



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE MASTER EN INGENIERÍA INDUSTRIAL

DISEÑO E IMPLEMENTACIÓN DE UN BANCO DE ENSAYOS PARA LA CARACTERIZACIÓN DE SISTEMAS MECÁNICOS

AUTOR: ALBERTO DÍAZ RODRÍGUEZ

TUTOR: ANA MARÍA PEDROSA SÁNCHEZ

COTUTOR: ANTONIO CORRECHER SALVADOR

Curso Académico: 2020-21

AGRADECIMIENTOS

*A mis padres, hermanos y a toda mi familia,
que siempre han estado cerca ante las adversidades.*

*A mis compañeros y amigos, por el apoyo
que han supuesto para recorrer este camino.*

A los profesores, por todo lo que me han enseñado.

*A mis tutores Ana y Antonio, y a Manuel por su inestimable
ayuda y conocimientos que se ven reflejados en este trabajo.*

Personas que han estado a mi lado todo este tiempo.

Gracias

RESUMEN

En el presente Trabajo de Fin de Máster se diseñará e implementará un banco de ensayos para la caracterización de sistemas mecánicos. El accionador del sistema será un motor lineal de alta potencia gobernado por un controlador de motor comercial. La adquisición de medidas, así como la coordinación del sistema se realizará desde un controlador de tiempo real CompactRio de National Instruments. La supervisión de todos los elementos se realizará desde un PC que comunicará con los distintos elementos mediante Ethercat para garantizar la máxima velocidad y sincronización. El resultado de este se incorporará a un prototipo de ensayos HIL que simula la interacción entre un pantógrafo real y una catenaria virtual.

Palabras clave: EtherCAT, LabVIEW, Motores lineales, CompactRIO.

RESUM

En aquest Treball de Fi de Màster es dissenyarà i implementarà un banc d'assajos per a la caracterització de sistemes mecànics. L'accionador del sistema serà un motor lineal d'alta potència governat per un controlador de motor comercial. L'adquisició de mesures, així com la coordinació del sistema es realitzarà des d'un controlador de temps real CompactRio de National Instruments. La supervisió de tots els elements es realitzarà des d'un PC que comunicarà amb els diferents elements mitjançant Ethercat per a garantir la màxima velocitat i sincronització. El resultat d'aquest s'incorporarà a un prototip d'assajos HIL que simula la interacció entre un pantògraf real i una catenària virtual.

Paraules clau: EtherCAT, LabVIEW, Motors lineals, CompactRIO.

ABSTRACT

In this Master's Degree Final Project, a test bench for the characterization of mechanical systems will be designed and implemented. The actuator of the system will be a high power linear motor governed by a commercial motor controller. The acquisition of measurements, as well as the coordination of the system, will be carried out from a National Instruments CompactRio real-time controller. The supervision of all the elements will be carried out from a PC that will communicate with the different elements through Ethercat to guarantee maximum speed and synchronization. The result of this will be incorporated into a HIL test prototype that simulates the interaction between a real pantograph and a virtual catenary.

Keywords: EtherCAT, LabVIEW, Linear motors, CompactRIO.

Índice

Capítulo 1. Introducción	8
1.1 Antecedentes	9
1.2 Presentación del proyecto.....	10
1.3 Contenido de la memoria	11
Capítulo 2. Marco teórico.....	13
2.1 Motores lineales	14
2.2 LabVIEW	18
2.3 EtherCAT.....	22
2.4 Controlador E1450-EC-QN-0S.....	24
2.5 Medida de la posición del vástago	26
2.6 SolidWorks	28
2.7 LinMot-Talk	29
Capítulo 3. Realización práctica.....	39
3.1 Elementos del banco de ensayos.....	40
3.2 Diseño del banco de ensayos	45
3.3 Conexión de los equipos.....	48
3.4 Selección del sensor de posición	51
3.5 Definición del fichero ENI	59
3.5.1 PDO de entrada	65
3.5.2 PDO de salida.....	66
3.6 Programa para el control del motor lineal.....	70
3.7 Descripción de un movimiento cíclico tipo rampa.....	81
3.8 Otras curvas de movimiento	83
3.9 Elementos adicionales del software	92
3.10 Incorporación del CompactRio	93
Capítulo 4. Conclusiones y líneas futuras	100
4.1 Conclusiones.....	100
4.2 Líneas futuras	100
Capítulo 5. Presupuestos.....	102
Capítulo 6. Anexo ampliación para MS01-1/D-SSI.....	104
Capítulo 7. Bibliografía	107

Capítulo 1. Introducción

1.1 Antecedentes

En el Centro de Investigación de Ingeniería Mecánica de la Universidad Politécnica de Valencia se plantearon hace unos años la posibilidad de optimizar el transporte ferroviario. Para ello, se enfocaron en los fenómenos que se producen en la interacción entre el pantógrafo y la catenaria. El pantógrafo es el elemento mecánico que, desde la parte superior del tren, toca el tendido eléctrico para poder accionar los motores, y la catenaria es la red que sustenta el tendido eléctrico a lo largo de toda la vía.

Profundizando un poco más en la idea, optimizando el contacto entre estos dos elementos, los cuales presentan una elevada relevancia para el coste de funcionamiento del medio de transporte, se podía reducir el coste económico. En concreto, por la disminución del desgaste de las piezas, con el consiguiente ahorro en mantenimiento, y por la mejora de la eficiencia energética con un contacto entre los elementos lo más constante posible, de modo que no se produzcan cortes en el suministro.

El tren, en su movimiento, provoca unas vibraciones tanto en el sentido de avance, como de forma perpendicular, así como oscilaciones que se mantienen a lo largo del recorrido, pero que no son constantes en todo momento.

Dentro del proyecto al que se ha hecho referencia, se diseñó un sistema que permitía simular la interacción de un pantógrafo real sobre una catenaria virtual, el cual incluye un modelo matemático de cálculo de la posición de la catenaria al que se hará referencia más adelante en el documento. El sistema diseñado obtuvo el premio Talgo a la innovación tecnológica en su decimoséptima edición [1], y el modelo matemático que desarrolló en su tesis el profesor Santiago Gregori Verdú, permite simular el movimiento que se produce en la catenaria, ante unas determinadas variables físicas relativas tanto al propio tren, como a la estructura sobre la que se sustenta la misma.

Por todo lo expuesto, resulta interesante la recreación de un banco de ensayos, sobre el que se pueda simular el comportamiento de una catenaria virtual, en su interacción con un pantógrafo, que en este caso es real.

El banco anterior presentaba limitaciones en el tipo de catenarias que se podían simular, y en la adquisición y procesado de datos. Por esta razón, se presenta este proyecto, con el objeto de avanzar más en el desarrollo de la investigación mencionada.

Después del estudio anterior, se desarrolló el proyecto del MINECO: VIRTUALCAT [2], en el que se recreó y mejoró el banco de ensayos diseñado para dotarlo de capacidad de simulación de catenarias más complejas y mejorar la adquisición y procesado de datos. En la siguiente imagen se puede ver el estado actual del banco de ensayos.



Figura 1: Banco de ensayos de la catenaria virtual.

1.2 Presentación del proyecto

El proyecto expuesto en el apartado anterior cuenta ya con años de desarrollo en los que cada vez los modelos, tanto teórico, como la implementación del banco de ensayos, son cada vez más precisos, lo cual se traduce en la posibilidad de extraer conclusiones más acertadas en lo relativo a la mejora de la eficiencia del transporte ferroviario.

Como se ha comentado, existe un modelo matemático que permite calcular a gran velocidad la posición teórica a la que se encontraría el punto de contacto de la catenaria ante unas determinadas condiciones relacionadas principalmente con el pantógrafo del tren que está circulando. El factor más restrictivo a la hora de recrear el ensayo en la realidad es que los elementos mecánicos reproduzcan los movimientos deseados a la velocidad y precisión que determina el modelo matemático.

Continuando con las ideas presentadas en el apartado anterior, en el momento en el que se comenzó el presente Trabajo de Fin de Máster, el banco de ensayos podía reproducir el movimiento vertical de la catenaria [3], y se pretendía de cara al futuro incorporar la posibilidad de que se pudiesen realizar movimientos en horizontal. Este tipo de movimiento también se presenta en la realidad al estar dispuesta la catenaria en zigzag, del modo en el que se muestra en la siguiente figura [4]. Por lo tanto, se debía incorporar un nuevo motor lineal al sistema.



Figura 2: Ejemplo de catenaria en zigzag.

Por otro lado, en el sistema establecido, aún cabía un gran margen de mejora en lo relativo a la velocidad de comunicación, para de ese modo poder recrear con la mayor fidelidad posible, la curva de movimiento que se obtiene a través del modelo matemático previamente citado. El bus de campo EtherCAT parecía una alternativa adecuada al sistema que estaba implantado previamente, ya que, por sus características de funcionamiento, la transmisión de la información entre el maestro y el esclavo es muy veloz. En concreto, es del orden de los microsegundos, pero depende del número de equipos presentes en la red. Por otro lado, resultaba adecuado para los intereses de la investigación, que el sistema funcionase en tiempo real, por lo que, en las etapas finales de este trabajo, resultaba adecuado que se pudiese implementar la programación en un equipo CompactRIO [5].

El tipo de motor lineal, así como la metodología de comunicación a emplear, no resultaban sencillos. Junto a esto, las fuerzas y velocidades que podía alcanzar el actuador, suponían un grado de peligrosidad el cual propiciaba la realización de un nuevo banco de ensayos independiente para el mismo, donde realizar las pruebas de funcionamiento. Además, esto permitiría, junto a la incorporación de esta idea a nivel de software, servir de banco de ensayos para la caracterización de este tipo de motores.

A lo largo de la memoria se irán presentando las diferentes etapas que atravesó el presente proyecto, así como las dificultades que hubo que solucionar en cada etapa del desarrollo.

1.3 Contenido de la memoria

La presente memoria del Trabajo de Fin de Máster se estructura en siete capítulos. En el primero de ellos se presenta el proyecto realizado, partiendo de la necesidad que se detectó y, posteriormente, se expone la idea inicial en torno a la que se desarrollará el trabajo.

El segundo capítulo analiza desde el punto de vista teórico, aquellos conceptos que intervienen y que se han tenido que estudiar para comprender, tanto a nivel software, como hardware, los elementos del sistema. Este desarrollo, en el que se llevó a la realidad las ideas expuestas, se

expone en el tercer capítulo, donde se presenta el desarrollo mecánico del banco de ensayos, y el software implementado en LabVIEW para el control de los equipos que componen el conjunto.

El cuarto capítulo ofrece una conclusión al proyecto, y analiza las posibles vías de mejora posibles, ya que como se ha expresado al comienzo de la memoria, el proyecto global es un desarrollo de varios años, y seguirá en el futuro, dado lugar de este modo, a un modelo cada vez más aprovechable para la mejora de la eficiencia del transporte ferroviario.

En el quinto capítulo se presenta un presupuesto en el que se analiza desde el punto de vista económico el proyecto, y se concluye el documento con un sexto capítulo, destinado a la recopilación de la bibliografía que ha resultado necesario para llevar a cabo este trabajo.

Capítulo 2. Marco teórico

2.1 Motores lineales

A la hora de buscar la mejor alternativa para el movimiento de elementos mecánicos en un proceso industrial, existen multitud de elementos de juicio a tener en cuenta, ya que las alternativas de actuadores disponibles para seleccionar son numerosas.

Principalmente, los accionamientos se pueden agrupar en dos grandes grupos [6], los rotativos y los lineales.

Los motores rotativos se suelen utilizar cuando existe una necesidad de realizar un movimiento de giro en un eje, no obstante, existen cadenas de transmisión, como por ejemplo el mecanismo husillo-tuerca, con el que se puede transformar un movimiento de rotación en uno de traslación. Algunos ejemplos de este tipo de accionamientos que se pueden enunciar son:

- Motores oleohidráulicos, que son turbinas movidas mediante un flujo de aceite a presión.
- Motores neumáticos, análogos a los anteriores, pero empleando aire comprimido.
- Motores eléctricos paso a paso, los cuales tienen la destacada característica, con motivo de su construcción, de permitir giro de ángulos controlados bastante precisos, y que por tanto posibilitan un control de la posición muy certero. El inconveniente es que habitualmente se tratan de actuadores que trabajan a una velocidad reducida.
- Motores de corriente continua, los cuáles se controlan en régimen de giro y par, mediante el control de la tensión e intensidad de alimentación. Su gran inconveniente es la necesidad de emplear escobillas, con el correspondiente gasto en mantenimiento que ocasiona.
- Motores de corriente alterna asíncronos, que consisten en un rotor estator bobinado alimentado con corriente alterna, y un rotor sin alimentación. Este sistema, al funcionar induciendo un campo magnético en el rotor, que por esta razón es pasivo, resuelve el problema de la necesidad de usar escobillas, pero la regulación del par y régimen de giro es más compleja que en el caso anterior.
- Motores de corriente alterna síncronos, en los que el rotor es activo, alimentado mediante corriente continua, lo cual genera un campo magnético que se sincroniza con el del estator. De nuevo, no resulta necesario el uso de escobillas. En este caso, la regulación de par y velocidad de giro es más sencilla que en el caso anterior, y se realiza actuando sobre la frecuencia, tensión e intensidad de alimentación.

Por otro lado, existen los accionamientos lineales, los cuales, como su nombre indica, describen movimientos lineales de elementos mecánicos. La dificultad para controlar los desplazamientos es superior a la de los accionamientos rotativos, pero la ventaja es la no necesidad de tener que emplear un sistema de transmisión del movimiento para convertirlo en lineal, lo cual genera pérdidas de eficiencia, limitaciones mecánicas de la dinámica, e imprecisiones. Esto último, en el proyecto que se desarrolla en este documento, resulta de gran importancia. Además, el movimiento que se pretende describir, tanto en el banco de ensayos, como en la implementación posterior, es completamente lineal, por lo que este tipo de accionamientos se ajusta a las necesidades del proyecto.

Los accionamientos lineales pueden ser hidráulicos (oleohidráulicos o neumáticos), o eléctricos, siendo este último tipo el que se ha escogido, ya que atendiendo a la aplicación para la que se requiere el motor, en la que se precisa de grandes velocidades y precisión, tanto en movimientos de avance como de retroceso, junto a que es más sencillo disponer de una fuente eléctrica que de un flujo hidráulico a presión, resulta más adecuado.

Entrando en un mayor grado de detalle, se ha seleccionado un motor lineal eléctrico de la marca LinMot [7]. Este fabricante, produce motores lineales tubulares, especialmente diseñados para aplicaciones de dinámica extremadamente elevada, lo cual es muy positivo para los intereses del proyecto. En concreto, cuentan con modelos de hasta 9 m/s de velocidad con aceleraciones que superan los 1010 m/s², fuerzas de hasta 2720 N, y recorridos máximos cercanos a los 2 metros.



Figura 3: Motor lineal del fabricante LinMot.

El diseño constructivo de este tipo de equipos permite convertir la potencia eléctrica en un trabajo de desplazamiento lineal sin la presencia de mecanismos intermedios. Para ello, dispone de unas bobinas presentes en el estator y unos imanes permanentes de neodimio presentes en este caso en la recta del eje, los cuales se pueden ver en la figura 6. La corriente que circula por las bobinas del estator genera un campo magnético, el cual interactúa con el campo magnético producido por los imanes permanentes antes mencionados. De este modo, se genera una fuerza lineal que desplaza el vástago del motor en un sentido marcado por la secuencia de fases de la corriente alterna trifásica de alimentación del estator.

En cuanto a la regulación, el fabricante también suministra controladores con los que se ajusta la intensidad y tensión aplicada a las bobinas de modo que la fuerza generada sea la deseada.

Junto a todo lo anterior, estos equipos cuentan con un codificador de movimiento tipo *encoder*, el cual permite la realización controlada de trayectorias, ya que este dispositivo permite poder acceder a la información del desplazamiento descrito desde el punto anterior en el que se encontraba el sistema.

A la hora de seleccionar el motor lineal correcto para una determinada aplicación, se debe tener en cuenta la fuerza que resulta necesario generar, la carrera máxima en la que se va a desplazar el vástago, las masas que se van a desplazar, la secuencia de movimientos que se va a describir,

el método de comunicación entre el motor y el resto de los equipos del sistema, así como las características ambientales del entorno de trabajo.

Atendiendo a las citadas consideraciones, el motor escogido fue el PS10-70X240U-BL-QJ-D01 [8], del cual, analizando la hoja de características, se puede destacar que su carrera máxima es de 1610 mm, la máxima fuerza que puede describir es de 1650 Nm, y mediante la alimentación trifásica que se va a emplear, se puede alcanzar una velocidad de 5.4 m/s. Todo ello con una resolución de 0.005 mm y una repetibilidad de ± 0.05 mm. Todas estas características cumplen con las necesidades de la aplicación final en la que se instalará el motor.

Resulta conveniente mencionar en este punto, que, para manejar el motor, es necesario emplear un controlador. En este caso, para el motor expuesto, y teniendo en cuenta que la comunicación se hará bajo el protocolo de comunicación EtherCAT, el cual se explicará en el apartado 2.3, se ha seleccionado el E1450-EC-QN-0S [9].



Figura 4: Controlador E1450-EC-QN-0S.

Este controlador, especialmente ideado para el accionamiento axial de los ejes de la marca, aunque también es posible emplearlo en otras aplicaciones, cuenta con una resolución de 32 bits para la posición del vástago, y una etapa de potencia en corriente alterna con la que es posible alimentar posteriormente al motor.

La comunicación entre ambos dispositivos se realizará mediante un cable KSS05-02/08-D15/J-3 [10] como el de la siguiente figura, y que transmite la información del encoder presente en el cuerpo del motor.



Figura 5: Cable KSS05-02/08-D15/J-3.

Previo a la finalización de este apartado del escrito, se va a detallar con mayor grado de detalle el funcionamiento del motor lineal empleado. Las diferentes partes del equipo se pueden observar en la siguiente imagen, la cual se ha extraído de un manual [11] elaborado por el fabricante.

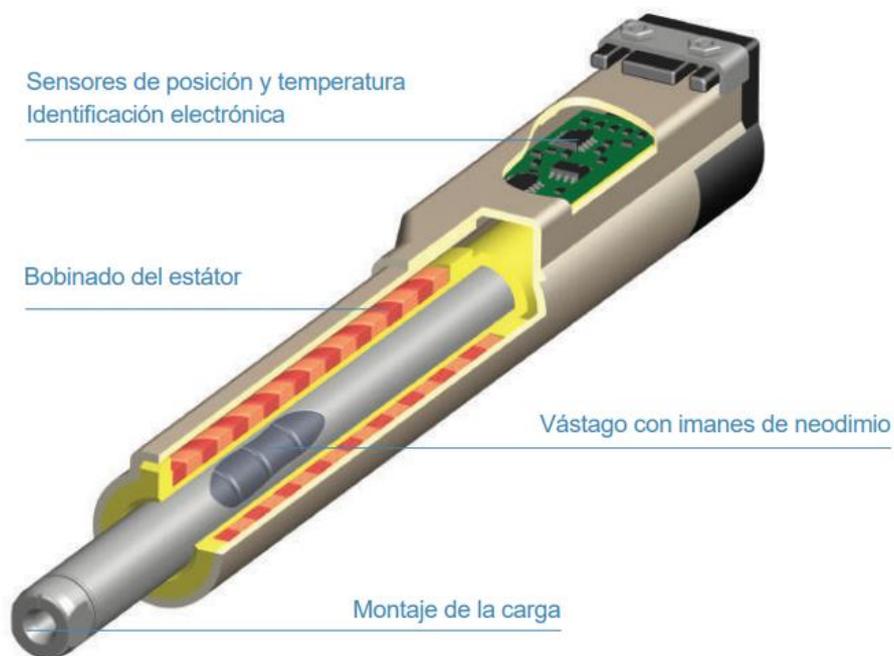


Figura 6: Esquema de funcionamiento de los motores lineales.

En la parte de la electrónica se encuentran los componentes que permiten actuar sobre el bobinado del estator, el encoder con el que se obtiene la información de la posición durante el funcionamiento, y sensores destinados a proteger el motor, por ejemplo, ante un sobrecalentamiento.

El principio físico bajo el que se rige el funcionamiento del equipo es el electromagnetismo. El paso de corriente alterna trifásica por el bobinado del estator genera un campo magnético que atrae al campo asociado al imán permanente presente en el vástago [12]. Dependiendo de la secuencia de las fases de la corriente de alimentación, el desplazamiento se producirá en un

sentido, o en el opuesto. Será el controlador del motor el que se encargue de realizar este tipo de ajustes en el estado del movimiento. La ventaja principal de este sistema es que el mantenimiento del sistema es bajo, ya que no es necesario incorporar escobillas para alimentar el vástago.

Por último, la zona del montaje de la carga será el punto en el que, en el banco de ensayos, se atornille el carrito que se desplaza siguiendo el movimiento del vástago.

2.2 LabVIEW

La programación para comunicar el ordenador con el controlador E1450-EC-QN-0S, y que este a su vez envíe las ordenes de movimiento al motor, se realizará en LabVIEW [13], que es un entorno de programación gráfico, desarrollado por National Instruments, en el que se puede controlar un proceso industrial realizando ensayos, toma de datos, o el control del funcionamiento del mismo. En este caso, será esta última característica la que más de aproveche.

Las dos funcionalidades más destacables de LabVIEW son la integración de dispositivos hardware, como en este caso será el controlador del motor, y el sensor de desplazamiento lineal, los cuales se podrán controlar desde el ordenador en el que se esté ejecutando el programa, incluyendo la posibilidad de realizar ajustes durante la ejecución. La otra gran característica es que es que la programación se realiza de forma gráfica, lo cual resulta muy intuitivo, definiendo las funcionalidades a través de diferentes bloques generando un diagrama de flujo, en el cual se pueden definir acciones que ocurren paralelamente. Estos bloques permiten establecer y finalizar la conexión con los diferentes dispositivos hardware, realizar operaciones matemáticas, crear representaciones gráficas de los datos, o introducir datos a través del teclado o ficheros de texto entre otras muchas funcionalidades.

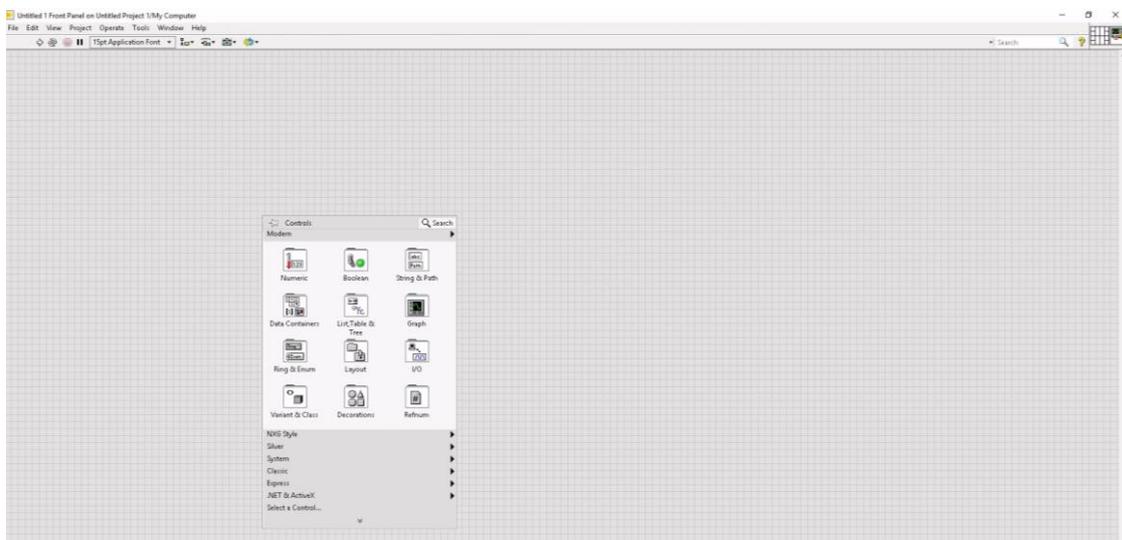


Figura 7: Panel frontal del LabVIEW.

En la anterior imagen se puede ver el panel frontal de la interfaz, donde se disponen los controles, lo cuales permiten modificar los parámetros de entrada del programa (botones, valores numéricos, textos, listas desplegadas y otros), tanto durante la ejecución del programa, como previamente. También permite definir los indicadores (Gráficas, iconos, diagramas y

otros) que se van a utilizar, si por ejemplo se quiere ver el valor de un dato del programa, o si se desea visualizar algún tipo de variable de un modo gráfico. Bajo el término de terminales se conoce al conjunto de controles e indicadores, y estos se encuentran en la paleta de controles que se puede ver en la imagen.

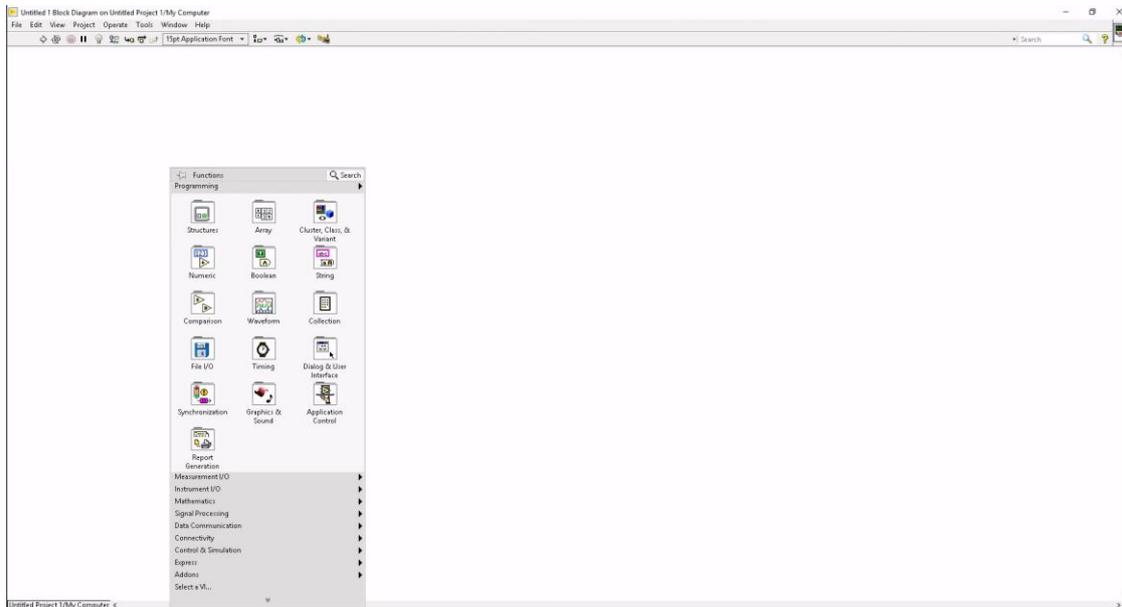


Figura 8: Panel de diagrama de bloques de LabVIEW.

Cuando ya se disponen de los elementos de entrada y representación de los datos necesarios, hay que definir la lógica interna del programa. Es decir, procesar la información para obtener los resultados objetivo del software. Esto se realiza en la otra ventana, la cual se muestra en la figura 8, y que se denomina diagrama de bloques, y en la que con el botón derecho del ratón se abre la paleta de funciones, donde se encuentran los bloques necesarios para el procesamiento de los datos, así como las estructuras necesarias para definir bucles si resulta necesario. Se puede alternar rápidamente entre ambas ventanas pulsando simultáneamente las teclas Ctrl y E del teclado.

En ambas ventanas es común la paleta de herramientas, a la que se accede con la combinación de teclas Mayús y botón derecho del ratón.



Figura 9: Paleta de herramientas de LabVIEW.

Algunos de los iconos más relevantes son los siguientes:

	Se utiliza para modificar los valores de un control o un texto.
	Permite posicionar los objetos, y también cambiar su tamaño.
	Permite añadir textos en la interfaz, los cuales pueden resultar aclaratorios de una determinada funcionalidad.
	Se utiliza para conectar los terminales de los distintos bloques que se utilicen en el programa.
	Permite desplazar los elementos de la interfaz e interactuar con los mismos.
	Se utiliza para establecer un punto de ruptura (" <i>breakpoint</i> ") en un punto del programa, de modo que cuando la ejecución llegue a dicho elemento se detenga. Es una herramienta de depuración.
	Permite ver los valores que toman las variables en ese punto de la ejecución, pero en este caso sin detenerla.
	Se utiliza cuando se quiere realizar una única ejecución del programa.
	Realiza una ejecución continuada (al terminar vuelve a iniciarse el ciclo).
	Detiene la ejecución del programa actual.
	Pausa el programa que se está ejecutando.

Existen multitud de tipos de datos, tipos de estructuras de programación, y terminales en LabVIEW. En este caso, en el tercer capítulo se puede encontrar un apartado en el que se analiza el software desarrollado para el control del motor. Será en dicho apartado en el que se analicen los diferentes elementos que han resultado necesarios.

Entrando a analizar el caso concreto de la aplicación desarrollada, ha sido necesario instalar la librería EtherCAT para LabVIEW [14], la cual permite el intercambio de información con esclavos de una red EtherCAT, siendo el ordenador en el que se ejecuta el programa el maestro de dicha red.

La instalación de la librería es sencilla, ya que el instalador se encarga de ubicar correctamente los archivos en la carpeta en la que se encuentra el software instalado. Una vez que se ha realizado esta operación, en la paleta de funciones se pueden encontrar los nuevos bloques, pertenecientes a la librería, los cuales ya se pueden probar ya que la librería incluye una serie de ejemplos, los cuales resultan de gran utilidad para entender el funcionamiento de los distintos bloques. Estos ejemplos se encuentran en la carpeta Archivos de programa (x86)\National Instrument\LabVIEW 2018\examples\Ackermann Automation\EtherCAT Library.

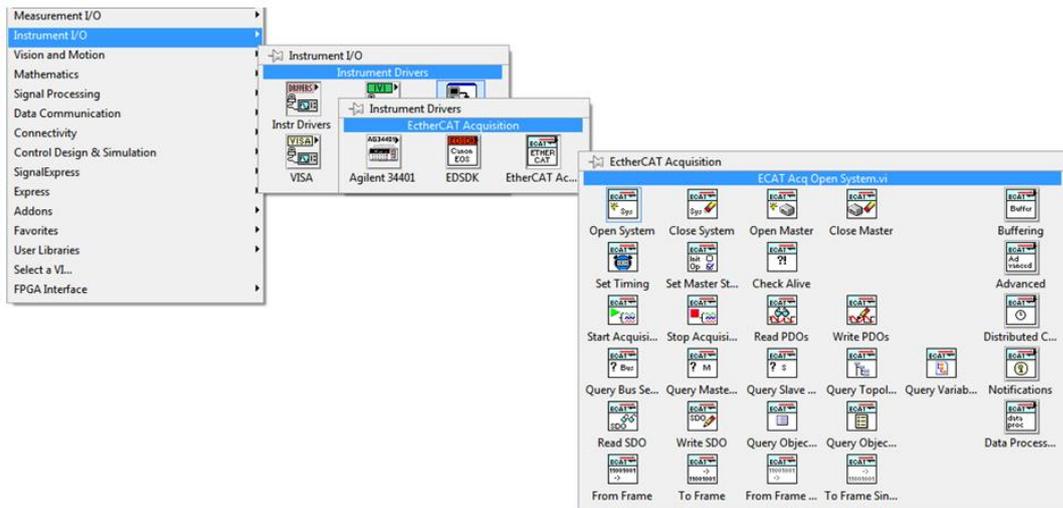


Figura 10: Bloques de la librería de EtherCAT para LabVIEW.

Por otro lado, también resultaba necesario instalar el controlador de tiempo real en la tarjeta de red alojada en el ordenador, lo cual solo es posible en el caso del sistema operativo Windows 10, a partir de la versión 20H2 [15]. Otro aspecto importante a considerar es que, la instalación de este controlador, limita las tarjetas de red posibles a únicamente las de los fabricantes Intel y Realtek.

Teniendo claras las anteriores consideraciones, se puede ejecutar el administrador del sistema, en el que se podrá ver una ventana como la que sigue.

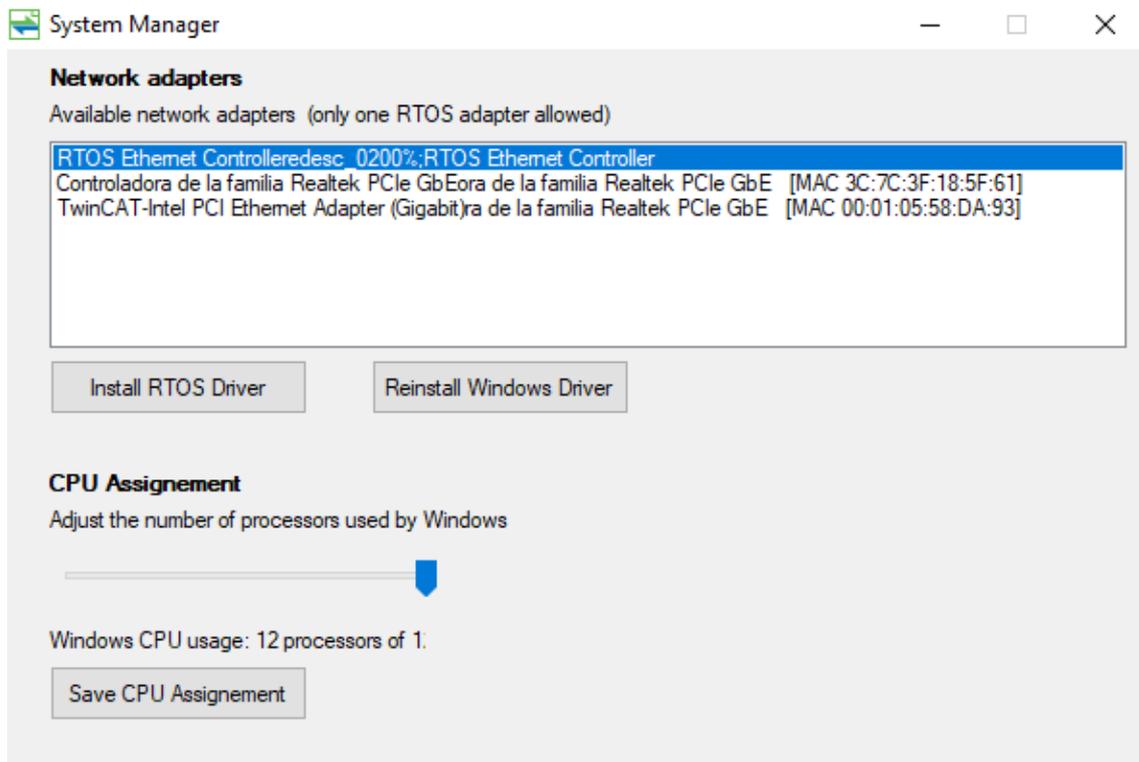


Figura 11: Instalador de controladores de la librería de EtherCAT para LabVIEW.

En este caso, se presenta en la imagen lo que se muestra después de hacer clic en la opción *Install RTOS Driver*, lo cual se puede saber por el término RTOS que aparece en el nombre del adaptador de red. Resulta conveniente mencionar que, si posteriormente se desea emplear el mismo puerto de la tarjeta de red para otras tareas, como por ejemplo en TwinCAT, se debe reinstalar el controlador de Windows con carácter previo.

Adicionalmente, para un correcto funcionamiento de la librería, en el manual se indica que se han de deshabilitar una serie de ajustes en la BIOS en el ordenador. En concreto, se hace referencia al soporte para USB heredado, la técnica de modulación de ensanchado de espectro, y las tecnologías C-STATE y *SpeedStep* de Intel. En el caso del equipo empleado, no todas estas características existían.

Con todo lo expresado en este apartado correctamente configurado, ya resulta posible probar los ejemplos que incluye la librería, o crear un nuevo programa incorporando los bloques añadidos.

