



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


ETSI Aeroespacial y Diseño Industrial

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

Desarrollo de experimentos para la identificación de una
articulación con Harmonic Drive.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Mecatrónica

AUTOR/A: Saneleuterio Temporal, Rafael Luis

Tutor/a: Zotovic Stanisic, Ranko

Cotutor/a: Gutiérrez Rubert, Santiago Carlos

CURSO ACADÉMICO: 2023/2024

Resumen

El objetivo de este proyecto es montar y programar una serie de experimentos que pongan a prueba un reductor Harmonic Drive para adquirir un gran volumen de datos que permitan su posterior identificación y la mejora del modelado teórico. Para ello se utilizan sensores de giro de alta precisión (encoders óptico y magnético), una tarjeta de adquisición de datos de la familia STM32 (Discovery kit) con un periodo de muestreo de 150 microsegundos gracias a interrupciones temporizadas, varios circuitos de acondicionamiento de señales, dos motores para cada lado del mecanismo (un BLDC y un servo motor) y sus respectivos controladores de potencia. La razón de estos dos motores es porque también se tiene como objetivo identificar el sistema poniendo a prueba la reversibilidad (back drivability), característica muy importante en robots colaborativos. Los ensayos se realizan sobre una estación Norelem para la máxima estabilidad de las mediciones y también se diseñan una serie de piezas para la unión de ciertos componentes. En conclusión, esta articulación elástica tan extendida en la robótica por sus altas prestaciones en poco espacio presenta parámetros que acomplejan el modelado dinámico y por ello es necesario entender e identificar el sistema de la forma más aproximada posible.

Palabras clave: Harmonic Drive, articulación elástica, back drivability, identificación, modelado, STM32, Matlab.

Resum

L'objectiu d'aquest projecte és construir i programar una sèrie d'experiments que posen a prova un reductor Harmonic Drive per adquirir un gran volum de dades que permeten la seua posterior identificació i la millora del modelatge teòric. Per a això s'utilitzen sensors de gir d'alta precisió (encoders òptic i magnètic), una targeta d'adquisició de dades de la família STM32 (Discovery kit) amb un període de mostreig de 150 microsegons gràcies a interrupcions temporitzades, diversos circuits de condicionament de senyals, dos motors per cada costat del mecanisme (un BLDC i un servo motor) i els seus respectius controladors de potència. La raó d'aquests dos motors és perquè també es té com a objectiu identificar el sistema posant a prova la reversibilitat (back drivability), una característica molt important en robots col·laboratius. Els assajos es realitzen sobre una estació Norelem per a la màxima estabilitat dels mesuraments i també es dissenyen una sèrie de peces per a la unió de certs components. En conclusió, aquesta articulació elàstica tan estesa a la robòtica per les seues altes prestacions en poc espai presenta paràmetres que acomplexen el modelatge dinàmic i per això cal entendre i identificar el sistema de la manera més aproximada possible.

Paraules clau: Harmonic Drive, articulació elàstica, back drivability, identificació, modelatge, STM32, Matlab.

Abstract

The objective of this project is to assemble and program several experiments to test a Harmonic Drive reducer to acquire a large volume of data that allows its subsequent identification and the improvement of theoretical modeling. To achieve this, high-precision rotation sensors (optical and magnetic encoders), a data acquisition module from the STM32 family (Discovery kit) with a sampling period of 150 microseconds thanks to timed interruptions, several signal conditioning, two motors for each side of the mechanism (a BLDC and a servo motor) and their respective power controllers are used. The reason for these two motors is because the objective is also to identify the system by testing the back drivability, a very important feature in collaborative robots. The tests are carried out on a Norelem station for maximum measurement stability and several pieces are also designed to join certain components. In conclusion, this elastic joint widely used in robotics due to its high performance in a small space, presents parameters that make dynamic modeling a complex work and therefore it is necessary to understand and identify the system as accurately as possible.

Keywords: Harmonic Drive, elastic joint, back drivability, identification, modeling, STM32, Matlab.

Índice general

1. Memoria	4
2. Planos	131
3. Pliego de condiciones	143
4. Presupuesto	152



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ETSI Aeroespacial y Diseño Industrial

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DESARROLLO DE EXPERIMENTOS PARA LA IDENTIFICACIÓN DE UNA ARTICULACIÓN CON HARMONIC DRIVE

1. MEMORIA

TRABAJO FINAL DEL

Máster Universitario en Ingeniería Mecatrónica

REALIZADO POR

Rafael Luis Saneleuterio Temporal

TUTORIZADO POR

Ranko Zotovic Stanisic

CODIRECTOR:

Santiago Carlos Gutiérrez Rubert

CURSO ACADÉMICO: 2023/2024

Índice memoria

1. Introducción	9
2. Objeto	13
3. Antecedentes	13
4. Estudio de necesidades, limitaciones y condicionantes	15
5. Planteamiento de soluciones alternativas y justificación de la solución adoptada	16
5.1. Alternativas en el entorno mecánico	16
5.2. Alternativas en el entorno electrónico	20
5.3. Alternativas en el entorno de programación	28
5.4. Alternativas en el entorno de control	36
6. Descripción y justificación detallada de la solución adoptada	41
6.1. Subsistema mecánico	41
6.1.1. Harmonic Drive	41
6.1.2. Piezas impresas	45
6.1.2.1. Bases	45
6.1.2.2. Ejes	46
6.1.2.3. Soportes de encoders	46
6.1.3. Norelem	47
6.1.4. Elementos auxiliares	47
6.2. Subsistema electrónico	48
6.2.1. Módulo de adquisición de datos	48
6.2.2. Encoder magnético	49
6.2.3. Encoder óptico	52
6.2.4. Motor BLDC	53
6.2.5. Servomotor	54
6.2.6. Controlador BLDC	55
6.2.7. Controlador servomotor	58
6.2.8. Circuitos de acondicionamiento	58
6.2.9. Alimentación	61
6.2.10. Diagrama de conexiones	62
6.3. Subsistema de programación	62
6.3.1. Configuración en STM32Cube	62
6.3.2. Código principal	65
6.3.3. Posproceso	70
6.4. Subsistema de control	73
6.4.1. Modelado	73
6.4.2. Ensayos y datos experimentales	77
6.4.3. Identificación	81
7. Conclusiones	94
8. Bibliografía	96

9. Anexos	100
ANEXO I. RELACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE	100
ANEXO II. CÓDIGO PRINCIPAL STM32CUBE IDE	102
ANEXO III. CÓDIGO POSPROCESO MATLAB	120
ANEXO IV. CÓDIGO IDENTIFICACIÓN MATLAB	123

Índice de figuras

Figura 1: Modelo de Harmonic Drive SE [2]	9
Figura 2: Despiece del Harmonic Drive [2]	11
Figura 3: Principio de funcionamiento del Harmonic Drive	11
Figura 4: Laboratorios V.5E.0.013 (izq.) y V.5E.0.066 (der.)	15
Figura 5: Montaje de primeras pruebas con el encoder magnético	17
Figura 6: Posición del encoder magnético, detalle de perturbación	17
Figura 7: Posición del encoder magnético con marcas en cada perturbación	18
Figura 8: Posición del encoder magnético tras 40 vueltas	18
Figura 9: Juntas para ejes del laboratorio	20
Figura 10: Entrada del harmonic drive	20
Figura 11: Pruebas con un módulo NUCLEO-F303K8	21
Figura 12: Cuadratura de dos canales de encoder	22
Figura 13: ESCON Module 50/5 usado en [6]	23
Figura 14: ESCON Module 50/5 incorporado a segunda etapa de potencia [22]	23
Figura 15: Pruebas con amplificadores para la señal del encoder magnético	24
Figura 16: Posición del encoder magnético usando amplificadores	25
Figura 17: Amplificación de la señal de referencia con filtro	25
Figura 18: Amplificación diferencial de la señal de corriente	26
Figura 19: Corriente usando AO y <i>buffers</i> (izq.) frente a conexión directa (der.)	27
Figura 20: Primeros ensayos con fuente Ruzizao	27
Figura 21: Captura de datos deformes en ESCON Studio	28
Figura 22: Versiones de configuración y programa en STM32CubeIDE	29
Figura 23: Evolución del valor de datos guardados en SDRAM	30
Figura 24: Programas utilizados para comunicación serie	31
Figura 25: Resultado de obtener valores absolutos a partir de relativos	32
Figura 26: Error de dato recibido en el posproceso de Matlab	34
Figura 27: Posiciones de encoders sin usar “flush”	34
Figura 28: Cálculo de la acción de control antes del tiempo de experimentación	35
Figura 29: Captura del efecto PWM en ESCON Studio	35
Figura 30: Captura del efecto DAC en ESCON Studio	36
Figura 31: Posiciones y velocidades sin aplicar filtro	37
Figura 32: Comparación de velocidades promediadas	38
Figura 33: Comparación de velocidades usando diferentes filtros	39
Figura 34: Datos del primario del reductor ante señal cuadrada	40
Figura 35: Organigrama general del proyecto	41
Figura 36: Harmonic drive HFUC-14-2UH real y en CAD	42
Figura 37: Rigidez torsional dependiente del par [13]	43
Figura 38: Flexspline encajado en circular spline para E1, E2 y E3	43
Figura 39: Proceso de despiece del Harmonic drive	44

Figura 40: Flexspline extraído real y en CAD	44
Figura 41: Flexspline (izq.) y wave generator (der.) en E4	44
Figura 42: Base E123 (izq.) y Base E4 (der.)	45
Figura 43: Eje E134 (izq.) y Eje E23 (der.)	46
Figura 44: Soporte encoder E1 (izq.) y Soporte encoder E2 (der.)	47
Figura 45: Base y piezas de Norelem del laboratorio para los ensayos	47
Figura 46: Estructuras de Lego para sujetar los lectores	48
Figura 47: Microcontrolador STM32F429 Discovery kit	49
Figura 48: Implementación de lector de encoder magnético según [33]	50
Figura 49: Lector de encoder magnético soldado en PCB	50
Figura 50: Anillo axial magnético incremental	50
Figura 51: Frecuencia de señales de salida del encoder	52
Figura 52: Frecuencias asignadas a periféricos de la Discovery	52
Figura 53: Lector y cabezal del encoder óptico	53
Figura 54: Motor BLDC en escuadra	54
Figura 55: Servomotor de potencia	54
Figura 56: Controlador ESCON 50/5 [25]	56
Figura 57: Configuración de parámetros del motor BLDC	57
Figura 58: Configuración del control y referencia del motor BLDC	57
Figura 59: Configuración de la salida analógica de la ESCON	57
Figura 60: Controlador Digitax HD con Keypad	58
Figura 61: Esquema de divisores de tensión (DT) para cada encoder	59
Figura 62: Pines del amplificador operacional dual integrado LM358 [40]	59
Figura 63: Esquema de circuito amplificador	60
Figura 64: Fuentes DC para Discovery, ESCON y acondicionamientos	61
Figura 65: Fuentes DC para Digitax HD	61
Figura 66: Diagrama electrónico de conexiones	62
Figura 67: Eliminar configuración por defecto	63
Figura 68: Configuración del reloj en STM32Cube	63
Figura 69: Configuración del timer para las interrupciones temporizadas	64
Figura 70: Configuración del timer para el encoder magnético	65
Figura 71: Código principal temporizado	66
Figura 72: Diagrama de flujo del <i>main</i>	67
Figura 73: Diagrama de flujo de programa temporizado	67
Figura 74: Diagrama de flujo de compresión de datos	69
Figura 75: Diagrama de flujo del cálculo de la referencia	70
Figura 76: Configuración PuTTY	71
Figura 77: Posproceso en Matlab	72
Figura 78: Función deslimitarU en Matlab	72
Figura 79: Simplificación aproximada de articulación elástica	73
Figura 80: Espacio de estados general para una articulación elástica	74
Figura 81: Representación aproximada de las hebras elásticas	76
Figura 82: Modelos de fuerza de rozamiento según la velocidad	77
Figura 83: Esquema de experimentos objetivo	78
Figura 84: Esquema de experimentos resultantes	79
Figura 85: Montaje experimentos E1 y E1'	80
Figura 86: Montaje experimentos E2' y E3'	80
Figura 87: Montaje experimento E4	80

Figura 88: Escalón insuficiente para iniciar movimiento	82
Figura 89: Punto del modo escalera en el que comienza el movimiento	82
Figura 90: Arranque con impulsos crecientes	83
Figura 91: Formación de una matriz de regresores	84
Figura 92: Representación de las curvas de posición reales, teóricas y estimadas	84
Figura 93: Detalle de las curvas de posición reales, teóricas y estimadas	85
Figura 94: Matrices de coeficientes estimados y teóricos	85
Figura 95: Modelo aproximado de la reducción del Harmonic Drive	86
Figura 96: Aproximación de la posición del primario a través de Correlation Models	86
Figura 97: Aproximación a través de Transfer Function Models	87
Figura 98: Aproximación a través de Process Models	87
Figura 99: Función de transferencia del modelo estimado de Process Models	88
Figura 100: Implementación de modelo ARX no lineal en Matlab	88
Figura 101: Velocidades y resonancia ante entrada senoidal (E1')	89
Figura 102: Velocidades y resonancia ante escalones con offset (E1')	89
Figura 103: Velocidades y resonancia con y sin filtro (E1')	90
Figura 104: Velocidades y resonancia ante entrada senoidal (E1)	90
Figura 105: Velocidades y resonancia ante escalones con offset (E1)	91
Figura 106: Modelo y fundamento de junta Oldham [13]	91
Figura 107: Errores de precisión en la medición del encoder magnético [33]	92
Figura 108: Errores de precisión en la medición del encoder magnético [33]	93
Figura 109: Velocidades con resonancia y diafonía ante escalones con offset (E1)	93

Índice de tablas

Tabla 1: Velocidades máximas según modelo de encoder [33]	51
Tabla 2: Datos del servomotor según su número de serie [37]	55

1. Introducción

En este proyecto final de máster se pone a prueba el sistema mecánico de un Harmonic Drive de reducción 30:1 de la empresa alemana Harmonic Drive SE (véase figura 1). Este mecanismo fue inventado por el prolífico americano Clarence Walton Musser en la década de los 50 [1] y en los años siguientes se comercializó a gran escala. Hasta la fecha, este reductor objeto de estudio todavía no tiene un modelo que lo defina al completo y es por ello que este proyecto pretende contribuir con material útil a esta línea de investigación. Para ello se llevan a cabo diferentes experimentos para capturar de la forma más precisa posible el desempeño del Harmonic Drive y almacenar los datos para que puedan usarse para tareas de identificación.



Figura 1: Modelo de Harmonic Drive SE [2]

Los reductores han desempeñado un papel muy importante dentro de la mecánica, permitiendo multiplicar la fuerza a la salida a partir de una entrada rotórica más rápida. Con la existencia de estos sistemas se ven beneficiados campos de la ingeniería tales como la robótica, aeronáutica, aeroespacial, defensa militar, automatización, máquinas herramienta, medicina y un largo etcétera. De todos estos campos existen una gran cantidad de aplicaciones que requieren motores para aportar movimiento a un sistema, en el caso que más se relaciona con este proyecto, la robótica, los actuadores más extendidos son los motores eléctricos.

Ahora bien, a la hora de seleccionar los componentes que van a dotar de movimiento a un sistema robotizado hay que tener en cuenta que depende del tipo de robot y de la aplicación en la que va a trabajar [3]. Sin embargo, en la mayoría de ocasiones el motor por sí solo no es capaz de aplicar la fuerza suficiente o directamente el uso de un motor demasiado grande puede comprometer otros aspectos de la máquina. Es aquí donde entran en juego los reductores, ya que la combinación reductor/motor suele asegurar una mejora en el peso, espacio y costes. Además, como los motores eléctricos como los *brushless* o servo motores pueden alcanzar velocidades del orden de miles de revoluciones por minuto, se puede

reducir la velocidad en varios múltiplos con un reductor sin repercusiones para obtener una multiplicación del par ejercido a la articulación. Esta técnica encaja perfectamente con el funcionamiento de los brazos robóticos industriales o colaborativos, que necesitan hacer movimientos a través de soluciones que equilibran la capacidad de fuerza y velocidad con las articulaciones de menor peso y volumen posible.

Es aquí donde el Harmonic Drive destaca por conseguir una gran reducción en un formato más compacto y ligero en comparación con los reductores de engranajes clásicos. Con una sola etapa de acoplamiento se consiguen reducciones de hasta 320:1 (al menos de forma comercializada por la principal distribuidora [2]).

A parte de lo ya mencionado, el Harmonic Drive tiene otras ventajas que en suma lo han situado como uno de los principales mecanismos en articulaciones robóticas:

- No *backlash*: este es un efecto que se da mucho en engranajes tradicionales; en cambio, los fabricantes del Harmonic Drive aseguran que para cualquier movimiento en la entrada del sistema siempre habrá efecto en la salida (es decir que la holgura es muy baja o nula). Esto es debido a la naturaleza de la disposición de los dientes los cuales están en continuo engrane.
- Reversible: característica también conocida como *back drivability*. Le permite al mecanismo ser accionado desde la salida y es importante que esto sea posible para robots colaborativos, para tener la posibilidad de que un humano pueda posicionar los eslabones manualmente.
- Ejes céntricos: el diseño del Harmonic Drive sitúa al eje de salida centrado con el eje de entrada. Este factor facilita enormemente el montaje y acoplamiento de estos reductores en las articulaciones de robots u otras máquinas.
- Gran capacidad de carga: este tipo de reductores pueden soportar pares muy elevados dado que la presión se reparte de forma óptima entre los dientes de los engranajes.

En contraste con los puntos anteriores, es importante mencionar cuáles son los puntos desfavorables del Harmonic Drive:

- Coste elevado: hasta la fecha, el Harmonic Drive sigue siendo una opción menos asequible que otros reductores.
- Resonancias: este tipo de reductores puede presentar vibraciones indeseadas en las frecuencias de resonancia de la transmisión armónica.
- Componente elástico: si bien es cierto que, según los fabricantes, no hay holgura en la transmisión ya que siempre hay dientes engranando, la pieza elástica del *flexspline* (ver figura 2) aporta una componente de error al sistema difícil de identificar y fuente de efectos no deseados como la histéresis en el movimiento.
- No linealidades: el harmonic drive es un sistema mucho más complejo que los reductores tradicionales, resolver el problema de identificar este componente no es una tarea trivial. En adición a esto, la respuesta del mecanismo puede variar dependiendo de la posición y sus condiciones, es decir, existe cierta incertidumbre.

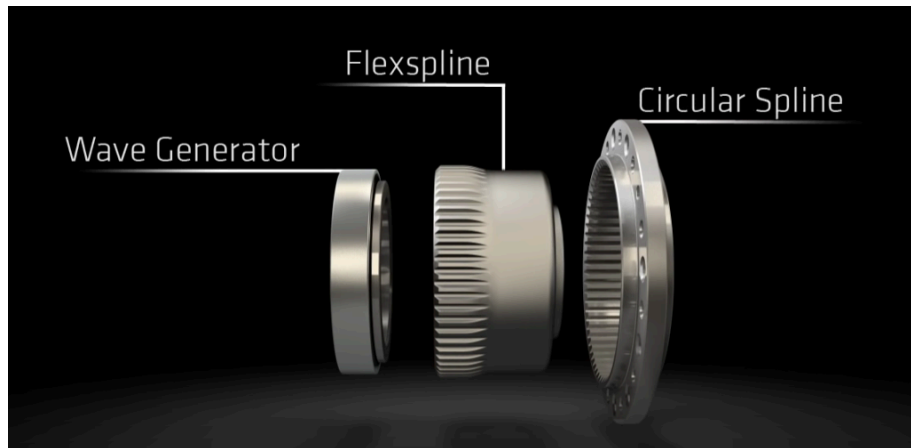


Figura 2: Despiece del Harmonic Drive [2]

A continuación se van a introducir las bases del funcionamiento del Harmonic Drive y cómo interactúan entre sí sus tres elementos principales: el wave generator, el flexspline y el circular spline (véase figura 3). Cabe decir que, dependiendo de qué componente se fije o de cuál se considere la entrada o la salida del reductor, el comportamiento será diferente. Con esto en mente, se procede a explicar el caso típico para entender los cimientos del sistema con el que se va a trabajar en el presente proyecto.

En primer lugar, la fuente de movimiento rotórico (un motor por ejemplo) hace girar una pieza con forma elíptica llamada wave generator que generalmente es un tipo de rodamiento. Este componente es la entrada del mecanismo y viene encajado dentro del flexspline, una pieza metálica flexible que adquiere esta forma elíptica giratoria [4].

En segundo lugar, entra en juego el acoplamiento de los dientes del perímetro exterior del flexspline con los dientes del interior del circular spline. La presente forma elíptica asegura que los dientes de áreas diametralmente opuestas entren en contacto. Por lo general, existe una diferencia de 2 dientes entre estos componentes que aseguran la transmisión y, cuantos más dientes tengan ambos elementos, mayor será la reducción. Concretamente, cada vuelta del primario se avanza una cantidad igual a la diferencia de dientes entre estos componentes. El circular spline suele posicionarse fijo y es el que hace de apoyo para que el flexspline tenga un movimiento relativo y actúe como la salida del reductor.

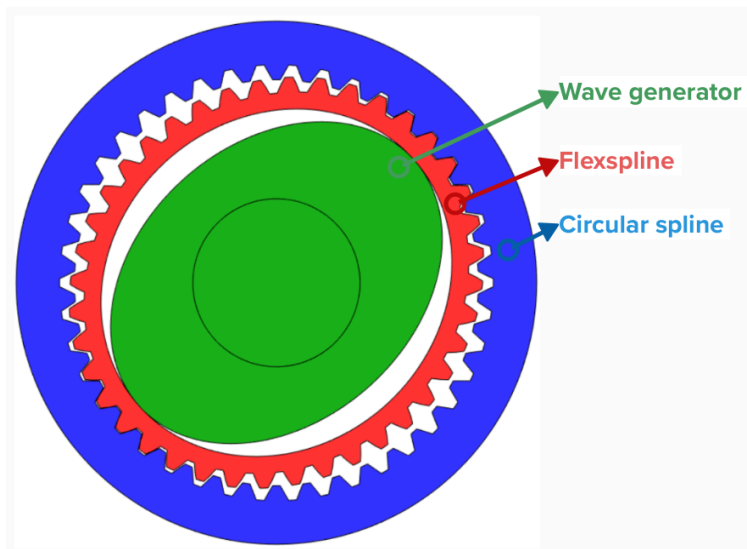


Figura 3: Principio de funcionamiento del Harmonic Drive

Por otro lado, pese a que el proyecto se centra más en el sector de la experimentación que en el subsistema de control (modelado e identificación), cabe contextualizar este aspecto para plantear correctamente los ensayos. Por tanto, el modelado e identificación de un sistema se refiere al proceso de crear un modelo matemático que representa el comportamiento dinámico de un sistema a partir de observaciones experimentales. Este modelo permite predecir cómo responderá el sistema a diferentes entradas o condiciones iniciales. El proceso de identificación comienza con la recopilación de datos experimentales que reflejan las entradas y salidas del sistema. Luego, estos datos se utilizan para ajustar los parámetros de un modelo matemático propuesto (que puede ser lineal o no lineal, determinístico o estocástico) de modo que el modelo sea capaz de replicar el comportamiento observado. Técnicas como la regresión lineal, el método de mínimos cuadrados y algoritmos de optimización son comúnmente utilizadas en esta etapa. Además, el Harmonic Drive presenta efectos que perturban el normal desempeño del modelo como pueden ser el rozamiento seco, la histéresis, el error cinemático, fricción no lineal, amortiguamiento y rigidez no lineal que añaden complejidad al proceso de identificación.

Para llevar a cabo la identificación del sistema de manera efectiva, es crucial considerar varios factores al experimentar y obtener datos [5]. En primer lugar, la calidad y cantidad de datos es fundamental: los datos deben ser precisos y abarcar un rango suficiente de condiciones operativas del sistema para asegurar que el modelo resultante sea robusto y generalizable. Además, las entradas al sistema deben ser variadas y suficientemente excitantes para asegurar que toda la dinámica del sistema sea capturada en los datos.

Cerrando el aspecto del control, es importante tener en cuenta la presencia de ruido y posibles errores de medición en los datos. El ruido puede ser causado por variaciones aleatorias en el sistema o por imprecisiones en los sensores. Los métodos de identificación de sistemas deben ser capaces de filtrar el ruido y enfocarse en la dinámica real del sistema. Además, es fundamental validar el modelo identificado mediante pruebas adicionales y, si es posible, compararlo con otros modelos para asegurar su precisión y fiabilidad. La validación puede incluir el uso de subconjuntos de datos no utilizados durante el proceso de identificación inicial y la comparación de las predicciones del modelo con observaciones experimentales adicionales.

Con todo, la realización del proyecto queda justificada por la creciente necesidad de mejorar la precisión de los robots, perfeccionando el modelo de los elementos mecánicos que los componen. Se trata, por tanto, de seguir la idea de contribuir a esta línea de investigación llevando a cabo experimentos con este gran sistema mecánico como es el Harmonic Drive que aporten más información y nuevos enfoques para que el modelo computacional se aproxime cada vez más a la complicada realidad.

Antes de proseguir con la propia memoria, cabe comentar brevemente su estructura. Para tratar de alcanzar una redacción completa de todos los aspectos del desarrollo del proyecto, el documento se divide en los apartados de objetivos, antecedentes, estudio de necesidades, soluciones alternativas y la descripción y justificación detallada de la solución adoptada. Por último, se encuentran unas conclusiones, la bibliografía y anexos.

Finalmente, para cada experimento, se va a llevar a cabo un ensamblaje físico del sistema, un montaje electrónico y una programación. Luego, tiene lugar una posterior interpretación de los datos de algunos experimentos con herramientas de identificación junto con un modelado pero, este apartado no profundiza en el aspecto de la identificación tanto como sería deseable debido a limitaciones de tiempo. En consecuencia, el peso del trabajo se

encuentra mayoritariamente en los apartados previos. Por tanto, el mencionado apartado de descripción detallada de la solución adoptada está dividido en cuatro grandes subsistemas: el mecánico, el electrónico, el de programación y el de control.

Para acabar, conviene señalar que, como es habitual en proyectos de ingeniería, aparte de la memoria, se incluyen los documentos de planos, pliego de condiciones y presupuesto.

2. Objeto

El objetivo general (OG) del presente trabajo es desarrollar una base de experimentos precisa que permita la obtención de numerosos datos para la identificación de fenómenos y parámetros del Harmonic Drive.

Como objetivos específicos (OE) del proyecto, se pretende:

- OE1. Montar una mesa de experimentación lo más precisa y estable posible para realizar pruebas con el reductor Harmonic Drive HFUC-14-2UH.
- OE2. Diseñar un subsistema electrónico robusto para el control y adquisición de datos dando uso también de sensores de alta precisión.
- OE3. Desarrollar un programa optimizado que, junto con la electrónica, consiga una frecuencia de muestreo y resolución superior a la presente en otros proyectos y artículos de esta área de estudio.
- OE4. Realizar ensayos aplicando por desde la entrada del reductor un control por corriente en vacío y con carga.
- OE5. Poner a prueba la back drivability del Harmonic Drive aplicando señales de par desde la salida con un servomotor de potencia.
- OE6. Experimentar con una versión simplificada del propio Harmonic Drive.
- OE7. Describir el modelado teórico con los parámetros que definen al Harmonic Drive.
- OE8. Interpretar datos adquiridos y obtener una identificación de parámetros y fenómenos del sistema que permitan alcanzar conclusiones provechosas para futuros trabajos.

3. Antecedentes

El principal trabajo predecesor del presente proyecto es [6]. Su desarrollo se centra en el modelado y simulación de articulaciones elásticas, abordando tanto la identificación de parámetros como la simulación de su comportamiento. Se desarrollan modelos matemáticos que representan la dinámica de las articulaciones elásticas y se emplean técnicas de identificación para ajustar los parámetros del sistema basadas en datos experimentales con un periodo de muestreo de 1 ms y encoders de máximo 2000 pulsos. Además, se realizan simulaciones para validar la precisión del modelo propuesto y se discuten las implicaciones prácticas de estos modelos en el diseño y control de sistemas robóticos. Como añadidura, hay ciertos componentes electrónicos en el presente proyecto que son reutilizados del trabajo de [6], por lo que existe más relación todavía entre ambos.

Sin embargo, el reto de modelar el harmonic drive no es reciente, ya desde el año 1992 se pueden encontrar trabajos como el extenso [7], que investiga con detalle el comportamiento de las transmisiones de engranajes de tipo harmonic drive, ya que son componentes clave

en muchos sistemas robóticos. El trabajo incluye un análisis profundo de las características no lineales y de fricción que afectan el rendimiento de estos mecanismos. Se desarrollan modelos detallados para capturar estas dinámicas complejas y se validan mediante comparaciones con datos experimentales con una precisión máxima de 0.0008° y una frecuencia de muestreo de 1 ms.

También, Ruderman y colaboradores han desarrollado una serie de trabajos desde 2009 relacionados con la experimentación y modelado de articulaciones elásticas. En primer lugar, el artículo [8] aborda el modelado y la identificación de juntas elásticas en robots, considerando efectos de histéresis y holgura (backlash). Los autores presentan un modelo matemático que incorpora estos efectos no lineales y emplean técnicas de identificación para ajustar los parámetros del modelo. Se valida el modelo mediante comparaciones con datos experimentales, mostrando su capacidad para reproducir con precisión el comportamiento de las juntas elásticas bajo diferentes condiciones operativas. En segundo lugar, [9] se centra en el modelado, observación y control de la torsión con histéresis en juntas elásticas de robots. Los autores desarrollan un modelo detallado que describe la histéresis torsional y proponen métodos de control para mitigar sus efectos. Se validan los modelos y métodos propuestos mediante experimentos a 500 microsegundos de periodo de muestreo y una resolución máxima de encoder de 2880000 pulsos por revolución, mostrando su utilidad para mejorar el control y la precisión de robots con juntas elásticas. En tercer lugar, [10] Ruderman e Iwasaki presentan un enfoque de control para robots con juntas elásticas no lineales utilizando desacoplamiento de par alimentado hacia adelante. El método propuesto mejora la precisión del control al compensar las no linealidades presentes en las juntas elásticas. Se validan los resultados mediante simulaciones y experimentos, demostrando la eficacia del enfoque para mejorar el rendimiento de robots con este tipo de juntas. Finalmente, [11] se centra en las fuerzas de arranque en sistemas de movimiento actuado que presentan fricción no lineal. Se desarrolla un modelo que captura el comportamiento de estas fuerzas de arranque y se discute su impacto en el rendimiento del sistema. Este estudio es importante para el diseño y control de sistemas de movimiento actuados, donde la fricción no lineal puede afectar significativamente la precisión y estabilidad del sistema.

En la misma línea, Preissner y colaboradores presentan en [12] un modelo de alta fidelidad para transmisiones de engranajes de tipo harmonic drive incluyendo efectos tales como fricción viscosa no lineal, rigidez no lineal, histéresis y error cinemático. La estrategia en este trabajo tiene un enfoque de caja negra (black box), se valida el modelo mediante comparaciones con datos experimentales, con una tasa de muestreo de 2048 Hz, y se demuestra su utilidad para el diseño y análisis de sistemas que utilizan transmisiones de engranajes de tipo harmonic drive.

En conjunto, estos trabajos demuestran la evolución y la diversidad de técnicas en el campo del modelado y control de sistemas elásticos y no lineales. Mientras algunos estudios se centran en el desarrollo de modelos matemáticos detallados y su validación experimental, otros avanzan hacia la implementación de métodos de control prácticos basados en esos modelos. Al seguir por esta línea de investigación se avanza en un mejor entendimiento y manejo de las complejidades inherentes a estos sistemas. La sinergia entre estos enfoques promete mejoras significativas en el diseño y control de sistemas robóticos y de transmisión, impactando positivamente en su precisión, eficiencia y robustez.

4. Estudio de necesidades, limitaciones y condicionantes

La industria está en continua evolución y un factor clave es la presencia de sistemas robóticos que ofrezcan las mejores prestaciones para cada aplicación. La precisión y fiabilidad de los movimientos mecánicos son propiedades necesarias para realizar las tareas dentro de cierto margen de error. Cuanto mejores sean los elementos utilizados en cada subsistema que forma un robot (mecánico, electrónico, computacional, control, lubricación, etc.), más se reducirá ese margen de error. En este proyecto se pretende aportar a la mejora del control de estos sistemas a través de una serie de experimentos sobre una articulación y la recolección de múltiples datos de entrada y salida: corriente de referencia, corriente real (excepto E3), posición angular de la entrada, posición angular de la salida (excepto E4) y tiempo. Todo ello para sacar conclusiones que puedan aprovechar futuros trabajos y mejorar el modelo de la articulación robótica y, por tanto, que se obtenga un control más preciso.

Una vez planteadas las necesidades generales que cubre el tema del proyecto es importante exponer los límites y factores que condicionan la manera en que este se ha llevado a cabo. Como normativa relacionada se tienen en cuenta documentos de la índole de instrumentación electrónica, medición, control, sector robótico, compatibilidad electromagnética y seguridad. Estos apartados se especifican en el documento de Pliego de condiciones del presente trabajo. También se respetan las normas y recomendaciones del laboratorio de control numérico V.5E.0.013 y el laboratorio de fabricación integrada V.5E.0.066 de la Universidad Politécnica de Valencia (figura 4) donde se ha llevado a cabo toda la experimentación. Por otro lado, las restricciones y circunstancias más relevantes en el desarrollo del presente proyecto nacen del material disponible y de la naturaleza de los propios experimentos.



Figura 4: Laboratorios V.5E.0.013 (izq.) y V.5E.0.066 (der.)

En primer lugar, las pruebas se realizan con un único modelo de harmonic drive, el HFUC-14-2UH, el cuál pertenece a la universidad y no se ha utilizado para ninguna otra aplicación. Así mismo, la escuela ofrece material tanto electrónico como mecánico para el desarrollo de los demás aspectos del trabajo, no obstante, la elección de estos componentes se detalla en el siguiente apartado de planteamiento de soluciones alternativas y justificación de la solución adoptada.

En segundo lugar, el tipo de pruebas que se tiene como objetivo realizar con el mecanismo son las siguientes: acción desde el primario en vacío (E1), acción desde el secundario en vacío (E2), acción desde ambos lados (E3) y acción desde el primario solo con las piezas

del *flexspline* y *wave generator* (E4). Todos estos ensayos se realizan con los ejes de giro en horizontal.

Por tanto, no se plantea un abanico de experimentos más amplio (por ejemplo, incluyendo eslabones) por limitaciones de tiempo. Por otra parte, se trabaja con una reducción de 1:30 y el engranaje está lubricado con grasa [13].

En cuarto lugar, todos los experimentos se realizan con un periodo de muestreo de 150 microsegundos y la resolución del sensor de posición del primario son 8192 cuentas (13 bits) por revolución mientras que en el secundario se tienen 589824 cuentas (72 x 13 bits). Las acciones de control aplicadas son de múltiples tipos (pulsos, escalones, rampas y senoidales) incluyendo también versiones aleatorias y con offset de estas en el sistema para una mejor identificación. El tipo de señal más excitante con el que se trabaja es la suma de tres señales senoidales de frecuencia y amplitud aleatorias que cambian su forma cada cierto tiempo durante cada experimento.

También conviene aclarar que ciertas gráficas de la memoria no disponen de leyenda en la propia imagen debido a que no se tuvo en cuenta este aspecto a la hora de representar los datos en su momento y replicarlos no sería posible o llevaría demasiado trabajo. Como enmienda a esta carencia, las gráficas afectadas tienen en el pie o cabeza de figura toda la información que concierne la interpretación de las mismas.

En quinto y último lugar, debido al alcance del proyecto y el enfoque que se le ha dado al resto de subsistemas, el apartado de control no incluye una identificación completa de todos los aspectos planteables del harmonic drive. Las limitaciones de tiempo han hecho que se apunte a un objetivo más acotado y que únicamente se desarrollen conclusiones relacionadas con: el modelado teórico, el control de los diferentes experimentos, el rozamiento estático, algunas estrategias de identificación, la reducción y fluctuaciones en la salida.

5. Planteamiento de soluciones alternativas y justificación de la solución adoptada

Con el motivo de analizar las soluciones posibles para la aplicación del proyecto planteado, se parte de los aspectos generales de los diferentes subsistemas y, en caso de ser conveniente, se plantean alternativas más específicas. Las posibles opciones dentro de cada subsistema pueden abarcar tanto ideas hipotéticas como alternativas implementadas físicamente, habiendo seleccionado la que mejores resultados ofrecía. Por ende, en este apartado se expone y justifica la elección de las diferentes ideas pero solo se detalla en profundidad la solución final adoptada en el siguiente apartado.

5.1. Alternativas en el entorno mecánico

En cuanto al primer subsistema, el montaje mecánico se debe hacer de manera que asegure la máxima estabilidad. De no ser así se podría dar lugar a perturbaciones indeseadas en las mediciones y el entorpecimiento de la identificación. Un ejemplo de usar material poco preciso es cuando se ensambló el prototipo de las primeras pruebas con el encoder magnético (ver figura 5). Esta solución usando Lego no tenía como objetivo ser la definitiva, simplemente era una forma dinámica de trabajar con el sensor antes de su implementación oficial.

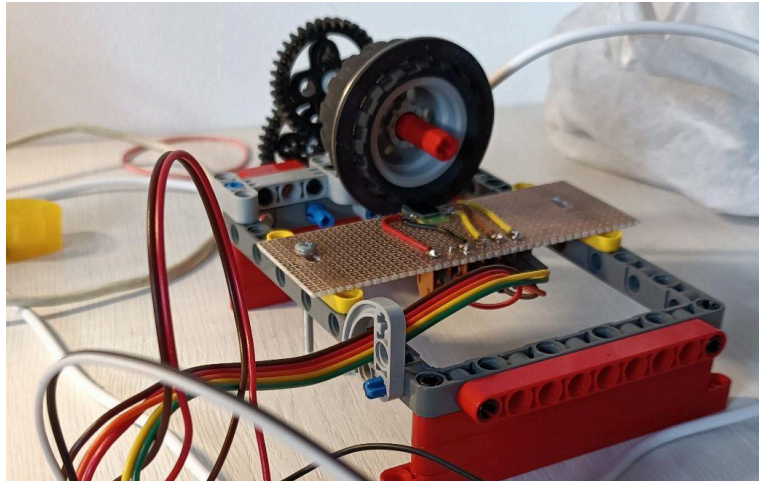


Figura 5: Montaje de primeras pruebas con el encoder magnético

Las consecuencias de usar este tipo de material en los componentes móviles puede generar defectos como los detectados en la figura 6. Aquí se puede observar como la curva de la posición medida presenta una pérdida de datos ya que el valor no cambia como lo que cabe esperar dada la tendencia. Luego, de la imagen 7 se puede ver que cada aproximadamente 581632 cuentas de encoder aparece esta perturbación. Este valor es equivalente a 71 polos del encoder magnético, un polo menos que los que debería contar (cada polo ofrece 13 bits de cuentas). La deducción de todo esto era declarar al encoder magnético como elemento defectuoso ya que un polo concreto no daba señales.

Pero nada más lejos de la realidad, este fenómeno no se trata de un error de electrónica ni de precisión mecánica. Cuando ya se ensambla la primera mesa de experimentación con el harmonic drive y elementos más robustos, las cuentas se adquieren sin perturbaciones ni errores acumulados.

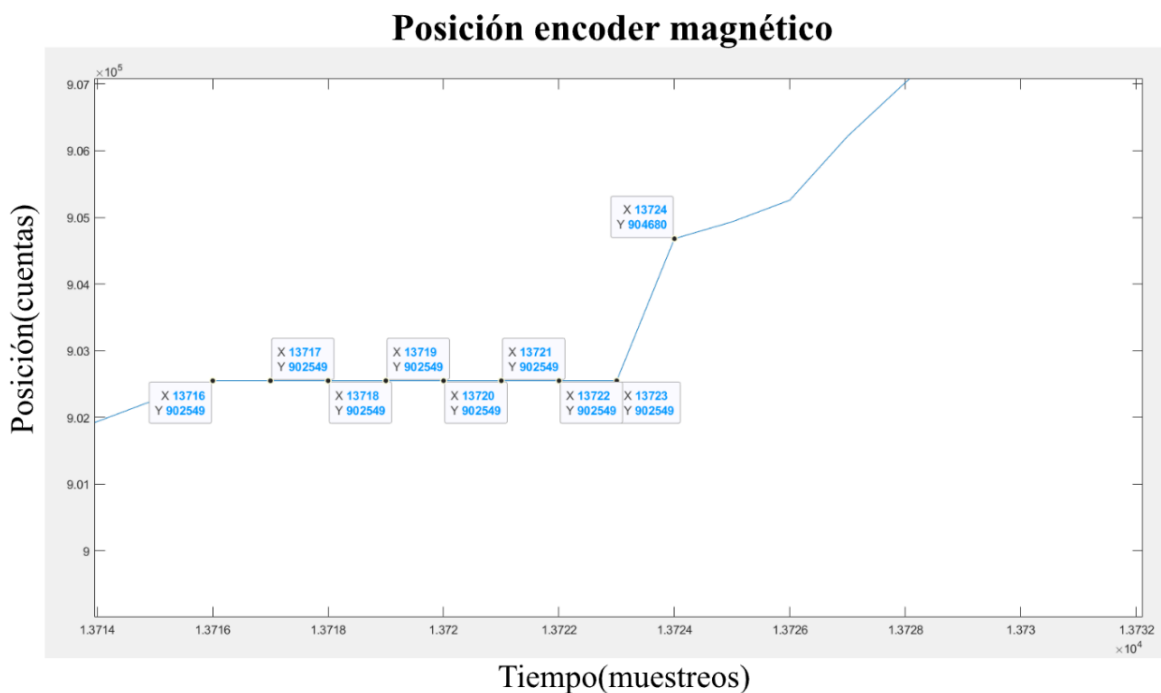


Figura 6: Posición del encoder magnético, detalle de perturbación

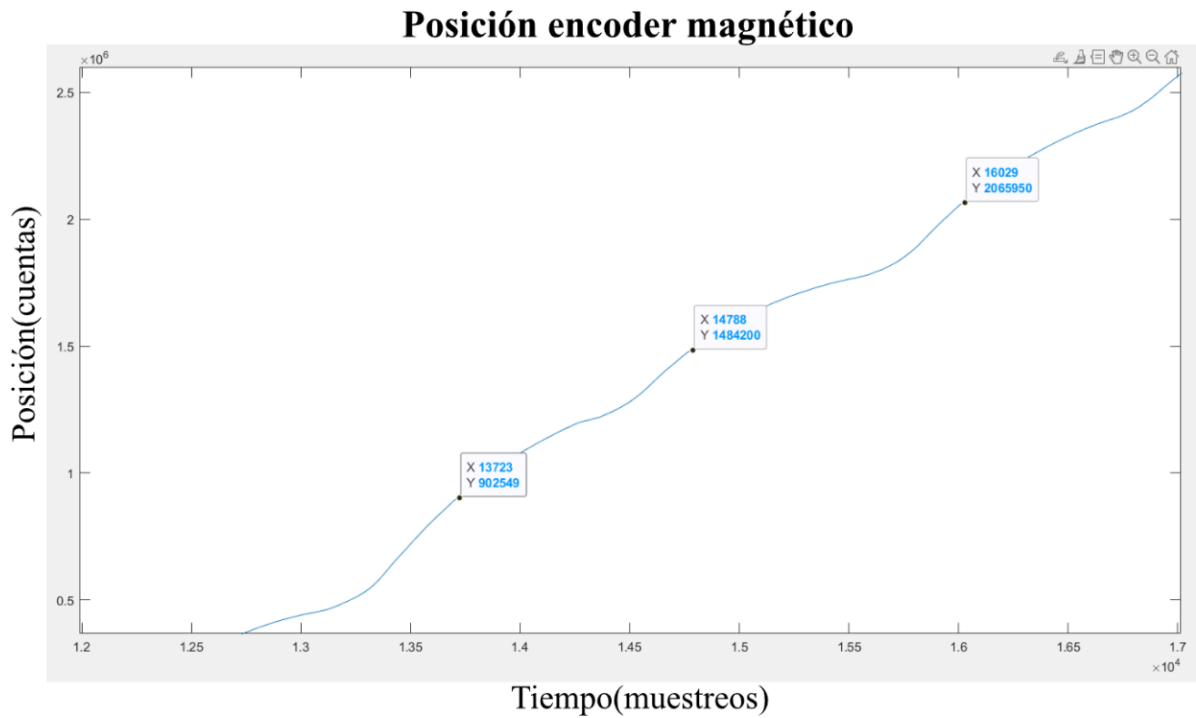


Figura 7: Posición del encoder magnético con marcas en cada perturbación

En la figura 8 se observa la posición del encoder magnético tras 40 vueltas y las cuentas corresponden con lo esperado (40×589824 cuentas) con un error de 0.045° , es decir que, a priori, no existen tendencias de acumulación de error. No obstante, aunque estas perturbaciones se hayan solucionado de esta manera es posible que, en efecto, haya un polo débil en el sensor magnético susceptible a mala estabilidad del montaje.

