



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN EN
LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4
ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION**

AUTOR: Carlos Babiera Sancho

TUTOR: Juan Manuel Herrero Durá

Curso Académico: 2015-16

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

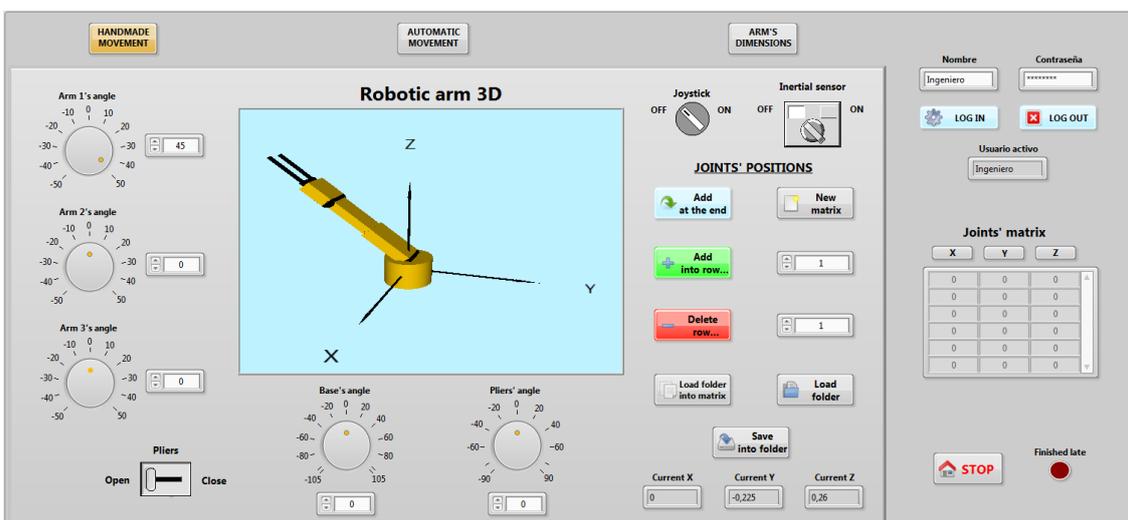
RESUMEN

En el presente proyecto se muestra el diseño de una aplicación programada en LabVIEW, con la cual se controla la posición del brazo robot S300105 de 5 ejes y 4 articulaciones desarrollado por Lynxmotion, Inc.

Esta aplicación permite el movimiento manual mediante la cinemática directa. También realiza la generación de trayectorias y el movimiento manual con un joystick usando la cinemática inversa.

Adicionalmente, se ha añadido una unidad inercial (IMU) CC3D para proporcionar al brazo robot una funcionalidad extra: Imitación de los giros aplicados al sensor.

Palabras clave: Brazo robot, sensor inercial, IMU, joystick, LabVIEW, aplicación, trayectorias, pinza, servomotores, control de la posición, cinemática inversa, cinemática directa.



Equipo utilizado y aplicación desarrollada.

DESENROTLLAMENT D'UNA APLICACIÓ EN LABVIEW PER AL CONTROL DE TRAJECTÒRIES DEL BRAÇ ROBOT, DE 4 ARTICULACIONS AMB PINÇA, MODEL S300105 DE LYNXMOTION

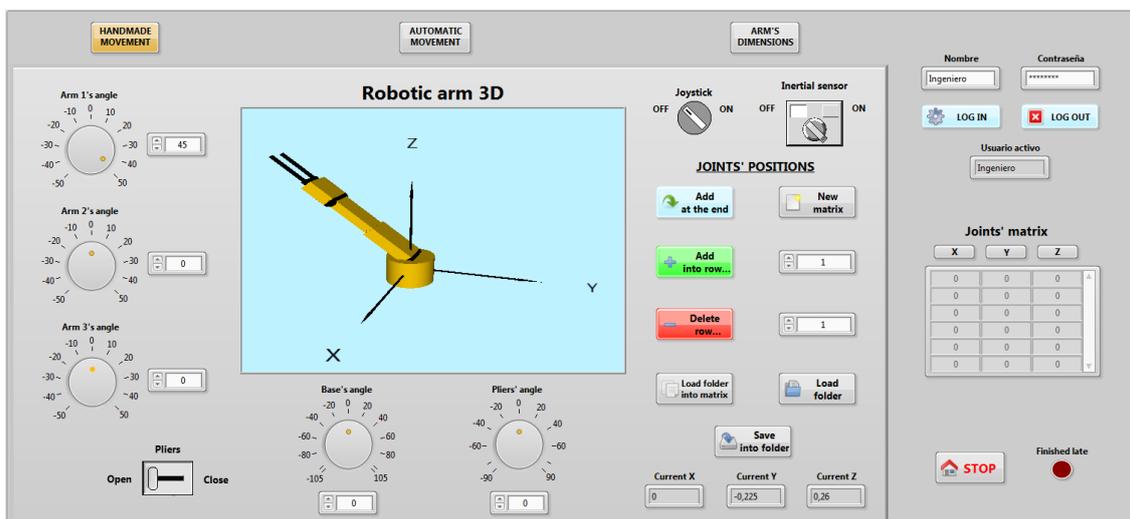
RESUM

En el present projecte es mostra el disseny d'una aplicació programada en LabVIEW, amb la qual es controla la posició del braç robot S300105 de 5 eixos i 4 articulacions desenvolupat per Lynxmotion, Inc.

Esta aplicació permet el moviment manual per mitjà de la cinemàtica directa. També realitza la generació de trajectòries i el moviment manual amb un joystick usant la cinemàtica inversa.

Adicionalment, s'ha afegit una unitat inercial (IMU) CC3D per a proporcionar al braç robot una funcionalitat extra: Imitació dels girs aplicats al sensor.

Paraules clau: Braç robot, sensor inercial, IMU, joystick, LabVIEW, aplicació, trajectòries, pinça, servomotors, control de la posició, cinemàtica inversa, cinemàtica directa.



Equip utilitzat y aplicació desenvolupada.

LABVIEW APPLICATION DEVELOPMENT FOR ROBOTIC ARM (4 JOINTS AND PLIERS) TRAJECTORY CONTROL, LYNXMOTION MODEL 2300105

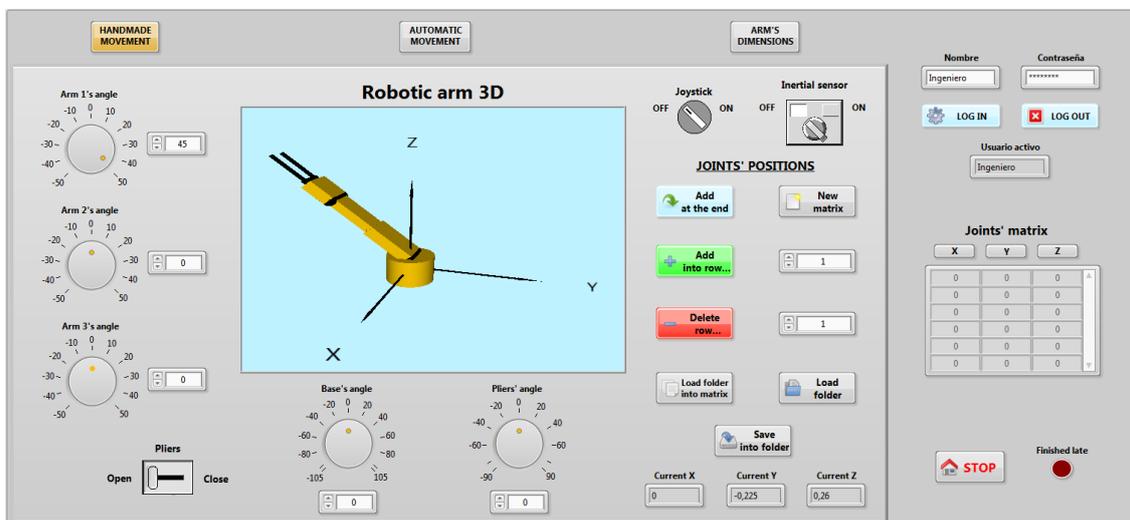
ABSTRACT

A LabVIEW design application is shown in this project in order to control the robotic arm position (S300105, 5 axis and 4 joints, developed by Lynxmotion, Inc).

This application allows the manual movement through forward kinematics. It also generates the trajectory and the manual movement with a Joystick using inverse kinematics.

Besides, an inertial unit (IMU) CC3D has been added to provide an extra functionality: Imitation of the turns applied to the sensor.

Keywords: Robotic arm, inertial sensor, IMU, joystick, LabVIEW, application, trajectory, pliers, servomotors, position control, inverse kinematics, forward kinematics.



Equipment used and application developed.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Contenido

1. Memoria
2. Presupuesto
3. Anexo de programación
4. Manual de usuario
5. Bibliografía

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA
APLICACIÓN EN LABVIEW PARA
EL CONTROL DE TRAYECTORIAS
DEL BRAZO ROBOT, DE 4
ARTICULACIONES CON PINZA,
MODELO S300105 DE
LYNXMOTION

Documento Nº1

MEMORIA

Autor: Carlos Babiera Sancho

Tutor: Juan Manuel Herrero Durá

Curso académico: 2015-16

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Contenido

1. Objeto	1
1.1 Especificaciones de diseño	1
2. Antecedentes	2
2.1 Descripción del brazo robot	2
2.2 Comunicación con servomotores	3
2.3 Sensor inercial (IMU)	4
2.4 Joystick	5
2.5 LabVIEW	6
2.6 Toolbox Robotics	7
2.7 Módulo DSC	8
2.8 Cinemática directa	9
2.9 Cinemática inversa	10
2.10 Planificación de trayectorias	11
3. Descripción de la solución adoptada	13
3.1 Estructura de la aplicación	13
3.2 Cinemática directa	15
3.3 Cinemática inversa	17
3.4 Control manual	22
3.4.1 Digital	22
3.4.2 Joystick	22
3.5 Seguimiento de trayectorias	24
3.6 Unidad inercial	25
3.7 Seguridad	27
3.8 Variables compartidas y uso mediante dispositivo Tablet	29
4. Conclusiones	30
4.1 Resultados obtenidos	30
4.2 Cambios en el diseño	30
4.3 Posibles mejoras	31

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

1. Objeto

La presente memoria tiene como objetivo introducir el equipo utilizado (brazo robot, aplicación, sensor y joystick) y los problemas a tratar del movimiento del brazo robot en la aplicación (generación de trayectorias y cinemática directa e inversa).

Asimismo también se discutirá la comunicación de los datos de los periféricos, la solución al problema de la cinemática directa e inversa adoptada, la generación de trayectorias, el sistema de seguridad y el control remoto de la aplicación desde un dispositivo Tablet.

El documento finalizará con las conclusiones obtenidas tras realizar este Trabajo Final de Grado, estudiando qué objetivos se han cumplido, qué cambios en el diseño han habido y qué posibles mejoras tendría el programa.

1.1 Especificaciones del diseño

Esta aplicación debe:

- Estar realizada por el software LabVIEW.
- Ser capaz de realizar un movimiento manual tanto en forma digital (a través de la aplicación) como con el uso de un Joystick.
- Ser capaz de guardar puntos en el espacio o bien cargarlos desde un fichero para realizar el seguimiento de trayectorias.
- Ser capaz de imitar los giros de un sensor inercial.
- Ser capaz de representar el brazo y sus movimientos en un entorno 3-D.
- Ser capaz de representar los puntos de la trayectoria a seguir en el espacio para un mejor seguimiento, así como generar una matriz de puntos de la trayectoria realizados para poder compararlos con los puntos originales y ver así el error cometido en la cinemática inversa.
- Tener un registro de usuarios con sus respectivos privilegios.
- Ser capaz de controlar una parte del programa mediante un dispositivo Tablet.
- Poder cambiar las dimensiones del brazo robot.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2. Antecedentes

A continuación se van a detallar los puntos más importantes que se tratarán en la aplicación, así como el equipo usado.

2.1 Descripción del brazo robot

El brazo robot utilizado es un brazo robot de 5 ejes y de 603 gramos de peso (sin baterías) desarrollado por Lynxmotion, Inc. (Correcher, 2005). Permite la rotación de la base, hombro, codo, muñeca y la apertura/cierre de una pinza.

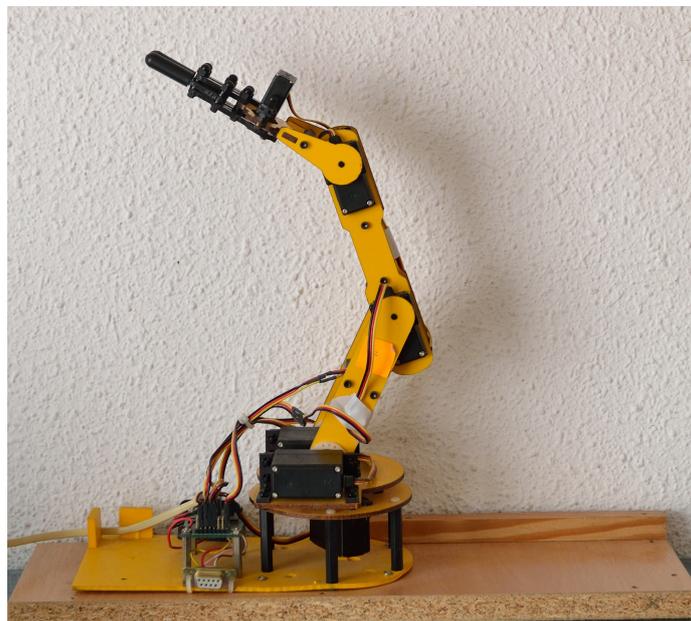


Figura 1. Brazo robot desarrollado por Lynxmotion, Inc.

El brazo es controlado por un circuito SSC (Serial Servo Controller) que puede llegar a controlar 8 servos independientes desde un puerto serie RS232 de un PC.



Figura 2. Placa SSC.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

La placa SSC se configura con tres jumpers, de manera que se pueden variar las siguientes características:

Jumper	Función	Off	On	Descripción
R	Rango	90º	180º	Rango de giro de las articulaciones
I	Identificación	0-7	8-15	Direcciones de los servos
B	Baud rate	2400	9600	Velocidad de información que transmite

Tabla 1. Posibles configuraciones de los jumpers de la placa SSC.

En este caso se ha usado un rango de 180º, una identificación de 0-7 y un "Baud rate" de 9600.

Cabe destacar que al poder cambiarse la identificación de los servos, la placa SSC permite el control de dos robots mediante el mismo cable serie.

La placa SSC consume 4.7 mA y los servos del brazo robot pueden llegar a consumir hasta 130 mA de corriente cada uno y 6 Voltios.

2.2 Comunicación con servomotores

La comunicación con los servomotores se realiza mediante puerto serie RS232 a un PC. En el envío de información se usan 8 bits de datos, sin paridad y 1 bit de stop.

Para mover un servo, se mandan los bits de sincronización (255), de dirección del servo (0-7 ó 8-15) y la posición del servo (0-254).

Los bits que nos informan de la posición del servo se envían a través de la placa SSC. Ésta transforma la información a un pulso PWM (que corresponderá a un ángulo dependiendo del tiempo en On y Off de la señal) y envía dicho pulso al servo motor. El servo mediante un bucle cerrado controla la posición con un regulador PI:

PWM

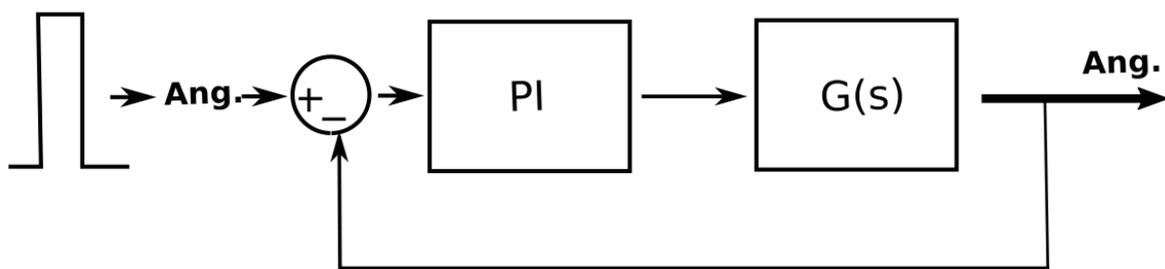


Figura 3. Bucle cerrado del control del ángulo en el servomotor.

De esta manera cada servo regula la posición requerida cuando se le envían los bits de información.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.3 Sensor inercial (IMU)

Se trata de un sensor modelo CC3D (*Jietai, 2015*) que consta de 3 giroscopios y 3 acelerómetros. Envía datos al ordenador vía mini USB.



Figura 4. Sensor inercial modelo CC3D.

La información se envía por protocolo UAV-talk. Con un código JavaScript se transforma la información a código ASCII y se elige la información que se quiere enviar (en nuestro caso, los ángulos de Euler en los tres ejes) y los envía en forma de UDP.

De esta manera que se envían paquetes de información UDP con un periodo de 50ms una cadena de caracteres ("string") de los ángulos de Euler en los ejes $[X, Y, Z]$ separados por comas.

Con las herramientas de LabVIEW adecuadas se transforman en tres números tipo "float" que se usarán posteriormente.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.4 Joystick

El modelo que se ha utilizado es un "Cyborg evo force" de Saitek (*Saitek, s.f.*). Este modelo permite el uso de 4 movimientos (se utilizarán todos) y 12 botones (de los cuales se usará únicamente uno) y envía la información vía USB.



Figura 5. Joystick Cyborg evo force desarrollado por Saitek.

El rango de movimientos son de 16 bits (es decir, se recibe una información desde -32767 hasta +32768), y el de los pulsadores 1 bit (se recibe un 1 mientras está apretado y un 0 cuando no lo está).

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.5 LabVIEW

LabVIEW es un programa que permite diseñar aplicaciones con un lenguaje de programación visual. Es un software creado por National Instruments pensado para el diseño de aplicaciones de control simulado usando lenguaje G (Lajara y Pelegrí, 2011).

Este programa se divide en dos componentes clave: Por un lado, el "Block Diagram" muestra el código que se ha implementado, y por otro el "Front Panel" muestra la interfaz gráfica generada en el "Block Diagram".

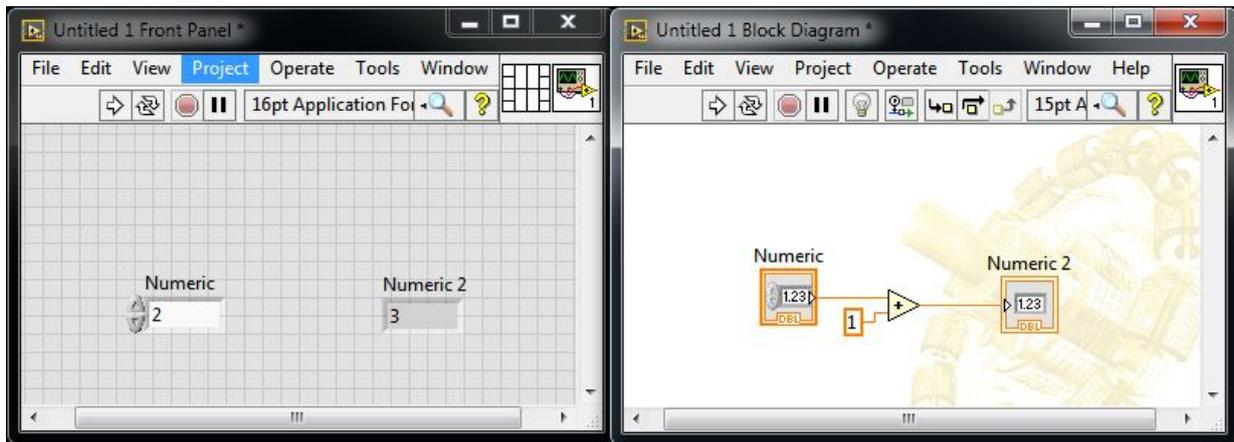


Figura 6. Front Panel (izquierda) y Block Diagram (derecha) de Labview.

Las aplicaciones creadas con este programa se denominan "VI" (Virtual Instruments), ya que se trata de la extensión de los archivos creados.

Se ha elegido este programa por los siguientes motivos:

- Pese a tener un lenguaje de programación simple e intuitivo, es un programa con un gran potencial para realizar tanto pequeños como grandes proyectos.
- Debido a sus muchas posibilidades y su versatilidad, siempre es posible mejorar el programa si se añaden más opciones u optimizando la interfaz del usuario ("Front Panel").
- Se puede simplificar mucho la visualización del diseño, ya que es posible que los VIs llamen a su vez a otro subVIs.
- Tiene la opción de añadir toolboxes y módulos (extensiones del programa) para aumentar su funcionalidad.
- Es muy potente: se puede trabajar a la vez con varios bucles con un bajo periodo con centenares de entradas y salidas.
- Es compatible con una gran cantidad de sensores, servos... Etc.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.6 Toolbox Robotics

Esta toolbox permite a LabVIEW trabajar más adecuadamente con brazos robot, sensores, arduino... Etc, además de poder representarlo en un entorno 3D.

