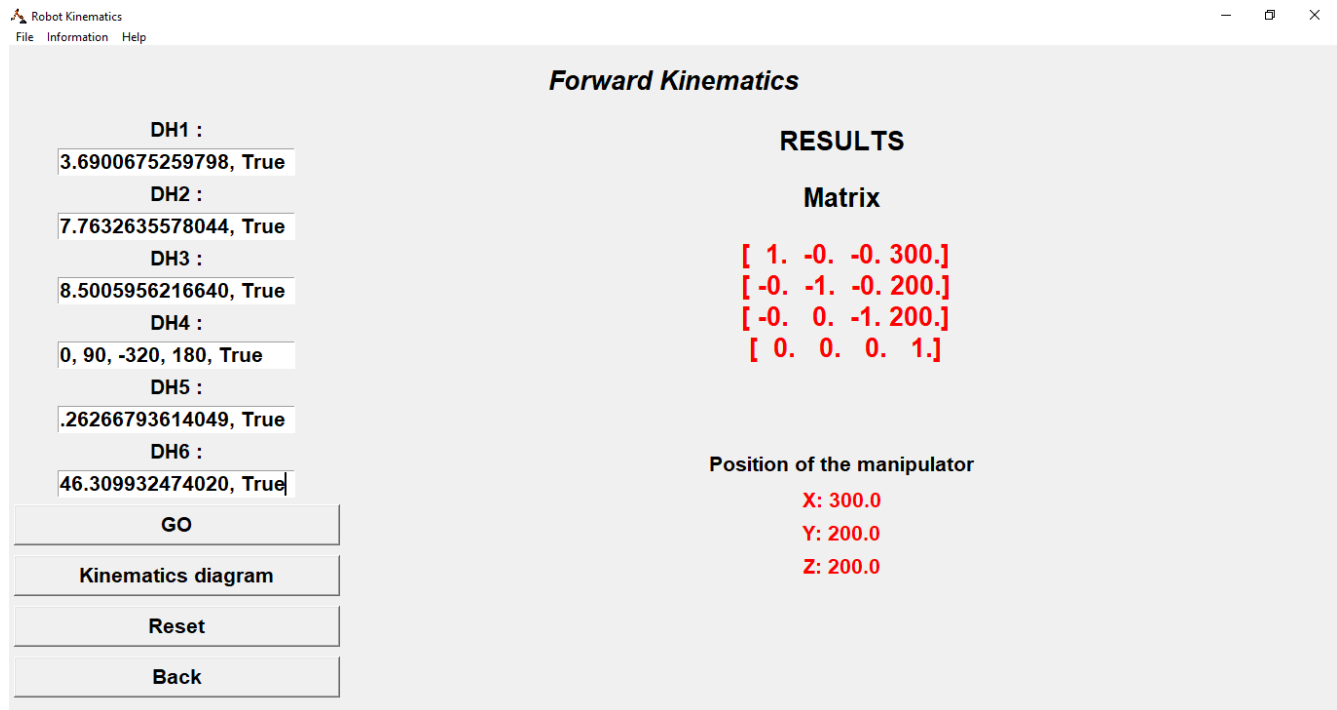
$$DH4 = 0, 90, -320, 180$$

$$DH5 = 0, 90, 0, 9.26266793614049$$

$$DH6 = 0, 0, 80, -146.309932474020$$

Como podemos observar en nuestros parámetros anteriores los ángulos están en grados por lo tanto cuando ingresemos estos parámetros en nuestras cajas de texto debemos de agregar un “True” esto nos indica que estamos ingresando grados, ya que si solo ingresamos los parámetros la interfaz lo va a interpretar que los ángulos están en radianes.

Para ver los resultados que nos manda esta ventana con los parámetros anteriores véase la **Ilustración 18. Ejemplo de la ventana de cinemática directa.**



Robot Kinematics
File Information Help

Forward Kinematics

DH1 :
3.6900675259798, True

DH2 :
7.7632635578044, True

DH3 :
8.5005956216640, True

DH4 :
0, 90, -320, 180, True

DH5 :
.26266793614049, True

DH6 :
46.309932474020, True

RESULTS

Matrix

[1. -0. -0. 300.]
[-0. -1. -0. 200.]
[-0. 0. -1. 200.]
[0. 0. 0. 1.]