2.3 EtherCAT

Tal y como se expuso al comienzo del documento, una de las razones fundamentales que justifican la realización del proyecto, es la imposibilidad que existía, con el equipo anterior, de transmitir los datos en tiempo real, a la velocidad suficiente como para que los resultados que se obtengan a partir de los ensayos puedan reflejar con fidelidad la interacción que se produce entre el pantógrafo y la catenaria en la realidad. Ante estas necesidades, el bus de campo EtherCAT [16] se presentaba como una alternativa viable.

El nombre EtherCAT hace referencia a *Ethernet for Control Automation Technology*, y define un protocolo de comunicación en tiempo real desarrollado en sus inicios por la empresa Beckhoff Automation. Viene recogido bajo el estándar IEC61158, y se planteó como un sistema especialmente indicado para procesos de muy corto periodo de ciclo, incluso inferiores a los 100 μ s, y con baja fluctuación para asegurar una sincronización precisa, con tiempos inferiores al microsegundo.

Atendiendo al principio de funcionamiento, el maestro EtherCAT envía un telegrama, el cual pasa por cada uno de los nodos de la topología de red. Cada esclavo lee la información cuya dirección está dirigida al mismo, y escribe los datos que quiere que vayan a algún otro nodo. El intercambio de información se hace mientras viaja el mensaje, por lo que los retardos que se producen en la comunicación son únicamente los producidos por las limitaciones del hardware que se esté empleando. Además, el maestro de la red usa una MAC (*Media Access Control*) estándar, por lo que se puede implementar el sistema en cualquier puerto Ethernet. Esta dirección MAC hace referencia a un identificador de 48 bits que permiten identificar de forma única a cualquier dispositivo conectado a una red.

La tecnología de EtherCAT permite realizar una comunicación bidireccional o dúplex completo (en adelante *full dúplex*), por lo que, si así se desea, al llegar el telegrama al último nodo, este puede regresar al maestro. En el caso del presente proyecto únicamente se dispondrá de un esclavo, el controlador del motor lineal, y el maestro de la red será un ordenador. Dada la característica indicada en este párrafo, únicamente será necesario un cable ethernet para

comunicar ambos equipos, y también implica que, desde el punto de vista teórico, sea posible alcanzar velocidades de transmisión de datos superiores a los 100 Mbits/s.

Toda la comunicación parte del maestro, ya que es el único habilitado para iniciarla. Esto ocurre ya que es la única manera de garantizar la característica de trabajo en tiempo real.

Entrando de forma más detallada en el protocolo de comunicación, el datagrama EtherCAT se incorpora dentro de un mensaje estándar de Ethernet (en adelante *frame*), el cual en su cabecera se identifica mediante un campo de dos octetos 0x88A4, que indica el protocolo que se está utilizando. Además, como se trata de optimizar el tiempo de ciclo, para que sea un protocolo válido para tasas de cambio de datos muy elevadas, los protocolos como TCP/IP (*Transmission Control Protocol*), o UDP/IP (*User Datagram Protocol*) se eliminan.

En la siguiente imagen se detalla con mayor grado de detalle los elementos del *frame* de comunicación. No obstante, no forma parte del proyecto el análisis a bajo nivel de la comunicación, ya que en esta parte del trabajo lo que será necesario realizar es un trabajo de integración de los bloques definidos en la librería de EtherCAT para LabVIEW.



Figura 12: Elementos de un datagrama EtherCAT.

Empleando EtherCAT, solo debe enviar un único telegrama con la información de salida al controlador MAC, y lo hace a través del DMA (*Direct Memory Access*). En el caso de que exista más de un esclavo en la red, es posible definir varios telegramas dentro del *frame* completo del mensaje. En la siguiente figura se puede ver una representación de esta idea. No obstante, hay que recordar que este no será el caso del diseño que se ha realizado, y que en el mismo solo existe un esclavo.

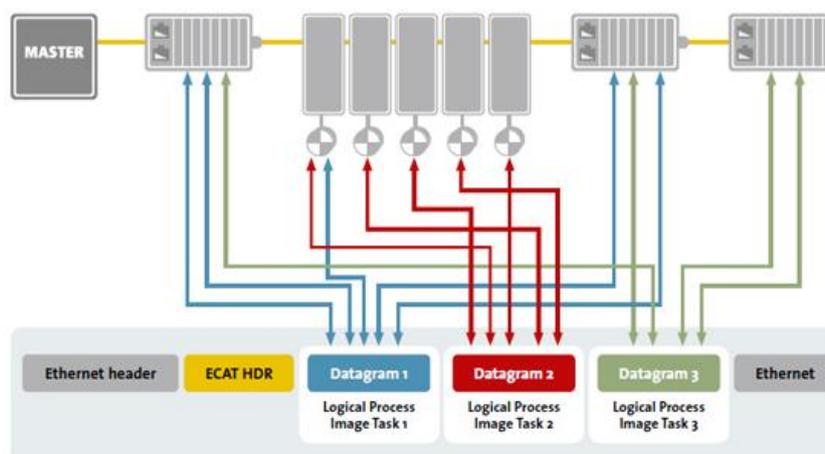


Figura 13: Envío de datagramas a diferentes esclavos de una red EtherCAT.

Como se ha mencionado, los únicos retardos que se producen bajo este bus de campo son los generados por el propio hardware empleado. Para ello, en cada nodo, los datos se deben

procesar “al vuelo”. Esto consiste en que, desde el momento en el que el maestro envía el mensaje completo, ya se incluye la dirección del nodo al que va dirigido cada datagrama, de modo que mientras este transita por los diferentes nodos, transmite la información necesaria, y junto a esto, recopila los datos que desean hacer llegar los esclavos al maestro. Esto se realiza sin detener la comunicación para realizar comprobaciones adicionales, y así posibilitar el trabajo en tiempo real del sistema.

Otra característica importante del bus de campo EtherCAT es que es flexible respecto a la topología de la red, permitiendo distintas configuraciones como línea, estrella, anillo o árbol, incluso pudiendo modificarla durante el funcionamiento.

2.4 Controlador E1450-EC-QN-0S

Al objeto de actuar sobre el movimiento que realiza el vástago del motor, resultaba necesario un controlador que recibiese los mensajes generados en el ordenador, los interpretase, y provocase el movimiento deseado del vástago. Adicionalmente, este mismo equipo, debía recopilar los datos de funcionamiento del motor, para poder monitorizarlos desde el ordenador de control.

El fabricante LinMot dispone de diferentes modelos, adecuados para el tipo de motor que se presentó en apartados anteriores. En este caso, una característica, que resultó fundamental a la hora de seleccionar el equipo más adecuado, es la característica de que pudiese transmitirse la información mediante el bus de campo EtherCAT. De esta forma, el modelo que mejor se ajustaba a los requerimientos era el E1450-EC-QN-0S [17], que es un controlador para motores lineales y rotativos, de 32 bits de resolución para la posición, y con la etapa de potencia trifásica, con la que posteriormente se alimenta el actuador, integrada.



Figura 14: Controlador EtherCAT.

El equipo también dispone de interfaces de procesamiento de señales de gran velocidad, como entradas analógicas y digitales, e incluso una entrada tipo disparo (en adelante *trigger*), lo cual

permite la posibilidad de emplear distintos tipos de control del movimiento, como por ejemplo a través de la magnitud de una señal de corriente.

En el caso de este proyecto, las órdenes de control se transmitirán a través de un cable ethernet, pero todas las interfaces mencionadas permiten que el controlador pueda operar de muchos modos diferentes.

Movimientos interpolados: En este modo, se alcanza la posición deseada en un comando único, y el desplazamiento se ajusta para no sobrepasar los límites máximos de velocidad y aceleración. El rango de movimiento es de ± 100 m, con unas resoluciones de posición de $0.1 \mu\text{m}$, de velocidad de $1 \mu\text{m/s}$, y de aceleración de $10 \mu\text{m/s}^2$, gracias a los 32 bits de los que se dispone.

Curvas a lo largo del tiempo: En el controlador se pueden almacenar hasta 100 curvas diferentes, las cuales describen la evolución de la posición del vástago a lo largo del tiempo. Estas curvas pueden disponer de hasta 16000 puntos, y se pueden activar mediante la comunicación serial, por cable Ethernet, o con la entrada de *trigger* por ejemplo. De nuevo, el rango de movimiento es de ± 100 m, con una resolución de posición de $0.1 \mu\text{m}$.

Movimientos ajustados: Se puede describir como una combinación de los dos modos anteriores ya que se realiza un único movimiento, a una posición determinada, pero la forma de alcanzar el punto de destino viene definida por una curva indicada por el usuario en un archivo CSV de Microsoft Excel®.

Transmisión del punto de consigna: Es un modo que permite seguir una curva a través del envío constante de los puntos a alcanzar de forma cíclica y temporizada. Es este el modo empleado cuando en este documento se haga referencia a los modos de funcionamiento *P Stream* y *PV Stream*, y que corresponde al control por parte del usuario de la posición (letra P) y la velocidad (letra V).

Pasos simples: Se pueden almacenar en el controlador hasta 8 diferentes posiciones o comandos de desplazamiento (una curva), para poder reproducirlos por el vástago cuando se envió la señal de control.

Tabla de comandos: Se define una secuencia de operaciones de hasta 255 comando individuales, los cuales se almacenan en una tabla, y se van ejecutando en orden. Este es el modo más adecuado para realizar movimientos que se pueden secuenciar, y que se necesitan realizar a muy alta velocidad. El tiempo de ciclo en este modo es de $100 \mu\text{s}$.

Sincronización con un *encoder* del maestro: Se puede describir una curva previamente definida, cuyos puntos se irán siguiendo por el vástago según vaya girando el *encoder* que controla el sistema. Una vuelta completa del *encoder* se corresponde con un ciclo completo de la curva a seguir.

Sincronización con una correa: En este caso, lo que se sincroniza es la velocidad del movimiento del vástago, según la velocidad a la que avanza una correa. Este modo resulta adecuado para aplicaciones como el llenado sincronizado de envases, etiquetado de botellas en una cinta transportadora y carga o descarga sincronizada entre otras aplicaciones.

2.5 Medida de la posición del vástago

El control en lazo cerrado de la posición del vástago, requiere de un sistema que permita conocer en cada momento el desplazamiento que se ha realizado, con el propósito de saber si se ha alcanzado la posición de consigna, o aún se debe seguir con el movimiento. Para ello, el motor del fabricante LinMot cuenta con un *encoder* interno.

Sin embargo, para la aplicación final, y tal como se presentará en el tercer capítulo resultaba necesario incorporar un dispositivo externo de medida de esta variable, de altas prestaciones, para cumplir con el objetivo con los requisitos que requiere el estudio que se está realizando sobre el banco de ensayos principal. En dicho capítulo se analizará la viabilidad de instalar modelos que trabajan con tecnologías muy diferentes, las cuáles se describirán brevemente en las siguientes líneas.

Un *encoder* lineal es un dispositivo que puede transmitir la información de desplazamientos, sobre un eje fijo, de forma absoluta (respecto a una referencia), o modo incremental (cada desplazamiento parte de una referencia móvil). Este segundo modo de funcionamiento es el que se empleará en la integración final del proyecto.

Existen diferentes tecnologías que permiten de forma indirecta conocer el desplazamiento que se ha efectuado.

Los *encoder* ópticos son los más habituales, y generan impulsos eléctricos al desplazarse debido a que, en su movimiento, un fotodiodo identifica tramos de luz y de sombra, los cuales, al ser conocido su tamaño, permiten saber, según el número de tramos de luz identificados, el desplazamiento que se ha realizado.

No obstante, el dispositivo que se ha seleccionado para el proyecto, por los motivos que se presentarán en el apartado 3.4, es un *encoder* magnético, que recibe este nombre porque se desplaza sobre una banda magnética, y cuyo principio fundamental de funcionamiento es el mismo que en el caso anterior. La única modificación es que, en lugar de detectar zonas de luces y zonas de sombras, la polaridad de la banda magnética va cambiando, con lo que el sensor pasa por zona de campo magnético positivo, y zonas de campo magnético negativo. Para que esto pueda ser detectado, el disco interno del *encoder* debe estar magnetizado, conteniendo varios polos, por lo que este disco comienza a rotar. De esta forma, y a través de un circuito electrónico, se transforman los pulsos que se reciben del sensor, transformándolos en la información que se estaba buscando ya que, el dato de a qué distancia equivale un pulso es conocida [18].



Figura 15: Encoder magnético.

Por otro lado, y destinado a otro tipo de aplicaciones, aunque resultaba factible para el proyecto, dadas los restrictivos requerimientos con los que se contaba, se encuentran los sensores láser. En estos equipos, la cabeza sensora emite un haz de luz, el láser, y este rebota en el objetivo, el cual, en el caso del presente proyecto, podría haber sido una placa metálica que se moviese solidaria al vástago, mientras en sensor permaneciese en una posición fija. En este caso, la medida directa que se realiza es el tiempo que tarda el haz de luz en regresar al receptor incorporado en el sensor, tras rebotar en el objetivo. Conociendo la velocidad de la luz, se puede despejar fácilmente la distancia, teniendo en cuenta que existe una distancia adicional condicionada porque el receptor de la luz no puede estar justo encima del emisor. No obstante esto se puede corregir mediante el cálculo con un modelo matemático que tenga en cuenta esto último.

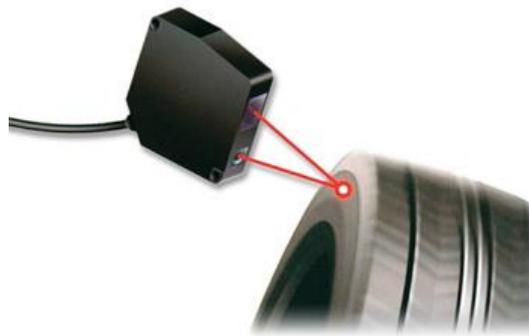


Figura 16: Sensor de desplazamiento láser.

Por último, la otra tecnología que se estudió como posibilidad en el proyecto, es la de la vibrometría, la cual presenta unas prestaciones en cuanto a resolución en posición y velocidad muy favorables para multitud de aplicaciones entre las que se encuentra este proyecto. Además, se trata de una medida lineal en todo el rango de actuación.

El principio de funcionamiento se basa en la medida de vibraciones a través del efecto Doppler. Profundizando un poco más, la frecuencia de una onda se ve alterada al rebotar en una superficie en movimiento, la cual se detecta en su vuelta para detectar la variación que se ha producido. En los equipos de Polytec [19] que se analizarán en el proyecto, la onda enviada se divide mediante espejos en dos partes, una de las cuales se lleva al receptor directamente para conocer la señal de referencia, mientras que la segunda señal se hace colisionar con el cuerpo en movimiento, el cual podría ser, del mismo modo que en el caso láser, una placa que se moviese solidaria al extremo del vástago. Según la velocidad y el desplazamiento que experimente la pieza sobre la que incidimos la onda, la frecuencia se ve modificada de una forma determinada. Esta nueva onda se dirige al mismo detector al que se hizo llegar a la señal original, de modo que se produce una modulación por la superposición de las ondas. Después de esto, solo resulta necesario un procesamiento de la señal para conocer la posición y la velocidad del objeto estudiado.

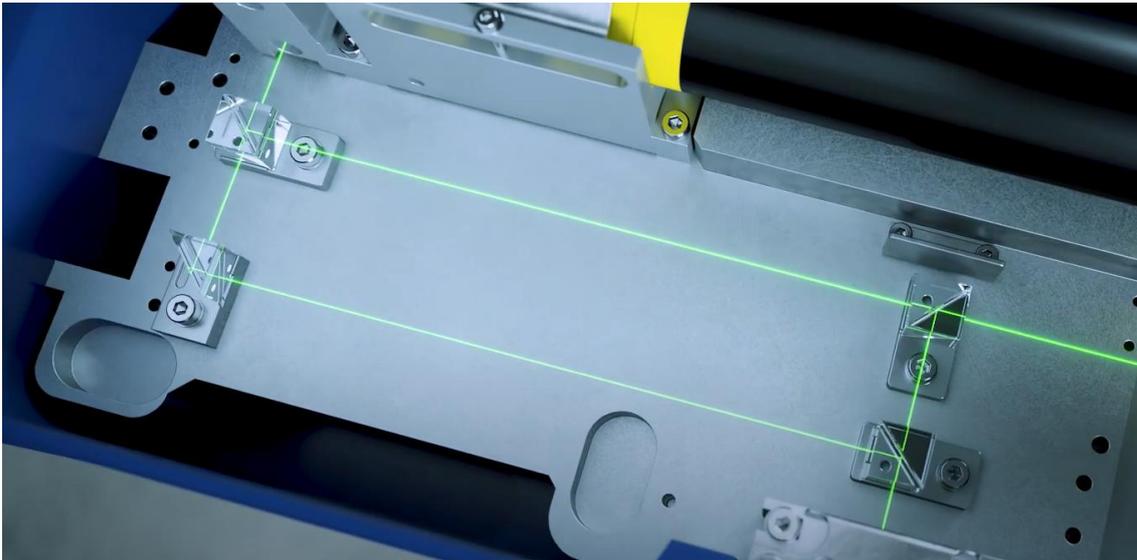


Figura 17: Direccionamiento del haz de luz en vibrómetros de Polytec.

2.6 SolidWorks

En la primera fase del presente proyecto, se diseñó un banco de ensayos en el que poder ensayar el motor lineal de forma segura, y que además se pudiese emplear en las pruebas de caracterización de dicho equipo.

SolidWorks [20] ha sido el software de modelización CAD 3D que se ha empleado para diseñar las piezas que componen el banco de ensayos, tanto para los elementos del montaje principal, como para las piezas adicionales destinadas al estudio del movimiento mediante un encoder.

En este caso, solo ha sido necesario emplear las capacidades de creación de sólidos del software, y para ello se ha empleado una licencia para estudiantes proporcionada por la Universidad Politécnica de Valencia, ya que esto ha sido lo necesario para el diseño del modelo. Estos últimos han tenido una gran relevancia, especialmente en el caso del diseño del montaje anexo para el encoder MS01-1/D-SSI, ya que este requiere de una gran precisión dimensional en cuanto a su posición respecto a la banda magnética sobre la que se desliza. Por esta razón se necesitaba emplear un software de diseño por ordenador.

Otras soluciones que incluye SolidWorks, es la posibilidad de crear ensamblajes, en los que además de visualizar el conjunto de piezas que componen el diseño, se puede analizar el movimiento de las piezas móviles, pudiendo de este modo identificar posibles defectos antes de fabricar las piezas.

En la siguiente figura se puede ver la interfaz del programa.

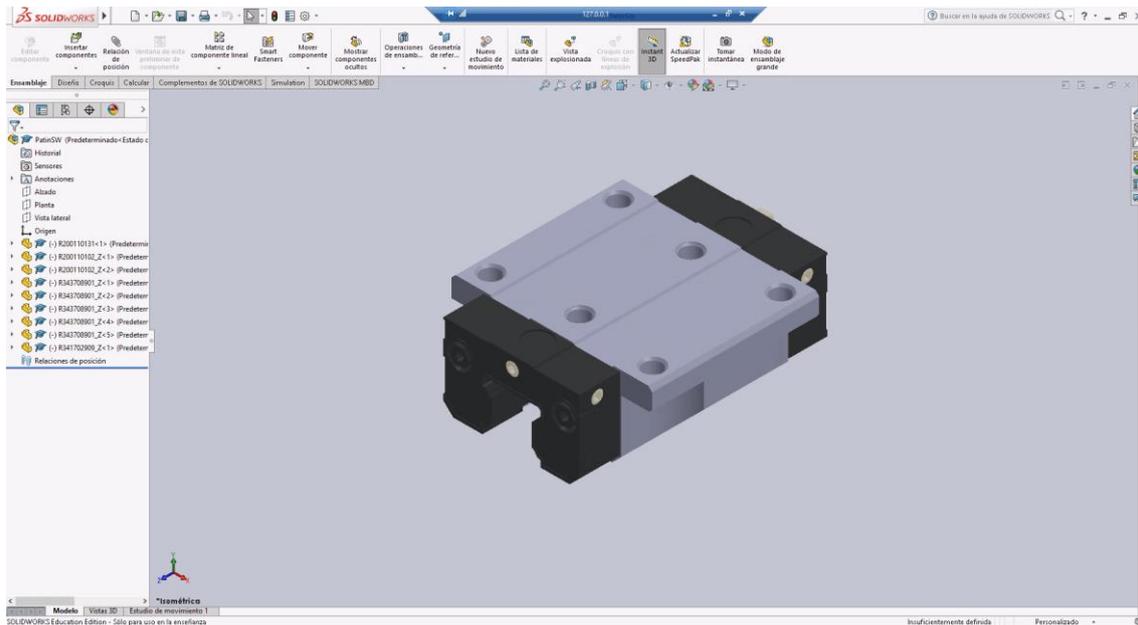


Figura 18: Interfaz del SolidWorks.

Como en cualquier otro software de modelado 3D, la definición de los sólidos parte del dibujo de un croquis en 2D de la base, y a partir de ella se realizan operaciones de formación de sólidos, como la extrusión, la revolución o chaflanes, que permiten obtener el objeto 3D deseado. En este caso, los carritos, sus guías, el motor lineal, y su carcasa, no ha sido necesario diseñarlos ya que los fabricantes ya proporcionan los modelos 3D en sus páginas webs. De este modo, en este trabajo, se han diseñado las piezas adicionales de apoyo, y se han realizado los ensamblajes necesarios para visualizar si el sistema planteado resultaba adecuado antes de realizar la solicitud de fabricación.

2.7 LinMot-Talk

Consiste en un software ofrecido por el fabricante LinMot [21] para la configuración de los controladores de su marca. Resulta especialmente conveniente utilizar este software ya que los equipos vienen recogidos en la base de datos del mismo, por lo que el reconocimiento del controlador, y su configuración resulta especialmente sencilla.

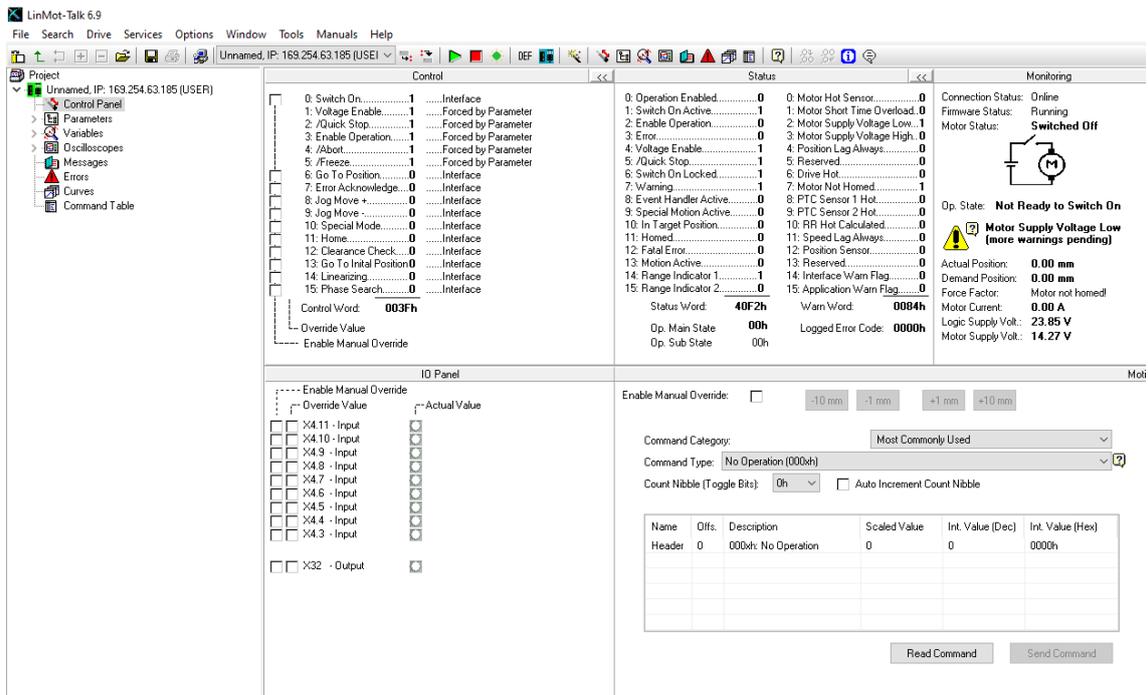


Figura 19: Ventana principal del software Linmot-Talk.

En la imagen anterior, se puede ver el panel frontal del LinMot-Talk. En el menú izquierdo se puede acceder a los parámetros de configuración del controlador, algunos de los cuales se comentarán más adelante. En el cuadro de control se permite al usuario realizar un control manual de algunas de las posibilidades de control que ofrece el panel, como poner el motor en modo de operación, ir a la posición de referencia, o anular una señal de error. También, en el caso de que el sistema esté siendo controlado por otro agente, como ocurre en este proyecto al emplear LabVIEW, este panel permite monitorizar el estado de estas variables.

El panel de entradas y salidas digitales, permite simular la activación de las entradas del puerto X4, lo cual podría utilizar como señal de *trigger*, para lanzar una orden de seguimiento de una curva de movimiento, o activar algún actuador entre otras muchas posibilidades.

El panel de estado indica el valor de los bits de estado, los cuales se corresponden con la información que se puede extraer del PDO de entrada. En el caso de este trabajo, ha resultado de gran relevancia el poder acceder al dato contenido en el bit denominado “*homed*”, ya que, en el software, se bloquea la entrada en el modo de funcionamiento hasta que el motor alcanza la posición de referencia como método de seguridad.

El cuadro de monitorización muestra el estado del motor, así como advertencias, e información en tiempo real de la posición actual y demandada del motor, la corriente consumida, y la suministrada.

Finalmente, el área ubicada en la zona inferior derecha permite el envío de órdenes al controlador de forma manual. Esta herramienta resultó de gran utilidad en este proyecto, tanto para comprobar el funcionamiento de los PDO de salida, como para poder realizar ensayos seguros antes de proceder con las pruebas desde el software implementado en LabVIEW.

Entrando en el detalle de la forma de configurar los parámetros básicos del funcionamiento del motor, esto se puede realizar, una vez se ha establecido la conexión al equipo, mediante una sucesión de ventanas denominada “Motor wizard”.

En la primera ventana, se selecciona el modelo de motor y de vástago utilizado, de esta forma el software ya puede acceder a los datos característicos de los equipos.

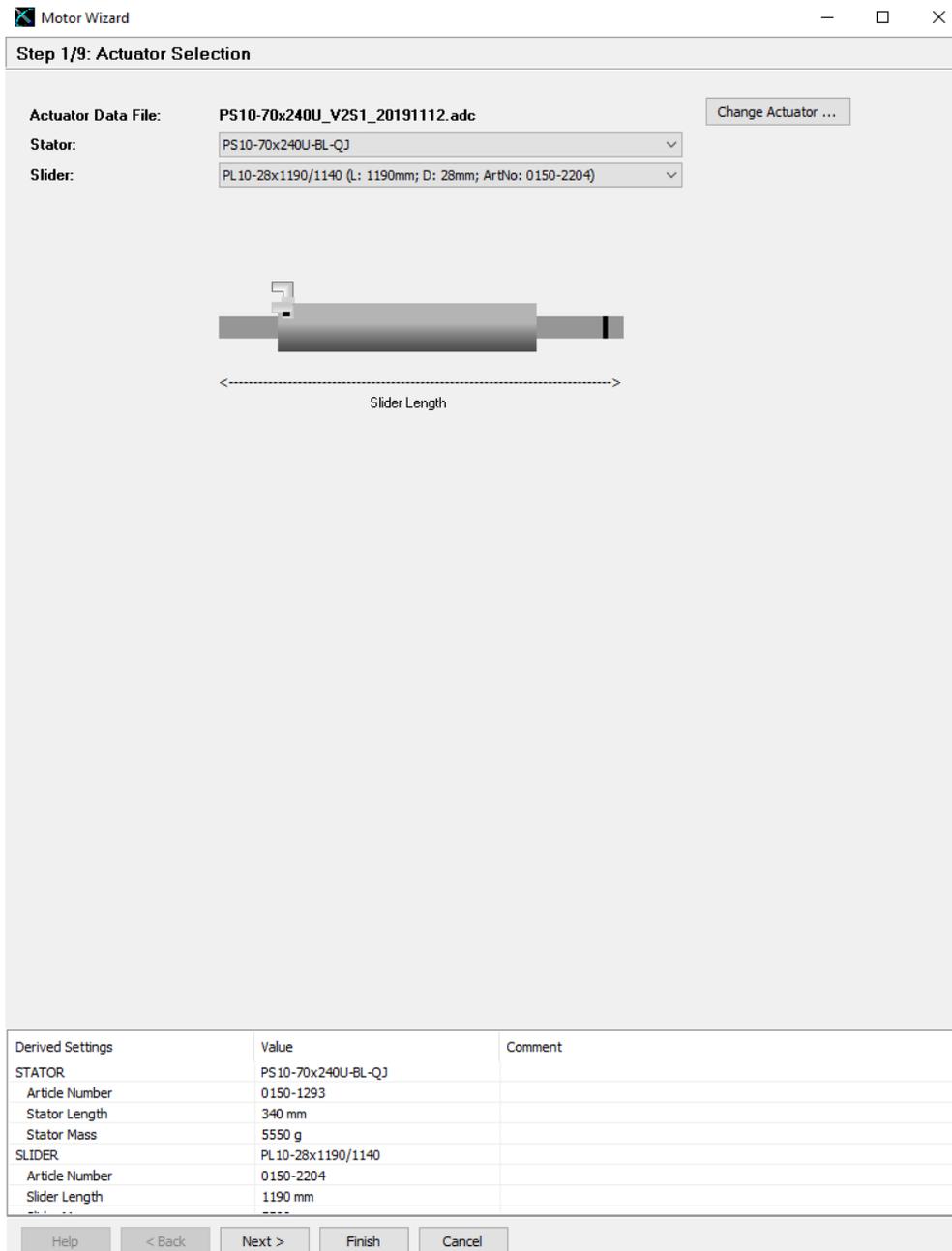


Figura 20: Ventana 1 del motor wizard en LinMot-Talk.

En la segunda ventana, únicamente se permite la introducción de un nombre para el controlador que se está configurando, y que se indique la tipología de alimentación que se está utilizando.

Motor Wizard

Step 2/9: Drive Settings

Drive Name:

Motor Supply:
Voltage:

Derived Settings	Value	Comment
MOTOR POWER SUPPLY VOLTAGE	3/PE AC 400V (±10%) / 50-60Hz / T...	
Motor Supply Voltage Low	420 V	
Motor Supply Voltage Too low	400 V	

Figura 21: Ventana 2 del motor wizard en LinMot-Talk.

En la siguiente etapa de la configuración, se puede indicar si se ha incorporado alguna extensión de los cables, lo cual podría tener alguna relevancia sobre el funcionamiento desde el plano eléctrico.

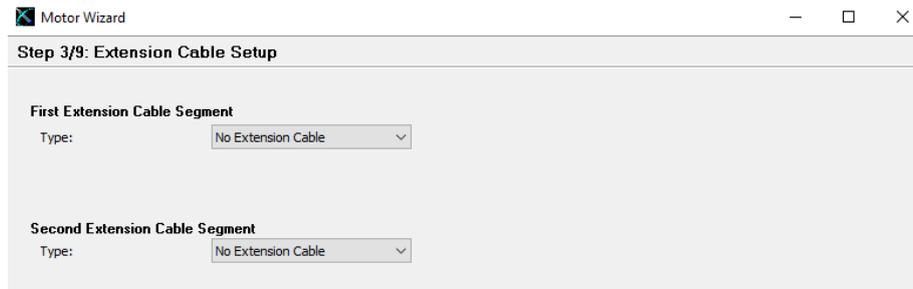


Figura 22: Ventana 3 del motor wizard en LinMot-Talk.

En la cuarta ventana se puede seleccionar el modelo de sensor de posición externo que se está usando, si es que este se ha incorporado. En el caso de este proyecto, se podría utilizar el sensor MS01-1/D-SSI al que ya se ha hecho referencia previamente, pero el propósito de dicho sensor no es mejorar las capacidades del encoder interno del motor, sino servir como fuente de información para una adquisición de datos realizada desde el equipo CompactRIO.

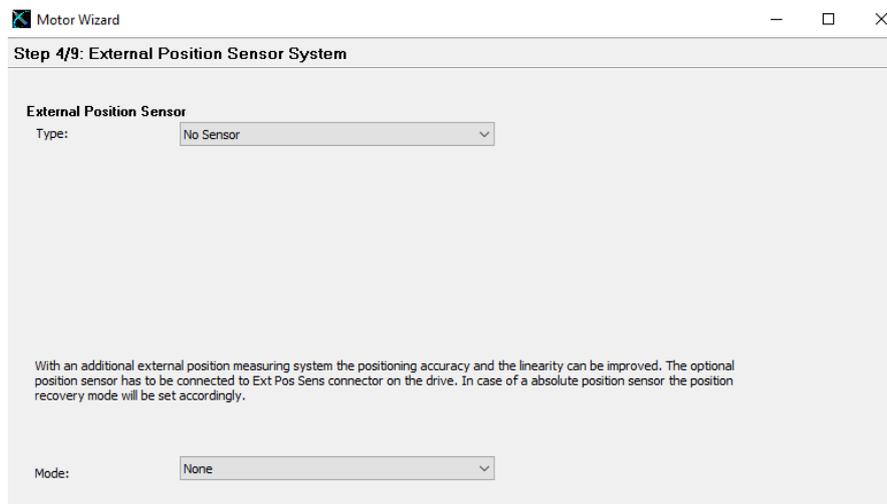


Figura 23: Ventana 4 del motor wizard en LinMot-Talk.

La pestaña que aparece en quinto lugar ofrece la posibilidad de indicar si el motor se va a utilizar para un desplazamiento vertical, como ocurre en el equipo instalado en el montaje principal, o si, por el contrario, trabajará en horizontal, como ocurre en el sistema diseñado. También se puede indicar la masa del cuerpo que tiene que arrastrar el vástago para que esta sea tenida en cuenta en los cálculos, y se presentan casillas para introducir datos de fuerzas externas como las de rozamiento.

Motor Wizard

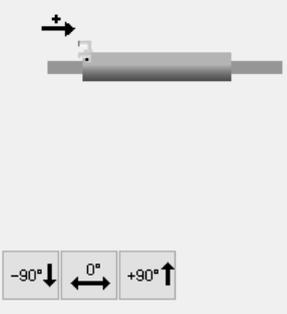
Step 5/9: Feed Forward Parameters

Mechanical Layout
 Moving Part of Motor: Slider
 Orientation Angle (-90°..+90°): 0 °

Moving Mass
 Slider: 5590 g
 Additional Load Mass: 0 g

Friction Forces
 Dry Friction: 0 N
 Viscous Friction: 0 N/(m/s)

MagSpring (or other constant force)
 External Constant Force: 0 N
 Force Direction: Negative



Derived Settings	Value	Comment
Total Moving Mass	5590 g	
Gravitation force in motor direction	0 N	
External Constant Force	0 N	
Sum of Constant Effective Forces	0 N	
FF Constant Force	0 A	Current for constant force compensation
FF Friction	0 A	Current for compensation of dry friction
FF Damping	0 A/(m/s)	Factor for compensation of viscous friction
FF Acceleration	0.095 A/(m/s^2)	Acceleration feed forward factor

Help < Back Next > Finish Cancel

Figura 24: Ventana 5 del motor wizard en LinMot-Talk.

La sexta ventana está destinada al ajuste de los parámetros del PID del controlador, así como a la determinación de la zona muerta, la cual hace referencia al límite a partir del cual, una orden de desplazamiento es tenida en cuenta, y por tanto reproducida por el vástago. Por defecto, los valores del PID son los que el software cataloga como funcionamiento rígido, y por las pruebas realizadas, se ajusta correctamente a las necesidades de funcionamiento de este proyecto.