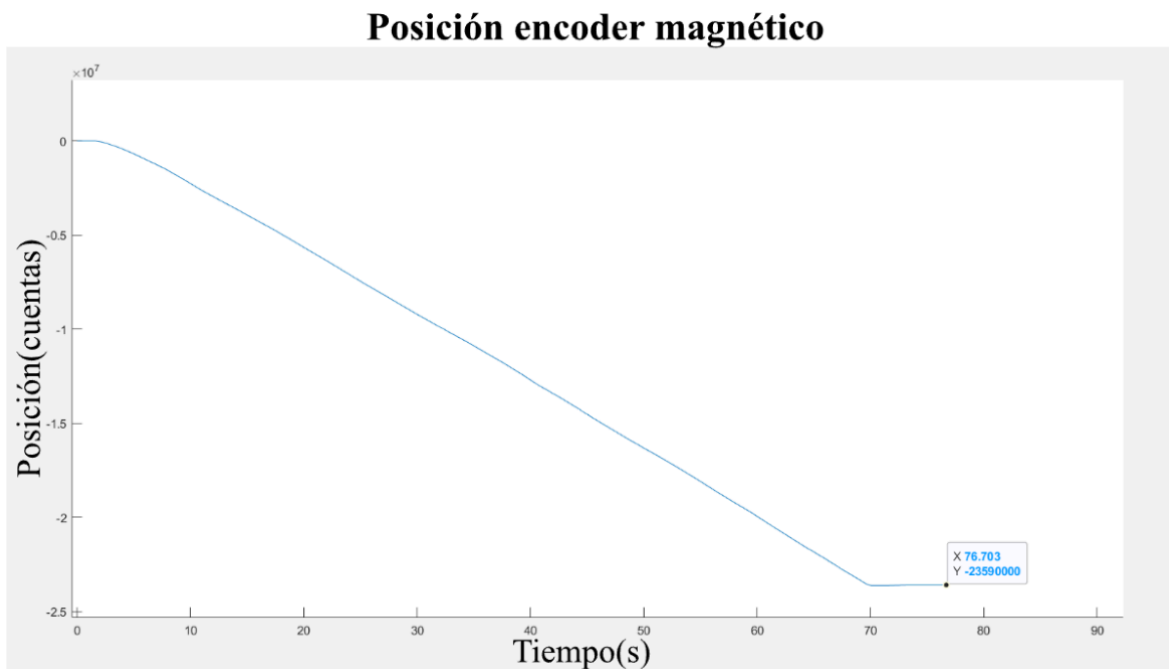


Figura 8: Posición del encoder magnético tras 40 vueltas

Por tanto, una vez evidenciada la necesidad de un montaje mecánico robusto, queda elegir qué materiales usar concretamente para cada soporte y unión. La primera alternativa que se tiene en cuenta es el uso de componentes fiables con la implementación de una base de experimentos basada en piezas de Norelem. Esta empresa especializada en la fabricación y distribución de componentes estándar y soluciones de ingeniería ofrece productos de altísima precisión que se utilizan en una gran variedad de aplicaciones industriales, incluyendo la fabricación, la ingeniería mecánica, la automoción, la robótica y la automatización [14]. La universidad dispone de varios kits de sistemas modulares cuya adquisición data de 1991. A día de hoy algunas de estas piezas del laboratorio ya no se distribuyen ni tampoco se incluyen sus referencias en los nuevos catálogos de la empresa, como el extenso “The big green book” [15] de Norelem.

Una vez se ensambla la mesa de experimentación con las piezas de Norelem llega un punto en que es necesario plantear las alternativas de la unión de los diferentes componentes a la base de pruebas. Utilizar más piezas de Norelem ya no es posible ya que el material disponible en el laboratorio, si bien habían decenas de piezas útiles para la mesa de ensayos, la variedad de piezas más pequeñas es limitada. Por tanto en este punto del proyecto se opta por combinar el montaje del Norelem con piezas auxiliares impresas en 3D diseñadas específicamente para cumplir las necesidades de cada uno de los experimentos objetivo.

La universidad ofrece un servicio de impresión 3D tanto del tradicional FDM (Fused Deposition Modeling) [16] como del innovador SLS (Selective Laser Sintering) [17]. De estas opciones se escoge la segunda ya que ofrece acabados mucho más precisos y el material utilizado en la impresión es el nylon, que tiene una gran resistencia, estabilidad dimensional y amortiguación mecánica [18].

Ahora bien, no todas las piezas auxiliares necesitan ser impresas en 3D. En algunos casos existen alternativas más simples ya que la función mecánica es muy general. El primer caso está en el motor BLDC que está dispuesto con el eje en horizontal gracias a una L de acero y que junto con una lámina de madera se eleva a la altura de trabajo. En el segundo caso, para ajustar la altura del segundo accionamiento, el servomotor, se elige la opción más simple de láminas de aluminio. Un último caso en el que se ha ahorrado la impresión 3D es para la sujeción del lector del encoder magnético que, como no está sometido ni a esfuerzos ni a movimientos mecánicos se ha posicionado con piezas de Lego y Meccano de forma que permitiera ajustar su posición en el lugar correcto. Es importante comentar que todas estas piezas auxiliares están fuertemente unidas a la mesa de ensayos con elementos de Norelem.

Como último análisis de alternativas mecánicas se van a comentar las referentes a los acoplamientos de los ejes. A la hora de conectar un motor a un mecanismo, si aparece alguna excentricidad entre los ejes de conexión, por pequeña que sea, a altas revoluciones puede dar problemas. Es por ello que existen uniones que corrigen estos efectos como la junta cardán, que se adapta a los desvíos de dirección [19] o el acoplamiento Oldham, que compensa las desalineaciones radiales [20]. En la salida del harmonic drive se puede prescindir de este tipo de juntas ya que la velocidad máxima de giro es muy lenta (menos de una vuelta por segundo) y el montaje mecánico se considera suficientemente alineado. De todas formas, en la entrada del mecanismo se trabaja a velocidades de casi 1500 rpm y conviene realizar una unión que absorba posibles imperfecciones del montaje. En la siguiente imagen se ven algunas de las opciones disponibles; desde acoplamientos Oldham hasta juntas flexibles:



Figura 9: Juntas para ejes del laboratorio

Sin embargo, el propio harmonic drive ya cuenta en el primario (figura 10) con un acoplamiento del estilo Oldham. Esta característica está referenciada en la propia ficha técnica del reductor [13] y ofrece la oportunidad de utilizar uniones sólidas que, además, facilitan la identificación del sistema al excluir otras componentes mecánicas articuladas o flexibles. Por tanto y de forma definitiva, se decide diseñar también los ejes de entrada y salida del harmonic drive en 3D con la misma técnica mencionada anteriormente.

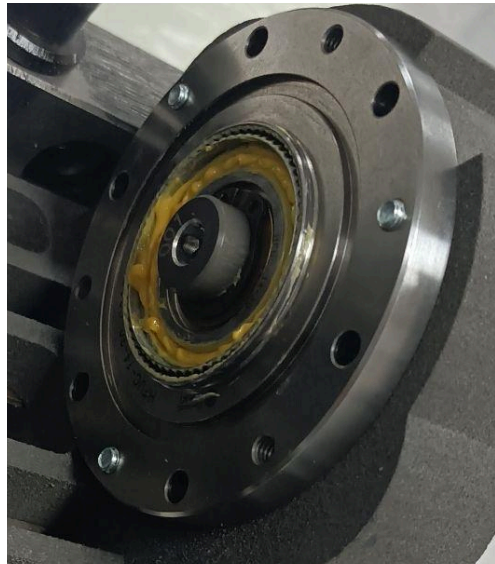


Figura 10: Entrada del harmonic drive

5.2. Alternativas en el entorno electrónico

En primer lugar, el componente electrónico central de todo el proyecto es el módulo de adquisición de datos. Como norma general se busca que la instrumentación electrónica tenga un funcionamiento lo más rápido y fiable posible y la selección de un microcontrolador capaz de cumplir con los objetivos de precisión propuestos es crucial. Como se puede observar en la imagen 11, las primeras pruebas se realizaron con dos módulos NUCLEO-F303K8.

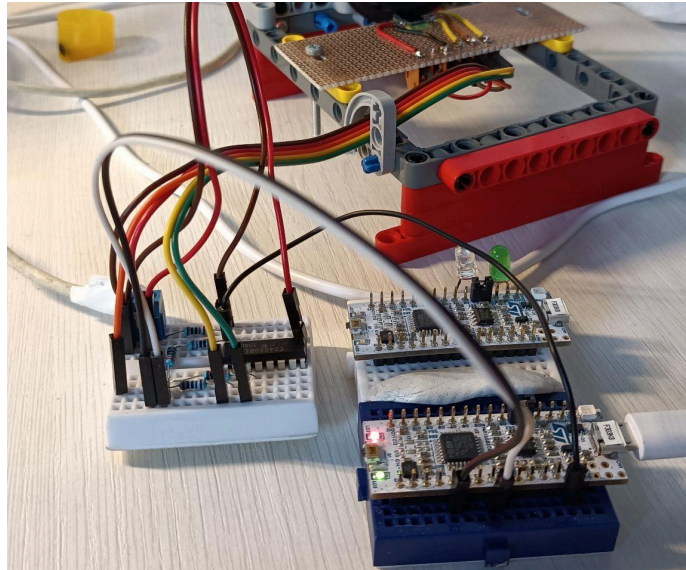


Figura 11: Pruebas con un módulo NUCLEO-F303K8

En la configuración del reloj de este microcontrolador se ha llegado a 64 MHz ya que para alcanzar los 72 MHz máximos se requiere un HSE (High-Speed External oscillator) [21]. Además, existe otra limitación que es la memoria del módulo. Solo tiene disponible una memoria flash de 64 KB y una SRAM de 16 KB. Si se quieren guardar los diferentes datos obtenidos hay que tener en cuenta que, al usar un periodo de muestreo que se supone muy pequeño va a haber un volumen considerable de valores a almacenar. Esta placa, pese a tener buenas prestaciones, no es suficiente para cumplir los objetivos.

En este punto se consideran otras alternativas dentro del material de la escuela. Finalmente se adquiere una placa de la misma familia STM32 que la anterior pero, esta vez, con características superiores, la STM32F429 Discovery kit. Este microcontrolador dispone de 2 MB de memoria flash, 256 KB de RAM y 64 Mb de SDRAM. Además la velocidad máxima del reloj del sistema es 180 MHz y es configurable con un HSE propio de la placa que además ofrece mucha estabilidad temporal.

En segundo lugar, tras la selección de la placa de programación se procede a plantear las alternativas en cuanto a los sensores. Para la implementación de los experimentos es crucial medir la posición del eje de entrada y salida del harmonic drive. El criterio aquí es único: usar los sensores disponibles con mayor precisión. En este caso, para medir la posición de giro existen opciones varias como encoders absolutos, encoders incrementales, resolvers, tacómetros, sensores Hall, etc. Finalmente, los sensores con mayor precisión que se han conseguido son un encoder óptico de 8192 cuentas (13 bits) por revolución y un encoder magnético de 589824 cuentas (72 x 13 bits), ambos encoders incrementales. Estos valores parten de la cuadratura de dos canales que permiten identificar el sentido de giro y cuadruplicar las medidas de los pulsos que detecta cada canal por separado. Esto se consigue ya que los pulsos de cada canal están desfasados 90° (figura 12) y se tiene en cuenta tanto los flancos de subida como los de bajada.

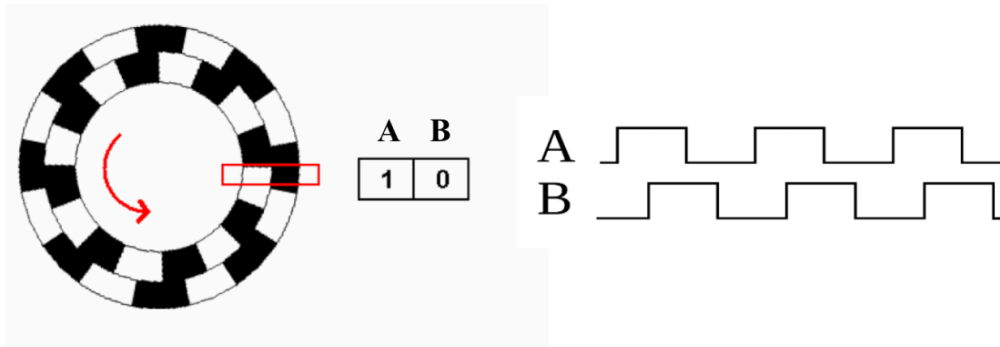


Figura 12: Cuadratura de dos canales de encoder

Tras la obtención de estos componentes de precisión, evidentemente, el encoder magnético se sitúa en el secundario del mecanismo ya que la salida gira 30 veces más lenta que la entrada y, por tanto, se requiere mayor resolución para captar las pequeñas variaciones que ayuden a identificar mejor el sistema.

Un último aspecto a comentar de los encoders es que el magnético dispone de una tercera salida llamada Z. Esta devuelve un pequeño pulso por cada paso por la referencia. Comúnmente la referencia es el paso por 0, es decir, la vuelta completa del encoder. Sin embargo, en este encoder las referencias están situadas en cada polo magnético del disco, por tanto, Z genera 72 pulsos por vuelta. Esto significa que no hay una referencia absoluta la cual podría ser utilizada para un *homing* (vuelta a la posición inicial) fácil de programar en los ensayos. De todas maneras, estas referencias se pueden utilizar combinadas con las cuentas del encoder para afianzar las medidas de posición (un intento de filtro de Kalman incompleto). La decisión final, tras numerosas pruebas combinando las cuentas y gestionando con interrupciones priorizadas, es no dar uso del pin Z. Esta conclusión se basa en que la fiabilidad del canal Z parece la misma (y no mayor) que los canales de cuadratura. Esto en parte se evidencia cuando se miden las cuentas en las primeras pruebas con poca estabilidad y aparecen las perturbaciones de la imagen 6 que provocan la nula medición de un polo entero usando tanto pulsos de Z como los canales de cuadratura.

En tercer lugar, las alternativas disponibles para la selección de los motores encargados de dotar de movimiento al harmonic drive son escasas. En cuanto a la entrada del reductor, se acopla un motor BLDC (brushless direct current) de la marca Maxon ya que las opciones disponibles son motores de menos prestaciones o sin encoder habilitado. Es importante destacar que el encoder incorporado en este motor tiene 13 bits de resolución por lo que en algunos de los experimentos objetivo se puede prescindir del encoder óptico adicional. Por otra parte, para mover el secundario del mecanismo, no es viable dar uso del mismo motor que en el primario por dos razones: la primera es que se trabajan con pares de fuerza 30 veces mayores que en el primario (en concreto se necesitan más de 2 Nm para empezar a mover el harmonic drive desde la salida frente a los 0.405 Nm de par continuo máximo del motor); la segunda razón es que el motor debería de ser capaz de tener la fuerza suficiente para mover sobradamente esta parte del mecanismo ya que, en caso contrario, no se tendría una acción suficientemente excitante para la buena identificación. Tras esta reflexión y aprovechando la disponibilidad de un servomotor de potencia y su respectivo controlador, se decide diseñar los experimentos de la *back drivability* incluyendo las dimensiones de este accionamiento.

En cuarto lugar, una decisión importante es qué controlador o *driver* usar para manejar el motor BLDC, ya que el servomotor ya viene acompañado de este componente de potencia de su misma marca. En una primera instancia, la opción lógica es la misma usada para controlar el mismo motor en [6], que es un controlador de la marca Maxon (igual que el propio motor) cuyo nombre comercial es ESCON Module 50/5 (figura 13). En cambio, pese a que en el *datasheet* se afirma que el módulo soporta 5 A continuos y 15 A de máxima, su

tamaño es muy reducido y no tiene disipadores de calor. Esto plantea la indagación de si es mejor para el desarrollo del proyecto contar con un controlador más confiable y, efectivamente, en la propia ficha técnica se plantea la opción de un *shield* o placa base donde incorporar el módulo (figura 14) que sí corresponde con una etapa de potencia robusta.



Figura 13: ESCON Module 50/5 usado en [6]

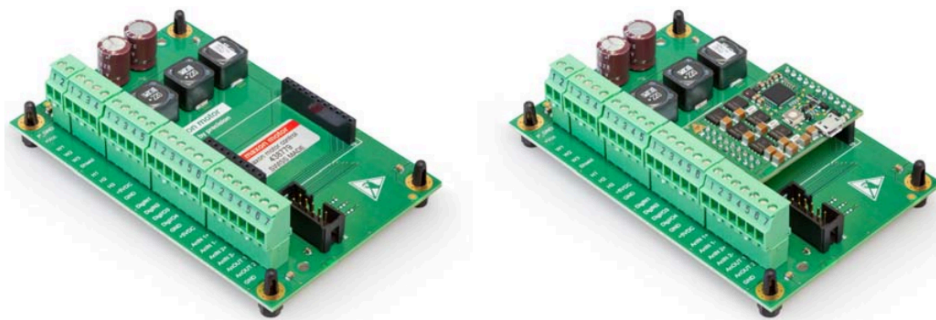


Figura 14: ESCON Module 50/5 incorporado a segunda etapa de potencia [22]

Tras las últimas consideraciones, finalmente se selecciona una variante del anterior módulo que engloba ambas etapas de potencia y que está disponible en la propia universidad. Se trata de la controladora ESCON 50/5 y su implementación, como la del resto de componentes, se explica en el apartado descripción detallada de la solución adoptada.

En quinto lugar, conviene hablar de las diferentes soluciones para conseguir un buen acondicionamiento de cada señal. Por un lado se tienen las señales digitales que tienen como salida los diferentes encoders. Cada uno tiene dos canales, como se ha comentado antes, y las salidas devuelven “0” o “1” lógico en el caso de detectar un pulso. El problema reside en que la lógica del encoder magnético y el del motor BLDC funciona a 5 V mientras que la Discovery lo hace a 3 V. Para reducir estas tensiones se optó en un primer lugar por un chip CD4050B [23] ya que se pensaba que a través de sus *buffers* internos se llegaría a traducir una señal lógica a otro nivel de tensión (figura 11). Pero este módulo no tiene la función esperada, más bien adapta las tensiones de la salida según la carga como se explica en [24]. Tras esta causa, se procede a dar uso de un divisor resistivo a pesar de su poca eficiencia y pérdidas de corriente para la entrada de la señal a la Discovery. Esta solución en un principio provisional permitió las primeras pruebas con los encoders y los resultados fueron satisfactorios.

Sin embargo, cuando se detectaron las perturbaciones vistas en el anterior apartado, donde parece que el encoder magnético falla por cada vuelta en uno de sus polos (hay que recordar que en un primer momento esto parece un fallo electrónico pero luego el problema se solventa con el montaje mecánico final), se plantea que el error puede residir en el acondicionamiento. En este punto se dio uso de amplificadores operacionales (AO) en forma de comparador o de atenuadores (amplificadores con ganancia entre 0 y 1). Algunas de las pruebas realizadas se pueden ver en la figura 15.

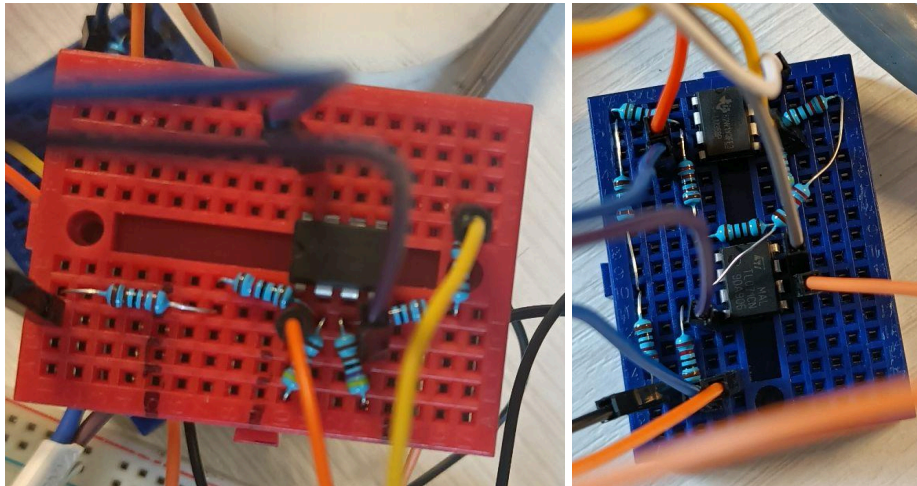


Figura 15: Pruebas con amplificadores para la señal del encoder magnético

También se hicieron pruebas usando amplificadores como *buffers* en el propio divisor resistivo para aumentar la corriente de entrada a la Discovery. Sin embargo, tras todo lo mencionado, parece que el uso de amplificadores no ofrece resultados satisfactorios. En la figura 16 se ve el resultado de medir un movimiento del encoder magnético de un lado a otro y parada en el lugar de inicio. Como se observa, no se han detectado correctamente los pulsos del encoder ya que hay una deformación de las oscilaciones (que se esperan medianamente uniformes pese haberse hecho a mano) y la posición final está lejos de situarse como al inicio. Como deducción de todo esto, el problema puede deberse a la latencia presente en los propios AO ya que una pequeña variación en la señal puede obstaculizar la detección los flancos de subida y bajada de todos los pulsos del encoder, sobretodo de un sensor con más de medio millón de cuentas por vuelta. Por tanto, se elige la alternativa de los divisores resistivos para acondicionar las señales de los encoders magnético y del motor ya que no generan errores acumulados en comparación con los anteriores.

Antes de pasar al siguiente aspecto, cabe mencionar que el encoder óptico adicional no necesita acondicionamiento ya que la tensión de su señal es igual a la de alimentación y este se puede alimentar con la misma lógica que la Discovery.

Posición encoder magnético

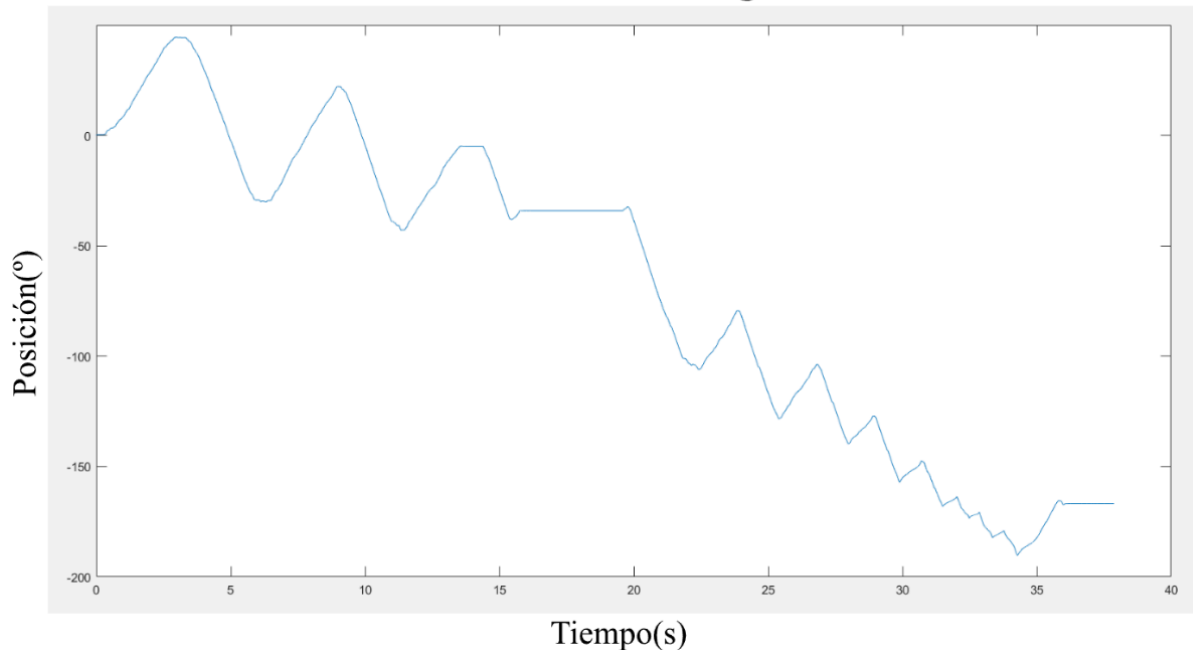


Figura 16: Posición del encoder magnético usando amplificadores

Por otro lado, hay que hablar de las alternativas para tratar las diferentes señales analógicas de los experimentos. La primera señal es la de control o de referencia que genera la Discovery que tiene un rango de 0 a 3 V mientras que los controladores de potencia tienen unas entradas analógicas de -10 V a +10 V. Esto significa que se pueden conectar directamente pero habrá una pérdida de la calidad de los datos ya que la máxima resolución en la entrada analógica de los controladores se consigue al dar uso de todo el fondo de escala [25]. Por tanto, se descarta la opción de conectar directamente y se implementa un amplificador que multiplique por 6.66 la señal. Además, la ESCON 50/5 tiene entradas analógicas diferenciales, es decir, que por un canal se puede introducir la señal y por otro una tensión constante (10 V en este caso) que desplace y centre la entrada. En adición a esto, se coloca un pequeño condensador a la entrada del AO junto con una resistencia para filtrar altas frecuencias que pueda contener la señal de control (por ejemplo con el uso de modulación de ancho de pulso PWM). En la imagen 17 se visualiza el estado de este acondicionamiento en el momento de su implementación.

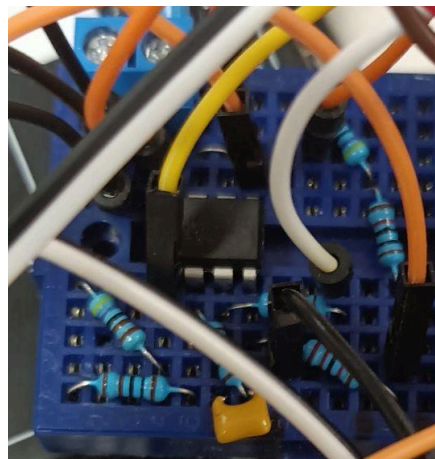


Figura 17: Amplificación de la señal de referencia con filtro

A pesar de todo lo comentado, esto resulta no ser suficiente para asegurar una señal de calidad. Resulta que, en la experimentación, al comparar los datos teóricamente enviados con los recibidos por la controladora se percibe cierto offset variable. Finalmente esto se soluciona aislando las etapas usando buffers para cada señal diferencial, aumentando la impedancia de entrada al amplificador con una resistencia de gran valor y quitando el condensador.

La decisión final de quitar el condensador no compromete a la experimentación porque, en el caso de que se utilice PWM, la naturaleza del propio motor ya actúa como un filtro para esta señal cuadrada por el bobinado de este. También comentar que, la inclusión de un filtro puede sumar un orden indeseado al sistema a identificar.

La segunda señal analógica es la corriente medida del motor. En el caso de la controladora del motor BLDC, se tiene una salida analógica programable para que devuelva el valor real o promedio de la corriente. En cualquier caso se tienen 12 bits distribuidos en un rango de -4 V a 4 V [25], lo que significa que conviene atenuar y elevar la señal para que la Discovery pueda aprovechar toda la resolución. En la figura 18 se puede ver la conexión de un amplificador operacional en modo diferencial que comprara la señal de la ESCON con un valor fijo de 4 V el cual está siendo alimentado con una fuente simétrica gracias al equipo del laboratorio. La razón por la que se descarta la implementación de un amplificador de instrumentación para esta señal diferencial es porque en este caso se necesita una ganancia menor que 1 para atenuar la señal y, por lo general, los de instrumentación solo amplifican.

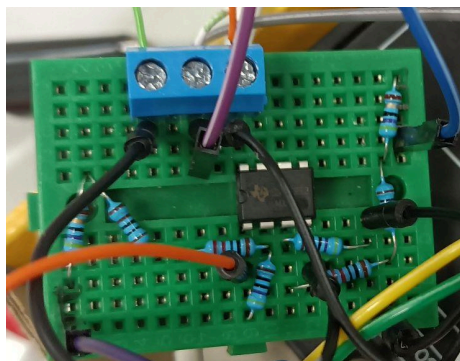


Figura 18: Amplificación diferencial de la señal de corriente

Por tanto, dada la falta de un buen aislamiento entre etapas de medición por usar un único AO esto se traduce en un mal comportamiento de la señal obtenida, habiendo mucho más ruido en la corriente. El ruido es normal en las señales de corriente y, en general, es bueno para la identificación pero en este caso parecen interferencias que nada tienen que ver con el motor, se deduce que es el AO el que acababa amplificando las frecuencias de ruido de manera desmesurada. Tras intentar aislar las etapas utilizando *buffers* y no encontrar solución, se procede a la conexión directa de la Discovery y la controladora. De esta manera se obtiene una señal más limpia pero con menos de la mitad de resolución (véase figura 19).

Corriente real (azul) y de referencia (naranja)

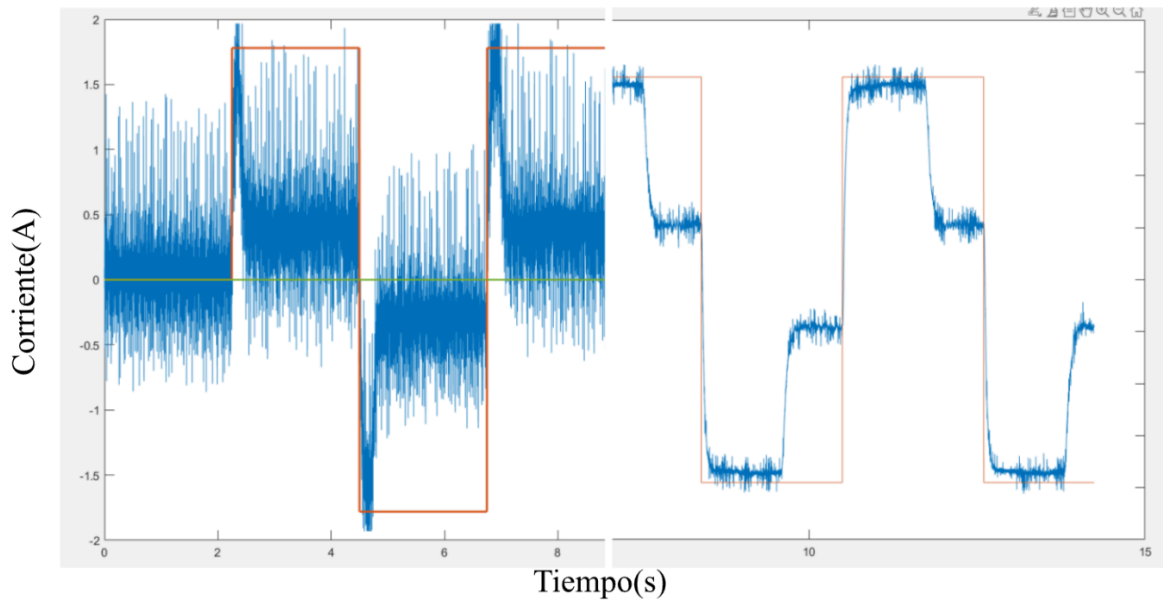


Figura 19: Corriente usando AO y buffers (izq.) frente a conexión directa (der.)

Por último, queda abordar la cuestión de la alimentación eléctrica de todos los componentes ya que, tras numerosos ensayos, ha sido motivo de errores en la experimentación. Conviene, por tanto, exponer cuál ha sido la mejor opción para alejarse de problemas derivados de este aspecto.

En los primeros ensayos, se utilizaba una sola fuente de alimentación (Ruzizao) como se puede ver en la figura 20. La fuente es regulable hasta los 60 V y puede alcanzar 5 A de trabajo. Esta alimentaba a la controladora con 36 V (que es lo que necesita el motor BLDC) y el resto de tomas eran derivadas de esta (menos la Discovery que también puede alimentarse por el puerto USB).

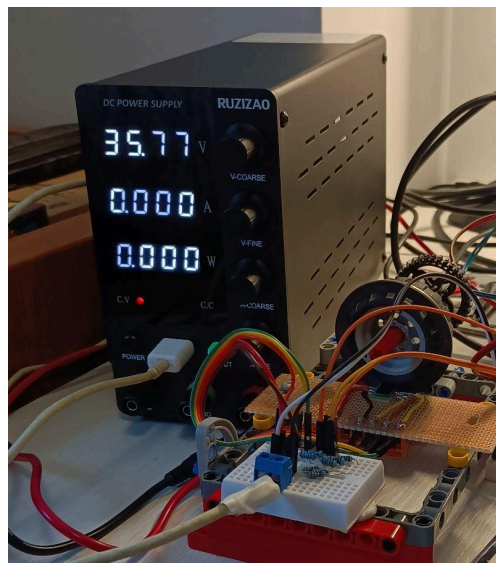


Figura 20: Primeros ensayos con fuente Ruzizao

Tras algunas pruebas con acciones de control excitantes, que exigen cambios de corriente bruscos, se observa en la figura 21 que la propia señal de referencia sufre deformaciones

ya que se espera una señal perfectamente cuadrada. Como el AO que amplifica la señal de control se alimenta con los 36 V de la fuente de alimentación y se detectaron con un multímetro pequeñas variaciones en esta en los picos de corriente se dedujo que el problema se originaba de una mala alimentación, que compensaba la demanda de corriente con un aumento de tensión.

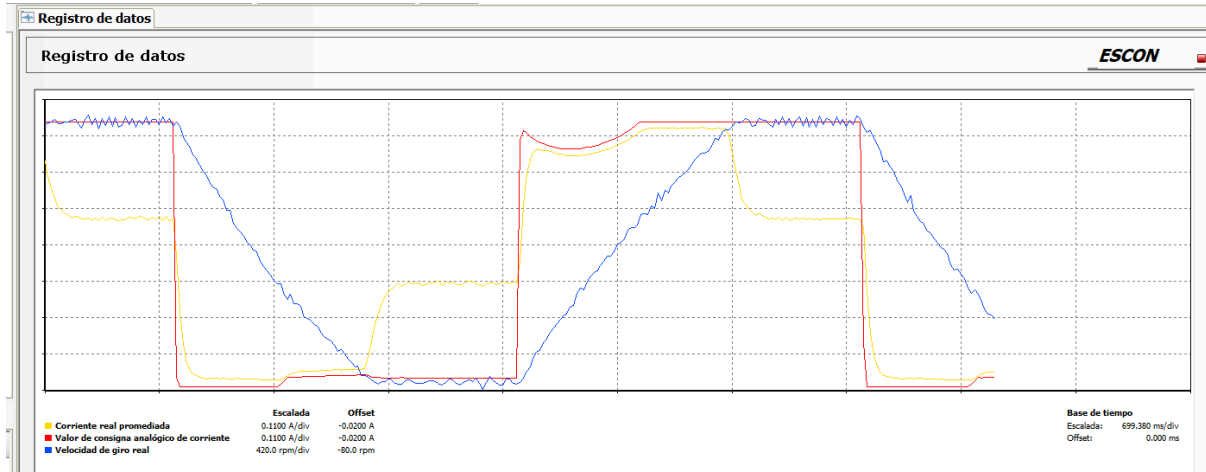


Figura 21: Captura de datos deformes en ESCON Studio

Primero se conectaron unos condensadores de gran capacidad en la salida de la fuente para ver si eran capaces de compensar los picos de corriente, pero no tuvo resultados notorios. Luego se sustituyó la fuente por otra (PULS) del laboratorio pero también deformaba la señal. La solución final para que no haya perturbaciones en la circuitería es utilizar una fuente (Ruzizao) para alimentar la etapa de potencia y otra diferente (PULS) para la amplificación y el resto de circuitos.

5.3. Alternativas en el entorno de programación

Las opciones de programación en cuanto al entorno de desarrollo de la placa STM32F429 Discovery kit son el STM32CubeMX (para la configuración de la placa) junto con el Keil (para la programación) o el STM32CubeIDE que engloba los dos aspectos en un solo programa. Por simplicidad y por ser el más actualizado para el desarrollo de proyectos se escoge la segunda opción. En cuanto a computación, configuraciones de *hardware* y posproceso también se da uso de otros programas como Matlab, PuTTY y ESCON Studio de los cuales se hablará más adelante. Para empezar se van a justificar las decisiones dentro de la programación de STM32CubeIDE la cual ha evolucionado con cada versión de código (figura 22), empezando con simples pruebas con interrupciones y mediciones, pasando por alternativas finalmente descartadas, que se comentan en este apartado, y llegando a la versión final que se describe en detalle en el apartado 6.3.

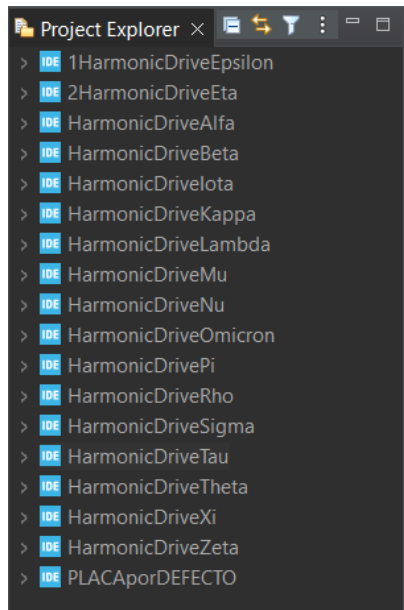


Figura 22: Versiones de configuración y programa en STM32CubeIDE

Para empezar, se va a exponer la justificación de por qué se opta por una transmisión serie en tiempo de ejecución de los experimentos ya que, en definitiva, es el aspecto que más puede comprometer a la calidad de las mediciones. Partiendo de que la programación mediante interrupciones temporizadas es una opción más dinámica que el método de *polling*, la idea inicial era dar uso de la memoria de la Discovery para almacenar a gran velocidad los datos de la identificación y minimizar el periodo de muestreo. Luego, estos datos se enviarían a través del puerto serie al ordenador a una velocidad evidentemente menor ya que la comunicación serie es más lenta que asignar unos valores a un espacio de memoria.

Con esta idea en mente y sabiendo que la Discovery dispone de 64 Mb de SDRAM la expresión que se puede deducir para calcular el tiempo que puede durar cada experimento es:

$$Duración(s) = \frac{Memoria(b) \cdot Tm(s)}{Paquete(b)} \quad (1)$$

Si se considera un paquete de datos de 10 bytes (10 x 8 bits) y un periodo de muestreo de 60 microsegundos, la Discovery puede guardar 48 segundos de experimento. Esta duración es más que suficiente, incluso se podría plantear aumentar más la frecuencia de muestreo pero esto no sería conveniente ya que la conversión ADC, de la cual se habla más adelante, precisa de aproximadamente 60 microsegundos para completar la medición.

Pero en el intento de utilizar la SDRAM, configurando todos los parámetros y periféricos tal como se recomienda en [26], aparece un problema inesperado. Los valores guardados en los espacios de memoria se alteran con el tiempo, llegando a un número estable decimal de 255. Este número corresponde a FF en hexadecimal u 8 valores lógicos a nivel alto, lo que significa que, de alguna manera, los espacios en memoria se rellenan de “unos” con el paso de unos milisegundos. Se puede deducir que hay un problema con el refresco de los condensadores que mantienen los datos guardados durante la ejecución del programa y, tras algunas pruebas modificando valores de la configuración de este periférico, no se ha

conseguido que la SDRAM funcione correctamente. En la imagen 23 se puede ver el resultado de guardar en tres espacios de memoria SDRAM el valor 1 en decimal y como, tras consultar estos valores periódicamente, se percibe esta supuesta corrupción en la memoria.

Data_2024-04-13_122300.txt: Bloc de notas					Data_2024-04-13_122300.txt: Bloc de notas					Data_2024-04-13_122300.txt: Bloc de notas				
Archivo	Edición	Formato	Ver	Ayuda	Archivo	Edición	Formato	Ver	Ayuda	Archivo	Edición	Formato	Ver	Ayuda
1	1	1			31	253	31			253	255	253		
1	1	1			253	31	253			255	253	255		
1	1	1			31	253	31			253	255	253		
1	1	1			253	31	253			255	253	255		
1	1	5			31	253	31			253	255	253		
1	5	1			253	31	253			255	253	255		
5	1	5			31	253	31			253	255	253		
1	5	1			253	31	253			255	253	255		
5	1	5			31	253	31			255	255	255		
1	5	1			253	31	253			255	255	255		
5	5	5			31	253	31			255	255	255		
5	5	5			253	31	253			255	255	255		
5	5	5			31	253	31			255	255	255		
5	5	5			253	95	253			255	255	255		
5	5	5			95	253	95			255	255	255		
5	5	5			253	95	253			255	255	255		
5	5	5			95	253	95			255	255	255		
5	5	5			253	95	253			255	255	255		
13	5	13			95	253	95			255	255	255		
5	13	5			253	95	253			255	255	255		
13	5	13			95	253	95			255	255	255		
5	13	5			253	95	253			255	255	255		
13	5	13			95	253	95			255	255	255		
5	13	5			253	95	253			255	255	255		
13	5	13			95	253	95			255	255	255		
5	13	5			253	95	253			255	255	255		
13	5	13			95	253	95			255	255	255		
5	29	5			253	95	253			255	255	255		
29	5	29			95	253	95			255	255	255		
5	29	5			253	95	253			255	255	255		
29	5	29			95	253	95			255	255	255		
5	29	5			253	95	253			255	255	255		
93	5	93			223	253	223			255	255	255		
5	93	5			253	223	253			255	255	255		
93	5	93			223	253	223			255	255	255		

Figura 23: Evolución del valor de datos guardados en SDRAM

Antes de plantearse la opción de adquirir un nuevo módulo de adquisición de datos, es conveniente seguir con la mejora del código y comparar los resultados. En este punto se decide concentrar los esfuerzos en conseguir un código optimizado que se ejecute en el menor tiempo posible y consiga enviar los datos por el puerto serie a la máxima velocidad permisible.