Position of the manipulator

X: 300.0
Y: 200.0
Z: 200.0

GO

Kinematics diagram

Reset

Back

Ilustración 18. Ejemplo de la ventana de cinemática directa

Como podemos observar en la ilustración anterior de que la ventana nos muestra como resultado una matriz que tiene todo el análisis de todos los grados de libertad que hemos utilizado para en el

manipulador. Aunque si se tiene noción del tema se puede deducir las coordenadas del manipulador, pero se diseñó con una alternativa si se daba el caso que el usuario no tenga mucho conocimiento sobre el asunto por lo tanto la ventana muestra la posición del manipulador con sus tres ejes X , Y , Z .

Ahora veremos la funcionalidad del botón “*Kinematics diagram*” que prácticamente nos muestra el diagrama cinemático o diagrama de alambre de nuestro manipulador y podemos dar una idea como luciría. Para este ejemplo véase la ***Ilustración 19. Diagrama cinemático (1)*** y la ***Ilustración 20. Diagrama cinemático (2)***.

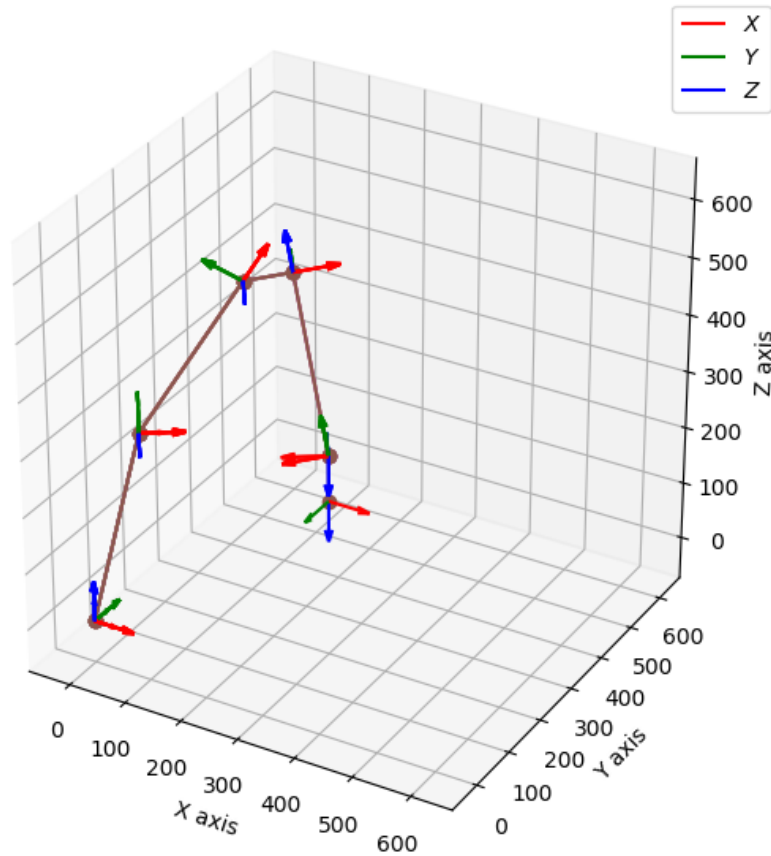


Ilustración 19. Diagrama cinemático (1)