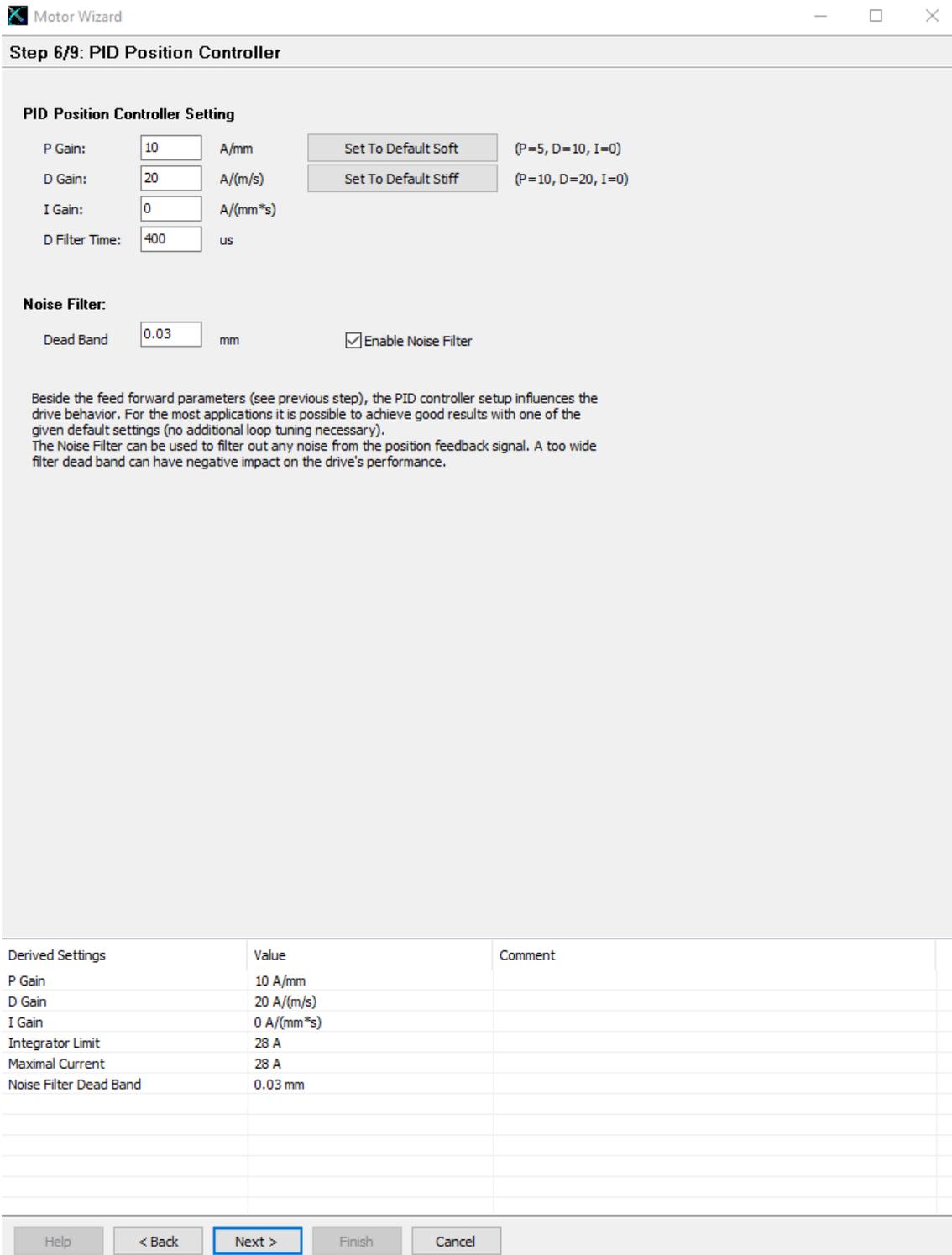


Figura 25: Ventana 6 del motor wizard en LinMot-Talk.

En la séptima ventana se puede indicar la velocidad a la que se desplaza el vástago al iniciar el procedimiento de búsqueda de la posición de *home*, así como la metodología para obtener una posición de referencia. En este caso, se ha seleccionado que el motor busque un tope físico desplazándose hacia la derecha. En ese momento, sabrá que está en el límite en ese sentido de desplazamiento. Se escogió la derecha en lugar de la izquierda ya que, en este segundo caso, por la configuración de los topes metálicos, podría no respectarse el margen mínimo de 15 cm

que debe sobresalir el vástago por el extremo opuesto del motor. Hacia la derecha, este margen asciende a 25 cm por lo que se puede cumplir holgadamente.

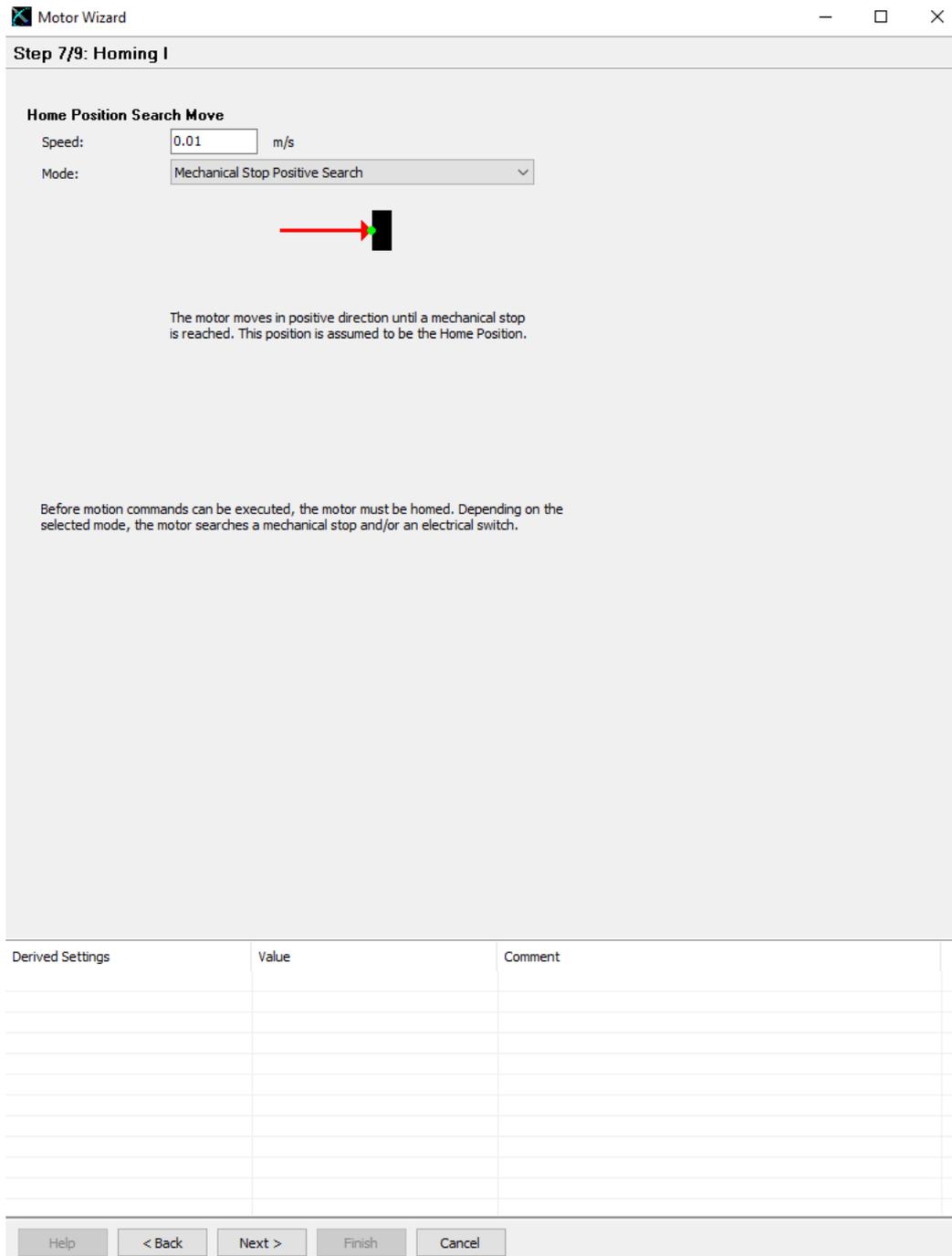


Figura 26: Ventana 7 del motor wizard en LinMot-Talk.

En la siguiente pantalla se debe establecer la posición del vástago cuando este encuentra el tope mecánico. Cuando se selecciona la distancia A, el software automáticamente completa la información de la distancia B, o viceversa, ya que conoce la longitud del vástago empleado.

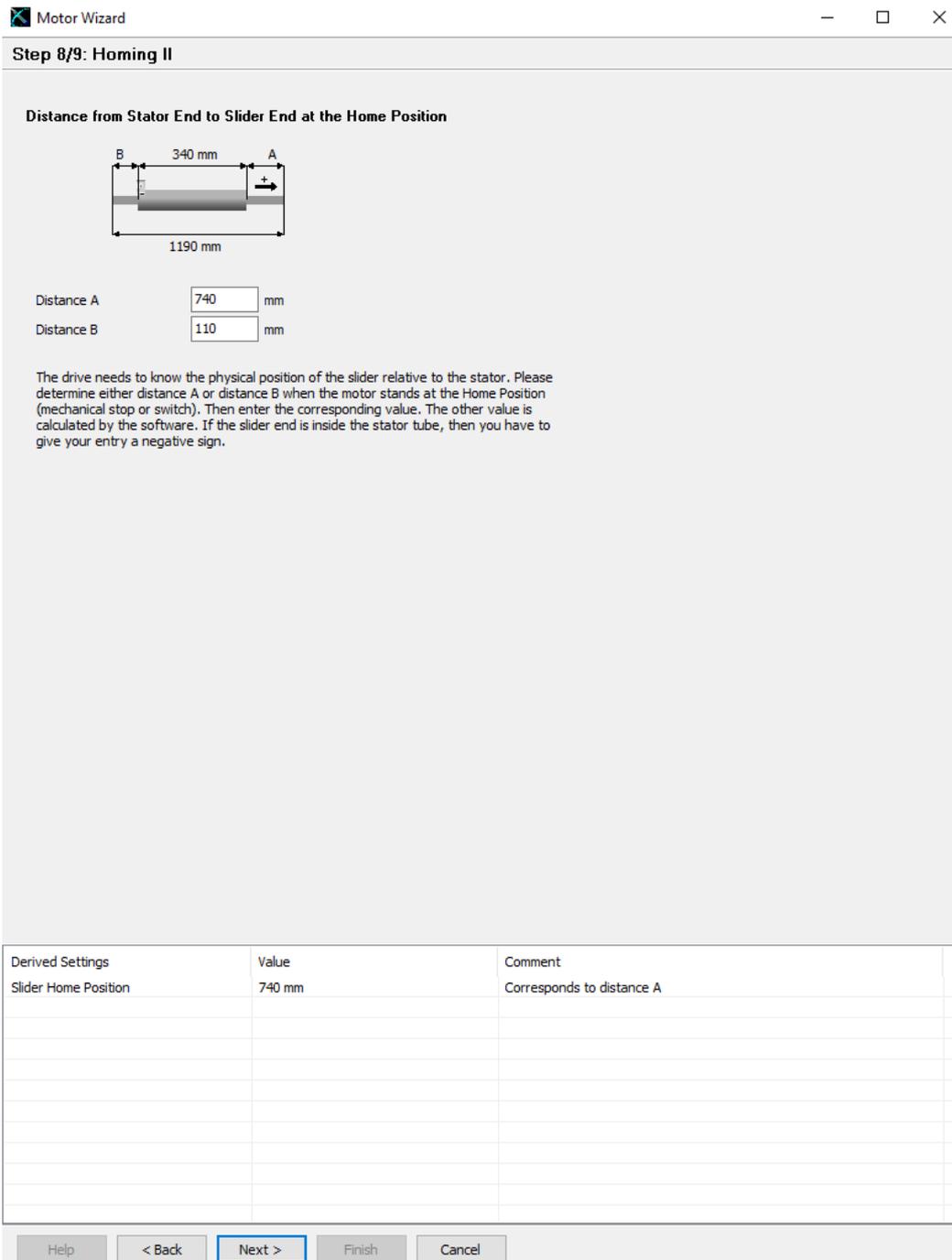


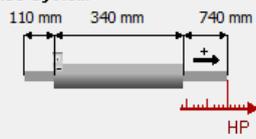
Figura 27: Ventana 8 del motor wizard en LinMot-Talk.

En la última ventana, se solicita qué distancia se desea recorrer tras encontrar el tope mecánico, para establecer en ese punto la posición del origen de coordenadas (en adelante *home*). En este caso, como se pretende que esta referencia quede en el centro de la guía del carrito, se debe retroceder hacia la izquierda 320 mm.

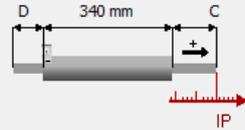
Esto último, deja un rango de posible desplazamiento de entre -405 y 405 mm. No obstante, se limita el rango de movimiento para evitar colisiones, dejando un margen de seguridad, y por ello como límites se establece el rango de -350 a 350 mm.

Step 9/9: Homing III

Definition of the Application Reference System

Home Position (HP): mm

Move to the Initial Position at the End of the Homing Procedure

Initial Position (IP): mm

You can define your application specific reference system by assigning any position value to the Home Position. All further position values are based on this system. At the end of the homing procedure the motor moves to the Initial Position. Then it is ready to execute the motion commands. If the motor has to be homed on a mechanical stop, then the Initial Position value should differ from the Home Position.

Minimal Position Error Enabled Minimal Position: mm (-405mm <= Position <= 405mm)

Maximal Position Error Enabled Maximal Position: mm

Derived Settings	Value	Comment
Minimal Position (stroke range limit)	-350 mm	
Maximal Position (stroke range limit)	350 mm	
Distance C	420 mm	
Distance D	430 mm	

Help < Back Next > Finish Cancel

Figura 28: Ventana 9 del motor wizard en LinMot-Talk.

Capítulo 3. Realización práctica

3.1 Elementos del banco de ensayos

Con motivo del riesgo derivado de las elevadas fuerzas y velocidades que es capaz de alcanzar el motor lineal descrito en el capítulo anterior, se justifica la necesidad de diseñar un banco de ensayos en el que se pudiesen realizar las primeras pruebas de funcionamiento de este. En este mismo sentido de las ideas, resultaba recomendable no incorporar el sistema diseñado en este Trabajo de Fin de Máster al montaje principal hasta que este no estuviese completamente controlado, reduciendo de este modo la posibilidad de que se produzca un accidente en el que algún elemento del banco de ensayos existente resultase dañado.

El banco de ensayos debía incorporar todos los equipos principales que resultaban necesarios para el funcionamiento del sistema, los cuales se describen en las siguientes líneas.

Como se mencionó en el capítulo anterior del documento, el motor empleado ha sido el PS10-70x240U-BL-QJ del fabricante LinMot, el cual permite que un vástago de 1190 mm de longitud y 28 mm de diámetro, en concreto el modelo PL10-28x1190/1140 del mismo fabricante, pueda realizar el movimiento lineal oscilatorio que se desea.



Figura 29: Motor PS10-70x240U-BL-QJ.



Figura 30: Vástago PL10-28x1190/1140.

El motor, en su funcionamiento, se encuentra dentro de una carcasa prismática, que facilita su apoyo sobre una superficie plana, lo protege, y además es un elemento que mejora la disipación de calor durante el funcionamiento. El modelo concreto que se ha utilizado es el PF10-70x270 de LinMot. El fabricante, adicionalmente, aporta soluciones que permiten disipar mayor cantidad de calor, incluyendo por ejemplo ventiladores en algunos casos. No obstante, esto no se ha considerado necesario en este proyecto, ya que el motor va a operar bastante alejado de las condiciones límite de uso en su funcionamiento.



Figura 31: Carcasa de motor lineal PF10-70x270.

A los elementos anteriores, los acompañan una serie de piezas, que no forman parte de los equipos que se debían utilizar, pero que se debían de añadir para que el banco de ensayos resultase útil como un sistema de prueba previo al uso final del motor.

El extremo del vástago cuenta con una rosca que permite atornillar el mismo a un carrito, que será el que describa un movimiento oscilatorio lineal, deslizándose sobre un rail metálico, correctamente lubricado.



Figura 32: Extremo del vástago empleado.

Concretando con una mayor profundidad el párrafo anterior, se ha empleado un patín modelo HGR15, del fabricante HIWIN, el cual viene acompañado de una guía, que se ajusta perfectamente a las dimensiones del mismo, con una longitud de 90 cm para el desplazamiento lineal del carrito.

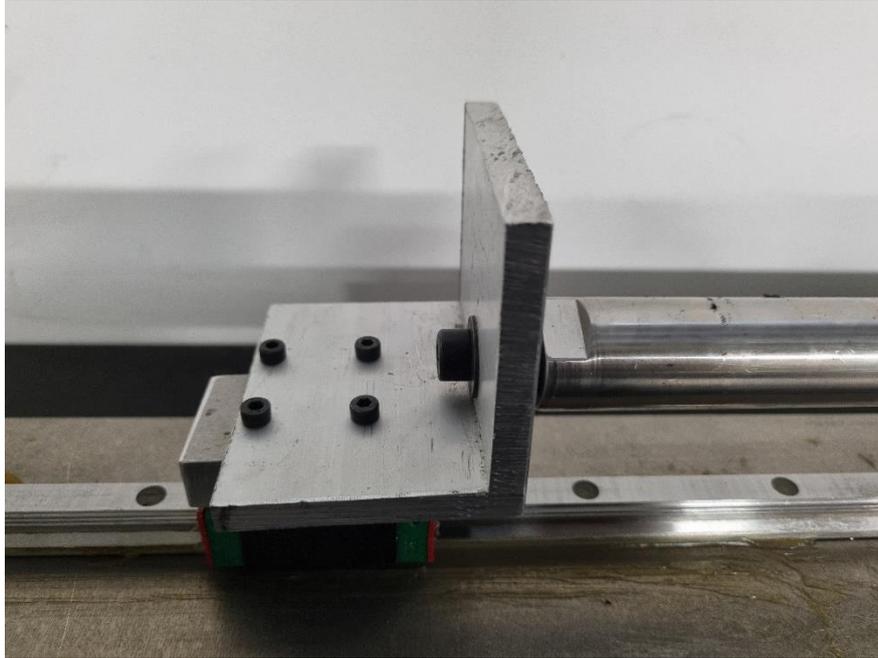


Figura 33: Perfil en L que transmite el movimiento del vástago al carrito.

Colocando todos los elementos enunciados de forma correcta, sobre una superficie plana, ya se cuenta con lo necesario para poder experimentar con el motor, y realizar las pruebas que resulten oportunas para su caracterización.

No obstante, en otro sentido de las ideas, parecía razonable incorporar al montaje elementos de seguridad para la protección de las personas durante el uso del banco de ensayos. Estas partes adicionales debían cumplir con la condición de no interferir en el correcto funcionamiento del sistema, lo cual se analizará con mayor grado de detalle en el siguiente apartado del documento.

En el lado del motor en el que se realiza la conexión al carrito, se colocaron unas piezas metálicas que realizan la función de topes para tratar de impedir que, en el caso de que fallase el sistema, o se cometiese un error en el control, no saliese disparado el vástago en ninguna de las dos direcciones en el sentido de movimiento del sistema. Estos topes metálicos se atornillan al rail sobre el que se traslada el carrito.

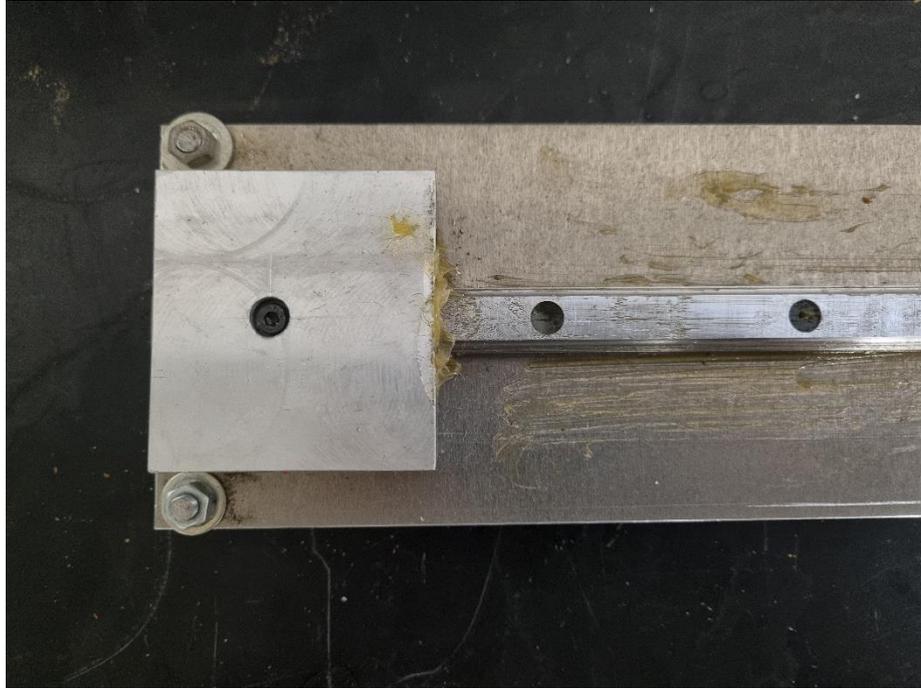


Figura 34: Topes metálicos de seguridad.

Mientras tanto, en el otro lado del motor, se encuentra uno de los extremos del vástago moviéndose en concordancia con lo que ocurre en el otro lado del motor, lo cual resulta peligroso atendiendo a las velocidades y fuerzas que es capaz de desarrollar el motor empleado. Por ello, se ha colocado un tubo de sección cuadrada, de aluminio para evitar la interacción con los imanes del vástago, de modo que el vástago se mueva en el interior del mismo, sin que exista la posibilidad de que entre en contacto con ningún elemento externo al banco de ensayos.



Figura 35: Tubo de protección.

Como se puede apreciar en la última fotografía, la estructura citada, incluye unas pequeñas placas metálicas elevadas gracias a la incorporación de unos tornillos, que, colocados en posición vertical, permiten regular esta parte del montaje. Esto último resulta importante ya que, no solo debe pasar el vástago por el interior del tubo sin colisionar con las paredes, sino que, adelantando un aspecto que se presentará con mayor grado de detalle más adelante en el documento, el vástago no puede pasar a menos de un centímetro de ninguna superficie metálica. Esta distancia se incrementa a dos centímetros en el caso de que dicha superficie fuese ferromagnética.

Adicionalmente, para evitar que alguien moviese las manos por los alrededores de la zona no protegida por el tubo magnético, se instaló una malla protectora que separa la zona de ensayo. Esta rejilla se mantiene vertical gracias a la incorporación de una perfilera metálica, y solo se retira cuando, estando la estación detenida, se quiere realizar alguna modificación mecánica.



Figura 36: Malla protectora.

En cuanto a la alimentación de los equipos, la alimentación trifásica se pasa a través de un filtro, ya que en las condiciones en las que se trabaja, los desfases entre las fases de la tensión podrían afectar al correcto funcionamiento del motor lineal, y por otro lado se cuenta con una fuente de tensión continua con la que se alimenta también al controlador. En ambos casos, se incorporan interruptores magnetotérmicos de protección en las líneas. Todo este conjunto se ha organizado dentro de un mismo cuadro, a excepción del filtro, el cual, debido a su tamaño, tuvo que ser puesto a parte.



Figura 37: Cuadro eléctrico del sistema.

3.2 Diseño del banco de ensayos

En los próximos párrafos, se describirá el montaje que se ha realizado para poder ensayar los elementos expuestos en el apartado anterior, así como los elementos auxiliares que han resultado necesarios para completar un banco de ensayos funcional con el que poder caracterizar el motor lineal PS10-70x240U-BL-QJ-D01, y también realizar las primeras pruebas de comunicación con el mismo.

En primer lugar, un requisito importante al que se debía atender fue el que, dado que el sistema iba a trabajar en algunos momentos describiendo movimientos oscilatorios, y en otras ocasiones movimientos alternativos bruscos, resultaba necesario que todas las piezas estuviesen correctamente ancladas a una superficie común, de modo que las vibraciones que se produjesen no afectasen al comportamiento del sistema, y por tanto pudiesen alterar las mediciones que se pretendían realizar posteriormente a partir del montaje diseñado.

En concordancia con lo anterior, el banco de ensayos parte de una pieza metálica rectangular, **cuyas medidas son**, un largo de 152.5 cm, un ancho de 10.1 cm, y un espesor de 2.5 cm. Lo cual supone un elemento suficientemente robusto para la aplicación en la que se va a emplear.

Sobre esta placa metálica, y en el eje de simetría longitudinal de la misma, se dispuso el rail sobre el que circulará el carrito. La unión de los elementos se realizó mediante tornillos de métrica 4, directamente atornillados a la base, sobre la que se realizaron los taladros, y posteriores roscados necesarios, con carácter previo.

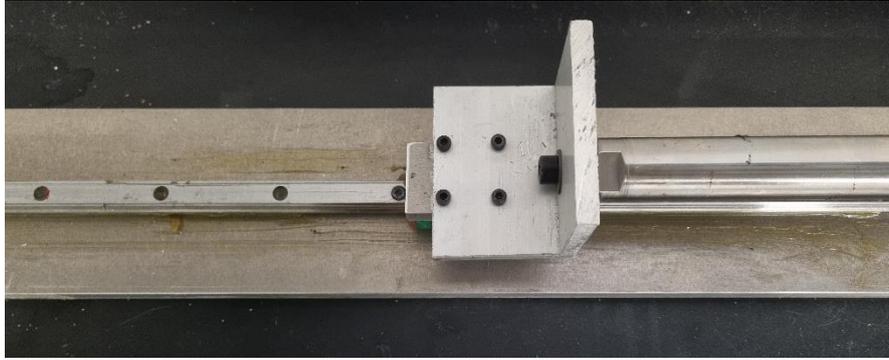


Figura 38: Carrito sobre el rail atornillado a la base del sistema.

En este punto ya se puede introducir uno de los elementos de seguridad que se han dispuesto, los topes metálicos a los que se hizo referencia en el apartado anterior. La incorporación de estas piezas al montaje resultó muy sencilla, ya que la forma de estas se ajusta correctamente para ser atornilladas a la base, aprovechando los roscados que se realizaron para el rail, e impidiendo que el carrito pueda salirse del rail por ambos extremos. En la siguiente imagen se puede ver esto último.

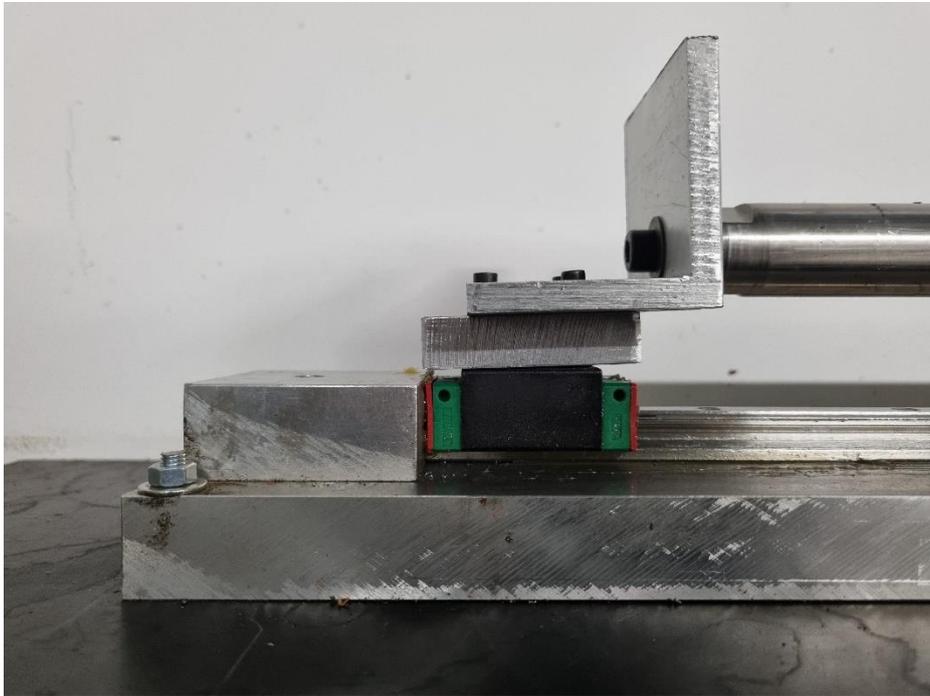


Figura 39: Tope metálico en contacto con el carrito.

Paralelamente a lo anterior, el otro elemento que se debe colocar sobre la misma base a la que se ha apelado, es el motor lineal, el cual se aloja en el interior de su carcasa metálica, la cual cuenta con cuatro agujeros para tornillos de métrica 6, en este caso, de mayor longitud, y que permitieron poder atornillar la carcasa, junto al motor, a la base, del mismo modo que se empleó para el rail. No obstante, sobre la base de lo indicado en el manual de instalación del motor, el vástago, para no ser interferido por fuerzas indeseadas en su movimiento, en concreto, ante la posibilidad de que se produjesen interacciones magnéticas, que perturbasen el control del sistema, debe cumplir con un requisito indicado por el fabricante. El vástago debe mantener siempre una distancia de 1 cm con cualquier superficie metálica, o de 2 cm en el caso de que

dicha superficie sea ferromagnética. En el caso concreto del diseño presentado, se contaba con un retal metálico de base de 45.5 cm y 10.1 cm de ancho. El espesor, que es la dimensión más relevante en este caso, era de 2.5 cm, por lo que resultaba apropiado para el montaje, permitiendo separar el vástago de los topes.

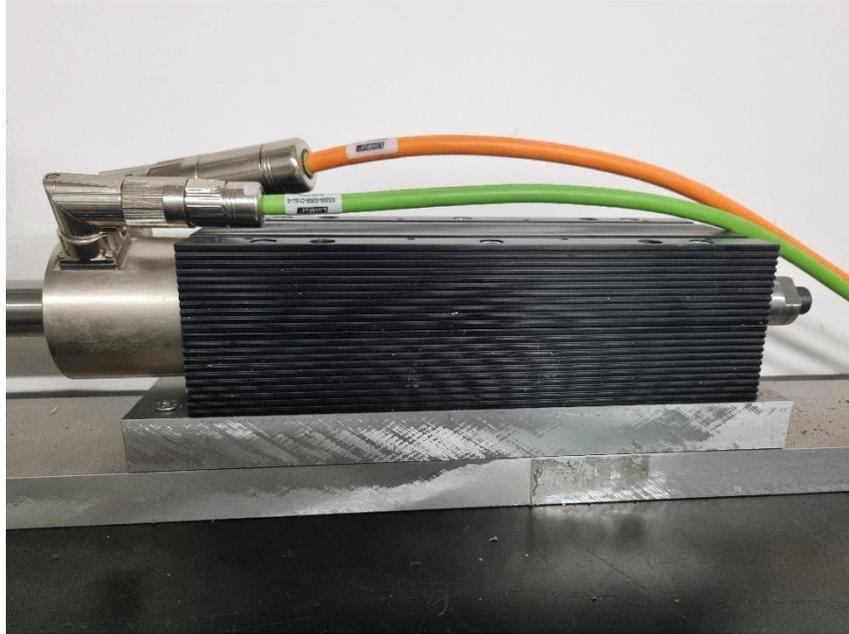


Figura 40: Estructura de soporte del motor instalado.

Habiendo definido todo lo anterior, lo cual supone la totalidad de los elementos que permitían realizar la configuración inicial del motor, así como las pruebas de funcionamiento que se pretendían ejecutar, se puede pasar a exponer el sistema de seguridad que se ideó para proteger a los usuarios del banco de ensayos. Mientras que la zona de experimentación es la que se ha ido describiendo en las líneas anteriores, en el extremo opuesto del motor, está la otra salida del vástago. Considerando que el motor permite realizar movimientos bruscos, a altas velocidades y fuerzas, los ensayos se han llevado a cabo teniendo en cuenta el riesgo existente, por lo que se han extremado las precauciones para impedir golpear a alguna persona o equipo, de modo que, durante los ensayos se ha liberado de obstáculos la posible trayectoria del vástago.

Adicionalmente, en relación a la problemática expuesta, se ha dispuesto un tubo metálico de aluminio, a través del cual circula el vástago. Conviene recordar la imposición establecida por el fabricante de respetar la distancia de 1 cm para materiales ferromagnéticos como en este caso. Para ello, el tubo se sostiene sobre unas pequeñas placas que se regulan en altura mediante unos tornillos dispuestos en posición vertical, del modo en el que se muestra en la siguiente figura.



Figura 41: Tornillo para la regulación en altura del tubo de seguridad.

Con todo lo expuesto, queda definido el banco de ensayos que se ha diseñado. Cabe mencionar que, dado que el vástago genera una fuerza de atracción magnética nada despreciable, no se deben acercar elementos metálicos.

En relación a la malla de separación a la que se hizo referencia en el apartado anterior, esta se coloca entre el montaje y el extremo de la mesa, en un punto en el que no moleste al recorrido del carrito durante el funcionamiento.

Tras todo lo anterior, en la siguiente fotografía se puede ver una muestra del resultado final en la que se incluyen los elementos principales del montaje.



Figura 42: Vista general del banco de ensayos diseñado.

3.3 Conexión de los equipos

En el presente apartado, se presentará el modo en el que se han conectado los diferentes equipos que incorpora el sistema. En un primer montaje, los elementos que se emplearon fueron, el motor y su controlador, un filtro para corriente trifásica, los cables de conexión, y el ordenador que haría las funciones de maestro. Posteriormente, ante la imposibilidad de trabajar

en tiempo real bajo el sistema operativo Windows, se sustituyó el ordenador por un equipo CompactRIO con el objetivo de que fuese este el sistema encargado de ejecutar el software de control.

En primer lugar, el motor cuenta con dos conexiones, una de ellas destinada a transmitirle la corriente trifásica mediante un cable KPS15-04-B/Q-5 [22] (de color naranja), de 5 metros de largo, cuyo terminal Q, consistente en un conector MC10-Q/f se conecta al terminal más ancho del motor, y el lado B, consistente en un conector MC10-B/m se conecta al controlador, mediante su adaptador, al puerto de corriente trifásica del controlador.

El otro cable, de color verde, cuyo propósito es transmitir la información del encoder interno del motor al controlador para poder efectuar el control de la posición, es el KSS05-02/08-D15/J-5 [23], también de 5 metros de largo, cuyo terminal J, que se trata de un conector MC10-J/f, se conecta al terminal más estrecho del motor, mientras que el lado D15, consistente en un conector MC01-D15/f, se conecta al puerto destinado al encoder del controlador a través de su adaptador.

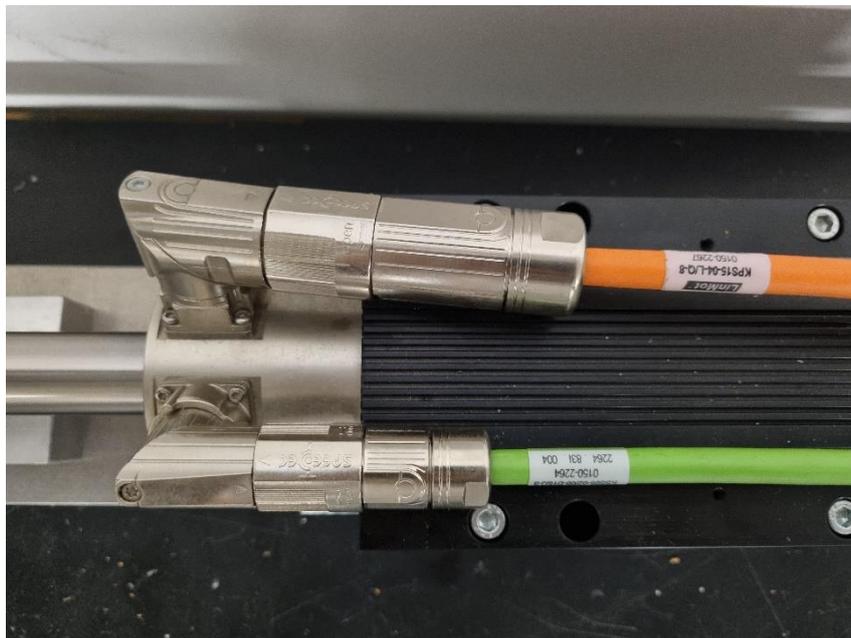


Figura 43: Conectores de alimentación y control del motor lineal.