También se añaden una serie de VIs que trabajan con sus opciones más básicas: calcular cinemática inversa, cinemática directa, generación de trayectorias, obtener la imagen en tres dimensiones de un brazo dadas las longitudes de sus brazos... Etc.

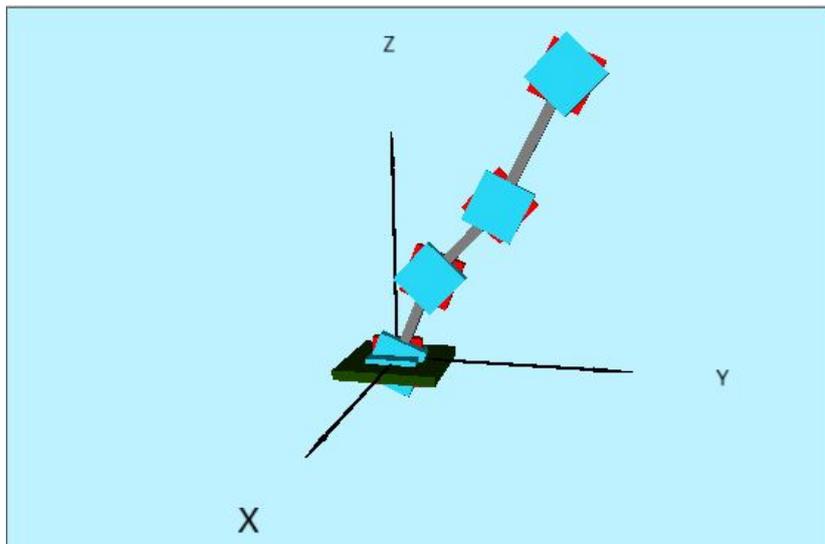


Figura 7. Esqueleto del brazo robot obtenido con Robotics.

Gracias a este módulo se pueden agilizar muchos cálculos y evitar muchos problemas.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.7 Módulo DSC

Este módulo permite un nuevo repertorio de opciones de interfaz de usuario y además permite añadir las cuentas de seguridad a la aplicación.

De esta manera el programa podrá ser controlado en mayor o menor medida dependiendo de qué usuario esté conectado en ese momento. En este programa se han creado tres niveles de seguridad: Operario, técnico e ingeniero. Se puede ver quién está conectado en todo momento además de poder conectarse o desconectarse.



Figura 8. Menú de seguridad de la aplicación.

Si el programa no está funcionando con uno de esos tres usuarios, no se podrá realizar ninguna acción de interés en el programa.

Los privilegios de cada usuario se explicarán más adelante.

2.8 Cinemática directa

El problema de la cinemática directa consiste en obtener los puntos $[x, y, z]$ del espacio a partir de los ángulos del brazo robot (Hernández, Ortiz, Calles y Rodríguez, 2015). Existen diversos métodos para la solución de la cinemática directa, como por ejemplo las matrices de transformación homogénea o la transformación de coordenadas (método Denavit – Hartenberg).

No obstante se ha escogido por simplicidad la resolución por geometría, es decir, la obtención del punto en el espacio mediante la trigonometría aplicada al brazo robot. Esto es, en el caso de un brazo con dos articulaciones:

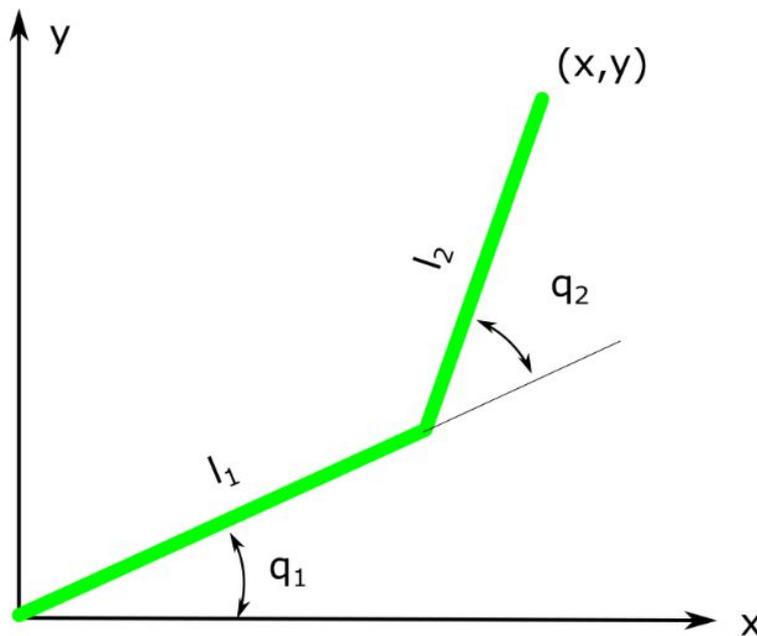


Figura 9. Ejemplo de un brazo con dos articulaciones.

$$x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2)$$

$$y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2)$$

Ecuación 1. Cinemática directa plana aplicada a un brazo de dos articulaciones (método geométrico).

2.9 Cinemática inversa

El objetivo de la cinemática inversa es la obtención de los ángulos del brazo robot a partir de un punto del espacio $[x, y, z]$ dado (Craig, 2006).

Nuevamente existen diferentes formas de resolver este problema, como por ejemplo soluciones numéricas (iterativas), desacoplamiento cinemático... Etc, pero se ha vuelto a escoger un método geométrico por simplicidad.

No obstante, cabe destacar que al contrario que en la cinemática directa, existen varias soluciones para un mismo punto, como mostramos en la siguiente imagen:

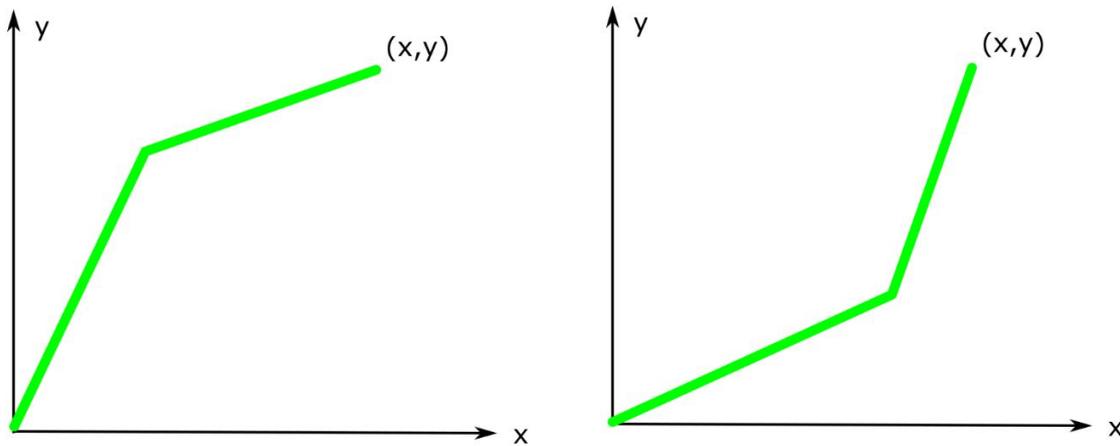


Figura 10. Ejemplo de dos posibles soluciones (mismo punto con diferentes ángulos).

Esto se debe a que el sistema de ecuaciones es compatible indeterminado, es decir, hay más incógnitas por resolver que ecuaciones. Es esto por lo que más adelante se supondrá una parte de la solución para ver si se llega a una solución lógica del problema.

2.10 Planificación de trayectorias

La planificación de trayectorias consiste en que el brazo robot realice unos movimientos adecuados y seguros entre dos puntos del espacio. Esto se consigue usando trayectorias continuas, que se generarán añadiendo puntos intermedios entre dos puntos inicial y final dados. Por lo general, la trayectoria tiene la forma que se muestra en la siguiente imagen:

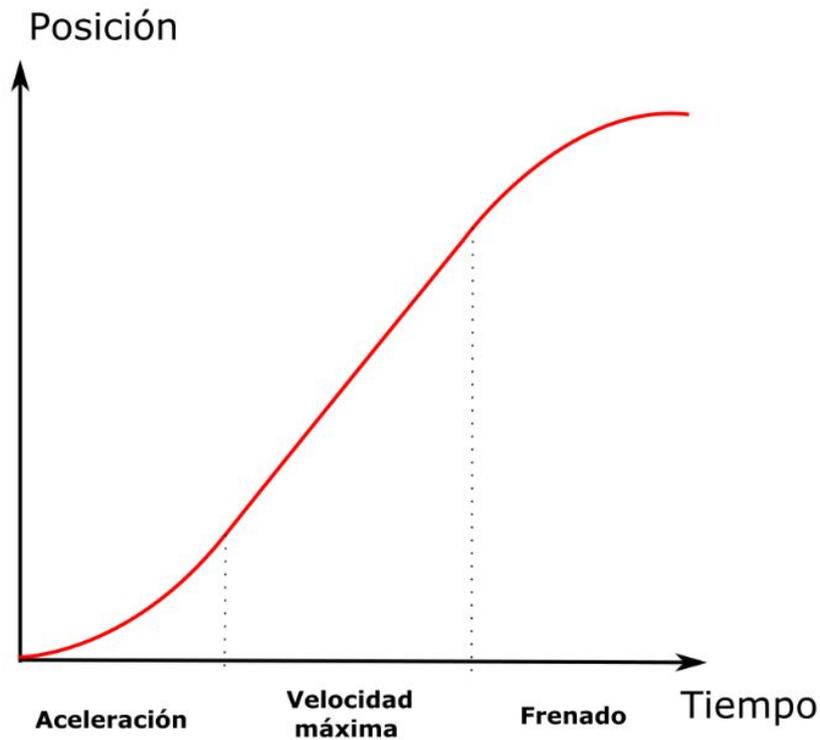


Figura 11. Trayectoria generada entre dos puntos.

Se observa una clara zona de aceleración inicial y una zona de deceleración final, siendo la velocidad constante y máxima en el tramo central.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Si se tiene un conjunto de puntos, se pueden elegir dos opciones para la creación de trayectorias:

- Por un lado, se puede generar la trayectoria punto a punto, es decir, creando una aceleración y una deceleración cada par de puntos. Ésta es la opción que se ha elegido en la elaboración de trayectorias en la aplicación.

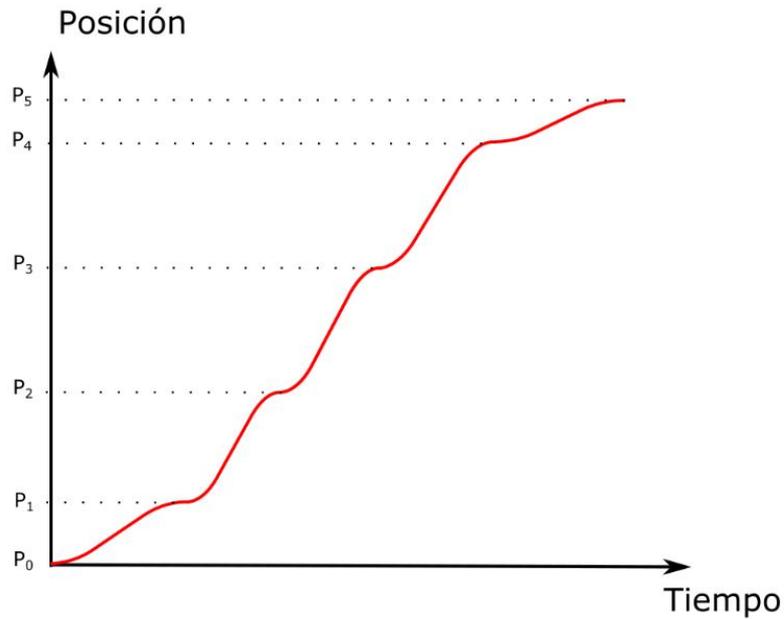


Figura 12. Trayectoria generada punto a punto.

- Por otro lado, se puede crear una trayectoria entre el punto inicial y final, de manera que el brazo robot pase por cada punto intermedio que haya a una velocidad constante.

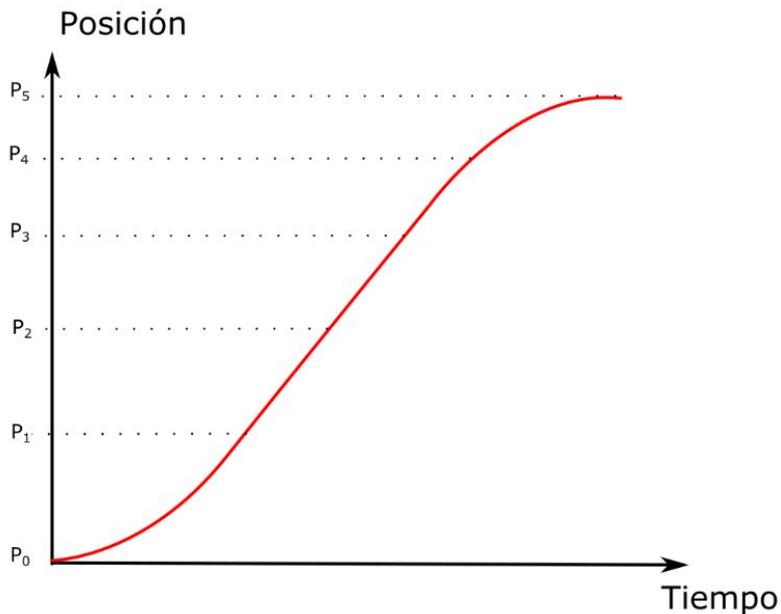


Figura 13. Trayectoria generada entre punto inicial y final.

3. Descripción de la solución adoptada

Seguidamente se van discutir las soluciones realizadas.

3.1 Estructura de la aplicación

Para el diseño de la aplicación se ha optado por el uso de bucles Do-While, de manera que el programa funcionará hasta que se pulse el botón de Stop. A continuación se van a enumerar las funciones de cada bucle

- Creación del brazo 3-D y envío de la información del movimiento manual digital de éste (incluida apertura/cierre de pinza) por medio de la cinemática directa, además de la posibilidad de enviar la información del movimiento del joystick por medio de la cinemática inversa.
- Transmisión de la información a los servo motores del brazo robot.
- Guardar los puntos en el espacio $[x, y, z]$ en una matriz y poder modificar ésta (eliminar filas, añadir en la fila deseada, cargar un fichero "txt", guardar en un fichero "txt"... Etc) y permitir la conexión y desconexión de usuarios registrados.
- Generación de trayectorias entre los puntos de la matriz de puntos generada y envío de la información del movimiento a seguir del brazo robot.
- Recibimiento de la información del sensor inercial (aceleraciones lineales en los ejes).
- Transformación de la información del sensor en movimiento del brazo robot.
- Ejecución de la cinemática inversa a la posición obtenida del sensor inercial y transmisión de la información del movimiento.
- Creación de la gráfica de los puntos en el espacio $[x, y, z]$ a realizar por la trayectoria.

Estos bucles se han combinado con las llamadas "Ocurrencias" de LabVIEW, que anulan algunos bucles en momentos determinados para ahorrar código y agilizar la aplicación (como por ejemplo cuando se usa el control manual con el Joystick se anula la generación de trayectorias, ya que se necesita que el cálculo de la cinemática inversa vaya lo más rápido posible).

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

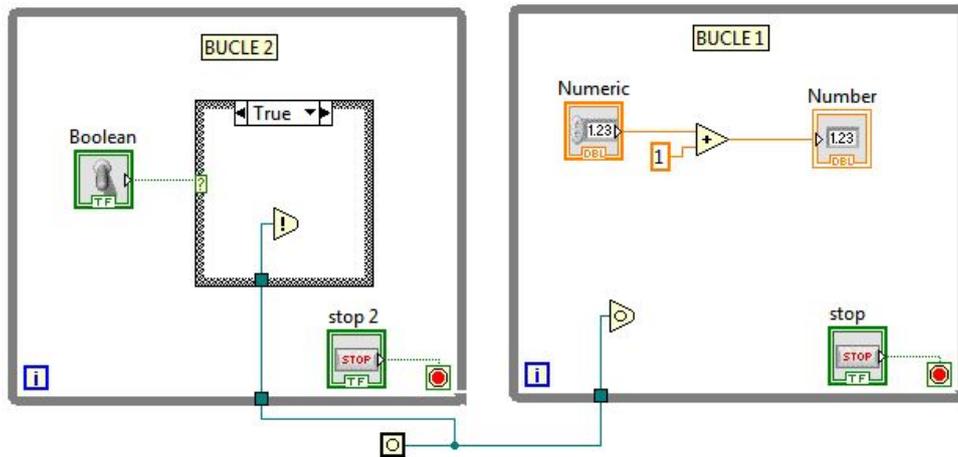


Figura 14. Ejemplo del uso de las ocurrencias.

En esta última imagen, como “BUCLE1” depende de la ocurrencia de “BUCLE2”, vemos que hasta que el interruptor “Boolean” no se active, el indicador “Number” no mostrará la operación realizada aunque cambiemos el control “Numeric”.

3.2 Cinemática directa

La solución adoptada para la cinemática directa de un robot de 4 articulaciones es la siguiente:

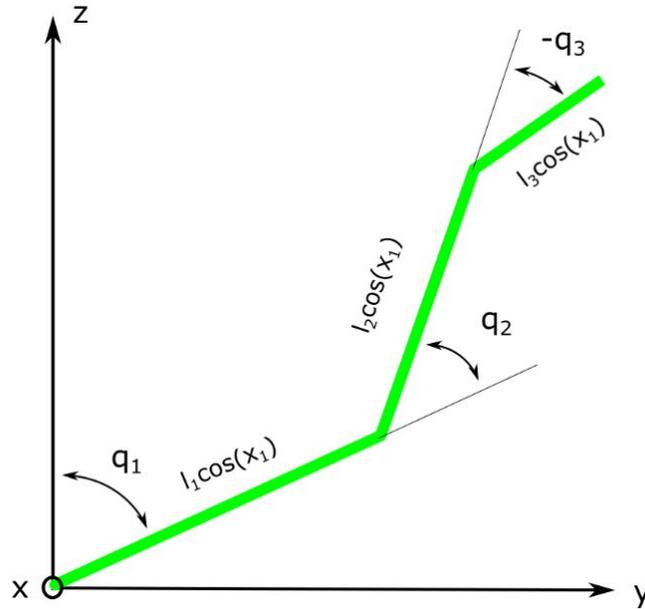


Figura 15. Problema de la cinemática directa. Plano ZY.

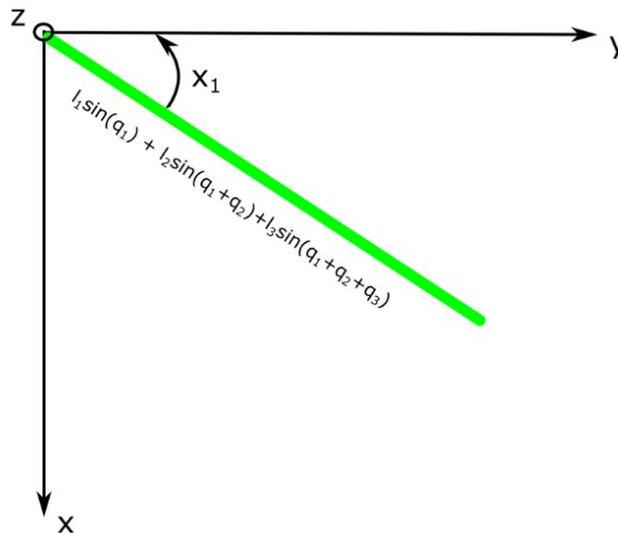


Figura 16. Problema de la cinemática directa. Plano XY.

De esta manera, sumando la longitud de la pinza a la longitud de la muñeca, se puede saber en todo momento el punto en el espacio $[x, y, z]$ en el que se está situado.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

$$\begin{aligned}
 x &= \{[l_1 \sin (-q_1)] * [\sin (-x_1)]\} + \{[(l_2 * \sin (-q_2 - q_1))] * [\sin (-x_1)]\} + \{[(l_3) * \sin (-q_3 - q_2 - q_1)] * [\sin (-x_1)]\} \\
 y &= \{[l_1 \sin (-q_1)] * [\cos (-x_1)]\} + \{[(l_2 * \sin (-q_2 - q_1))] * [\cos (-x_1)]\} + \{[(l_3) * \sin (-q_3 - q_2 - q_1)] * [\cos (-x_1)]\} \\
 z &= [l_1 \cos (q_1)] + Zbase + [l_2 * \cos (q_1 + q_2)] + [l_3 * \cos (q_1 + q_2 + q_3)]
 \end{aligned}$$

Ecuación 2. Cinemática directa aplicada al brazo robot (método geométrico).