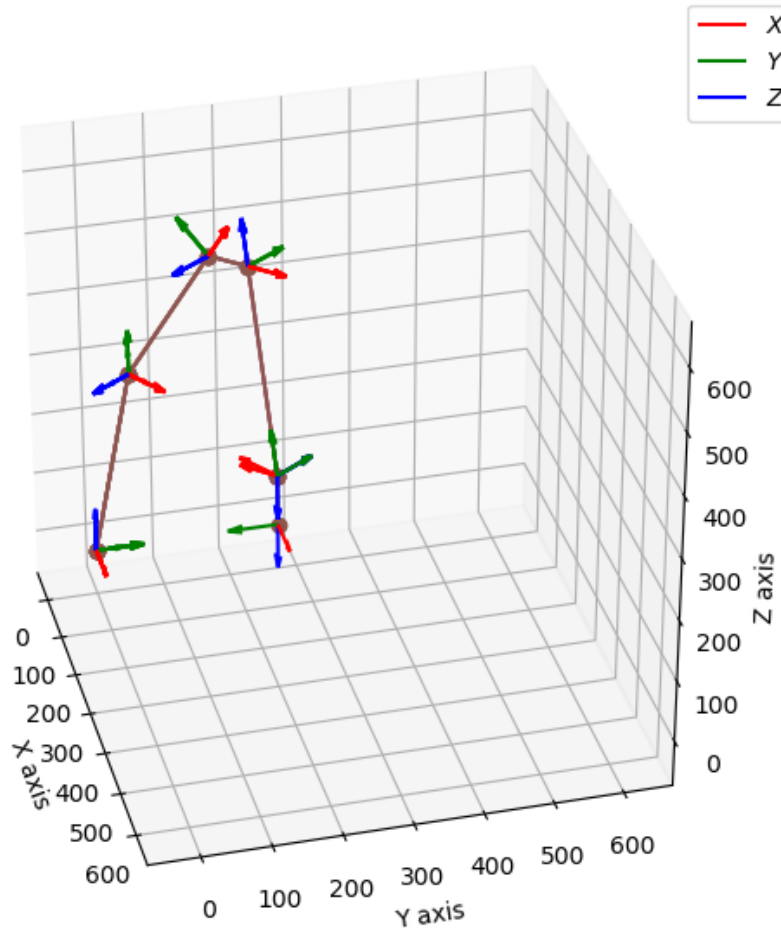


Ilustración 20. Diagrama cinemático (2)

Como podemos observar en las ilustraciones de nuestro diagrama cinemático nos muestra también los sistemas de ejes de cómo están orientados cada uno de nuestros grados de libertad. El espacio en 3D se programó para que luciera se forma interactiva y lo pudiéramos rotar y de esa manera poder ver diferentes tipos de perspectiva.

1.3 Cinemática inversa

En esta ventana tendremos la funcionalidad de realizar el análisis de un manipulador por cinemática inversa, esta ventana nos puede ofrecer dos métodos posibles de realizar este análisis que son: el método numérico de Newton – Raphson y la función “root” de la librería Scipy esta función realiza los cálculos matemáticos gracias al método “hybr”, aunque no es el único método con el que trabaja esta función se pueden usar más métodos si tiene alguno en específico.

Si quiere ver como luce la ventana originalmente véase la ***Ilustración 21. Ventana de cinemática inversa.***

Como podemos observar en la ilustración tenemos dos botones donde el botón que tiene “Cinemática inversa (N – R)” hace referencia en que ese botón nos va a mandar a la ventana donde nos hará la cinemática inversa por medio del método de Newton – Raphson. El botón que tiene el nombre de

“Cinemática inversa (mixed)” hace referencia a que nos mandara a la ventana donde nos hará la cinemática inversa por medio de la función de Scipy.