Estas mismas conexiones que se muestran en la anterior representación, llegan en su terminal opuesto al controlador, y estas se deben conectar del modo en el que se muestra a continuación.



Figura 44: Conexiones de alimentación y control en el lado del controlador.

La alimentación trifásica llega al motor a través del controlador ya que este incorpora en su interior una etapa de potencia. De este modo, se ha de alimentar al controlador, a través del puerto de suministro de trifásica mediante el adaptador que incluye el set de conectores para el controlador E1450-EC-QN-0S. No obstante, la conexión a la red no se realiza directamente, sino que existen dos etapas intermedias. La primera de ellas es un interruptor magnetotérmico de seguridad ante la posibilidad de que se produzca una sobrecarga o un cortocircuito, y posteriormente, se hacen pasar las 3 fases a través de un filtro de red trifásica, el cual atenúa los niveles de perturbación de la señal, y en concreto se ha utilizado el modelo DLC 3016 del fabricante EMIKON [24].

Continuando con la etapa de alimentación, es necesario suministrar el controlador con una tensión continua de 24 voltios, la cual permite realizar todas las funciones disponibles de los equipos a excepción del movimiento del vástago, para lo cual si resulta necesaria la corriente trifásica que se comentó en el párrafo anterior. A través de una conexión a la red del edificio, lo que se puede obtener es una alimentación alterna, por lo que resulta necesaria una etapa previa a la conexión al controlador. En concreto, se ha empleado una fuente de alimentación conmutada del fabricante Lovato electric, y el modelo seleccionado fue el PSL1M 100 24 [25], que trabaja ofreciendo una tensión continua de salida de 24 voltios, una corriente nominal de 4.2 amperios, y por tanto, una potencia de 100 vatios. La salida se conecta al puerto X4 del controlador, a los pines +24VDC (X4.2) y DGND (X4.1).

Terminando con la exposición del conexionado eléctrico de los componentes, se conectaron las tomas a tierra de todos los equipos.

En otro sentido de las ideas, las comunicaciones se realizan físicamente a través de cables ethernet estándares. Para la configuración del software interno del controlador, se conecta el terminal X15 (*Config Ethernet*), a un puerto de red del ordenador. De este modo, mediante programas como *LinMot-Talk* se puede realizar la configuración de parámetros importantes de funcionamiento como la posición de referencia, las características físicas del equipo empleado, ajustes del PID de control y la masa que desplaza el motor entre otro, o mediante el TwinCAT generar el archivo ENI, necesario para que el programa conozca la tipología de la red y el formato de PDO empleado.

El otro cable ethernet necesario, es con el que se realiza la comunicación durante el funcionamiento, y este debe unir el puerto X17 (*Real Time Ethernet*) del controlador, con la conexión de la tarjeta de red del ordenador en la que se haya instalado el controlador de la librería de EtherCAT para LabVIEW.

3.4 Selección del sensor de posición

El nivel de precisión con el que se trabaja en la estación de simulación de la interacción entre pantógrafo y catenaria exige contar con una extraordinaria precisión a la hora de conocer la posición exacta a la que se encuentra la catenaria, la cual es la parte del sistema cuyo movimiento se simula. Por su parte, el pantógrafo sí que es real.

En este mismo sentido de las ideas, el motor lineal instalado cuenta con un encoder interno que permite un control de la posición del vástago, el cual ha sido suficiente a nivel de precisión para los ensayos descritos en este TFM, siendo este el motivo por el que no se ha incorporado al sistema un equipo adicional. No obstante, de cara a la implementación futura en el sistema principal, resulta necesario añadir un hardware externo, de mayores prestaciones para monitorizar esta variable.

Existen diferentes tipos de sensores, cuya señal de salida permite, de forma directa o indirecta, la posición del vástago respecto a un punto de referencia, el cual interesa que sea la posición de *home* establecida a nivel de software.

Entrando en el detalle de las especificaciones con las que debía contar el sensor, la precisión necesaria era de 5μ y un periodo de muestreo de por lo menos 0.1 ms, es decir, una frecuencia de toma de datos superior a los 10000 Hz. La señal de salida debía ser analógica, pudiendo recogerse el dato en nivel de tensión o de corriente, y el rango de medida mínimo necesario era de 5 cm, lo cual se corresponde con la amplitud dentro de la que suele oscilar la altura de la catenaria debido a los apoyos sobre los que se sustenta, y a la fuerza que le aplica el pantógrafo.

En el motor lineal que controla el movimiento vertical de la catenaria, el cual ya estaba instalado en el sistema principal en el momento de comienzo de este proyecto, se empleaba para el propósito de medida de la posición un sensor láser de la marca Panasonic®. Por esta razón, se comenzó analizando el catálogo de este fabricante con el propósito de comprobar si existía un modelo que se ajustase a las características mencionadas.

En primer lugar, se estudió la serie HL-G1 [26], los cuales trabajan con un método de triangulación, en el que se conoce la distancia por el ángulo al que se refleja el haz de luz, y que permiten medidas de alta resolución para superficies en las que resulta indiferente el color o la las condiciones de su superficie.

El modelo HL-G112-A-C5, cumple los requisitos de contar con salida analógica y el rango de medida es de 6 cm centrados en una referencia a 12 cm del láser, pero no permite trabajar a la frecuencia de muestreo y resolución deseadas.

Sensor láser con salida analógica (modelo estándar)					
Modelo estándar	HL-G103-A-C5	HL-G105-A-C5	HL-G108-A-C5	HL-G112-A-C5	HL-G125-A-C5
Altas prestaciones	HL-G103-S-J	HL-G105-S-J	HL-G108-S-J	HL-G112-S-J	HL-G125 S-J
Clase láser	2				
Salida analógica	0 - 10V / 4 - 20mA				
Rango de medida	30 ± 4mm	50 ± 10mm	85 ± 20mm	120 ± 60mm	250 ± 150mm
Diámetro del spot	0,1 x 0,1mm	0,5 x 1mm	0,75 x 1,25mm	1,0 x 1,5mm	1,75 x 3,5mm
Frecuencia de muestreo	200µs, 500µs, 1ms, 2ms		>100 US		
Resolución	0,5µm	1,5µm	2,5µm	8µm	20µm
Linealidad	+/- 0,1% F.S.				
Longitud de onda del láser	655nm				
Máx. potencia del elemento emisor	1mW				
Salida a transistor	máx. 50mA				
Interfaces de comunicación	Solo en el modelo de altas prestaciones RS422 o RS485 (seleccionable)				
Material	Carcasa: PBT / Cubierta Frontal: Acrílico / Cable: PVC				
Protección	IP67				
Tamaño (AlxAxL)	60 x 57 x 20,4mm				
Cable	Modelo estándar: 5m Modelo de altas prestaciones: cable con conector, 0,5m				
Cable de extensión	El cable opcional permite prolongar la longitud hasta los 20m (el cable del modelo estándar no se puede prolongar).				
Tensión de operación	24V DC (+/-10%)				
Temperatura ambiente	-10°C a +45°C, almacenamiento: -20°C a +60°C				
Humedad	de 35 a 85% RH, almacenamiento: de 35 a 85% RH				
Peso (aprox.)	HL-G1xx-A-C5 320g y HL-G1xx-S-J 110g				

Figura 45: Serie de sensores láser HL-G1. Modelo HL-G112-A-C5.

De entre los modelos de altas prestaciones, el HL-G108-RS-J, permite bajar a una resolución de 2.5 µm, pero el rango de medida se reduce a únicamente 1 cm.

Sensor láser con salida analógica (modelo de altas prestaciones)			
Standard type	HL-G103A-RA-C5	HL-G105A-RA-C5	HL-G108A-RA-C5
High functionality	HL-G103A-RS-J	HL-G105A-RS-J	HL-G108A-RS-J
Clase láser	2		
Salida analógica	0 - 10V / 4 - 20mA		
Rango de medida	26,3 ± 2mm	47,30 ± 5mm	82,9 ± 10mm
Diámetro del spot	0,1 x 0,1mm		0,2 x 0,2mm
Frecuencia de muestreo	200µs, 500µs, 1ms, 2ms		
Resolución	0,5µm	1,5µm	2,5µm
Linealidad	±0,1% F.S.		
Longitud de onda del láser	655nm		
Máx. potencia del elemento emisor	1mW		
Salida a transistor	máx. 50mA		
Interfaces de comunicación	Solo en el modelo de altas prestaciones RS422 o RS485 (seleccionable)		
Material	Carcasa: PBT / Cubierta Frontal: Acrílico / Cable: PVC		
Protección	IP67		
Tamaño (AlxAnxL)	60 x 57 x 20,4mm		
Cable	Standard type: 5m High functionality type: cable with connector, 0.5m		
Cable de extensión	El cable opcional permite prolongar la longitud hasta los 20m (el cable del modelo estándar no se puede prolongar).		
Tensión de operación	24V DC (+/-10%)		
Temperatura ambiente	-10°C to +45°C, storage: -20°C to +60°C		
Humedad	de 35 a 85% RH, almacenamiento: de 35 a 85% RH		
Peso (aprox.)	HL-G1xx-A-C5 380g y HL-G1xx-S-J 160g		

Figura 46: Serie de sensores láser HL-G1. Modelo HL-G108-RS-J.

Viendo que ninguno de los modelos cumplía simultáneamente todos los requisitos, se pasó a analizar la serie HG-C [27], que consisten en equipos más compactos que los anteriores.

El HG-C1200, tanto en su variante NPN como en la PNP, cuentan con un rango de medida de 8 cm y salida analógica, pero de nuevo, la repetitividad con la que cuenta, así como el tiempo de respuesta, imposibilitan su uso para esta aplicación.

NPN	HG-C1030	HG-C1050	HG-C1100	HG-C1200	HG-C1400
PNP	HG-C1030-P	HG-C1050-P	HG-C1100-P	HG-C1200-P	HG-C1400-P
Distancia al centro	30mm	50mm	100mm	200mm	400mm
Rango de medida	+/-5mm	+/-15mm	+/-35mm	+/-80mm	+/-200mm
Repetitividad	10µm	30µm	70µm	200µm	300µm (200-400mm) 800µm (400-600mm)
Linealidad	±0,1% F.E.				
Deriva térmica	±0,03% F.E.				
Fuente de luz	Láser semiconductor rojo (655nm), clase 2 (JIS/IEC/GB)/clase II (FDA)				
Haz	~ Ø 50µm	~ Ø 70µm	~ Ø 120µm	~ Ø 300µm	~ Ø 500µm
Tensión de alimentación	de 12 a 24VDC +/-10%				
Salida de control	1 salida (conmutable Con Luz/En Ocuridad), Corriente de carga: 50mA o menos				
Salida analógica	Salida de tensión: 0 a 5V (Alarma: +5,2V) / Impedancia de salida: 100Ω Salida de corriente: 4 a 20mA (Alarma: 0mA) / Impedancia de salida : máx. 300Ω Seleccionable vía configuración				
Tiempo de respuesta	Alta velocidad: 1,5ms / Estándar: 5ms / Alta precisión: 10ms (Conmutable)				
Entrada externa	1 entrada (Ajuste a cero, Aprendizaje, Láser ON/OFF, Selección de la función de Trigger)				
Grado de protección	IP67				
Cable	Cabe de 5-hilos, 2m				
Peso	35g (sin cable), 85g aprox. (con cable)				

Figura 47: Serie de sensores láser HG-C1.

El siguiente modelo que se estudió es el HL-C1 [28], el cual, como se puede ver en la siguiente imagen, cumple todos los requisitos ya que cuenta con salida analógica, un rango de medida de 20 cm, una frecuencia de muestreo de 10 kHz y salida analógica de corriente. El inconveniente en este caso es que se trata de un producto descatalogado, por lo que también hubo que descartar esta opción.

	HLC105BBK	HLC108BBK	HLC135BBK	HLC105B	HLC108B
Tipo	propósito general y caucho negro			reflexión especular	
Clase láser	2				
Rango de medida	50 ± 5mm	85 ± 20mm	350± 200mm	46 ± 4mm	81.5 ± 16 mm
Diámetro del spot	70 x 120µm	100 x 140µm	400 x 200µm	70 x 120µm	100 x 140µm
Frecuencia de muestreo	10kHz				
Resolución	1µm	6µm	1µm	1µm	3µm
Longitud de onda del láser	685nm				
Máx. potencia del elemento emisor	1mW				
Material	Aluminio				
Protección	IP67				
Tamaño (HxWxL)	87 x 82 x 26.6mm				
Cable	0.5m con conector				
Temperatura	0°C a +50°C				
Características de la temperatura	0.01 % F.S. / °C		0.02 % F.S. / °C	0.01 % F.S. / °C	
Peso (aprox.)	500g				

DISCONTINUADO

Características:

- Alta resolución de 1µm independiente de la frecuencia de muestreo
- Frecuencia de muestreo de 10kHz
- No se ve afectado por el color o la superficie del objeto
- Salida analógica de corriente 4-20mA
- Error de linealidad máx. +0.2% F.S. (fondo de escala).
- Diseño compacto

Opciones:

- Cable (2 / 5 / 10 / 20 / 30m)

Especificaciones:

Figura 48: Serie de sensores láser HL-C1.

El fabricante recomienda pasar a la serie HL-C2 [29], los cuales pasan a ser dos dispositivos, el sensor y su controlador, por separado. En este caso, se trata de unos dispositivos con una tecnología muy avanzada que permite contar con grandes prestaciones. El inconveniente es que, para poder trabajar dentro del rango de medida deseado, hay que escoger el modelo de mayores prestaciones de la gama, el HL-C235CE-W, o el HL-C235CE-WMK, que son las variantes con haz de luz fino, o lineal respectivamente. El rango de medida es de 20 cm, la resolución es de 2 µm, el periodo de muestreo se puede seleccionar en 100 µs, y dispone de salida analógica tanto en tensión como en corriente.

Referencias Modelo de haz fino	HL-C208B [E]	HL-C208C[E]		HL-C211F [E]		HL-C211F5[E]		HL-C235CE-W
Referencias Modelo de haz lineal	HL-C208B[E]-MK	HL-C208C[E]-MK		HL-C211F[E]-MK		HL-C211F5[E]-MK		HL-C235CE-WMK
Setup mode	Reflexión especular	Reflexión difusa	Reflexión especular	Reflexión difusa	Reflexión especular	Reflexión difusa	Reflexión especular	Reflexión difusa
Rango de medida	81,4 ± 6mm	85 ± 20mm	81,4 ± 6mm	110 ± 15mm	106,7 ± 14,5mm	110 ± 15mm	106,7 ± 14,5mm	350 ± 200mm
Resolución *)	0,15µm			0,1µm		2µm		
Clase láser	2	3R		2		3R		
Diámetro del haz Modelo de haz fino	ø100µm			ø80µm		ø400µm		
Beam size linear beam spot type	100 x 1200µm			80 x 1700µm		400 x 6500µm		
Longitud de onda láser	658nm							
Máx. potencia del elemento emisor	1mW	5mW		1mW		5mW		
Dimensiones (Al xAn xProf)	90 x 74 x 26mm							
Peso (aprox.)	300g (incluido el cable)							

*) Las referencias con [E] tienen una resolución de 0,25 µm

SENSOR

Modelo	Con puerto RS-232C		Con puerto Ethernet	
	Controlador (NPN) Hasta dos cabezas sensoras 2 HL-C2	Controlador (PNP) Hasta dos cabezas sensoras 2 HL-C2	Controlador (NPN) Hasta dos cabezas sensoras 2 HL-C2	Controlador (PNP) Hasta dos cabezas sensoras 2 HL-C2
Referencia	HL-C2C	HL-C2C-P	HL-C21C	HL-C21C-P
Número de cabezas sensoras	Máx. 2			
Tensión de alimentación	24 V CC ±10 % incluido el rizado 0.5 V (P-P)			
Consumo de corriente	Con 2 cabezas sensoras: 500mA aprox/ con 1 cabeza sensora: 350 mA aprox. (si se conecta la mini consola el consumo aumenta en unos 100 mA aprox.)			
Periodo de muestreo	10 µs, 20 µs, 40 µs, 100 µs, 200 µs, 400 µs, 1 ms, 2 ms			
Salida analógica	Salida de tensión: -5 a +5 V y salida de corriente: 4 a 20 mA			
Salidas digitales	Alarma, salida de resultado (OK/NG), estrobo, máx. 100mA 30V CC			
Entradas digitales	Bloqueo remoto, control del láser, temporizador, reset, cambio de memoria			
Indicadores	Potencia, radiación láser en la cabeza sensora, alarma			
Interfaces	Velocidad de transmisión: 9600, 19200, 38400, 115200 bit/s		IEEE802.3u, 10 Base-T / 100 Base-TX RJ45	
Puerto USB	USB 2.0 Full-speed			
Temperatura ambiente	de 0°C a 50°C			
Accesorios	CD-ROM: 1 pieza, cable USB (2m): 1 pieza, soporte: 1 pieza		CD-ROM: 1 pieza, cable USB (2m): 1 pieza, soporte: 1 pieza Núcleos de ferrita (E04SR200935A, de Seiya Electric Mfg. Co., Ltd.: 3 piezas	
Dimensiones (AlxAnxProf)	120x59x95mm (D = 105,5mm incluido el terminal a tornillo)			

CONTROLADOR

Figura 49: Serie de sensores láser HL-C2.

La razón por la que hubo que hubo que prescindir de esta opción es que, al tratarse del modelo más caro de la gama, el precio era excesivo para la dotación presupuestaria del proyecto con el que se está trabajando. Se entendió de este modo que no se estaba buscando la tecnología de medición de distancias más adecuada para las peculiaridades del sistema implementado. Se pasó de este modo a estudiar los sensores basados en la vibrometría. En concreto, se examinó la tecnología de vibrometría láser por efecto Doppler de Polytec [30]. Se anuncia como un método que ofrece muy buenas resoluciones de posición y velocidad, además de que la medida es lineal. Todo esto de forma independiente de la distancia de medida, e integrado en equipos robustos que permiten el trabajo en diferentes tipos de ambientes.

Existen seis tipologías principales de funcionamiento de la vibrometría, los equipos de punto único, los diferenciales, los de escaneo, los rotacionales, los microscópicos y los en plano. El primero de ellos consiste en la toma de la medida en un objeto que se desplaza en la dirección del haz emitido por el sensor, siendo esta la forma más simple de tomar la medida. Se pueden combinar varios sensores de este tipo para obtener un vibrómetro multipunto. Los vibrómetros diferenciales describen la medida en dos puntos relativos entre ellos. Existe un primer método,

en el que el haz de referencia del interferómetro se guía hacia el objeto, obteniéndose una gran fidelidad de la medida, lo cual es adecuado para altas frecuencias. El segundo método, la diferencia entre las dos medidas relativas de calcula de forma electrónica con dos interferómetros independientes, lo cual permite una mayor versatilidad a la hora de escoger una determinada configuración espacial de los elementos del sistema. En tercer lugar, la tecnología por escaneo consiste en que el haz de luz escanea toda la superficie mediante equipos de punto único, obteniéndose funciones de transferencia para cada uno de los puntos de medición. Los rotacionales obtienen la velocidad y desplazamiento angulares en estructuras rotatorias, lo cual es útil en estructuras de eje como generadores eléctricos o turbinas. En los microscópicos, la principal singularidad es el uso de lentes de microscopio ya que se emplea en componentes de pequeñas dimensiones. Por último, la tecnología en plano permite mediciones en la dirección perpendicular al eje de sensado, y es empleado para la medición de deformaciones de elevada dinámica. Se podría considerar una séptima categoría consistente en los vibrómetros multipunto, pero como se mencionó, se componen mediante la asociación de múltiples sensores de punto único.

En el caso del proyecto analizado, el modelo que mejor se ajusta a la necesidad de medir el desplazamiento lineal del vástago, el vibrómetro de punto único [31]. El fabricante Polytec ofrece tres modelos diferentes, los cuales se exhiben en la siguiente figura.



Figura 50: Modelos de vibrómetros del fabricante Polytec.

El VibroFlex es el nuevo modelo de la marca, y es el más flexible en cuanto a que es modular, pudiendo incorporar los elementos que se requieran para cada aplicación concreta. Puede medir simultáneamente la posición, la velocidad y la aceleración, además de que cumple con las características necesarias para su integración en el sistema presentado en este escrito. El problema de nuevo es que el precio es incluso mayor que el equipo que se presentó en último lugar con tecnología láser.

El modelo IVS-500 y el VibroGo son similares en cuanto a las posibilidades que ofrece, siendo el segundo de ellos el modelo más nuevo de la marca, y que sustituye, con un diseño más compacto y algunas ventajas en cuanto a prestaciones, al modelo IVS-500. En cualquiera de los dos casos, se cumplen los requisitos técnicos, pero exceden el coste sigue siendo muy elevado. Tras una reunión con el distribuidor en España de los productos expuestos, el coste era superior al modelo láser, aunque las prestaciones son superiores, por lo que ambas opciones podrían ser buenas en otro tipo de proyecto.

Atendiendo a todas las consideraciones realizadas a lo largo del presente apartado, queda claro que resultaba necesario escoger un sensor que estuviese especialmente preparado para la

aplicación en la que se quería instalar para, de ese modo, poderse ajustar a los requerimientos sin dificultad, y que por ello el coste no fuese excesivo como ha ocurrido en las otras alternativas.

La solución se encontró dentro del mismo fabricante del motor, se trata del encoder MS01-1/D-SSI [32], cuyo funcionamiento es mucho más simple que en los casos anteriores, como ya se comentó en el apartado 2.5.



Figura 51: Encoder MS01-1/D-SSI.

En relación con el cumplimiento de los requisitos necesarios, cuenta con una resolución de 1 μm , o de 5 μm trabajando en el modo incremental o absoluto respectivamente. En el modo absoluto, en todo momento la distancia es respecto a una referencia, mientras que, en el modo incremental, cada desplazamiento es independiente y el final de un traslado es la posición de referencia para el siguiente movimiento. Este último método es el que se usará en el proyecto, una vez se instale el motor en el sistema principal. Puede medir recorridos de hasta 10.24 metros, y la salida es digital, pudiendo muestrear a la frecuencia deseada y la información se tomaría muestreando la señal a través de los pines con los que cuenta el dispositivo.

En la siguiente gráfica se puede ver el comportamiento de las señales digitales de salida A y B, las cuales forman una secuencia gracias a la que se puede saber si se ha dado un paso y en qué sentido. En cuanto a las señales /A y /B, son las señales A y B respectivamente, pero negadas, por lo que para esta aplicación no se emplearán.

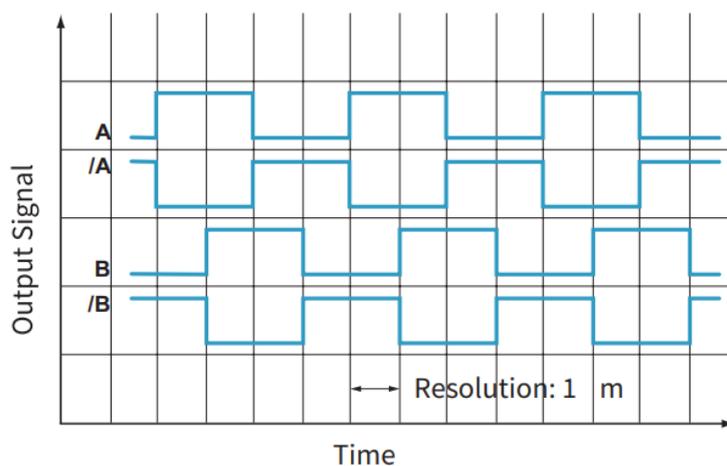
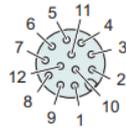


Figura 52: Señales de salida que ofrece el encoder MS01-1/D-SSI.

En la siguiente tabla, extraída de la hoja de características del dispositivo, se muestra a qué señal hace referencia cada uno de los pines del conector con el que cuenta el dispositivo, así como el código de colores empleado, lo cual será de gran utilidad para cablear correctamente los equipos.

En concreto, la señal A está en el pin 7, la señal B en el pin 9, la alimentación, que puede ser desde 4.5 voltios a 30 voltios, se realizaría a través del pin 5, y el cable a 0 voltios se encuentra en el pin 12.

Connector wiring



Pin	Connector wiring	Wire color Connection cable KSS01-12.../ABS- ENC-10	
1	nc	grey-pink	
2	D+	pink	
3	D-	grey	
4	T-	black	
5	+UB	white	
6	/A	green	
7	A	yellow	
8	/B	blue	
9	B	red	
10	Config	GND The sensor is in the SSI mode.	red-blue
		+UB (while encoder supply is being turned on) The sensor is in the boot loader mode for the first 10s (installing new firmware is enabled), then it changes over to the service mode.	
		 Setting of the position value to the calibration value (only if the sensor is in the SSI mode)	
11	T+	purple	
12	OV	brown	

Figura 53: Pines de conexión del encoder MS01-1/D-SSI.

Resulta muy importante leer el documento de instalación del dispositivo, ya que en él se detallan instrucciones muy precisas en relación al montaje, ya que las distancias y ángulos existentes entre el encoder, la banda magnética sobre la que se desplaza, y otros objetos externos, deben estar muy controlados. En la siguiente figura se resumen las dimensiones clave a considerar para que la medida ofrecida por el sensor cumpla con las características que anuncia el fabricante.

INSTALLATION

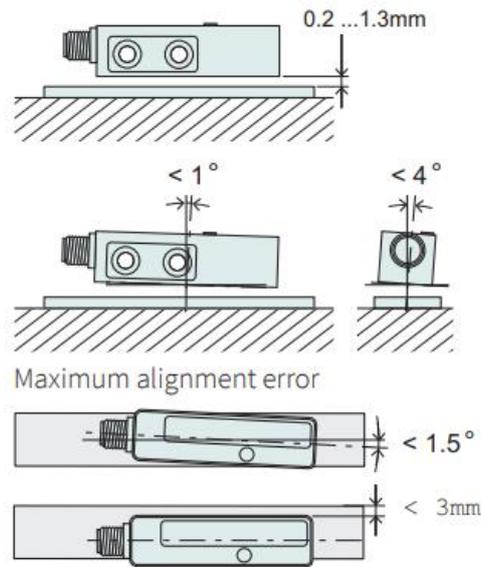


Figura 54: Tolerancias de montaje del encoder MS01-1/D-SSI.

En concreto, la separación entre el encoder y la banda magnética sobre la que se desplaza debe estar dentro del rango de entre 0.2 mm y 1.3 mm. La inclinación en el plano perpendicular al conector debe ser inferior a 1° , y respecto al plano del conector de 4° . La inclinación respecto al eje de la banda magnética debe ser inferior a 1.5° , y el dispositivo no se debe alejar en ningún caso más de 3 mm respecto al borde de la guía magnética.

3.5 Definición del fichero ENI

Para el control del motor lineal incorporado al diseño presentado en los apartados anteriores, se ha definido una topología de red muy sencilla, en la que únicamente se conecta el ordenador principal, el cual actúa como maestro, al controlador del motor, que es el esclavo de la red EtherCAT.

A fin de emplear los módulos de comunicación existentes en la librería de EtherCAT para LabVIEW, se debe indicar, a objeto de establecer la conexión entre los dos nodos de la red, mediante el denominado archivo ENI (*EtherCAT network Information*), la topología de red utilizada.

Existen diferentes formas de generar este archivo. En este proyecto se ha empleado el software TwinCAT. Se trata de un programa de BECKHOFF destinado a emplearse en la ejecución de programas de control en tiempo real, y que incluye herramientas para el diagnóstico y configuración de estos equipos. Será esta última, la única funcionalidad que se aprovechará para este proyecto, ya que el control y monitorización del sistema se realizará desde LabVIEW.

En cuanto al software, en primer lugar, se instaló la versión 2 PLC, que es la base, y posteriormente se incluyó la actualización a TwinCAT 3. Para comenzar la configuración se debe

conectar mediante cable ethernet, el puerto X17 del controlador, al puerto de la tarjeta de red que se va a utilizar en el ordenador.

Junto con el software LinMot-Talk, el cual se presentó en el apartado 2.7, se incluye un archivo de extensión .xml, con la descripción del dispositivo que se está conectando, la cual resulta necesaria para poder realizar la configuración del controlador en TwinCAT. El archivo NTIL_LM_SERVOS_V1_7.xml, presente en la carpeta EtherCAT/XML, ubicada en la carpeta de instalación de LinMot-Talk, se debe copiar a la ubicación C:\TwinCAT\3.1\Config\Io\EtherCAT\. Tras esto, el software podrá reconocer los controladores necesarios automáticamente al conectar el dispositivo LinMot al ordenador.

El controlador del motor lineal se alimenta mediante tensión continua de 24 voltios, y se conecta al ordenador del modo en el que se indicó en párrafos anteriores. Tras esto, y posterior a la creación de un nuevo proyecto en TwinCAT XAE Shell, se puede buscar automáticamente el dispositivo conectado con la opción "Scan" que se muestra en la figura 55.

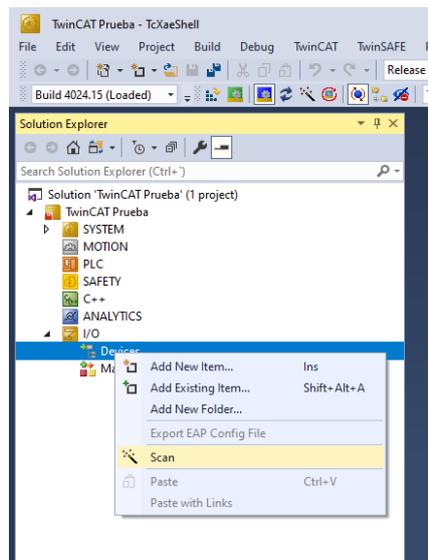


Figura 55: Búsqueda de dispositivos compatibles en TwinCAT.

De este modo, se encontrará el dispositivo, y se aceptará a la solicitud del software de comenzar a buscar los paquetes asociados al controlador.

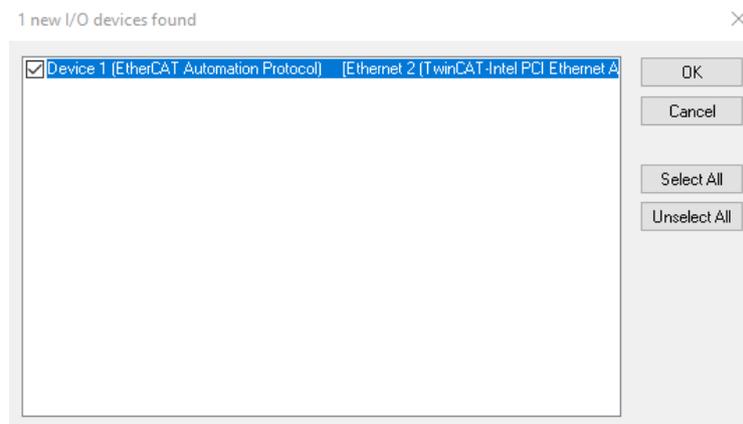


Figura 56: Dispositivos encontrados por el software TwinCAT.

Una vez que se han realizado estos pasos, se puede acceder a la configuración del E1450-EC-QN-0S. La acción más relevante para el proyecto, la cual se debe realizar mediante el software TwinCAT, es la configuración del PDO (*Process Data Object*) que se empleará en el intercambio de mensajes entre los dos nodos de la topología de red. Este archivo quedará configurado en el maestro (ordenador principal), y será transmitido al controlador en el momento del arranque del sistema. De este modo, durante el funcionamiento, tanto el maestro, como el esclavo, podrán interpretar correctamente los mensajes que son transmitidos. Resulta conveniente mencionar que, cuando en el manual de TwinCAT, o en el propio software, se menciona “entradas”, estas se refieren desde el punto de vista del maestro, por lo que las “salidas” son los envíos desde el maestro hacia el esclavo.

En esta configuración se puede escoger entre diferentes tipos de mensajes de entrada y de salida. Sin embargo, en este caso resulta adecuado trabajar con las entradas y salidas por defecto, las cuales, además son con las que se trabaja en los ejemplos que incluye la librería de EtherCAT para LabVIEW.

3.1.3 Default Inputs 0x1B00

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B00	18	-	Variables	RECORD
0x1B62:00	2	0	StateVar	Uint16
0x1D51:00	2	2	StatusWord	Uint16
0x1D8E:00	2	4	WarnWord	Uint16
0x1B8A:00	4	6	DemandPosition	Int32
0x1B8D:00	4	10	ActualPosition	Int32
0x1B93:00	4	14	DemandCurrent	Int32

Figura 57: PDO de entrada por defecto.

En la tabla anterior, extraída del manual de EtherCAT [9], se describe la entrada por defecto. Esta consiste en una cadena de 18 bytes donde los dos primeros bytes sirven para monitorizar el motor en cuanto al estado de conexión, del firmware, del motor (apagado o encendido), y el estado de operación. Estas variables se muestran también en la siguiente ventana del LinMot-Talk.

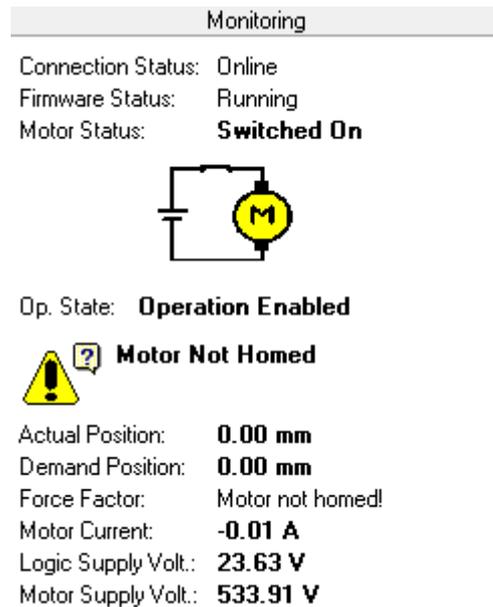


Figura 58: Cuadro de monitorización del LinMot-Talk.

En segundo lugar, se puede acceder a la palabra de estado del motor, consistente en 2 bytes que permiten conocer la información mostrada en la siguiente figura.

Status		<<	
0: Operation Enabled.....	1	0: Motor Hot Sensor.....	0
1: Switch On Active.....	1	1: Motor Short Time Overload...	0
2: Enable Operation.....	1	2: Motor Supply Voltage Low...	0
3: Error.....	0	3: Motor Supply Voltage High...	0
4: Voltage Enable.....	1	4: Position Lag Always.....	0
5: /Quick Stop.....	1	5: Reserved.....	0
6: Switch On Locked.....	0	6: Drive Hot.....	0
7: Warning.....	1	7: Motor Not Homed.....	1
8: Event Handler Active.....	0	8: PTC Sensor 1 Hot.....	0
9: Special Motion Active.....	0	9: PTC Sensor 2 Hot.....	0
10: In Target Position.....	0	10: RR Hot Calculated.....	0
11: Homed.....	0	11: Speed Lag Always.....	0
12: Fatal Error.....	0	12: Position Sensor.....	0
13: Motion Active.....	0	13: Reserved.....	0
14: Range Indicator 1.....	1	14: Interface Warn Flag.....	0
15: Range Indicator 2.....	0	15: Application Warn Flag.....	0
Status Word:	40B7h	Warn Word:	0080h
Op. Main State	08h	Logged Error Code:	0000h
Op. Sub State	00h		

Figura 59: Ventana de estado del motor en LinMot-Talk.