En cuanto a este último objetivo, en primer lugar se va a medir el valor analógico de la corriente del motor de la forma que mejor rendimiento ofrezca al resto del código. Tras una serie de pruebas y con la configuración de máxima frecuencia del reloj del sistema se consigue una conversión ADC de 12 bits en 60 microsegundos. No obstante, si se configura una llamada a la interrupción cuando la conversión haya finalizado, se evita la espera en la ejecución del programa. De todas formas, la conversión sigue tardando 60 microsegundos pero, esta vez, el periférico trabaja en segundo plano.

En segundo lugar, se escoge la mejor opción de transmisión y recepción de la comunicación serie. Por un lado, la velocidad máxima de baudios que puede manejar la Discovery es de 11.25 Mb/s en el USART1 (Universal Synchronous and Asynchronous serial Receiver and Transmitter), una frecuencia elevada y muy por encima de lo que los programas de ordenador se pueden permitir, presentándose aquí un verdadero cuello de botella. En la imagen 24 se puede ver como se han instalado y comparado seis de los programas más extendidos para comunicación serial y otras (Realterm, Termite, PuTTY, CoolTerm, Serial Port Monitor y Hercules).

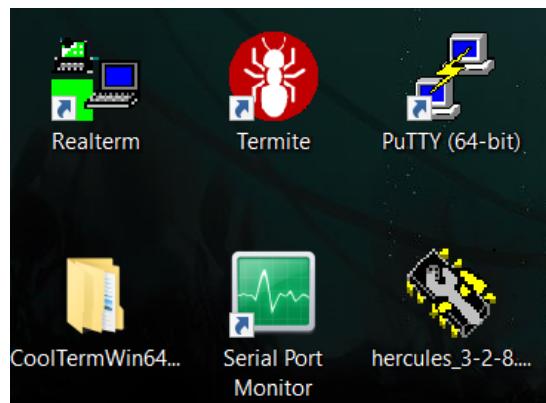


Figura 24: Programas utilizados para comunicación serie

Se ha podido comprobar que el PuTTY puede trabajar con la tasa más alta de entre los programas, siendo 2 Mb/s el máximo valor que no presenta errores de recepción de los datos. A través de la expresión siguiente se puede calcular el valor máximo teórico del paquete de datos que se puede enviar en un cierto periodo de muestreo y tasa de baudios:

$$Paquete(b) = Baudios(b/s) \cdot T_m(s) \quad (2)$$

Si se considera el periodo de muestreo mínimo de 60 μ s y los máximos baudios del PuTTY se obtiene un tamaño de paquete máximo de 120 bits (15 bytes). No obstante, estos valores estimados están justo en el límite de fallo; más adelante se expone qué margen de seguridad se deja para evitar fallos de comunicación. También es importante comentar que el PuTTY presenta características en la interfaz de usuario muy variadas y, la más destacable, es poder configurar un autoguardado de los datos recibidos por su terminal. Estos se pueden guardar en documentos de texto con nombres únicos (asignando el día, hora, minuto y segundo en el nombre) por cada comunicación lo que facilita la ejecución de muchos experimentos seguidos y la automatización de parte del proceso de adquisición de datos. La duración de los experimentos no tiene límite más que el espacio disponible en el ordenador para los documentos de datos.

En este punto conviene decidir qué estructura va a tener cada paquete de datos de forma que sea lo más compacta posible sin comprometer a la adquisición. Si se parte de un esqueleto en formato CSV (Comma-Separated Values) como el siguiente:

$$datos = [tiempo,referencia,encoder1,encoder2,corriente\r\n] \quad (3)$$

Se puede hacer una estimación aproximada de cuánto ocupa esta opción teniendo en cuenta que el formato de transmisión y recepción de datos por el puerto serie se basa en caracteres en código ASCII, los cuales pueden adquirir 127 valores (7 bits) [27]. En la práctica, se considera un bit adicional para cada carácter enviado llamado bit de *stop* para identificar dónde acaba cada carácter por lo que cada dígito dentro de la cadena de datos

representa 8 bits (1 byte) [28]. También existe la opción de ASCII extendido [29] que dispone del doble de caracteres añadiendo un bit pero tras algunas pruebas se ha comprobado que la configuración de STM32Cube precisa de un bit de parada como mínimo y al enviar caracteres especiales del ASCII ampliado la comunicación serial da uso de dos bytes.

Con esta contextualización, al rellenar inicialmente una cadena de caracteres como en (3) resulta en: 32 bits de tiempo (10 caracteres), 12 bits de referencia (4 caracteres), 32 bits de encoder1 (10 caracteres), 32 bits de encoder2 (10 caracteres), 12 bits de corriente (4 caracteres), 4 comas (4 caracteres), retorno de carro y salto de línea (2 caracteres). Todo ello suma un total de 44 dígitos, es decir, 44 bytes que se deberían transmitir por cada periodo de muestreo. Evidentemente esto se da en el caso más extremo de uso de datos absolutos ya que se ha considerado la máxima capacidad numérica para el tiempo y los encoders.

Por tanto, en este punto se plantea mejorar el paquete simplificando los datos pero sin llegar a perder nada de información de los experimentos. La primera medida es eliminar el dato del tiempo, esto se justifica porque se considera conocido el periodo de muestreo. Por tanto, cada paquete de datos recibido va a estar siempre a una distancia temporal igual al periodo de muestreo y esto se puede posprocesar una vez recibidos los datos para añadir una columna de las marcas temporales.

La segunda medida es trabajar con datos relativos en lugar de absolutos, de esta manera los números no tienen porqué ser tan extensos. Para hacer esto, el dato a manipular debe de tener cambios relativamente graduales, por tanto, se excluyen la referencia y la corriente ya que pueden dar saltos que hagan perder el hilo del posproceso. La idea es que los datos (ahora ya solo de encoders) se envíen con una cuenta que use menos dígitos y que el posproceso se encargue de sumar o restar gradualmente según convenga para general las posiciones absolutas. En la figura 25 se ve un ejemplo de transformación de los datos de posición recibidos a absolutos. En el apartado 6.3 se describe con detalle esta estrategia.

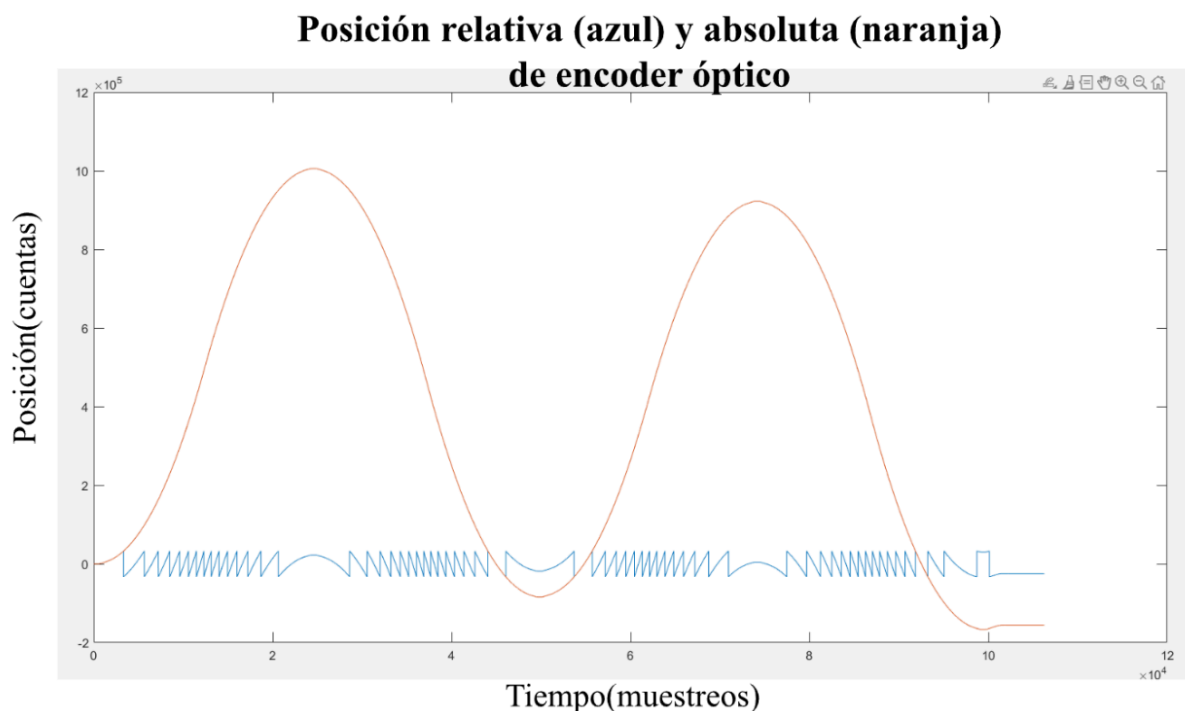


Figura 25: Resultado de obtener valores absolutos a partir de relativos

Ahora conviene conocer cuál es el máximo salto que pueden dar las cuentas de los encoders para seleccionar un mínimo número de datos relativos que asegure no perder información. En la siguiente expresión se puede calcular el número de cuentas por medición dependiendo de la velocidad de giro, de la resolución del encoder y del periodo de muestreo:

$$Cuentas/Medición = \frac{Velocidad(rpm) \cdot Resolución(cuentas) \cdot Tm(s)}{60} \quad (4)$$

Si se considera una velocidad igual a la máxima del motor (1560 rpm), la resolución del primer encoder (8192 cuentas) y un periodo de muestreo de 60 μ s; se obtienen unas 13 cuentas por medición para el primer encoder. Para el segundo encoder se toma una velocidad 30 veces menor, dada la reducción, y la resolución de este (589824 cuentas); se obtienen unas 31 cuentas por medición. Esto significa que, en un principio, si se utilizan valores de posición relativa de 2 dígitos, se asegura una compresión de los datos muy significativa (5 veces menos tamaño que antes) sin riesgo de perder información.

Ahora bien, con el objetivo de compactar todavía más los datos, se propone dar uso del máximo número de caracteres posible por cada dígito. Hasta este punto solo se empleaban números decimales del 0 al 9 por cada espacio de la cadena de caracteres pero esto se puede mejorar. Como se ha comentado antes, el código ASCII incorpora 127 valores que, en forma de caracteres, incluye números, letras, símbolos y comandos de control [29]. Tras algunas pruebas se concluye que se puede dar uso de 91 valores seguidos del código ASCII. Se descarta el resto ya que los caracteres de control no son válidos para transmitir datos ya que tienen otras funciones; los valores 34 y 126 dan problemas en el posproceso ya que Matlab los interpreta incorrectamente. Se escoge, por tanto, desde el carácter 35 hasta el 125 (de “#” a “}”) ya que los valores seguidos facilitan la programación.

A la hora de aprovechar este nuevo abanico de símbolos, se procede a coger las dos primeras cifras (las de mayor peso) del número decimal de cada dato y mapear su posición en el código ASCII con un *offset* de 35. Esas dos cifras decimales pueden ir desde 00 hasta 90 y el resto de cifras no serán convertidas de esta manera para que puedan tomar valores de 0 a 9 cada una. El resultado de todo esto es que se pueden enviar decimales de 2 dígitos (máximo 91 valores) con un byte, de 3 dígitos (máximo 910 valores) con 2 bytes y de 4 dígitos (máximo 9100 valores) con 3 bytes. Evidentemente, en todo momento se comprueba cuántos microsegundos tarda en ejecutarse cada acción del código temporizado para comparar alternativas. El proceso de conversión ASCII descrito demora 3 μ s mientras que los bytes de datos adicionales transmitidos suman aproximadamente 5 μ s cada uno (20 μ s entre los cuatro datos). Se concluye, por tanto, que el código es más veloz con la solución implementada. Este proceso de compresión se detalla en el apartado 6.3.

Como últimas medidas de compresión, se eliminan las comas ya que en el posproceso se puede deducir a qué dato pertenece cada dígito, y se sustituyen el retroceso de carro y el salto de línea por un espacio. De esta manera, en un principio se pueden enviar 8 bytes de datos separados por un byte que se almacenan en documentos generados por el PuTTY en formato de una sola línea que luego Matlab transforma a columna, separa por variables y descomprime el ASCII.

Tras aplicar todo lo expuesto resulta que hay que ampliar el periodo de muestreo porque se trabaja muy cerca del límite de fallo. Los errores de transmisión o recepción de datos a estas altas velocidades pueden darse por causas que conciernen a la Discovery o, más probablemente, al programa PuTTY. Estos errores son fácilmente detectables ya que, si la cadena de caracteres sufre cambios en su tamaño, el código de posproceso diseñado en Matlab lo puede identificar (figura 26). Por otro lado, si un dato se corrompe y cambia su valor manteniendo el tamaño de la cadena, sólo puede identificarse con la interpretación de los datos una vez procesados. Es decir, que si en la curva de un dato aparecen valores pico

que no tienen sentido (no ocurre) se puede identificar este tipo de fallo. También es importante dar uso de la función “fflush” para forzar la transmisión de todo el contenido del buffer de salida en cada periodo. De lo contrario, los datos se envían a trozos como se aprecia en la figura 27.

```

Index exceeds the number of array elements. Index must not exceed 7.

Error in descomprimirASCIIQuad (line 36)
    valor(8) = double(cadena(8))-48;

Error in ScriptoRomulusQuad (line 21)
    A = descomprimirASCIIQuad(Aascii);

```

Figura 26: Error de dato recibido en el posproceso de Matlab

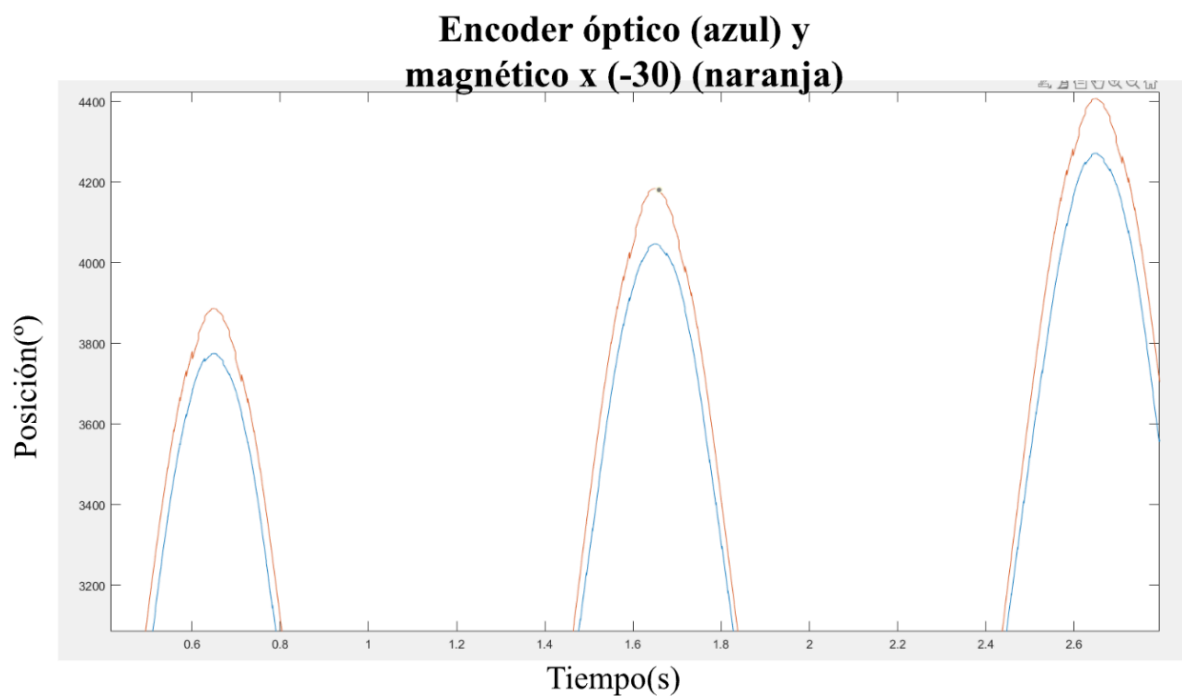


Figura 27: Posiciones de encoders sin usar “flush”

Por tanto, se amplía el periodo de muestreo hasta que se asegure que no aparece ningún error en la recepción de datos. De esta manera, se alcanzan pues 150 μ s de periodo de muestreo y, con ello, se debe ampliar el *counter period* del encoder magnético ya que, al disminuir la frecuencia de muestreo, ahora hay riesgo de contar saltos con valores más altos que en la primera suposición. El resultado es un paquete de datos de 10 bytes (9 bytes de datos y un byte de espacio separador) que se envía cada 150 microsegundos.

El último aspecto a comentar de este apartado es la referencia de corriente o acción de control: su cálculo y aplicación. Por un lado, hay que disponer de una gran variedad de acciones de control para una buena identificación y para ello una buena opción es implementar una función que contenga todos los modos y se pueda elegir entre ellos con un argumento de entrada. También es importante generar números aleatorios para mayor excitación. Para evitar utilizar cadenas de números pseudoaleatorios [30], las cuales pueden determinarse y no ofrecer el suficiente desorden, se decide utilizar el periférico RNG (Random Number Generator) de la STM32. Este generador de números aleatorios utiliza una fuente de ruido analógico para la obtención de valores de 32 bits completamente

impredecibles y tarda un microsegundo en completar su proceso. El cálculo de la referencia puede demorarse 73 μ s en el peor caso que es en la suma de tres senoidales aleatorias.

En este punto es conveniente intentar trasladar el cálculo fuera del tiempo de ejecución. El uso de la SDRAM se ha visto que no es una opción pero todavía quedan el resto de memorias (que tienen mucha menos capacidad). Tras llevar a cabo diferentes pruebas (figura 28) se ha comprobado que el valor límite del periodo de muestreo sigue siendo 150 μ s, con o sin la inclusión del cálculo de la acción de control. La diferencia es que, si se depende de la memoria, la duración de los experimentos se ve limitada. Como en el código temporizado hay tiempo suficiente para la incorporación de este cálculo, se decide trabajar de esta manera.

```
387 while(i < 90000){//90000--> 13.5 s (periodo de 150 micros)////9s (periodo de 100 micros)
388     controlCalc(MODO_SENOIDAL, 1, &duty_array[i]);//segundo arg. 20000 --> 20000*150us = 3s
389     //duty_array[i] = 4096 - 360;
390     i++;
391 }
392 i = 0;
393 HAL_TIM_Base_Start_IT(&htim8);
```

Figura 28: Cálculo de la acción de control antes del tiempo de experimentación

Finalmente, la aplicación de esta acción de control tiene dos alternativas: PWM y DAC. Por un lado, al dar uso de una modulación de ancho de pulso (PWM), se perciben oscilaciones en el ESCON Studio como se puede ver en la figura 29. En un principio esto puede parecer un defecto importante pero solo es debido al muestreo del programa ya que, al cambiar este cambia la forma de la onda y cabe recordar que, en esencia, el PWM es una señal cuadrada a alta frecuencia. Además, la regulación de corriente no intenta seguir estas variaciones, lo que significa que solo es un problema de visualización. La modulación por ancho de pulso es muy utilizada en motores ya que sus variaciones de alta frecuencia no afectan a la dinámica del proceso por el efecto filtro del bobinado. Por otro lado, el DAC presenta una aplicación más limpia y rápida. Mientras el PWM necesita que se completen 4096 cuentas (12 bits) de reloj para completar un solo ciclo (23 μ s aprox.) el DAC completa la conversión en alrededor de un microsegundo (según la depuración). Sin embargo, se puede encontrar un defecto en la conversión digital-analógica y es que los valores en los extremos se deforman. En la imagen 30 se muestra cómo el DAC achata la señal en las puntas. Como decisión final, se va a implementar una conversión DAC para aplicar la acción de control pero recortando 360 valores en cada extremo para alejarse de la zona de deformidad; por tanto, en lugar de 4096 posibles valores ahora se tiene una resolución de 3376 divisiones.

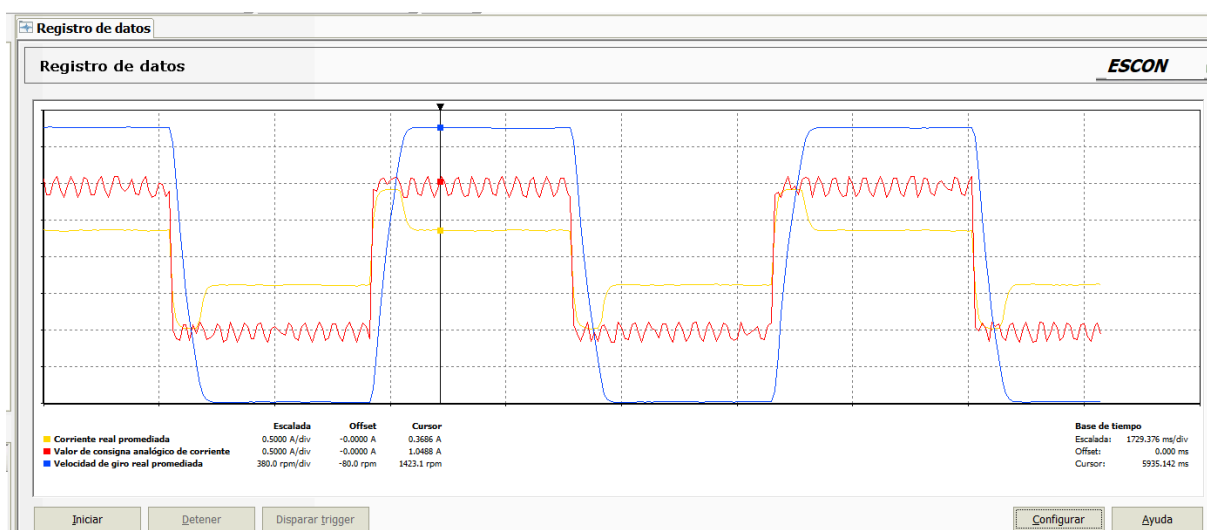


Figura 29: Captura del efecto PWM en ESCON Studio

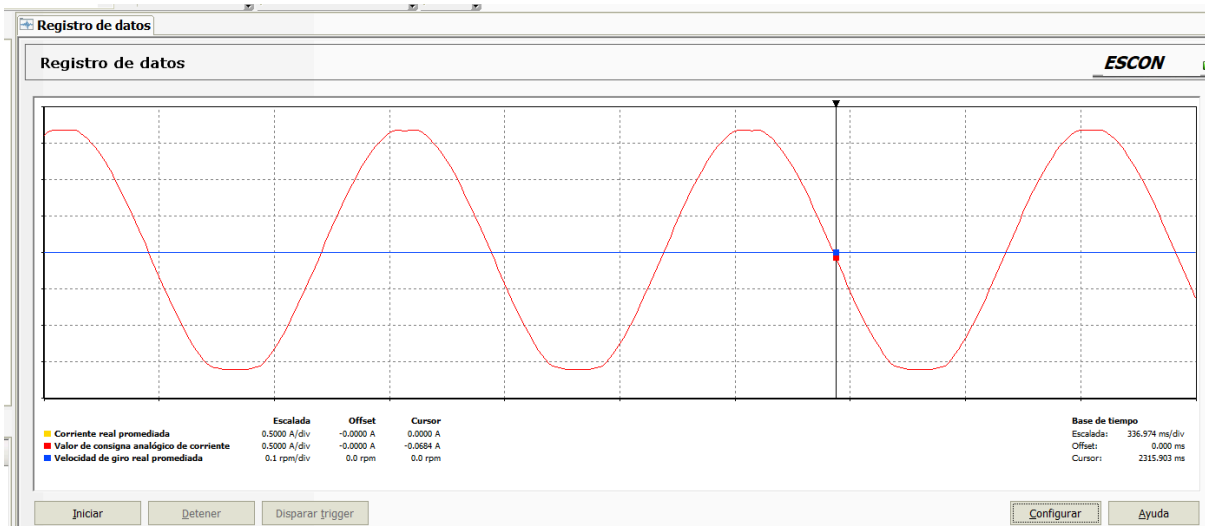


Figura 30: Captura del efecto DAC en ESCON Studio

5.4. Alternativas en el entorno de control

El enfoque que tiene el propio subsistema de control en este proyecto, el cual se basa en interpretar de diferentes maneras los datos recolectados, en sí mismo ya contempla diferentes estrategias alternativas. De todas maneras, antes de llegar a la descripción detallada de este entorno, conviene que en este apartado se incluyan alternativas generales de algunos aspectos de la identificación. Es importante recalcar que el subsistema de control, que alcanza tanto aspectos de la identificación como de modelado, no está plenamente desarrollado y, por limitaciones de tiempo, se le ha dado más peso a el resto de subsistemas para asegurar una buena obtención de datos experimentales. Por tanto, en este proyecto, no se lleva a cabo una identificación profunda del modelo, los parámetros, o el rozamiento del harmonic drive.

En primer lugar, existen varias estrategias generales para la identificación de un proceso. En este caso se trata de si llevar a cabo un enfoque de caja negra (black box) o caja gris (grey box). La primera estrategia se trata de partir de información nula sobre el sistema a identificar. No obstante, en este proyecto se va a abordar el enfoque de grey box ya que se conocen ciertas expresiones teóricas que aproximan el modelo del harmonic drive como articulación elástica y reductor, definiendo también de qué forma afectan los parámetros de los componentes al sistema y cómo interactúan las diferentes variables entre sí. Además, se tiene constancia de ciertos modelos de rozamiento y, junto con lo anterior, proporcionan información muy útil para la base de la identificación.

En segundo lugar, se plantean las diferentes alternativas de filtrado de las señales. En un principio, las posiciones de los encoders no presentan ruido ni perturbaciones indeseadas de alta frecuencia pero, al calcular la velocidad, aparece ruido característico que distorsiona la forma que cabría esperar. En la figura 31 se pueden comparar las posiciones y velocidades de ambos lados del reductor.

El ruido presente en las velocidades se debe al error de cuantización, el cual es un efecto perfectamente esperable tras el tratamiento de una señal discreta. Este aparece al calcular la diferencia entre una posición y la anterior en un periodo de muestreo concreto. Como la posición medida no puede tomar valores infinitamente precisos (la continuidad real) el resultado oscila entre valores concretos. En el caso del secundario, donde se da uso del encoder magnético de alta precisión, se capturan variaciones más pequeñas que con el encoder del motor.

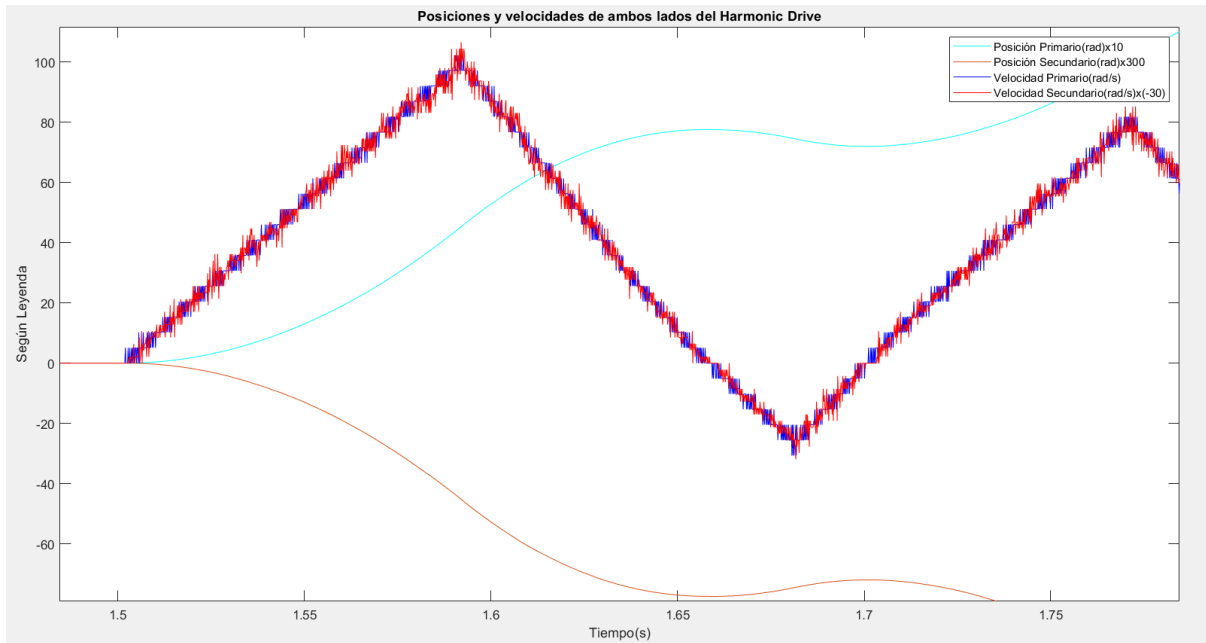


Figura 31: Posiciones y velocidades sin aplicar filtro

Siguiendo por la misma línea de alternativas y para aclarar el efecto del ruido de cuantización se escoge el peor caso: el encoder del motor. En la imagen 32 se puede ver en detalle qué forma tiene el cálculo utilizando tres formas diferentes de filtrado sencillo a través de las fórmulas basadas en la clásica de velocidad en función de la posición:

$$velocidad_i = \frac{posición_i - posición_{i-1}}{T_m} \quad (5)$$

$$velocidad_i = \frac{posición_i - posición_{i-10}}{T_m \cdot 10} \quad (6)$$

$$velocidad_i = \frac{posición_i - posición_{i-100}}{T_m \cdot 100} \quad (7)$$

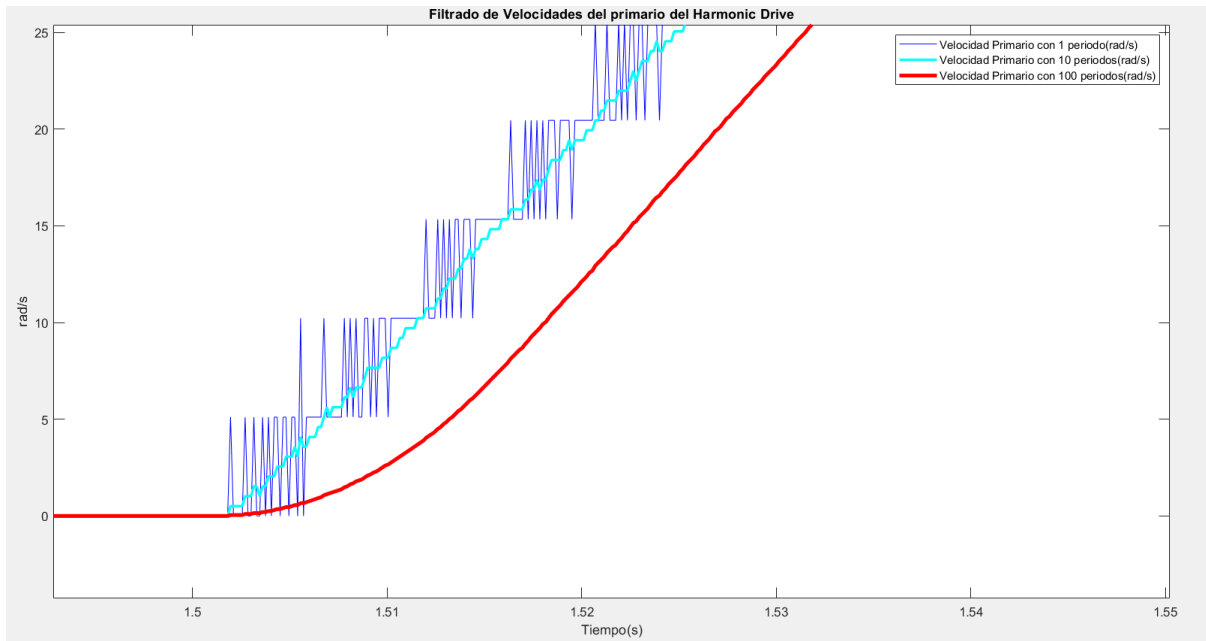


Figura 32: Comparación de velocidades promediadas

En la figura anterior se ve más de cerca el efecto del ruido de cuantización cuando se utiliza un solo periodo de muestreo (se alterna entre dos valores como si de una modulación de ancho de pulso se tratara). El promedio de los valores de este ruido es la velocidad aproximada en el continuo. Si, en lugar de comparar las posiciones con las del periodo anterior, se escogen con más tiempo entre ellas (un periodo más grande) se lleva a cabo este promediado y, con ello, un efecto de filtrado. Nuevamente, en la figura anterior se puede comprobar que a más periodos de muestreo, el error de cuantización se mejora pero se incrementa el retraso en el dato de velocidad.

Conviene en este punto, implementar otros filtros para ver de qué forma se suaviza mejor la señal de velocidad. En la imagen 33 se comparan 4 nuevas formas de estimar la velocidad. Por un lado, se tiene un promediado similar a los anteriores pero seleccionando periodos anteriores y posteriores de manera simétrica. De esta manera se centran mejor los datos pero deforma la señal hacia los laterales. Por otro lado, el filtro FIR (Finite Impulse Response) suaviza de forma efectiva la señal a coste de un retraso inevitable. Si se utiliza un filtro Butterworth se consiguen configuraciones de frecuencia de corte y orden de la etapa que son más efectivas por lo general que los resultados del FIR. Además, también se han hecho pruebas aplicando la función "movmean" de Matlab (para considerar una media móvil de ciertas ventanas de datos) o el primer promediado al filtro Butterworth pero los resultados no muestran mejoras. Finalmente, se prueba aplicar un filtro no causal con el mismo objetivo de encontrar la curva que mejor se ajuste a la recta esperada de velocidad. Este último tipo de filtro, "filtfilt" en Matlab, realiza un filtrado de izquierdas a derechas y de derechas a izquierdas de los datos haciendo que para múltiples aplicaciones sea una opción muy efectiva. El problema viene con la no causalidad del proceso, es decir, datos futuros afectan a la estimación de datos anteriores (se adelanta la señal). Este efecto colateral no es deseable en un proceso de identificación ya que, se busca determinar con precisión causas que rigen el comportamiento de cada variable.

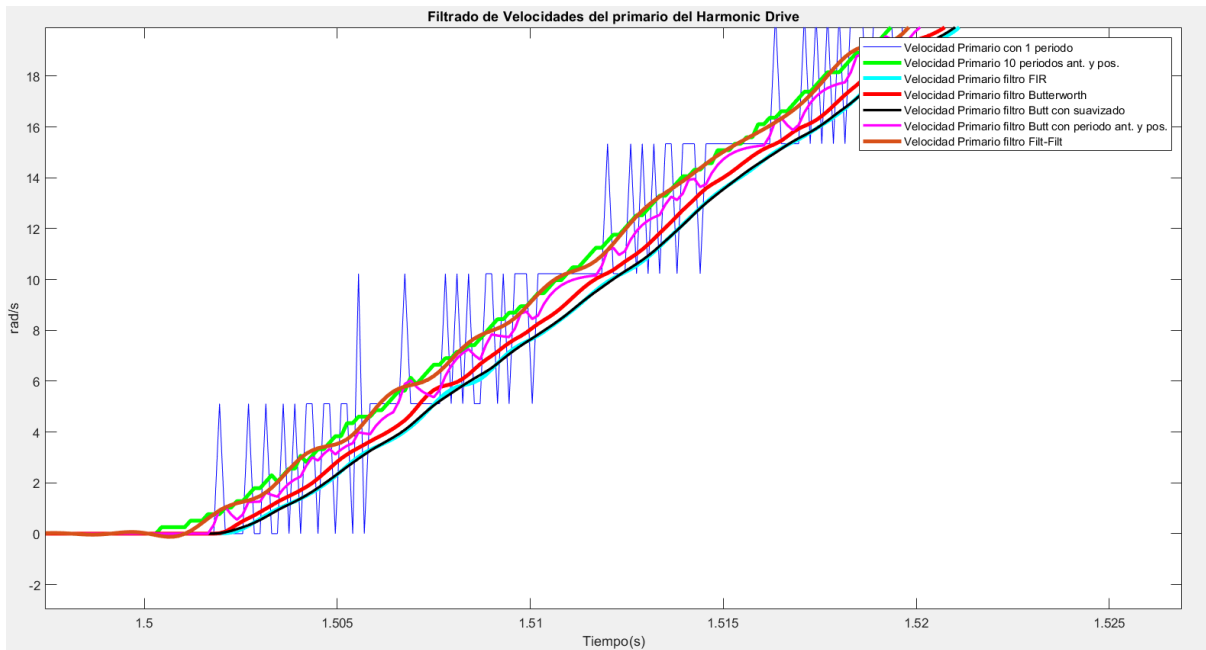


Figura 33: Comparación de velocidades usando diferentes filtros

Además de lo anterior, también se han intentado estrategias de interpolación lineal de los datos de posición. Este método se basa en la creación artificial de nuevos puntos de muestreo sobre una recta imaginaria trazada entre los datos reales obtenidos. De esta manera se crea el efecto no real de un periodo de muestreo menor. Para poder ver mejoras, esta técnica se combinaba con un filtrado Butterworth para suavizar la nueva señal interpolada. En las pruebas, se ha interpolado dividiendo el periodo de muestreo 2000 veces y el resultado levemente mejor pero el coste computacional es excesivamente alto. Esta técnica, por tanto, se descarta para la identificación.

De forma análoga a las pruebas hechas con la velocidad, con la aceleración se obtienen conclusiones parecidas en cuanto al desempeño de cada filtro. Sin embargo, en el caso de la aceleración el error de cuantización es exponencialmente más problemático dada su expresión matemática general:

$$aceleración_i = \frac{posición_i - 2 \cdot posición_{i-1} + posición_{i-2}}{T_m^2} \quad (7)$$

Por tanto, se espera que la aceleración tenga un ruido de cuantización mucho más elevado y es necesario un filtro más agresivo para visualizar la curva evitando este efecto. En la imagen 34 se presentan los datos obtenidos en cuanto a corriente, posición, velocidad y aceleración (filtrados) ante una señal de control de escalones.

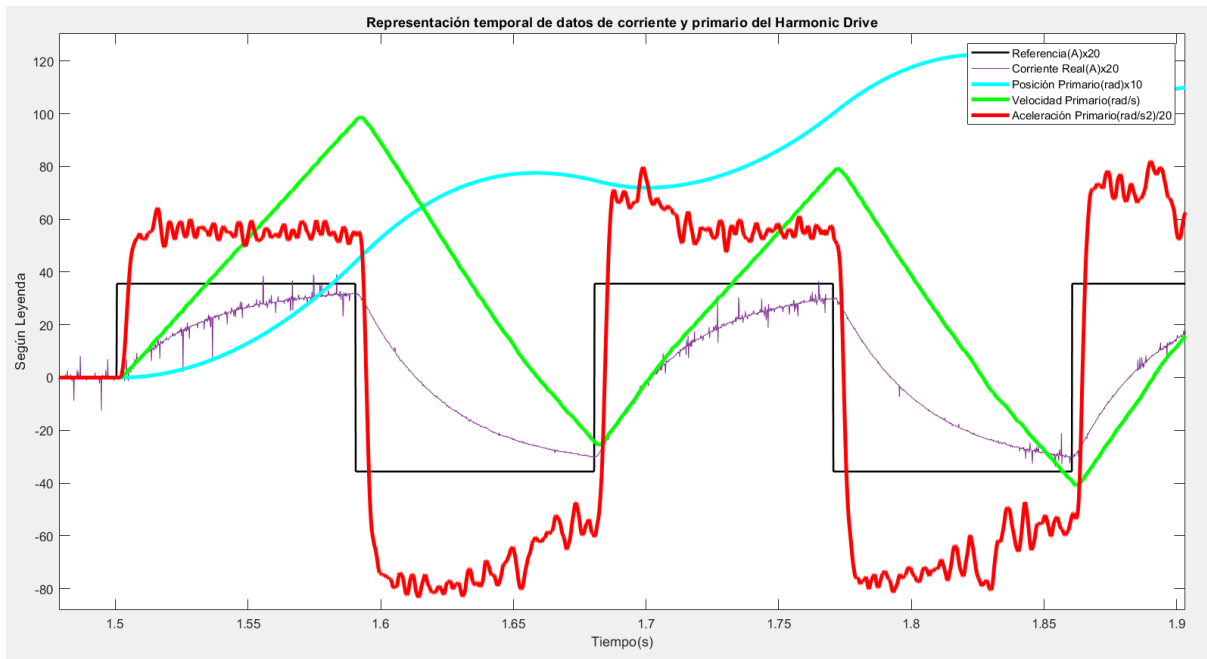


Figura 34: Datos del primario del reductor ante señal cuadrada

Pese a todo el desarrollo descrito de alternativas de limpieza de señales, el proceso de identificación en ocasiones puede filtrar de forma intrínseca los datos (como el método de mínimos cuadrados). Es por eso que, por lo general, se van a utilizar las señales originales sin filtrar ya que de esta manera no se añaden órdenes adicionales al sistema y los propios regresores de los métodos de identificación se aproximan a un promedio prácticamente común usando filtros o prescindiendo de ellos.

6. Descripción y justificación detallada de la solución adoptada

En este apartado se explica de forma precisa cada uno de los componentes y apartados resultantes de las alternativas de los cuatro diferentes subsistemas. En la figura 35 se plantea el organigrama general en el que se estructura este apartado.

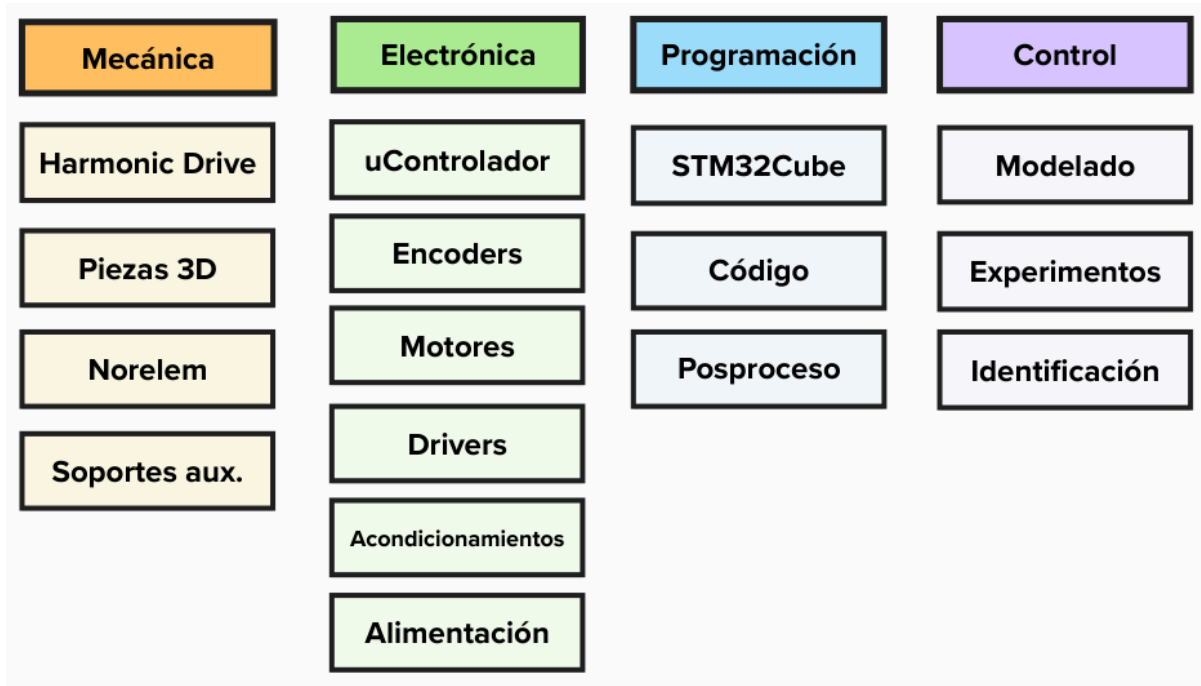


Figura 35: Organigrama general del proyecto

6.1. Subsistema mecánico

En este apartado se describen todas las partes que conforman la estructura mecánica de los diferentes experimentos objetivo.