Donde “Zbase” es la altura de la base del brazo robot.

Además, con estas ecuaciones se pueden obtener las posiciones en el espacio de los extremos de cada articulación del brazo. Esto es necesario para poder representar el brazo en tres dimensiones:



$$\begin{aligned}
 X &= \{[l_1 \sin (-q_1)] * [\sin (-x_1)]\} + \{[(l_2 * \sin (-q_2 - q_1))] * [\sin (-x_1)]\} + \{[(l_3) * \sin (-q_3 - q_2 - q_1)] * [\sin (-x_1)]\} \\
 Y &= \{[l_1 \sin (-q_1)] * [\cos (-x_1)]\} + \{[(l_2 * \sin (-q_2 - q_1))] * [\cos (-x_1)]\} + \{[(l_3) * \sin (-q_3 - q_2 - q_1)] * [\cos (-x_1)]\} \\
 Z &= [l_1 \cos (q_1)] + Zbase + [l_2 * \cos (q_1 + q_2)] + [l_3 * \cos (q_1 + q_2 + q_3)]
 \end{aligned}$$

Figura 17. Ecuaciones del punto fraccionadas en las articulaciones del brazo robot.

3.3 Cinemática inversa

Primeramente se eligió un VI de la toolbox de Robotics para realizar la cinemática inversa. Pero al tratarse de un método iterativo (a veces se obtienen resultados ilógicos si se ejecutan a tiempo real) y tampoco tenía en cuenta las limitaciones físicas del brazo robot, se optó por crear un VI propio resolviendo la cinemática inversa.

Para empezar, se ha diseñado el ángulo de rotación de la base para que el brazo se "encare" hacia el punto. Esto se consigue calculando el arco-tangente entre "x" e "y". En caso de no poder hacerlo por llegar a su límite físico ($\pm 105^\circ$), lo hará hacia el ángulo opuesto y dará un aviso para cambiar de signo los otros ángulos calculados.

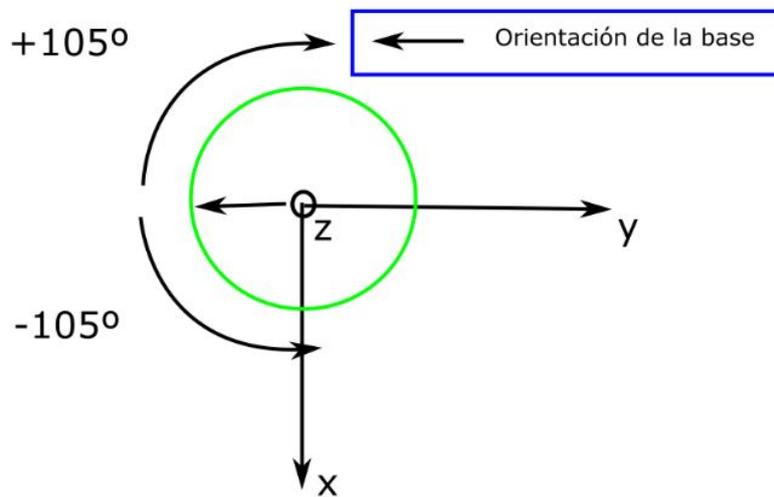


Figura 18. Problema de la cinemática inversa. Posición inicial de la base.

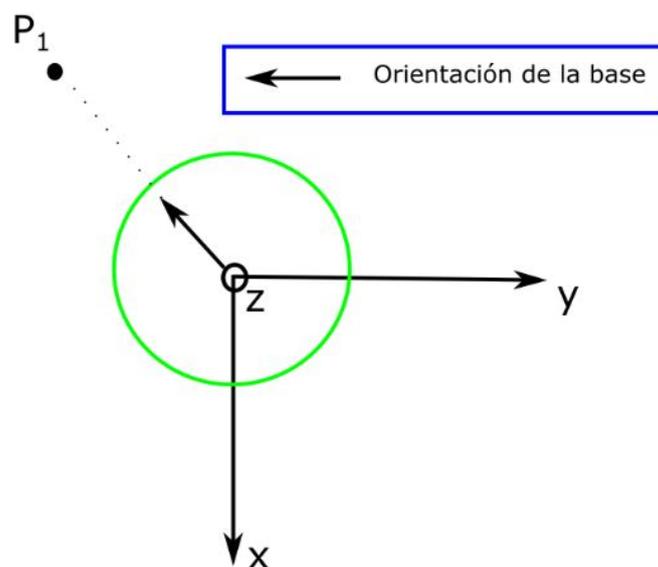


Figura 19. Problema de la cinemática inversa. Caso de un punto alcanzable para la base.

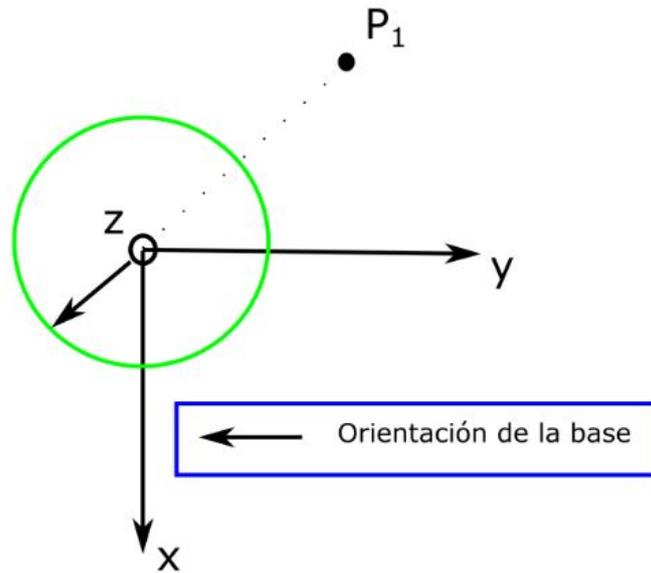


Figura 20. Problema de la cinemática inversa. Caso de un punto inalcanzable para la base.

De esta manera se tendrá que resolver un problema plano en vez de tres dimensiones, donde el extremo del brazo estará a una altura "z" y a una distancia lateral de " $r = \sqrt{x^2 + y^2}$ "

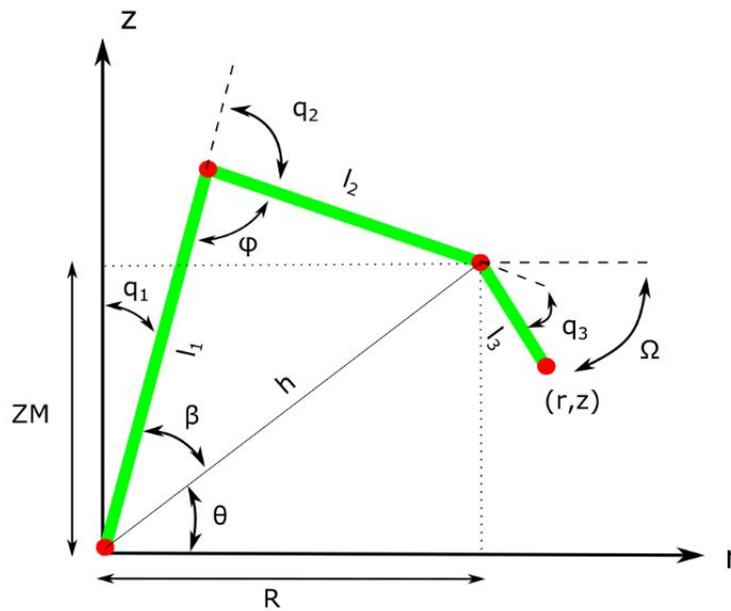


Figura 21. Problema de la cinemática inversa. Plano Zr.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Para la resolución del problema en dos dimensiones resultante, se ha escogido el método geométrico por simplicidad. No obstante, como se ha dicho anteriormente, el sistema de ecuaciones es compatible indeterminado: se tienen 3 incógnitas (ángulos del brazo q_1, q_2, q_3) y 2 ecuaciones (correspondientes a las coordenadas (r, z)):

$$r = l_1 \cos(90 - q_1) + l_2 \cos(90 - q_1 - q_2) + l_3 \cos(90 - q_1 - q_2 - q_3)$$

$$z = l_1 \sin(90 - q_1) + l_2 \sin(90 - q_1 - q_2) + l_3 \sin(90 - q_1 - q_2 - q_3)$$

Ecuación 3. Cinemática inversa plana aplicada a un brazo de tres articulaciones (método geométrico).

Es por esto que se dará por sabida el ángulo con respecto a la horizontal de la muñeca (denominada "Ángulo de cabeceo", Ω) y se calcularán los ángulos del brazo consecuentes aplicando trigonometría. Si los ángulos calculados están dentro de los límites físicos adoptados del brazo ($\pm 50^\circ$ en inclinación de hombro, codo y muñeca) entonces se dará por resuelto.

No obstante, si no se encuentra una solución, se le suma al ángulo de cabeceo 0,5 y se volverá a calcular (empieza con un mínimo de -60° hasta un máximo de 90°). Las ecuaciones utilizadas (en orden de uso) son:

$$R = r - [\cos(\Omega) * l_3]$$

$$ZM = z - [\sin(\Omega) * l_3]$$

$$h = \sqrt{R^2 + ZM^2}$$

$$\vartheta = \arctg\left(\frac{ZM}{R}\right)$$

$$\beta = \arccos\left[\frac{(l_1^2 - l_2^2 + h^2)}{2 l_1 * h}\right]$$

$$q_1 = 90 - \vartheta + \beta$$

$$\phi = \arccos\left[\frac{(l_1^2 + l_2^2 - h^2)}{2 l_1 * l_2}\right]$$

$$q_2 = 180 - \phi$$

$$q_3 = 90 - q_1 - q_2 - \Omega$$

Ecuación 4. Cinemática inversa plana aplicada al brazo robot (método geométrico).

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Sin embargo, hay ciertos puntos que no se pueden llegar a alcanzar de esta manera, ya que el resultado es tener todos los ángulos positivos o, si la base está orientada opuestamente al punto, todos negativos. Hay veces que el brazo necesita "contraerse" sobre sí mismo para alcanzar un punto, como el siguiente ejemplo:

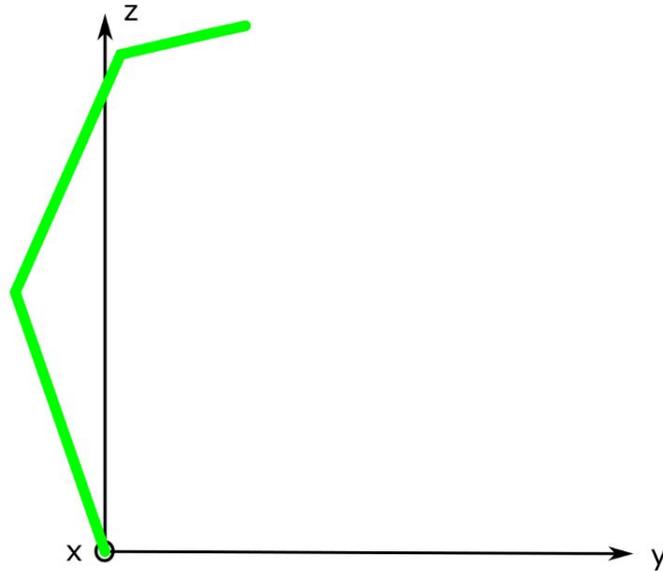


Figura 22. Ejemplo de un brazo contraído.

Es por esto que se ha delimitado una zona, de manera que si está fuera de ella, el VI calcula el caso en el que el ángulo de inclinación del hombro es negativo.

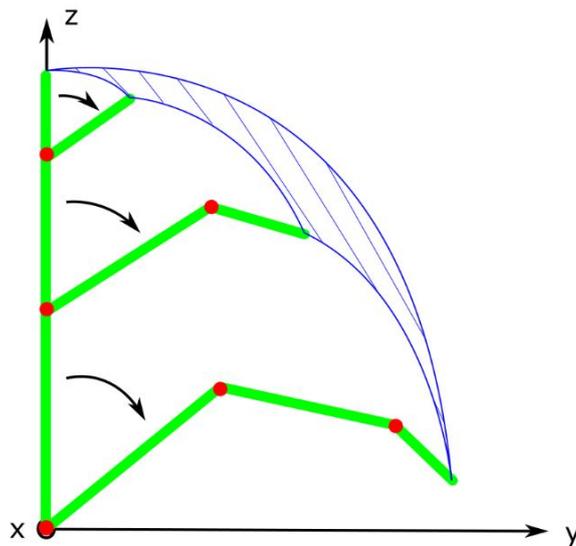


Figura 23. Representación de la zona límite del brazo robot.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Dentro de la zona rallada, se calculará como se ha mostrado anteriormente, pero fuera de ella, se cambiará una única operación: El cálculo del ángulo del hombro “ q_1 ”.

$$q_1 = -(90 + \vartheta + \theta)$$

Ecuación 5. Cambio en el cálculo de la cinemática inversa del brazo robot cuando éste se contrae.

Este cambio afectará a los ángulos q_2 y q_3 posteriormente calculados.

Con todo esto, el VI nos devuelve los ángulos calculados. Cabe destacar que también nos dice dónde se sitúa el punto en caso de no poder calcularlo, es decir, si está fuera (por ejemplo, a 1 metro de distancia) o si está dentro (como podría ser el punto (0, 0, 0), inalcanzable físicamente para el brazo robot).

3.4 Control manual

3.4.1 Digital

El control manual digital se realiza mediante controladores digitales de LabVIEW, que indican el número que se introduce en los cálculos mostrados anteriormente:

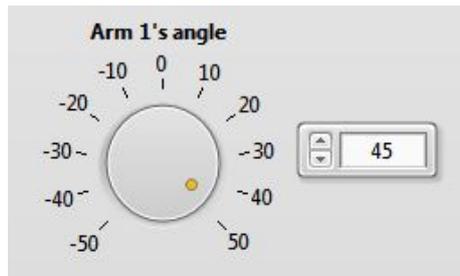


Figura 24. Control "Knob" de LabVIEW (usado en la aplicación).

3.4.2 Joystick

Se ha diseñado de manera que el joystick suma o resta coordenadas $[x, y, z]$ en mayor o menor medida (hasta llegar a unos límites diseñados) dependiendo del movimiento que se le aplica al Joystick. Esto se resume en que si se realiza el máximo movimiento del Joystick hacia la derecha, el bucle sumara en cada ciclo mucha coordenada "x", mientras que si se mueve levemente hacia la izquierda, restara un poco de "x" en cada ciclo.

Adicionalmente, el joystick tiene la opción de abrir o cerrar la pinza con un botón, así como variar el ángulo de rotación del hombro rotando el propio Joystick.

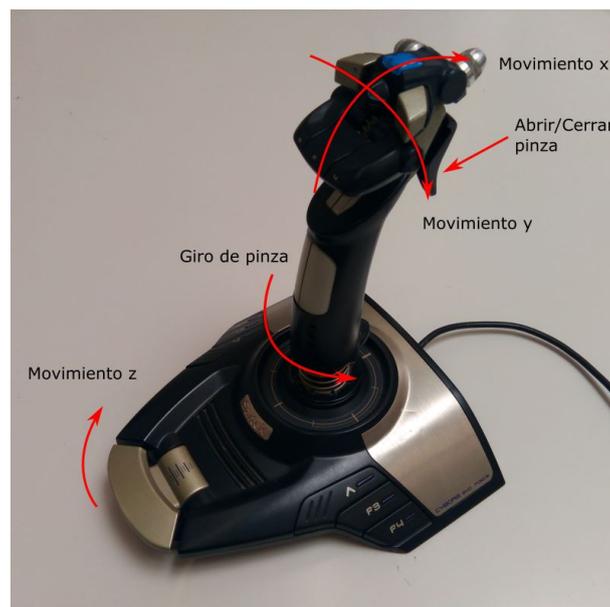


Figura 25. Movimientos del Joystick usados en la aplicación.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Para llevar a cabo la cinemática inversa se ha usado el VI comentado anteriormente. De este modo si el VI devuelve que el punto es imposible y está dentro, el brazo quedará inmóvil hasta que haya un punto al que pueda llegar, mientras que si el punto es imposible y está fuera, obligaremos a que el brazo vaya a un punto alcanzable cercano.

Este punto alcanzable se trata del límite de la semiesfera positiva en el eje Z del brazo robot, que no es más que un conjunto de los puntos más alejados que puede llegar a alcanzar el brazo robot.

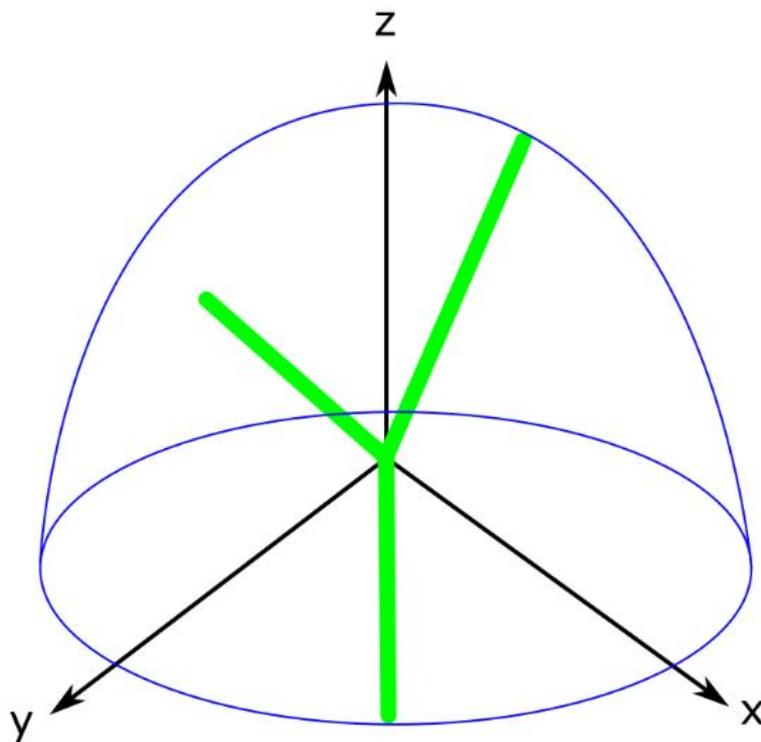


Figura 26. Representación de la semiesfera positiva en z que tiene como radio la longitud del brazo robot.

A partir de la "x" e "y" que tengamos, calcularemos la "z" mediante la siguiente fórmula (fórmula de una esfera):

$$R^2 = x^2 + y^2 + z^2$$

Ecuación 6. Esfera de radio "R".

Donde "R" es el radio de la esfera y, como "z" no puede ser negativa, se limita a la semiesfera superior.

3.5 Seguimiento de trayectorias

Una vez se obtienen los puntos a seguir en la trayectoria, se transforman en ángulos mediante el VI de cinemática inversa y se genera una trayectoria de ángulos desde los ángulos iniciales hasta los ángulos finales de cada par de puntos mediante un VI de la toolbox de Robotics de LabVIEW.

Este VI genera una trayectoria con el número de puntos intermedios que queramos. De esta manera se puede generar una trayectoria más o menos agresiva dependiendo del número de puntos, tal y como se puede observar en las siguientes imágenes:

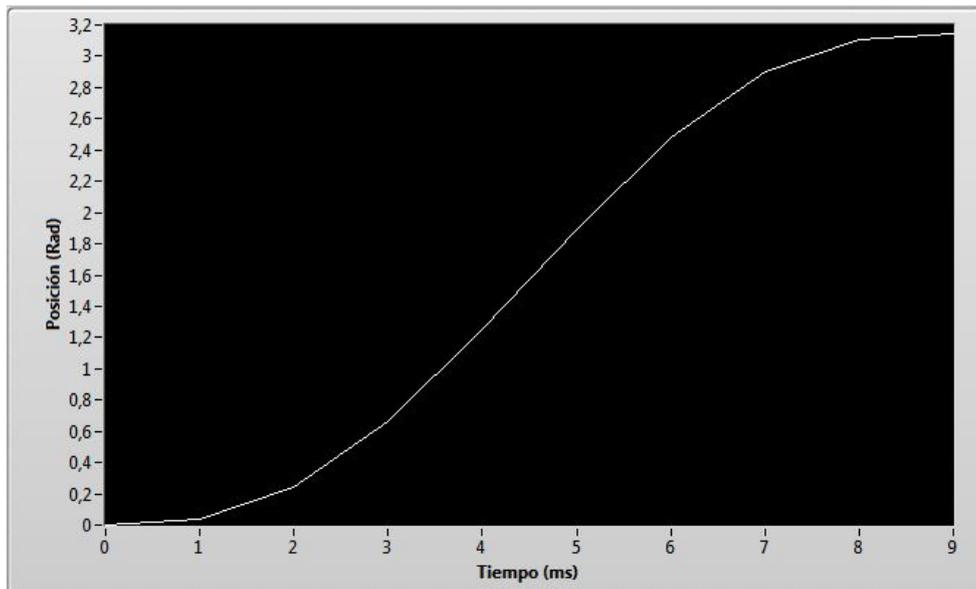


Figura 27. Trayectoria agresiva (sólo con 10 puntos).