Por otro lado, la palabra de advertencia, en la que se indica el código de error existente en el caso de que exista algún problema que impida el correcto funcionamiento del motor.

A continuación de lo anterior, se encuentran tres espacios de 4 bytes destinados al envío de información relativa a la posición que se está demandando alcanzar, la posición actual del vástago respecto a la referencia establecida, y la demanda de corriente que está solicitando el motor en cada instante de funcionamiento. Este último dato presenta una gran relevancia para el estudio de la caracterización del motor que se quiere realizar.

Actual Position: **0.08 mm**
 Demand Position: **0.00 mm**
 Force Factor: 100.00 %
 Motor Current: **-0.83 A**
 Logic Supply Volt.: **23.74 V**
 Motor Supply Volt.: **534.19 V**

Figura 60: Datos de funcionamiento del motor en LinMot-Talk.

En otro sentido de las ideas, y bajo el mismo criterio que para las entradas, se definió activando la casilla 0x1700 del cuadro que se muestra en la siguiente figura, que se quería trabajar con la salida por defecto.

Sync Manager:

SM	Size	Type	Flags
0	128	MbOut	
1	128	MbIn	
2	24	Outputs	
3	18	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1B21	4.0	Input Command Table Var 2	F		0
0x1602	8.0	Cyclic Position Mode	F		0
0x1603	12.0	Cyclic Position Velocity Mode	F		0
0x1604	14.0	Cyclic Position Velocity Mode	F		0
0x1606	6.0	Cyclic Torque Mode	F		0
0x1620	0.0	User Defined Outputs			0
0x1700	24.0	Default Outputs	F	2	0
0x1701	26.0	NC Outputs	F		0
0x1703	2.0	Control Word	F		0
0x1704	6.0	Motion Command Intf #6	F		0
0x1708	8.0	Config Module	F		0
0x1710	4.0	Max Current Set A	F		0

PDO Assignment (0x1C12):

- 0x1602 (excluded by 0x1700)
- 0x1603 (excluded by 0x1700)
- 0x1604 (excluded by 0x1700)
- 0x1606 (excluded by 0x1700)
- 0x1620
- 0x1700
- 0x1701 (excluded by 0x1700)
- 0x1703 (excluded by 0x1700)
- 0x1704 (excluded by 0x1700)
- 0x1708
- 0x1710

PDO Content (0x1700):

Index	Size	Offs	Name	Type	Default (hex)
0x1D52:00	2.0	0.0	Control Word	UINT	
0x1DB0:...	2.0	2.0	Motion Command Header	UINT	
0x1E40:00	4.0	4.0	Motion Command Par 1	DINT	
0x1E41:00	4.0	8.0	Motion Command Par 2	DINT	
0x1E42:00	4.0	12.0	Motion Command Par 3	DINT	
0x1E43:00	4.0	16.0	Motion Command Par 4	DINT	
0x1E44:00	4.0	20.0	Motion Command Par 5	DINT	
		24.0			

Figura 61: Selección del PDO de salida por defecto en TwinCAT.

3.2.6 Default Outputs 0x1700

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1700	24	-	Variables	RECORD
0x1D52:00	2	0	ControlWord	Uint16
0x1DB0:00	2	2	MotionCommandHeader	Uint16
0x1E40:00	4	4	MotionCommand Par 1	Word32
0x1E41:00	4	8	MotionCommand Par 2	Word32
0x1E42:00	4	12	MotionCommand Par 3	Word32
0x1E43:00	4	16	MotionCommand Par 4	Word32
0x1E44:00	4	20	MotionCommand Par 5	Word32

Figura 62: Elementos del PDO de salida por defecto.

El PDO de salida lo compone una cadena de 24 bytes, donde el primer byte es el número 39 de la cadena completa que se envía a través del bloque correspondiente en LabVIEW. Los dos primeros bytes hacen referencia a la palabra de control que se quiere enviar. Esto sirve para poner en funcionamiento el motor, anular una advertencia de un error que se ha producido, o realizar el procedimiento de alcanzar el punto de referencia al arranque del sistema, entre otras posibilidades.

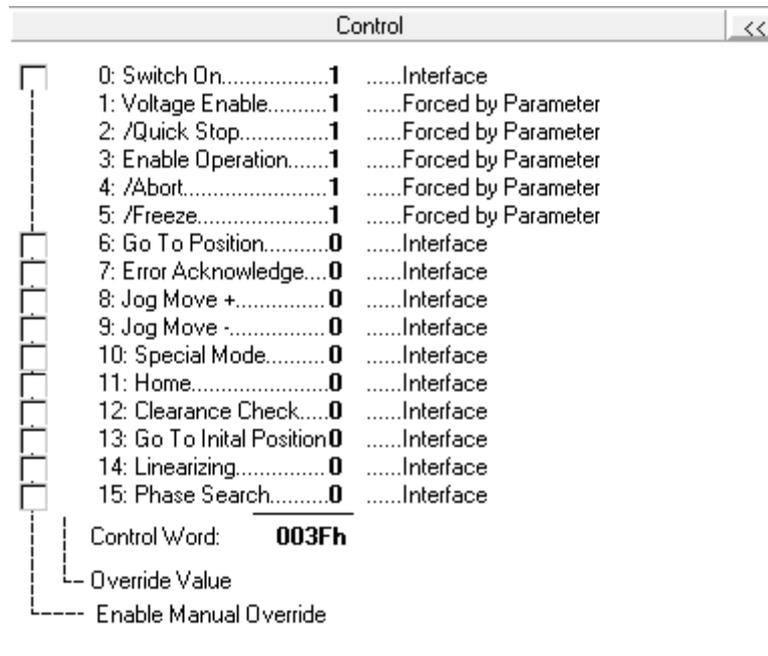


Figura 63: Panel de control del motor en LinMot-Talk.

A continuación de lo anterior se encuentra la cabecera, en la cual se indica el comando de movimiento que se pretende utilizar. Además, el último dígito de esta parte del mensaje debe cambiar entre cada envío de mensaje para que el controlador realice los cambios indicados en los bytes que siguen a estos. La forma en la que se ha resuelto esto último, es alternar entre 0 y 1 en cada envío de un mensaje, el bit de la posición menos significativa de la cabecera.

Además, según el tipo de comando de movimiento escogido, los siguientes tramos del mensaje tendrán un significado u otro. Por ejemplo, para la cabecera (0300h), correspondiente al comando de movimiento *P Stream With Slave Generated Time Stamp*, los cuatro primeros bytes del comando de movimiento hacen referencia a la posición a la que se quiere mover el vástago, mientras que para la cabecera (0310h), correspondiente al comando *PV Stream With Slave Generated Time Stamp*, los cuatro primeros bytes coinciden con los del caso anterior, pero además se emplean los cuatro bytes siguientes para indicar a la velocidad a la que se desea realizar la orden de movimiento.

Con todo lo anterior configurado, se puede proceder a generar el fichero ENI necesario para la librería de EtherCAT para LabVIEW. Se accede a la ventana mostrada en la siguiente figura, y en la opción *PCI Bus/Slot* se selecciona el puerto de la tarjeta de red en el que se encuentra conectado el controlador a través del cable ethernet.

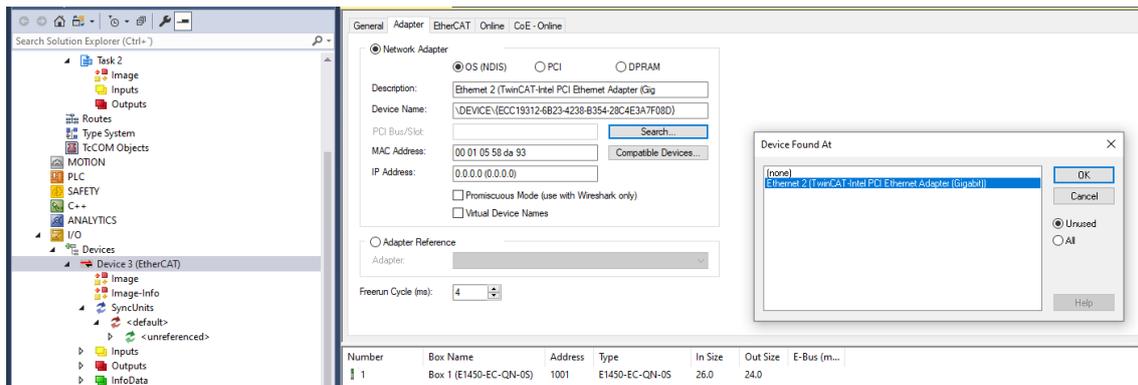


Figura 64: Selección del puerto con la tarjeta de red donde se encuentra el controlador.

Posteriormente, se accede a la siguiente ventana, en la que ya es posible seleccionar la opción *Export Configuratio File*, que devolverá el archivo con extensión .XML necesario.

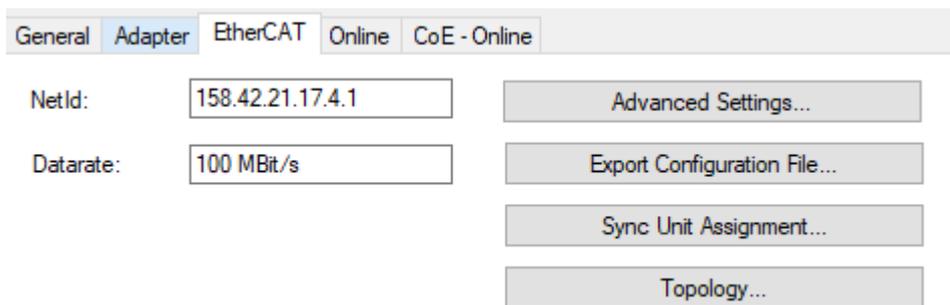


Figura 65: Generación del fichero ENI en TwinCAT.

3.5.1 PDO de entrada

En este proyecto, aunque el PDO de salida adquiriera un mayor protagonismo, es importante analizar también la forma en la que se construye el PDO de entrada, ya que se debe poder acceder a ciertos valores de dicho mensaje con el objetivo de monitorizar ciertas variables relacionadas con el funcionamiento del motor.

En este sentido, ya al inicio de este apartado se analizó el significado de cada uno de los 18 bytes que componen este mensaje por defecto. De este modo, en las siguientes líneas lo que se analizará es la información que resulta necesaria recopilar para el proyecto que se está describiendo.

En primer lugar, para poder llevar a cabo el estudio de la caracterización del motor lineal, resultaba necesario conocer la corriente que demanda el motor en cada momento, durante el funcionamiento. Para este propósito se puede acceder a los 4 bytes que muestran esta información, y que se ubican a partir del byte número 14 del mensaje. Pasando los valores hexadecimales de estas posiciones a su correspondiente equivalencia en notación decimal, se puede conocer los miliamperios demandados por el motor para la descripción de la cinemática que se solicite desde el controlador.

No obstante, y tal y como se presentará más adelante en el escrito, el programa de control y monitorización implementado en LabVIEW, genera una matriz con toda la información que

contiene el PDO de entrada, empleando para ello los bloques de la librería de EtherCAT para LabVIEW. De este modo, para conocer el valor de corriente demandada, únicamente resulta necesario acceder al valor almacenado en la posición (5,5) de dicha matriz. Los datos almacenados son cadenas de caracteres, por ello, antes de llevar el valor, por ejemplo, a un indicador en el panel frontal de LabVIEW, hay que emplear el bloque que convierte las cadenas de caracteres en notación decimal, a su correspondiente valor en formato numérico.

3.5.2 PDO de salida

Al inicio de este apartado, se presentó el PDO (*Process Data Object*) de salida con el que se trabajará, el cual se debe generar en el programa implementado en LabVIEW para posteriormente enviarlo a través de la red EtherCAT al controlador, y que el motor ejecute las órdenes deseadas.

Como se ha expuesto previamente, para el sistema que se quiere implementar, consta de cuatro campos principales.

En primer lugar, la palabra de control, de dos bytes, y en la que se configuran las órdenes de control del motor. Los dieciséis bits se pueden configurar a 1 o a 0 según si se quiere activar o desactivar respectivamente los siguientes ajustes:

- Bit0: Activar el motor.
- Bit1: Alimentación habilitada. Se mantiene a 1 siempre.
- Bit2: Parada rápida habilitada. Se mantiene a 1 siempre.
- Bit3: Modo de operación habilitado. Se mantiene a 1 siempre.
- Bit4: Función de abortar movimiento habilitada. Se mantiene a 1 siempre.
- Bit5: Función de parar el movimiento habilitada. Se mantiene a 1 siempre.
- Bit6: Ir a una posición previamente definida en la memoria interna del controlador.
- Bit7: Anular mensaje de error.
- Bit8: Mover el vástago en sentido positivo unos mm definidos previamente en la memoria interna.
- Bit9: Igual que el bit anterior, pero en sentido negativo, es decir, hacia la izquierda en la configuración escogida.
- Bit10: Activar modo de funcionamiento especial.
- Bit 11: Realizar el procedimiento de *home* (ir a la posición de referencia).
- Bit 12: Control de seguridad habilitado.
- Bit 13: Ir a la posición inicial. Es necesario haber realizado el procedimiento de *home* previamente.
- Bit 14: Linealización de las órdenes de movimiento.
- Bit 15: Búsqueda de fase.

De este modo, mediante una representación en hexadecimal, el modo desconectado se consigue con 003E. Para pasar al modo de operación se mandaría la orden 003F. Una vez el motor está habilitado, para anular un mensaje de error se mandaría la orden 00BF, y para iniciar el procedimiento de *home* 083F. Posteriormente, y si todo el procedimiento ha sido correcto, se puede volver al estado 003F, y pasar a modificar los otros campos del PDO.

En concreto, los dos bytes siguientes a la palabra de control son la cabecera. En ella, se indica el tipo de comando de movimiento que se desea enviar. Entrando en un grado de detalle un poco mayor, los dos bytes más significativos indican el identificador del grupo de comandos, y el siguiente byte indica el comando concreto dentro del grupo indicado. Por ejemplo, los dos modos que más se han estudiado en este Trabajo de Fin de Máster han sido el *P Stream With Slave Generated Time Stamp*, cuya cabecera comenzaría por 030, y el *PV Stream With Slave Generated Time Stamp*, con la cabecera 031 en hexadecimal. Finalmente, el último byte de la cabecera se emplea para que el controlador sepa que se quiere enviar una nueva orden. Por esta razón, en cada envío de PDO, este byte debe cambiar ya que, en caso contrario, el controlador no haría caso de la orden que le llegue. No es necesario ir cambiando el valor de 0 a F, hay que recordar que se está trabajando con notación hexadecimal, sino que alternando entre 0 y 1 en ese último byte, para cada envío de un mensaje es suficiente, siendo esta la alternativa que se ha implementado en el programa de LabVIEW.

El siguiente tramo del PDO de salida es el conjunto de 4 bytes de la primera parte del comando de movimiento. El significado de esta parte del mensaje cambia según la cabecera escogida, pero en los dos ejemplos presentados en el apartado anterior el significado es el mismo, y es la posición a la que se quiere desplazar el vástago, respecto a la referencia fijada en el procedimiento de *home*. La mejor forma de saber el valor que se debe enviar es que en decimal consiste en el número de milímetros de la posición a alcanzar, multiplicado por diez mil. Ese número, que estará en notación decimal, se convertirá a hexadecimal posteriormente.

Si se emplea el comando *PV Stream With Slave Generated Time Stamp*, el PDO incluirá un elemento más, el segundo tramo del comando de movimiento. El funcionamiento es muy similar al anterior, pero en ese caso concreto, esta zona del mensaje indica la velocidad a la que se quiere realizar el movimiento. En decimal, se puede calcular multiplicando por mil el número de milímetros por segundo al que se quiere configurar el movimiento.

En otro sentido de las ideas, y atendiendo a la implementación realizada en LabVIEW, el bloque de envío de datos con dirección al controlador que se emplea recibe una cadena de números decimales, donde cada uno de ellos representa un byte de la cadena. Es decir, un número de entre 0 y 255. Es importante indicar que el primer byte, de la palabra de control, es el número 39 de la cadena completa que se envía. De este modo, la cabecera comienza en el 41, la primera parte del comando de movimiento en el 43, y la segunda parte, si es necesaria, en el byte 47.

Habiendo definido todos los elementos que intervienen en la definición del PDO de salida, en los siguientes párrafos, se expondrá el método empleado para empaquetar todos los datos una única cadena de bytes, para posteriormente realizar el envío del mensaje hacia el controlador. El motivo que justifica la necesidad de enviar toda la información junta es que el tiempo es una variable de mayúscula relevancia en este proyecto. Hay que recordar que la velocidad de la comunicación es uno de los motivos de mayor peso a la hora de haber decidido trabajar con un controlador EtherCAT. Enviar toda la información en un único mensaje, permite que se actualice toda la información de la posición, y velocidad en el caso

del modo *PV Stream*, en cada iteración, y por tanto, a la máxima velocidad de comunicación permitida por el hardware.

La palabra de control comienza en el byte 39 del mensaje de salida, por lo que, previo a estos datos, se debe generar una cadena de 39 ceros para las posiciones 0 a 38. Una vez se han unido estas dos partes, los siguientes dos bytes, el 41 y 42, son la cabecera. A partir de ahí, dependiendo del modo de funcionamiento cambia el significado del resto de posiciones, pero para las dos posibilidades con las que se está trabajando, los bytes 43, 44, 45 y 46 son la posición que debe alcanzar el vástago. Finalmente, para el caso del modo *PV Stream*, se puede seleccionar la velocidad con la que se desea alcanzar la nueva posición. Esta información se debe introducir en los bytes 47, 48, 49 y 50.

Por otro lado, antes de introducir la información en el mensaje completo, hay que dividirla en sus correspondientes bytes. Por ejemplo, la palabra de control en decimal puede ser 129 si se quiere salir de un estado de error, pero ese valor hay que separarlo en los bytes 39 y 40. En este ejemplo resulta muy sencillo ya que, al ser un valor inferior a 255, el byte más significativo es 0, y el menos significativo 129. Las cuatro partes del mensaje se introducen en notación decimal en el programa. Por ello, resulta necesario un algoritmo que separe automáticamente el dato en 2 o 4 bytes según corresponda. Para ello se definió el siguiente procedimiento en LabVIEW.

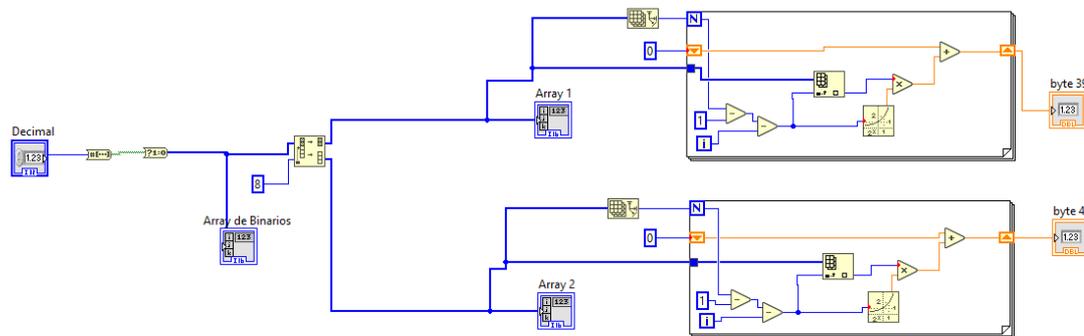


Figura 66: Algoritmo de separación en 2 bytes en LabVIEW.

En el sentido de las ideas que se han ido presentado, las cuatro partes principales del PDO de salida se introducen en una única variable decimal. Este valor se pasa a notación binaria en un primer paso ya que, el objetivo es obtener el valor que se debe enviar en cada byte del mensaje. En este ejemplo, se muestra el procedimiento para la palabra de control y la cabecera, ya que estas disponen de dos bytes de espacio. Por esta razón, los ocho bits menos significativos se separan en un primer array, de los ocho bits más significativos. Para la información de la posición y velocidad, que cuentan con 4 bytes, el procedimiento es el mismo, y se muestra en la siguiente imagen, en la que la única diferencia es que hay 4 grupos diferentes de ocho bits que ordenar.

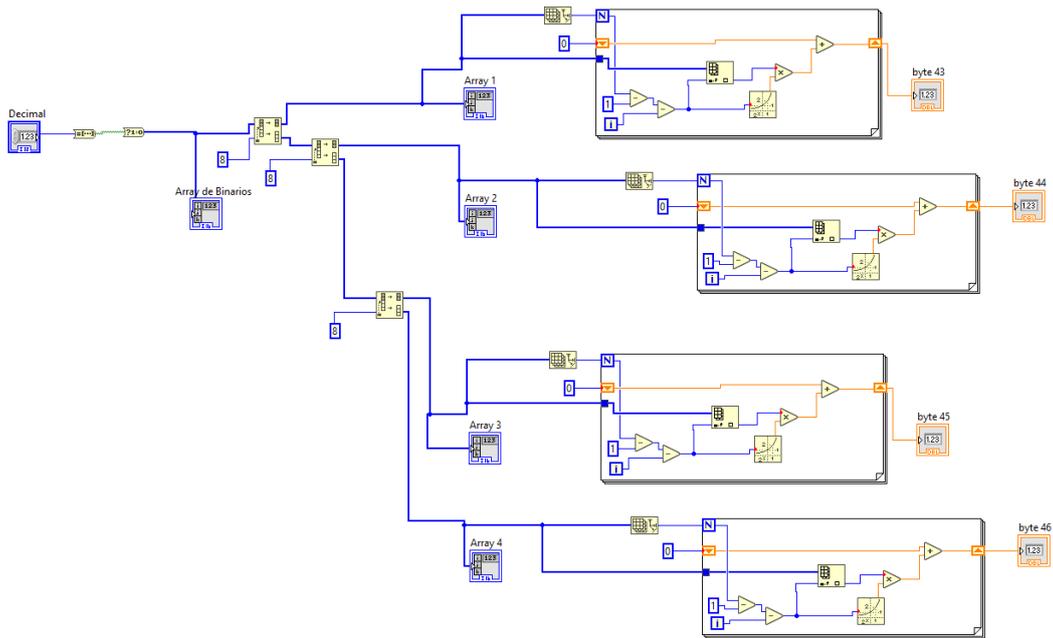


Figura 67: Algoritmo de separación en 4 bytes en LabVIEW.

Una vez que ya se dispone de los grupos de ocho bits separados, es decir, cada byte de forma independiente, se puede proceder a volver a pasar a notación decimal. Para ello, se ha propuesto un algoritmo que, en un bucle *for*, va multiplicando el valor del bit por 2 elevado a la posición del bit, siendo esta la técnica habitual empleada para pasar de notación binaria a decimal.

De este modo, se puede conseguir el dato que se debe enviar en las posiciones desde el byte 39 al 50, pero falta unir toda la información.

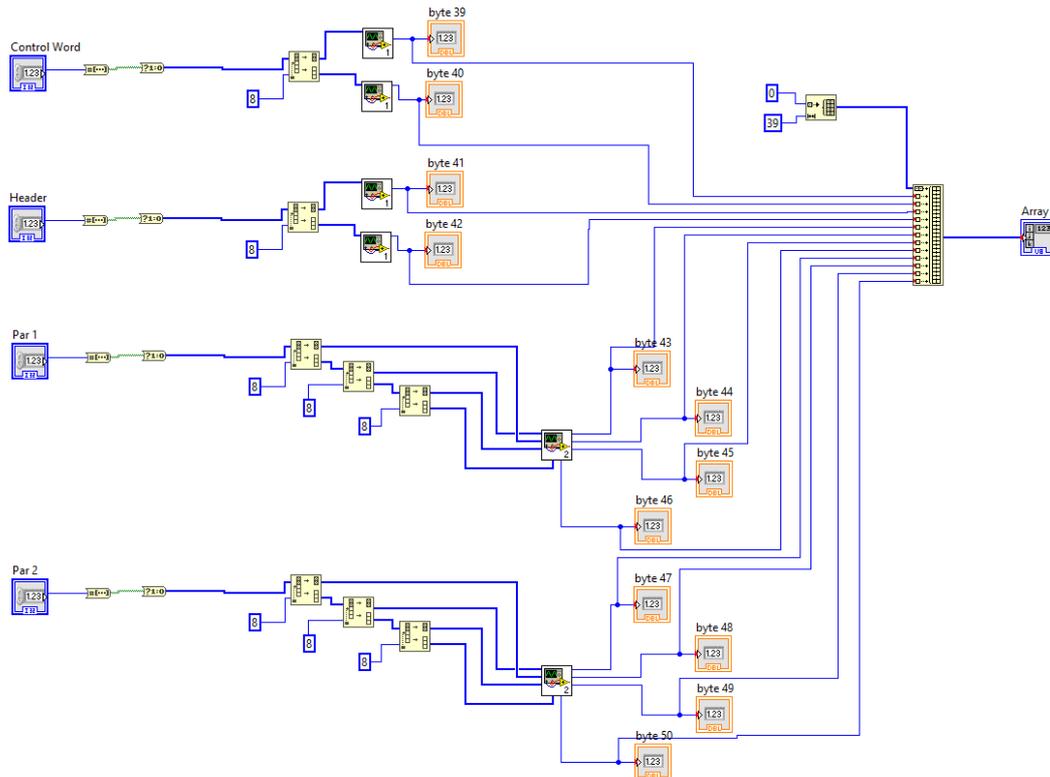


Figura 68: Generación del array con el PDO de salida en LabVIEW.

El procedimiento empleado es el que se muestra en la figura X, en la que se han definido algunos instrumentos virtuales para la simplificación del esquema, en la parte del algoritmo destinado a pasar de notación binaria a decimal.

Continuado con las explicaciones, es necesario definir en primer lugar una cadena de 39 posiciones, en las que se encuentre un valor nulo en cada posición, ya que la palabra de control comienza en la posición número 39 de la cadena. Por ello, y mediante un bloque de construcción de vectores (en adelante *arrays*), en primer lugar, se insertan los ceros de las posiciones 0 a 38, y luego los bytes 39 a 50 correspondientes al mensaje que se quiere enviar. Posteriormente, el bloque de la librería de EtherCAT se encargará de completar la cadena para que el controlador pueda interpretar correctamente el mensaje generado.

3.6 Programa para el control del motor lineal

En este apartado se describirá el programa diseñado en LabView con el objetivo de realizar las primeras pruebas de funcionamiento del motor. Partiendo de las ideas propuestas en los ejemplos incluidos en la librería de EtherCAT para LabView, se acondicionó el código para que pudiese ejecutar movimientos cíclicos del motor. También hubo que realizar múltiples ajustes para solventar algunas situaciones que se producían, ante las cuales el motor entraba en un estado de error.

En los siguientes párrafos, se presentará cada una de las partes en las que se secuenció la implementación realizada, y se comentarán las consideraciones que ha sido necesario tener en cuenta.

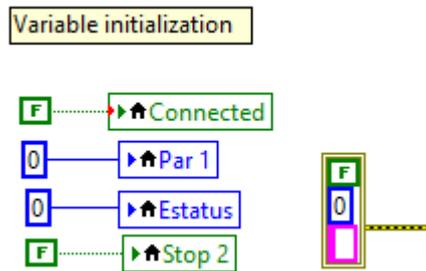


Figura 69: Inicialización de variables.

En primer lugar, se inicializan las variables locales del sistema. En concreto, se apaga el LED indicador de que la comunicación se ha establecido correctamente, y se asigna el valor 0, o falso en el caso del booleano, a las variables locales del sistema. Con Par 1 se hace referencia a la posición a la que se tiene que desplazar el vástago en cada iteración, el Estatus marca si el vástago está desplazándose hacia la derecha, o hacia la izquierda respecto al sistema de referencia establecido, y Stop 2 permite, desde el panel frontal, detener el movimiento del motor. Se han utilizado variables locales en estos tres últimos casos ya que, como se mostrará más adelante, el tramo principal del código se divide en dos bucles diferentes, que se ejecutan simultáneamente, y que se comunican a través de estas variables. Poniendo en falso la variable Connected, simplemente se paga el LED indicador de que se ha establecido la conexión con el controlador por si esta hubiese quedado encendida en la ejecución anterior.

También se crea un *cluster*, que consiste en la asociación de diferentes tipos de variables en un espacio común ya que existe una relación entre ellas en cuanto a uso que se les va a dar, compuesto por un booleano, iniciado en falso, un entero, iniciado en 0, y un array vacío, donde se almacenará posteriormente, y si corresponde, una advertencia de error, junto a su código y mensaje asociado. Es este el método que emplea la librería para controlar los errores.

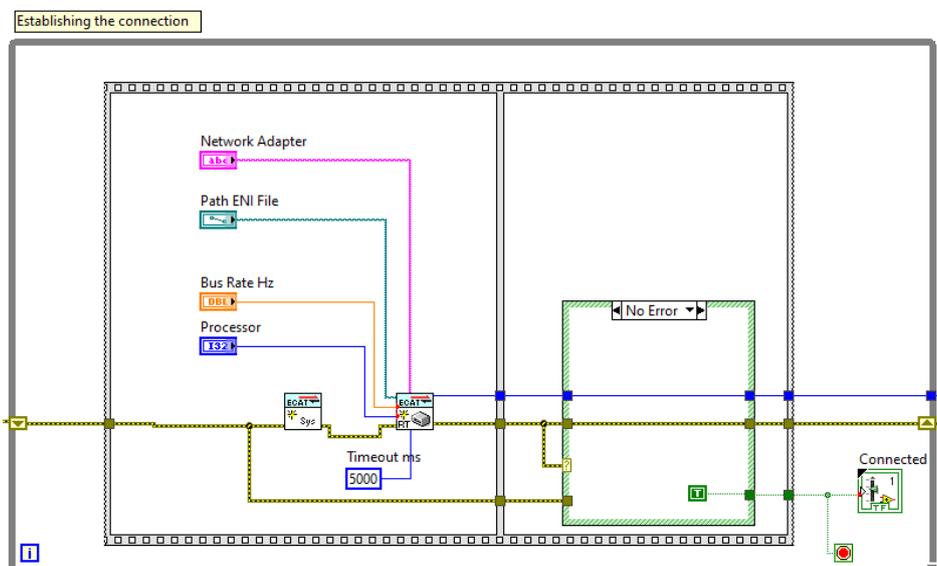


Figura 70: Establecimiento de la conexión con el controlador.

En segundo lugar, se encuentra el tramo de establecimiento de conexión, el cual se ha implementado sobre un bucle de tipo *while* debido a que es habitual que, tanto empleado los ejemplos de la librería, como el programa que se está presentando en este apartado, se presentase un error al tratar de establecer la conexión bajo la descripción de “Ethernet link. Cable disconnected”, lo cual se conseguía resolver tras tratar de iniciar el código de LabVIEW repetidamente, siendo esto de lo que se encarga el bucle, tratando de definir el maestro de la red hasta que lo consigue.

El primer bloque tan solo se encarga de activar los drivers necesarios, comprobando que en el ordenador se encuentra activa una licencia de uso de la librería.

Antes de continuar, conviene presentar el panel frontal del programa, ya que en esta sección se toman numerosos datos introducidos por el usuario.

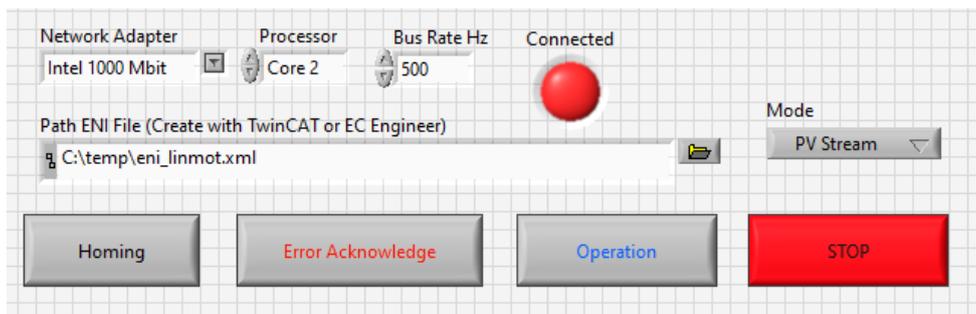


Figura 71: Panel frontal diseñado en LabVIEW.

Como entrada del bloque de apertura del maestro de la red EtherCAT, los parámetros de entrada son el adaptador de red y el núcleo del procesador empleado, la frecuencia a la que se enviarán los datos, y el archivo ENI, que incluye una descripción de la topología de la red, así como la definición de los PDO empleados.

El código de error pasa a través del bloque, y como salida se obtiene el código de referencia que se empleará posteriormente en el código. Además, se activa la variable que permite la salida del bucle y que enciende el LED indicador del panel frontal. Todo esto es en el caso de que no se produzca un error, ya que, en el caso contrario, lo que ocurrirá será que se mantendrá el *cluster* de error previo, pero se volverá a intentar establecer la conexión correctamente como se puede comprobar en la siguiente figura.

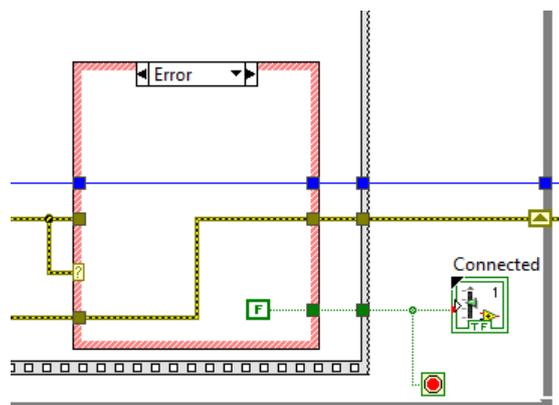


Figura 72: Bucle de intentos sucesivos de conexión.

Al resto de elementos del panel frontal se hará referencia en el momento en el que estos se presenten en el tramo del diagrama de bloques que se esté describiendo.

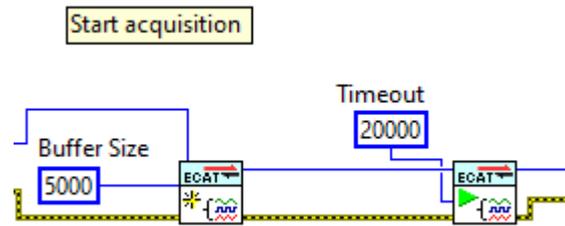


Figura 73: Inicialización de la adquisición de datos.

A continuación, se atraviesa un primer bloque en el que se crea el *buffer* para el intercambio de información, y un segundo bloque en el que inicia la adquisición, poniendo al maestro en el modo de operación, y dejando un tiempo de hasta 20 segundos para ello, lo cual es suficiente para que en ningún caso se produzca un error.

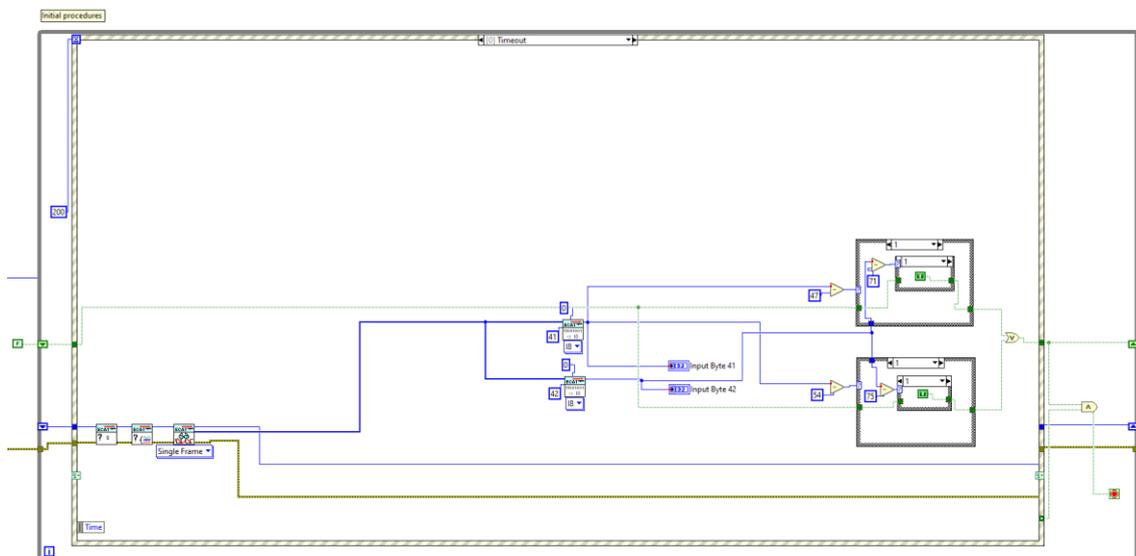


Figura 74: Comprobación de haber alcanzado la posición de home.