Los planos, donde se acotan de forma detallada las piezas diseñadas y se representa el conjunto físico entre los diferentes componentes, van en el segundo documento del proyecto tras la memoria. Las especificaciones que detallan cómo poner a punto toda la experimentación y que complementan a los planos se encuentran en el documento de Pliego de condiciones.

6.1.1. Harmonic Drive

El modelo de reductor harmonic drive con el que se va a trabajar es el HFUC-14-2UH de ratio 30:1 y lubricado con grasa [13]. En la imagen 36 se puede ver este componente junto con su modelado en CAD en formato STEP, disponible en la web del fabricante [2].

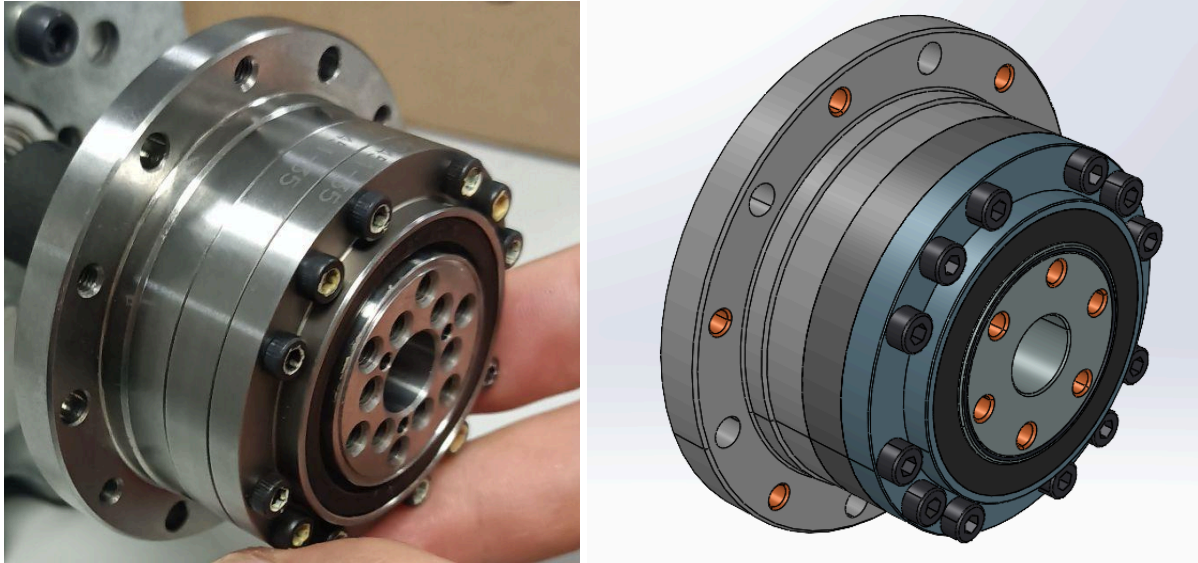
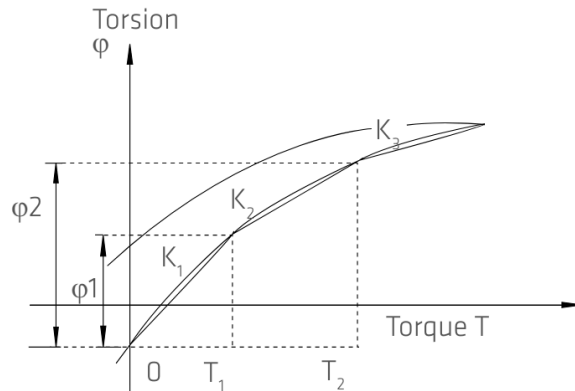


Figura 36: Harmonic drive HFUC-14-2UH real y en CAD

Las características más relevantes de este engranaje son:

- Reducción: 30:1 con sentido invertido
- Velocidad de entrada media: 3500 rpm
- Velocidad de entrada máxima: 8500 rpm
- Par de inicio mínimo en la entrada: 0.06 Nm
- Par reversible mínimo sin carga (back drivability): 2.4 Nm
- Par de trabajo medio: 6.8 Nm
- Par momentáneo máximo: 17 Nm
- Momento de inercia: $3.3 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$

En la hoja de características del harmonic drive no figuran todos los parámetros que conciernen a una articulación elástica como el coeficiente de fricción viscosa (b_e) o el coeficiente de amortiguamiento (B). Por otra parte, hay parámetros que no tienen por qué ser constantes en todo momento del desempeño del mecanismo y el fabricante afirma que la rigidez torsional (K) es uno de estos casos. En la gráfica presentada en la figura 37 se puede ver cómo este parámetro depende del par aplicado en la entrada. Como en las experimentaciones se trabaja con niveles de par bajos y siguiendo las indicaciones de [13] se selecciona K_1 , es decir 3400 Nm/rad.



K_1, K_2, K_3 = Torsional stiffness, w = Output angle
 ϕ_1 = Torsion angle, with output torque T_1
 ϕ_2 = Torsion angle, with output torque T_2

Figura 37: Rigidez torsional dependiente del par [13]

En la figura 38 se puede ver el flexspline encajado en el circular spline listo para introducir el eje del motor con el wave generator.

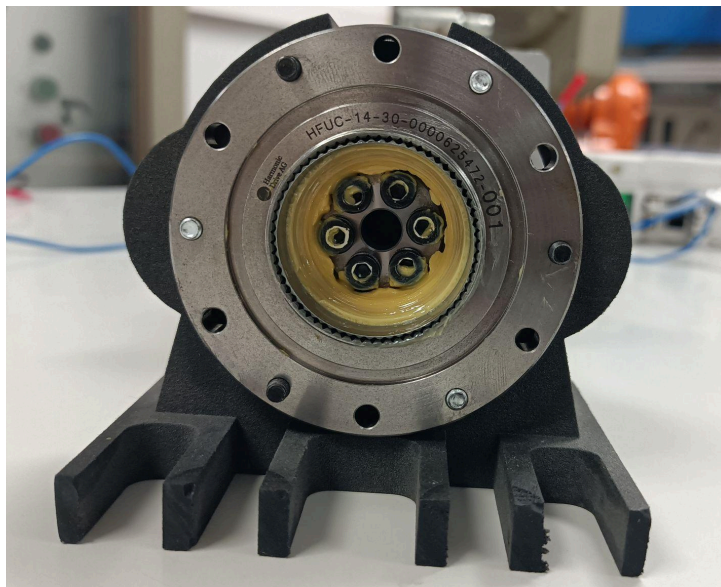


Figura 38: Flexspline encajado en circular spline para E1, E2 y E3

Antes de seguir con el resto de componentes, cabe comentar que en un punto de los ensayos se lleva a cabo un despiece del harmonic drive como se ve en la figuras 39 y 40 (una vez realizados todos los experimentos que requieren el reductor completo). El objetivo es disponer de una versión más simple de este (wave generator y flexspline) y realizar pruebas para el E4 (figura 41).

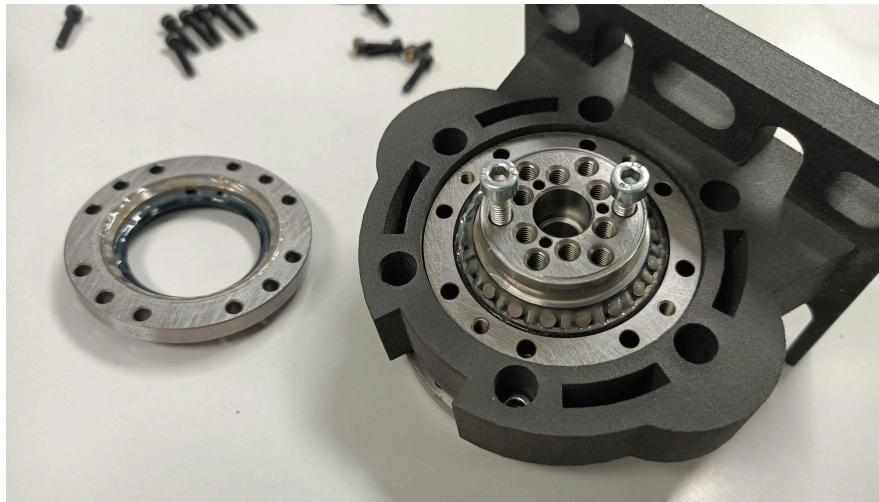


Figura 39: Proceso de despiece del Harmonic drive

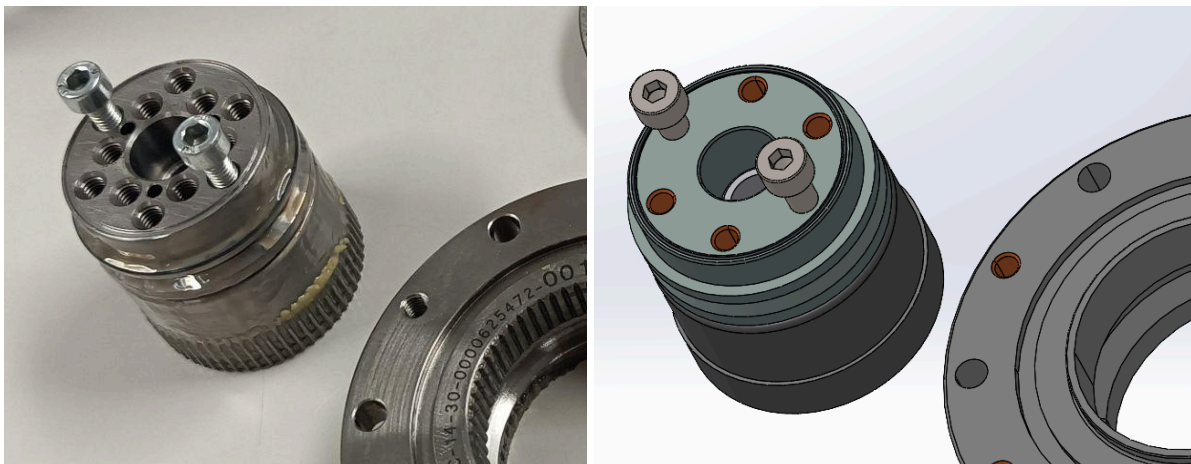


Figura 40: Flexspline extraído real y en CAD

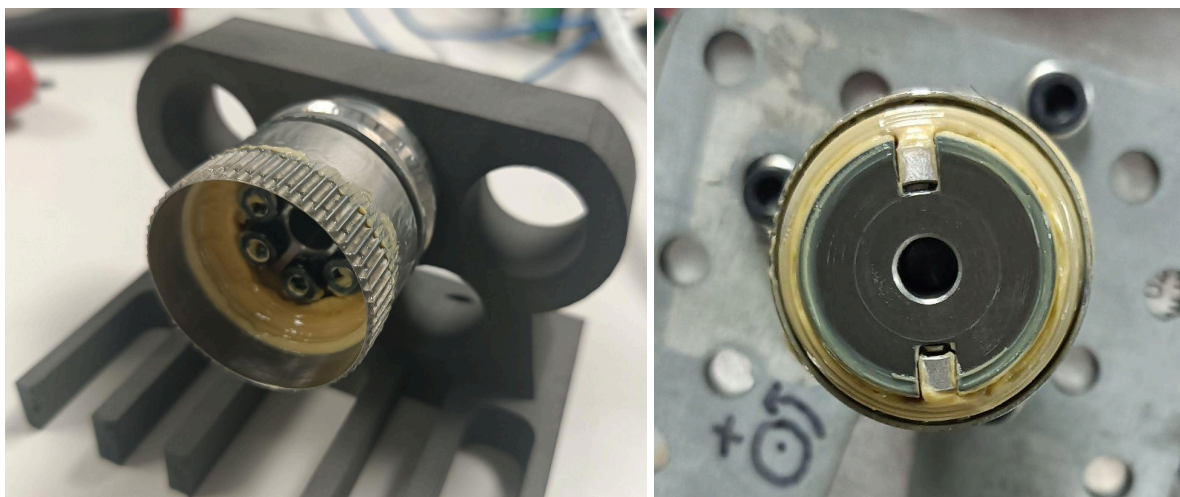


Figura 41: Flexspline (izq.) y wave generator (der.) en E4

6.1.2. Piezas impresas

Con el objetivo de fijar y unir los diferentes componentes de los experimentos, se han diseñado una serie de piezas en el *software* Solidworks. Posteriormente, se han impreso usando el método de sinterizado por láser selectivo (SLS), servicio disponible en la UPV. Además, el material utilizado ha sido nylon que, junto a esta técnica de impresión, asegura unos resultados muy resistentes y rígidos [17], propiedades necesarias para el buen desempeño de los experimentos.

Cabe comentar que los nombres de las diferentes piezas son la letra “E” y un número seguido que hace referencia a qué experimentos dan uso de estas. También se pueden encontrar inscripciones en algunas piezas que aclaran el nombre de la pieza y dónde se encajan, por ejemplo: primario del Harmonic Drive (HD1), servomotor (SM) o motor *brushless* (BL).

Los planos de piezas y conjuntos, donde se visualizan y acotan de forma detallada los elementos diseñados y su disposición en las diferentes pruebas, van en el segundo documento del proyecto tras la memoria.

6.1.2.1. Bases

Las bases tienen la función de encajar el reductor y que permanezca completamente fijo y estable sobre la mesa de trabajo. Existen dos modelos: para los primeros experimentos se tiene el E123 para encajar el circular spline y para la última prueba el E4 que sólo fijará el flex spline (ver figura 42).

Cabe comentar que el apoyo de estas piezas tiene una anchura progresivamente más gruesa. Este efecto tiene el objetivo de encajar de forma firme entre las piezas del Norelem. Finalmente se disponen de 6 agujeros para introducir pernos de métrica M4 para la parte del harmonic drive y unas ranuras en la base para introducir pernos M10 para la parte del Norelem y que permita ajustar la posición a lo largo de la dirección del eje de giro si fuera necesario. No obstante, tras la impresión 3D, fue conveniente cortar aproximadamente 1 cm de material sobrante de la parte delantera para dar más espacio al servomotor y que pudiera apoyarse en la propia mesa de Norelem.

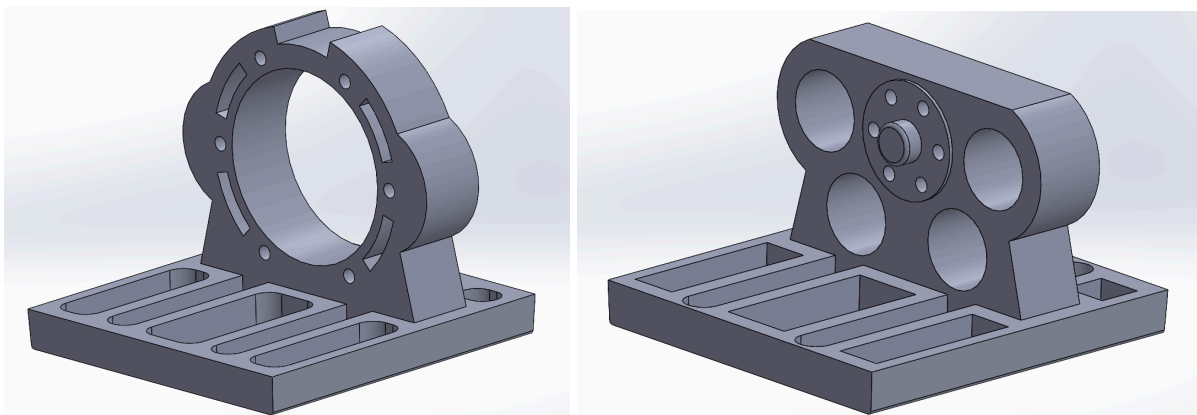


Figura 42: Base E123 (izq.) y Base E4 (der.)

6.1.2.2. Ejes

Los ejes tienen como función acoplar los motores al harmonic drive (ver figura 43). Por un lado, el modelo E134 une el BLDC con el primario del reductor. Este dispone de dos pasadores para tornillos M3 para la parte del wave generator y, para la parte del motor, un orificio que se va estrechando para que su eje quede lo más solidario posible y que no deslice.

Por otro lado, el modelo E23 sigue la misma línea del diseño anterior pero esta vez para encajar con el servomotor de potencia con el secundario del reductor. Para ello, hay seis agujeros para pasar pernos M4 y alrededor de estos, una base de revolución para encajar el disco del encoder magnético. Por último, para poder roscar los pernos sin dificultades se ha vaciado material del eje en una dirección abierta desde cada uno de los respectivos agujeros.

Por último, cabe comentar que la primera versión de estas piezas no resultó satisfactoria debido a que los ejes de los motores se deslizaban o quedaban holgados. Este inconveniente inesperado fue ocasionado por la dilatación del nylon que, tras la impresión de piezas presenta un ligero error que contrae las piezas y ensancha los agujeros. Es por ello que se repitió el diseño de las piezas reduciendo el tamaño de los orificios y, tras una segunda impresión 3D, las piezas estaban listas para encajarse. Eso sí, esta vez había sido necesario un lijado del material interior para que los ejes encajaran hasta una profundidad suficiente.

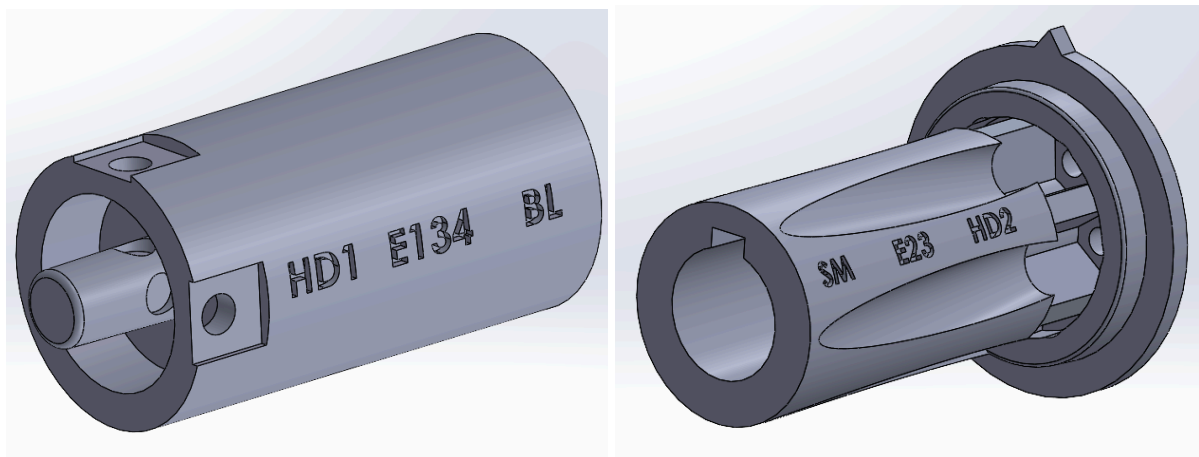


Figura 43: Eje E134 (izq.) y Eje E23 (der.)

6.1.2.3. Soportes de encoders

Estos elementos tienen el objetivo de unir cada lado del motor con los discos de los encoders correspondientes cuando estos no están conectados a ningún motor. Las formas en el diseño son análogas a las anteriores pero eliminando el acoplamiento con los motores para reducir perturbaciones. Cabe resaltar que en el E2 (ver figura 44) se sitúa un encoder diferente que en el E1, es por ello que los diámetros no son los mismos.

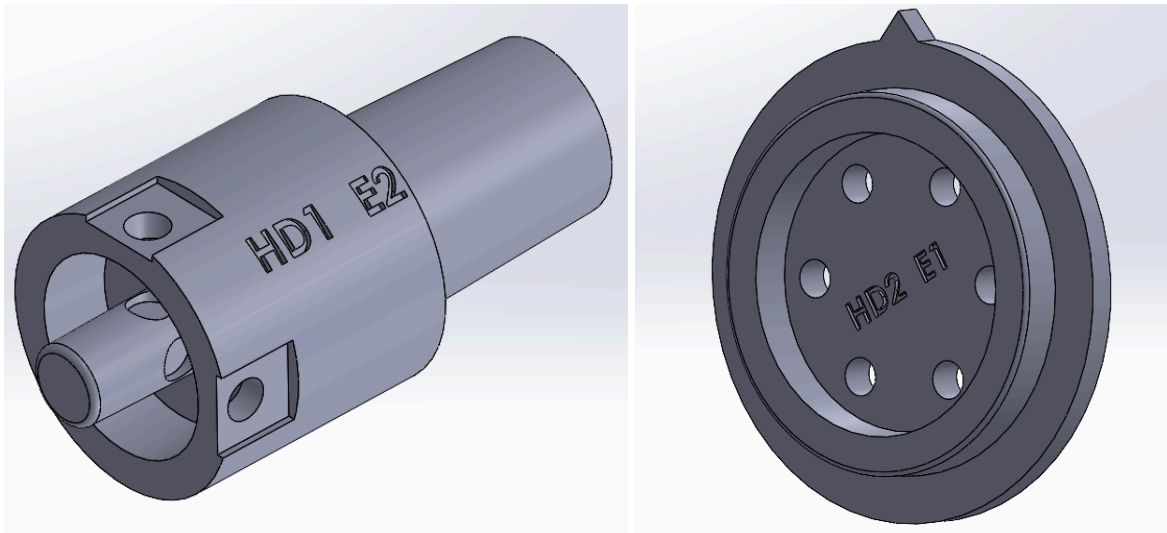


Figura 44: Soporte encoder E1 (izq.) y Soporte encoder E2 (der.)

6.1.3. Norelem

En este punto se escogen las piezas de Norelem del laboratorio que mejor sujeten a los diferentes elementos. La variedad disponible es limitada pero con los elementos que se pueden ver en la imagen 45 se construye una buena base de experimentación. Las referencias exactas de cada componente y forma de ensamblaje se encuentran en el Pliego de condiciones.

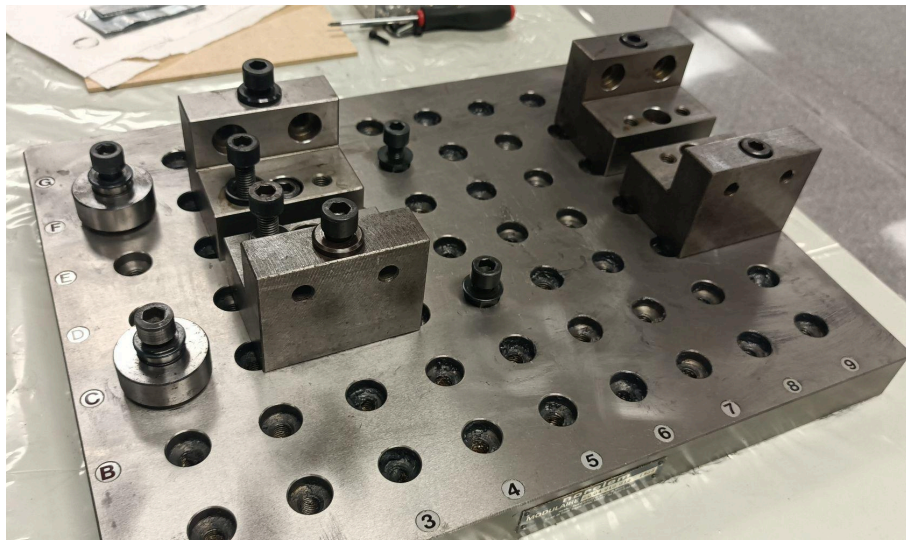


Figura 45: Base y piezas de Norelem del laboratorio para los ensayos

6.1.4. Elementos auxiliares

A parte de los componentes mecánicos anteriores, para completar el presente subsistema es necesario mencionar la presencia de algunos elementos auxiliares:

- Una escuadra reforzada de acero para sujetar el motor BLDC.
- Una lámina de madera para elevar la escuadra a la altura correcta.
- Unas láminas de papel de aluminio para ajustar los motores a la altura correcta.

- Unas estructuras de Lego y Meccano para la sujeción de los lectores de posición (figura 46).
- Una tapa de cartón para albergar parte de los componentes electrónicos.

Para finalizar, cabe aclarar que las propiedades y condiciones detalladas de cada uno de los elementos de este subsistema se manifiestan en el tercer documento de este proyecto: el Pliego de condiciones.

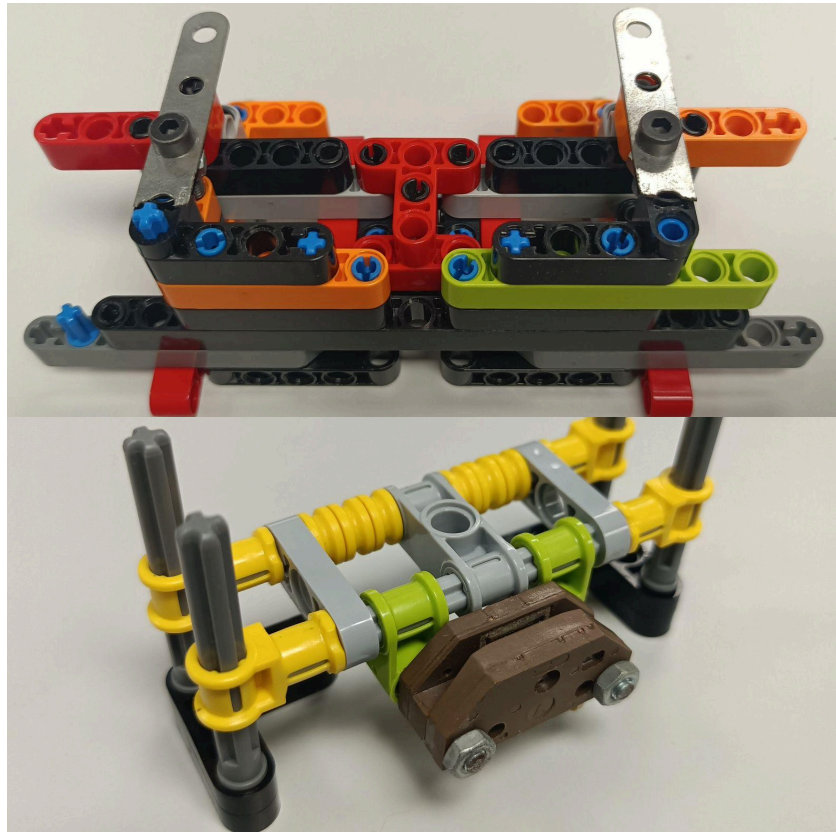


Figura 46: Estructuras de Lego para sujetar los lectores

6.2. Subsistema electrónico

En este apartado se van a desarrollar cada una de las partes resultantes de las alternativas de la parte electrónica.

6.2.1. Módulo de adquisición de datos

La pieza central de la electrónica es el microcontrolador STM32F429 Discovery kit (figura 47) [31] el cual se va a utilizar para aplicar las diferentes referencias de corriente y adquirir los datos de los experimentos. Esta tarjeta dispone de una pantalla implementada y muchas de sus funcionalidades están configuradas por defecto para su uso. Por ello, lo primero que conviene hacer para disponer de todos los periféricos y canales es limpiar toda la configuración de pines.

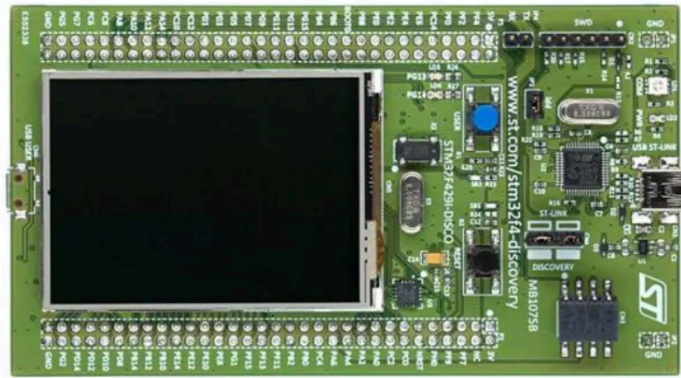


Figura 47: Microcontrolador STM32F429 Discovery kit

Esta placa trabaja en un entorno de programación C y tiene altas funcionalidades de depuración las cuales han sido necesarias en múltiples ocasiones para identificar errores en el código, vigilar variables en periodo de ejecución y medir tiempos. Las características más relevantes de este componente [32] que conciernen al proyecto son:

- Alimentación: 5 V
- Lógica: 3 V
- Frecuencia de reloj interno: 180 MHz
- Puerto de depuración con ST-LINK/V2-B incorporado (debug)
- Interrupciones
- Múltiples temporizadores (timers)
- Convertidor analógico digital (ADC)
- Convertidor digital analógico (DAC)
- Transmisión serie (USART)

6.2.2. Encoder magnético

El encoder magnético de alta precisión, usado para medir la posición del secundario, está formado por dos piezas fundamentales: un lector (*readhead*) y un disco magnético. Por un lado, el redhead es uno de los modelos “RLC2HD” y es la parte fija del encoder que se encarga de captar los pulsos del disco incremental. En la imagen 48 se puede ver como es necesario según [33] soldar el sensor en una placa de circuito impreso PCB para tener acceso a los pines. En la imagen 49 se puede ver el resultado de esta soldadura en una placa de prototipos.

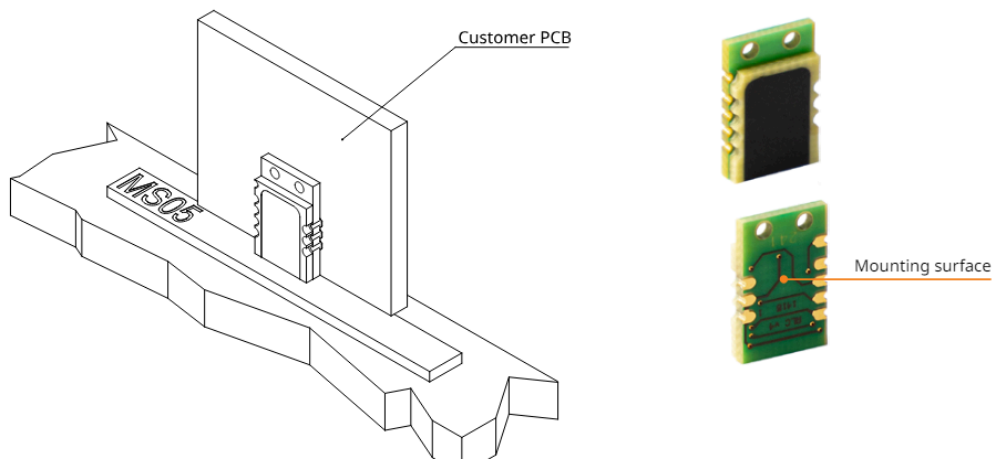


Figura 48: Implementación de lector de encoder magnético según [33]

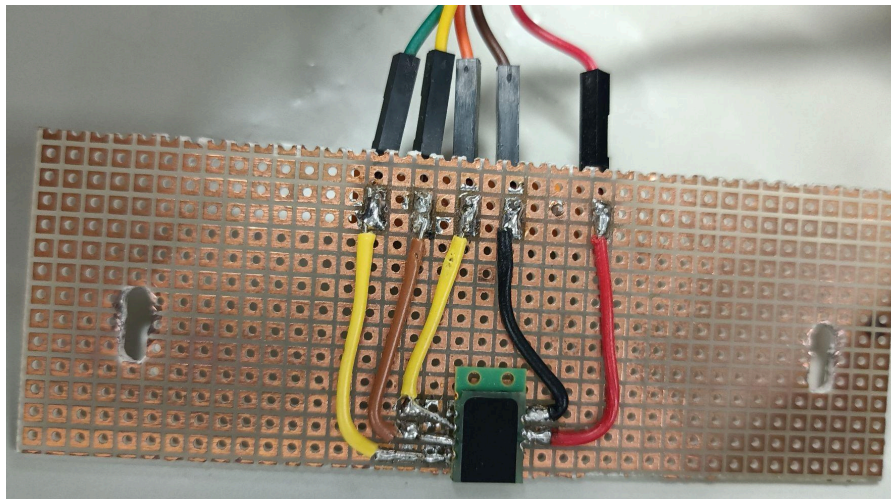


Figura 49: Lector de encoder magnético soldado en PCB

Por otro lado, el disco magnético tiene la función móvil del encoder. En concreto, se trata de un modelo “MR” llamado anillo axial magnético incremental [34] con 72 polos y 11 bits de resolución de pulsos por polo por cada canal (véase figura 50). Por tanto, al interpolar ambos canales por cuadratura se obtienen 8192 cuentas por polo, es decir, 589824 cuentas por revolución.

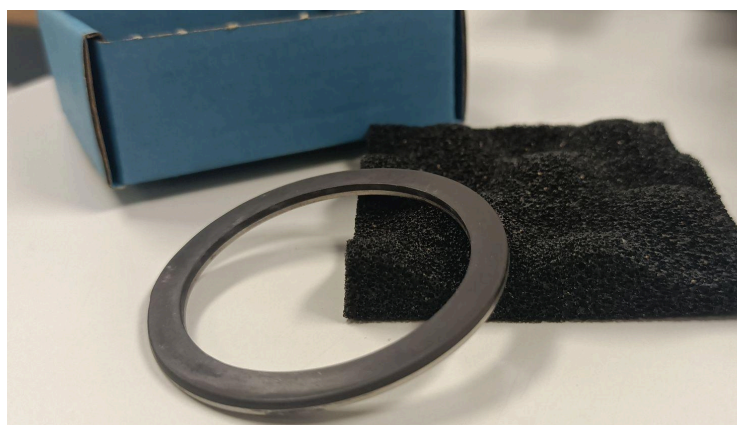


Figura 50: Anillo axial magnético incremental

Ordering code	Interpolation	Resolution		Maximum rpm								
		cpr	ppr	NOTE: Values in brackets are valid for LM13 standard version only.								
13B	8192	589824	147456	760	380	95	47	25	13	10	5	3
12B	4096	294912	73728	1520	760	190	95	50	25	20	10	5
11B	2048	147456	36864	3040	1520	380	190	99	51	40	20	10
2D0 (001)	2000	144000	36000	3113	1553	389	194	101	52	41	21	10
1D6	1600	115200	28800	3887	1947	486	243	127	65	51	26	13
10B	1024	73728	18432	6073	3040	760	380	198	101	80	40	20
1D0 (002)	1000	72000	18000	6220	3113	780	389	203	104	82	41	21
D80	800	57600	14400	7780	3887	973	486	254	130	102	52	26
09B	512	36864	9216	12153 (9300)	6073	1520	760	396	203	160	81	41
D50	500	36000	9000	12447 (9300)	6220	1553	780	406	207	164	83	41
D40 (005)	400	28800	7200	13000 (9300)	7780	1947	973	507	259	205	103	52
D32	320	23040	5760	13000 (9300)	9720 (9300)	2433	1213	634	324	256	129	65
08B	256	18432	4608	13000 (9300)	12153 (9300)	3040	1520	793	405	320	161	81
D20 (010)	200	14400	3600	13000 (9300)	13000 (9300)	3887	1947	1013	519	409	206	104
D16	160	11520	2880	13000 (9300)	9720 (9300)	2433	1213	634	324	256	129	65
07B	128	9216	2304	13000 (9300)	13000 (9300)	6073	3040	1587	810	640	323	162
D10 (020)	100	7200	1800	13000 (9300)	13000 (9300)	3887	1947	1013	519	409	206	104
D08	80	5760	1440	13000 (9300)	9720 (9300)	2433	1213	634	324	256	129	65
06B	64	4608	1152	13000 (9300)	13000 (9300)	12153 (9300)	6073	3173	1620	1279	645	324
D04 (050)	40	2880	720	13000 (9300)	9720 (9300)	2433	1213	634	324	256	129	65
05B	32	2304	576	13000 (9300)	13000 (9300)	13000 (9300)	12153 (9300)	6340	3241	2558	1290	649
04B	16	1152	288	n/a	13000 (9300)	13000 (9300)	13000 (9300)	12680 (9300)	6482	5117	2580	1297
03B	8	576	144	n/a	n/a	13000 (9300)	13000 (9300)	13000 (9300)	12964 (9300)	10233 (9300)	5161	2594
Minimum edge separation (µs)				0.07	0.13	0.50	1	2	4	5	10	20
Maximum count frequency (MHz)				15	8	2	1	0.5	0.25	0.2	0.1	0.05
Part numbering				K	A	B	C	D	E	F	G	H

Tabla 1: Velocidades máximas según modelo de encoder [33]

Esta configuración permite una velocidad máxima de 760 rpm en el secundario (tabla 1), valor que nunca se alcanzará en ningún experimento lo cual ofrece un buen margen de seguridad. Por ejemplo, situándose en el experimento del motor en el primario, teniendo en cuenta la reducción del harmonic drive que es 30, el motor en el primario podría girar hasta 22800 rpm (velocidad en vacío del motor: 1560 rpm).

Por otro lado, si en el experimento el mecanismo es accionado desde la salida (utilizando el servomotor) la velocidad máxima en el secundario lo imponen otros factores. La limitación que hay que considerar es que a este lado del mecanismo existen muchas más pérdidas y puede que el harmonic drive no esté preparado para estas velocidades o fuerzas aplicadas para alcanzar esas mismas velocidades. En su hoja de características solo se comentan valores de par mínimo para iniciar el movimiento, por tanto, se procede a trabajar cerca de este mínimo.

El fabricante (figura 51) incluye la forma de calcular la frecuencia máxima a la que el encoder emitirá las señales a medir. Como caso crítico, se va a tomar la velocidad máxima de 1560 rpm entre 30 (52 rpm) en el secundario y esto da como resultado una frecuencia de 511 KHz (periodo de 1,956 microsegundos). Esta frecuencia no tiene que ver con la de muestreo del código temporizado, la cual es 6.66 KHz (150 µs de periodo); si no con la frecuencia a la que los timers de la Discovery pueden trabajar. En concreto, se utiliza el timer 2 que viene manejado por el bus de periféricos avanzado (APB) número 1 [31] y, como se ve en la figura 52, está asignado a 90 MHz.

Output Frequency calculator

Let's assume the encoder speed (RPM) is 1000 and the counts per revolution (CPR) is 2000.

The output frequency of an encoder can be calculated using the formula:

$$\text{Output Frequency (Hz)} = (\text{Encoder Speed (RPM)} * \text{Counts Per Revolution}) / 60$$

Let's plug in the values:

$$\text{Output Frequency (Hz)} = (1000 \text{ RPM} * 2000 \text{ CPR}) / 60 = 33333,33 \text{ Hz}$$

Therefore, the output frequency of the encoder in this example would be approximately 33,33 KHz.

Remember that this calculation assumes that every count in the CPR is being read (4x encoding). Depending on the encoding mode, this number could be different.

Figura 51: Frecuencia de señales de salida del encoder [33]

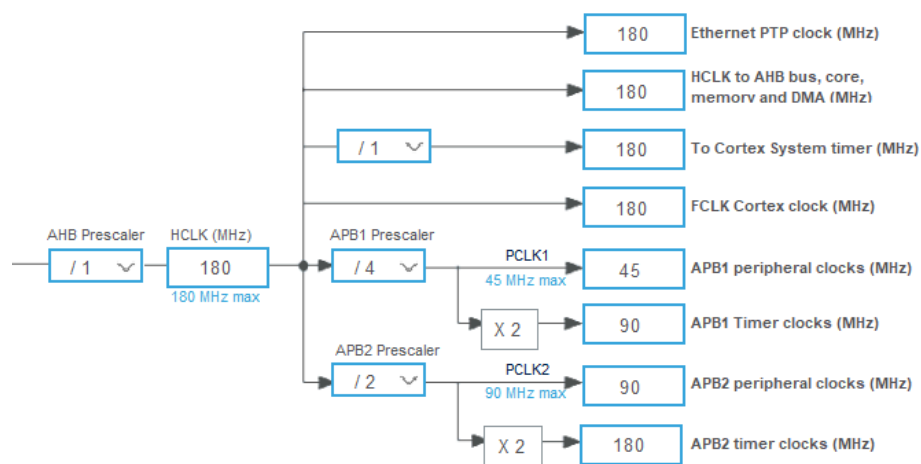


Figura 52: Frecuencias asignadas a periféricos de la Discovery

Por último, cabe comentar que, como el encoder trabaja a 5 V, es necesario un circuito de acondicionamiento el cual se expone más adelante.

6.2.3. Encoder óptico

Para el experimento E2, el objetivo es mover el secundario del mecanismo y que el primario esté en vacío. Al no tener el motor BLDC conectado y, por tanto, no disponer del encoder que lleva incorporado, se tiene a mano un encoder adicional para este experimento. Se trata de un encoder óptico incremental de 12 bits de resolución por vuelta (figura 53).

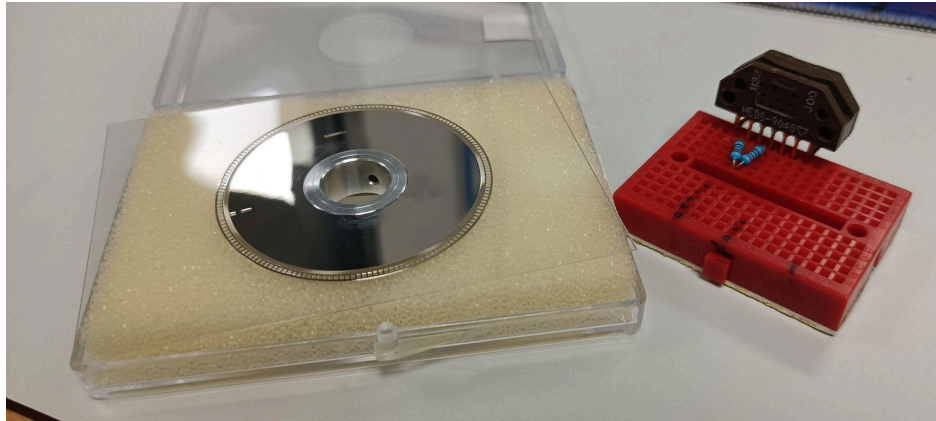


Figura 53: Lector y cabezal del encoder óptico

El cabezal de sensor puede funcionar en un rango de tensiones variado, incluido la lógica de 3 V, por tanto, es directamente compatible con la Discovery. El único detalle que tener en cuenta es la aplicación de dos resistencias de pull-up [35] en las salidas de los canales de datos.

Cabe comentar que, en el planteamiento final de los experimentos, este encoder finalmente no es implementado ya que se usa directamente el incorporado por el motor BLDC.