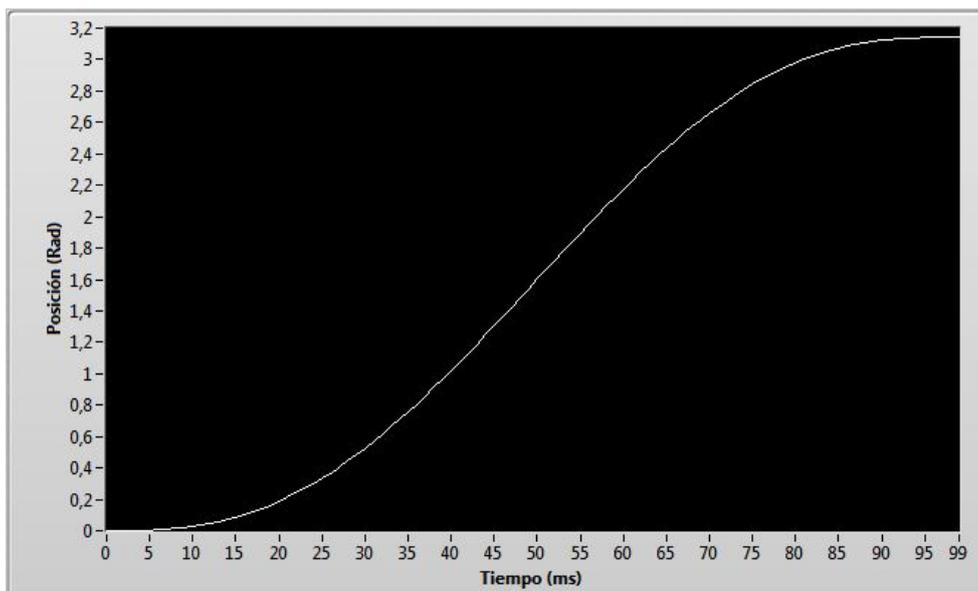


Figura 28. Trayectoria menos agresiva (con 100 puntos).

3.6 Unidad inercial

Cuando se activa el modo sensor, éste proporciona los ángulos de Euler en los tres ejes. La aplicación utiliza dos de ellos: la rotación de la base (plano XY) y la rotación del plano XZ.

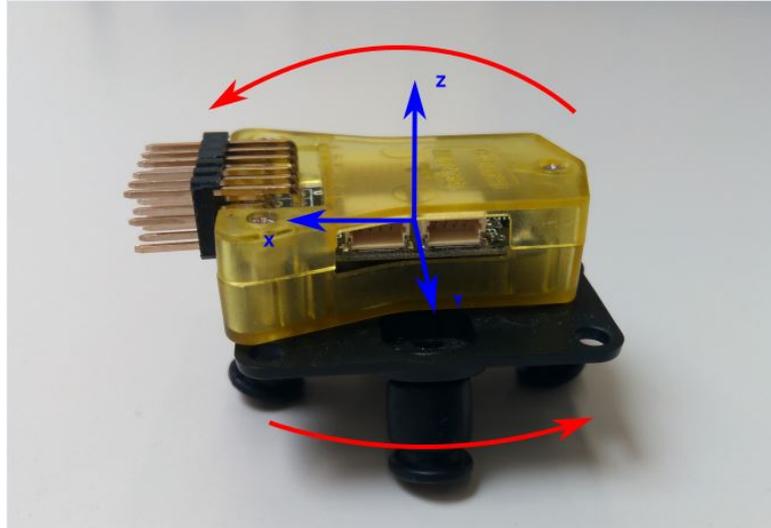


Figura 29. Movimientos realizados sobre el sensor inercial.

Mediante las transformaciones y la programación necesaria, la aplicación transforma la rotación del plano XY en la rotación de la base del brazo robot y la rotación del plano XZ en inclinación del hombro, codo y muñeca del brazo robot.

Un posible uso de esta aplicación sería, por ejemplo, en un brazo robot que trabaja en una balsa en el mar.

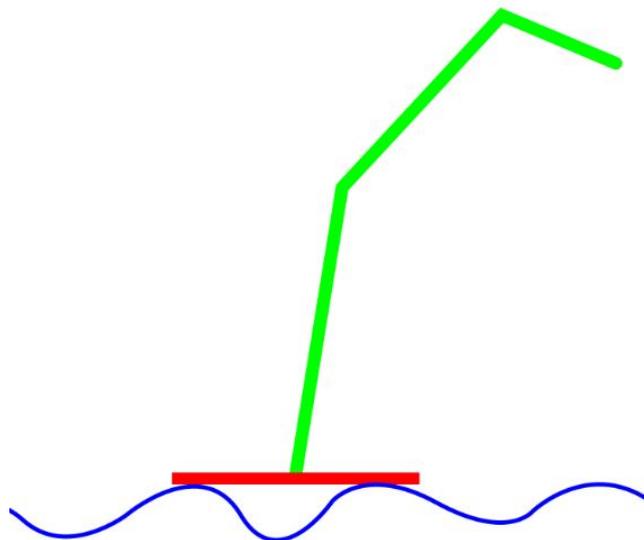


Figura 30. Ejemplo de un brazo robot situado en una balsa en el mar.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Cuando el oleaje produzca un giro en la balsa, éste podrá ser detectado por la unidad inercial y corregirlo en consecuencia:

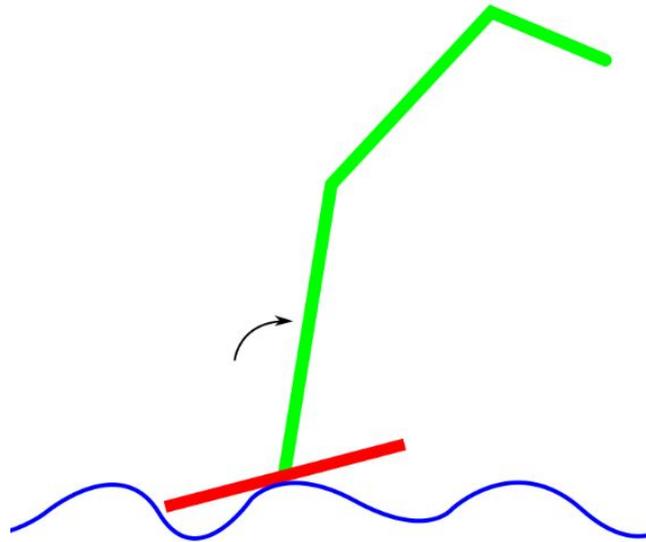


Figura 31. Ejemplo de un brazo robot en una balsa en el mar que se ha inclinado debido al oleaje.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

3.7 Seguridad

Se han implementado tres niveles de seguridad en la aplicación: Operario, Técnico e Ingeniero.

El operario sólo podrá realizar las acciones más básicas, como son mover de forma digital el brazo robot y poder cargar puntos en la matriz de puntos para realizar trayectorias pre-establecidas.

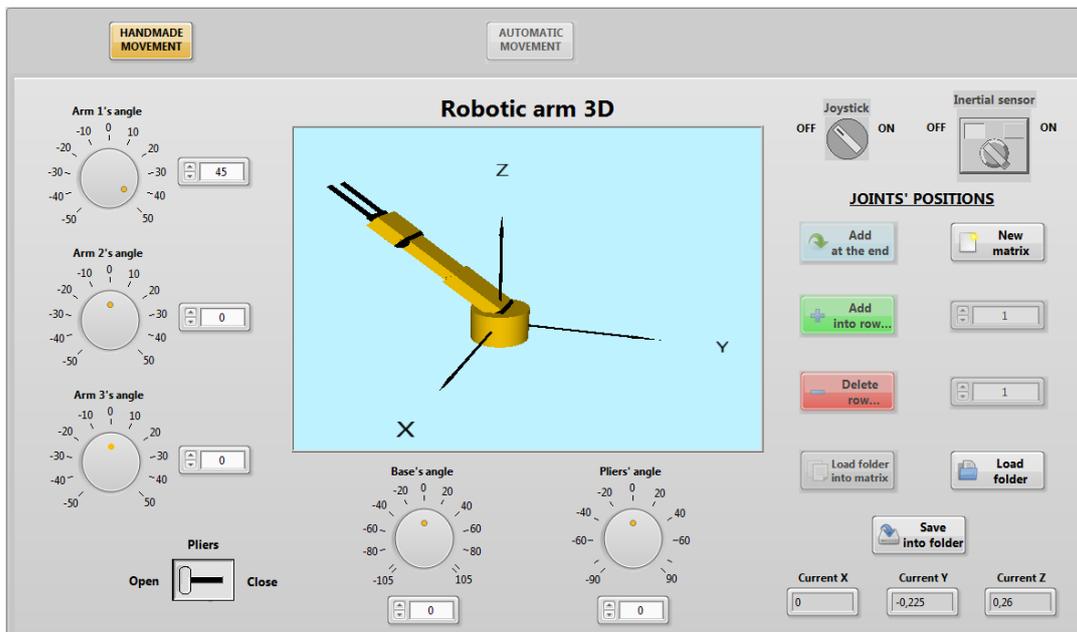


Figura 32. Visualización del programa desde el usuario "Operario".

El técnico podrá además acceder tanto al control del Joystick como al de la unidad inercial, así como la posibilidad de crear su propia matriz de puntos añadiendo o eliminando los puntos que considere.

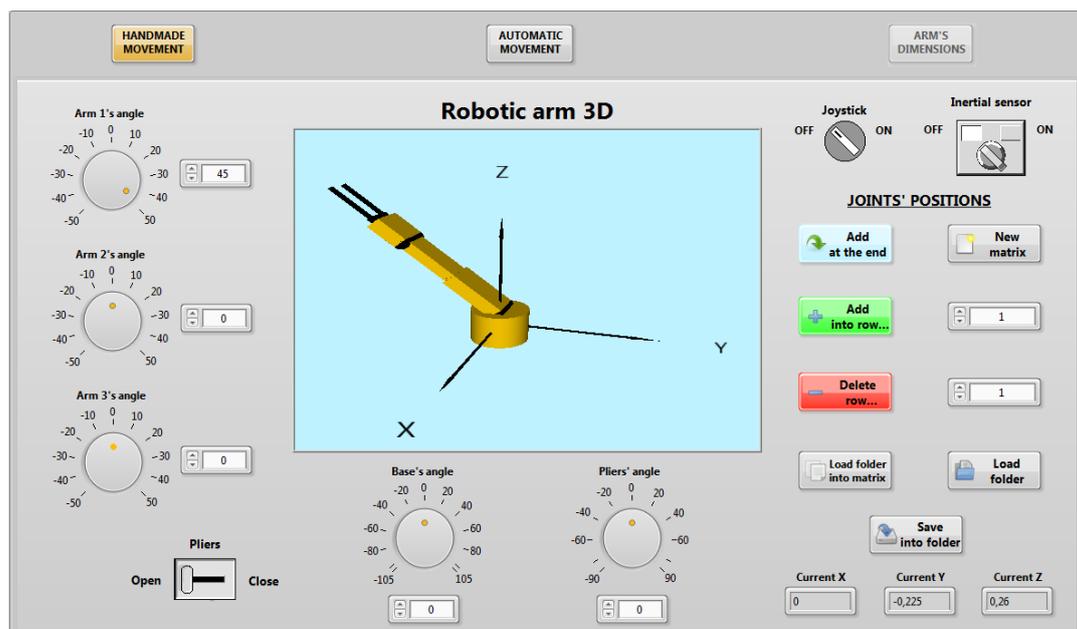


Figura 33. Visualización del programa desde el usuario "Técnico".

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Finalmente, el Ingeniero podrá cambiar las dimensiones del brazo robot y la velocidad entre los puntos de la trayectoria, es decir, tendrá completo acceso a todo el programa.

Cabe destacar que, como se ha mencionado anteriormente, si no está activo ningún usuario no se podrá realizar ninguna acción de interés en el programa. De hecho, no se podría ni desactivar el programa (Se deshabilita el botón de "Stop").

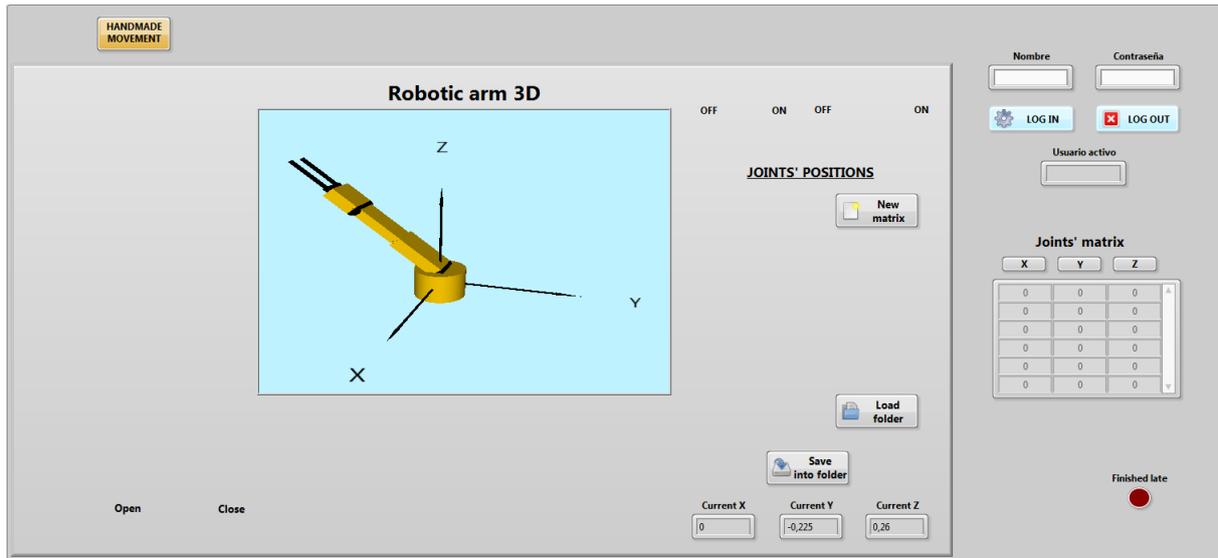


Figura 34. Visualización del programa sin ningún usuario conectado.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

3.8 Variables compartidas y uso mediante dispositivo Tablet

Las variables compartidas de LabVIEW son unas variables que pueden usarse en más de 2 programas simultáneamente mediante red y, por tanto, son capaces de modificar los controles de una aplicación principal mediante una aplicación más pequeña de forma remota.

En este caso, se ha creado una pequeña aplicación de LabVIEW que puede mover de manera digital el brazo-robot, así como abrir o cerrar la pinza a voluntad.

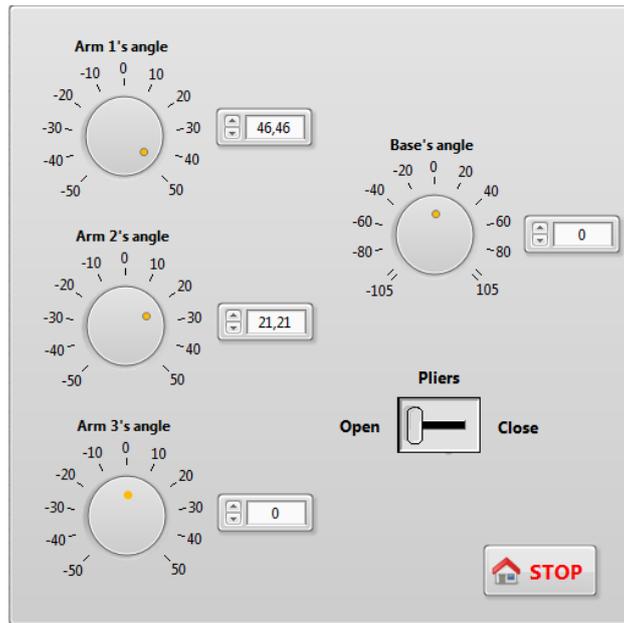


Figura 35. Interfaz de la aplicación creada para dispositivo Tablet.

4. Conclusiones

4.1 Resultados obtenidos

Se ha realizado un programa que:

- Permite el control digital del brazo, pudiendo guardar los puntos en el espacio (x, y, z) en una matriz para posteriormente generar una trayectoria entre éstos. Además, esta matriz se puede guardar en un fichero “txt”, al igual que se puede obtener cargando éste tipo de fichero.
- Permite el cambio de velocidad de las trayectorias generadas.
- Permite el control del brazo robot mediante un Joystick.
- Permite el cambio de dimensiones del brazo.
- Permite el movimiento del brazo dependiendo de una unidad inercial.
- Hace uso de la seguridad mediante el registro de usuarios.
- Puede ser controlado remotamente desde un dispositivo Tablet.

4.2 Cambios en el diseño

- Como se ha mencionado anteriormente, para el uso de la cinemática inversa se usó primeramente un “VI” de la toolbox de Robotics. No obstante, debido a que no tenía en cuenta las limitaciones físicas del brazo robot y al tratarse de un proceso iterativo (tenía errores cuando se usaba en tiempo real con el uso del Joystick), se cambió por un “VI” propio.
- Primeramente, se había pensado usar las aceleraciones lineales del sensor inercial para imitar movimientos rectos del propio brazo humano. No obstante, debido a la inexactitud del sensor y al error residual acumulativo cometido en las integraciones, la posición tendía a $\pm\infty$.
- Se ha añadido un mapa 3-D para el seguimiento de las trayectorias. En este mapa se representan los puntos del espacio (x, y, z) a seguir como puntos rojos y, a medida que el brazo llega a ellos, se vuelven verdes.

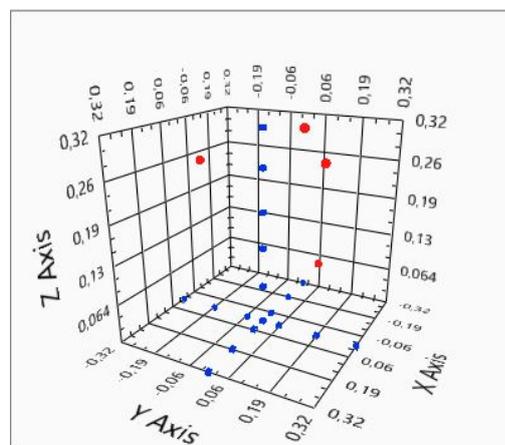


Figura 366. Gráfica de puntos en el espacio de la trayectoria.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

- Se ha añadido una matriz de puntos de trayectoria, que indica los puntos que ha obtenido al realizar la cinemática inversa. De esta manera se ve más fácilmente el error cometido en cada punto.
- Se ha añadido una funcionalidad adicional a la representación en 3D del brazo robot: Giro de la pinza. Además, se ha habilitado el movimiento de éste mediante la rotación del Joystick.

4.3 Posibles mejoras

- Aunque el uso de las ocurrencias ha supuesto una gran mejora del coste computacional, se puede seguir ahorrando con un código más eficiente. De esta manera, se podría hacer una cinemática inversa con un error mucho menor haciendo que el ángulo de cabeceo recorra un rango de ángulos mayor (es decir, incrementar el ángulo de cabeceo en 0,1 en vez de 0,5 cada vez que no se encuentra una solución).
- El uso de las ocurrencias para mover el brazo robot mediante el Joystick ha provocado que se inhabilite el bucle del uso de la matriz de puntos (crearla, guardarla, cargarla... Etc). Si se mejorara la eficiencia del código no haría falta inhabilitar éste bucle y mejoraría la utilidad de la aplicación.
- Cuando se realiza la trayectoria entre puntos, el programa se bloquea y no permite hacer nada hasta que éste acabe. Se podría mejorar el código para que siguiera cumpliendo su función sin tener que bloquear el programa.
- Para poder realizar el movimiento del brazo robot mediante las aceleraciones lineales del sensor de inercia, se podría haber usado un sistema GPS para corregir los errores residuales mencionados anteriormente.
- Con una mejora del código aplicado al Joystick, se podrían conseguir unos movimientos más naturales de éste.
- Si se realizara la parte de la generación de trayectoria sin el uso de la toolbox de Robotics, se ahorraría el coste del módulo en el presupuesto.