Se llega de este modo a una parte importante del programa, y en la que hay que tomar precauciones, debido a que ya resulta posible enviar mensajes con órdenes de movimiento al controlador, pero al mismo tiempo, no es conocida la posición en la que se encuentra el vástago, por lo que, antes de comenzar a realizar algún movimiento, hay que realizar el procedimiento de búsqueda de la posición de *home*.

En este mismo sentido de las ideas, se genera un bucle *while*, del cual no se puede salir hasta que se cumplan dos condiciones simultáneamente, que el motor esté en la posición de referencia, y que el usuario, a través del panel frontal, accione el botón *Operation*. Este, y el resto de los botones a los que se hará referencia, se pueden visualizar en la representación de la figura 71. De este modo, se ha definido una estructura de tipo evento, en la que, mediante una condición de tiempo, concretamente cada 200 ms, se entra en el caso mostrado en la última figura.

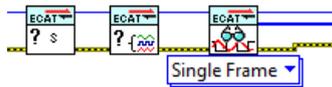


Figura 75: Bloques de consulta de información de los esclavos de la red.

En primer lugar, se pasa por este conjunto de tres bloques, donde el primero consulta el estado del esclavo de la topología de red, el segundo recopila la información relativa al *buffer* que se está empleando, y el tercero lee el PDO de entrada, en este caso en un *array* de 1 dimensión. En el caso de los dos primeros bloques, no se aprovechan los datos de salida que ofrecen, pero resulta necesario incluirlos en esta parte del código para su correcto funcionamiento.

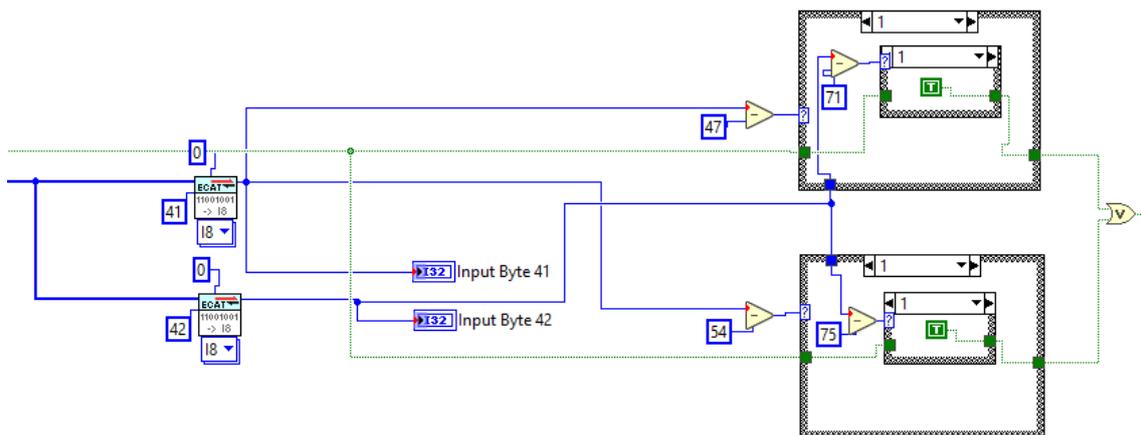


Figura 76: Reconocimiento de la posición de home.

En este caso, lo que se quiere comprobar es si el motor está en la posición de *home*, para lo cual hay que consultar la palabra de estado del PDO de entrada, presente en los bytes 41 y 42. Como se puede ver en la anterior imagen, son estos los bytes que se consultan, y se accede a ellos a través de unos bloques ofrecidos por la librería, en el que se puede indicar el byte, el bit de referencia, que será 0 en este caso, y el tipo de notación en la que se quiere recibir el dato, en este caso un entero procedente de un dato de 8 bits. Por último, se comprueba si los datos recogidos son (48,72) correspondiente al estado de *home* con modo de operación desactivado, o (55,76) que corresponde al estado de *home* con el modo de operación activado, siendo este último el que se alcanza de forma natural en el programa. La otra situación se puede dar únicamente si manualmente, mediante otro software como en LinMot-Talk, se cambia el estado de operación del motor.

Como los condicionales de las estructuras *case* comparan el valor recibido con 1 o 0, los valores anteriores se restan con el entero inmediatamente anterior, de modo que, si el resultado es 1, el número es el buscado. De este modo, cuando se alcanza la posición de *home*, se activa el booleano que permite la salida del bucle, el cual solo puede volver al estado de falso si se vuelve a realizar el procedimiento de búsqueda de la posición de referencia. Se puede acceder a este proceso mediante la interrupción de la estructura de eventos, de pulsar el botón *Homing*.

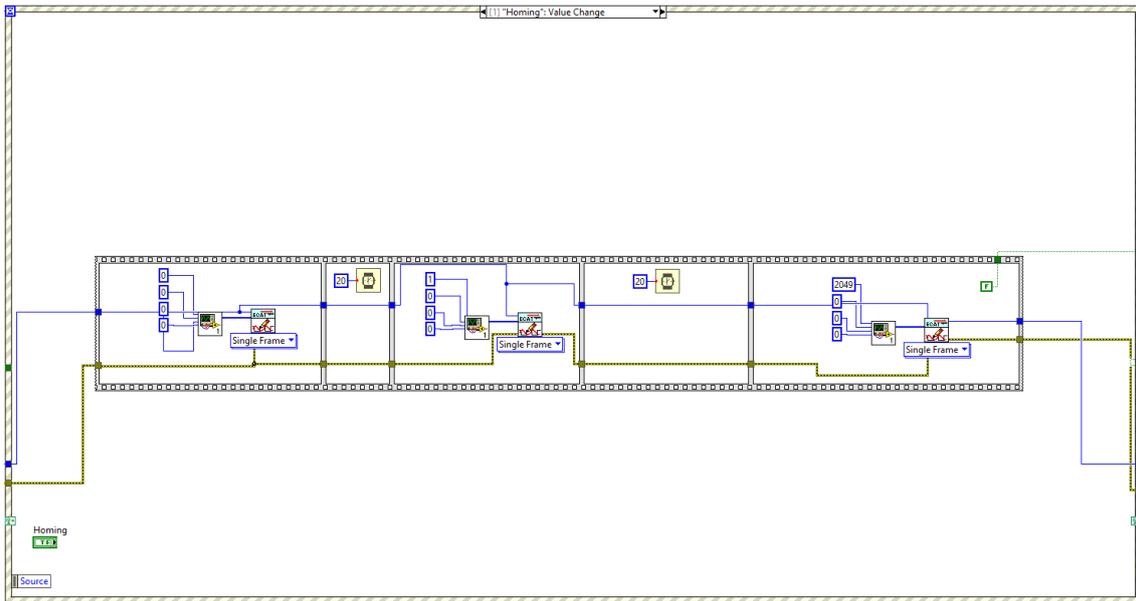


Figura 77: Secuencia de instrucciones para alcanzar el home.

El procedimiento consiste en tres comunicaciones con el controlador, separadas entre ellas 20 ms para evitar saturar el *buffer* de salida y que falle la comunicación. Ya en apartados anteriores se presentó el funcionamiento del bloque de envío de mensajes, por lo que, en este caso, únicamente se indicará el propósito de cada envío.

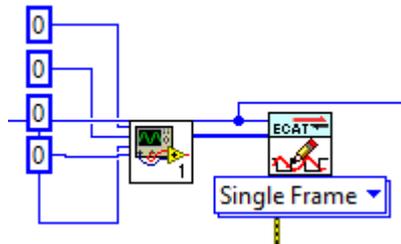


Figura 78: Envío inicial de mensaje al controlador.

El primer paso es enviar todos los valores a 0 para que el motor esté preparado para pasar al modo de operación, siendo esto último lo que se realiza enviando una palabra de control de 1 en la siguiente instrucción.

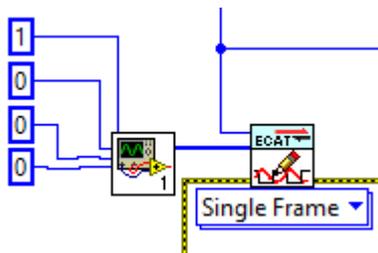


Figura 79: Puesta del motor en el modo de operación.

Tras los anterior, se puede realizar el procedimiento de *home*, manteniendo el controlador en el estado de operación, lo cual se consigue con una palabra de control de 2049 como se muestra en la siguiente representación.

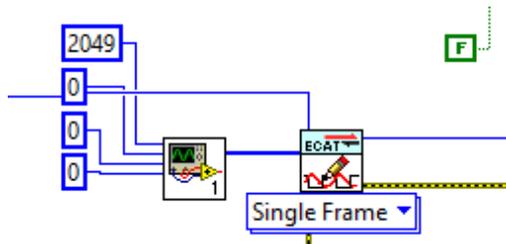


Figura 80: Inicio del proceso de búsqueda del home.

Como se puede apreciar en la última imagen, también aparece una constante de tipo falso, la cual se envía para desactivar la condición de que se ha alcanzado la posición de referencia. Esto es una protección ante la posibilidad de que el usuario, tras realizar el procedimiento, vuelva a pulsar en el botón *Homing*, lo cual alejaría el vástago de la posición de referencia, y por tanto hay que impedir que se pueda comenzar el control del movimiento del mismo.

Es posible que, debido a algún problema que se produjese en una ejecución del código previa, o ante algún otro tipo de contratiempo, el motor entre en un estado de error, impidiendo de ese modo que se pueda actuar sobre el controlador. De este modo, resultaba necesario incluir alguna forma de salir de dicho estado de error. Para ello, se ha definido otro caso de la estructura de eventos, consistente en pulsar el botón *Error Acknowledge*, el cual activa la siguiente secuencia.

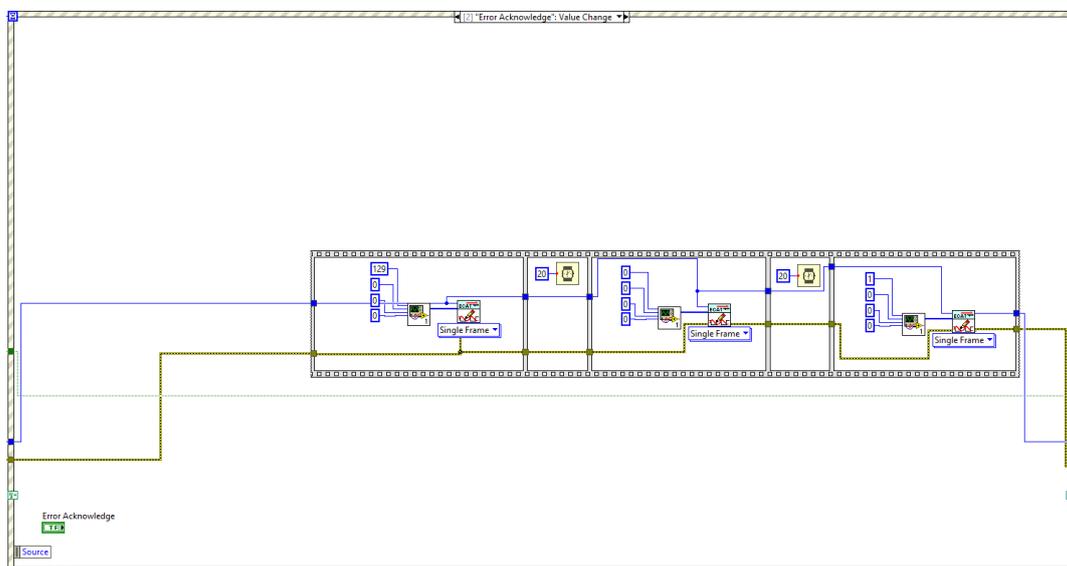


Figura 81: Secuencia para la supresión de errores.

El proceso es muy similar al descrito en los párrafos anteriores, en cuando a que se trata de tres envíos de palabras de control separados entre ellos por retardos de 20 ms.

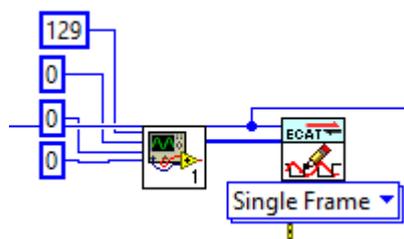


Figura 82: Indicar al controlador que el error es conocido.

En primer lugar, con el valor 129 se anula el error, manteniendo el estado en operación, y posteriormente se prepara al controlador para volver a entrar en operación, pero sin errores, enviando para ello la palabra de control 0.

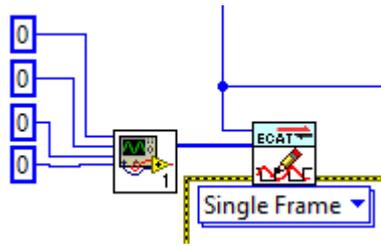


Figura 83: Salida del motor del modo de operación.

Finalmente, se vuelve a poner el motor en el estado de operación, en este caso ya sin errores, a excepción de que haya algún problema, como por ejemplo que no se esté alimentando el controlador con la tensión trifásica, que devolvería al sistema a un estado de error.

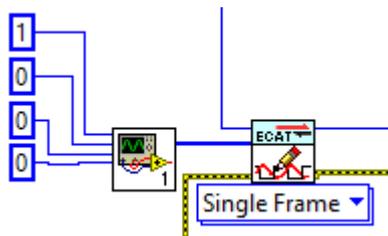


Figura 84: Puesta del motor en el modo de operación.

Finalmente, el último caso de la estructura, el cual se activa pulsando el botón *Operation*, únicamente manda un valor verdadero al booleano de salida, el cual se compara con el booleano de estar en la posición de *home*. De este modo, si se cumplen ambas condiciones se permite la salida del bucle.

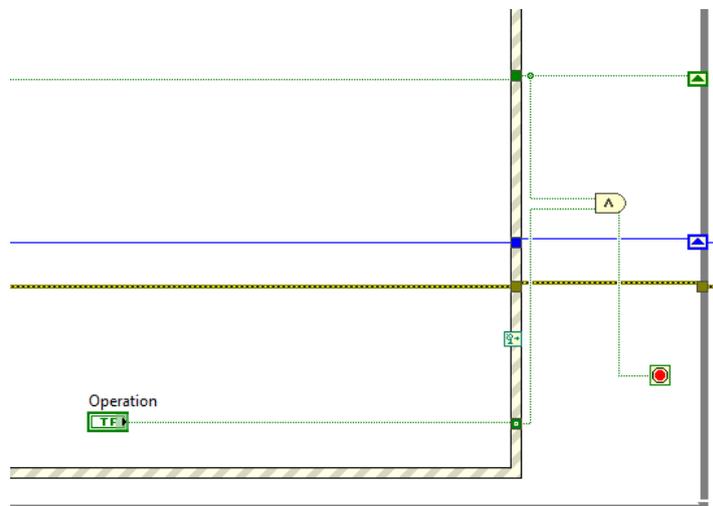


Figura 85: Salida del bucle y paso al modo de operación.

Con todo lo anterior realizado, se puede enviar una orden de poner el motor en el estado de operación, previo al envío de órdenes que se realizará más adelante, y antes de la entrada en

los bucles principales se pausa el programa 10 ms, tratando de sincronizar lo máximo posible la entrada en los dos bucles.

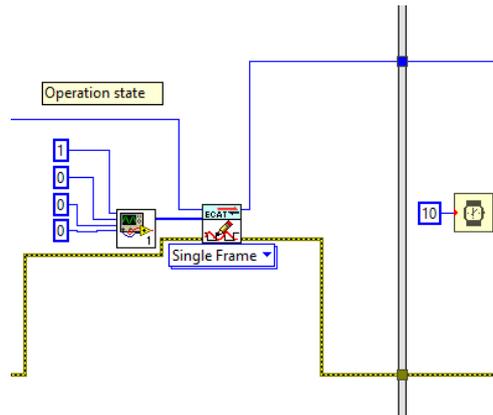


Figura 86: Puesta final del motor en el modo de operación.

Los dos bucles consisten en estructuras de tipo *while* temporizadas, las cuales se comunican entre ellas a través de variables locales.

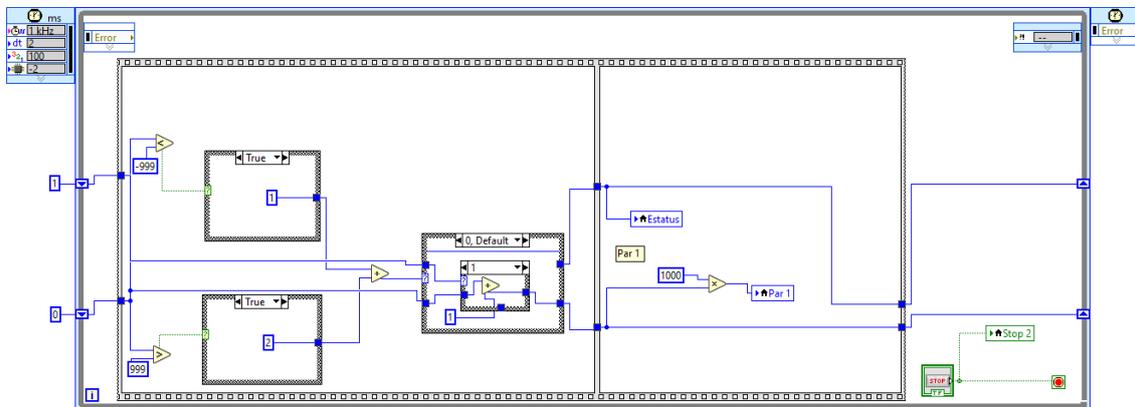


Figura 87: Bucle de cálculo de la posición del vástago en la siguiente iteración.

El bucle que se presenta en la figura 87, corresponde al encargado de generar la posición a la que se tiene que desplazar el vástago.

En cada iteración, se almacena en la variable local Par 1 la posición que debe alcanzar el vástago, y en la variable Estatus, un valor indicativo de si el motor está provocando un movimiento hacia la derecha o hacia la izquierda, lo cual interviene en la decisión del signo de la velocidad que se ha de indicar al emplear el modo de funcionamiento *PV Stream* tras seleccionarlo en el desplegable del panel frontal. También se almacena en la variable Stop 2, la señal que se genera al pulsar el botón STOP, de modo que, eso permita salir simultáneamente de los dos bucles.

Por otro lado, se ha querido separar el cálculo de la posición, con los procesos de comunicación entre el maestro y el esclavo, siendo esto último lo que se realiza en el segundo bucle.

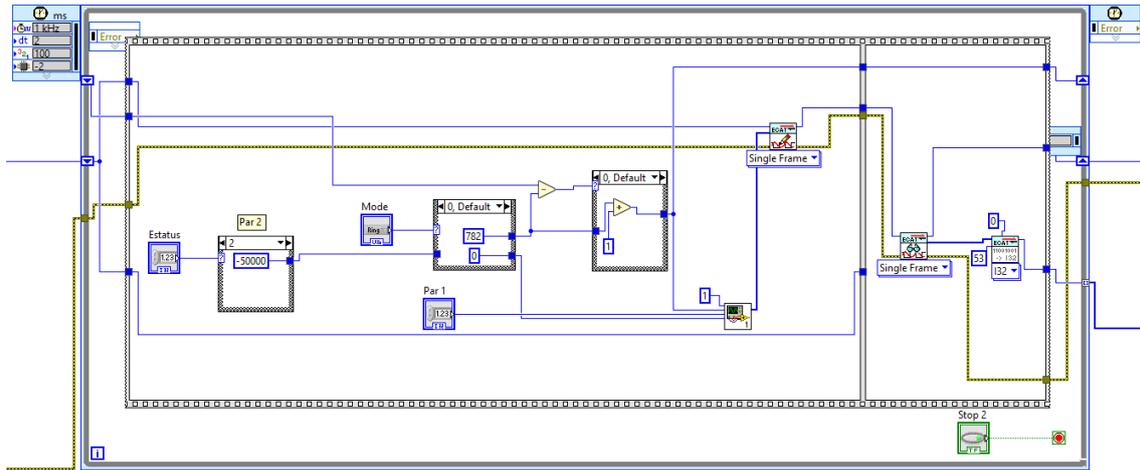


Figura 88: Bucle de envío y recepción de PDOs con el controlador.

El periodo de este segundo bloque tiene que ser de 2 ms como mínimo, coincidiendo con el límite establecido para los modos de funcionamiento con los que se está trabajando, mientras que el anterior se puede configurar al valor deseado, tomando el bucle de envío de datos, los valores que haya almacenados en las variables locales en cada momento.

La estructura mostrada en la figura X, consta de dos partes secuenciadas, de modo que, en primer lugar, se conforma el mensaje y se envía el PDO de salida, y posteriormente, se lee el PDO de entrada y se accede a los datos que se desean conocer.

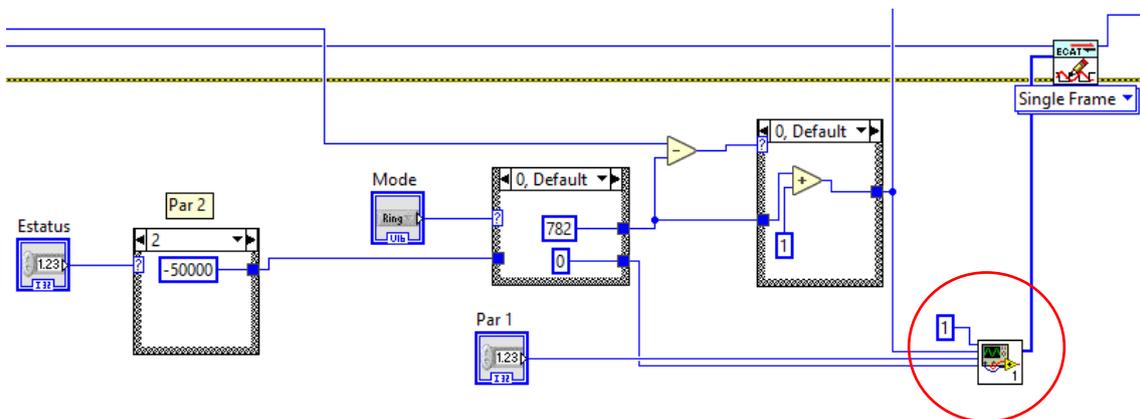


Figura 89: Conformado del mensaje de salida previo al envío al controlador.

En la zona marcada con la circunferencia roja de la anterior imagen, es donde realiza la conformación del mensaje, y posteriormente, este se transmite al bloque que realiza el envío del PDO al controlador.

En cuando a la creación del mensaje, la palabra de control siempre será 1, ya que se pretende mantener el motor en todo momento en el estado de operación. Por otra parte, mediante el desplegable *Mode* del panel frontal, se puede seleccionar trabajar con el modo *P Stream* o *PV Stream*. En el primer caso, la cabecera se alterna entre 782 y 783 en cada iteración, y en el segundo, se alterna entre 784 y 785. Esto se consigue empleando un *shift register*, que almacena el valor de la iteración anterior, para cambiarlo en la siguiente del modo en el que se muestra.

En relación con la velocidad, en el modo *P Stream*, que es el caso número 0, y por tanto el mostrado en la figura, se envía un 0, ya que es el controlador el que decide la velocidad a la que

se desplaza el vástago, mientras que en el modo *PV Stream*, se escoge la velocidad con la que se desea que el vástago llegue a la posición objetivo. En este ejemplo, como el vástago se desplaza 0.1 mm en cada iteración de 2 ms, se ha escogido la velocidad media que es de 0.05 m/s. Este valor se envía positivo cuando el vástago se está desplazando hacia la derecha, o negativo cuando el modo es 2, y por tanto se está desplazando a la izquierda, siendo este último caso el representado en la figura 89.

Tras finalizar este proceso, se procede a realizar la lectura del PDO de entrada, para lo cual se emplea el bloque de lectura, y los de conversión a entero que ya se presentaron en párrafos anteriores.

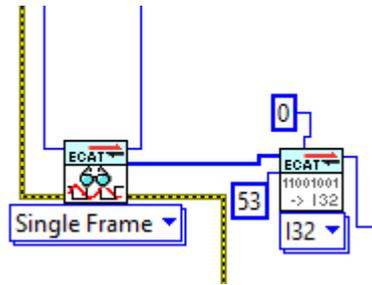


Figura 90: Bloque de lectura del PDO de entrada.

En este caso, el valor que se lee, es un entero de 32 bits, que comienza en el byte 53 del PDO, lo cual se corresponde con el valor de la corriente demandada en cada momento por el motor. Se escogió este valor para realizar pruebas con las que poder verificar que la lectura de datos funcionaba correctamente. Los valores recogidos en cada iteración se almacenan en un *indexing tunnel*, para de este modo, tras finalizar la ejecución poder almacenarlos en un array, y poder acceder a estos datos, por ejemplo, mediante un fichero de tipo Excel.

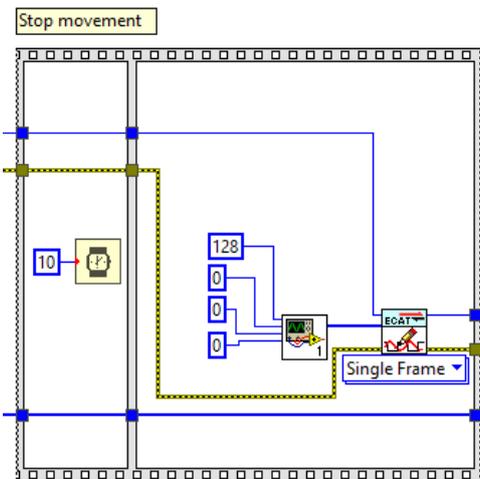


Figura 91: Anular error en el motor al finalizar la ejecución del software de LabVIEW.

Tras presionar, el botón STOP, se sale de ambos bucles simultáneamente, y tras un retardo de 10 ms, se envió un último mensaje con todos los elementos a 0, excepto la palabra de control, que es de 128, correspondiente a sacar al motor del estado de operación, y anular errores. Esto último se hace ya que, si durante la ejecución del programa, este no recibe los mensajes con suficiente velocidad, se activa una señal de error, y es fácil que esto ocurra al finalizar la

ejecución del programa. De este modo, se desactiva la señal de error, para no tener que anularla en la siguiente ejecución del programa.

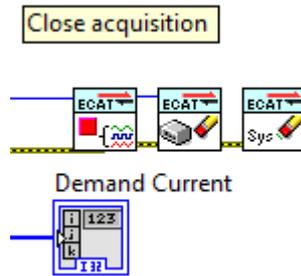


Figura 92: Finalización de la adquisición.

Finalmente, se cierra la comunicación mediante los tres bloques mostrados en esta última figura. El primero de ellos cierra el bus de comunicación, el segundo cierra el maestro y el tercero cierra el sistema EtherCAT.

Por otro lado, se muestra el array que se mencionó antes, para almacenar en este ejemplo, los valores de corriente demandada por el motor.

3.7 Descripción de un movimiento cíclico tipo rampa

A partir de los ejemplos incluidos en la librería para LabVIEW utilizada, se pudo comprender el funcionamiento de los distintos bloques que incluye, así como que la comunicación entre el ordenador y el controlado funcionaba correctamente. Durante la realización de las pruebas, se monitorizaba el estado del controlador mediante el software LinMot-Talk. Esto permitió, aparte de poder verificar que los mensajes enviados estaban siendo recibidos correctamente, poder conocer la razón de por qué en muchas ocasiones el motor entraba en un estado de error.

Para evitar posibles accidentes, los primeros ensayos se realizaron sin conectar la alimentación trifásica al motor. Esto permitía realizar las comprobaciones antes descritas, pero imposibilitaba que el motor se moviese. Además, al tratar de poner el motor lineal en el modo de operación, el controlador entraba en error ante la falta de esta alimentación. Por esta razón, una vez estaba comprobado el correcto funcionamiento de la transmisión de información, se alimentó el sistema con la tensión trifásica. De este modo, se pudo ver que para el programa desarrollado el motor entraba correctamente en el modo de operación, y realizaba el procedimiento para alcanzar la posición *home* sin problemas. Sin embargo, cuando posteriormente a estos procedimientos, se pretendía que el vástago se desplazase hacia una posición determinada, de nuevo se producía un error porque, según se indicaba en la ventana del LinMot-Talk, el controlador no estaba recibiendo los comandos de desplazamiento a suficiente velocidad.

Lo descrito en el párrafo anterior, se debía a que el controlador está preparado para trabajar recibiendo órdenes de movimiento de forma cíclica a gran velocidad. En concreto, para los modos de funcionamiento *P Stream* y *PV Stream*, se debe mandar un mensaje en un intervalo de tiempo de entre 2 y 5 ms. Además, por defecto viene configurado en el controlador, que se

produzca un error, si se pierde más de un mensaje cada milisegundo, como se puede ver en la siguiente imagen.

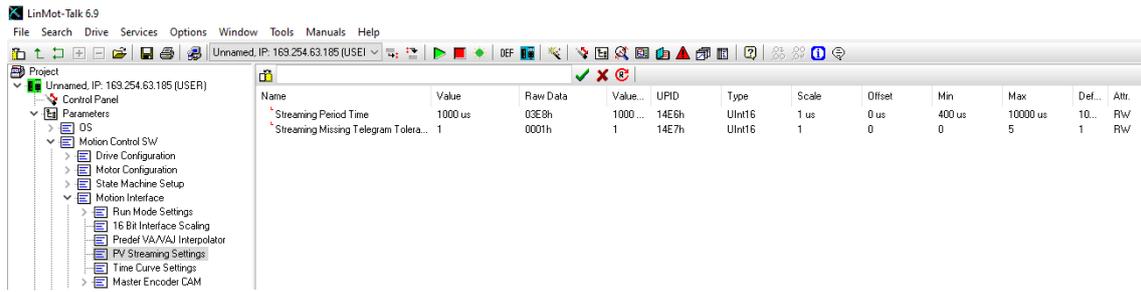


Figura 93: Panel de ajuste del modo PV Stream en LinMot-Talk.

Atendiendo a las razones presentadas, no resultaba posible realizar pruebas de desplazamiento con un programa preparado para enviar comandos de forma aislada, y sin un control de tiempo. Más adelante en el documento, se presentará el programa diseñado, en el que gracias a un bucle *while* temporizado, se puede garantizar el envío de información de forma cíclica en intervalos de tiempo que respetan las necesidades del sistema. Por otro lado, resultaba necesario idear algún método que generase las distintas posiciones que debía alcanzar el vástago en cada instante de tiempo. En este sentido, se decidió que un movimiento periódico, triangular, sería una forma adecuada de probar la respuesta del motor ante los distintos modos de funcionamiento que se pretendían probar en los ensayos.

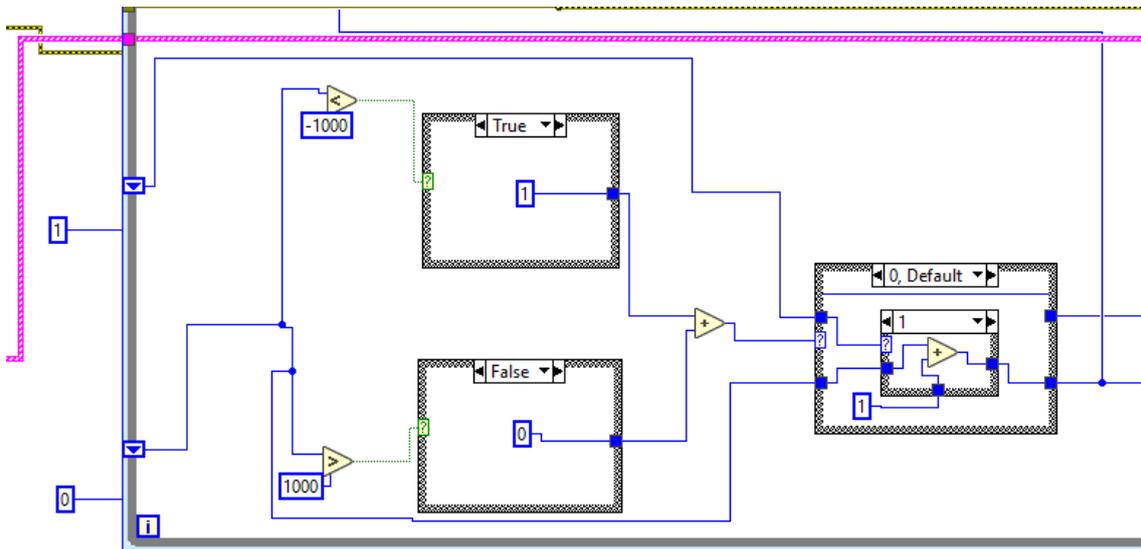


Figura 94: Generación de los valores de la curva triangular.

En la anterior imagen, se puede visualizar el método empleado para generar los valores de posición a alcanzar por el vástago. Inicialmente, se inicia la cuenta en 0, y se establece el estado del ciclo en aumentar el valor de la misma. En cada iteración, se comprueba si la cuenta ha superado el valor de límite superior, en el ejemplo 1000, o si la cuenta está por debajo del límite inferior, en la figura -1000. En el primer caso, el modo se pone en 2, es decir, reducir el valor, mientras que, en el segundo caso, se establece el modo 1, cuyo significado es aumentar el valor. En el caso en el que la cuenta se encuentra entre los valores límites, se mantiene el modo anterior. De esta forma, se consigue un contador que va de 0 a 1001, después se va reduciendo

hasta -1001, tras ello comienza a aumentar repitiéndose el ciclo constantemente. El valor de la cuenta se multiplica por 1000, con el objetivo de que cada iteración, la posición del vástago cambie en 0.1 mm, y este valor llega al bloque de envío de datos al controlador, concretamente se ubica en los bytes destinados a la información de la posición de destino del vástago.

Este sistema permite definir el tamaño del paso en cada iteración modificando el multiplicador a la salida del bucle, los límites superior e inferior del movimiento, actuando sobre los límites de la cuenta, y el número de pasos realizado en cada iteración, modificando el valor que aumenta la cuenta en cada ciclo del programa.

Con este archivo fue posible comenzar a realizar pruebas de movimiento del vástago, tanto para el modo *P Stream* como para el *PV Stream*. El modo de funcionamiento, en la aplicación a la que va destinada el motor será muy diferente, ya que la simulación requiere que las posiciones alcanzadas por el vástago sean las determinadas por un algoritmo que predice la posición a la que se mueve la catenaria debido a su interacción con el pantógrafo real, y a los factores físicos que intervienen en el modelo.

3.8 Otras curvas de movimiento

Anteriormente, se expuso el programa diseñado para poder realizar las primeras pruebas de funcionamiento del motor, describiendo para ello una curva triangular periódica, lo cual no resulta muy exigente para las capacidades de los equipos empleados. En este orden de los acontecimientos, tras verificar que el comportamiento observado en el vástago era correcto, a excepción de algunos problemas con el control del tiempo en la ejecución, los cuales se describen en el apartado 3.10, se actualizó el software para que este incluyese nuevos tipos de curvas con los que poder analizar el comportamiento del motor ante diferentes situaciones.

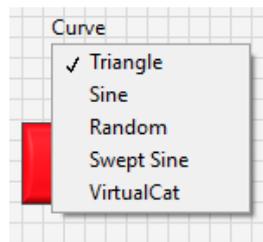


Figura 95: Selector de curvas de movimiento en el panel frontal.