6.2.4. Motor BLDC

El accionamiento del primario en los experimentos es un motor de corriente continua sin escobillas o *brushless* (BLDC) de la marca Maxon (figura 54) [36]. Tiene un encoder incorporado con 8192 cuentas por revolución. Las características más relevantes son:

- Tensión nominal: 36 V
- Velocidad nominal: 1240 rpm
- Velocidad en vacío: 1560 rpm
- Corriente nominal: 1,78 A
- Constante de par: 0,217 Nm/A
- Inercia del rotor: 3060 gcm^2
- Coeficiente de fricción viscosa: 0.02044 Nm/(rad/s)
- Tensión de encoder: 5 V
- Resolución de encoder: 2048 pulsos por revolución (8192 cuentas por cuadratura)

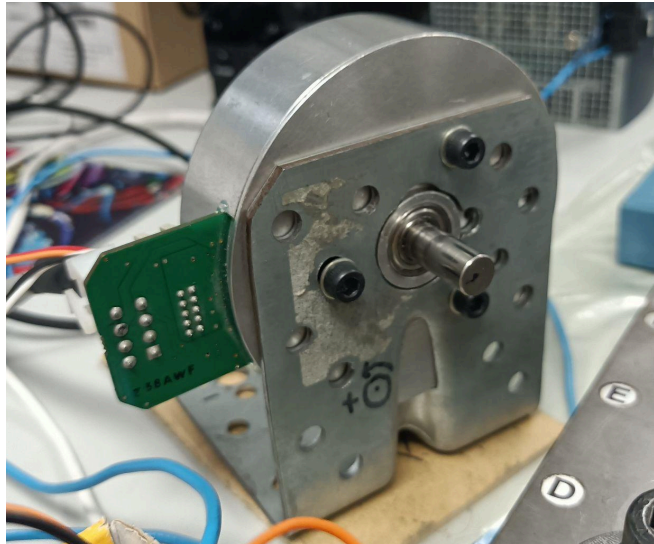


Figura 54: Motor BLDC en escuadra

6.2.5. Servomotor

Para accionar el Harmonic Drive desde la salida se utiliza un servomotor de potencia (figura 55) de la marca Nidec (Control Techniques) [37]. Este dispone de un encoder de 4096 pulsos por revolución pero no es relevante ya que se utiliza el encoder magnético para el secundario. Las características más relevantes aquí son:

- Velocidad nominal: 3000 rpm
- Par nominal: 5.6 Nm
- Constante de par: 1.6 Nm/A
- Inercia del rotor: 3700 gcm^2

En la hoja de características los datos se vinculan con el número de serie del motor, como se puede ver en la tabla 2.

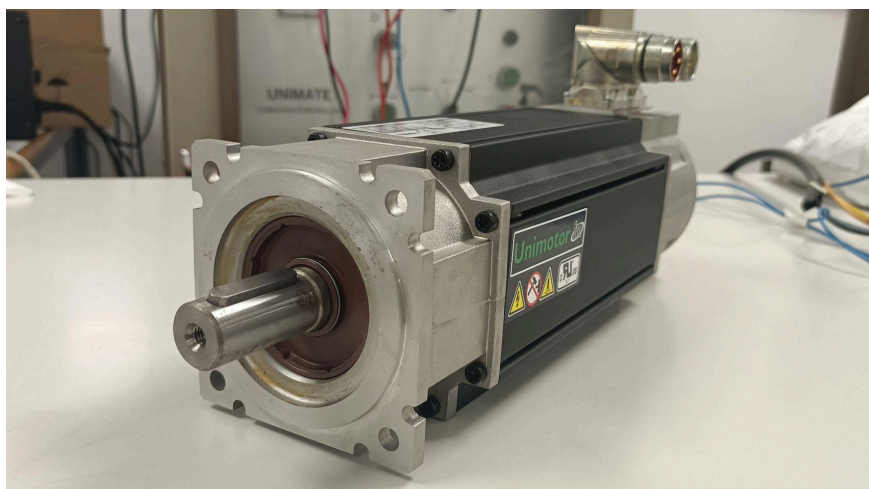


Figura 55: Servomotor de potencia

095	U3	A	30	5	B	A	CA	A	100	190
Frame size	Motor voltage	Stator length	Rated speed (rpm)	Brake	Connection type	Output shaft	Feedback device	Inertia + Temp sensor	BCD	Shaft Diameter (ex. 110=11.0 mm)
075	E3 = 230 V	A	20 = 2000	0 = No Brake	B = Power and Signal 90° and rotatable size 1.0	A = Keyway w/ Full Key installed	EC = Inductive Absolute Multi-turn EnDat	A = Standard + PTC	075 Std	075 Frame
095	U3 = 460 V	B	30 = 3000	5 = 24 Vdc Parking brake	C = Power 90° rotatable and signal vertical size 1.0	B = No Keyway	FC = Inductive Absolute Single-turn EnDat	B = High + PTC	080 Opt	110 23 A
115	250 Frame	C	40 = 4000		V = Power and Signal Vertical size 1.0	E* = Keyway w/ Half Key installed	VF = Capacitive Hiperface SinCos Multi-turn	C = Standard + KTY84-130 thermistor**	085 Opt	140 30 B-D
142		D	60 = 6000*		D = Hybrid power and signal 90° size 1.0**	F* = Keyway w/ Half and Full Key included	WF = Capacitive Hiperface SinCos Single-turn	D = High + KTY84-130 thermistor**	095 Frame	190 40
190	U3 = 460 V	095 -142 Frames	250 Frame		J = 1.5 size Power 90° and signal 90° and rotatable		075-250 Frames		100 Std	095 Frame
250		A	10 = 1000		E = Hybrid power and signal 90° size 1.5**	N = Power 90° rotatable and signal vertical size 1.0	AE = Resolver	CA = Incremental Encoder (Std) 4096 ppr	098 Opt	140 30 A
		B	15 = 1500		M = Power and Signal Vertical size 1.0	EB = Optical Absolute Multi-turn EnDat	ECN 1325		115 Opt	190 40 B-E
		C*	20 = 2000 ^o			FB = Optical Absolute Single-turn EnDat	ECN 1313		115 Std	115 Frame
		D*	25 = 2500 ^o			RA = Optical SinCos Multi-turn Hiperface	SRM 50		130 Opt	142 Frame
		E*				SA = Optical SinCos Single-turn Hiperface	SRS 50		165 Std	240 50 D, E
		190 Frame				EF = Inductive Serial Multi-turn EnDat	EQI 1331**		149 Opt	142 Frame
		A				FF = Inductive Serial Single-turn EnDat	ECI 1319**		190 Frame	240 50 A-E
		B				GB = Optical Multi-turn EnDat	EQN 1337**		215 Std	320 58 D, E
		C*				HB = Optical Single-turn EnDat	ECN 1325**		250 Frame	190 Frame
		D*							300 Std	320* 80* A-H*
		E*								320 58 A-H
		F*								380 58
		G*								250 Frame
		H*								480 110 D-F

Tabla 2: Datos del servomotor según su número de serie [37]

6.2.6. Controlador BLDC

Como etapa de potencia para que la Discovery pueda manejar el motor BLDC se utiliza el driver ESCON 50/5 de la misma marca (figura 56) [25]. Las propiedades más importantes a destacar son:

- Tensión de trabajo máxima: 56 V
- Corriente de salida continua máxima: 5 A
- Corriente de salida máxima (<20 s): 15 A
- Dos entradas analógica diferencial: 12 bits (de -10 V a +10 V)
- Dos salidas analógicas: 12 bits (de -4 V a +4 V)
- Medición de corriente del motor



Figura 56: Controlador ESCON 50/5 [25]

Además, dispone de un puerto USB para conectarse y con el programa de ESCON Studio poder configurar los pines de entrada y salida así como las características del motor accionado. Además hay herramientas de auto tuning para una rápida calibración del control del motor y una ventana de visualización de datos como la velocidad, la corriente real o de referencia, etc.

Para poner a punto la controladora, primero se debe establecer la conexión. Luego se selecciona motor de tipo EC y se escriben sus parámetros (imagen 57). En tercer lugar, se escoge la polaridad, que solo se refiere al sentido de giro, y se define la resolución del encoder interno. En este caso, se selecciona la entrada digital 2 como la habilitación y paro del motor.

En las siguientes ventanas se selecciona el control de modo corriente (imagen 58) que es equivalente a regular el par del motor. Luego se selecciona la entrada analógica 2 para poder introducir por ella la referencia de corriente y se ajusta la escala para que las señales tengan el máximo en la corriente nominal del motor (hay que recordar que la señal de referencia no alcanza los 10 V por el defecto del DAC). Luego, es conveniente aplicar un offset de la referencia para centrar en cero la señal (0.015 en este caso).

Finalmente, en la imagen 59 se puede ver cuál es la configuración de la salida analógica 1 para la medida de la corriente y su escala.

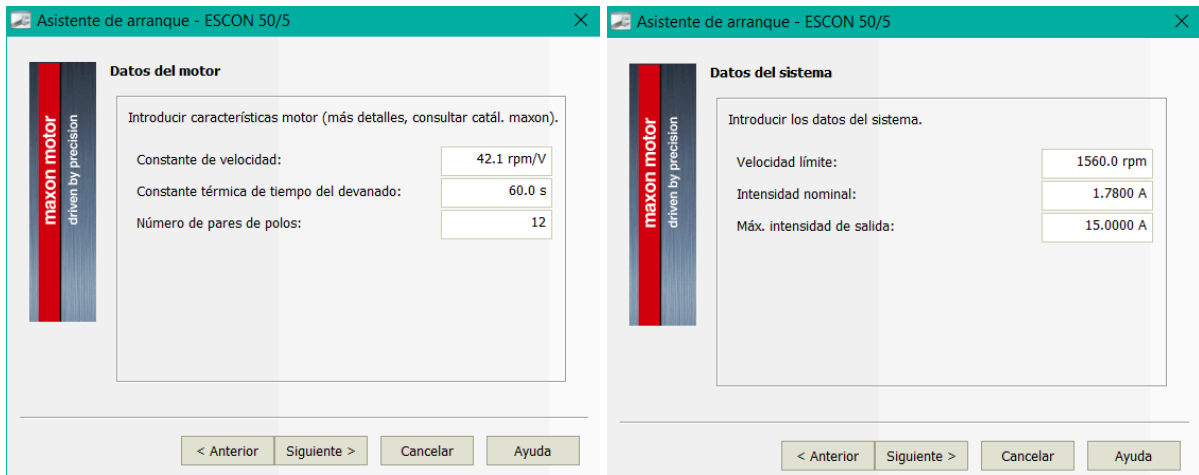


Figura 57: Configuración de parámetros del motor BLDC

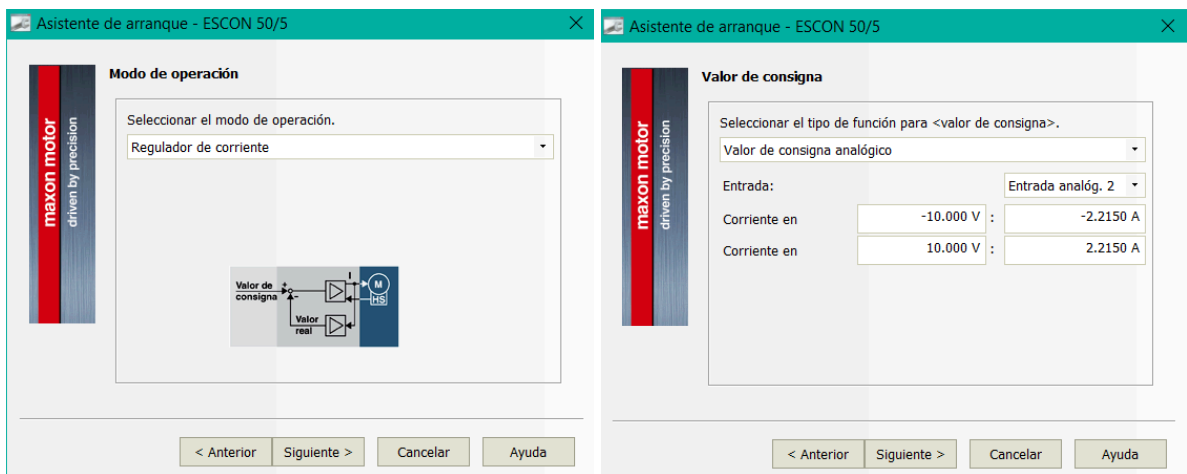


Figura 58: Configuración del control y referencia del motor BLDC

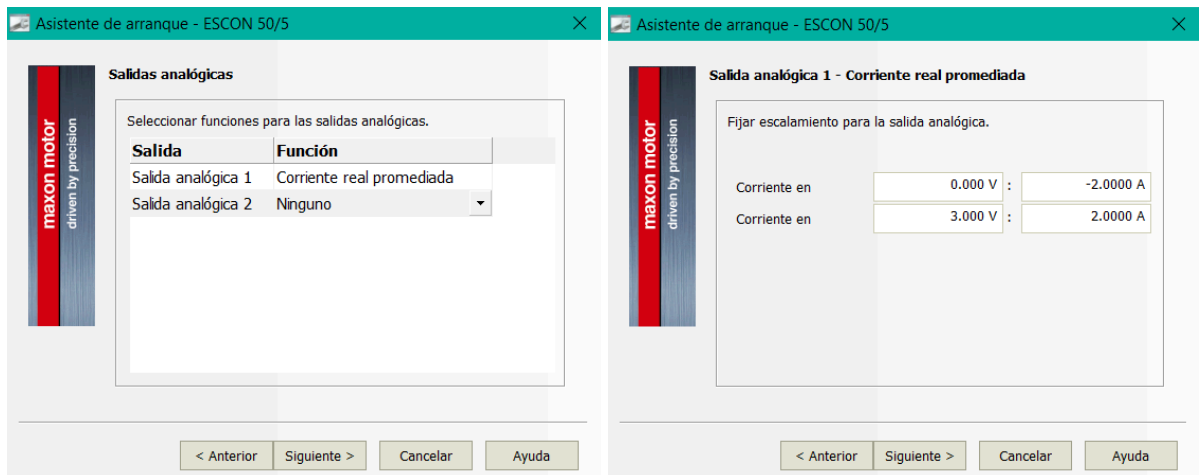


Figura 59: Configuración de la salida analógica de la ESCON

6.2.7. Controlador servomotor

La etapa de potencia del servomotor se trata del driver Digitax HD M751 de la misma marca que el propio motor [38]. Las características a tener en cuenta son:

- Tensión de potencia: 400 V trifásica
- Tensión circuito de control: 24 V
- Corriente nominal: 4.2 A
- Entrada analógica diferencial: 12 bits (de -10 V a +10 V)

Además, para configurar los parámetros del sistema es necesario disponer de un teclado remoto (Remote-Keypad RTC) y conectarlo mediante ethernet. El controlador no dispone de una salida analógica incorporada y, al no disponer de tiempo para adquirir el módulo adicional para este cometido que ofrece la propia empresa, no es posible adquirir el dato de la corriente real. En la imagen 60 se puede ver la conexión del controlador con el teclado.



Figura 60: Controlador Digitax HD con Keypad

Se configura el equipo inicialmente a parámetros por defecto. Luego, se parametriza la placa motor, se configura modo de funcionamiento teclado y se realiza un autoajuste rotativo [39]. Después, se cambia al modo a RFC-S y después se ajusta que la entrada analógica 1 (pr. 07.010) controle la referencia de par (pr. 04.008), se configura también el pr. 04.011 que es el selector de modo de par a "1" que es el modo de control de par (que es equivalente al control por corriente).

Finalmente, en el parámetro 04.024 se ajusta el rango de par que se manipula, en este caso 150% para tener una respuesta efectiva al accionar el secundario de Harmonic Drive [39].

6.2.8. Circuitos de acondicionamiento

En primer lugar, se reduce la tensión de las señales de los encoders de 5 V a 3.3 V a través de varios divisores de voltaje formados por resistencias de 220 Ω (figura 61). La expresión teórica de la operación es la siguiente:

$$V_o = V_i \frac{R}{R+R+R} = 5V \frac{220\Omega}{3 \cdot 220\Omega} = 3.3V \quad (8)$$

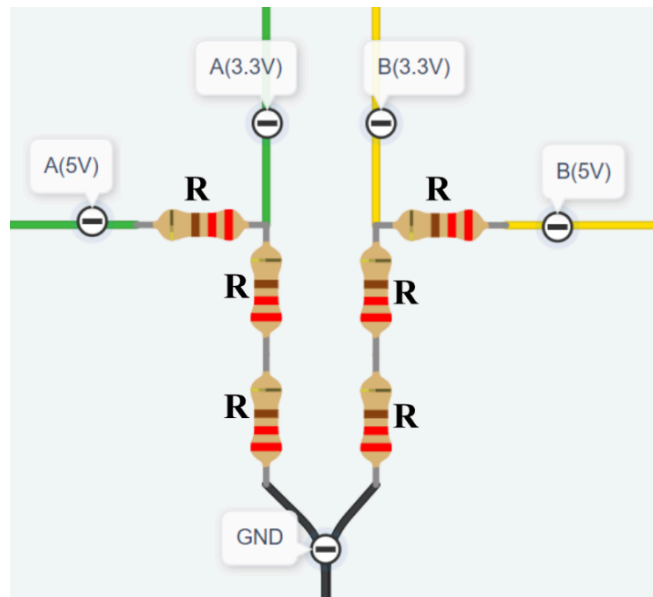


Figura 61: Esquema de divisores de tensión (DT) para cada encoder

En segundo lugar, se amplifica la señal de referencia con un amplificador operacional dual integrado, el LM358 de Texas Instruments. En la figura 62 se puede ver la distribución de pines de este chip.

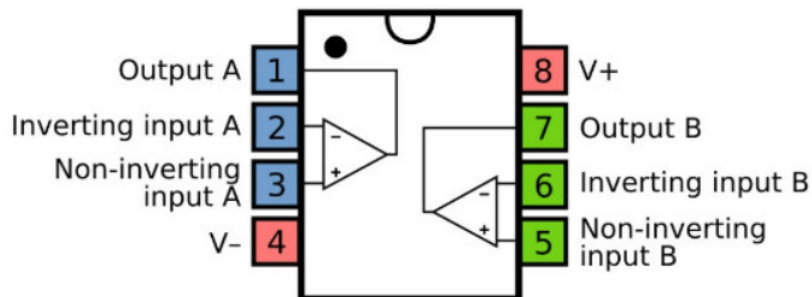


Figura 62: Pines del amplificador operacional dual integrado LM358 [40]

El proceso de diseño de este acondicionamiento empieza por definir la ganancia objetivo que, si la Discovery entrega de 0 a 3 V como valor analógico y se requiere un rango de 20 V para aplicar a los controladores (de -10 V a +10 V), la ganancia debe ser 6.66. Ambos drivers permiten una entrada analógica diferencial por lo que se decide trabajar con un valor fijo de 10 V (terminal restador) y la señal de datos amplificada de 0 a 20 V (terminal sumador). Cabe mencionar que el rango de tensión en modo común es un parámetro que no limita a la ESCON para su entrada analógica pero, en cambio, la entrada del Digitax sí presenta problemas con el rango de 20 V. Por ello, en el caso del experimento 2 (E2) las tensiones de referencia se reducen a un rango de 13 V [39].

En este punto se define el uso de cada AO del integrado. Por un lado, a través de un divisor resistivo se obtienen 10 V fijos de la tensión de alimentación de 36 V. Luego, este pasa por

uno de los AO que está en modo buffer para aislar la etapa resistiva. La expresión es la siguiente:

$$V_{ref1} = V_{cc} \frac{R1}{R1+R2+R3} = 36 \frac{22K}{22K+10K+47K} = 10.025 V \quad (9)$$

Por otro lado, la señal de referencia entra en el integrado atravesando tres grandes resistencias de 470 KΩ (R4) con el objetivo de aumentar la impedancia de entrada y reducir el ruido y las perturbaciones. Finalmente, la ganancia (G) viene definida por la siguiente expresión y combinación de resistencias que más se aproxima a la solución teórica:

$$G = 1 + \frac{R6+R7}{R5} = 1 + \frac{1K+4K7}{1K} = 6.7 V/V \quad (10)$$

El esquema de este circuito de acondicionamiento queda de la siguiente manera:

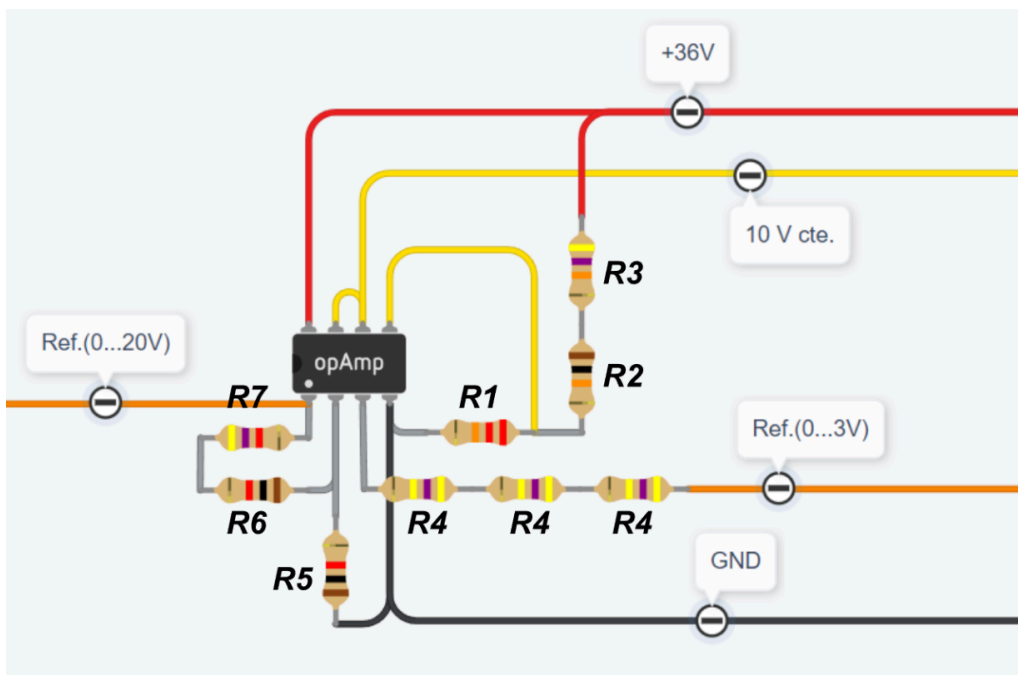


Figura 63: Esquema de circuito amplificador

6.2.9. Alimentación

Es importante alimentar a toda la electrónica lógica de manera aislada de la parte de potencia, para ello se da uso de dos fuentes de alimentación DC a 36 V(imagen 64). Por su parte, el controlador Digitax alimenta su potencia a través de la red trifásica del laboratorio y también da uso de una tercera fuente de alimentación para sus circuitos lógicos como se ve en la imagen 65(alimentación a 24 V).

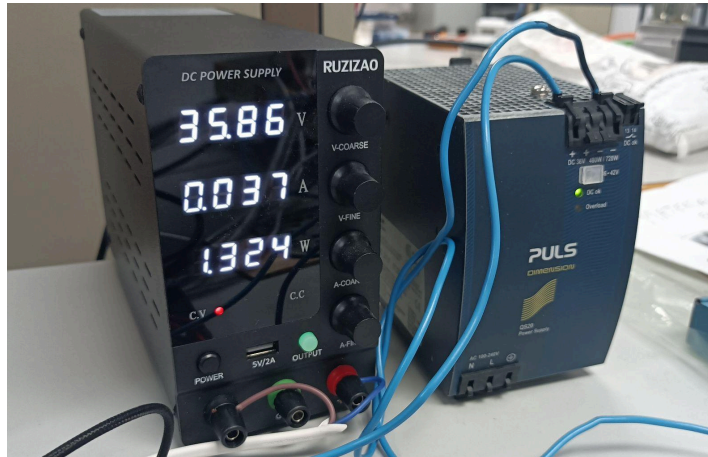


Figura 64: Fuentes DC para Discovery, ESCON y acondicionamientos

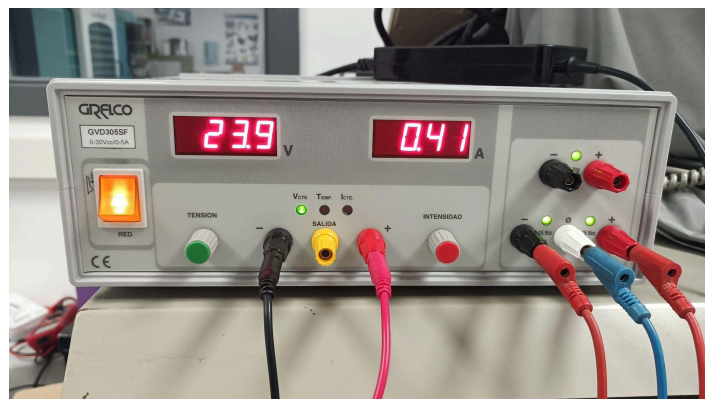


Figura 65: Fuentes DC para Digitax HD

6.2.10. Diagrama de conexiones

Finalmente, el montaje electrónico general queda representado en la figura 66. El diagrama se vincula con el E1, para el resto, es análogo a excepción de los pasos descritos en el documento del pliego de condiciones, donde también se especifican con detalle las interconexiones entre pines y componentes.

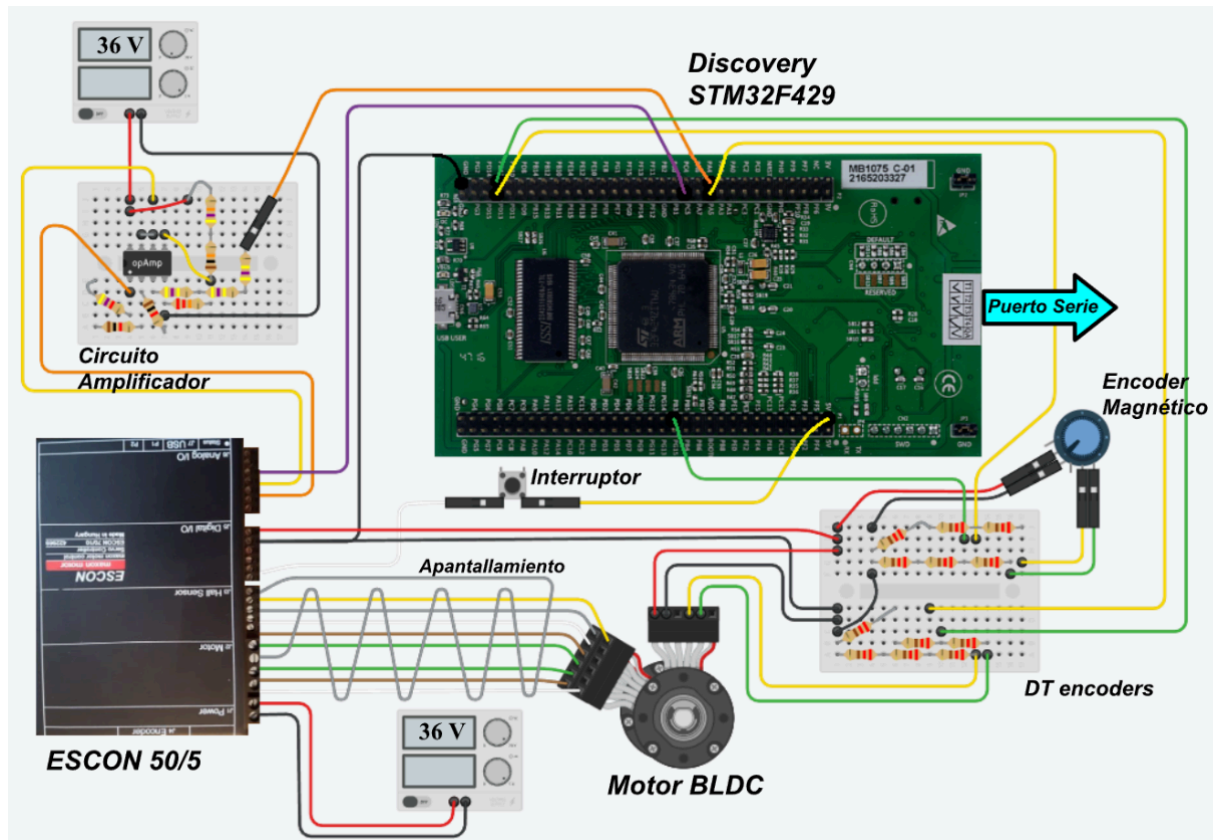


Figura 66: Diagrama electrónico de conexiones

6.3. Subsistema de programación

En el este apartado se describen las configuraciones del microcontrolador y su respectiva programación. Aquí se expone el código más relevante y, con el fin de no entorpecer el desarrollo de la memoria, se destina el resto a los anexos del proyecto.

6.3.1. Configuración en STM32Cube

En primer lugar se crea un proyecto seleccionando la tarjeta STM32F429 Discovery kit y antes de continuar es necesario hacer un reset de todas las configuraciones de los pines para disponer libremente de ellos (figura 67).

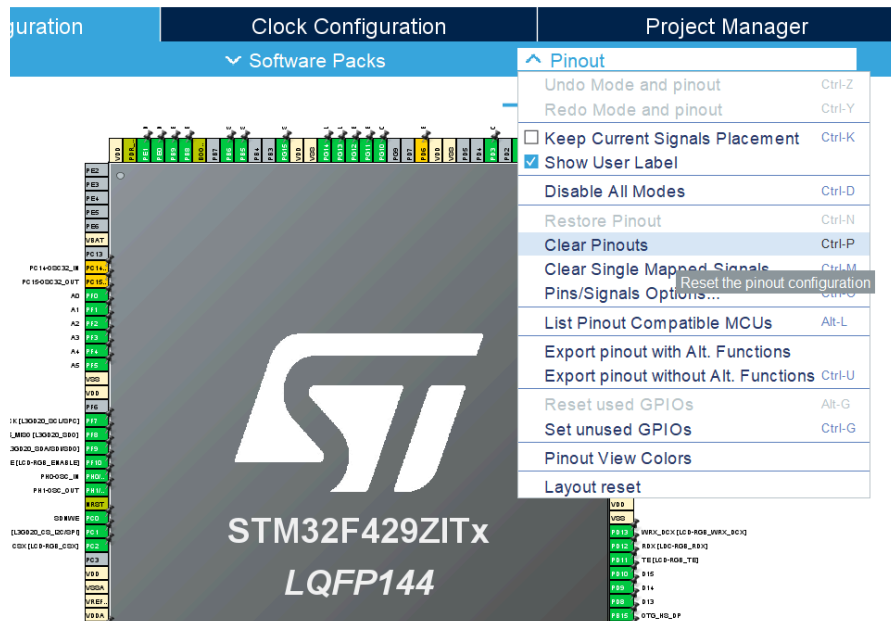


Figura 67: Eliminar configuración por defecto

En segundo lugar se habilita el reloj de alta velocidad (HSE) con el oscilador de cristal/cerámico en el apartado RCC (Reset and Clock Control) de *System Core*. De esta manera ya se puede optar a una frecuencia de 180 MHz muy precisa escribiendo este valor en el recuadro azul (HCLK) de la configuración del reloj y automáticamente el programa encuentra la combinación correcta de *prescalers* y multiplicadores (figura 68). Además, en el proyecto se trabaja con la fuente de base de tiempos de “SysTick” el lugar del timer por defecto.

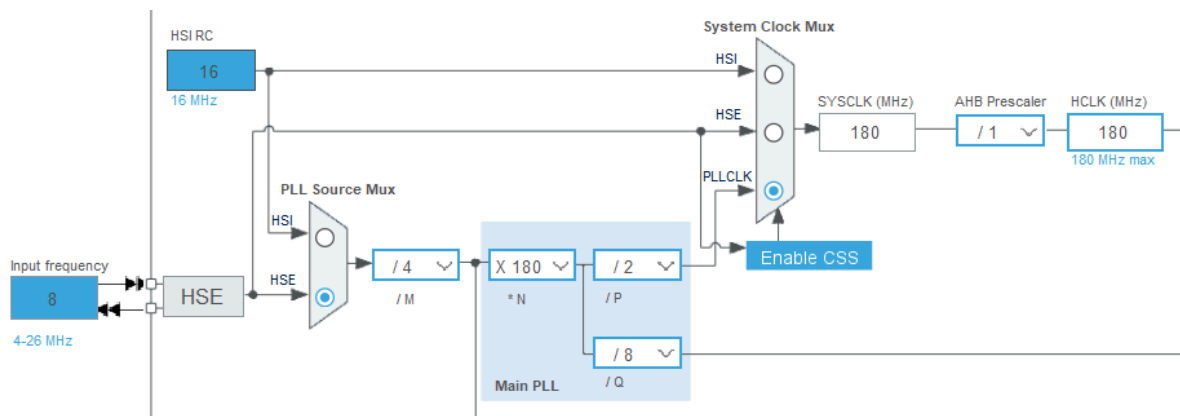


Figura 68: Configuración del reloj en STM32Cube

A continuación, ya se habilitan los periféricos de los cuales se hace uso: el convertidor analógico a digital ADC2 en la entrada 15 y su interrupción, el convertidor digital a analógico DAC en la salida 2, la comunicación serie USART1 a 2 millones de baudios y el generador de números aleatorios RNG. Además, se deben configurar un total de tres temporizadores (timers): TIM8 que marca el ritmo de las interrupciones de muestreo y el resto para funcionar en modo encoder. Por un lado, el TIM8 tiene el objetivo de marcar el momento en el que se ha de disparar la interrupción cada 150 μ s que da comienzo al código temporizado. Para conseguir esto, se configura un prescaler de 180 para dividir el reloj del sistema (180 MHz) y obtener un periodo de 1 μ s para posteriormente ajustar el *counter*

period a 150 cuentas. De esta manera, se crea un contador que desborda a las 150 cuentas que vienen dirigidas por una frecuencia de 1 MHz. Cabe comentar que, a la hora de aplicar estos valores, en el prescaler se debe restar una unidad ya que en la expresión interna se parte de uno para evitar la división entre cero [41]; en el caso del counter period se le resta la unidad porque se empieza a contar desde el 0. En la figura 69 se puede visualizar la configuración referente al TIM8.

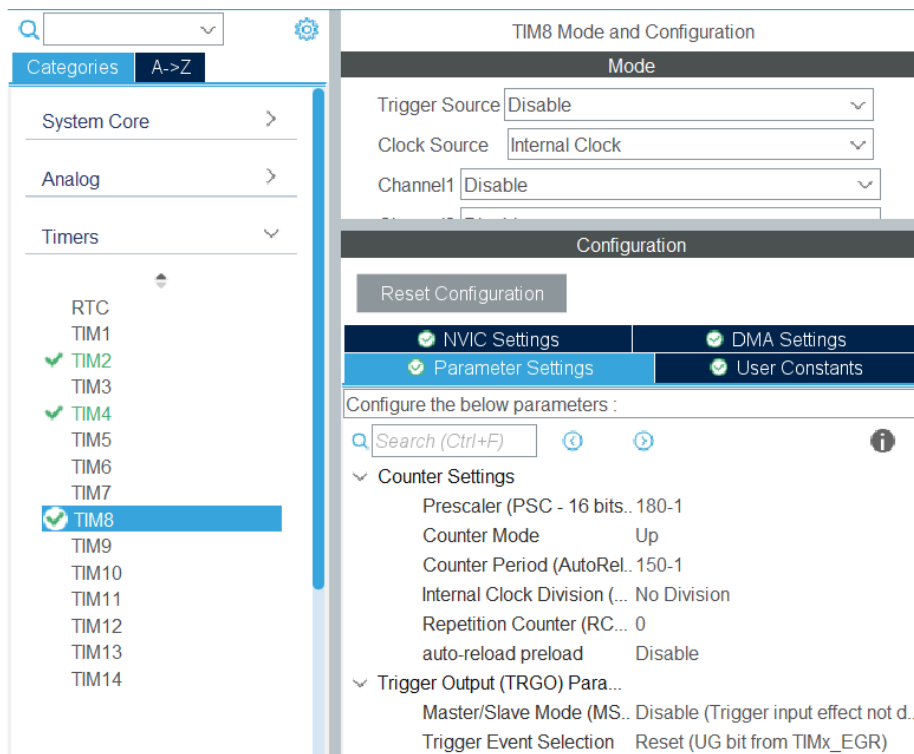


Figura 69: Configuración del timer para las interrupciones temporizadas

Por otro lado, el TIM2 y TIM4 se configuran en modo encoder para el magnético y el óptico respectivamente. En ambos se selecciona “Mode TI1 y TI2” porque se tienen dos canales por cada uno y se ajusta un counter period de 909 cuentas para el encoder magnético y 90 cuentas para el óptico. En la imagen 70 se ve esta configuración en el caso del encoder magnético.

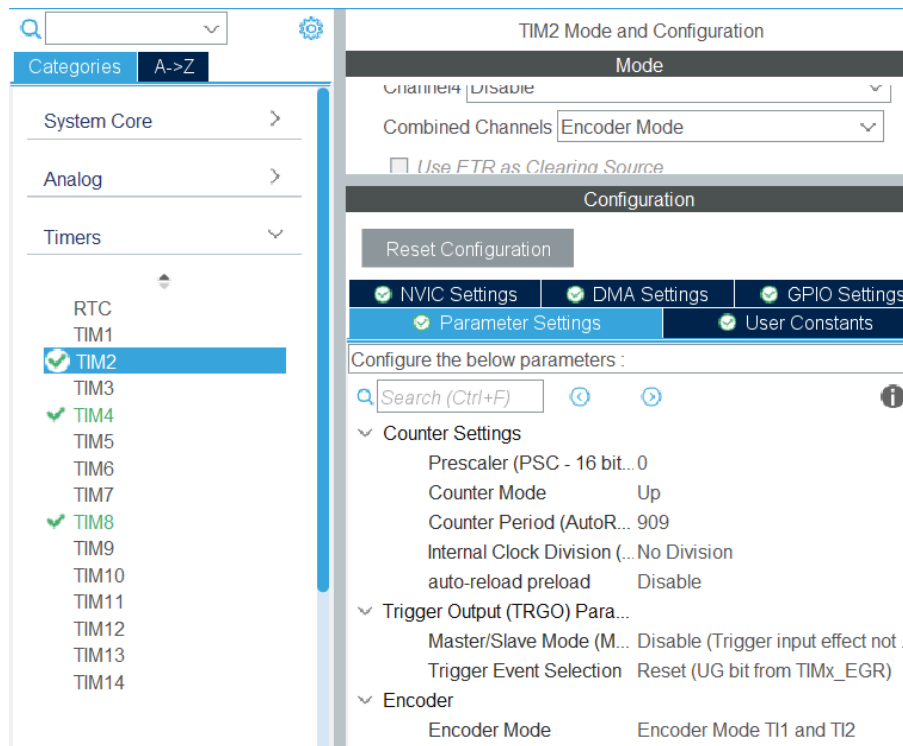


Figura 70: Configuración del timer para el encoder magnético

Antes de guardar y generar el código, se debe ordenar la jerarquía de interrupciones en NVIC (Nested Vectored Interrupt Controller). Se asegura, por tanto, que las interrupciones del TIM8 (la temporización de muestreo) tengan mayor prioridad que las del ADC. En el entorno de STM32, tienen prioridad las interrupciones con un número de preferencia menor.

6.3.2. Código principal

Una vez generado el código, se tienen múltiples carpetas que contienen todos los programas y librerías necesarias en lenguaje C para el funcionamiento de la placa. De ellas solo se va a manipular el fichero principal "main.c" y se va a hacer uso, en general, de funciones de la librería HAL (Hardware Abstraction Layer) que es una biblioteca de software proporcionada por STMicroelectronics como parte de su paquete de software STM32Cube. El objetivo de HAL es ofrecer una interfaz de programación de aplicaciones (API) que permita a los desarrolladores interactuar con el hardware del microcontrolador de una manera más dinámica y portable.

Ahora sí, la primera parte del programa se basa en incluir las librerías necesarias las cuales son: "stdbool.h", "stdlib.h", "stdio.h" y "math.h". Seguidamente se definen constantes y se declaran las variables. También es importante escribir las cabeceras de las funciones e inicializar todos los timers y periféricos que se vayan a utilizar.

En la imagen 71 se tiene el código que se ejecuta cada 150 μ s. La ejecución es la siguiente:

1. Se aplica la acción de control a través del DAC.
2. Se empieza a leer el valor analógico de la corriente en segundo plano.
3. Se leen las cuentas de los encoders.
4. Se guarda el valor analógico de corriente y que hace referencia al valor del periodo anterior ya que el de este periodo todavía sigue con la conversión.

5. Se imprime por el puerto serie la cadena con todos los datos guardados del muestreo anterior y se fuerza la transmisión del buffer de salida.
6. Se comprimen los datos guardados en una cadena de caracteres.
7. Se calcula la acción de control que es aplicada en el siguiente periodo de muestreo.

```

565 /* USER CODE BEGIN 4 */
566
567 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
568     if (htim->Instance == TIM8) {
569         HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, accion);
570         //1 micros
571         HAL_ADC_Start_IT(&hadc2);
572         //Va a tardar unos 60 us en completar la conversión (en segundo plano)
573         counter1 = __HAL_TIM_GET_COUNTER(&htim4);
574         counter2 = __HAL_TIM_GET_COUNTER(&htim2);
575         readCurrent = readADC; //El valor de la corriente está retrasado un periodo
576         //3 micros
577         printf("%s ", data);
578         //11 micros sustituyendo
579         fflush(stdout); //fflush es necesario
580         //64 micros
581         //En esta versión se aumentan los bytes de corriente y duty
582         //va que puede dar cambios muy bruscos, se va a enviar un número de 12 bits
583         compresionASCII(accion, counter1, counter2, readCurrent, data);
584         //67 micros
585         controlCalc(MODO_ESCALON, 1, 0, 1); //segundo arg. 20000 --> 20000*150us = 3s
586         //tercer arg. OFFSET (ejemplos -->220, 275, 300, 950, 1200, 3736(MAX)) E4(140)
587         //cuanto arg. Atenuación (ejemplo: 8 es ocho veces más pequeño) E4(25)
588         //THE END: el código temporizado se ejecuta en 140 micros en el peor caso (Random Mix)
589         //time = __HAL_TIM_GET_COUNTER(&htim8);
590     }
591 }

```

Figura 71: Código principal temporizado

Los puntos 6 y 7 son códigos albergados en dos funciones diferentes: `compresionASCII` y `controlCalc`. En la primera se introducen los cuatro datos que se quieren transmitir (acción de control, contador 1, contador 2 y corriente) y se convierten de números decimales a una versión de menos dígitos aprovechando más símbolos del código ASCII que se introducen en una sola cadena de caracteres sin espacios (`data`). Por limitaciones del rango de símbolos que se puede utilizar (91) solo se comprimen a ASCII las dos cifras más significativas que podrán ir desde 00 hasta el 90 en decimal. Para ello, se separan todos los dígitos decimales de cada dato con ayuda del operador de resto (%) y divisiones. Luego, a la combinación de las dos cifras mayores se introducen en la cadena de caracteres sumando 35 para que el rango de símbolos adquirido sea de uno perfectamente válido que va de los “#” hasta los “}” (de 00 a 90). El resto de números decimales se colocan uno a uno sumando 48 para que coincida con su símbolo decimal ya que no sufren compresión.