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA
APLICACIÓN EN LABVIEW PARA
EL CONTROL DE TRAYECTORIAS
DEL BRAZO ROBOT, DE 4
ARTICULACIONES CON PINZA,
MODELO S300105 DE
LYNXMOTION

Documento N°2

PRESUPUESTO

Autor: Carlos Babiera Sancho

Tutor: Juan Manuel Herrero Durá

Curso académico: 2015-16

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Contenido

1. Objeto.....	1
2. Presupuesto.....	2
2.1 Unidades de obra.....	2
2.2 Cuadro de precios	2
2.2.1 Materiales.....	2
2.2.2 Mano de obra.....	3
2.2.3 Tabla de precios descompuestos	4
2.2.4 Tabla de precios unitarios	4
3. Coste final.....	5

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

1. Objeto

En el presente documento se pretende realizar el presupuesto del proyecto realizado. Para ello se ha tenido en cuenta:

- Que el proyecto ha sido llevado a cabo por un Graduado de Ingeniería en Tecnologías Industriales.
- Que algunos componentes (los cuales se especificarán más adelante) han sido proporcionados por la empresa.
- Pese a ser un Trabajo Final de Grado, se incluirá el IVA en los precios calculados.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2. Presupuesto

A continuación se van a crear las unidades de obra, se van a realizar los cuadros de precios y finalmente se calcularán los precios unitarios.

2.1 Unidades de obra

Se diferencian dos:

1. Software creado: Es la programación del código de la aplicación, que incluye todo el tiempo necesario para crearla (búsqueda de información y programación).
2. Equipo usado: Se considerará que tanto el brazo robot como el sensor y el Joystick lo proporciona la empresa. Esta unidad de obra incluye licencias informáticas, ordenador, fuente de alimentación, montaje, transporte y revisión.

2.2 Cuadro de precios

2.2.1 Materiales

Por un lado, se tiene el software. En este cuadro de precios se consideran la amortización de licencias en un año (aproximadamente 1850 horas).

En la siguiente tabla se muestra el precio por hora de cada software:

LICENCIA	COSTE (€)	PRECIO (€/h)
Windows 10	62	0,034
LabVIEW	995	0,538
Toolbox Robotics	345	0,186
Módulo DSC	950	0,514

Tabla 1. Cuadro de precios de las licencias.

Por otro lado, se tiene el hardware. Como se supone que la empresa proporciona el brazo robot, sensor y el joystick empleados, sólo se tiene en cuenta el ordenador y sus periféricos.

Éste equipo se considera a amortizar en 2 años (aproximadamente 3700 horas), por lo que el cuadro de precios queda:

HARDWARE	COSTE (€)	PRECIO (€/h)
Ordenador	900	0,243

Tabla 2. Cuadro de precios del hardware.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

2.2.2 Mano de obra

Como mano de obra directa se tiene a un graduado en Ingeniería en Tecnologías Industriales, que realizará la tarea de programación.

Se considera el tiempo del proyecto de un mes y medio. Esto junto a los aproximadamente 22 días laborables que tiene cada mes, hacen:

$$22 \text{ días/mes} * 8 \text{ horas} * 1,5 \text{ meses} = 264 \text{ horas}$$

Como mano de obra indirecta se tiene a un técnico que emplea 3 horas de su horario semanal para supervisar, realizar el montaje, ayudar en el proyecto y realizar tareas de mantenimiento. El número de horas que trabajará el supervisor serán (considerando que un mes tiene aproximadamente 4,3 semanas):

$$1,5 \text{ meses} * 3 \text{ horas/semana} * 4,3 \text{ semanas/mes} = 19,35 \text{ horas}$$

GASTOS	PRECIO INGENIERO (€)	PRECIO TÉCNICO
Sueldo (220 días/año)	24.000	13.000
Seguridad social	8.000	4.700
Horas extra	1.400	1.000
Transporte	1.200	800
Total por año	34.600	19.500
Por jornada (8 horas)	157,273	88,636
Por hora	19,660	11,080

Tabla 3. Cuadro de precios de la mano de obra.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

2.2.3 Tabla de precios descompuestos

Es la tabla que muestra los precios (€/h) de cada parte de cada unidad de obra. En nuestro caso tenemos:

RECURSOS	PRECIO (€/h)
<u>U.O.1 Elaboración de la aplicación</u>	
Programador	19,660
Amortización equipo	0,243
Amortización licencia Windows 10	0,034
Amortización licencia LabVIEW	0,538
Amortización licencia Robotics	0,186
Amortización licencia Módulo DSC	0,514
<u>U.O.2 Montaje, supervisión y mantenimiento</u>	
Técnico	11,080

Tabla 4. Cuadro de precios descompuestos.

2.2.4 Tabla de precios unitarios

Esta tabla muestra los precios (€/h) de cada unidad de obra y las mediciones (horas invertidas) de cada una de ellas para obtener el gasto. La tabla queda:

UNIDAD DE OBRA	PRECIO (€/h)	MEDICIÓN (h)	COSTE TOTAL (€)
Elaboración de la aplicación	21,175	264	5.590,2
Montaje, supervisión y mantenimiento	11,080	19,35	214,398

Tabla 5. Cuadro de precios unitarios.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

3. Coste final

Se tiene que el presupuesto de ejecución material es:

$$\text{Presupuesto de ejecución material} = 5.590,2 + 214,398 = 5.804,598 \text{ €}$$

El presupuesto de ejecución por contrata, considerando un beneficio industrial del 6% y un gasto general del 16%:

$$\text{Presupuesto de ejecución por contrata} = 1,22 * 5.804,598 = 7.081,610 \text{ €}$$

Finalmente, añadiendo un 21% de IVA

$$\text{Coste total} = 1,21 * 7.081,610 = 8.568,748 \text{ €}$$

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA
APLICACIÓN EN LABVIEW PARA
EL CONTROL DE TRAYECTORIAS
DEL BRAZO ROBOT, DE 4
ARTICULACIONES CON PINZA,
MODELO S300105 DE
LYNXMOTION

Documento N°3

ANEXO DE PROGRAMACIÓN

Autor: Carlos Babiera Sancho

Tutor: Juan Manuel Herrero Durá

Curso académico: 2015-16

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Contenido

1. Objeto	1
2. Antecedentes.....	2
2.1 Event structure.....	2
2.2 Case structure	2
2.3 Shift registrar	3
2.4 Formula node	3
3. Modelo 3D	4
4. Creación de la matriz de puntos	7
4.1 Nueva matriz.....	7
4.2 Añadir al final.....	7
4.3 Eliminar fila... ..	8
4.4 Cargar matriz	9
4.5 Guardar matriz	10
5. Generación de trayectorias.....	11
6. Gráfica de puntos de la trayectoria	14
7. Control y comunicación de la posición	16
8. Movimiento con Joystick.....	18
9. Movimiento con sensor inercial	21
10. Seguridad.....	23
11. Programa para dispositivo Tablet	25

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

1. Objeto

En el presente anexo se pretende explicar el código de programación utilizado en la aplicación, facilitando así una mejor comprensión para que el futuro programador pueda entender, cambiar e incluso mejorar el programa.

Este anexo está hecho para un usuario que ya sepa las nociones básicas del programa. Por ello, se recomienda que se vean los tutoriales disponibles (National Instruments, s.f.), si no se ha programado anteriormente con este software.

2. Antecedentes

2.1 Event structure

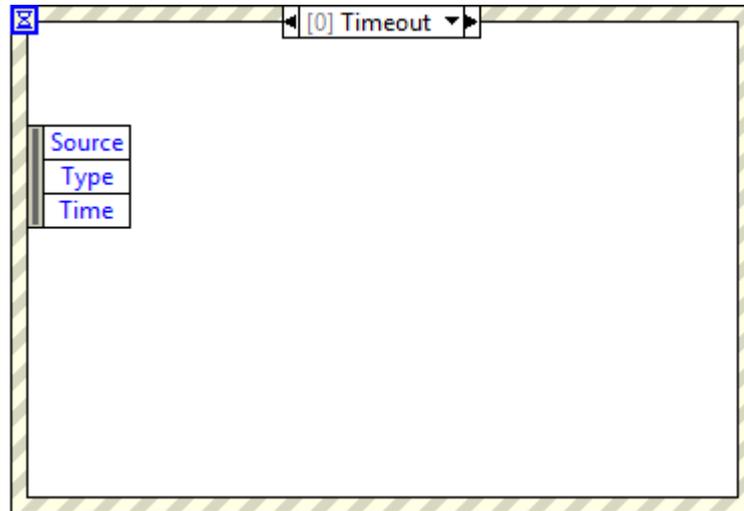


Figura 1. Caja Event structure.

Éste bucle permite añadir acciones a posibles estados, es decir, se ejecutará lo que indique el código de la programación si se realiza el cambio programado (como podría ser, por ejemplo, la acción de un pulsador). Predeterminadamente, tiene un estado "Timeout" que se refiere a lo que hará en el caso de que no se ejecute nada.

2.2 Case structure

Éste bucle es la representación gráfica del famoso bucle "If... Else" en cualquier lenguaje de programación.

En este ejemplo, si el booleano introducido es "True", el indicador mostrará 1. Si es "False", mostrará 0.

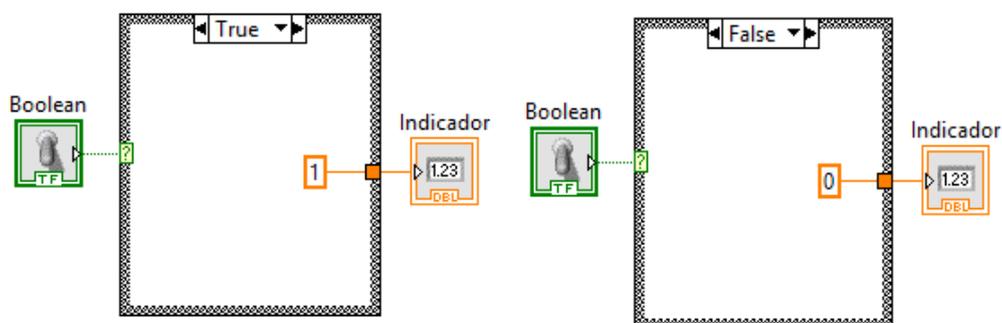


Figura 2. Ejemplo de funcionamiento del Case structure.

2.3 Shift register

Sólo se puede usar en bucles For o While, y consiste en la retroalimentación de un valor.

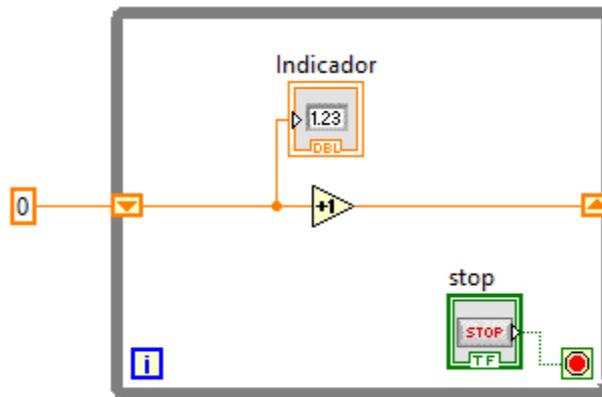


Figura 3. Ejemplo de funcionamiento del Shift register.

En este ejemplo, el indicador mostrara inicialmente 0, y a cada iteración se sumará 1 al resultado anterior hasta que se finalice el bucle, tendiendo así a $+\infty$.

2.4 Formula node

Permite el uso de lenguaje Matlab para facilitar la comprensión del código utilizado.



Figura 4. Ejemplo de funcionamiento del Formula node.

En este ejemplo, el indicador mostrará el número dos.

3. Modelo 3D

Para explicar de una manera más sencilla la creación del modelo 3D usado en la aplicación creada en LabVIEW, se usara un ejemplo más sencillo.

Este ejemplo consiste en la representación de un cilindro unido a una esfera en el extremo de éste, pudiendo girar el cilindro (y en consecuencia, la esfera) en forma de inclinación y de rotación (como si del hombro del brazo robot se tratara). Está compuesto por los mismos componentes que se han usado en la representación del brazo de tres articulaciones con la base y la pinza, pero de una forma más simple.

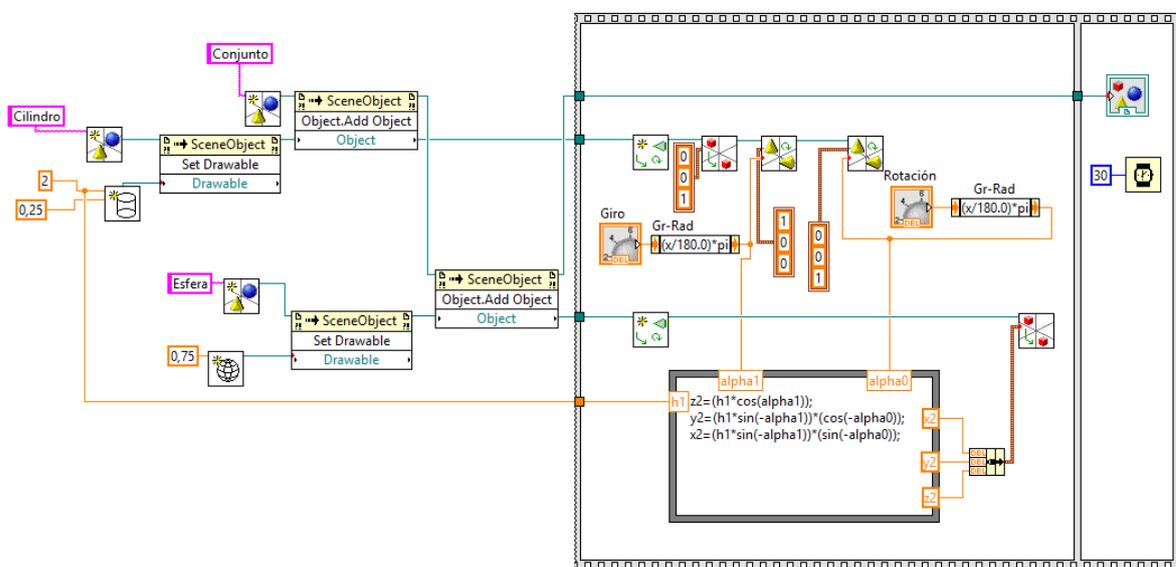


Figura 5. Ejemplo de código para la creación de una figura 3D.

Se pasa a explicar parte por parte el código utilizado:

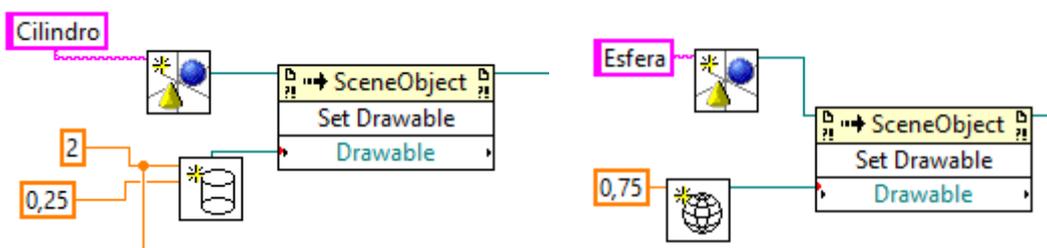


Figura 6. Creación del cilindro y de la esfera.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Primeramente en la Figura 6 se muestra la creación del cilindro (con el radio y la altura elegidos) y de la esfera (con el radio elegido).

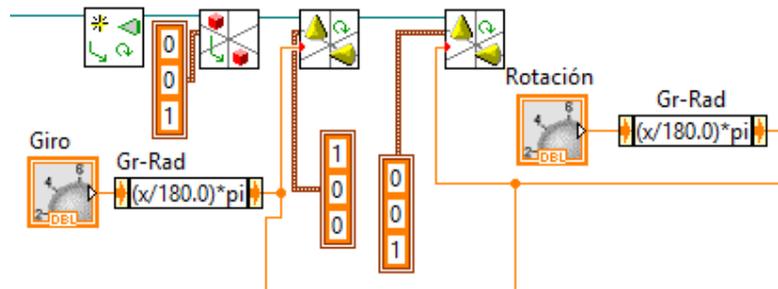


Figura 7. Configuración del movimiento del cilindro.

Seguidamente, como se ve en la Figura 7, se configura el primer elemento, es decir, el cilindro. Éste se eleva la mitad de su altura para que el centro de giro se sitúe en su base. Se gira alrededor del eje Z y del eje X mediante unos controles.

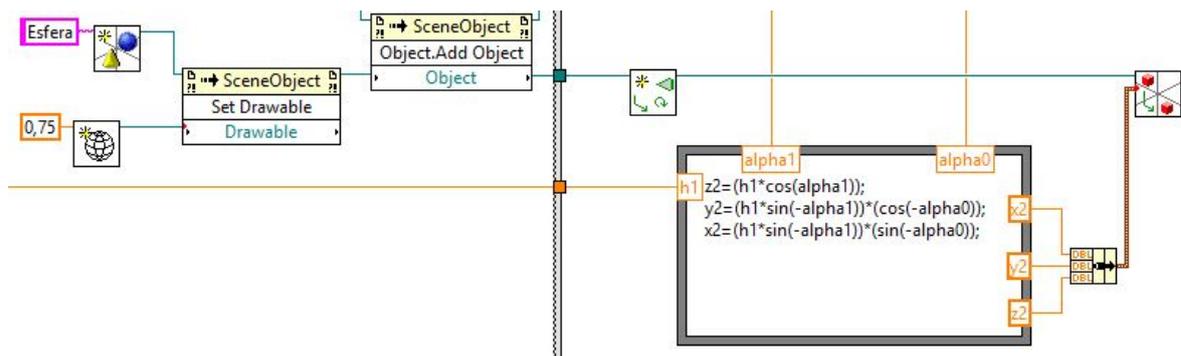


Figura 8. Configuración del movimiento de la esfera.

Finalmente en la Figura 8 se unen los elementos geométricos y se configura la posición de la esfera en todo momento mediante trigonometría, y se representa la figura en 3D.

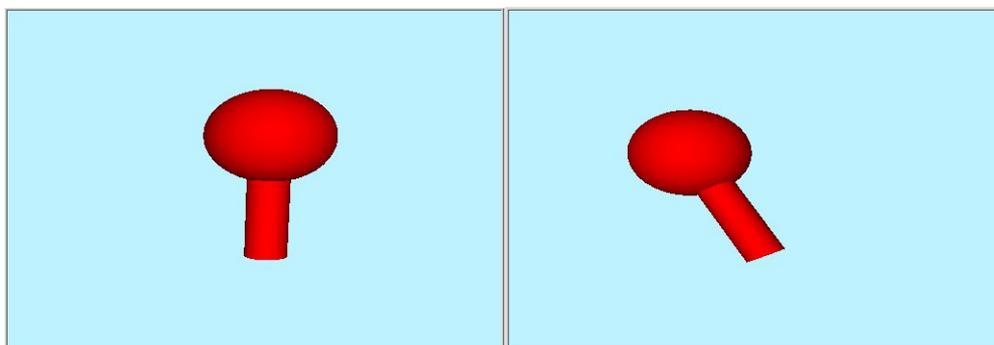
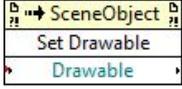
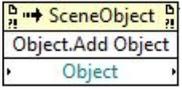


Figura 9. Posibles movimientos de la figura 3D creada.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

A continuación se explicarán todas las funciones de LabVIEW utilizadas en la creación del brazo robot en 3D:

- Creación de figuras geométricas  : Crea un objeto, que es "rellenado" por un dibujo u otro conjunto de objetos.
- Create cylinder  : Crea el dibujo de un cilindro cuyas dimensiones y color se escogen.
- Create box  : Crea el dibujo de una caja cuyas dimensiones y color se escogen.
- Set drawable  : Incluye un dibujo dentro de un objeto.
- Add object  : Añade un objeto a otro.
- Clear transformation  : Elimina todas las transformaciones previas.
- Translate object  : Traslada el objeto al lugar indicado.
- Rotate object  : Gira el objeto alrededor del eje o ejes escogidos la cantidad elegida.
- Create 3D axis  : Crea ejes ortogonales X, Y, Z con la longitud y el grosor deseados.

4. Creación de la matriz de puntos

Como se ha visto anteriormente, al crear la imagen en 3 dimensiones del brazo robot se obtienen las coordenadas del final de las extremidades. Estos puntos se pueden guardar en la matriz cuando el usuario pulsa el booleano creado para ello. A continuación, se va a explicar cómo se han creado los botones más representativos de la creación de la matriz.