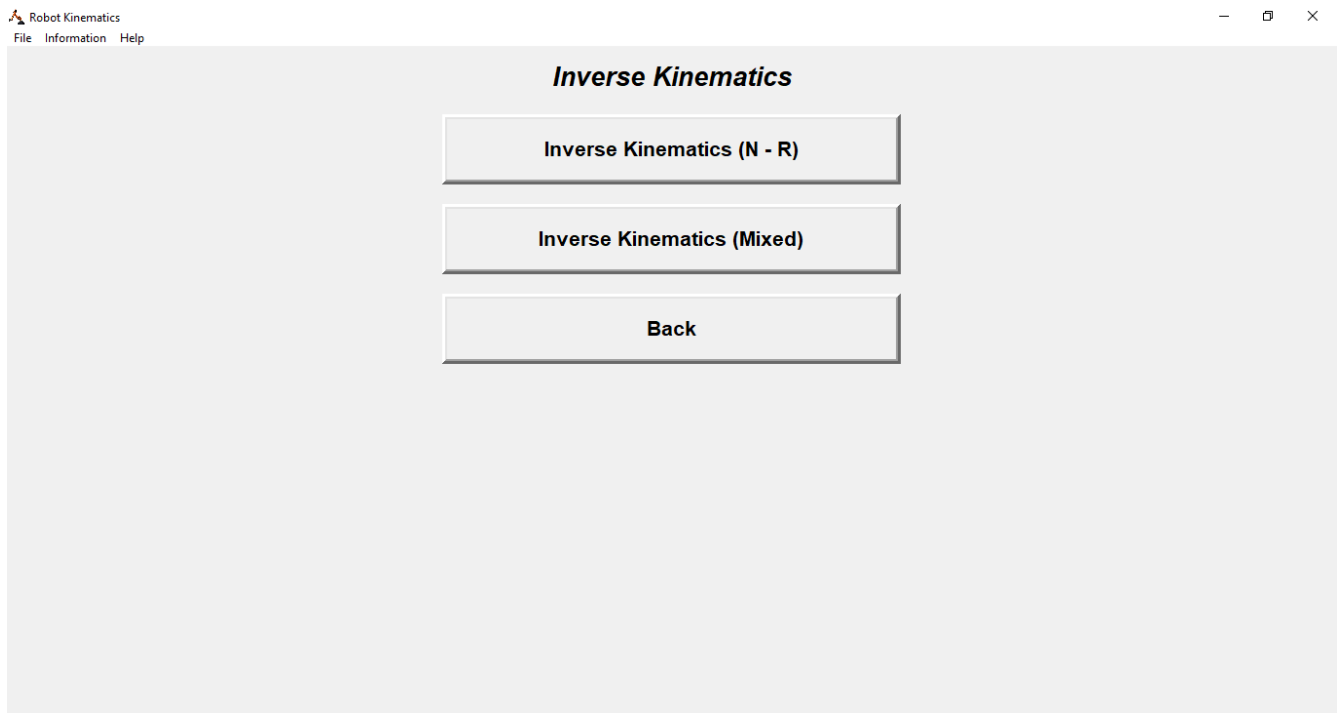


Ilustración 21. Ventana de cinemática inversa

1.3.1 Cinemática inversa por el método de Newton – Raphson

Como ya se comentó anteriormente, esta ventana nos hará el análisis de la cinemática inversa por medio del método de Newton – Raphson, pero tiene un pequeño inconveniente y es que debemos de ingresarle nosotros mismos la matriz jacobiana de nuestro análisis. Si se quiere ver esta ventana como luce originalmente véase la *Ilustración 22. Ventana de cinemática inversa por el método de Newton - Raphson*.

Inverse Kinematics for NEWTON - RAPHSON

J (Matrix J):

b (Matrix F):

Number of variables

Initials Values:

GO

Reset

Back

RESULTS

Approximate values:

Error range by variable:

Number of iterations:

Ilustración 22. Ventana de cinemática inversa por el método de Newton - Raphson

La funcionalidad de los botones es la siguiente:

- **GO:** Este botón tiene la funcionalidad de ejecutar los valores ingresados en las cajas de texto y es el que se encarga de dar la orden de que empiece hacer los determinados cálculos necesarios para esa ventana.
- **Reset:** Tiene la funcionalidad de borrar todo aquello modificado a la ventana original, las modificaciones que se le pueden hacer a la ventana son las cajas de texto y los resultados que muestra la ventana.
- **Back:** Este botón tiene la funcionalidad de regresar una ventana anterior que en este caso sería la ventana de cinemática inversa.

Como podemos ver en la ilustración anterior tenemos varias cajas de texto para ingresar nuestros valores a calcular. En esta ventana debemos de indicar el número de variables a calcular y deben de calcular en las ecuaciones ingresadas en “b (Matrix F)” y también el número de variables de coincidir en la matriz jacobiana que le ingresemos. El nombre de cada una de las variables ya está definido que son: var1, var2, ..., var6, para cuando se esté requiriendo el nombre de la variable se debe de poner dicho nombre a la hora de llenar nuestras cajas de texto.

Nota: Si se llegan a pedir en las ecuaciones que encuentre el valor de algún ángulo debe hacer lo siguiente: “Ingresar los valores iniciales en radianes” si no se hace los descrito anteriormente la interfaz le marcará error.

Esta interfaz nos manda como resultado los valores aproximados de las variables, error de las variables (entre más cercanas a cero es mejor) y el número de iteraciones en pocas palabras son las veces que la interfaz hizo el método para poder obtener los resultados reales de las variables.

Ejemplo. Para este problema supondremos que tenemos un manipulador RR (revoluta, revoluta) donde en ambos grados de libertad no conoceremos el valor de sus ángulos. Para este problema ya daremos a conocer la matriz jacobiana y la matriz b, solo para facilitar la explicación de este ejemplo.

$$J = \begin{bmatrix} -100\sin(var1) - 100\sin(var1 + var2) & -100\sin(var1 + var2) \\ 100\cos(var1) + 100\cos(var1 + var2) & 100\cos(var1 + var2) \end{bmatrix}$$

$$b = 100\cos(var1) + 100\cos(var1 + var2) - 100 \quad 100\sin(var1) + 100\sin(var1 + var2) - 100$$

Como podemos ver tenemos dos variables por lo tanto en la caja de opciones debemos elegir el número 2.

Los valores iniciales que también que vamos a ocupar son los siguientes: var1 = 1.047197551, var2 = 1.570796327. Estos valores iniciales de nuestros ángulos son iguales a 60° y 90° respectivamente.