La segunda función, `controlCalc` se encarga de calcular la acción de control o referencia de corriente que se ha de aplicar en el DAC. Tiene cuatro argumentos para generar de forma dinámica las señales de control:

- Modo: permite seleccionar qué modo de los 12 predefinidos aplicar para que dentro de la función se seleccione mediante un *switch* si se quiere una señal de escalera, escalones, rampas, senoidales, suma de tres senoidales, sus versiones aleatorias variables, un modo arranque de pulsos, modo de pequeñas rampas para detección de histéresis y barrido de frecuencias.

- Counter Period: como el periodo de muestreo es tan reducido, el cálculo de la referencia puede efectuarse demasiado rápido y no ser efectivo. Para ello, este periodo define cada cuántos muestreos se aplica un cambio en la señal de referencia y, de esta manera, tener un control más flexible y efectivo.
- Offset y atenuación: estos dos últimos argumentos ayudan a la fácil manipulación de la forma de la señal alejándose del cero y reduciendo su amplitud para no saturarla.

Las siguientes figuras presentan los flujogramas explicativos del programa principal y las funciones. Como se ha comentado anteriormente, las líneas del código completas se encuentran en los anexos del proyecto.

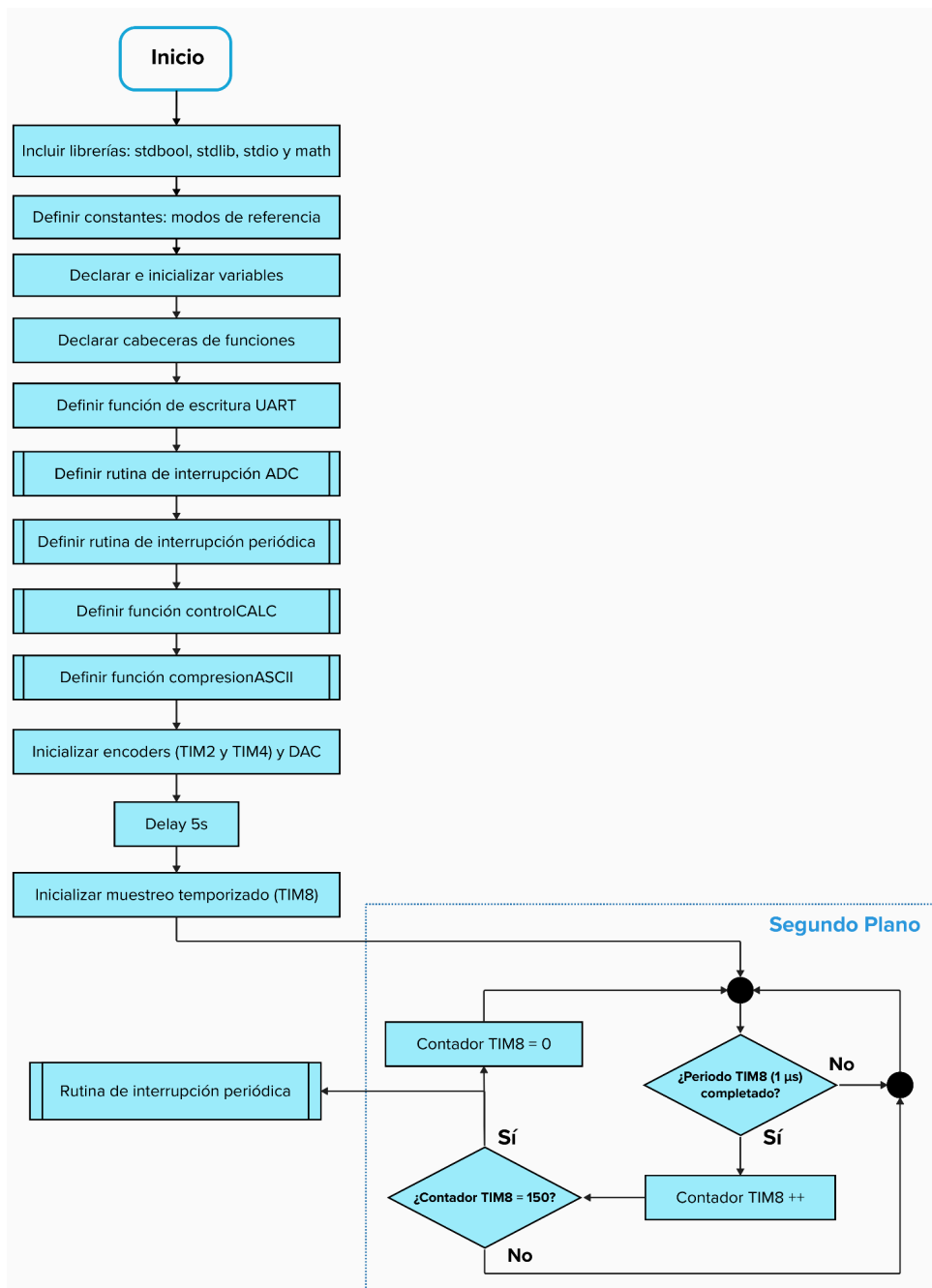


Figura 72: Diagrama de flujo del *main*

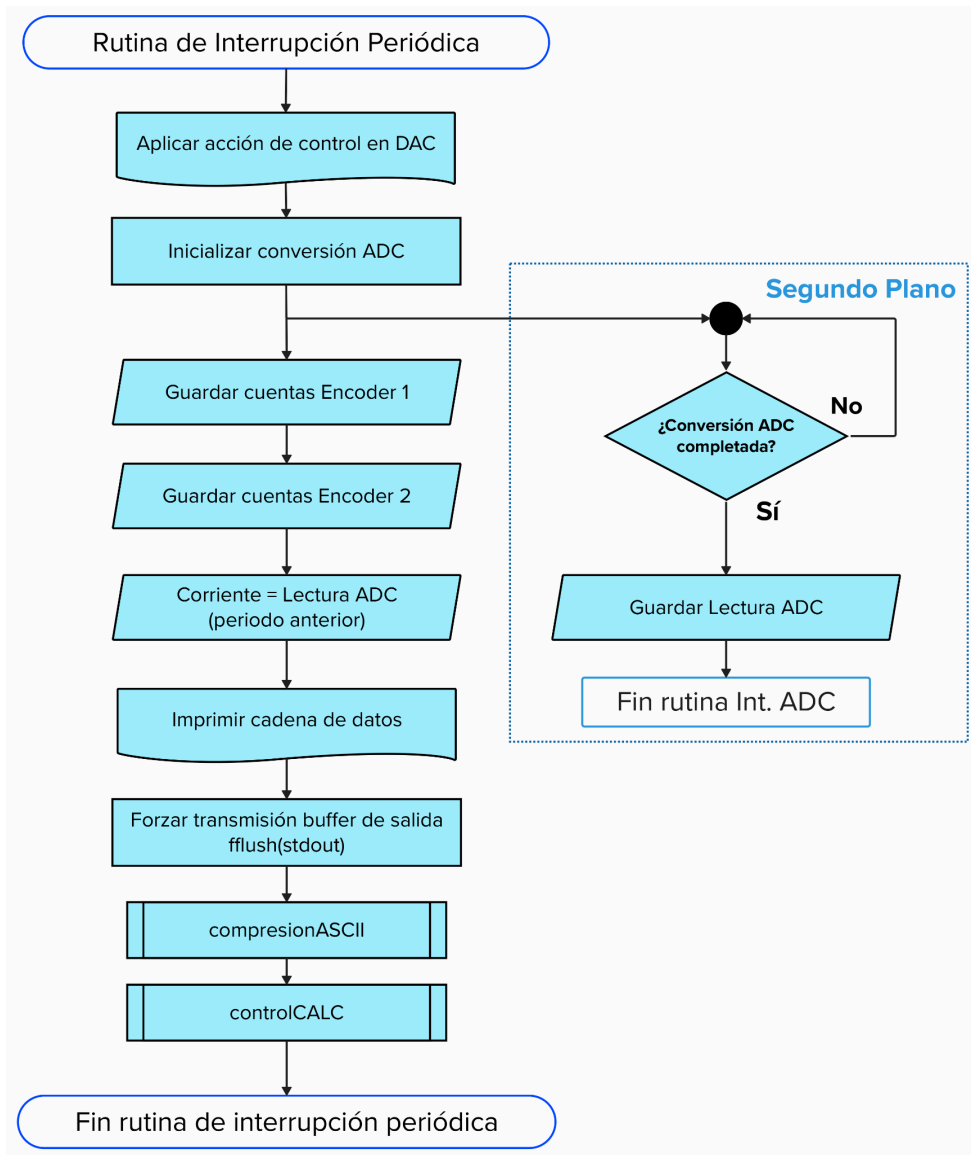


Figura 73: Diagrama de flujo de programa temporizado

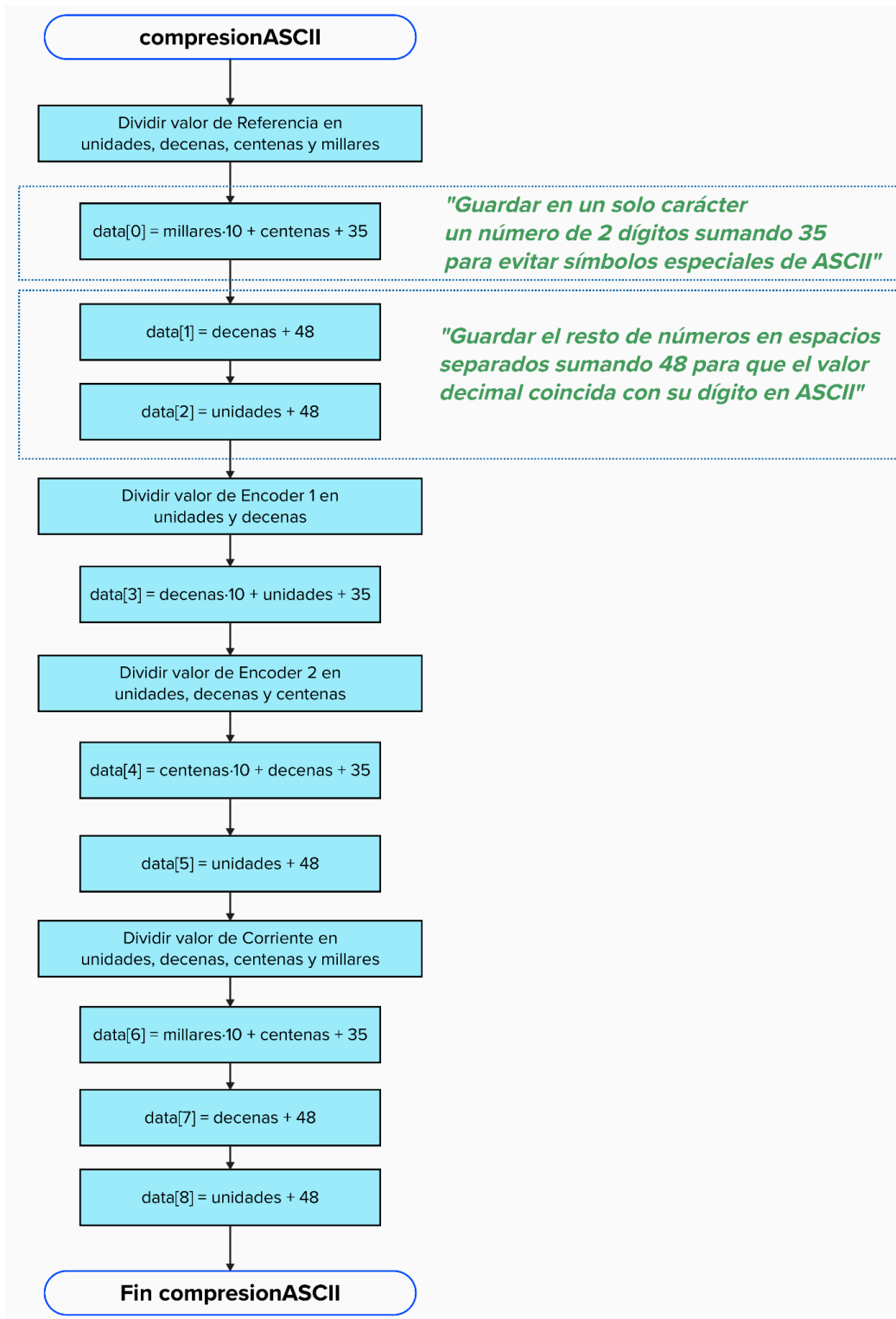


Figura 74: Diagrama de flujo de compresión de datos

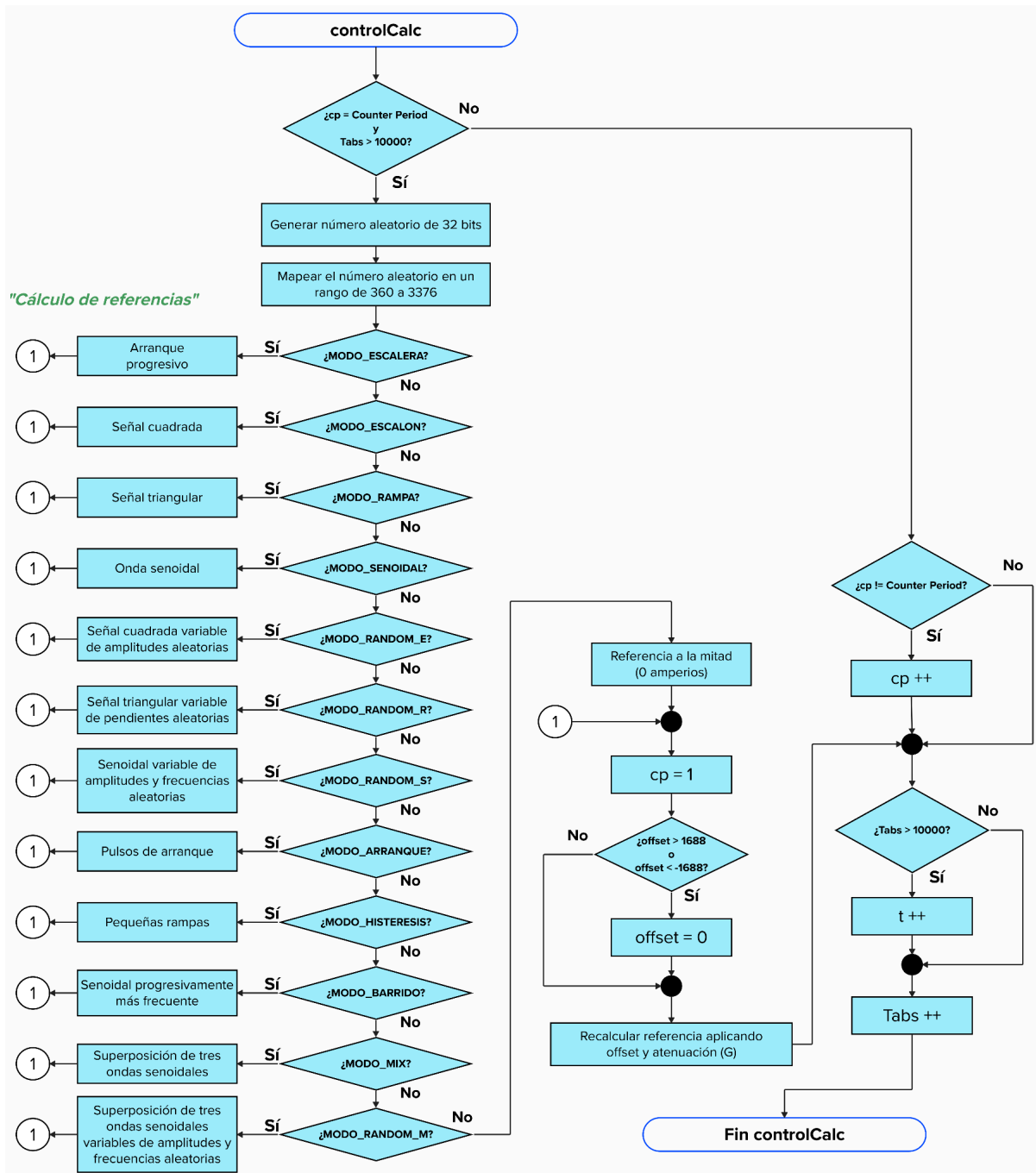


Figura 75: Diagrama de flujo del cálculo de la referencia

6.3.3. Posproceso

En primer lugar, para que se puedan almacenar los datos recibidos por el puerto serie se configura el PuTTY a 2 millones de baudios y para que suba los datos del terminal a un fichero cuyo nombre sea diferente para cada experimento, para ello, se usa “&Y-&M-&D_&T” que anota la fecha y hora de la comunicación serie (figura 76).

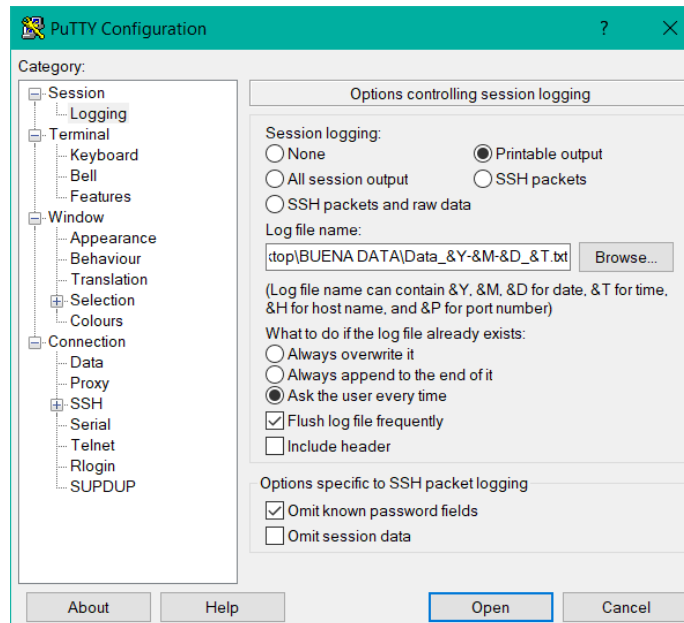


Figura 76: Configuración PuTTY

En segundo lugar, se deben interpretar correctamente los datos en Matlab. Para convertir cada fichero generado en la comunicación serie a una matriz de datos, se debe de convertir primero la tabla importada de una larga fila de datos separados por espacios a una columna. En este punto se tendrán varios cientos de miles de filas, albergando cada una de ellas los datos de cada muestreo (figura 77). Cada fila contiene una cadena de 9 símbolos donde están codificados los cuatro datos de referencia, posiciones y corriente real.

En tercer lugar, se aplica la función `descomprimirASCIIQuad`, especialmente diseñada para recorrer fila por fila los muestreos, dividirlos por los cuatro datos, deshacer la conversión llevada a cabo en en la Discovery e introducir todo en una matriz "A" para poder ya trabajar con los datos experimentales.


```

1  %%DATOS PSEUDO CODIFICADOS ASCII
2  clear all;
3  close all;
4  clc
5  FICHERO = 'Data_2024-07-01_210048.txt';
6  addpath(genpath('C:\Users\Asus\Desktop\TEFEEME\MATLABO'));
7  addpath(genpath('C:\Users\Asus\Desktop\BUENA DATA'));
8  addpath(genpath('C:\Users\Asus\Desktop\Datos Postproceso'));
9  opts = detectImportOptions(FICHERO);
10 opts = setvartype(opts,'string');
11 opts.Delimiter = " ";
12 opts.VariableNamesLine = 0;
13 opts.DataLine = 1;
14 %opts.VariableNamingRule = 'preserve';
15 DATOS=rows2vars(readtable(FICHERO,opts));
16 Aascii = DATOS(:,2);
17 %%
18 A = descomprimirASCIIQuad(Aascii);
19 time = (1:size(A, 1)');
20 B = [time, A];
21 %%14 segundos de procesado para 10 s de exp (4 min para 80 s
22 %%
23 C=B;
24 rango3=91;
25 B3=B(:,3);
26 C(:,3) = deslimitarU(B3,rango3);
27 rango4=910;
28 B4=B(:,4);
29 C(:,4) = deslimitarU(B4,rango4);

```

Figura 77: Posproceso en Matlab

Finalmente, se han de convertir las unidades de los datos al formato de trabajo pero, antes, se deben eliminar los límites de las cuentas de los encoders. Para ello se ha escrito una función llamada `deslimitarU` a la cual se le pasa una fila de datos y un rango. Lo que se hace aquí es, como se comentaba en las alternativas, pasar los datos relativos a absolutos teniendo en cuenta el rango en el que se producen los saltos de la señal recibida. La estrategia de programación seguida es comparar si la diferencia de posiciones consecutivas es mayor a la mitad del rango; si es así, se considera que el contador del encoder ha desbordado y se actúa en consecuencia sumando o restando la diferencia real según si el *overflow* ha sido superior o inferior (figura 78).

```

1  function C = deslimitarU(B, rango)
2
3  [long, ~] = size(B);
4  diff = 0;
5  big = B(1);
6  C(1) = 0;
7  half = rango/2;
8
9  for i = 2:long
10     if abs(B(i)-B(i-1))>half && B(i)>half
11         diff = rango-abs(B(i)-B(i-1));
12         big = big - diff;
13     elseif abs(B(i)-B(i-1))>half && B(i)<half
14         diff = rango-abs(B(i)-B(i-1));
15         big = big + diff;
16     else
17         big = big + (B(i)-B(i-1));
18         diff = 0;
19     end
20     C(i) = big;
21 end
22 end

```

Figura 78: Función `deslimitarU` en Matlab

6.4. Subsistema de control

En este último apartado se describe de forma matemática el modelado de una articulación elástica con reductor así como comentar algunos modelos de rozamiento, se presentan y justifican los ensayos con los respectivos datos experimentales y se hace un acercamiento a la identificación del sistema y de ciertos parámetros del harmonic drive.

6.4.1. Modelado

El modelo básico de articulaciones rígidas presenta la estructura en (11), pero esto cambia cuando se considera un factor elástico en la unión. La elasticidad está presente en mayor o menor medida en los procesos mecánicos reales y, normalmente, se desprecia su efecto. Sin embargo, el harmonic drive presenta un funcionamiento que se basa en la deformación de un componente metálico flexible (flexspline). Es por ello que el sistema se aproxima a un modelo elástico como el de la figura 79 regido por las expresiones matemáticas de (12) y (13).

$$\tau(t) = J \frac{d\omega(t)}{dt} + b\omega(t) \quad (11)$$

El par (τ) en un eje rígido depende del momento de inercia (J), la velocidad angular (ω) y la viscosidad (b).

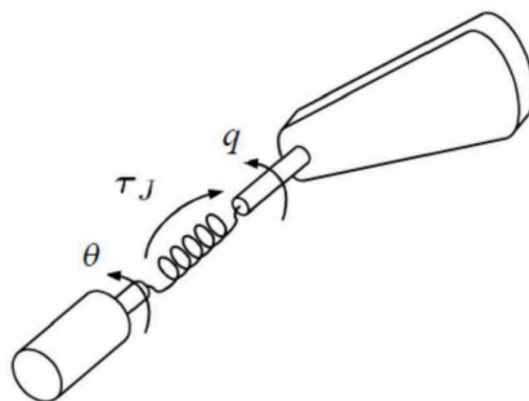


Figura 79: Simplificación aproximada de articulación elástica

$$\tau_m(t) = J_m \theta''(t) + b_m \theta'(t) - K(q(t) - \theta(t)) - B(q'(t) - \theta'(t)) \quad (12)$$

$$0 = J_e q''(t) + b_e q'(t) + K(q(t) - \theta(t)) + B(q'(t) - \theta'(t)) \quad (13)$$

En las expresiones de modelo elástico entran en juego nuevos parámetros que interrelacionan las diferentes entradas y salidas del sistema: el par motor (τ_m); el momento de inercia del motor y del eslabón (J_m y J_e) que influyen al sistema de forma proporcional a las aceleraciones del lado del motor y del eslabón (θ'' y q'') respectivamente; los coeficientes de fricción viscosa del lado del motor y del eslabón (b_m y b_e) que van relacionadas con las velocidades (θ' y q'); la rigidez torsional (que puede tener diferente magnitud según el rango de par [13]) y depende de las posición relativa entre elementos (θ

y q) ya que aquí se define la deformación de la transmisión mecánica; y un coeficiente de amortiguamiento elástico (B) el cuál afecta junto con las velocidades, sin embargo, también hay que considerar que su valor puede no ser constante.

De las expresiones anteriores se puede deducir una forma ejemplo de espacio de estados del modelo de una articulación elástica:

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{K}{J_m} & -\frac{b_m + B}{J_m} & \frac{K}{J_m} & \frac{B}{J_m} \\ 0 & 0 & 0 & 1 \\ \frac{K}{J_e} & \frac{B}{J_e} & -\frac{K}{J_e} & -\frac{b_e + B}{J_e} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u$$

Figura 80: Espacio de estados general para una articulación elástica

No obstante, en este punto cabe aplicar la reducción de velocidad presente el harmonic drive (N) y reconsiderar la estructura de las ecuaciones donde la reducción ahora afecta a los argumentos elásticos, que serán N veces menos significativos en el primario del motor. A su vez, en el cálculo diferencial de posiciones y velocidades de la expresión teórica, ahora se considera también la reducción del primario para que se pueda extraer la resta práctica. Con esto en mente, la inercia desde el lado del motor (J_m) comprende ahora la inercia del rotor y uniones más la del wave generator. Por la otra parte del mecanismo, la inercia en la salida J_e suma el flexspline y los elementos unidos a este otro lado. Además de lo comentado, también se puede considerar en cada momento de inercia, la componente del lado contrario vista desde cada lado (misma estrategia para el coeficiente de rozamiento). La deducción es la siguiente:

$$J_e = \frac{\tau_e}{\omega_e'} = \frac{N\tau_m}{\frac{\omega_m'}{N}} = N^2 J_e^{(1)} \quad (14)$$

$$J_{total} = J_m + J_e^{(1)} = J_m + \frac{J_e}{N^2} \quad (15)$$

De la misma manera:

$$b_{total} = b_m + b_e^{(1)} = b_m + \frac{b_e}{N^2} \quad (16)$$

Por tanto, esto se aplica de la siguiente forma:

$$\tau_m = J_{total} \omega_m' + b_{total} \omega_m \quad (17)$$

Aclarado esto, en la nueva expresión polinómica se aprovecha para incluir el par de rozamiento (τ_{roz}) como perturbación externa y se traduce el par motor a la corriente del motor (i) ya que ambos datos son proporcionales por la constante de par (K_t):

$$i(t)K_t = J_m \theta''(t) + b_m \theta'(t) - \frac{K}{N} (q(t) - \frac{\theta(t)}{N}) - \frac{B}{N} (q'(t) - \frac{\theta'(t)}{N}) - \tau_{roz} \quad (18)$$

$$0 = J_e q''(t) + b_e q'(t) + K(q(t) - \frac{\theta(t)}{N}) + B(q'(t) - \frac{\theta'(t)}{N}) \quad (19)$$

Otra forma de presentar las expresiones anteriores es a través de funciones de transferencia (FDT) en el dominio de Laplace. No obstante, la FDT no puede depender de una perturbación como es el par rozamiento ya que se considera independiente a la entrada. Se tiene, por tanto, la posición del motor (θ) con respecto a la acción de control y la posición del eslabón (q) respecto a la misma entrada:

$$\frac{\theta(s)}{i(s)} = \frac{Kt \cdot (Je \cdot s^2 + (B+be)s + K)}{Je \cdot Jm \cdot s^4 + (Je(\frac{B}{N^2} + bm) + Jm(B+be))s^3 + (K(\frac{Je}{N^2} + Jm) + bm(B+be) + \frac{B \cdot be}{N^2})s^2 + K(\frac{be}{N^2} + bm)s} \quad (20)$$

$$\frac{q(s)}{i(s)} = \frac{Kt/N \cdot (B \cdot s + K)}{Je \cdot Jm \cdot s^4 + (Je(\frac{B}{N^2} + bm) + Jm(B+be))s^3 + (K(\frac{Je}{N^2} + Jm) + bm(B+be) + \frac{B \cdot be}{N^2})s^2 + K(\frac{be}{N^2} + bm)s} \quad (21)$$

Para acabar esta parte, también se obtiene la expresión polinómica deduciendo las velocidades y aceleraciones de los datos de posición y, como se trabaja en el entorno discreto ya que la experimentación no puede adquirir datos del continuo tiempo, se procede a manejar la deducción considerando la presencia del periodo de muestreo (Tm):

$$\theta z^{-2} = \left(\frac{-K \cdot Tm^2}{N^2 \cdot Jm} - Tm \frac{bm + \frac{B}{N^2}}{Jm} - 1 \right) \theta + \left(Tm \frac{bm + \frac{B}{N^2}}{Jm} + 2 \right) \theta z^{-1} + \frac{K \cdot Tm^2 + Tm \cdot B}{N \cdot Jm} q + \frac{Tm \cdot B}{N \cdot Jm} q z^{-1} + \frac{Kt \cdot Tm^2}{Jm} i - \tau_{roz} \quad (22)$$

$$q z^{-2} = \left(\frac{K \cdot Tm^2}{N \cdot Je} + \frac{Tm \cdot B}{N \cdot Je} \right) \theta - \frac{Tm \cdot B}{N \cdot Je} \theta z^{-1} - \left(Tm \frac{Tm \cdot K + be + B}{Je} + 1 \right) q + \left(Tm \frac{be + B}{Je} + 2 \right) q z^{-1} \quad (23)$$

Antes de finalizar, conviene repasar algunos de los modelos del rozamiento que afectan al harmonic drive. En primer lugar, el modelo de fricción de Lugre es un enfoque dinámico que describe el rozamiento como la fuerza media de deformación de resortes elásticos [42]. A nivel microscópico (y no medible), se considera que los numerosos puntos de contacto entre las superficies están formados por hebras elásticas que se oponen al movimiento (figura 81). Cuando una fuerza tangencial actúa sobre estas cerdas flexibles, se deforman como si fueran resortes y, si la deformación es lo suficientemente grande, comienzan a deslizarse. La deformación promedio de las hebras elásticas en estado estacionario se determina en función de la velocidad y, a mayor rapidez, menos oposición de las cerdas. Si la fuerza de fricción viscosa sigue un comportamiento lineal, el modelo de fricción de Lugre se puede representar mediante el siguiente sistema de ecuaciones:

$$\frac{dz}{dt} = v - \left(\sigma_0 \cdot |v| \cdot \frac{1}{g(v)} \right) \cdot z \quad (24)$$

$$F = \sigma_0 z + \sigma_1 \frac{dz}{dt} + F_v v \quad (25)$$

$$g(v) = F_c + (F_s - F_c) e^{-(v/v_s)^2} \quad (26)$$

En las expresiones anteriores, z es la variable que representa la deflexión promedio de las hebras elásticas, v es la velocidad relativa entre las superficies de contacto, σ_0 es el coeficiente de rigidez, $g(v)$ es la función de la velocidad que modela el efecto de Striebeck, F la fuerza de rozamiento, σ_1 el coeficiente de amortiguamiento y F_v un coeficiente de proporcionalidad.

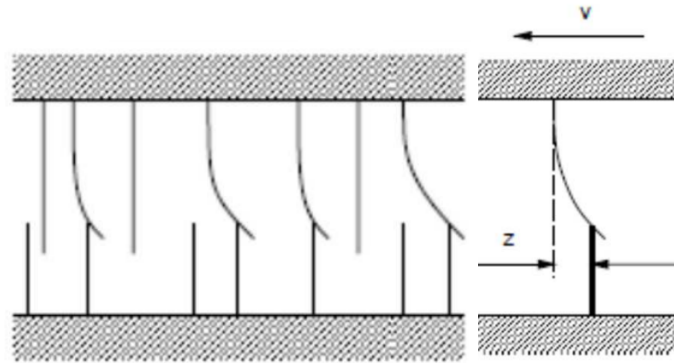


Figura 81: Representación aproximada de las hebras elásticas

No se va a profundizar más en los modelos de rozamiento y sus expresiones ya que no es el objetivo del proyecto. Sin embargo, sí que cabe hacer una última exposición de la fricción presente en los harmonic drives, comentando el efecto Striebeck y las fuerzas de Coulomb.

Cuando se habla de fricción, se puede diferenciar entre rozamiento estático y dinámico. El primero es aquel que debe ser superado para iniciar el movimiento y el segundo corresponde con la fricción que afecta una vez en movimiento y puede tener un valor diferente. En la figura 82 se pueden ver los diferentes efectos de la fuerza de rozamiento considerando el efecto estático y dinámico de Striebeck. Cabe comentar que las fuerzas representadas aquí se opondrán al movimiento y es por ello que, en los modelos matemáticos se deben considerar de signo contrario a la dinámica. En este proyecto, se valora el modelo básico a) que comprende las fuerzas de Coulomb como efecto constante con la velocidad, y el caso c) y d) que presentan fuerzas estáticas mayores que las dinámicas ya que se prevé que el harmonic drive demanda mayor par en los inicios del movimiento según los valores de torque de la hoja de características.

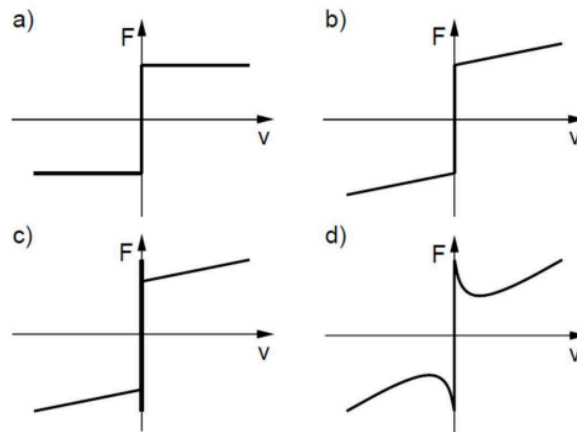


Figura 82: Modelos de fuerza de rozamiento según la velocidad

Para acabar, si se pone a prueba la back drivability, la fuerza de rozamiento cuando el harmonic drive es accionado por el secundario afecta de forma diferente que si se hace por el primario. Cabe recordar que, como reductor, se amplifica el par a coste de reducir la velocidad. La relación de velocidades es estrictamente 1:N, mientras que la relación de pares es N como forma ideal, pero en la realidad existen pérdidas en el par transmitido. Se habla, por tanto, de una fuerza de rozamiento que, vista desde la salida, se amplifica en N.

6.4.2. Ensayos y datos experimentales

En este apartado se explica qué forma tienen los experimentos objetivo (E1, E2, E3 y E4) y cómo se han desarrollado en la implementación física. En la imagen 83 se pueden ver los cuatro montajes: E1 consiste en el accionamiento del reductor desde el primario, dejando el lado del eslabón en vacío; E2 acciona el harmonic drive desde la salida mientras el primario está en vacío; E3 es similar al primero pero esta vez se aplica cierta carga mediante el motor de E2; finalmente, en E4 se prescinde del circular spline y se hacen pruebas como en E1 pero con el flexspline bloqueado. El objetivo de este último montaje es simplificar el sistema a identificar reduciendo el número de elementos que lo componen.

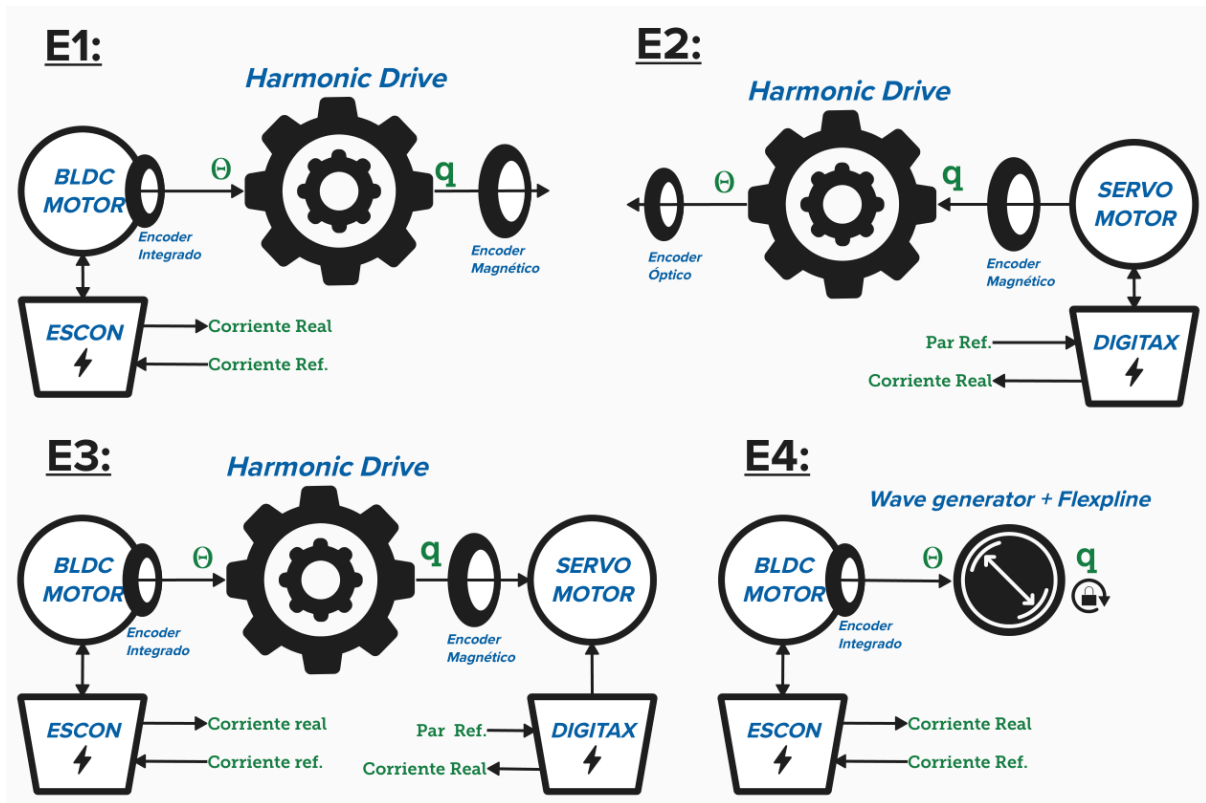


Figura 83: Esquema de experimentos objetivo

Una vez definido el planteamiento de experimentación, en la implementación real se ha precisado hacer cambios por falta de tiempo y de material. En primer lugar, se consigue la implementación de E1, pero tras la recolecta de datos y realizando pruebas de estabilidad adicionales se ha detectado una mala alineación del motor con el harmonic drive. Esta leve desalineación entre ejes se compensa por la junta Oldham que lleva integrada el harmonic drive pero introduce vibraciones al sistema que pueden ser minimizadas reajustando ligeramente la base de experimentación en tiempo de ejecución. Se decide, por tanto, incluir en los resultados del proyecto los datos de E1 (ejes bien alineados) y E1' (ejes desalineados). Evidentemente en E1 los ejes no están alineados a la perfección por limitaciones físicas ni en E1' están peligrosamente desalineados, no obstante, la diferencia en vibraciones es muy significativa.

En segundo lugar, la formas de E2 y E3 han tenido que ser modificadas. Por falta de tiempo no se ha podido colocar el primario de E2 en vacío con el encoder óptico y se ha optado por la solución de mantener el motor BLDC unido para dar uso de su encoder integrado. Además, tampoco se ha podido obtener el complemento necesario del controlador Digitax que devuelve el valor de analógico de la corriente real y, con lo anterior mencionado, el resultado es E2' y E3'.

En último lugar, el E4 sí se implementa al completo sin cambios. En la imagen 84 se presenta el resultado del planteamiento y forma final de los experimentos que en este punto están conformados por E1, E1', E2', E3' y E4.