En primer lugar, se incluyó una curva senoidal periódica, con rango desde 10 cm hasta -10 cm respecto a la referencia, siendo este el mismo rango que se empleaba en el caso de la señal triangular, y el que existe en todos los casos que se irán describiendo en los próximos párrafos.

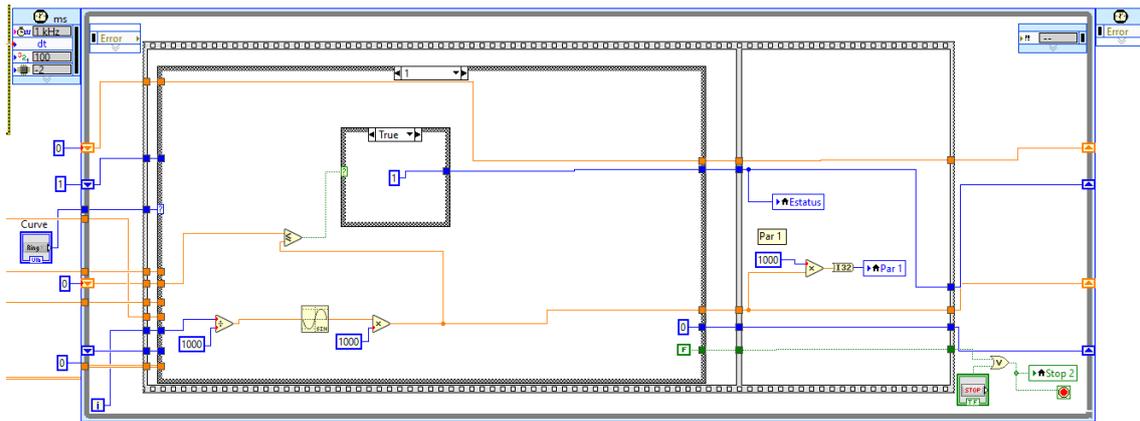


Figura 96: Generación de una curva de movimiento senoidal.

Comenzando por el caso de la señal senoidal, se aprovecha el contador ascendente del bucle while temporizado, y tras dividirlo por 1000, se calcula el seno. Tanto esta división, como la posterior multiplicación, también por 1000, sirven para ajustar el rango de valores generado al mismo que en el caso de la señal triangular, y que de ese modo cada paso del vástago fuese de 0.1 mm, y lo cual también permite que el código posterior se pudiese reutilizar.

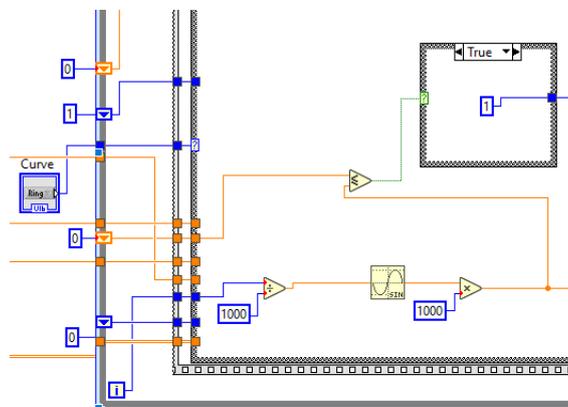


Figura 97: Zoom del cálculo de la posición en la curva senoidal.

Por otro lado, se compara el valor de posición actual, con el de la iteración anterior, asignado el valor 1 al estado, si se está desplazando a la derecha (aumenta el valor de posición), o un 2, si se está moviendo hacia la izquierda (disminuye el valor de posición).

Un aspecto que cabe destacar, es que cuando ya está generado el nuevo valor de posición, al haber un cálculo de un seno en el proceso, el resultado no es un número entero, sino uno real. Por ello, antes de realizar el envío a la variable local que almacena el dato de posición a enviar, se convierte a entero, lo cual es necesario ya que la resolución máxima a la que puede trabajar el equipo es de 0.1 μm , coincidiendo con el valor del bit menos significativo. Esta distancia es inapreciable por lo que, por lo menos para estos ensayos, no supone un problema. Este proceso se realiza para todos los casos que se van a describir, incluyendo el de la triangular, ya que como lo generado inicialmente es un número entero, el paso añadido no tiene ningún tipo de efecto sobre el valor.

Otra modificación que cabe destacar es que, en el caso de que se escoja el modo de funcionamiento *PV Stream*, se debe indicar la velocidad a la que se quiere alcanzar la posición objetivo. Una forma de que el desplazamiento del vástago en este modo sea suave, es indicar la velocidad media. En el caso de la señal triangular, es muy sencillo conocer este dato ya que, en cada iteración, que dura 2 ms, el paso es de 0.1 mm, por lo que la velocidad es de 0.05 m/s. Sin embargo, en el caso senoidal, y en algunos de los otros modos, este valor cambia en cada momento ya que, aunque se mantiene la frecuencia del envío de datos, no es constante el valor del desplazamiento. En estos casos, la velocidad media se puede obtener de un modo aproximado, pero suficientemente preciso, dividiendo el incremento de la posición del vástago entre los 2 ms de tiempo que dura cada iteración.

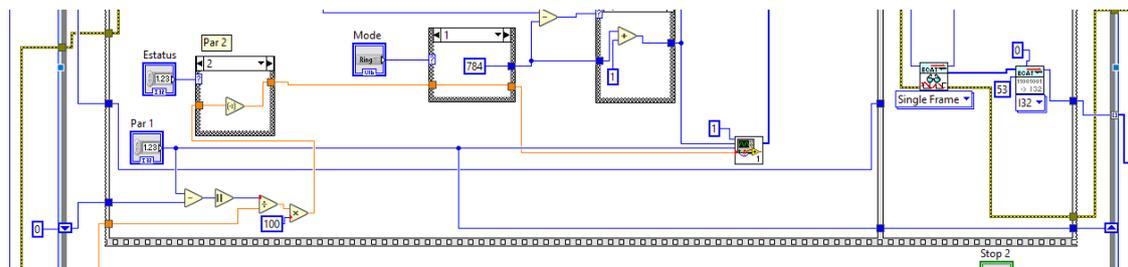


Figura 98: Cálculo automatizado de la velocidad media en el modo *PV Stream*.

Como se muestra en la anterior imagen, el proceso descrito se ha implementado mediante un *shift register* que permite comparar el valor actual con el anterior, y que, tras realizar el módulo, se puede realizar la operación descrita, y corriendo las unidades enviar el dato al controlador. Como se puede ver, cuando el estado es 2, el valor de velocidad se multiplica por -1 ya que el vástago se está desplazando en sentido negativo.

Con esta curva se puede estudiar con más detalle, la influencia de los cambios en la velocidad de desplazamiento, así como el efecto al realizar cambios de sentido a velocidad nula en los extremos.

Continuado con el resto de las curvas de movimiento incorporadas al programa, los cambios únicamente se encuentran en la secuencia de cálculo de la nueva posición del bucle temporizado secundario, siendo esta la razón de que las explicaciones desde este punto se centren en dicha parte del código.

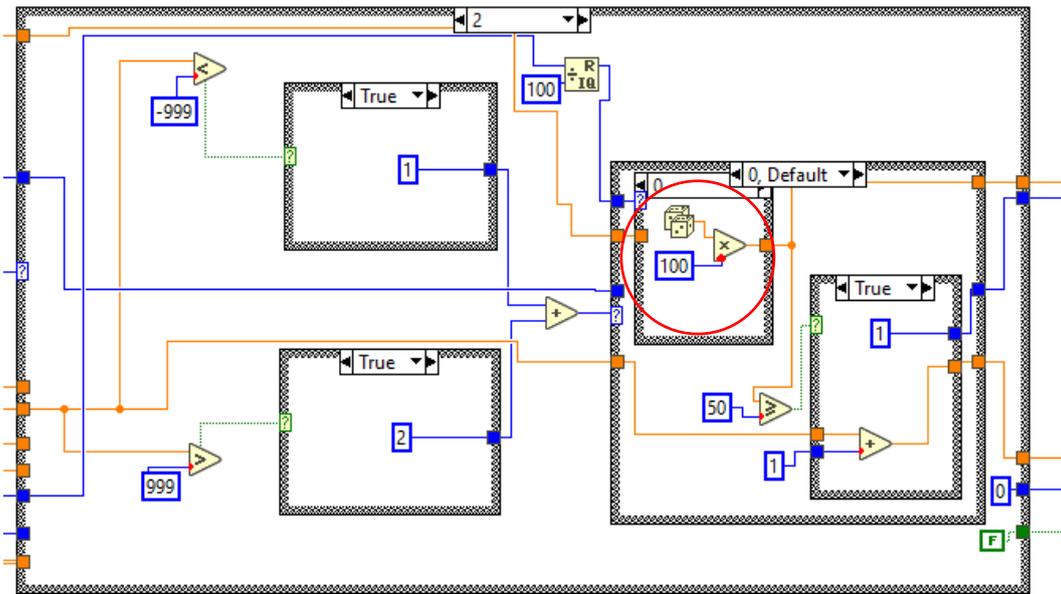


Figura 99: Cálculo de la curva de movimiento aleatoria.

La tercera curva de movimiento resulta bastante diferente a las anteriores ya que incorpora una componente de aleatoriedad. En este caso, se genera un número aleatorio entre 0 y 100 en la zona rodeada en rojo. Si este número es mayor o igual a 50, se desplaza el vástago hacia la derecha, y si es inferior a 50 hacia la izquierda. Como cada paso es de únicamente 0.1 mm, lo cual es muy poco para apreciarlo bien a simple vista, lo que se hace es que una vez que se decide hacia qué lado desplazar el vástago, las siguientes 99 iteraciones se mueve el eje 0.1 mm en el mismo sentido. De este modo se consigue que cada desplazamiento sea como mínimo de 1 cm. Una vez recorrida esa distancia, aleatoriamente irá 1 cm más en el mismo sentido o en el contrario. Junto a lo anterior, se establecen los límites de la posición a ± 10 cm. De este modo, cuando el vástago alcanza una de estas posiciones, en lugar de tomar la decisión de forma aleatoria, se impone que el siguiente centímetro sea hacia la posición de referencia, alejándose de este modo del borde.

En el caso de esta curva, se incrementan notoriamente los cambios bruscos de sentido, siendo esta la principal característica que se puede destacar, y la cual se asemeja un poco más a lo que ocurre al simular el comportamiento de una catenaria.

Continuando con las siguientes curvas implementadas, la cuarta propuesta es una senoidal, idéntica en cuanto a amplitud al segundo caso, pero que aumentase de frecuencia con cada iteración.

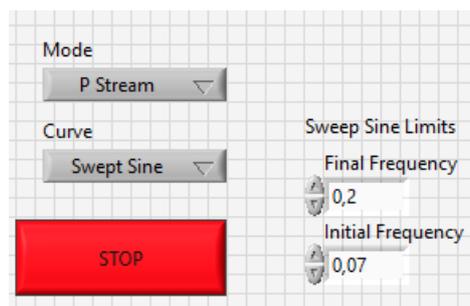


Figura 100: Añadidos al panel frontal para la curva de barrido en frecuencia.

En la figura 95, se muestra como en el panel frontal se puede seleccionar esta curva, denominada *Swept Sine*, y en la parte derecha se muestran los controles con los que se puede seleccionar la frecuencia al inicio del movimiento, y con la que finalizará el ensayo. En este caso, el programa termina tras un tiempo, el cual se ha establecido en 3 minutos. Por esta razón, internamente se calcula el incremento de la frecuencia necesario en cada iteración para cumplir con este propósito. Este proceso se puede ver en la siguiente imagen, en la que además se puede apreciar que los datos que finalmente se emplearán en el bucle temporizado secundario son los límites de frecuencia, y el incremento a aplicar en cada iteración.

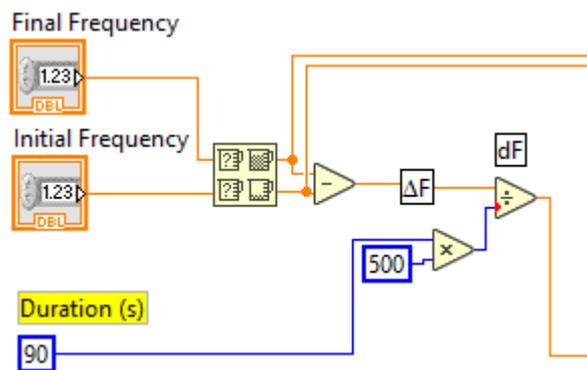


Figura 101: Cálculos previos de ajuste del barrido en frecuencia.

De este modo, resulta necesario calcular la frecuencia en cada iteración, y posteriormente se calcula la siguiente expresión:

$$Pos = \sin(2 \cdot \pi \cdot (f_0 + n\Delta f) \cdot n \cdot 2ms) \quad \text{donde } n \text{ es la iteración.}$$

El resto de pasos aplicados coinciden con lo que se indicó para el caso de la senoidal, pero en este caso, el barrido en frecuencias sería de gran utilidad para poder realizar un estudio de caracterización del motor, siendo este también uno de los objetivos que perseguía el presente Trabajo de Fin de Máster.

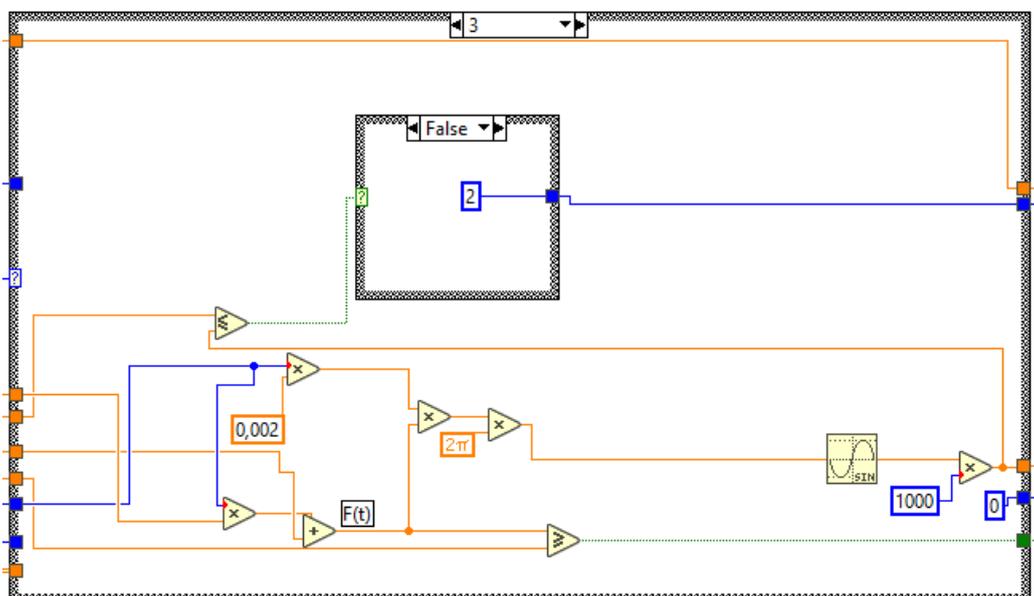


Figura 102: Cálculo de la curva de barrido en frecuencia.

Por último, y dado que el objetivo final es incorporar el motor al montaje principal, simulando el movimiento de la catenaria a partir del modelo matemático al que se hizo referencia en el capítulo de introducción, que tiene en cuenta los fenómenos físicos debidos al punto de la catenaria respecto a los apoyos, y a la interacción entre esta y el pantógrafo. A partir de este modelo, se generó una secuencia de posiciones correspondiente a una catenaria común, y de pequeña longitud, ya que únicamente se pretendía realizar un ensayo rápido en el que apreciar la respuesta del motor ante una secuencia de punto similar a la aplicación final.

A partir de un código en Matlab en el que está implementado el código con la ecuación que define las posiciones de la catenaria, se generó una secuencia de puntos, en la que cada uno correspondía a la posición a la que se tenía que desplazar el vástago cada milisegundo, y con una duración total de algo más de 4 segundos.

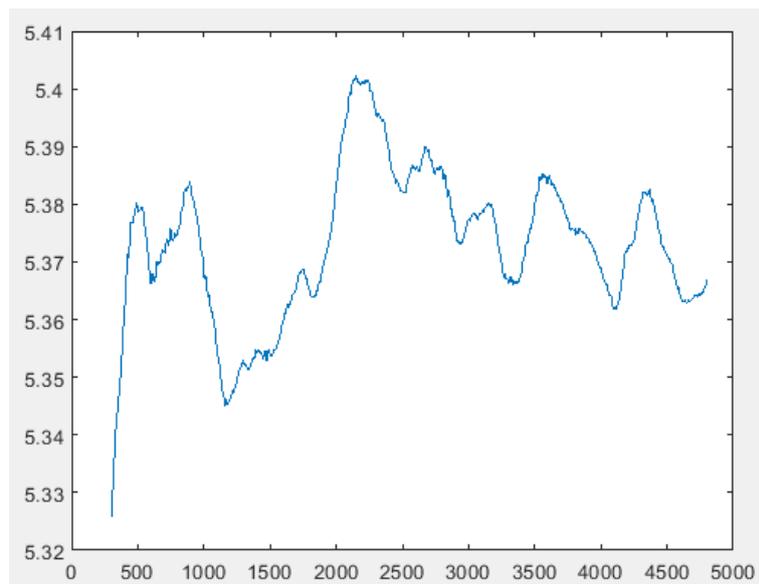


Figura 103: Curva de muestra de una catenaria.

Para poder adaptar estos datos al programa de LabVIEW, en primer lugar, se eliminó una parte inicial de datos nulos, y también los primeros datos de la secuencia hasta que el primer valor y el último coincidiesen. El objetivo de esto último es poder reproducir cíclicamente la curva, y que por tanto el ensayo no termine hasta que el usuario presione el botón Stop del panel frontal.

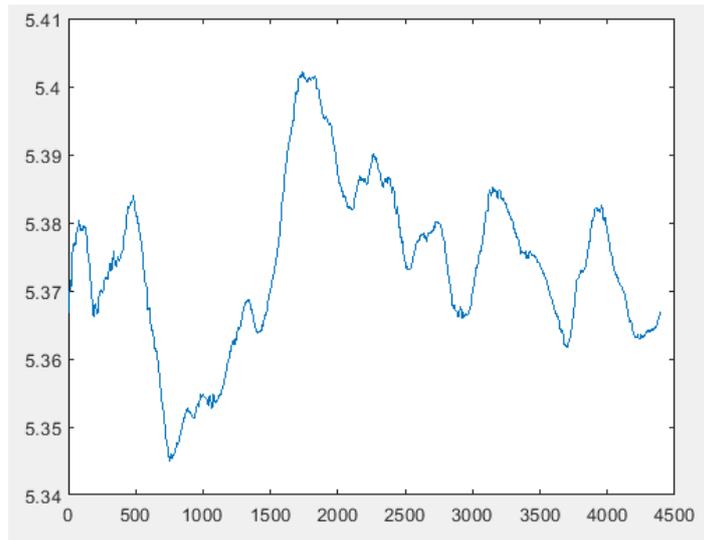


Figura 104: Ajuste de la curva de muestra de una catenaria para poder reproducirla cíclicamente.

Lo siguiente fue restar a todos los puntos el valor de la primera posición, de modo que el punto inicial fuese 0, y el movimiento partiese desde ese punto. También se multiplican los datos por 1000, después de la resta, para que las unidades en las que están los datos se correspondan con lo que espera el bucle implementado en LabVIEW.

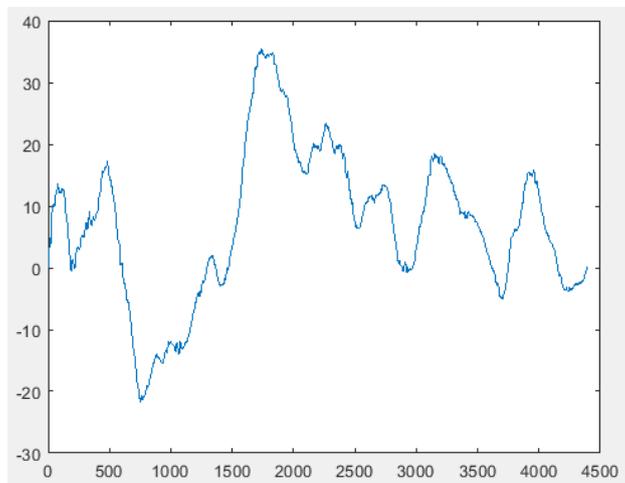


Figura 105: Modulación de los valores de la curva de muestra de una catenaria.

Estos datos se exportaron a Excel, donde se eliminó uno de cada dos datos, ya que la simulación envía una posición objetivo cada 2 ms, mientras que los datos que se habían generado estaban pensados para una iteración cada milisegundo. Al guardar el archivo, al efecto de que posteriormente el programa LabVIEW pueda interpretar correctamente los datos, se debe guardar como *Texto (delimitado por tabulaciones)*, y posteriormente se debe cambiar la extensión del archivo de .txt a .lvm.

La función de LabView que lee este tipo de ficheros es el *Read From Measurement File*. En este caso, se configura del modo en el que se muestra a continuación.

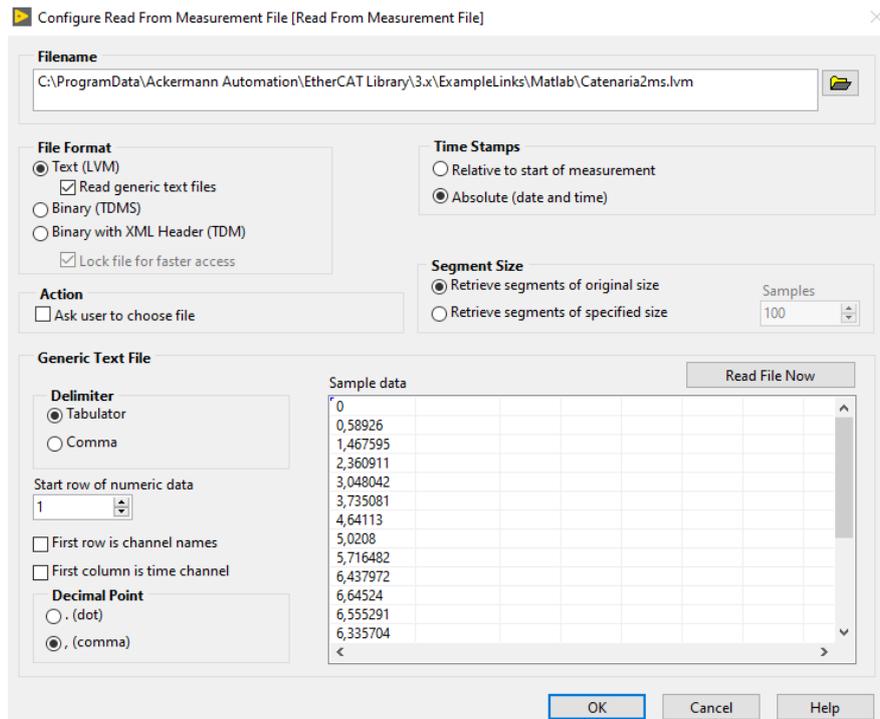


Figura 106: Bloque de lectura de hoja de cálculo en LabVIEW.

En cuanto al diagrama de bloques, se realiza la lectura en un bucle While, hasta que se cumpla la condición de haber finalizado la lectura del fichero (EOF manda una señal True). Se convierten los datos de tipo dinámico a numérico, y se van uniendo al array para posteriormente trabajar con el conjunto de datos.

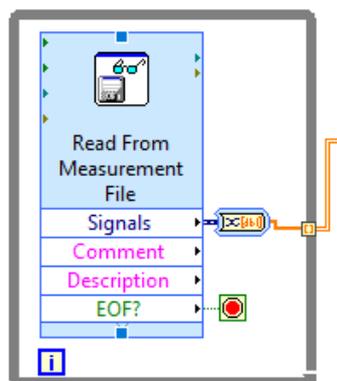


Figura 107: Vista desde el diagrama de bloques del bucle de lectura de hoja de cálculo en LabVIEW.

En cuanto al bucle temporizado secundario, el funcionamiento es el que se muestra en la imagen que sigue a este párrafo. En la parte superior, como en todos los casos, se asigna el valor del estado, según si el vástago se está desplazando hacia la derecha o hacia la izquierda. En la parte inferior, llega la cadena de valores, la cual se debe ir leyendo cíclicamente, es decir, cuando se llegue al último dato, se repite de nuevo la secuencia. Por esta razón, en primer lugar, se pasa la cadena por un bloque que devuelve la información de las dimensiones (filas y columnas), entramos de este modo a la posición 1, donde se pueden conocer las filas. Si la posición a la que estamos accediendo es inferior a la última, accedemos a la posición siguiente de la cadena en la

columna 0, mientras que, si se alcanza la última posición, se vuelve a poner la cuenta a cero como se muestra en la figura 109.

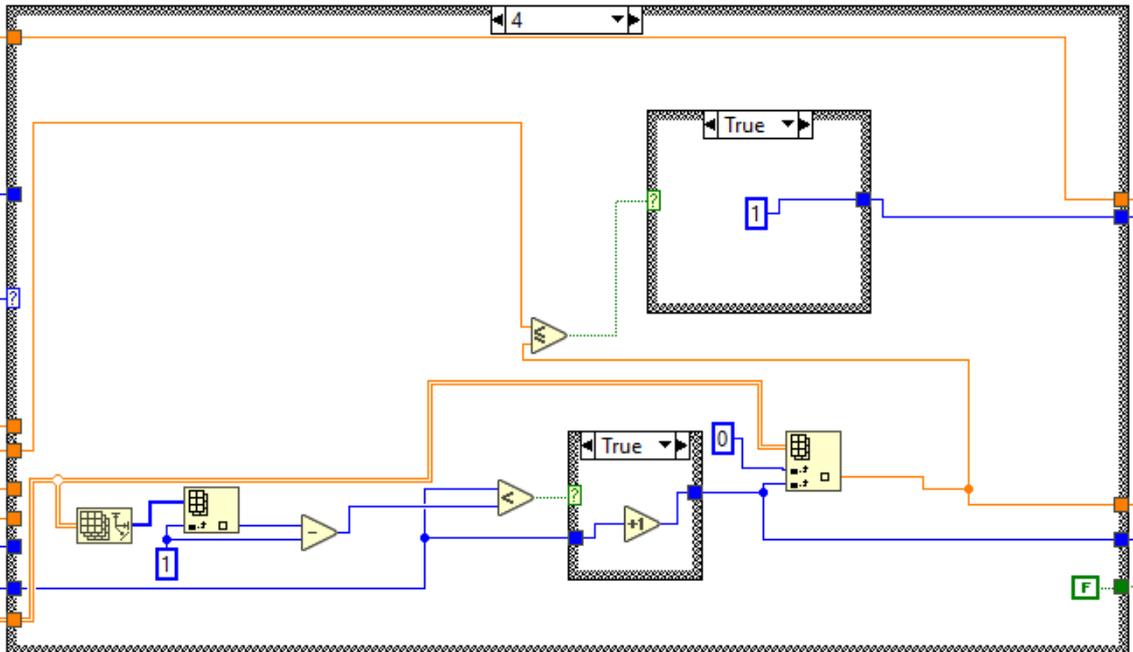


Figura 108: Control de los límites del rango de movimiento en LabVIEW.

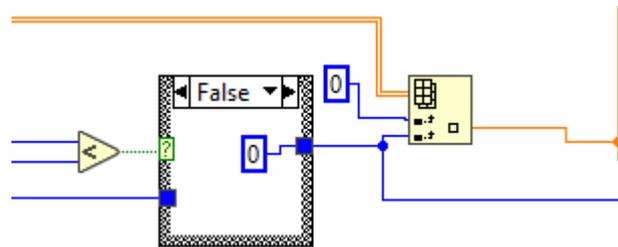


Figura 109: Puesta a cero de la cuenta al llegar al último dato del modelo de catenaria.

De este modo, se puede simular el comportamiento que tendría un tramo genérico de catenaria, y ver, por tanto, la respuesta que ofrece el motor lineal empleado.

Resulta complicado plasmar en un texto el comportamiento mecánico del motor. Por ello, será en la presentación de la defensa el Trabajo de Fin de Máster en la que se presentarán videos que muestren el sistema en funcionamiento. En la siguiente imagen, se puede ver el carrito en la posición de referencia.

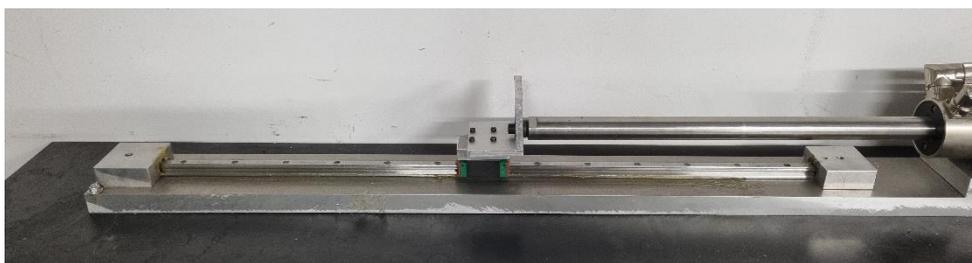


Figura 110: Motor lineal en la posición de referencia.

Cada curva de movimiento es diferente, pero siempre se describe un movimiento oscilatorio en el que se alcanza el límite hacia el sentido positivo (la derecha), y hacia el sentido negativo (la izquierda). En las siguientes dos fotografías se muestran estos estados.



Figura 111: Motor lineal en el límite derecho del rango de movimiento.



Figura 112: Motor lineal en el límite izquierdo del rango de movimiento.

3.9 Elementos adicionales del software

En el apartado 3.6, se expuso el programa elaborado en LabVIEW para el ensayo del motor. Posteriormente, se presentaron las distintas curvas que se añadieron como posibilidad para seleccionar en el panel frontal. En este caso, existen algunas modificaciones que se realizaron posteriormente, principalmente para evitar fallos de funcionamiento.

Un problema que se presentó en el transcurso de la realización de este Trabajo de Fin de Máster, es que, de forma aleatoria, aparecía un ruido durante el movimiento del vástago, el cual sin ninguna explicación aparente desaparecía, y volvía a producirse. Posteriormente, se descubrió que la causa era que no se estaba seleccionando correctamente la frecuencia del bus de comunicación al establecer la conexión con el esclavo. Esto se debía a que, internamente en el programa, se modificaba manualmente el periodo del ciclo de envío y recepción de los PDO, y existía por tanto la posibilidad de que no coincidiese con el indicado en el panel frontal.

Ante esta situación, se sustituyó el control del panel frontal en el que se indicaba la frecuencia, por uno en el que se introdujese el periodo. De este modo, este dato se introduciría en los bucles temporizados directamente, y calculando la inversa y corrigiendo las unidades, se podía conocer la frecuencia, siendo este dato el que se incorporaba en el establecimiento de la conexión con el esclavo.

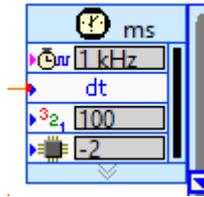


Figura 113: Cuadro de ajuste del bucle temporizado en CompactRIO.

Como se puede apreciar en la anterior imagen, ya no se muestra el periodo del ciclo introducido manualmente, sino que aparece el término dt que hace referencia a que ese dato se toma como una entrada, la cual, en este caso, se introduce en el panel frontal.

Otra característica muy importante, y ciertamente relacionada con lo que se está comentando, es que el periodo mínimo con el que se puede trabajar empleando el bus de campo EtherCAT con los equipos descritos, es de 2 ms. Ante esto, y para evitar seleccionar un valor incorrecto desde el panel frontal, se realiza una comprobación durante la ejecución para poder conocer si se incumple esta condición.

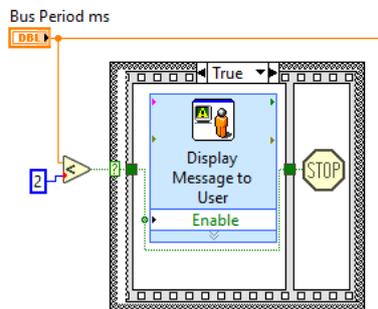


Figura 114: Cuadro de advertencia de error de introducción de velocidad de comunicación muy baja.

Como se puede apreciar, si el valor de periodo seleccionado es inferior a 2 ms, se muestra un mensaje por pantalla avisando al usuario, y posteriormente se suspende la ejecución del programa.

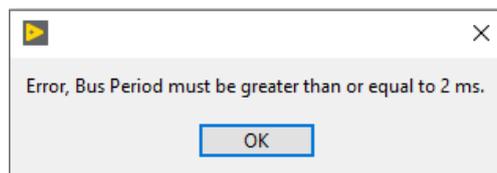


Figura 115: Ventana emergente de error ante velocidad de comunicación muy baja.

3.10 Incorporación del CompactRio

El compactRIO es un controlador del fabricante National Instruments que trabaja en tiempo real, y que cuenta con módulos de entrada y salida configurables, FPGA, puertos ethernet, y con la posibilidad de añadirle módulos adicionales según las necesidades.

Tal y como se ha descrito a lo largo de todo el documento, las sucesivas versiones del software de control del movimiento del motor lineal, permitieron describir correctamente las curvas

deseadas. No obstante, si se podían apreciar ciertos problemas, los cuales alteraban ligeramente el ensayo, pero de forma claramente apreciable.

Profundizando un poco más en lo expuesto en el párrafo anterior, durante el movimiento cíclico del motor, en cualquiera de las variantes de curvas implementadas, se reconocían pequeños saltos de posición incorrectos. Estos se podían identificar principalmente a través del sonido, ya que este, en vez de ser constante, se producían golpes similares a los que reproducía el vástago al realizar un cambio de sentido de movimiento en los extremos.

Por otro lado, y como se reveló posteriormente, a causa del mismo problema, al iniciar el programa, en cualquiera de las variantes de movimiento seleccionables desde el panel frontal de LabView, en algunas ocasiones, el movimiento se detenía, o tras un corto periodo de tiempo se llegaba a esta misma situación. En ambos casos, se encendía el indicador luminoso identificativo de error del motor. Dentro de un apartado disponible dentro del software LinMot-Talk, se puede acceder a un registro de los estados de error en los que ha entrado el motor, junto a una breve descripción. En este caso, el motivo de que se detuviese el movimiento es que el envío constante de mensajes de movimiento estaba fallando. En concreto, no llegaban suficientes órdenes de movimiento. El registro que toma el programa como referencia viene establecido por defecto en una pérdida de mensaje por cada 2000 μ s, es decir 2ms.

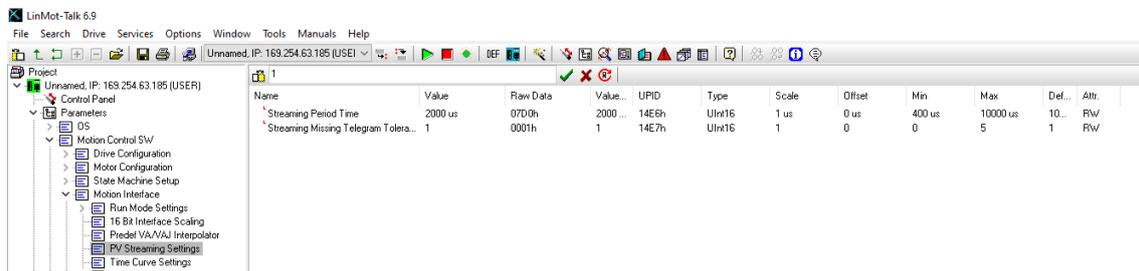


Figura 116: Cuadro de ajuste del número máximo admisible de mensajes perdidos.