4.1 Nueva matriz

En esta opción simplemente se crea un vector vacío y se transforma a matriz. También se reinicializa el Shift registrer a cero.

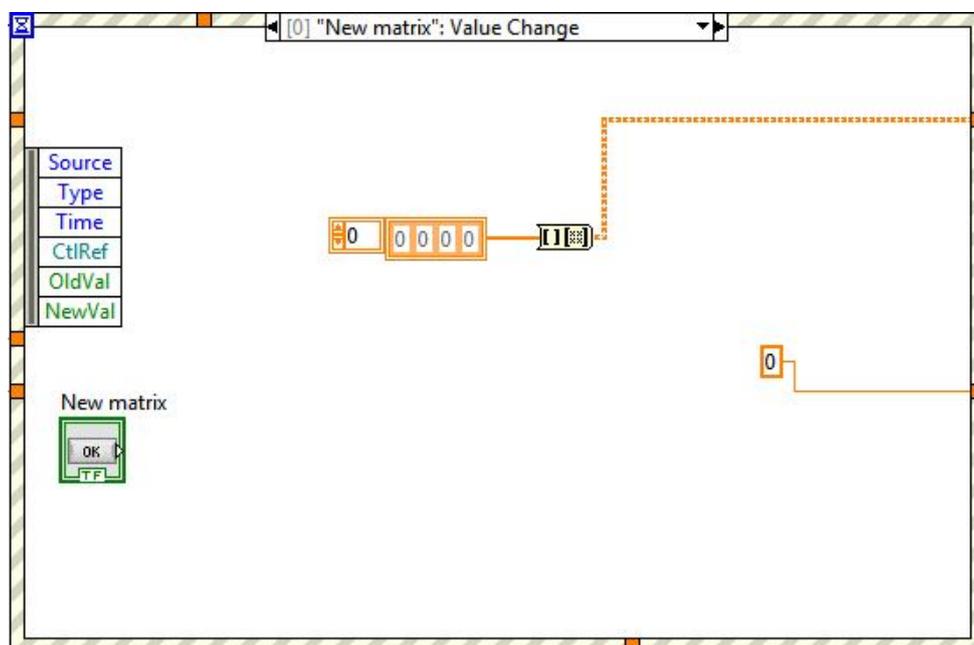


Figura 10. Código usado para la creación de una nueva matriz vacía.

4.2 Añadir al final

Inicialmente (o al seleccionar "New matrix") se tiene una matriz vacía con un contador de filas a cero. Cuando se pulsa este botón, se añaden las coordenadas (x , y , z) del punto actual que vendrá dado por la figura 3D.

Se añade a la matriz con la opción "Set matrix elements", en la fila que indica el "Shift register". Al acabar esta operación sumará 1 a la fila de la matriz.

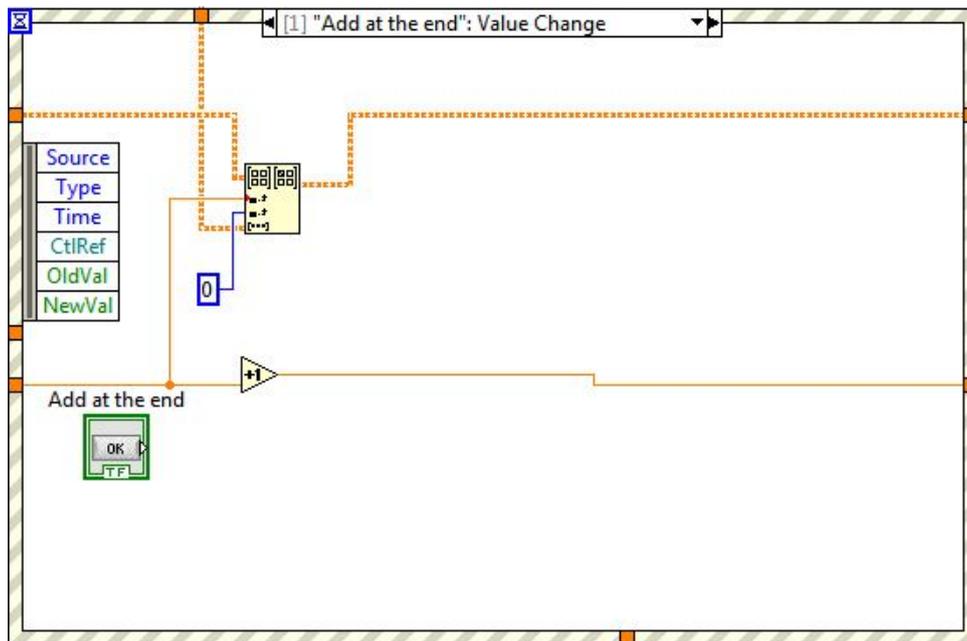


Figura 11. Código usado para la adición de un punto al final de la matriz.

4.3 Eliminar fila...

Con la acción "Delete from array", se eliminará el número de fila que indique el control y se restará 1 al número de fila del "Shift register" para escribir en la matriz desde la posición correcta.

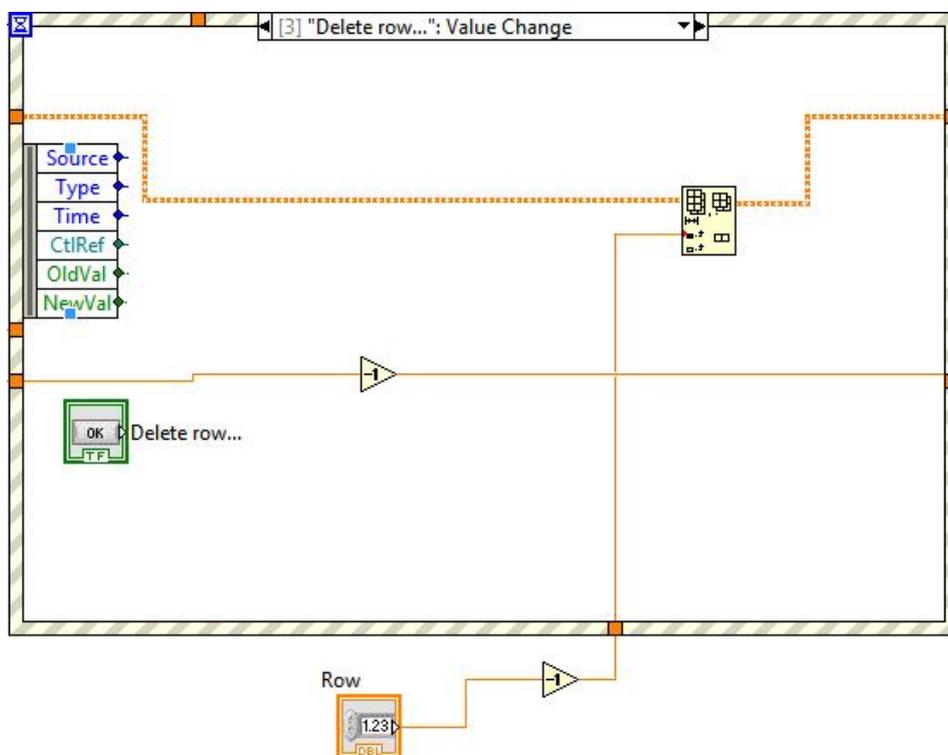


Figura 12. Código usado para la eliminación de la fila deseada de la matriz.

4.4 Cargar matriz

Gracias al VI "Read from spreadsheet File", podemos leer el archivo "txt" y cargarla sobre nuestra matriz. Actualizamos el número de filas del "Shift registrer" con el número de filas de la matriz leída.

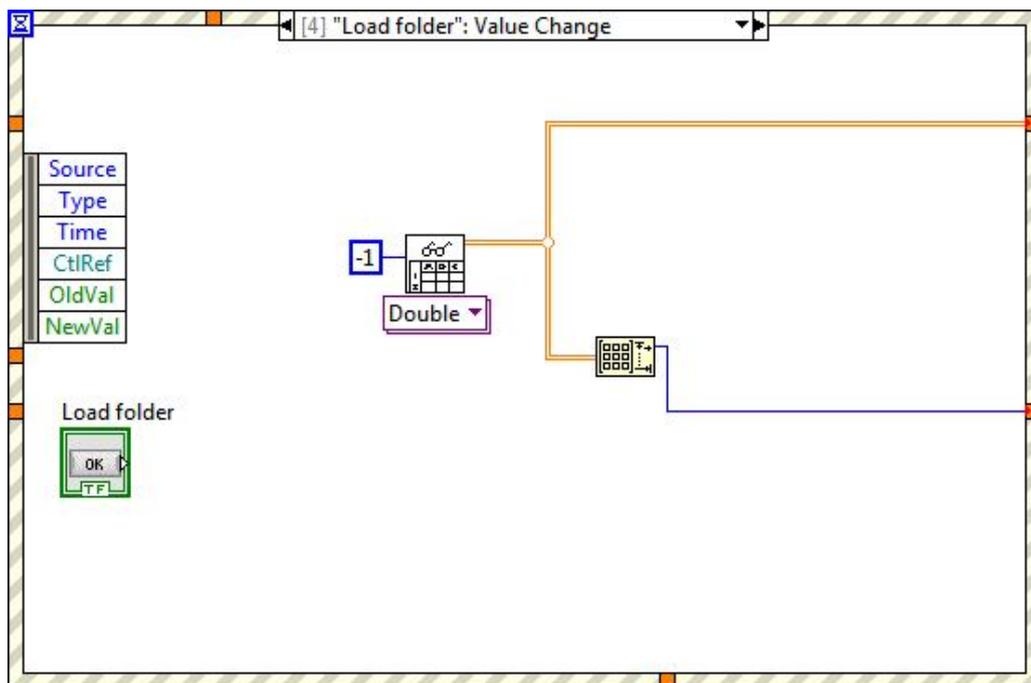


Figura 13. Código usado para cargar una matriz a partir de un fichero "txt".

El "-1" mostrado en el VI significa que tiene que cargar todas las filas del fichero.

4.5 Guardar matriz

Esto se consigue con el VI "Write to spreadsheet", que guarda el fichero. Como el número de filas no se ve afectado en esta operación, el "Shift registrer" no varía.

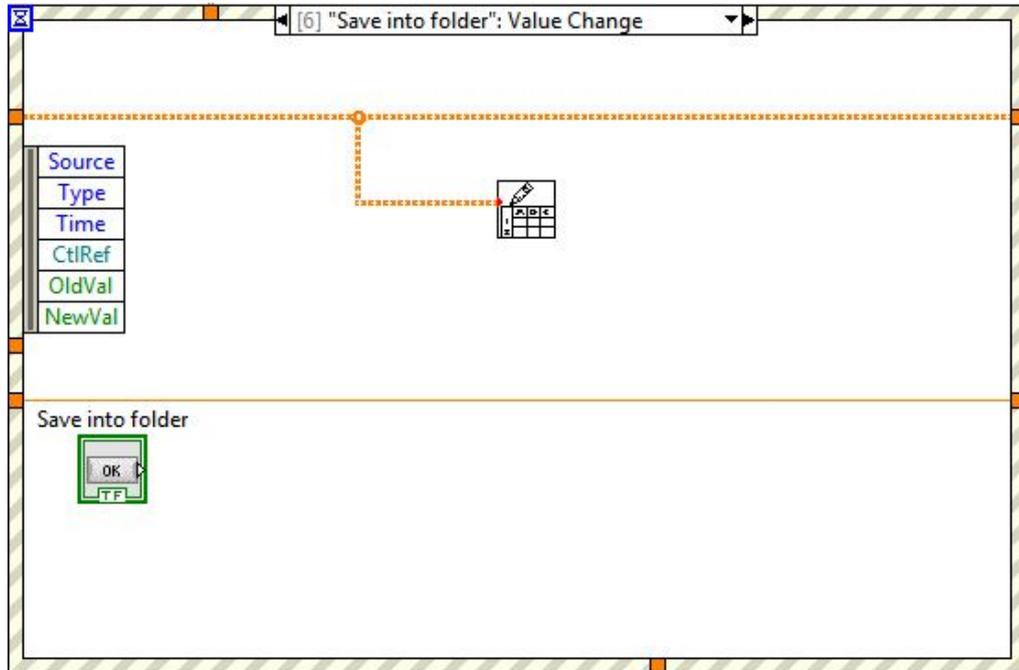


Figura 14. Código usado para guardar la matriz en un fichero "txt".

5. Generación de trayectorias

En nuestro caso, al apretar el botón de "Search points" se ejecutará la generación de la trayectoria, que consiste en dos bucles For (uno dentro del otro).

El primer bucle se repetirá el número de puntos que contenga la matriz. Para saber éste número de puntos se utiliza "Matrix size", que devuelve el número de filas y columnas de la matriz.



Figura 15. Icono de "Matrix size".

Cada iteración del primer bucle For, escogerá desde el primer hasta el último punto para generar una trayectoria desde el punto anterior hasta el actual (para el primer punto, se toma como punto anterior la posición inicial del brazo pre-establecida).

Para escoger los puntos de la matriz, se escoge el "Get matrix elements", cambiando el número de la fila a escoger a cada iteración gracias al "Shift register". Esta operación se realiza dentro del VI creado "Posiciones".

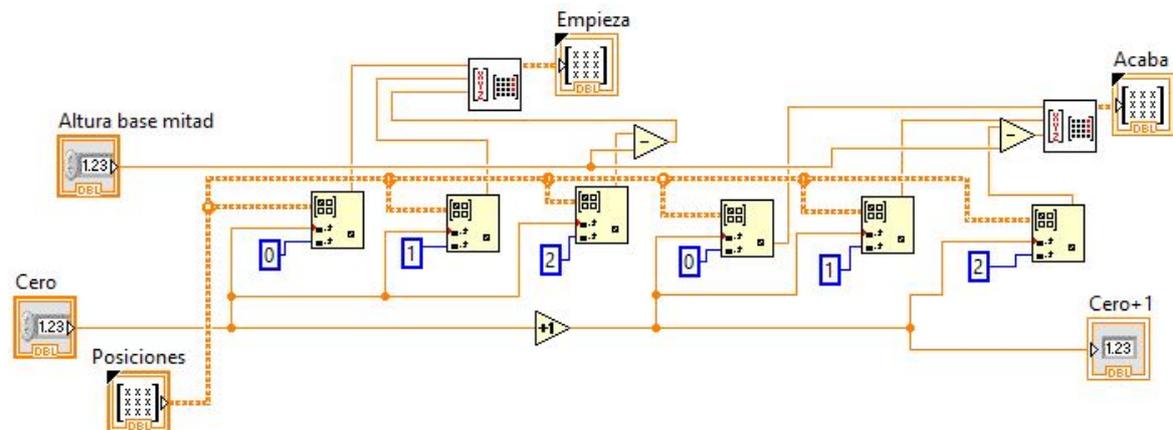


Figura 16. Código usado para la obtención de puntos a través de la matriz de puntos.

Se transforman las coordenadas (x, y, z) obtenidas a ángulos aplicando el VI de cinemática inversa creado.

Mediante la toolbox de Robotics se genera la trayectoria entre los dos ángulos obtenidos con el VI "Generate joint trajectory".



Figura 17. Icono de "Generate joint trajectory".

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Esta trayectoria pasa al segundo bucle, que basará su número de iteraciones en función de la longitud del vector de trayectoria.

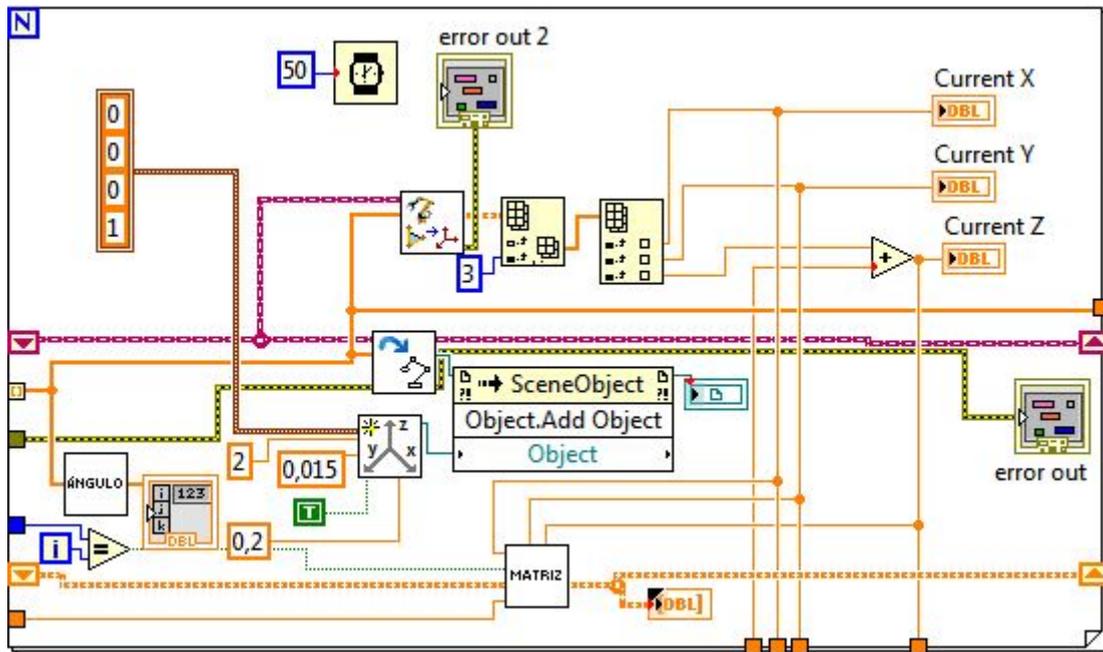


Figura 18. Segundo bucle For en la generación de trayectoria.

Éste bucle se encargará de recorrer punto a punto la trayectoria generada y mostrar la representación del "esqueleto" del brazo nuevamente mediante un VI de la toolbox de Robotics: "Update plot".



Figura 19. Icono de "Update plot".

Además, en la representación también se usan previamente otros VI de la toolbox de Robotics:

- Initialize serial arm : Define los posibles grados de libertad de las articulaciones del brazo robot y la distancia entre ellas.
- Zero out friction : Anula la posible fricción en el movimiento del esqueleto del brazo robot.
- Initialize plot : Inicializa la representación del esqueleto del brazo robot.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

El bucle también muestra las coordenadas (x, y, z) recorridas, además de guardar estos puntos una vez llegan a su destino a la matriz de trayectoria.

Para guardar estos puntos se usa el VI "Forward kinematics" de la toolbox de Robotics que usa la cinemática directa para calcularlos.



Figura 20. Icono de "Forward kinematics".

Una vez obtenidos los puntos en el espacio (x, y, z) , se muestran en pantalla y se añaden a la matriz de trayectoria mediante VIs ya explicados.

7. Control y comunicación de la posición

Para controlar la posición del brazo robot, se le envían los ángulos obtenidos de las diferentes funcionalidades de la aplicación (movimiento directo y automático).

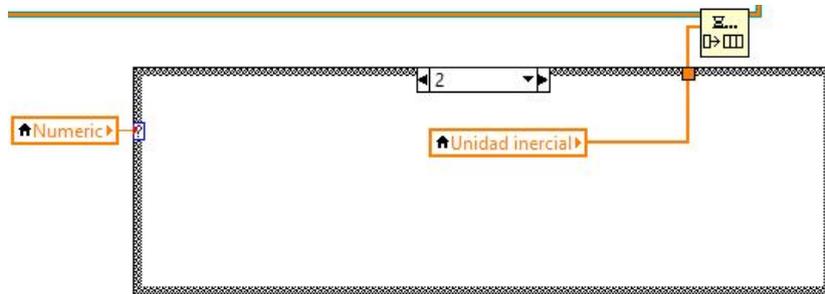


Figura 25. Código usado para enviar los datos de los ángulos a la cola de datos.

Este bucle añade los ángulos enviados de la funcionalidad que es de interés en ese momento y los añade a la cola de datos con el VI "Enqueue element" mostrado en la Figura

Éstos datos se obtienen de la cola de datos con el VI "Dequeue element".



Figura 26. Icono de "Dequeue element".