La manera correcta de ingresar nuestras matrices en nuestras cajas de texto es la de la siguiente manera:

$$J = [-100 * \sin(var1) - 100 * \sin(var1 + var2), -100 * \sin(var1 + var2)], [100 * \cos(var1) + 100 * \cos(var1 + var2), 100 * \cos(var1 + var2)]$$

$$b = [100 * \cos(var1) + 100 * \cos(var1 + var2) - 100, 100 * \sin(var1) + 100 * \sin(var1 + var2) - 100]$$

$$x0 = 1.047197551, 1.570796327$$

Si se quiere comprobar que exactamente se ingresan los valores anteriores en la ventana véase la **Ilustración 23. Ejemplo de la ventana de cinemática inversa por el método de Newton - Raphson.**

Inverse Kinematics for NEWTON - RAPHSON

J (Matrix J):

b (Matrix F):

Number of variables

Initials Values:

RESULTS

Approximate values:
 Variable 1 = -0.0
 Variable 2 = 90.0

Error range by variable:
 Variable 1 = 0.0
 Variable 2 = -0.0

Number of iterations:
 7

GO
Reset
Back

Ilustración 23. Ejemplo de la ventana de cinemática inversa por el método de Newton - Raphson

1.3.2 Cinemática inversa por la función de Scipy

En esta ventana vamos a realizar en análisis de cinemática inversa, pero por medio de una función llamada “root” de la librería de Scipy. La ventaja de usar esta ventana respecto a la ventana por medio del método de Newton – Raphson es que con esta función nos evitamos hacer un análisis extra que sería el de crear nuestra propia matriz jacobiana, mientras que en esta ventana como es claro la ventaja que tenemos es que nos evitamos la molestia de crear nuestra propia matriz jacobiana. También otra gran ventaja es que la misma ventana nos evitamos otra pequeña molestia la cual es definir el número de variables que se tiene en las ecuaciones.

Si se quiere tener un conocimiento mejor sobre la utilización de esta función le recomendamos que revise el siguiente enlace: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.html>

Si se quiere ver como luce la ventana originalmente véase la **Ilustración 24. Ventana de cinemática inversa por la función root**. Si podemos observar en la ilustración que se hizo referencia anteriormente podemos ver que solo tenemos dos cajas de texto las cuales nos pide únicamente nuestras ecuaciones y los valores iniciales para esas variables.

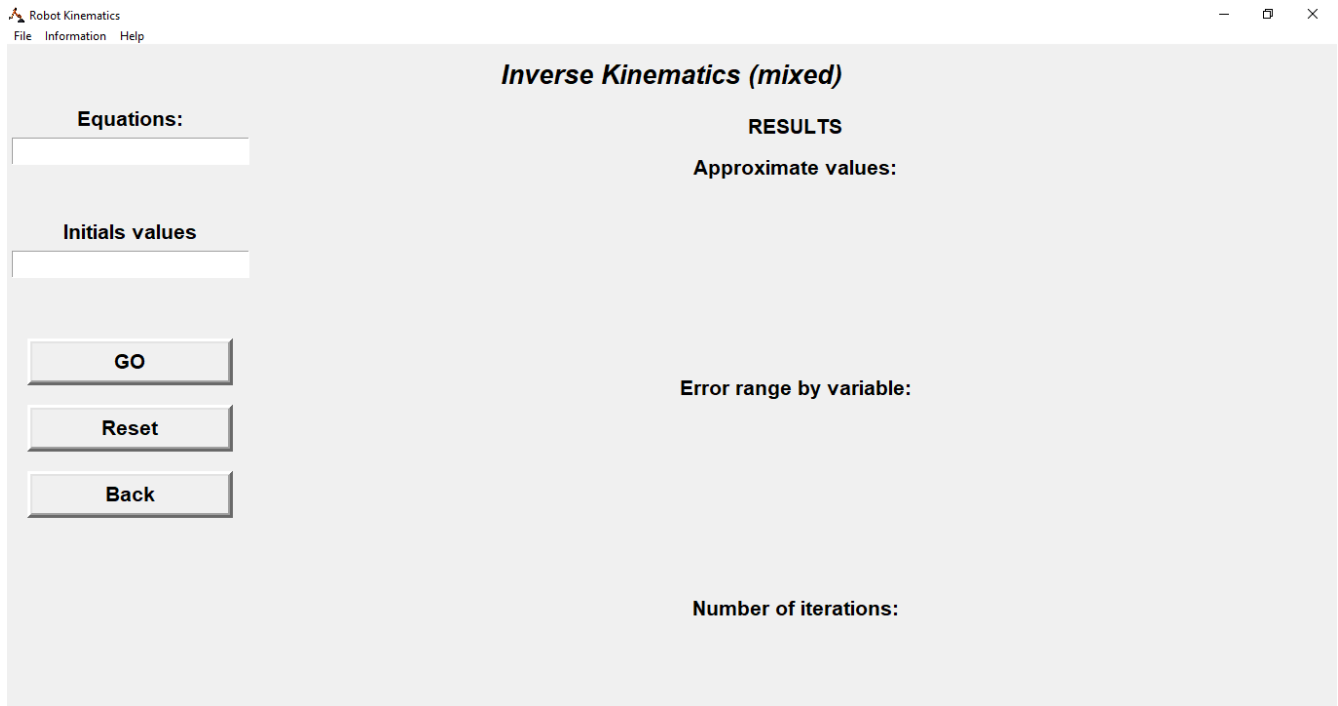


Ilustración 24. Ventana de cinemática inversa por la función root

La funcionalidad de los botones es:

- **GO:** Ejecuta el programa con los valores ingresados que le estamos dando en nuestras cajas de texto. En esta ocasión al dar clic en este botón no nos mandará ningún mensaje emergente.
- **Reset:** Nos borra todas aquellas modificaciones realizadas en la ventana a comparación como estaba originalmente, estas modificaciones pueden ser en las cajas de texto y en todos nuestros resultados mostrados por la ventana.
- **Back:** Este botón tiene la funcionalidad de regresarnos a una ventana anterior y que en este caso sería la ventana de cinemática inversa.

En esta ventana al momento de ingresar nuestras ecuaciones será en forma de una vector de una fila por n número de columnas si es que se quiere ver en forma de matriz, en este caso al ingresar nuestras variables por así decirlo ya tienen un nombre definido que sería x , pero como es una sola variable para definir el número de variable debemos escribir las variables de la siguiente forma: $x[0]$, $x[1]$, ..., $x[n]$, esta ventana solo debe cumplir una condición que se deba cumplir el mismo número de variables que de ecuaciones.

Nota: Si se llegan a ingresar variables de ángulos debemos de ingresar los valores iniciales en radianes y los resultados por default nos lo regresa en radianes.

Ejemplo. Las ecuaciones con sus respectivas variables son las siguientes:

$$\text{Ecuación 1} = x[0]\cos(x[1])\sin(x[2])$$

$$\text{Ecuación 2} = x[0]\sin(x[1])\sin(x[2])$$

$$\text{Ecuación 3} = 10 - (x[0]\cos(x[2]))$$

Nuestros valores iniciales serían los siguientes:

$$X0 = 10, \frac{\pi}{6}, \frac{\pi}{4}$$

Y la forma correcta de ingresar nuestros valores a nuestras cajas de texto es de la siguiente forma:

- Las ecuaciones los escribimos nuestras tres ecuaciones en la misma caja de texto, pero solo separamos a cada ecuación por una coma como a continuación se presenta:

$$x[0] * \cos(x[1]) * \sin(x[2]), x[0] * \sin(x[1]) * \sin(x[2]), 10 - (x[0] * \cos(x[2]))$$

- Los valores iniciales exactamente como los escribimos anteriormente así se escriben.

$$80, \pi/6, \pi/4$$

Si se quiere ver que exactamente se ingresan los valores a las cajas de texto como se acaba de explicar anteriormente revise la *Ilustración 25. Ejemplo de la venta de cinemática inversa por medio de la función root*.

The screenshot shows the 'Inverse Kinematics (mixed)' window in the Robot Kinematics software. The interface is divided into several sections:

- Equations:** A text box containing the equations: $x[2]), 10-(x[0]*\cos(x[2]))$.
- Initials values:** A text box containing the initial values: $80, \pi/6, \pi/4$.
- Buttons:** Three buttons labeled 'GO', 'Reset', and 'Back' are located below the initial values.
- RESULTS:** A section on the right side of the window showing the results of the calculation.
 - Approximate values:**
 - Variable 1 = 10.0
 - Variable 2 = 0.5236
 - Variable 3 = -0.0
 - Error range by variable:**
 - Variable 1 = -0.0
 - Variable 2 = -0.0
 - Variable 3 = -0.0
 - Number of iterations:** 17