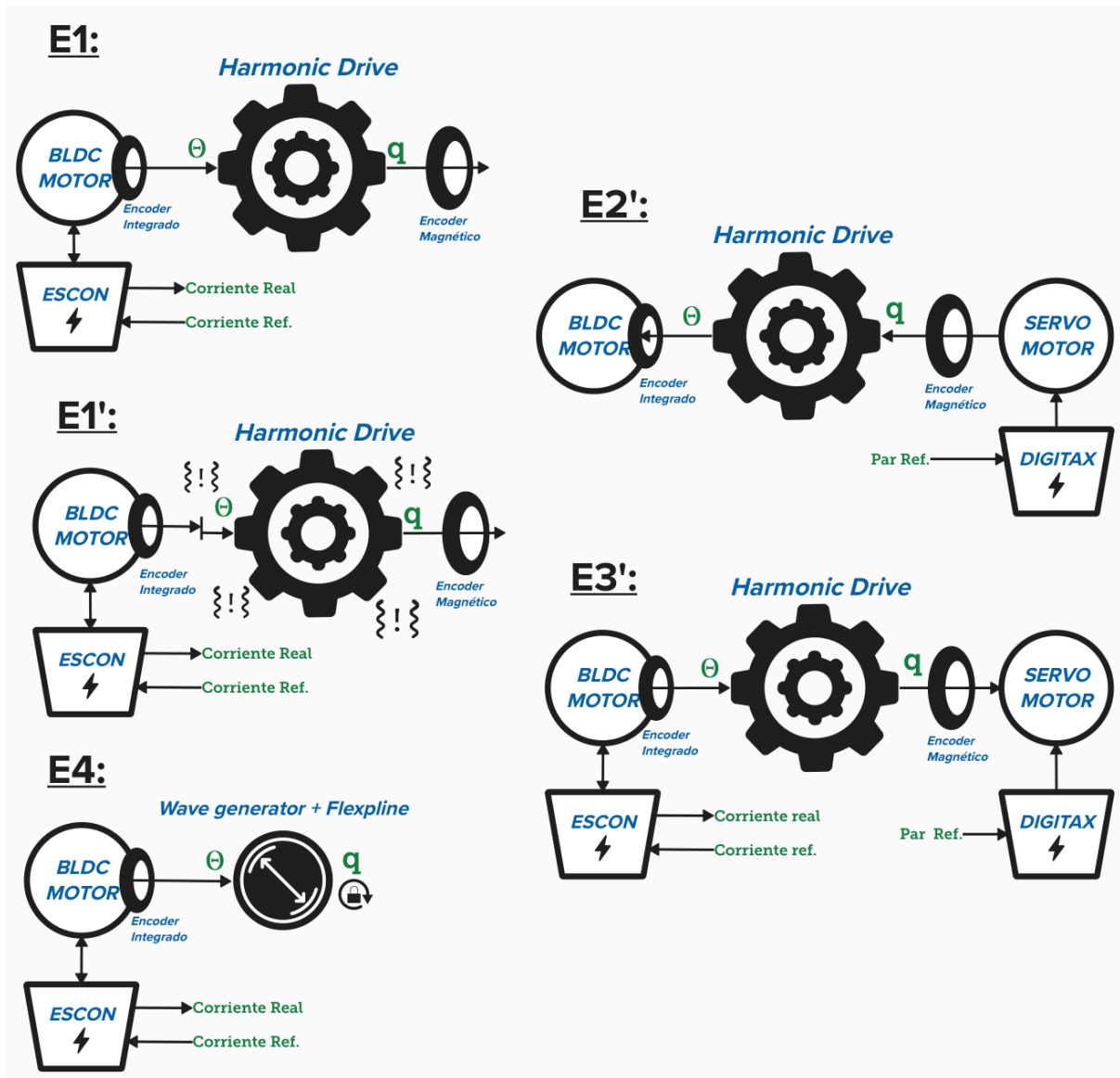


Figura 84: Esquema de experimentos resultantes

En las siguientes imágenes se presenta el aspecto de los montajes finales de los experimentos resultantes:

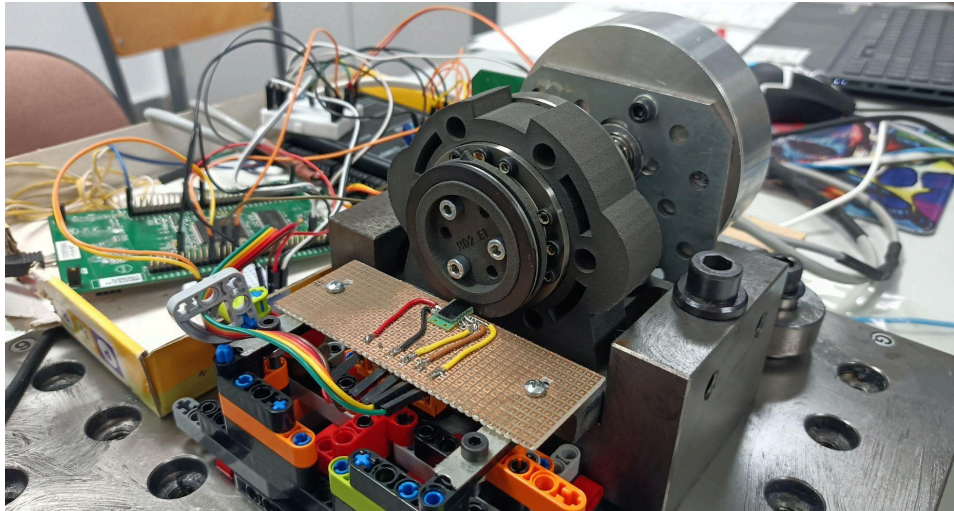


Figura 85: Montaje experimentos E1 y E1'

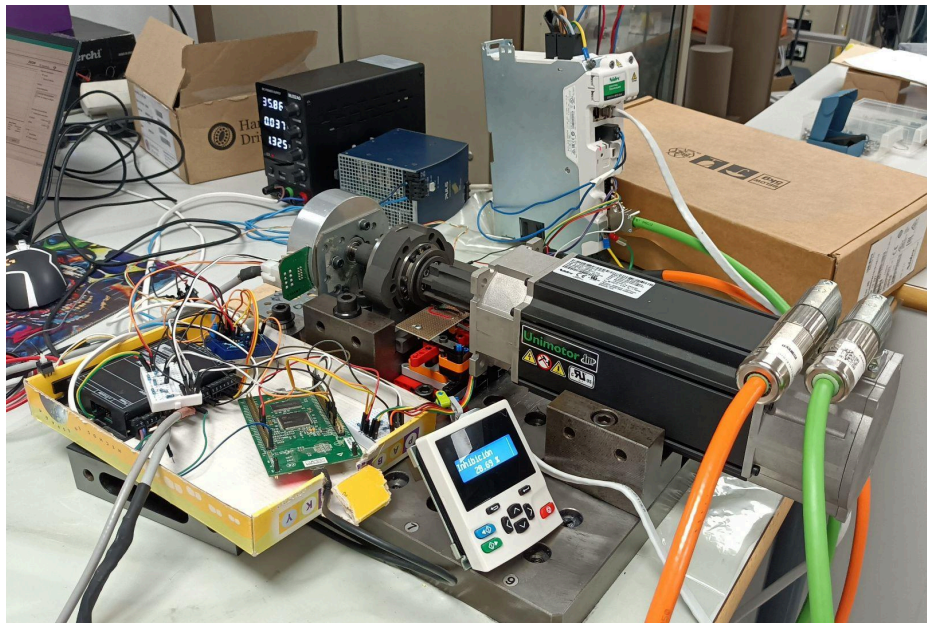


Figura 86: Montaje experimentos E2' y E3'

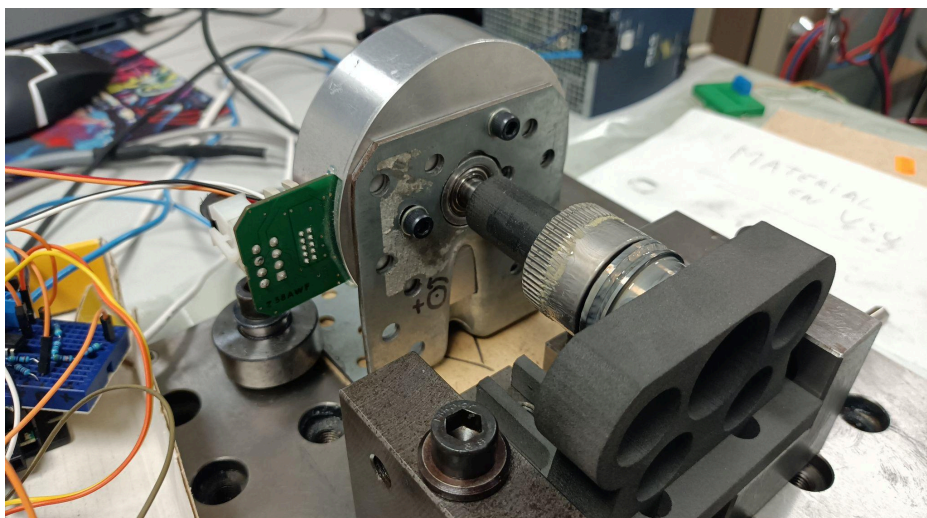


Figura 87: Montaje experimento E4

Finalmente, las acciones de control aplicadas en cada uno de los experimentos son análogas entre sí. La intención es dotar de la excitación suficiente al sistema y, por ello, se trabaja con pulsos, escalones, rampas y senoidales incluyendo también versiones aleatorias y con offset de estas en el sistema para una mejor identificación. El tipo de señal más excitante con el que se trabaja es la suma de tres señales senoidales de frecuencia y amplitud aleatorias variables en el transcurso de cada ensayo. Cuanto más alto es el orden de la señal introducida mejor es la identificación de sistemas de órdenes superiores. Además, la señal de control es una referencia de corriente (proporcional al par) por lo que se lleva a cabo un control modo corriente (o modo par en el caso del servomotor). Esta referencia es proporcional a la aceleración del eje de los motores y, por tanto, hay que tener en cuenta que la velocidad puede llegar a saturar en su valor máximo de funcionamiento. Ante esta situación, el controlador del motor BLDC baja automáticamente la corriente para mantener la velocidad máxima y el controlador del servomotor, por otro lado, hace un paro de emergencia. En los ensayos y adquisición de datos se evita esta zona de saturación de la etapa de potencia aplicando acciones de control rápidas o con una baja exigencia de par para que los motores no aceleren hasta la velocidad máxima.

Todos los datos obtenidos y posprocesados suman más de 300 ensayos y superan el GB de memoria y, por ello, se han guardado en la nube para el libre acceso y aportación a futuros proyectos en esta línea de investigación. Así pues, se pueden encontrar los archivos con los datos experimentales de los diferentes ensayos en el siguiente enlace:

[Datos Posproceso Harmonic Drive](#)

6.4.3. Identificación

Como ya se ha comentado con anterioridad, el presente trabajo no tiene como objetivo llevar a cabo una identificación completa de todos los aspectos que conciernen al harmonic drive. De todas maneras, sí se busca llevar a cabo ciertas interpretaciones de los datos adquiridos y hacer un acercamiento a algunas de las estrategias de identificación. Para ello, solo se trabaja con los datos adquiridos de E1 y E1' dadas las limitaciones de tiempo mientras que se deja el resto a disposición de futuros trabajos.

Para empezar, como se tiene la medición de la corriente real del motor (al menos con el uso de la ESCON 50/5), conviene tenerla en cuenta como entrada en lugar de la referencia o acción de control. Esto es así porque no se tiene la intención de identificar la etapa de potencia (controlador y motor); lo que realmente se pretende es tratar el sistema mecánico del harmonic drive, dependiendo lo mínimo posible de factores externos como es la dinámica interna del motor y el control que integren los drivers.

Ahora ya, en el siguiente punto conviene empezar analizando las pruebas de arranque para estimar el par mínimo necesario para iniciar el movimiento (tanto en E1 como en E1'). En la imagen 88 se puede ver cómo al no aplicar corriente suficiente (proporcional al par necesario), el mecanismo hace un intento de movimiento pero no lo consigue. En la imagen 89, sin embargo, la corriente ha ido en aumento progresivo con en modo escalera hasta conseguir mover el harmonic drive en aproximadamente 0.4815 A que, aplicando la constante de par Kt del motor BLDC (0.217 Nm/A) corresponde con unos 0.1045 Nm.

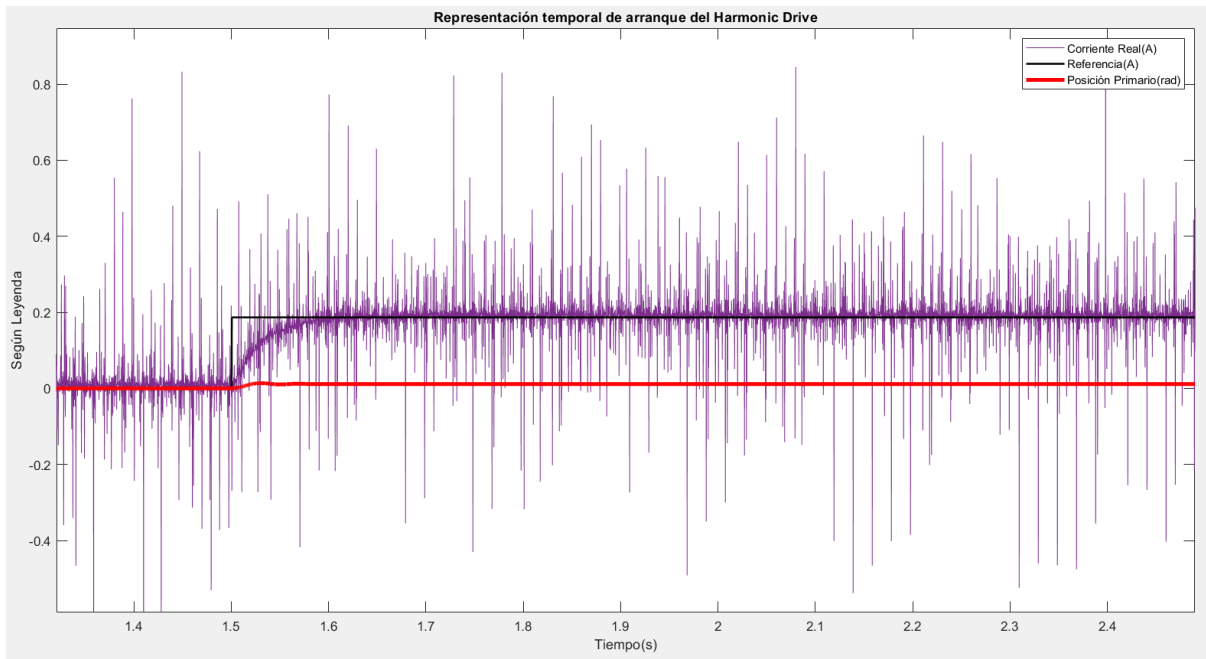


Figura 88: Escalón insuficiente para iniciar movimiento

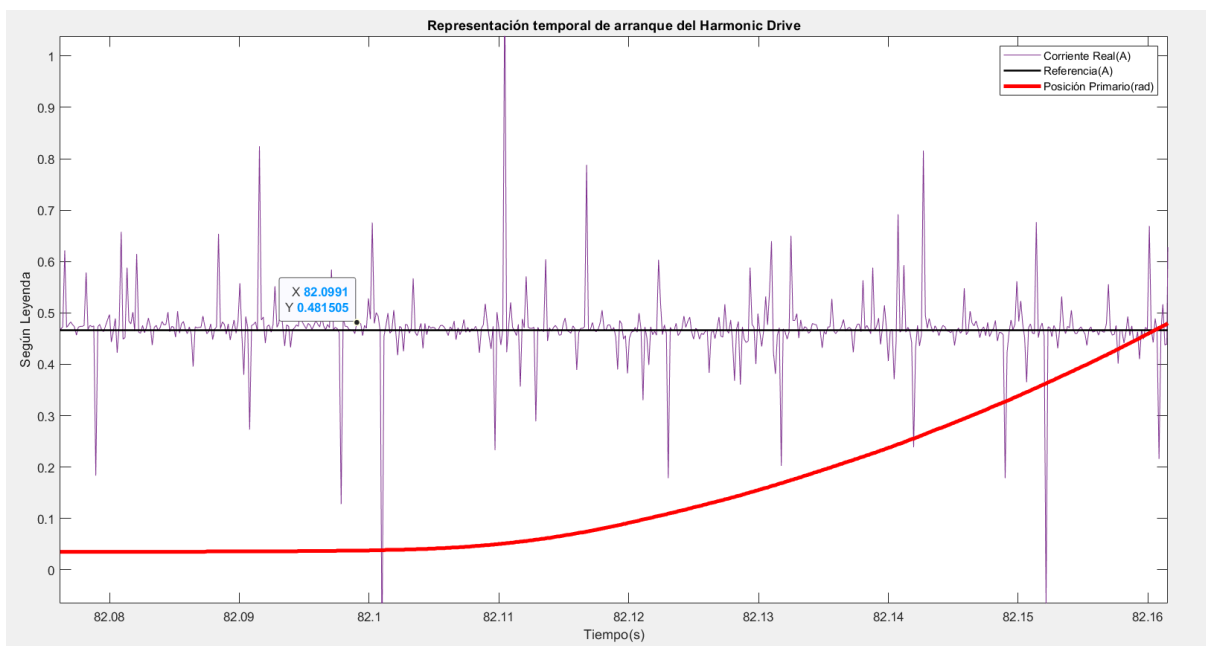


Figura 89: Punto del modo escalera en el que comienza el movimiento

Tras comparar numerosas señales en escalera partiendo desde diferentes posiciones y en ambos sentidos se obtiene un par de arranque que está en un rango aproximado entre 0.0611 Nm a 0.1519 Nm. No obstante, esto es aplicando un par que aumenta lentamente, si se aplican pulsos de 0 a una amplitud creciente intermitentes se recolectan datos en un rango de 0.0533 Nm a 0.0797 Nm aproximadamente. Los valores obtenidos son más bajos debido a que el arranque ahora es más agresivo y, por ello, el movimiento empieza con pares en un rango ligeramente menor. En la figura 90 se demuestra que el par de arranque no es el mismo cuando se alcanzan diferentes posiciones a través de los pulsos mencionados.

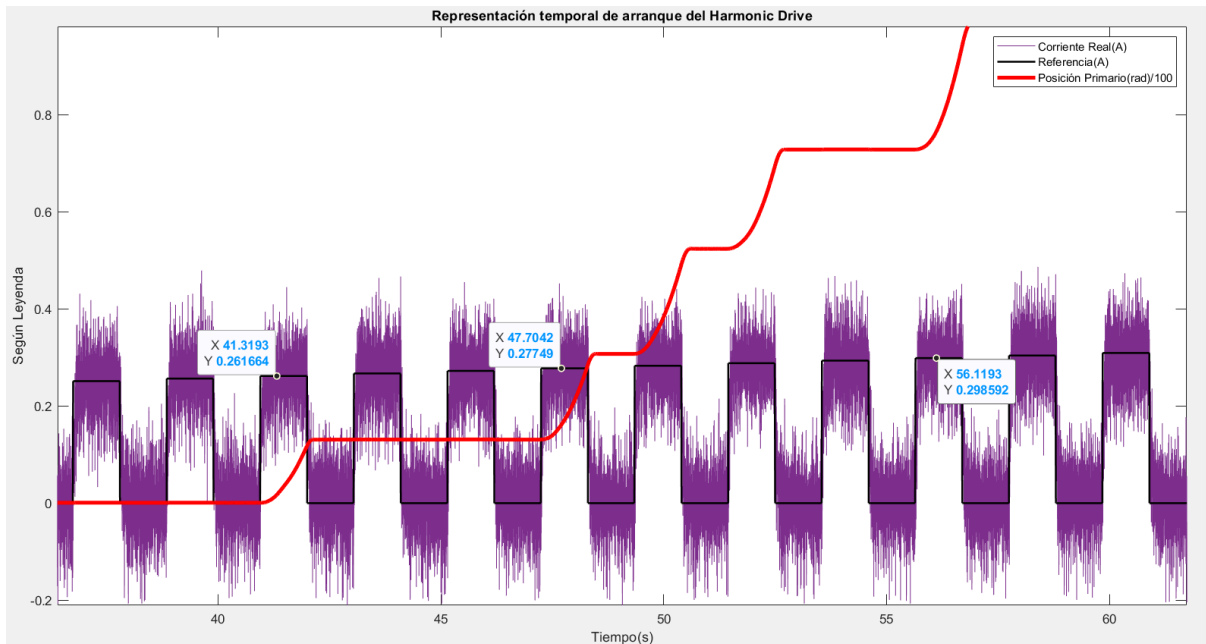


Figura 90: Arranque con impulsos crecientes

El par de inicio en vacío del harmonic drive según la hoja de características es 0.064 Nm y el promedio aproximado del par de arranque experimental mediante impulsos es 0.0665 Nm, el cual se aproxima bastante. Quizás el valor es ligeramente superior por la presencia de elementos extra en el giro como el Soporte E1, el Eje E134, tornillos y encoder magnético.

Por tanto, este valor de par identificado puede considerarse equivalente al parámetro de la fuerza de rozamiento estática presente en el modelo de Striebeck y que se espera que tenga un valor menor que la fricción dinámica una vez iniciado el movimiento. La fuerza de rozamiento de Coulomb en el modelo básico se considera igual al iniciar el movimiento que luego, pero otros modelos del propio efecto Striebeck sí que se considera un esfuerzo inicial mayor.

De todas maneras, conviene llevar a cabo una identificación de los parámetros incluyendo el rozamiento en el proceso dinámico. Para ello, se programa de forma manual un código que da uso del método de mínimos cuadrados y se construye también el modelo del sistema en cuanto a una articulación elástica con reductor. Cabe aclarar que, en este caso, para poder incluir el par de rozamiento, se trabaja con las expresiones polinómicas discretas obtenidas en el apartado de modelado.

En la imagen 91 se puede ver cómo se ha construido la matriz de regresores teniendo en cuenta un posible retraso entre la reacción de las diferentes variables. Además, siguiendo la estructura del modelo polinómico, se puede considerar como entradas (matriz Phi) las posiciones actuales, las anteriores, la corriente y el par de rozamiento que se puede dejar como una constante con una columna de unos. En la matriz de salida Y se han introducido como salidas las posiciones de dos periodos anteriores.

```

DELAYIth = 1;
DELAYthq = 3 + DELAYIth;%21,32,10,7

NL = length(T);
Phi = zeros(NL, 6); % Matriz de regresores
Phi(1:(end-DELAYIth), 1) = th((DELAYIth+1):end);
Phi(2:(end-DELAYIth), 2) = th((DELAYIth+1):end-1);
Phi(1:(end-DELAYthq), 3) = q((DELAYthq+1):end);
Phi(2:(end-DELAYthq), 4) = q((DELAYthq+1):end-1);
Phi(:, 5) = I;
Phi(:, 6) = ones(NL,1);

Y = zeros(NL, 2);
Y(3:(end-DELAYIth), 1) = th((DELAYIth+1):end-2);
Y(3:(end-DELAYthq), 2) = q((DELAYthq+1):end-2);

PhiT=Phi.';

```

Figura 91: Formación de una matriz de regresores

En las siguientes imágenes se comparan las respuestas de los datos reales con los datos teóricos y estimados. Como no se conocen los valores del coeficiente de fricción viscosa, del coeficiente de amortiguamiento elástico ni del rozamiento del harmonic drive, se han considerado valores arbitrarios en el modelo teórico.

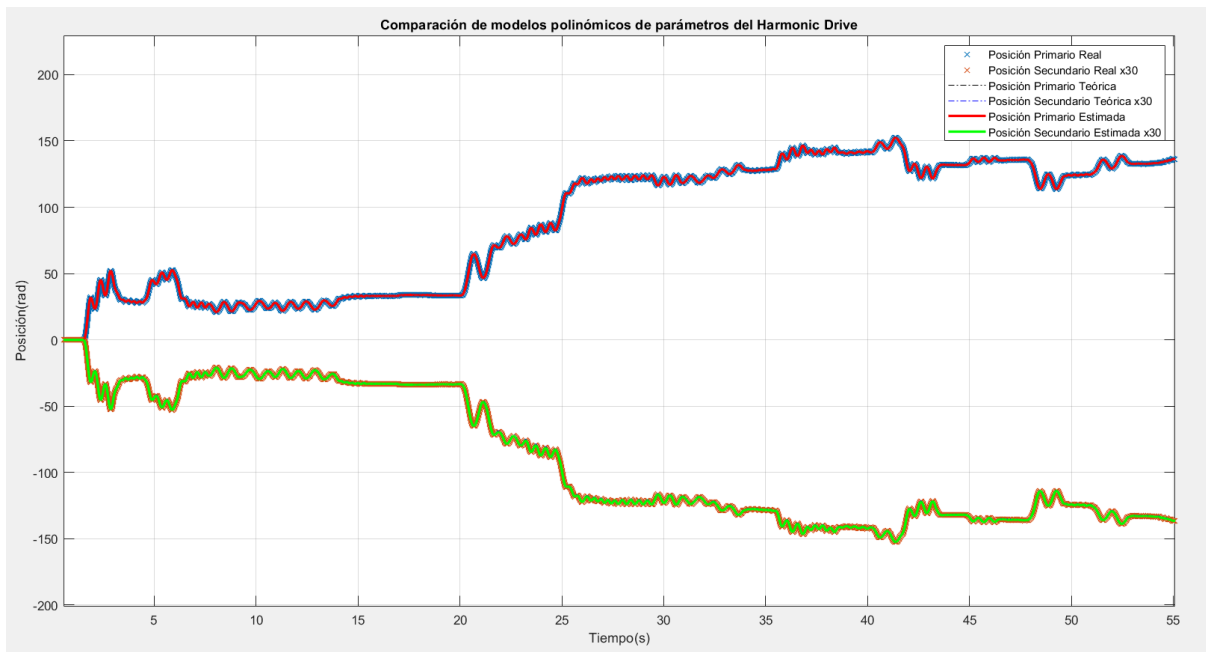


Figura 92: Representación de las curvas de posición reales, teóricas y estimadas

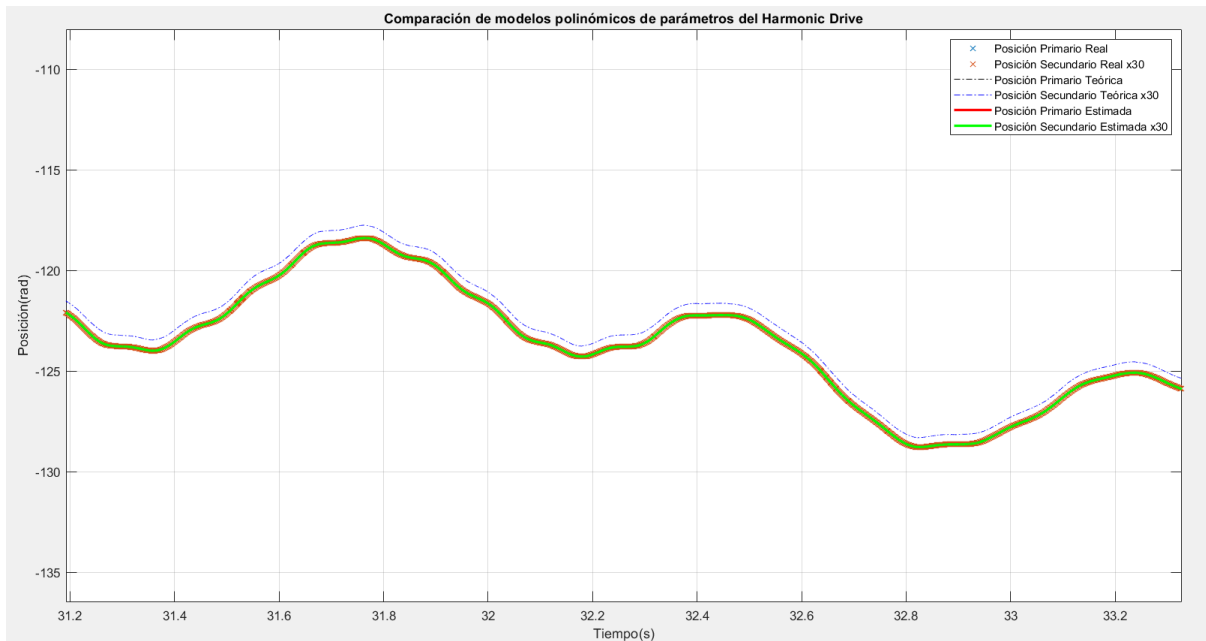


Figura 93: Detalle de las curvas de posición reales, teóricas y estimadas

Se ha visto que las estimación copia con precisión a la tendencia real. Sin embargo, esto es un resultado sesgado. La justificación radica en que, al dar uso de un sistema polinómico y dejar que la estimación por mínimos cuadrados interrelacione las diferentes variables y encuentre la solución aproximada, los resultados pueden no tener una causalidad correcta al no hacer una correcta clasificación de las entradas y salidas. En la imagen 94 se muestra la comparación entre las matrices de coeficientes (cada uno es una combinación de parámetros según el modelo) estimados y teóricos. Al no haber una buena correlación se concluye que es necesario invertir en otras estrategias de identificación.

$XE =$ $\begin{bmatrix} -0.349549334927500 & 0.025126584664368 \\ 1.349549925717071 & -0.025126807003948 \\ 19.277194087124464 & -0.238691308631147 \\ -19.277176066273128 & 1.238684666362601 \\ -0.000022392951117 & 0.000001805052409 \\ -0.000001599342174 & 0.000000140990343 \end{bmatrix}$	$XE_teorica =$ $\begin{bmatrix} -1.010768632545966 & -0.362980769230769 \\ 2.010492837607614 & 0.240384615384615 \\ -0.024497079818300 & -19.100961538461540 \\ -0.016223231667748 & 16.423076923076923 \\ 0.000015841985724 & 0 \\ 0.064000000000000 & 0 \end{bmatrix}$
--	--

Figura 94: Matrices de coeficientes estimados y teóricos

En Matlab se tiene la herramienta de System Identification Toolbox. Como primera prueba, se selecciona como entrada la posición del primario y como salida el secundario del harmonic drive. El enfoque va a ser una simple ganancia con el método de Process Models ya que se quiere comprobar que esta corresponde con la reducción del mecanismo (1:30). En la imagen 95 se ve como la aproximación del modelo (validado como es normal con un conjunto de datos diferentes a la estimación) es de prácticamente el 100%. Se ha notado que, cuantos menos datos se seleccionan para la identificación el resultado es peor. Así pues, la ganancia resultante es de -0.033358 que cumple con lo esperado y el signo negativo es por el conocido cambio de sentido a la salida.

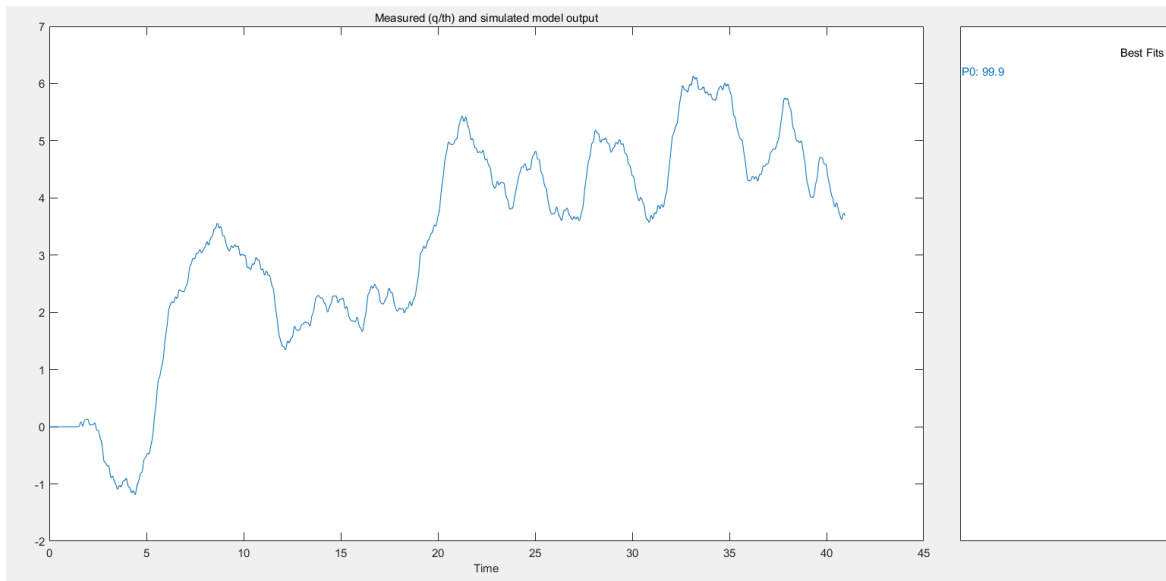


Figura 95: Modelo aproximado de la reducción del Harmonic Drive

Siguiendo con las pruebas con el Toolbox, para simplificar se escoge como única entrada la corriente y como única salida la posición del primario. A través del método Correlation Models se obtienen muy buenos resultados (figura 96) pero el modelo resulta tener 45 polos, lo cual no tiene relación alguna con el modelo esperado de 4 polos.

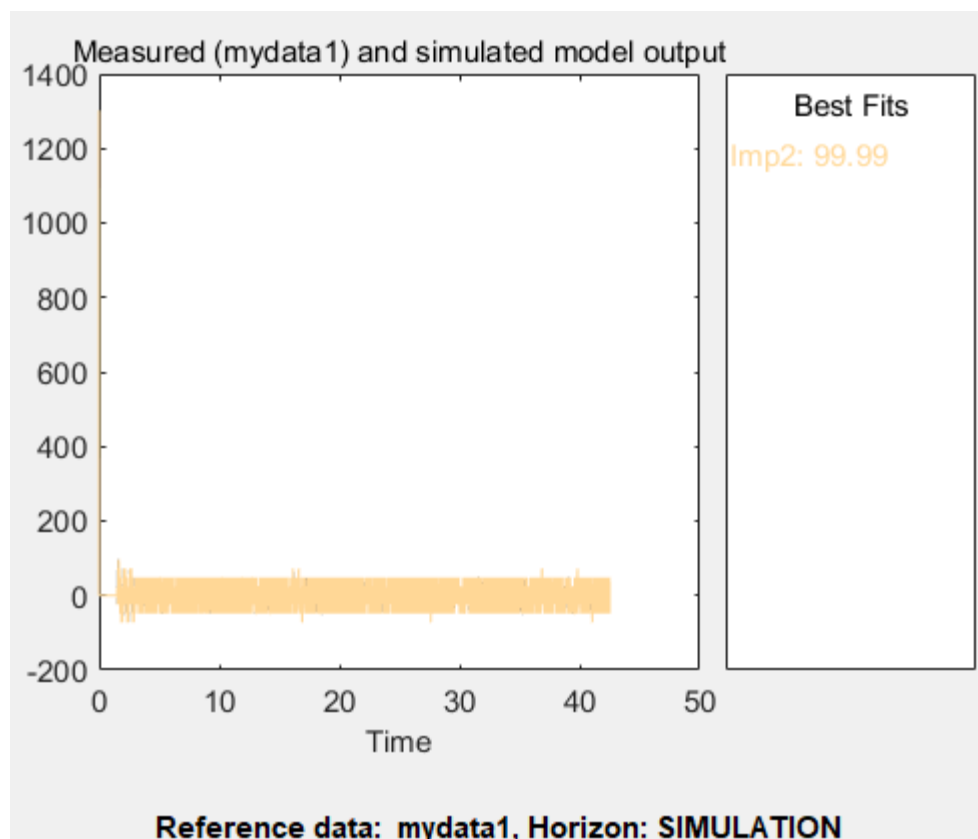


Figura 96: Aproximación de la posición del primario a través de Correlation Models

En este punto se decide cambiar el enfoque a escoger la velocidad del secundario como salida, de esta manera el modelo elimina el integrador y se supone un modelo de 3 polos y un cero. Entonces, con esta nueva estructura se procede a utilizar el método de Transfer Function Models y estimar. En la imagen 97 se puede ver como se aproxima cada modelo de prueba, teniendo un peso muy significativo la consideración de un retraso (delay). El mejor resultado es cuando se aplica un delay de 2 periodos de muestreo. Sin embargo el resultado está lejos de considerarse fiable ya que no llega al 60% de aproximación.

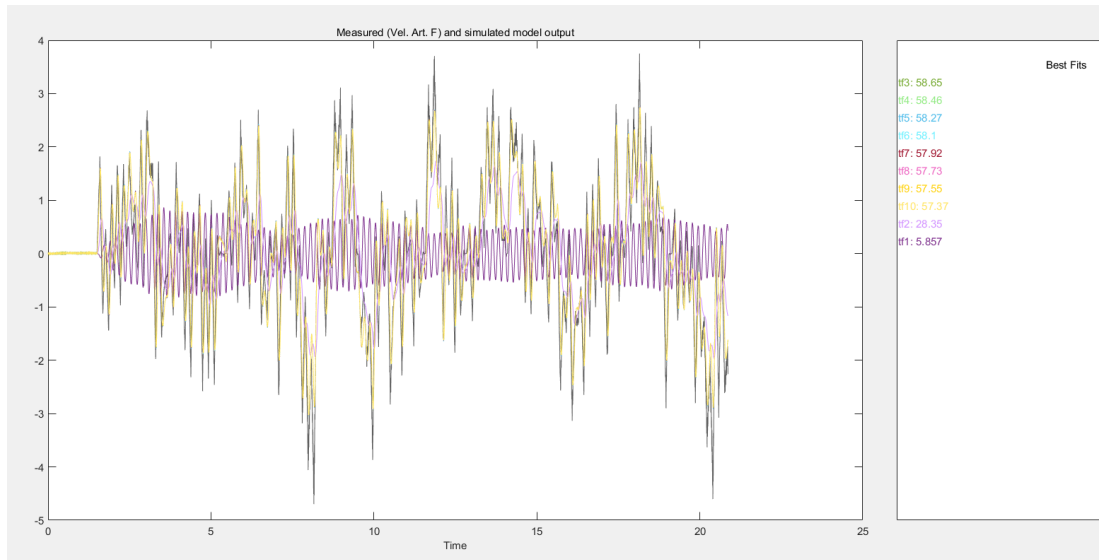


Figura 97: Aproximación a través de Transfer Function Models

Se hacen, por tanto, unas últimas pruebas en el Toolbox utilizando ahora Process Models y los resultados ya superan el 80% de aproximación como se ve en la figura 98 (sigue siendo una estimación insuficiente). Este método deja aplicar una estimación extra de un modelo de perturbación de grado uno o dos que se podría considerar para el rozamiento; de todas formas, se ha visto que esto empeora los resultados.

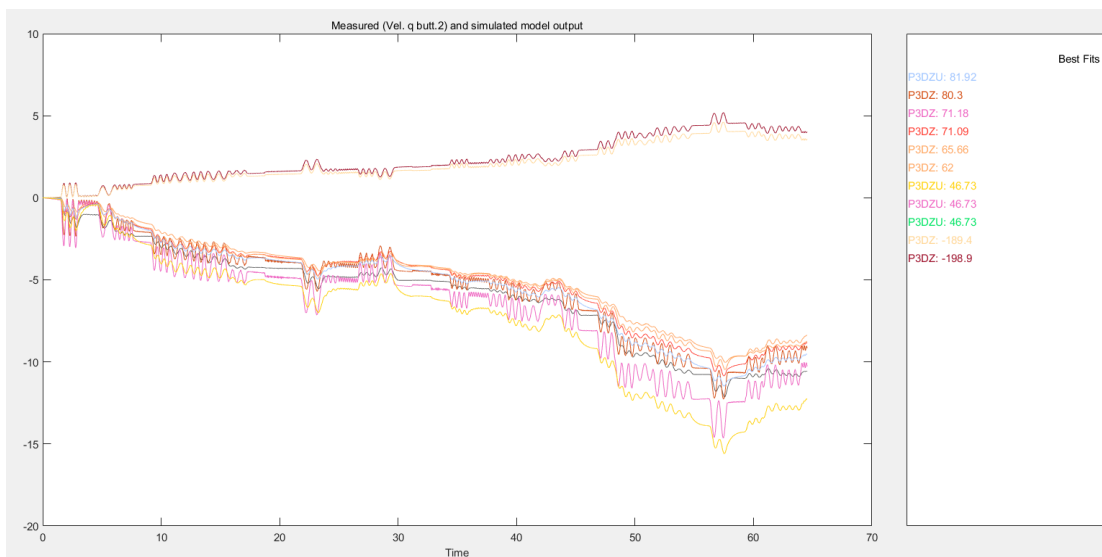


Figura 98: Aproximación a través de Process Models

El mejor modelo conseguido de esta manera se basa en una función de transferencia con un polo real y dos complejos conjugados, además de considerar un retraso nulo (figura 99).

Process model with transfer function:

$$G(s) = K_p * \frac{1+T_z*s}{(1+2*\zeta*T_w*s+(T_w*s)^2) (1+T_p3*s)} * \exp(-T_d*s)$$

$K_p = -4547.6$
 $T_w = 22.296$
 $\zeta = 11.789$
 $T_p3 = 0.00042781$
 $T_d = 0$
 $T_z = 0.22966$

Figura 99: Función de transferencia del modelo estimado de Process Models

Como última prueba de identificación, se hace un pequeño acercamiento a la identificación del sistema como modelo no lineal. Se decide utilizar, por tanto, el comando nlarx (Nonlinear Autoregressive with eXogenous inputs) en Matlab, que es una extensión de los modelos lineales ARX para sistemas dinámicos no lineales. Se utiliza para modelar sistemas en los que la salida depende no solo de sus valores pasados (autorregresivos) sino también de entradas externas y funciones no lineales de estas variables. El número de unidades en la función de la red neuronal *Wavelet* se puede especificar explícitamente en lugar de ser elegido automáticamente por el algoritmo de estimación [43].

Se aplica, por tanto, una matriz de órdenes (NN3) considerando su forma [na nb nk] que, como se tratan ahora 1 entrada (corriente) y dos salidas (posición del primario y secundario), debe tener un tamaño de 2x4 [43]. A su vez, en las dos primeras columnas se especifica que se trabaje con dos retrasos de cada salida, en la tercera columna que solo se tenga en cuenta un valor de la entrada y en último lugar el retraso de la entrada. Todo ello da como resultado una estimación interesante (figura 100) pero bastante lejos de ser satisfactoria.

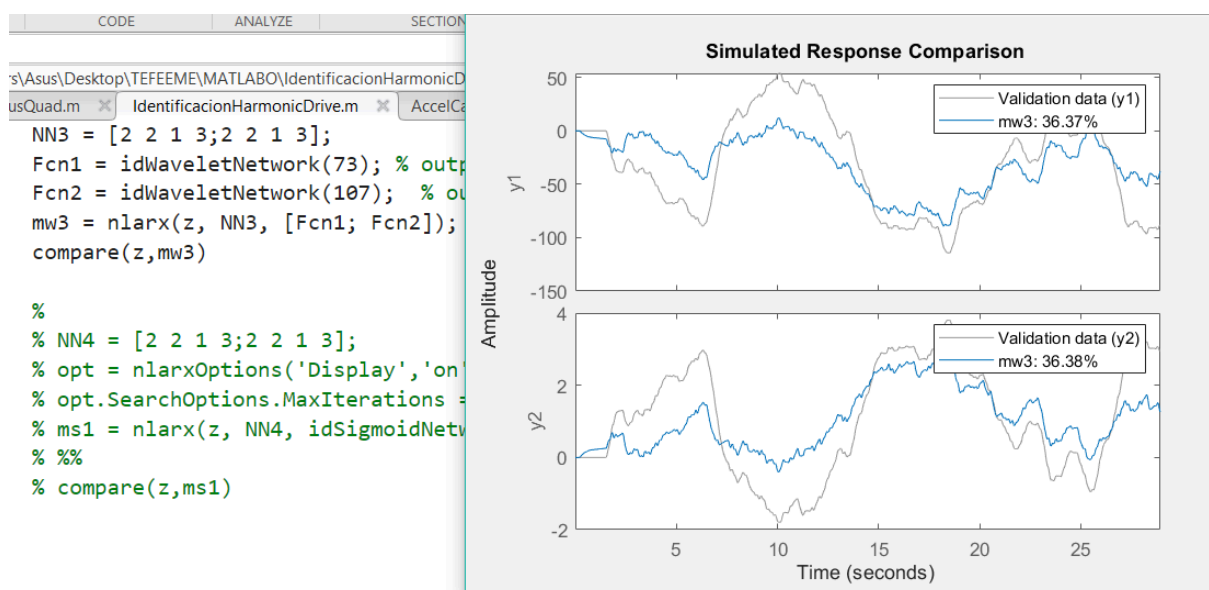


Figura 100: Implementación de modelo ARX no lineal en Matlab

Para acabar, se va a comentar un efecto de resonancia encontrado en el secundario del harmonic drive. Si bien las posiciones medidas no presentan errores sistemáticos aparentes, al estimar la velocidad se puede identificar una componente oscilante que acompaña a la curva y cuya frecuencia aumenta con la velocidad (figuras 101 y 102).