Cuando se obtienen datos de la cola, se transforman a radianes (originalmente son grados) y se envían los datos al puerto serie mediante el siguiente VI:

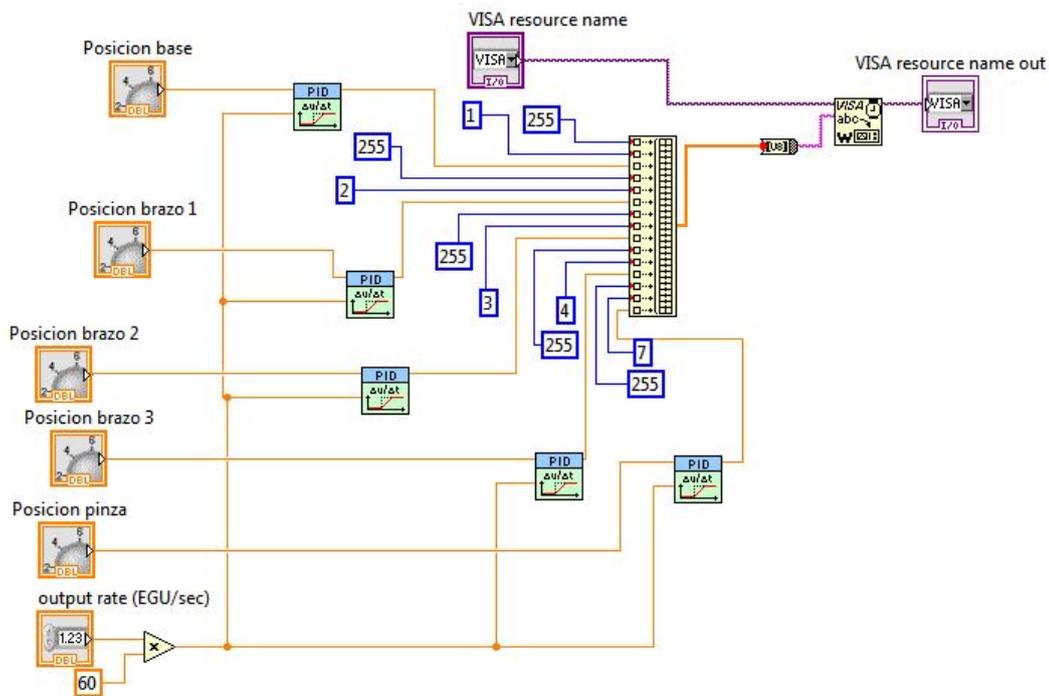


Figura 27. Código usado para el envío de las posiciones angulares al brazo robot.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Que hace que se envíen los bits necesarios a la placa SSC vía puerto serie.

Para que el movimiento de una posición a otra no sea demasiado rápida (haciendo que hasta el brazo robot se tambalee) se regula velocidad con el VI "PID Output rate limiter"



Figura 28. Icono de "PID Output rate limiter".

8. Movimiento con Joystick

Primeramente, se leen los bits recibidos del joystick con el siguiente VI

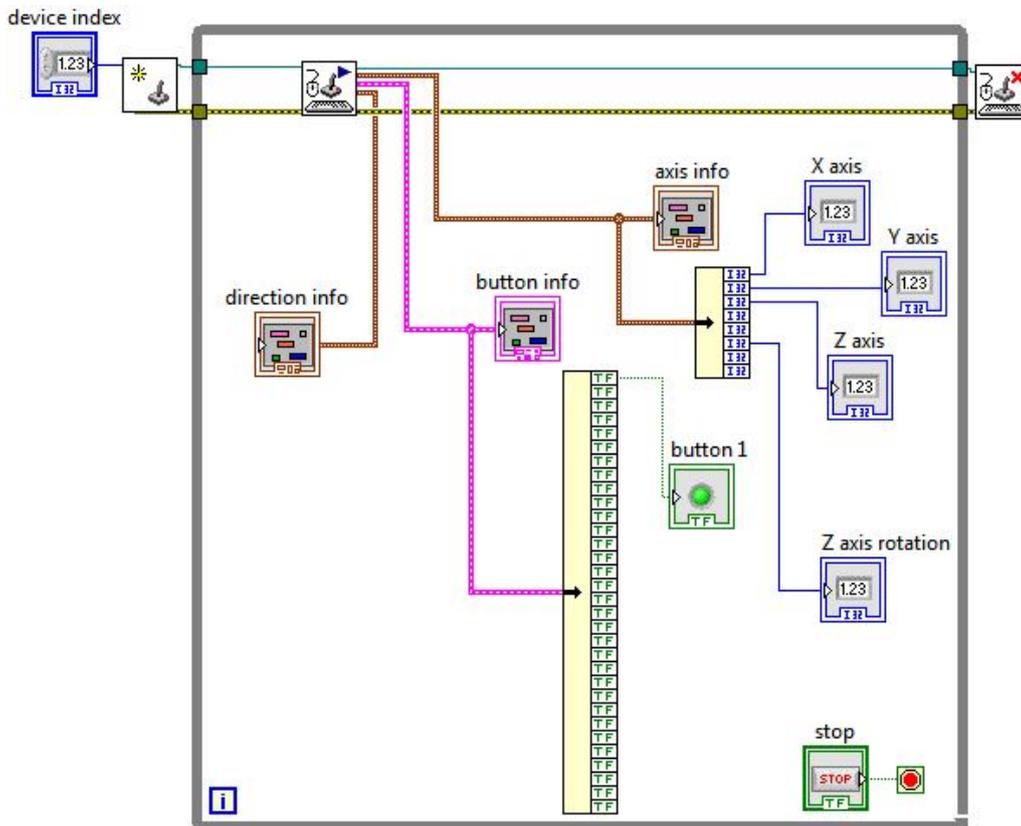


Figura 29. Código usado para recibir los datos del joystick.

Se realiza una transformación de los bits enviados por el mando (-32767 a +32768) a una suma o resta infinitesimal mediante una recta:

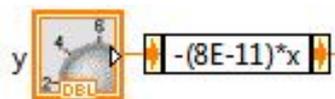


Figura 30. Recta usada para transformar los bits en un incremento de la posición.

De esta manera, mediante el uso del Shift register se suma más o menos longitud en un eje dependiendo del valor que esté enviando el Joystick.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Además, a la salida de la suma acumulada se limita la salida con el siguiente VI:

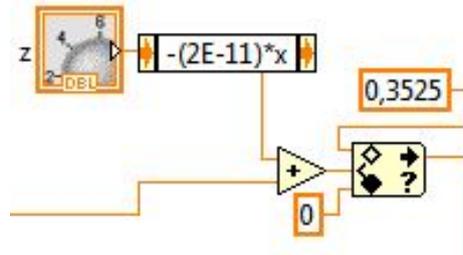


Figura 31. Código usado para transformar los bits en un incremento de la posición y limitar el resultado.

Con esto acotamos la posición en las coordenadas (x, y, z) a sus límites físicos.

Seguidamente se realiza la cinemática inversa aplicando el VI que se ha creado para ello. No obstante éste VI puede devolver que la solución no existe dentro de los límites del brazo robot, es decir, es un punto inalcanzable.

En este caso, como ya se ha explicado en la memoria, la coordenada z se recalcula para que esté dentro de una semiesfera positiva según las coordenadas x e y. Esto se realiza mediante el siguiente "Case structure":

Los booleanos mostrados están unidos de manera que, una vez encuentre una solución, cambien las coordenadas a recalcular, sea imposible de calcular el punto o haya que recalcular la coordenada z, se vuelva a calcular el punto.

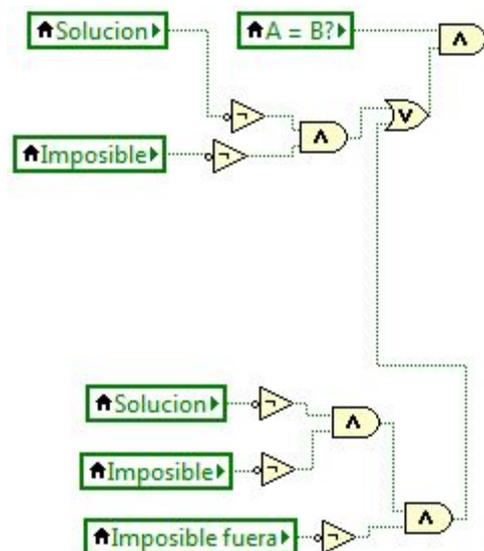


Figura 32. Código usado para relacionar los Booleanos.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Además se ha añadido que cuando las coordenadas x e y lleguen al extremo de la circunferencia que tiene como radio sus límites físicos ($\pm 0,294$) dejen de ir más allá del círculo con este "Case Structure".

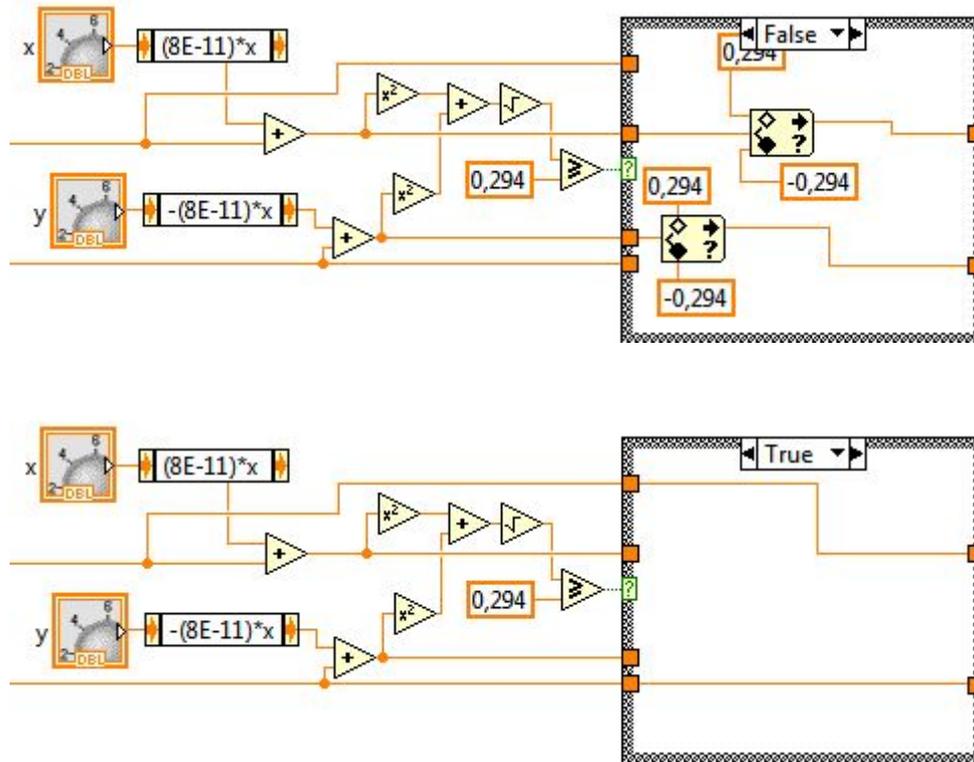


Figura 33. Código usado para limitar el movimiento conjunto de x e y.

9. Movimiento con sensor inercial

Primeramente, como el sensor envía información al puerto "6454" (así se ha impuesto en el código de JavaScript), se obtienen los datos del sensor mediante el siguiente VI:

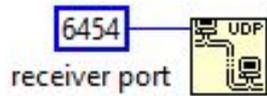


Figura 34. VI usado para obtener los datos del sensor mediante el puerto 6454.

A través de este puerto se recibe un "string" que contiene los tres ángulos de Euler separados por comas. Para transformar este dato a tipo "float", se usa el siguiente VI:

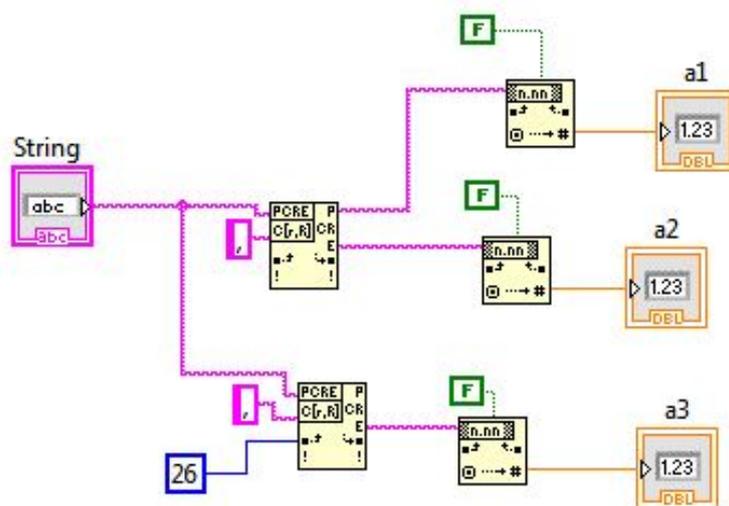


Figura 35. Código usado para convertir el string recibido en tres datos tipo float.

Este VI hace dos funciones:

1. Separa el string cuando encuentra un símbolo "," a una cierta distancia (por ello en uno de ellos empieza a buscar en la posición 26 y el otro, al no poner nada, busca desde el principio).
2. Transforma el "string" resultante a "float".

Escogemos dos movimientos a tener en cuenta: La rotación de la base y la inclinación.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Como en la rotación en la base se tiene mucha exactitud, se ha decidido hacerla directa, así que sólo se han limitado los ángulos que ofrece:

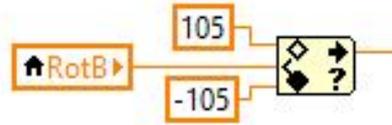


Figura 36. Código usado para transformar la rotación de la base del sensor en la rotación del brazo robot.

Para la inclinación, debido a la inexactitud del sensor, en los casos de $\pm 90^\circ$ ofrece $\pm 80^\circ$.

Por ello, primeramente se limita el valor a $\pm 80^\circ$ y, seguidamente, se transforma mediante una recta a $(-150^\circ) - 150^\circ$.

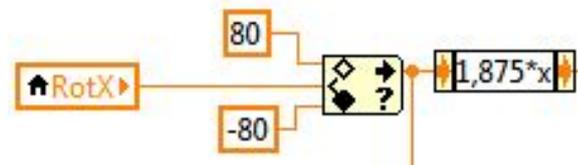


Figura 37. Transformación aplicada a la inclinación del sensor.

Si se obtiene un número de 0 a 150° , el brazo se inclinará hacia la derecha, en un orden de 50° cada articulación por cada 50° nos ofrezca la recta.

Esto significa que si, por ejemplo, obtenemos 106° de la recta, el hombro y codo se inclinarán 50° , mientras que la muñeca se inclinará 6° . El caso negativo es homólogo (sólo se cambia el signo de los ángulos).

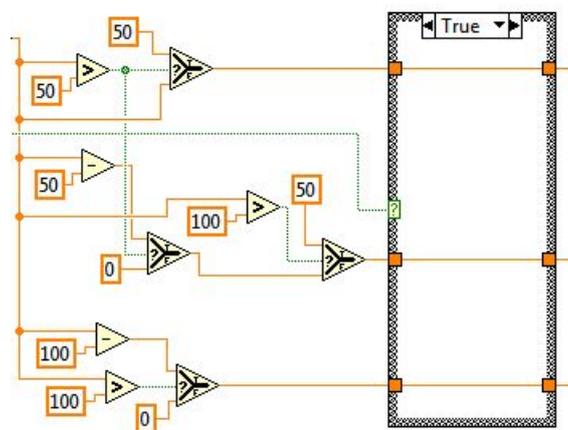


Figura 38. Código usado para transformar la inclinación del sensor en una inclinación de hombro, codo y muñeca del brazo robot.

10. Seguridad

Para implementarla, se ha definido en el dominio tres usuarios: Operario, Técnico e Ingeniero.

Para poder conectarse y desconectarse del sistema de seguridad se ha aprovechado el Event structure que ya había creado.

Para conectarse se ha usado el VI del módulo DSC de LabVIEW “NI Security programmatic login”

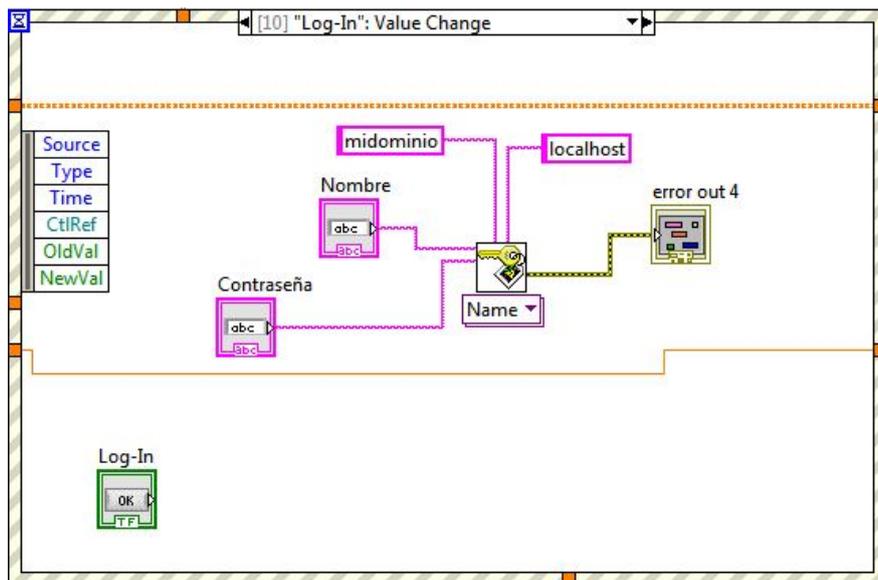


Figura 39. Código usado para la conexión del usuario.

Para desconectarse, se utiliza el VI “NI Security programmatic logout”

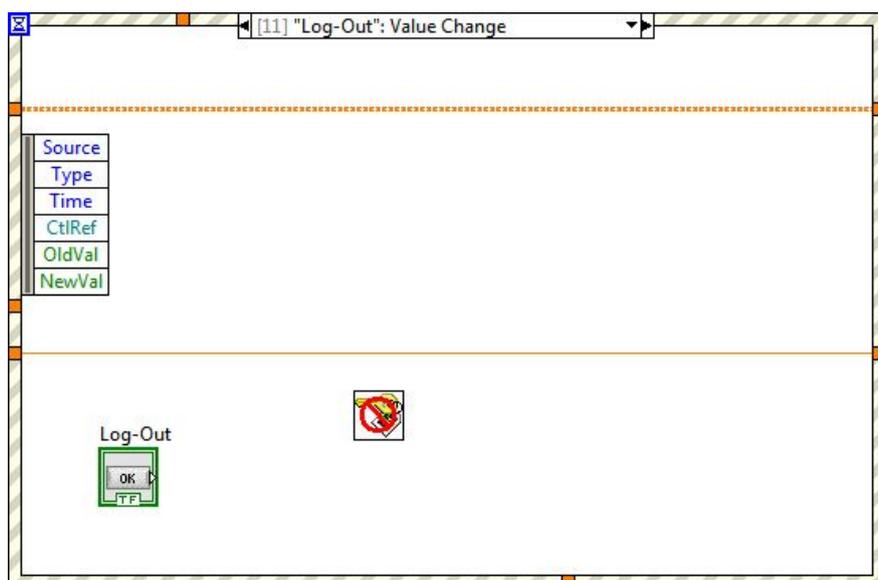


Figura 40. Código usado para la desconexión del usuario.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Finalmente, para saber qué usuario está activo en todo momento, se utiliza el VI “NI Security interactive user info”. Este VI nos da los datos del usuario (hora de conexión, nombre... Etc), de todos ellos se escoge el nombre y se muestra con un indicador.

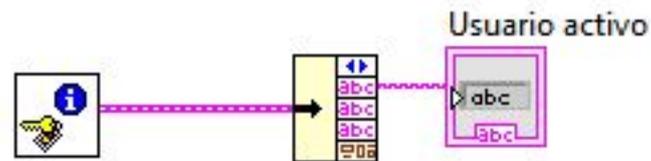


Figura 41. Código usado para obtener la información del usuario activo.

11. Programa para dispositivo Tablet

Para su diseño, se han creado variables compartidas en el proyecto de LabVIEW. Estas variables permiten compartir información de los controles e indicadores entre varios programas.