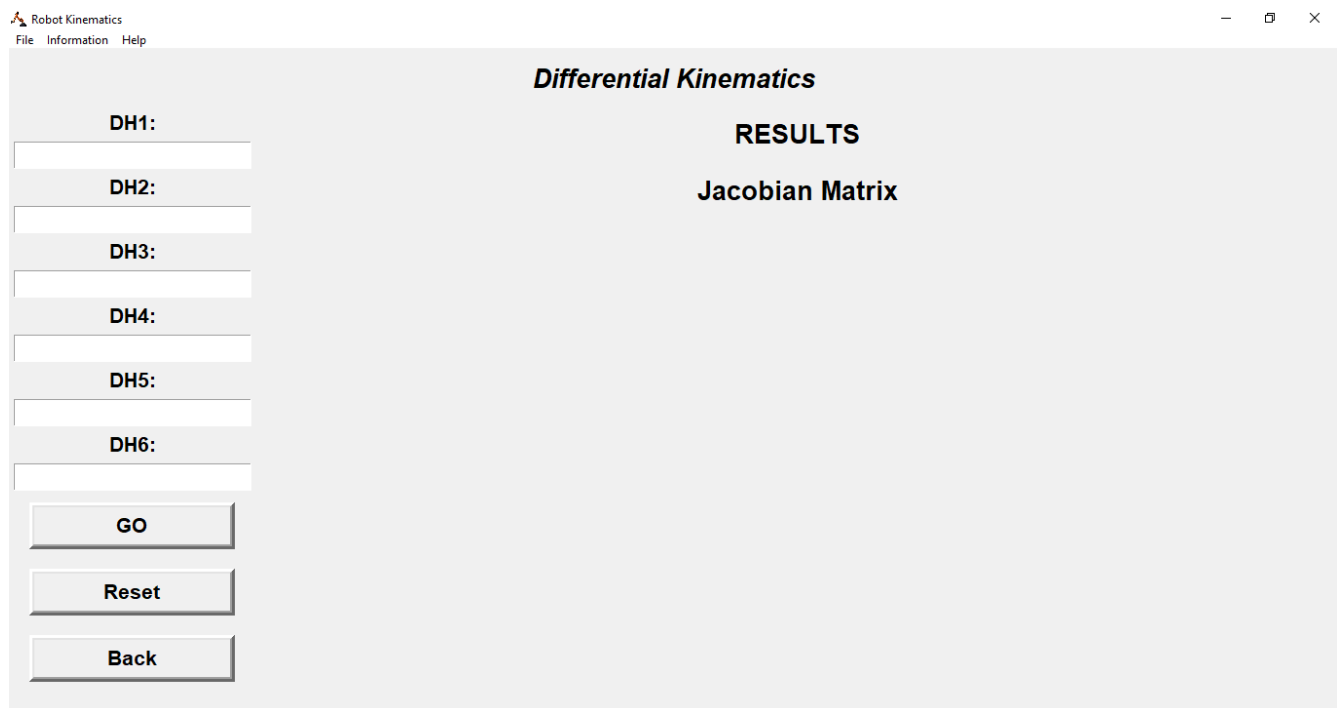
Ilustración 25. Ejemplo de la venta de cinemática inversa por medio de la función root

Consejo: Si el error de cada una de las variables es más cercano a cero es que es más correcto y si el error efectivamente es cero es porque la aproximación del valor de la variable efectivamente es el valor correcto con el que pudiéramos resolver nuestras ecuaciones.

1.4 Cinemática diferencial

En esta ventana tendremos la funcionalidad de poder calcular la matriz jacobiana del manipulador que estemos analizando. Los valores de entrada que le debemos entregar a la ventana son los parámetros de DH por cada uno de los grados de libertad que tenemos, con la matriz jacobiana podemos obtener demasiados datos como las velocidades de las articulaciones del robot ya sean revolutas o primaticos, también podemos obtener las velocidades del extremo del manipulador. Si se quiere tener un poco más de información le recomendamos que revise el siguiente enlace: <http://wiki.robotica.webs.upv.es/wiki-de-robotica/cinematica/>.

Para poder visualizar la ventana como es originalmente véase la **Ilustración 26. Ventana de cinemática diferencial**.



The screenshot shows a software window titled "Robot Kinematics" with a menu bar containing "File", "Information", and "Help". The main content area is titled "Differential Kinematics" and "RESULTS Jacobian Matrix". On the left side, there are six input fields labeled "DH1:", "DH2:", "DH3:", "DH4:", "DH5:", and "DH6:". Below these fields are three buttons: "GO", "Reset", and "Back".

Ilustración 26. Ventana de cinemática diferencial

La funcionalidad de los botones es:

- GO: Este botón tiene la funcionalidad de ejecutar nuestro programa para que la interfaz realice los cálculos debidos con respecto a los parámetros que le estemos ingresando.

- Reset: Este botón tiene la funcionalidad de borrar todas aquellas modificaciones que se le hacen a la ventana principal como los valores ingresados a nuestras cajas de texto y la matriz jacobiana resultante.
- Back: Este botón tiene la funcionalidad de regresarnos una ventana anterior a la actual donde estamos trabajando en este caso nos regresaría a la ventana principal de la interfaz.