La interpretación que se realizó de los problemas expuestos es que se estaban produciendo fallos puntuales en la comunicación. A objeto de resolver estos inconvenientes, en primer lugar, se ajustó correctamente la velocidad de comunicación indicada en el establecimiento de la comunicación con el controlador. La razón que explicaba que no coincidiese este parámetro, con la frecuencia de envío de datos real, era que estos dos parámetros se encontraban desagregados dentro del software, de modo que, al realizar pruebas a las diferentes velocidades de comunicación, ya que, para altas velocidades, el ordenador parecía no poder responder a la exigencia computacional, y a muy bajas velocidades el motor se detenía, ya que el mínimo exigido por el bus de comunicación es una orden cada 10 ms. Esto se corrigió ajustando ambos parámetros a través de la misma entrada de datos ubicada en el panel frontal del software. Por ejemplo, para la velocidad más alta de comunicación que permite el hardware, se indica un tiempo de 4 ms, el cual se introduce como dato para los bucles principal y secundario de funcionamiento, así como se calcula la frecuencia que se debe indicar en el establecimiento de la comunicación con el controlador, la cual es de 250 Hz.

En el mismo sentido de las ideas que se han ido presentado, el ordenador parecía no poder trabajar a un periodo de comunicación de 4 ms, ya que, a pesar de que no se consumían los recursos hardware del ordenador, al realizar cualquier interacción con el mismo, como mover el

ratón, durante la ejecución del programa, en muchas ocasiones se detenía el programa. Por otro lado, la ejecución del software se estaba realizando en un sistema operativo Windows, el cual ejecuta numerosas tareas en segundo plano, las cuales podían estar interfiriendo con los objetivos de comunicación en tiempo real que se deseaban alcanzar.

Ante la situación expresada, se tantearon diferentes opciones para evitar estos problemas, y dado que para el sistema principal de catenaria virtual se estaba empleando un equipo CompactRIO, se decidió probar esta alternativa. En relación a las situaciones planteadas anteriormente, este equipo permite trabajar en tiempo real, y sin que se ejecuten procesos en segundo plano ajenos al programa implementado en LabVIEW.

Como se puede interpretar a partir de lo mencionado, ya existía un sistema para la ejecución de un software LabView desde CompactRIO dentro del proyecto de simulación de la catenaria virtual. Para utilizarlo, se deben instalar una serie de añadidos y librerías de EtherCAT a LabView.

De este modo, y tal y como se muestra en la siguiente figura, se puede acceder a las etiquetas que permiten acceder a los datos contenidos en el PDO de entrada, o enviar al PDO de salida. Específicamente, estos se encuentran dentro del CompactRIO, Maestro EtherCAT, y el dispositivo conectado, que en este caso es el controlador del motor, identificado como dispositivo número 2.

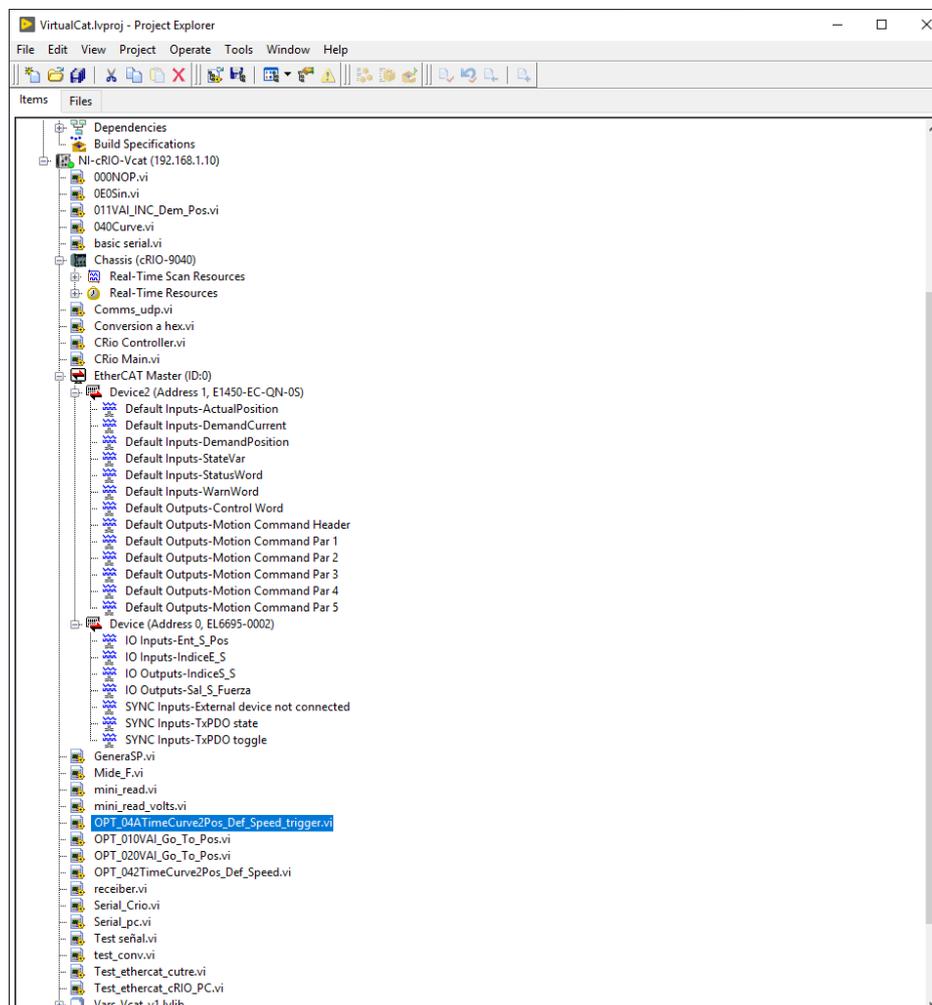


Figura 117: Explorador de proyectos en LabVIEW para CompactRIO.

El tener que acceder, o enviar los datos del PDO por tramos, en lugar del mensaje completo, que el modo en el que se había concebido el software, supuso la necesidad de realizar algunas modificaciones en la programación.

Junto a lo anterior, y tal y como se puede apreciar en la imagen que sigue a este párrafo, se modificaron algunos elementos del código que no funcionaban al ejecutar el programa con en el CompactRIO, como por ejemplo los botones del panel frontal. Esto se debe a que, el software trabaja enviando constantemente el PDO de salida y leyendo el PDO de entrada de forma cíclica a un periodo establecido. Ante esta situación se sustituyeron los botones por listas desplegables en las que seleccionar la acción deseada, y se simplificó el programa. En cualquier caso, esta prueba únicamente tenía el propósito de verificar si los problemas identificados eran causados por las consideraciones expuestas al inicio de este apartado del documento.

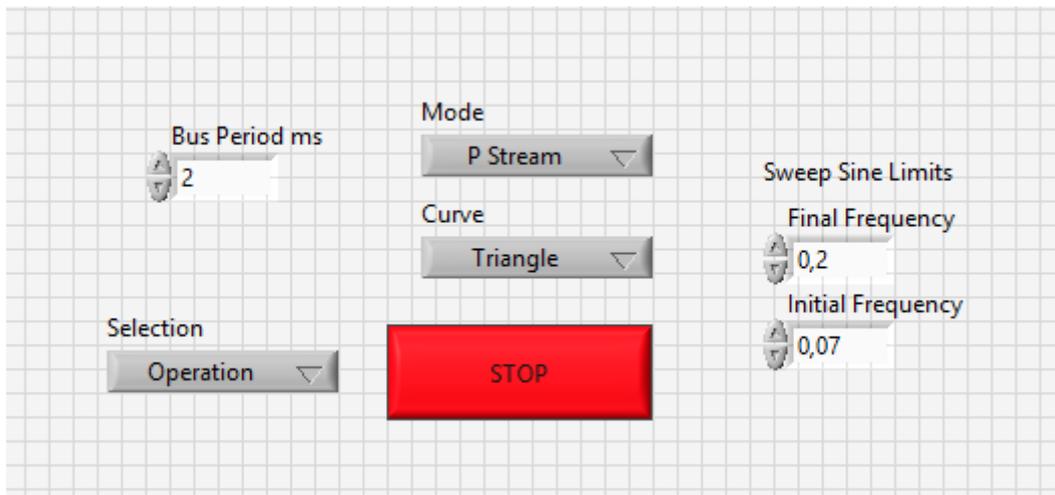


Figura 118: Panel frontal de LabVIEW para la programación en CompactRIO.

En la anterior imagen es posible ver el nuevo panel frontal, resultante del proceso de simplificación al que se ha hecho referencia. Como se mencionó, se sustituyeron los botones por un desplegable que incluye las mismas alternativas de control de la ejecución del software, y cuyo funcionamiento se comentará en los próximos párrafos.

La estructura general del programa es la misma que se expuso en los apartados anteriores. No obstante, la integración del mismo en el equipo CompactRIO supuso la necesidad de adaptar el diagrama de bloques. En primer lugar, en cuanto al PDO de entrada, el software ofrece la posibilidad de acceder directamente a la palabra de estado, y a partir de ella, extraer los bytes 47 y 54, que son los que resultaba necesario conocer a objeto de conocer si el vástago ha alcanzado la posición de *home*. Lo comentado es lo que se muestra en la siguiente figura.

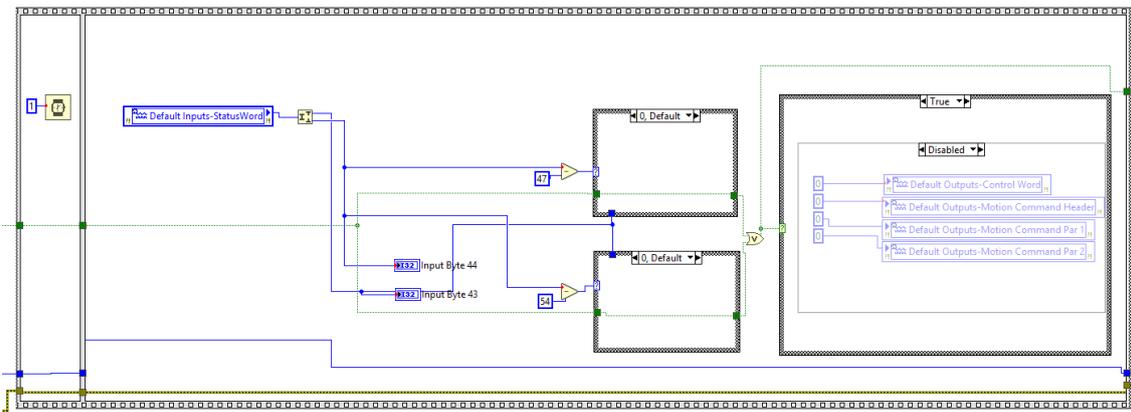


Figura 119: Bucle de comprobación de posición de home usando CompactRIO.

La imagen anterior corresponde al bucle que se ha de consultar constantemente, ya que, por seguridad, para evitar accidentes, no se permite entrar en el modo de operación hasta haber concluido el procedimiento de *home*, que es el que se muestra en la siguiente figura.

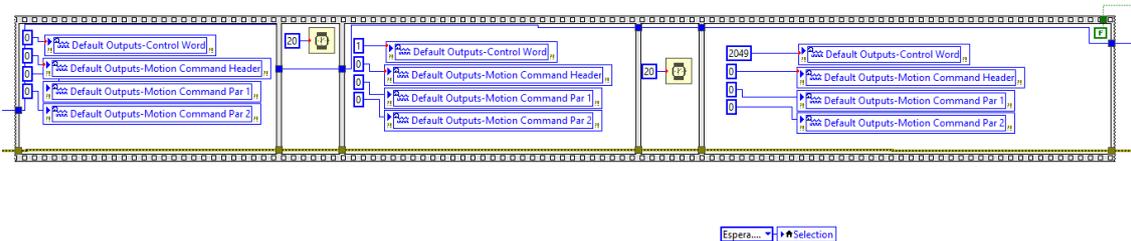


Figura 120: Secuencia para alcanzar el home usando CompactRIO.

Como se puede apreciar, los valores enviados a la red EtherCAT son los mismos que en la versión ejecutada en Windows del programa, pero si se ve modificada la forma de realizar el envío, ya que, en lugar de emplear los bloques de la librería de EtherCAT para LabVIEW, se conectan los datos a unos terminales similares a los empleados en el caso del PDO de entrada. Resulta conveniente mencionar que, como el envío de PDOs de salida es constante, llegando al controlador de forma cíclica el último PDO que se asignó en el software, nada más concluir las instrucciones que se exponen en la imagen, se debe salir de ese tramo del programa, y volver a la sección de espera, lo cual se logra con las instrucciones que se encuentran en la parte inferior derecha de la figura 120.

En cuanto a la secuencia de instrucciones para abandonar un estado de error, el funcionamiento es idéntico al del programa general, teniendo en cuenta las mismas consideraciones que para el procedimiento de *home*.

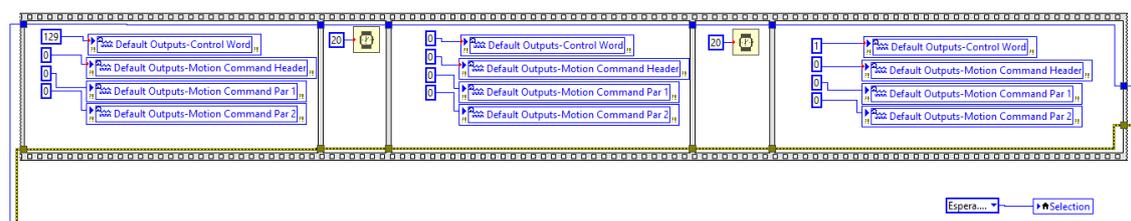


Figura 121: Secuencia de instrucciones para abandonar estado de error usando CompactRIO.

Al pasar al modo de funcionamiento, se establece el modo de operación del motor mediante las siguientes instrucciones.

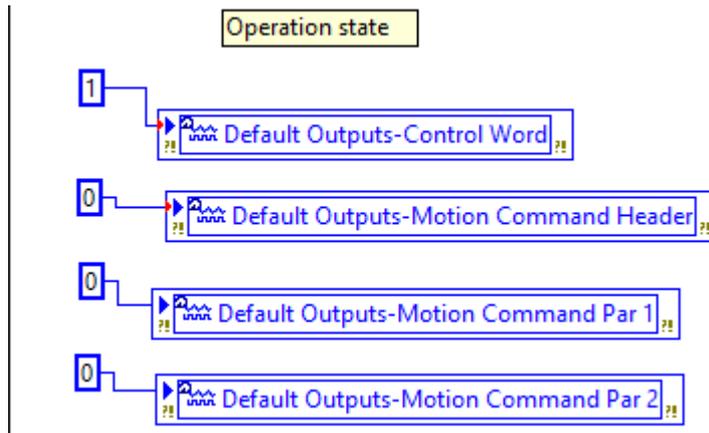


Figura 122: Paso al modo de operación usando CompactRIO.

De este modo, se llega a los bucles temporizados principal y secundario del programa. Sin embargo, en esta ocasión, como se puede ver en los cuadros de ajuste del bucle (rodeados con círculos rojos), el reloj escogido ahora es el de tiempo real que ofrece la librería instalada para el uso del dispositivo CompactRIO.

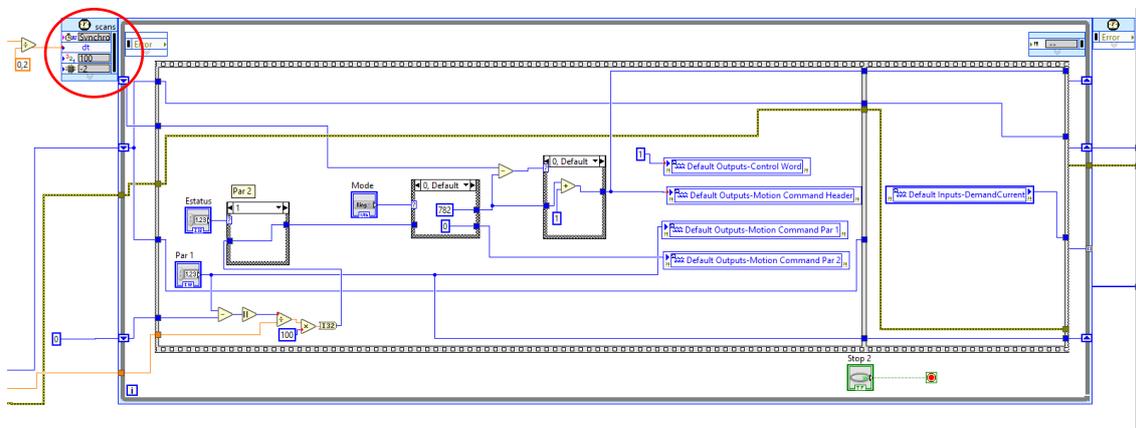


Figura 123: Bucle temporizado principal usando CompactRIO.

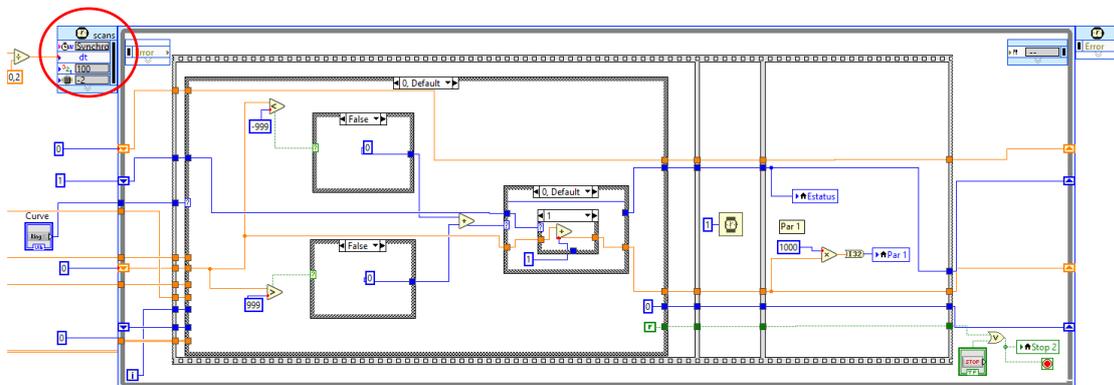


Figura 124: Bucle temporizado secundario usando CompactRIO.

Concluyendo la exposición de las modificaciones que se realizaron en el programa, tras presionar el botón STOP mostrado en la figura 118, se detiene el movimiento del motor y se abandona el estado de operación.

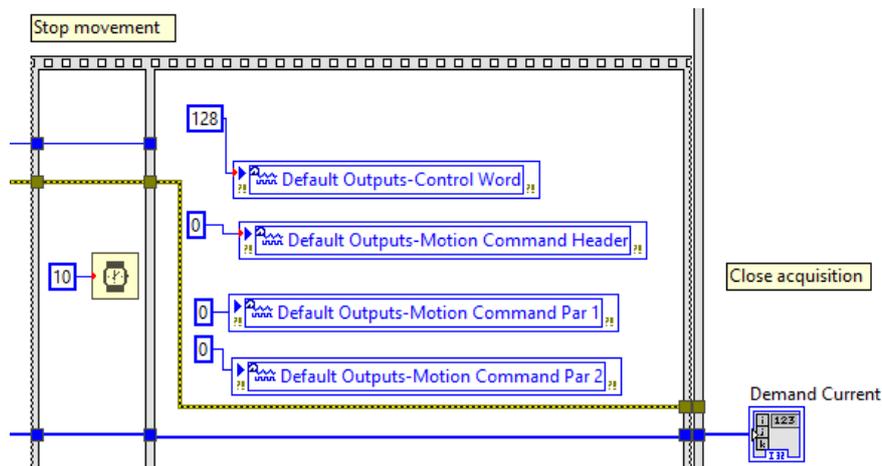


Figura 125: Finalización del software de control del motor usando CompactRIO.

Rememorando la razón de ser del ensayo descrito en este apartado del documento, el uso del CompactRIO eliminaba los ruidos que se producían durante el funcionamiento, y ya no se reproducían las detenciones en la ejecución de forma inesperada. Esto se puede justificar atendiendo a que el uso de un sistema que puede trabajar en tiempo real permite un control de la sincronización en el envío y recepción de los mensajes mucho más preciso, por lo que el controlador es capaz de realizar las interpolaciones de la posición a alcanzar en cada momento de un modo más exacto que en el caso previo. En cuanto a la no suspensión de la ejecución del programa, probablemente el uso de un equipo dedicado en la totalidad de su hardware a la tarea especificada, resulta mucho más robusto, evitando que interfieran tareas secundarias con el proceso principal que se está ejecutando.

Se puede concluir de este modo que, los dos principales dilemas que se producían mediante el equipo con sistema operativo Windows, se podrían solventar mediante esta vía, y que no se tratan de errores en la comunicación, ni en la programación efectuada en LabVIEW.

Capítulo 4. Conclusiones y líneas futuras

4.1 Conclusiones

En el presente Trabajo de Fin de Máster, se han atendido a los principales objetivos del proyecto, expuestos en el primer capítulo, de forma satisfactoria. En primer lugar, se ha diseñado un banco de ensayos para motores lineales, el cual cuenta con medidas de seguridad para la protección de los usuarios, y donde se ha ensayado el motor PS10-70X240U-BL-QJ-D01 en todas las pruebas de funcionamiento que se han ido detallando a lo largo del documento.

Junto a lo anterior, se ha programado el controlador de dicho motor como esclavo de una red EtherCAT, y se ha podido comprobar que la comunicación ha sido satisfactoria. Este nuevo bus de campo empleado, supone un incremento en la velocidad de comunicación frente a la situación del motor instalado en el sistema principal considerable. En la versión final del software desarrollado en LabVIEW, se envía una orden de movimiento cada 2 ms.

Los únicos problemas que se manifestaban en el funcionamiento del motor se pudieron solucionar cuando se sustituyó el ordenador por un equipo CompactRIO, que además es el sistema que se espera utilizar en el banco de ensayos principal en el futuro.

Por otro lado, dentro del software, se accede al PDO de entrada, con la información procedente del motor, y se ha mostrado el ejemplo de cómo se podría acceder al valor de la corriente consumida. Este sistema se podría emplear como método para la caracterización de los motores que se acoplen al banco de ensayos diseñado para este proyecto.

4.2 Líneas futuras

Atendiendo a las líneas de desarrollo en torno a las que podría evolucionar el proyecto [33], en primer lugar, se podrían citar aquellas propuestas que se han mencionado en la memoria, pero que no han sido implementadas a nivel práctico. Se presenta una propuesta en el plano para el anexo al montaje en el que, en un segundo rail, se desplace el encoder MS01-1/D-SSI. Sin embargo, para la implementación, se deberá conectar el equipo a un sistema de adquisición de datos, y que este se comunique con el software de cálculo de la posición de la catenaria, de modo que se pueda obtener en cada momento la siguiente posición de la forma más fiel al modelo matemático posible. También sería necesario fabricar las piezas diseñadas para alojar el segundo rail con su carrito, así como la banda magnética sobre la que se desplace el *encoder*.

Por otro lado, se ha concebido el banco de ensayos diseñado como una estación para la caracterización de motores lineales, con lo que se podría comenzar a realizar ese tipo de análisis, extrayendo la información necesaria para cada estudio hacia un fichero externo, y así poder analizarlo posteriormente.

Como se mencionó al inicio del documento, uno de los objetivos del proyecto recientemente concedido al grupo de investigación, es la incorporación del movimiento horizontal del cable de

corriente, tal y como ocurre en la realidad. Esta consideración requiere modificar el banco de ensayos actual, así como diseñar nuevas piezas para añadir tanto el accionamiento como el sistema de sujeción, y los elementos necesarios para transmitir el movimiento al cabezal que simula la posición de la catenaria. Todo lo mencionado en este último párrafo abriría una nueva vía de estudios a realizar, y permitiría llegar aún más lejos en el objetivo de mejorar la eficiencia del transporte ferroviario.

Capítulo 5. Presupuestos

En este trabajo se ha realizado el diseño de un banco de ensayos funcional en el que se incluyen equipos industriales de altas prestaciones ya que así lo requerían las características del proyecto. Para el cálculo del presupuesto se diferenciará entre los costes de materiales y los de desarrollo.

Para los primeros, se han tomado los precios totales recogidos en las facturas de los equipos, junto a estimaciones que ha resultado necesario realizar para algunos elementos de menor precio.

Componente	Precio (€)
CompactRIO	6118,46
Motor + Controlador + Cables	5270,03
Tarjetas EtherCAT	100
Coste de pantalla protectora	30
Equipos eléctricos de alimentación y caja	216
Presupuesto de materiales	11734,49

Tabla 1: Presupuesto de materiales.

Por otro lado, atendiendo a los costes de desarrollo, se ha estimado un precio por hora trabajada de 23 euros, el cual resulta razonable para un trabajo de ingeniería.

Horas de trabajo	Precio/hora (€/h)	Coste total de desarrollo (€)
300	23	6900

Tabla 2: Presupuesto de desarrollo.

Atendiendo a estos costes, el balance global de coste es el siguiente.

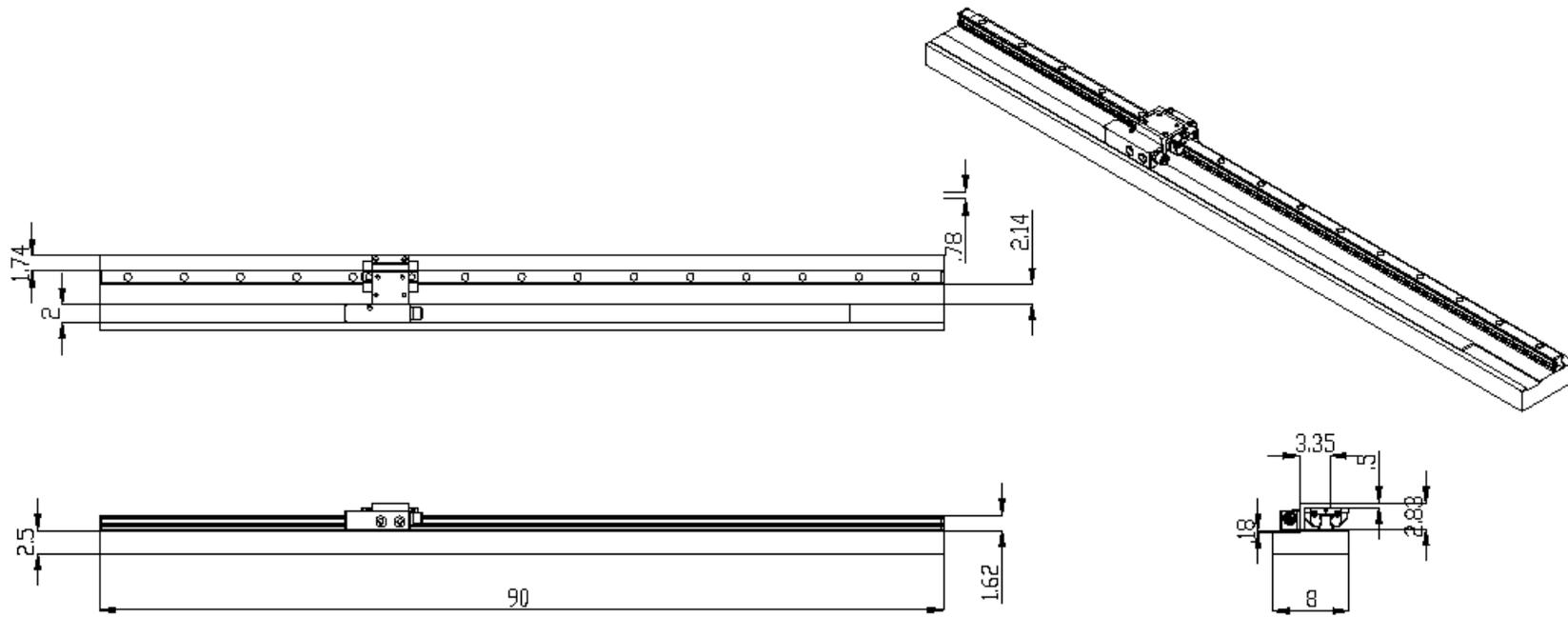
Concepto	Total
Presupuesto de materiales	11734,49
Costes del desarrollo	6900,00
Coste parcial del proyecto	18634,49
Beneficio industrial (6%)	1118,07
Gastos generales (14%)	2608,83
Coste antes de impuestos	22361,39
Impuestos (21%)	4695,89
Coste total del proyecto	27057,28

Tabla 3: Presupuesto total del proyecto.

Tras sumar los costes de materiales y desarrollo se han aplicado un 6% y un 14 % debido a los conceptos de beneficio industrial y gastos generales respectivamente. Finalmente, a la cantidad resultante se le ha aplicado el sobre coste correspondiente al IVA del 21%. De este modo, el coste total del proyecto es de 27057,28 euros.

Capítulo 6. Anexo ampliación para
MS01-1/D-SSI

En este capítulo, se adjuntará una propuesta de diseño del módulo de ampliación para el *encoder* MS01-1/D-SSI. La fabricación de las piezas involucradas, con las tolerancias dimensionales necesarias para cumplir los requisitos que se han ido presentando, no ha podido formar parte de este Trabajo de Fin de Máster debido a los plazos de fabricación de las mismas. Por esta razón, el plano únicamente se trata de una propuesta de diseño, para la cual, únicamente se incluyen las cotas de piezas que se deberían fabricar. Esto es, la base metálica donde se colocarán todos los elementos y la pieza de enganche del *encoder* al patín.



TRABAJO FINAL DE MÁSTER EN INGENIERÍA INDUSTRIAL



ESCUELA TÉCNICA
SUPERIOR
INGENIERÍA
INDUSTRIAL
VALENCIA

Proyecto: Diseño e implementación de un banco de ensayos para la caracterización de sistemas mecánicos

Plano: Ampliación para encoder MS01-1/D-SSI

Fecha: Septiembre 2021

Nº Plano:

Autor: Alberto Díaz Rodríguez

1

Capítulo 7. Bibliografía

[1] UPV Radiotelevisió, *Premio Talgo* [online], Fecha de consulta septiembre 2021. Disponible en:

https://www.youtube.com/watch?v=3Ymrh8YqBB8&ab_channel=UPVRadiotelevisi%C3%B3n%28oficial%29

[2] M.T. Valiente et al, TRA2017-84736-R-AR: “Desarrollo de un sistema de ensayos HL de pantógrafos con catenarias virtuales”. Entidad financiadora Ministerio de economía, industria y competitividad, 2018.

[3] M.T. Valiente et al, PID2020-113458RB-I00: “Ensayos HIL de pantógrafos con modelos avanzados de catenarias virtuales”. Entidad financiadora Ministerio de ciencia e innovación, 2021.

[4] A. Martínez, *Últimas instalaciones en la línea de alta velocidad* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.elcorreogallego.es/galicia/ultimas-instalaciones-en-la-linea-de-alta-velocidad-EB5650745>

[5] NI, *Información general del CompactRIO* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.ni.com/es-es/shop/compactrio.html>

[6] González Contreras, F.; Rosado Castellano, P. (2020). “Control numérico. Marco y fundamentos”. 2ª EDICIÓN. Editorial Universitat Politècnica de València. Fecha de consulta septiembre 2021. Disponible en:

<http://hdl.handle.net/10251/146827>

[7] Larraioz electronika, *Motores lineales* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://larraioz.com/linmot/productos/motores-lineales>

[8] LinMot, *Motor PS10-70x240U-BL-QJ-D01* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag1000.70.240/linear-motors/linear-motors-p10-70/stators-ps10-70x240/ps10-70x240u-bl-qj-d01.htm>

[9] LinMot, *Controlador E1450-EC-QN-0S* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag7000.e14/servo-drives/drives-for-motors-p10/series-e1400/e1450-ec-qn-0s.htm>

[10] LinMot, *Cable KSS05-02/08-D15/J-3* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag6000.10.7e/motor-cables/motor-cables-for-p10-motors/encoder-cables-for-p10-70/kss05-02-08-d15-j-3.htm>

[11] LinMot, *Motores lineales industriales* [online], Fecha de consulta septiembre 2021. Disponible en:

https://linmot.com/fileadmin//user_upload/Downloads/Brochures/0185-1100-ES_1V2_Product_Overview.pdf

[12] Alcázar Ortega, M.; Cañas Peñuelas, CS.; Escrivá Escrivá, G.; Fuster Roig, VL.; Roger Folch, J. (2019). *“Generación, transporte y distribución de energía eléctrica”*. Editorial Universitat Politècnica de València. Fecha de consulta septiembre 2021. Disponible en:

<http://hdl.handle.net/10251/118519>

[13] NI, *¿Qué es LabVIEW?* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.ni.com/es-es/shop/labview.html>

[14] Ackermann Automation, *EtherCAT Library for LabVIEW* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://ackermann-automation.de/ethercat-labview.htm>

[15] Ackermann Automation, *Documentation EtherCAT Library for LabVIEW 3* [online], Fecha de consulta septiembre 2021. Disponible en:

<http://www.ackermann-automation.de/pdf/Documentation-EtherCAT-Library-LabVIEW.pdf>

[16] EtherCAT Technology Group, *EtherCAT – the Ethernet fieldbus* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.ethercat.org/en/technology.html#1.2>

[17] LinMot, *Controlador E1450-EC-QN-0S* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag7000.e14/servo-drives/drives-for-motors-p10/series-e1400/e1450-ec-qn-0s.htm>

[18] NI, *Encoder Measurements: How-To Guide* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x1riCAA&l=es-ES>

[19] Polytec, *Laser Doppler vibrometry* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.polytec.com/eu/vibrometry/technology/laser-doppler-vibrometry>

[20] Dassault systemes, *SolidWorks* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.solidworks.com/es>

[21] LinMot, *LinMot Talk (Drive-configuration)* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://linmot.com/es/download/linmot-talk-drive-configuration/>

[22] LinMot, *Cable KPS15-04-B/Q-5* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag6000.10.7p/motor-cables/motor-cables-for-p10-motors/power-cables-for-p10-70/kps15-04-b-q-5.htm>

[23] LinMot, *Cable KSS05-02/08-D15/J-3* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag6000.10.7e/motor-cables/motor-cables-for-p10-motors/encoder-cables-for-p10-70/kss05-02-08-d15-j-5.htm>

[24] Emikon, *Series DLC* [online], Fecha de consulta septiembre 2021. Disponible en:

http://www.emikon.com/es/productos/emc_technology-filtro_emi-series_book_style-filtro_emi_potencia_monofasico_trifasico_dlc_em.htm

[25] Lovato electric, *Fuentes de alimentación conmutadas* [online], Fecha de consulta septiembre 2021. Disponible en:

https://www.electricalautomationnetwork.com/PDF/PIM/LOV/22-Fuentes-de-alimentacion-conmutadas-22-enero-2018_ES.pdf

[26] Ibérica de Automatismos, *Sensor laser Panasonic HL-G1* [online], Fecha de consulta septiembre 2021. Disponible en:

<http://www.ibericadeautomatismos.com/producto/hl-g1-sensor-laser-panasonic/>

[27] Ibérica de Automatismos, *Laser de medida Panasonic HG-C* [online], Fecha de consulta septiembre 2021. Disponible en:

<http://www.ibericadeautomatismos.com/producto/hg-c-laser-de-medida-panasonic/>

[28] Ibérica de Automatismos, *Laser de medida Panasonic HL-C1* [online], Fecha de consulta septiembre 2021. Disponible en:

<http://www.ibericadeautomatismos.com/producto/hl-c1-laser-medida-panasonic/>

[29] Ibérica de Automatismos, *Laser de medida Panasonic HL-C2* [online], Fecha de consulta septiembre 2021. Disponible en:

<http://www.ibericadeautomatismos.com/producto/hl-c2-laser-medida-panasonic/>

[30] Polytec, *Laser Doppler vibrometry* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.polytec.com/eu/vibrometry/technology/laser-doppler-vibrometry>

[31] Polytec, *Single-point vibrometers* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://www.polytec.com/eu/vibrometry/products/single-point-vibrometers>

[32] LinMot, *Encoder MS01-1/D-SSI* [online], Fecha de consulta septiembre 2021. Disponible en:

<https://shop.linmot.com/E/ag4800.1.enc/linear-motors-accessories/external-position-sensor/ms01-1-d-ssi.htm>