De esta manera, se han usado tanto en el programa principal como en el programa para el dispositivo Tablet:

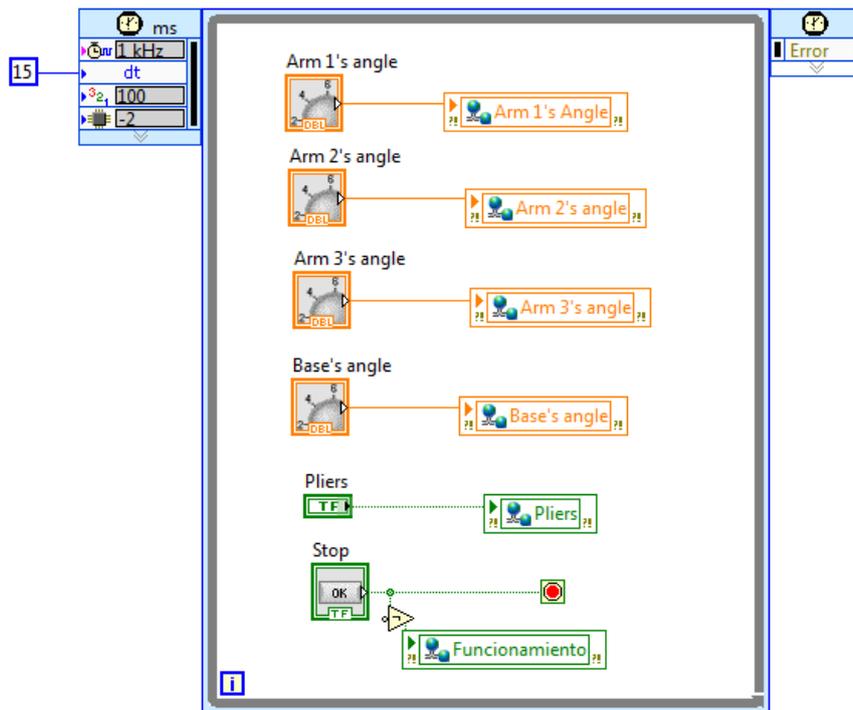


Figura 42. Código usado en la aplicación para dispositivo Tablet.

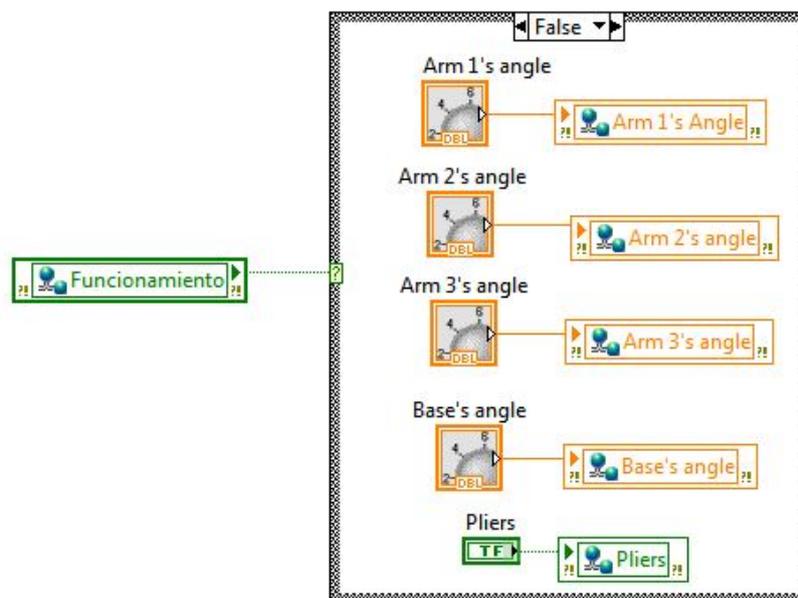


Figura 43. Código usado en la aplicación principal para poder realizar el uso de las Shared variable.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

No obstante, como tanto en el programa principal como en el programa para el dispositivo Tablet escriben sobre la misma variable compartida, necesitamos priorizar una de las dos para que el programa sepa cuál de los dos controles obedecer. Esto se consigue con la variable compartida "Funcionamiento", que hará que los controles del programa principal escriban a la variable compartida sólo si el programa del dispositivo Tablet esta desconectado.

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA
APLICACIÓN EN LABVIEW PARA
EL CONTROL DE TRAYECTORIAS
DEL BRAZO ROBOT, DE 4
ARTICULACIONES CON PINZA,
MODELO S300105 DE
LYNXMOTION

Documento Nº4

MANUAL DE USUARIO

Autor: Carlos Babiera Sancho

Tutor: Juan Manuel Herrero Durá

Curso académico: 2015-16

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

Contenido

1. Objeto.....	1
2. Interfaz	2
2.1 Handmade movement.....	3
2.2 Automatic movement.....	6
2.3 Arm's dimensions.....	8

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE
TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO
S300105 DE LYNXMOTION

1. Objeto

Este anexo tiene como objetivo servir como guía al usuario que vaya a usar la aplicación. Se pretende explicar el uso de todos los controles así como el significado de cada indicador o gráfica realizada por la aplicación.

2. Interfaz

Para iniciar el programa, se tiene que apretar el botón "Run"  en la parte superior izquierda de la aplicación.

Para pararlo, se puede apretar el botón "Stop"  o la "Parada de emergencia"  situado en la parte superior izquierda de la aplicación.

El indicador "Finished late" situado en la zona inferior derecha indica si hay un error en la ejecución de la aplicación.



Figura 1. Indicador de Finished late.

Cabe destacar que la aplicación está constantemente controlada por un sistema de seguridad de usuarios, situado en la parte superior derecha del programa.



Figura 2. Sistema de seguridad empleado en la aplicación.

En este sistema de seguridad hay registrados tres usuarios: Operario, Técnico e Ingeniero. Cada uno tiene sus privilegios (el Operario y el Técnico no puede ver ciertos controles que se explican a continuación).

Se puede saber el usuario con el que se está utilizando el programa en todo momento, así como conectar y desconectarse del sistema de seguridad.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.1 Handmade movement

Al iniciar el programa, automáticamente nos enviará a la pestaña de "Handmade movement".

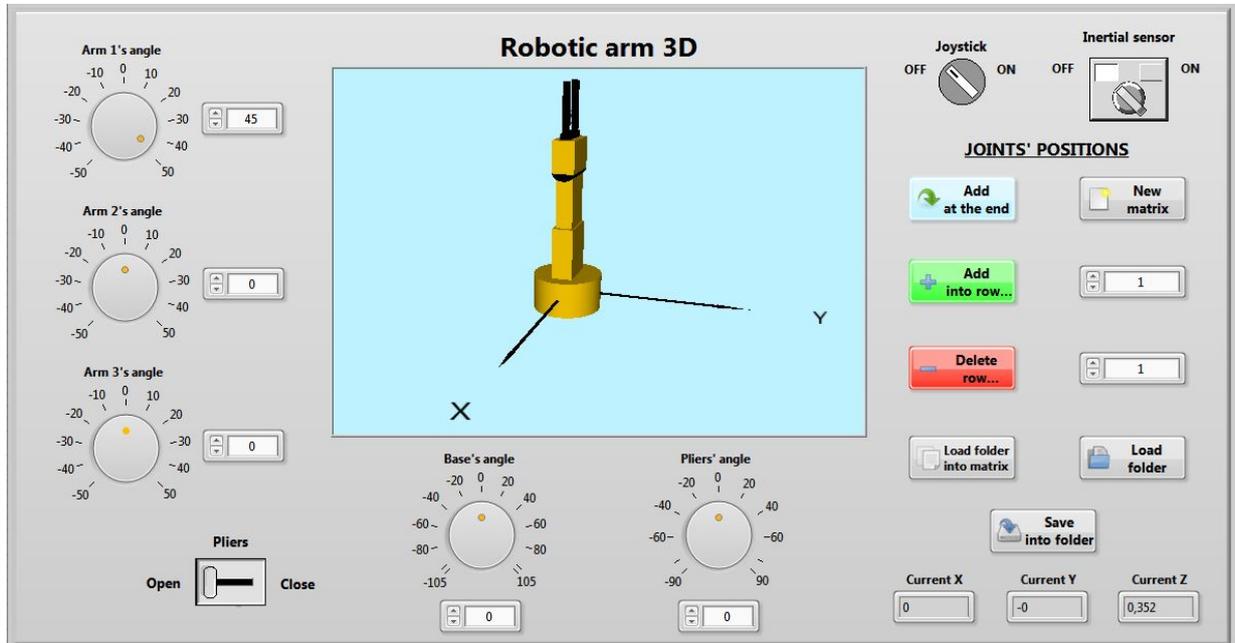


Figura 3. Pestaña de Handmade movement de la aplicación.

En esta pestaña tenemos varias opciones:

- 1 Mover el brazo robot con los controles que nos ofrece la aplicación. Se puede controlar la inclinación del hombro, codo y muñeca, así como la rotación del hombro y de la pinza. También ofrece la opción de abrir o cerrar la pinza a voluntad mediante un interruptor.



Figura 4. Control "Knob" e interruptor usados en la aplicación.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

- 2 Activar el movimiento por joystick mediante el siguiente pulsador:



Figura 5. Interruptor del joystick.

- 3 Activar el movimiento por sensor mediante el siguiente pulsador:

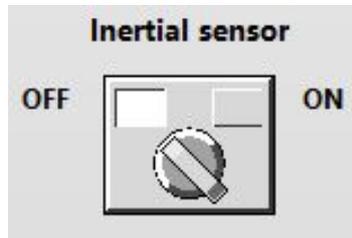


Figura 6. Interruptor del sensor.

- 4 Crear una matriz de puntos del espacio (x, y, z) para, posteriormente, realizar una trayectoria con ellos. Esta matriz se puede visualizar en cualquier pestaña debido a su importancia en la aplicación.

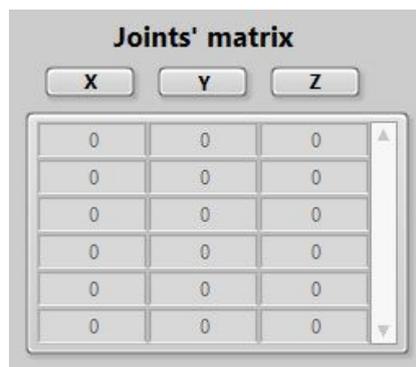
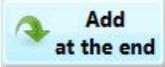


Figura 7. Matriz de puntos.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Para crear esta matriz, disponemos de diferentes pulsadores:

-  Permite añadir el punto actual al final de la matriz actual.
-  Permite eliminar la fila de la matriz seleccionada.
-  Permite añadir el punto a la fila de la matriz seleccionada.
-  Permite el reinicio de la matriz a una vacía.
-  Permite guardar la matriz en un fichero "txt".
-  Permite cargar un fichero "txt" en la matriz, reemplazando la matriz actual.
-  Permite cargar un fichero "txt" en la matriz, añadiendo sus puntos a la matriz actual.

Cabe destacar que la opción Joystick se anulará si se activa la opción Sensor y viceversa.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.2 Automatic movement

En esta pestaña se mostrará lo siguiente:

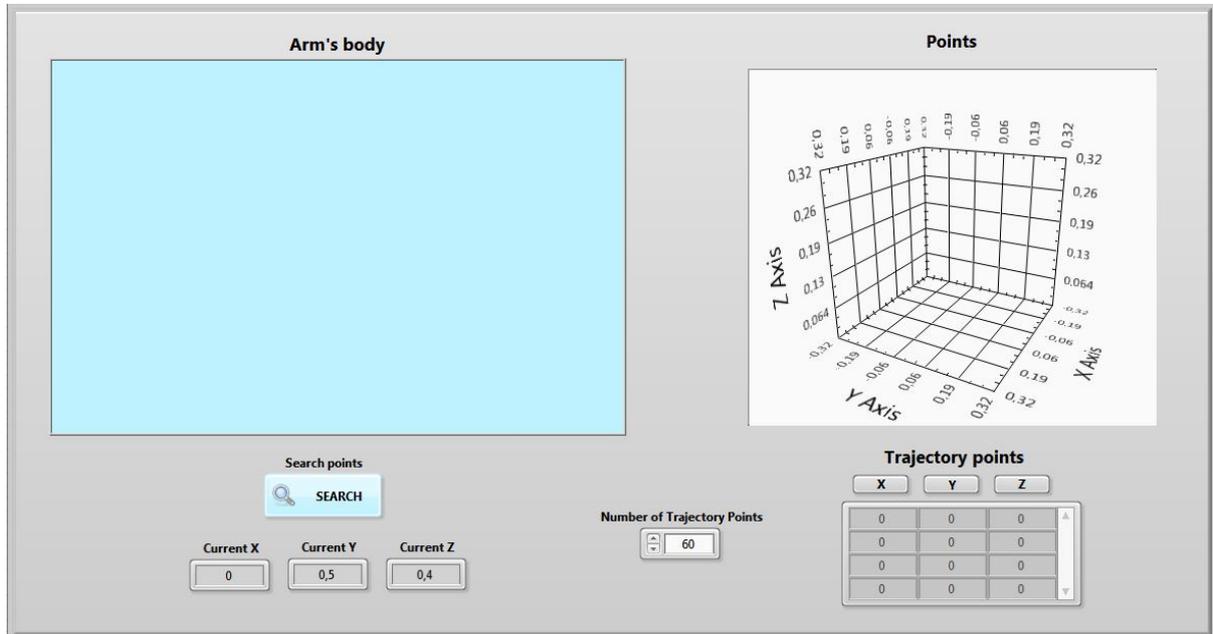


Figura 8. Pestaña Automatic Movement de la aplicación.

Esta pestaña realiza la trayectoria y muestra los puntos en el espacio recorridos (x, y, z) tanto de forma gráfica como en forma de matriz.

Tras pulsar el pulsador "Search" , se representa el esqueleto del brazo:

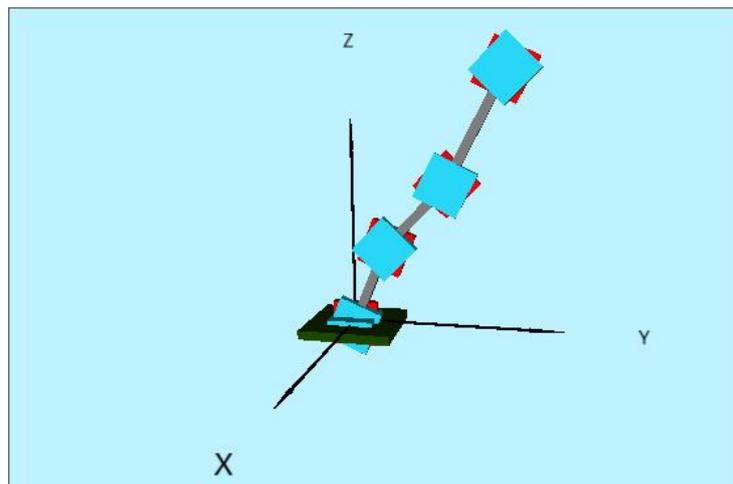


Figura 9. Representación del esqueleto del brazo robot.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

La única modificación que se puede hacer de la trayectoria es a través del número de puntos intermedios de ésta, que derivará en un cambio a su velocidad.

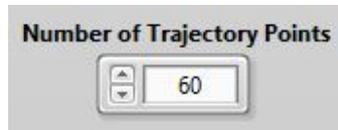


Figura 10. Control de los puntos intermedios de la trayectoria.

La gráfica de puntos muestra el camino recorrido por la trayectoria. Los puntos rojos mostrados son los puntos a alcanzar, mientras que los puntos verdes representan los puntos que ya se han alcanzado.

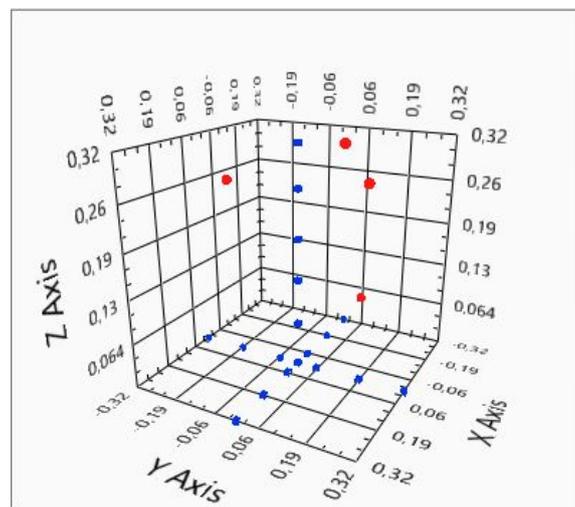


Figura 11. Gráfica de puntos en el espacio de la trayectoria.

A medida que la matriz de puntos se crea, se van creando los puntos rojos a alcanzar en la gráfica.

También se puede observar que, para una mejor representación, se han creado puntos azules a modo de ejes X, Y, Z.

Y finalmente, también se muestra la matriz de los puntos alcanzados por la trayectoria para que se pueda comparar de una manera más sencilla el error cometido al realizar la cinemática inversa.

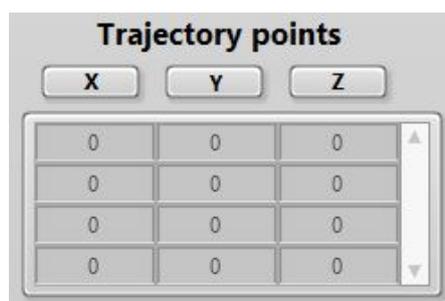
A screenshot of a LabVIEW control panel titled "Trajectory points". It features three buttons labeled "X", "Y", and "Z" at the top. Below these buttons is a 4x3 grid of input fields, each containing the number "0". A vertical scrollbar is visible on the right side of the grid.

Figura 12. Matriz de puntos de la trayectoria.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

2.3 Arm's dimensions

En esta pestaña se configura las dimensiones del brazo robot.

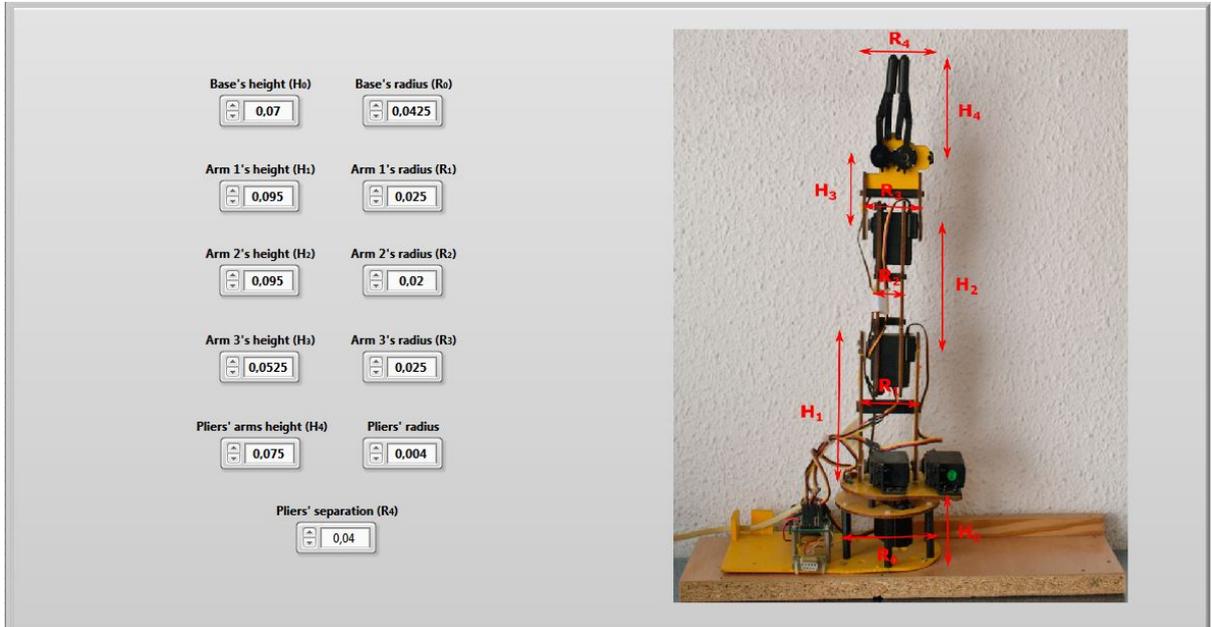


Figura 13. Pestaña Arm's dimensions de la aplicación.

Como se puede observar, hay una imagen para facilitar la comprensión de las dimensiones cambiadas.

Cabe mencionar que los cambios no se efectuarán mientras se ejecuta el programa: Es necesario un reinicio de éste.

DESARROLLO DE UNA APLICACIÓN EN LABVIEW PARA EL CONTROL DE TRAYECTORIAS DEL BRAZO ROBOT, DE 4 ARTICULACIONES CON PINZA, MODELO S300105 DE LYNXMOTION

Bibliografía

Correcher, A. (2005). *Manual de uso de los robots S300105*.

Craig, J. (2006). *Robótica*.

Hernández, M., Ortiz, M. B., Calles, C. A. y Rodríguez, J. C. (2015). *Robótica. Análisis, modelado, control e implementación*.

Jietai (2015). *CC3D flight control board. User's Manual*.

Lajara, J. R., Pelegrí, J. (2011). *LabVIEW. Entorno gráfico de programación*.

National Instruments (s.f.). Obtenido el 1 de Octubre de 2015, de <https://www.ni.com/getting-started/labview-basics/esa/>

Saitek (s.f.). Obtenido el 13 de Junio de 2016. *Cyborg evo Force. User manual*.