Al momento que vayamos a ingresar nuestros parámetros DH debemos de tener cuidado en especial cuando sean los ángulos ya que debemos ingresarlo siempre en radianes para que nos pueda entregar correctamente nuestros resultados. Los parámetros que les estamos dando deben de ser ya conocidos y también debemos de definir qué tipo de junta o par cinemático es, si es revoluta o prismático solo debemos de ingresar después de nuestros cuatro parámetros debemos de poner una coma y después un r si es revoluta o una p si es prismático, estas letras se deben de acompañar con unas comillas, por ejemplo: 'r' o "r". Si no se llega a poner ninguna de las dos letras la interfaz lo interpretará como una revoluta.

Ejemplo. Para este caso utilizaremos únicamente dos grados de libertad y que será un manipulador RRP (revoluta, revoluta, prismático). Para parámetros que vamos a ingresar en nuestras cajas de texto son las siguientes:

$$DH1 = 0, 1.570796327, 100, 0.785398163, 'r'$$

$$DH2 = 0, 1.570796327, 0, 1.047197551, 'r'$$

$$DH3 = 0, 0, 150, 0, 'p'$$

Como se acaban de escribir anteriormente los parámetros es la misma forma que vamos a ingresarlos a nuestras cajas de texto.

Si se quiere ver es exactamente igual y ver cuál es el resultado que nos muestra nuestra ventana le recomendamos que ve la *Ilustración 27. Ejemplo de la ventana de cinemática diferencial*.

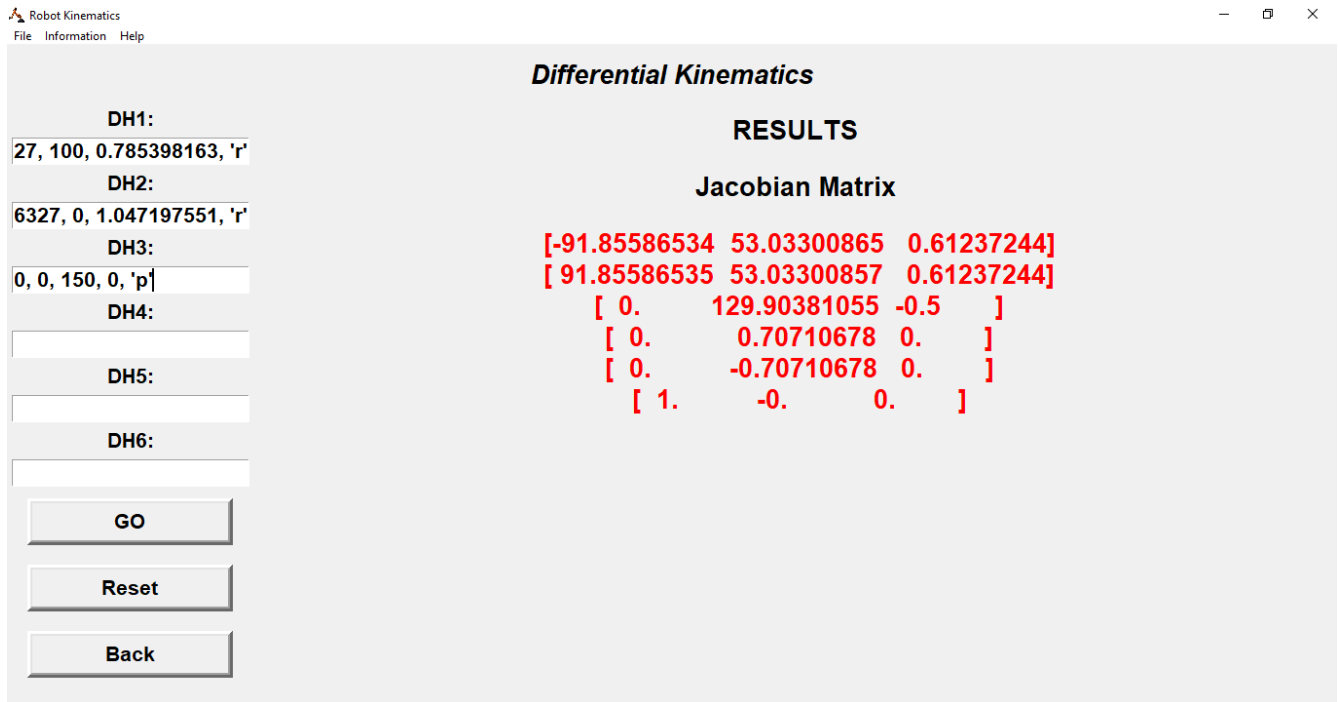


Ilustración 27. Ejemplo de la ventana de cinemática diferencial

Como podemos observar la ventana de la interfaz nos devuelve una matriz jacobiana de la cual podemos obtener los datos que necesitemos y de esa manera poder obtener la matriz de una manera más rápida.