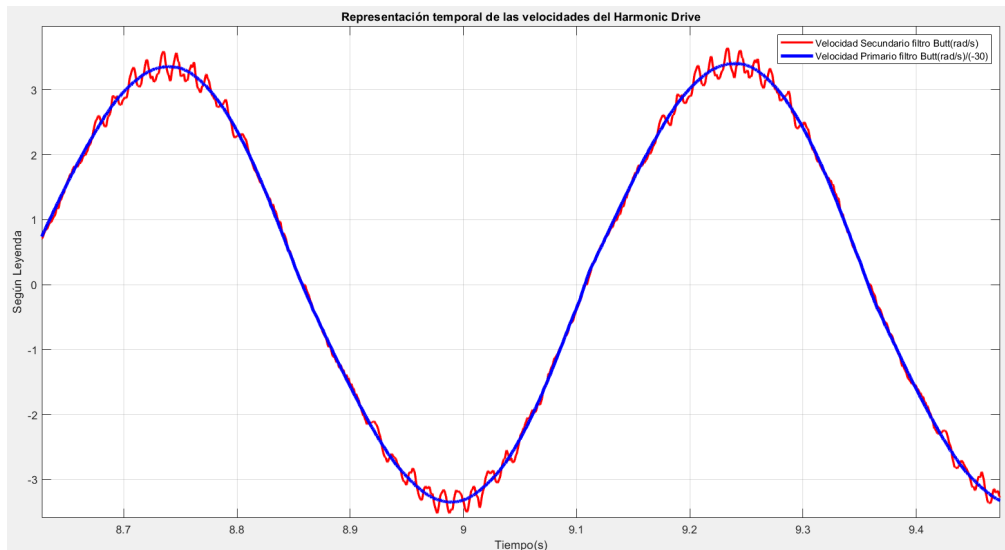


Figura 101: Velocidades y resonancia ante entrada senoidal (E1')

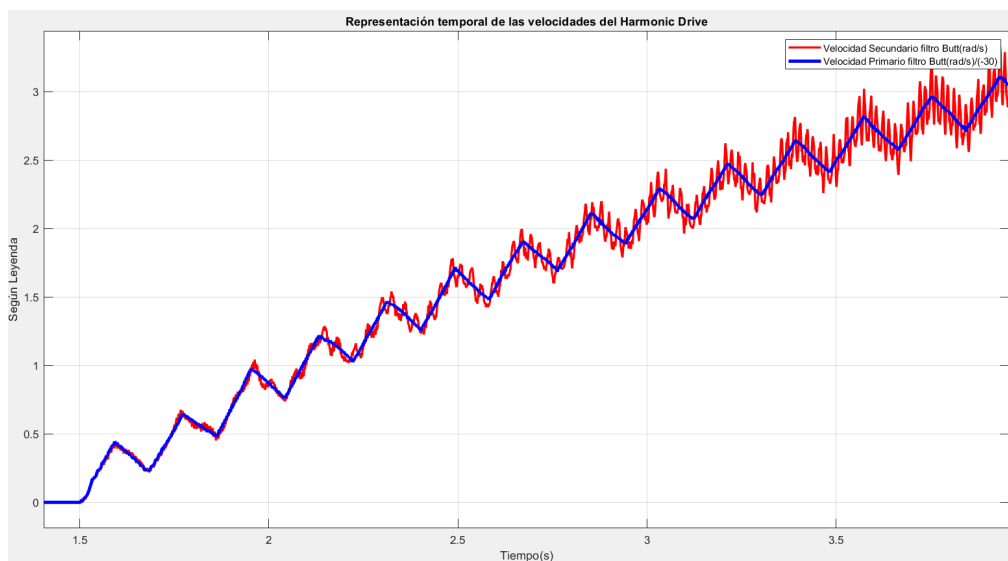


Figura 102: Velocidades y resonancia ante escalones con offset (E1')

Esta anomalía presenta una frecuencia que aumenta de forma proporcional a la velocidad de trabajo y, tras comprobarlo en diferentes puntos de las curvas e intentar buscar una correlación, esta frecuencia de resonancia es siempre 5 veces la frecuencia de giro del primario. En este punto, cabe asegurarse de que no se trata de parte del ya mencionado error de cuantización en el cálculo de la velocidad. Para ello, en la figura 103, se puede identificar que por un lado se tienen los escalonamientos característicos producidos por el ruido de cuantización y que, por otro lado, se describe una senoidal de orden diferente cuyo origen se plantea exógeno.

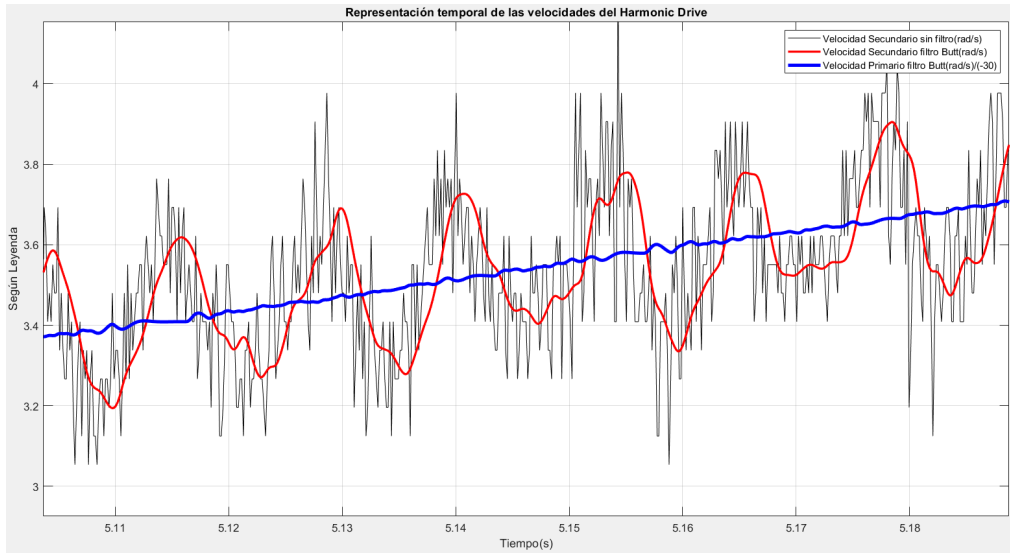


Figura 103: Velocidades y resonancia con y sin filtro (E1')

En cambio, cuando se comparan los datos de E1 y E1', se perciben grandes mejoras en la atenuación de esas oscilaciones (figuras 104 y 105). Puede que el origen de estas perturbaciones sea la mala alineación de los ejes o, que como mínimo, este sea un factor agravante.

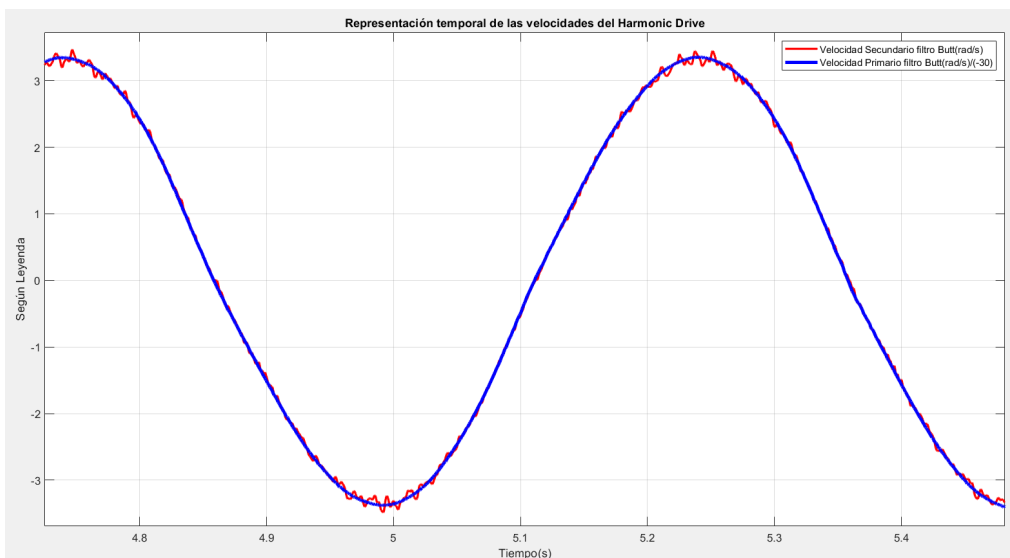


Figura 104: Velocidades y resonancia ante entrada senoidal (E1)

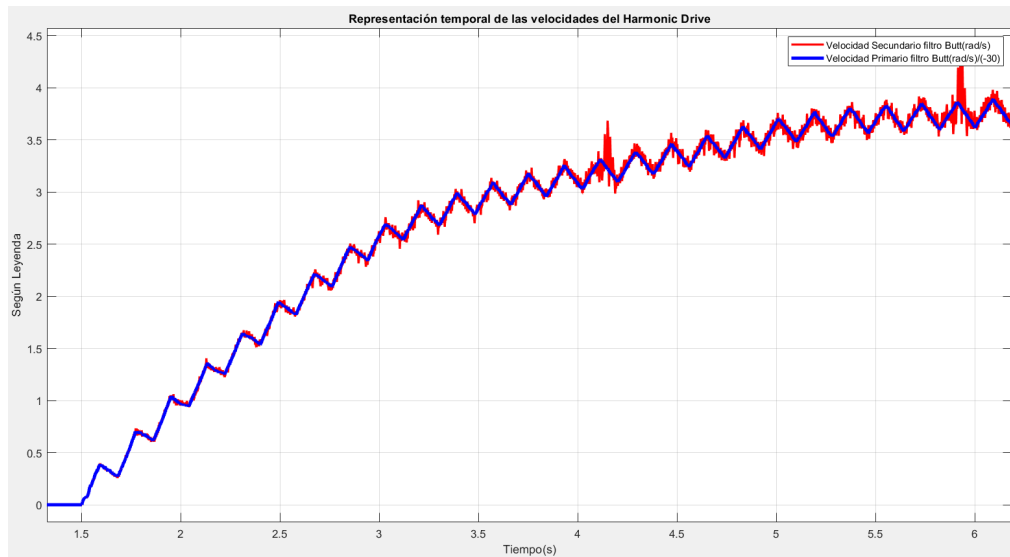


Figura 105: Velocidades y resonancia ante escalones con offset (E1)

Por tanto, el origen de las oscilaciones indeseadas en la velocidad del secundario puede deberse a la junta Oldham (figura 106) que incorpora el harmonic drive en el primario. La razón por la que no se observa la resonancia en las medidas de la entrada del reductor puede ser debido a las condiciones de muestreo del Teorema de Nyquist-Shannon. Este establece el criterio bajo el cual una señal continua en el tiempo (analógica) puede ser completamente reconstruida a partir de sus muestras discretas (digitales) [44]. El mencionado criterio se basa en trabajar con frecuencias de muestreo al menos el doble de las frecuencias máximas de las señales que se esperan medir, en caso contrario, puede darse el efecto indeseado del aliasing (distorsión en la reconstrucción de la señal original).

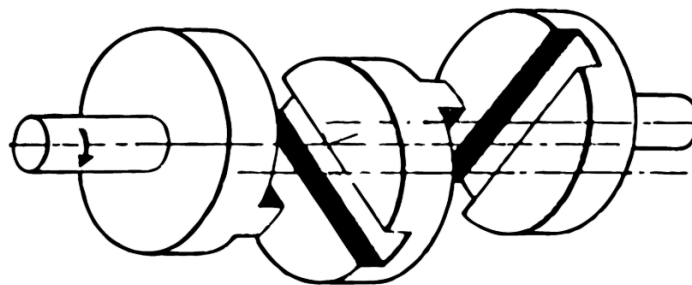


Figura 106: Modelo y fundamento de junta Oldham [13]

Entonces, si bien es cierto que la resolución del encoder del primario es 72 veces peor que en el secundario, este gira 30 veces más rápido. Por lo tanto, la relación de pulsos obtenidos por unidad de tiempo es 2.4 veces mayor en el secundario, una proporción bastante equilibrada. De todas maneras, es evidente que en la entrada no se captura con tanta precisión los fenómenos por cada vuelta del eje comparado con la salida. Si a esto último se le suma que el encoder del primario está situado más al lado del motor, antes de la junta Oldham del harmonic drive, es posible que el resultado sea la no detección de esas resonancias en la entrada.

Sin embargo, al revisar las características del encoder magnético, se encuentran tres posibles errores de precisión en la medición: error de magnetización, excentricidad y SDE (figura 107).

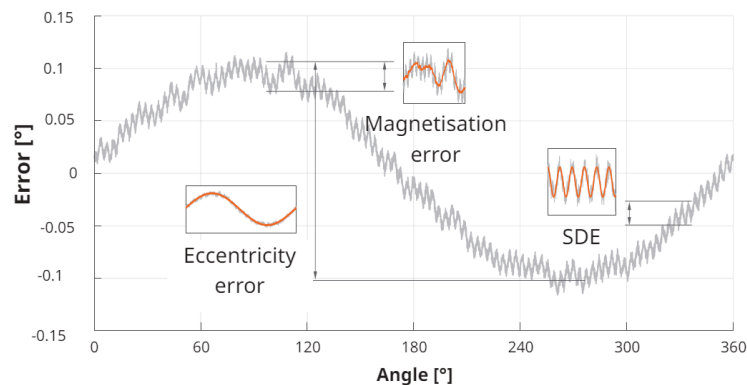


Fig. 1. Typical accuracy error plot.

Figura 107: Errores de precisión en la medición del encoder magnético [33]

En primer lugar, el error de magnetización es causado por imperfecciones en el material de elasto-ferrita y posibles desviaciones resultantes del proceso de magnetización [33]. Este error no incluye la excentricidad del montaje del anillo. Los siguientes factores influyen en el resultado:

- La inhomogeneidad magnética de la capa de elasto-ferrita.
- Las tolerancias de instalación del anillo durante el proceso de magnetización.
- La incertidumbre de medición del sistema de magnetización durante el proceso de fabricación.
- La calidad del sistema de magnetización.

En segundo lugar, la instalación y el ajuste del anillo y del cabezal de lectura, además del error específico del codificador, normalmente tienen un efecto significativo en la precisión general del sistema [33]. Tiene importancia aquí la excentricidad de instalación y el efecto de las deformaciones resultantes del montaje del anillo.

En tercer lugar, el error de subdivisión (SDE) o error de interpolación es un error de precisión periódico [33]. Está influenciado por los siguientes factores:

- La longitud de los polos.
- La homogeneidad y definición del ciclo de los polos magnéticos.
- La distancia de detección del cabezal de lectura instalado.
- La calidad del procesamiento de la señal.
- Las características del sensor magnético interno (AMR).

El SDE provoca fluctuaciones de velocidad en aplicaciones donde el codificador se usa como retroalimentación de velocidad, por ejemplo, en bucles de control de velocidad [33]. Para los anillos axiales, el SDE está fuertemente influenciado por la altura de conducción.

Pese a que en el control del motor se aplica una regulación de corriente y no de velocidad, es evidente que al aproximar la velocidad en el proceso de identificación se pueden presentar igualmente los efectos del SDE que son amplificados por la excentricidad que tenga el montaje experimental.

Finalmente, la documentación del encoder magnético también expone un error adicional por cada vuelta del disco, la diafonía o *crosstalk*, que es un efecto indeseable de la magnetización de la marca de referencia sobre la magnetización de la pista incremental, lo que conduce a picos de imprecisión (figura 108). Depende tanto de la altura de conducción como del desplazamiento lateral [33].

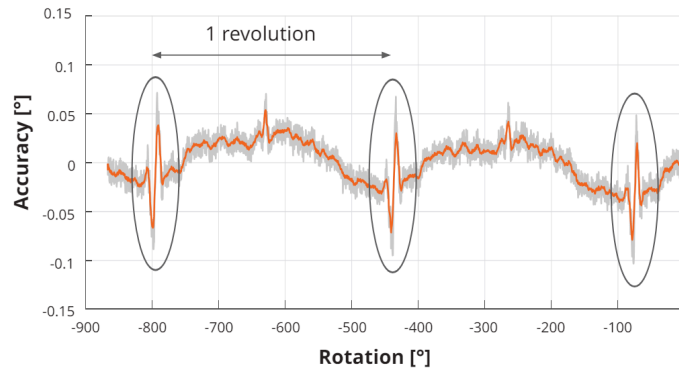


Fig. 2. Crosstalk representation. The crosstalk (Ri magnetisation) is circled.

Figura 108: Errores de precisión en la medición del encoder magnético [33]

Estos picos se pueden percibir fácilmente en los datos obtenidos cuando los ejes giran constantemente en un sentido (señales con offset) y, de esta manera, el encoder magnético da varias vueltas que generan varios errores de diafonía. En la figura 109, se ve este efecto de picos en la estimación de la velocidad del encoder magnético. Se comprueba, por tanto, que el periodo de tiempo entre los picos (1.35 segundos aproximadamente) que, haciendo el producto con la velocidad promedio de giro (4.658 rad/s) se obtienen 6.288 radianes, es decir que, efectivamente se corresponde con una vuelta completa de la salida (2π radianes).

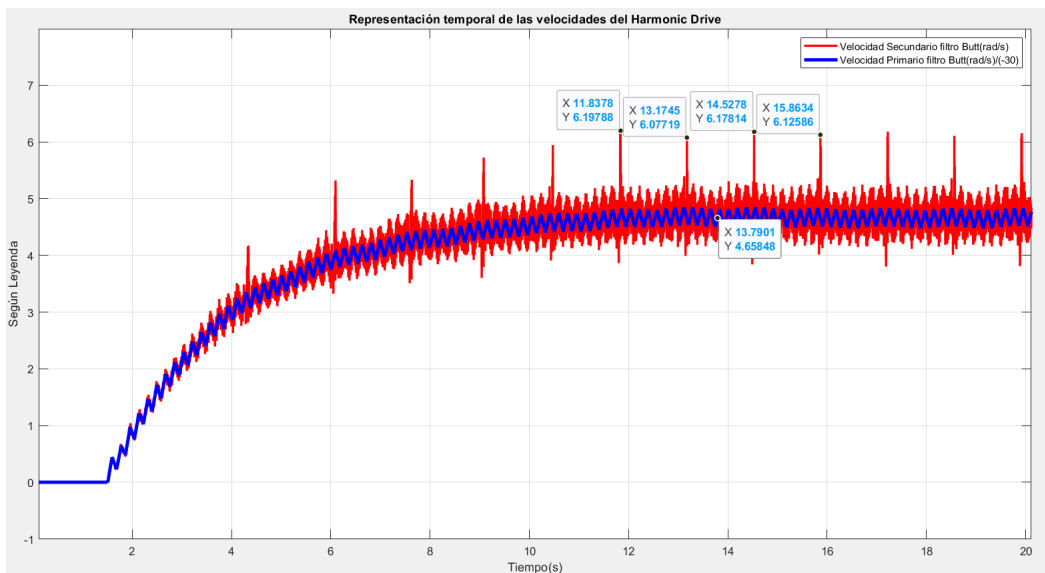


Figura 109: Velocidades con resonancia y diafonía ante escalones con offset (E1)

7. Conclusiones

A modo de cierre, se exponen una serie de conclusiones sobre la evolución del diseño de los experimentos, de manera que puede afirmarse, en resumen, que se ha alcanzado el objetivo general (OG), ya que se ha desarrollado una base de experimentos precisa que permite la obtención de numerosos datos para la identificación de fenómenos y parámetros del Harmonic Drive.

En cuanto a los objetivos específicos (OE), en primer lugar, se ha montado una mesa de experimentación lo más precisa y estable posible para realizar pruebas con el reductor Harmonic Drive HFUC-14-2UH, como se ha visto en el apartado del subsistema mecánico. Es decir, se ha logrado el OE1.

En segundo lugar, respondiendo al OE2, se ha diseñado un subsistema electrónico robusto para el control y adquisición de datos dando uso también de sensores de alta precisión.

En tercer lugar, se ha desarrollado un subsistema de programación optimizado que, junto con la electrónica, consigue una frecuencia de muestreo y resolución superior a la presente en otros proyectos y artículos de esta área de estudio, situándose en 150 μ s y 589824 cuentas por revolución. De esta forma, se cumple con el OE3.

En cuanto al OE4, se han realizado ensayos aplicando par desde la entrada del reductor a través de un control por corriente en vacío y con el servomotor como carga, todo ello mediante los experimentos E1, E1' y E3'.

Por su parte, el OE5, se ha puesto a prueba la back drivability del Harmonic Drive aplicando señales de par desde la salida con un servomotor de potencia gracias al experimento E2'.

Por último, también el OE6 ha quedado cumplido, ya que se ha experimentado con una versión simplificada del propio Harmonic Drive, realizando un despiece de este y trabajando solamente con el wave generator y flexspline para simplificar el sistema a identificar en E4.

Sin embargo, los objetivos OE7 y OE8 no han sido alcanzados con plena satisfacción. Por un lado, se ha descrito el modelado teórico (de forma polinómica, espacio de estados y Laplace) con los parámetros que definen una articulación elástica como el Harmonic Drive, pero no se ha podido indagar mucho más en este aspecto ni tampoco se han desarrollado todos los modelos de rozamiento que conciernen a este tipo de reductores. Por otro lado, se han interpretado varios datos adquiridos y se han identificado ciertos parámetros y fenómenos del sistema como el rozamiento estático, la reducción de velocidad y fluctuaciones en la salida. No obstante, por limitaciones de tiempo y extensión del proyecto solo se han analizado los experimentos E1 y E1' y no se ha profundizado lo suficiente en las estrategias de identificación para la obtención precisa de más parámetros o de un modelo del Harmonic Drive fiable.

Además de los objetivos principales del proyecto, su desarrollo ha permitido alcanzar diversos fines de aprendizaje, todos valiosos como la culminación del máster universitario en Ingeniería Mecatrónica. Entre estos logros se encuentran la experiencia de trabajar con distintos componentes electrónicos, enfrentar diversos retos en el entorno mecánico, fortalecer las habilidades de programación y adquirir más conocimientos sobre el modelado e identificación de sistemas. Todas estas experiencias en el desarrollo de experimentos de alta precisión han proporcionado, sin duda, una visión práctica y cercana de cómo abordar un desafío de ingeniería, a pesar de las numerosas limitaciones.

Dada la actual extensión del proyecto y el tiempo con el que se ha contado para su desarrollo, no se han podido explotar ciertas posibilidades. Por un lado, se han adquirido los datos por transmisión serie directamente por el mal funcionamiento de la memoria SDRAM del microcontrolador STM32F429. Por otro lado, existe la limitación de no poder medir el valor de la corriente real en el experimento E3' ya que el controlador de motores Digitax HD no incorpora una salida analógica. Además, en cuanto al subsistema control, este no ha tenido un desarrollo merecido ya que, como ya se ha estado comentando, el peso del proyecto se encuentra en el propio desarrollo de los experimentos.

Por consiguiente, una vez expuestas las limitaciones más relevantes del proyecto, queda abierta la posibilidad de futuras líneas de trabajo. La primera es que, si se busca aumentar la frecuencia de muestreo, se baraje la posibilidad de trabajar con un módulo de adquisición de datos nuevo, con los periféricos intactos y más veloz si cabe.

La segunda propuesta conveniente para seguir mejorando es adquirir el módulo adicional del controlador Digitax para entradas y salidas analógicas y poder muestrear la corriente real del servomotor de potencia cuando este actúa sobre la salida del Harmonic Drive.

Para finalizar, como se ha podido intuir en el transcurso de la memoria, este proyecto puede llegar a ser un provechoso material de ayuda a la indagación en esta línea de trabajos. Las futuras líneas de investigación y desarrollo para la identificación y modelado preciso de este gran sistema mecánico que es el reductor Harmonic Drive, pueden llegar a ver un gran beneficio en la ejecución de los experimentos de este proyecto, así como en los resultados obtenidos.

8. Bibliografía

- [1] «C. W. Musser, the Father of the HarmonicDrive® | Harmonic Drive Systems», ハーモニック・ドライブ・システムズ. Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.hds.co.jp/english/development/hd_skill/c_w_musser/
- [2] «Unidad HFUC-2UH», Harmonic Drive SE. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://harmonicdrive.de/es/producto/reductor-con-rodamiento-exterior/hfuc-2uh>
- [3] Roydisa, «Elementos principales a la hora de elegir Reductores Mecánicos.», Roydisa. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.roydisa.es/archivos/3356>
- [4] «Harmonic Drive | Description, Advantages, & Facts | Britannica». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.britannica.com/technology/Harmonic-Drive>
- [5] «Introducción a la identificación de sistemas», TÉCNICA INDUSTRIAL. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.tecnicaindustrial.es/introduccion-a-la-identificacion-de-sistemas/>[6]R. García, «Modelado, identificación y simulación de articulaciones elásticas», ene. 2019, Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://riunet.upv.es/handle/10251/115461>
- [6] R. S. Celis García (2019). Modelado, identificación y simulación de articulaciones elásticas: «<https://riunet.upv.es/bitstream/handle/10251/115461/Modeling%20identification%20and%20simulation%20of%20elastic%20joint.pdf>»
- [7] T. D. Tuttle, «Understanding and Modeling the Behavior of a Harmonic Drive Gear Transmission».
- [8] M. Ruderman, F. Hoffmann, y T. Bertram, «Modeling and Identification of Elastic Robot Joints With Hysteresis and Backlash», *IEEE Trans. Ind. Electron.*, vol. 56, n.º 10, pp. 3840-3847, oct. 2009, doi: 10.1109/TIE.2009.2015752.
- [9] M. Ruderman, T. Bertram, y M. Iwasaki, «Modeling, observation, and control of hysteresis torsion in elastic robot joints», *Mechatronics*, vol. 24, n.º 5, pp. 407-415, ago. 2014, doi: 10.1016/j.mechatronics.2014.02.009.
- [10] M. Ruderman y M. Iwasaki, «Control of Nonlinear Elastic Joint Robots using Feed-forward Torque Decoupling», *IFAC-Pap.*, vol. 48, n.º 11, pp. 137-142, ene. 2015, doi: 10.1016/j.ifacol.2015.09.173.
- [11] M. Ruderman, «On break-away forces in actuated motion systems with nonlinear friction», *Mechatronics*, vol. 44, pp. 1-5, jun. 2017, doi: 10.1016/j.mechatronics.2017.03.007.
- [12] C. Preissner, T. J. Royston, y D. Shu, «A High-Fidelity Harmonic Drive Model», *J. Dyn. Syst. Meas. Control*, vol. 134, n.º 011002, dic. 2011, doi: 10.1115/1.4005041.
- [13] «2014_12_ED_1019644_HFUC_2UH.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://harmonicdrive.de/fileadmin/user_upload/2014_12_ED_1019644_HFUC_2UH.pdf

- [14] «norem ES | Elementos estándar para la ingeniería mecánica». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://norem.es/es/>
- [15] «THE BIG GREEN BOOK - Pedir catálogo de productos ahora | norem ES». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://norem.es/es/pedido-del-catalogo>
- [16] «¿Qué es FDM? Tecnología de Fused Deposition Modeling para impresión 3D». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.stratasys.com/es/guide-to-3d-printing/technologies-and-materials/fdm-technology/>
- [17] «Descripción general de la sinterización selectiva por láser (SLS) | 3D Systems». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://es.3dsystems.com/selective-laser-sintering>
- [18] C. H. Fernández, «¿Qué es el nailon o nlyon? Usos y formatos», Servei Estació. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://serveiestacio.com/blog/que-es-el-nailon-o-nylon/>
- [19] «¿Qué es la junta cardán?», HR Motor. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.hrmotor.com/que-es/junta-cardan/>
- [20] «MECAPEDIA • Acoplamiento de Oldham». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.mecapedia.uji.es/pages/acoplamiento_de_Oldham.html
- [21] «How to use STM32CubeMX to configure HSE (High-Speed External Oscillator) and LSE (Low-Speed External Oscillator)». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://community.st.com/t5/stm32-mcus/how-to-use-stm32cubemx-to-configure-hse-high-speed-external/ta-p/49604>
- [22] «ESCON Module 50/5 Hardware Reference», 2015.
- [23] «10 Piezas CD4050 Compuertas lógicas DIP-14 : Amazon.es: Informática». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.amazon.es/Piezas-CD4050-Compuertas-l%C3%B3gicas-DIP-14/dp/B0C7491TNP?pd_rd_w=Lqaps&content-id=amzn1.sym.f818f2fc-e249-43fe-8b7a-354660a73fc6&pf_rd_p=f818f2fc-e249-43fe-8b7a-354660a73fc6&pf_rd_r=2Z1VPM0T3JY8A436FXTR&pd_rd_wg=qAH63&pd_rd_r=cd28936d-ce30-4a6a-b640-e59193b1b05e&pd_rd_i=B0C7491TNP&pvc=1&ref_=pd_bap_d_grid_rp_0_1_ec_pd_nav_hcs_rp_3_t
- [24] *Drive LEDs Without Resistors - Hex Buffer CD4050B Has A Trick Up Its Sleeve - Simply Put*, (3 de abril de 2020). Accedido: 3 de julio de 2024. [En línea Video]. Disponible en: https://www.youtube.com/watch?v=_spbp3Kyfd4
- [25] «409510-ESCON-50-5-Referencia-del-Dispositivo-Es.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.maxongroup.de/medias/sys_master/root/8834332327966/409510-ESCON-50-5-Referencia-del-Dispositivo-Es.pdf
- [26] «How to set up the FMC peripheral to interface with the SDRAM IS42S16800F-6BLI from ISSI». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://community.st.com/t5/stm32-mcus/how-to-set-up-the-fmc-peripheral-to-interface-with-the-sdram/ta-p/49457>
- [27] «ASCII y UTF-8», Apuntes de Programador. Accedido: 3 de julio de 2024. [En línea].

Disponible en: <https://apuntes.de/golang/ascii-y-utf8/>

- [28] «Bits de inicio, parada y marca - Documentación de IBM». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.ibm.com/docs/es/aix/7.1?topic=parameters-start-stop-mark-bits>
- [29] «El código ASCII Completo, tabla con los codigos ASCII completos, caracteres simbolos letras ascii, ascii codigo, tabla ascii, codigos ascii, caracteres ascii, codigos, tabla, caracteres, simbolos, control, imprimibles, extendido, letras, vocales, signos, simbolos, mayusculas, minusculas, alt, teclas, acentos, agudo, grave, eñe, enie, arroba, dieresis, circunflejo, tilde, cedilla, anillo, libra, esterlina, centavo, teclado, tipear, escribir, español, ingles, notebook, laptop, asccii, asqui, askii, aski,20240703». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://elcodigoascii.com.ar/>
- [30] M. T. Ortiz, *7.1 Números pseudoaleatorios | Estadística Computacional*. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://tereom.github.io/est-computacional-2018/numeros-pseudoaleatorios.html>
- [31] «um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf
- [32] Reference Manual STM32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439: «dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf
- [33] «Miniature Incremental Magnetic Encoder Module | RLC2HD encoder - www.rls.si». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.rls.si/eng/rlc2hd-miniature-linear-and-rotary-pcb-level-incremental-magnetic-encoder?partNumbers=RLC2HDA13BK00C00>
- [34] «Axial magnetic rings- www.rls.si». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.rls.si/eng/axial-magnetic-rings>
- [35] «HEDS-9040#J00 - Codificador Incremental Óptico, Serie HEDS-9040, 3 Canales, 1024 CPR, Detección de Posición Rotativa». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://es.farnell.com/broadcom-limited/heds-9040-j00/encoder-rotary-1024ppr-3ch/dp/1654869>
- [36] «Maxon EC90 Flat 90watt With Encoder | Descargar gratis PDF | Electrical Components | Electronic Engineering», Scribd. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.scribd.com/document/30713567/Maxon-EC90-Flat-90Watt-With-Encoder>
- [37] «Nidec Control Techniques Unimotor HD». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://variablefrequencydrive.com/control-techniques/servo-motors/unimotor-hd/>
- [38] «Digitax HD M751 Base». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://acim.nidec.com/es-ES/drives/control-techniques/Products/Servo-Drives/Digitax-HD/Digitax-HD-M751-Base>

- [39] «3737-servoregelaars-digitax-hd-m751-control-user-guide-en-iss3-0478-0460-03.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.nidec-netherlands.nl/media/3737-servoregelaars-digitax-hd-m751-control-user-guide-en-iss3-0478-0460-03.pdf>
- [40] «LM358 Dual Opamp Pinout». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://ecstudiosystems.com/discover/circuits/pinouts/lm358-pinout/>
- [41] bilaljo, «Prescaler vs Counter Period (STM32)», Electrical Engineering Stack Exchange. Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://electronics.stackexchange.com/q/632838>
- [42] «Memoria%2F05+-+Modelado+de+la+fricción.pdf». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://biblus.us.es/bibing/proyectos/abreproy/4022/fichero/Memoria%252F05+-+Modelado+de+la+fricci%C3%B3n.pdf>
- [43] «Motorized Camera - Multi-Input Multi-Output Nonlinear ARX and Hammerstein-Wiener Models - MATLAB & Simulink Example - MathWorks España». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://es.mathworks.com/help/ident/ug/motorized-camera-multi-input-multi-output-nonlinear-arx-and-hammerstein-wiener-models.html>
- [44] «Adquirir una Señal Analógica: Ancho de Banda, Teorema de Muestreo de Nyquist y Aliasing». Accedido: 3 de julio de 2024. [En línea]. Disponible en: <https://www.ni.com/es/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>

9. Anexos

ANEXO I. RELACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X

ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.

ODS 3. La salud y bienestar tienen una relación notoria con el cometido de este proyecto. Mejorar el modelo de los harmonic drives conlleva también, una mejora de las prestaciones de los dispositivos que los usan, como pueden ser robots que ayudan a la movilidad de personas con diversidad funcional. Además, es importante destacar una de las características más estudiadas en este trabajo, la reversibilidad (*back drivability*) la cual es fundamental en prótesis robóticas para conseguir una buena interfaz humano/máquina.

ODS 8. Con un impacto considerable, el objetivo de trabajo decente y crecimiento económico se ve afectado por las mejoras de los modelos teóricos. Perfeccionando la precisión de los sistemas simulados se pueden alcanzar resultados más precisos y eficientes en las aplicaciones reales sin la necesidad de realizar cambios en los componentes físicos.

ODS 9. El objetivo más relacionado con el proyecto es el de industria, innovación e infraestructuras. Estos sectores de la ingeniería se pueden ver altamente beneficiados por cada mejora que se pueda conseguir en cuanto a las prestaciones de las máquinas o robots utilizados. Una mejor identificación y una mejora del modelado pueden dar lugar a tareas más complejas y ofrecer mayor fiabilidad en su control. Además, la identificación de la reversibilidad puede ser un buen incentivo para aumentar el uso de robots colaborativos en la industria.

ANEXO II. CÓDIGO PRINCIPAL STM32CUBE IDE

```
/* USER CODE BEGIN Header */
/**
 * *****
 * @file      : main.c
 * @brief     : Main program body
 * *****
 * @attention
 *
 * Copyright (c) 2024 STMicroelectronics.
 * All rights reserved.
 *
 * This software is licensed under terms that can be found in the LICENSE file
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is provided AS-IS.
 *
 * *****
 */
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdbool.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
#define MODO_ESCALERA    1
#define MODO_ESCALON    2
#define MODO_RAMPA      3
#define MODO_SENOIDAL   4
#define MODO_RANDOM_E   5
#define MODO_RANDOM_R   6
#define MODO_RANDOM_S   7
#define MODO_ARRANQUE    8
#define MODO_HISTERESIS 9
#define MODO_BARRIDO     10
#define MODO_MIX         11
#define MODO_RANDOM_M    12
```

```

/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
ADC_HandleTypeDef hadc2;

DAC_HandleTypeDef hdac;

RNG_HandleTypeDef hrng;

TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim4;
TIM_HandleTypeDef htim8;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
uint16_t time=0;

char data[] = "748##0748";
//Compresión de los dos primeros dígitos de cada número
//#-->00
//7-->20

//ADC
uint16_t readCurrent = 4096/2, readADC = 4096/2;

//ACCIÓN DE CONTROL DAC
uint16_t accion = 4096/2, duty = 4096/2, max_duty = 4095 - 360, min_duty = 360, middle_duty =
4096/2, aux = 4096/2;
uint16_t A = 4096/2 - 360, Ar1 = (4096/2 - 360)/2, Ar2 = (4096/2 - 360)/4, Ar3 = (4096/2 - 360)/4;
uint16_t Tr = 500, Tr1 = 500, Tr2 = 500, Tr3 = 500, alter = 0;
uint32_t i = 0, t = 0, tabs = 0, T = 500, cp = 1, randomN = 360, macroetapa = 1, incremento = 0;
bool etapa = 1;

//ENCODER ÓPTICO
uint16_t counter1 = 0;

//ENCODER MAGNÉTICO
uint16_t counter2 = 0;

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM2_Init(void);
static void MX_USART1_UART_Init(void);

```



```

static void MX_TIM8_Init(void);
static void MX_TIM4_Init(void);
static void MX_ADC2_Init(void);
static void MX_RNG_Init(void);
static void MX_DAC_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */
void controlCalc(uint8_t modo, uint32_t CP, int16_t offset, int16_t G);

void compresionASCII(uint16_t number1, uint16_t number2, uint16_t number3, uint16_t number4,
char *ascii);

int _write(int file, char *ptr, int len)
{
    HAL_UART_Transmit(&huart1, (uint8_t*)ptr, len, HAL_MAX_DELAY);
    return len;
}

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc){
    //60 micros
    readADC = HAL_ADC_GetValue(&hadc2);
}

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

```

```

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_TIM2_Init();
MX_USART1_UART_Init();
MX_TIM8_Init();
MX_TIM4_Init();
MX_ADC2_Init();
MX_RNG_Init();
MX_DAC_Init();
/* USER CODE BEGIN 2 */

HAL_TIM_Encoder_Start_IT(&htim4,TIM_CHANNEL_ALL);
HAL_TIM_Encoder_Start_IT(&htim2,TIM_CHANNEL_ALL);
HAL_DAC_Start(&hdac, DAC_CHANNEL_1);

HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, accion);

__HAL_TIM_SetCounter(&htim4, 0);
__HAL_TIM_SetCounter(&htim2, 0);

HAL_Delay(5000);
HAL_TIM_Base_Start_IT(&htim8);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage

```

```

*/
__HAL_RCC_PWR_CLK_ENABLE();
__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

/** Initializes the RCC Oscillators according to the specified parameters
 * in the RCC_OscInitTypeDef structure.
 */
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLM = 4;
RCC_OscInitStruct.PLL.PLLN = 180;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 8;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Activate the Over-Drive mode
 */
if (HAL_PWREx_EnableOverDrive() != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
 */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC2_Init(void)
{
    /* USER CODE BEGIN ADC2_Init 0 */

```

```

/* USER CODE END ADC2_Init 0 */

ADC_ChannelConfTypeDef sConfig = {0};

/* USER CODE BEGIN ADC2_Init 1 */

/* USER CODE END ADC2_Init 1 */

/** Configure the global features of the ADC (Clock, Resolution, Data Alignment and number of
conversion)
*/
hadc2.Instance = ADC2;
hadc2.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
hadc2.Init.Resolution = ADC_RESOLUTION_12B;
hadc2.Init.ScanConvMode = DISABLE;
hadc2.Init.ContinuousConvMode = DISABLE;
hadc2.Init.DiscontinuousConvMode = DISABLE;
hadc2.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
hadc2.Init.ExternalTrigConv = ADC_SOFTWARE_START;
hadc2.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc2.Init.NbrOfConversion = 1;
hadc2.Init.DMAContinuousRequests = DISABLE;
hadc2.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
if (HAL_ADC_Init(&hadc2) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank in the sequencer and its
sample time.
*/
sConfig.Channel = ADC_CHANNEL_15;
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN ADC2_Init 2 */

/* USER CODE END ADC2_Init 2 */

}

/**
 * @brief DAC Initialization Function
 * @param None
 * @retval None
 */
static void MX_DAC_Init(void)
{

```

```

/* USER CODE BEGIN DAC_Init 0 */

/* USER CODE END DAC_Init 0 */

DAC_ChannelConfTypeDef sConfig = {0};

/* USER CODE BEGIN DAC_Init 1 */

/* USER CODE END DAC_Init 1 */

/** DAC Initialization
*/
hdac.Instance = DAC;
if (HAL_DAC_Init(&hdac) != HAL_OK)
{
    Error_Handler();
}

/** DAC channel OUT1 config
*/
sConfig.DAC_Trigger = DAC_TRIGGER_NONE;
sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;
if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN DAC_Init 2 */

/* USER CODE END DAC_Init 2 */

}

/**
 * @brief RNG Initialization Function
 * @param None
 * @retval None
 */
static void MX_RNG_Init(void)
{

/* USER CODE BEGIN RNG_Init 0 */

/* USER CODE END RNG_Init 0 */

/* USER CODE BEGIN RNG_Init 1 */

/* USER CODE END RNG_Init 1 */
hrng.Instance = RNG;
if (HAL_RNG_Init(&hrng) != HAL_OK)
{
    Error_Handler();
}
}

```

```

/* USER CODE BEGIN RNG_Init 2 */

/* USER CODE END RNG_Init 2 */

}

/**
 * @brief TIM2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM2_Init(void)
{

/* USER CODE BEGIN TIM2_Init 0 */

/* USER CODE END TIM2_Init 0 */

TIM_Encoder_InitTypeDef sConfig = {0};
TIM_MasterConfigTypeDef sMasterConfig = {0};

/* USER CODE BEGIN TIM2_Init 1 */

/* USER CODE END TIM2_Init 1 */
htim2.Instance = TIM2;
htim2.Init.Prescaler = 0;
htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
htim2.Init.Period = 909;
htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
sConfig.IC1Polarity = TIM_ICPOLARITY_RISING;
sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
sConfig.IC1Filter = 0;
sConfig.IC2Polarity = TIM_ICPOLARITY_RISING;
sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
sConfig.IC2Prescaler = TIM_ICPSC_DIV1;
sConfig.IC2Filter = 0;
if (HAL_TIM_Encoder_Init(&htim2, &sConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM2_Init 2 */

/* USER CODE END TIM2_Init 2 */

```

```

}

/**
 * @brief TIM4 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM4_Init(void)
{
    /* USER CODE BEGIN TIM4_Init 0 */

    /* USER CODE END TIM4_Init 0 */

    TIM_Encoder_InitTypeDef sConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM4_Init 1 */

    /* USER CODE END TIM4_Init 1 */
    htim4.Instance = TIM4;
    htim4.Init.Prescaler = 0;
    htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim4.Init.Period = 90;
    htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim4.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    sConfig.EncoderMode = TIM_ENCODERMODE_TI12;
    sConfig.IC1Polarity = TIM_ICPOLARITY_RISING;
    sConfig.IC1Selection = TIM_ICSELECTION_DIRECTTI;
    sConfig.IC1Prescaler = TIM_ICPSC_DIV1;
    sConfig.IC1Filter = 0;
    sConfig.IC2Polarity = TIM_ICPOLARITY_RISING;
    sConfig.IC2Selection = TIM_ICSELECTION_DIRECTTI;
    sConfig.IC2Prescaler = TIM_ICPSC_DIV1;
    sConfig.IC2Filter = 0;
    if (HAL_TIM_Encoder_Init(&htim4, &sConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM4_Init 2 */

    /* USER CODE END TIM4_Init 2 */

}

```

```

/**
 * @brief TIM8 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM8_Init(void)
{

    /* USER CODE BEGIN TIM8_Init 0 */

    /* USER CODE END TIM8_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM8_Init 1 */

    /* USER CODE END TIM8_Init 1 */
    htim8.Instance = TIM8;
    htim8.Init.Prescaler = 180-1;
    htim8.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim8.Init.Period = 150-1;
    htim8.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim8.Init.RepetitionCounter = 0;
    htim8.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim8) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim8, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim8, &sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN TIM8_Init 2 */

    /* USER CODE END TIM8_Init 2 */

}

/**
 * @brief USART1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART1_UART_Init(void)

```



```

{

/* USER CODE BEGIN USART1_Init 0 */

/* USER CODE END USART1_Init 0 */

/* USER CODE BEGIN USART1_Init 1 */

/* USER CODE END USART1_Init 1 */
huart1.Instance = USART1;
huart1.Init.BaudRate = 2000000;
huart1.Init.WordLength = UART_WORDLENGTH_8B;
huart1.Init.StopBits = UART_STOPBITS_1;
huart1.Init.Parity = UART_PARITY_NONE;
huart1.Init.Mode = UART_MODE_TX;
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart1.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART1_Init 2 */

/* USER CODE END USART1_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

/* GPIO Ports Clock Enable */
__HAL_RCC_GPIOH_CLK_ENABLE();
__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOC_CLK_ENABLE();
__HAL_RCC_GPIOD_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}

/* USER CODE BEGIN 4 */

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    if (htim->Instance == TIM8) {

```

```

    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, accion);
    //1 micros
    HAL_ADC_Start_IT(&hadc2);
    //Va a tardar unos 60 us en completar la conversión (en segundo plano)
    counter1 = __HAL_TIM_GET_COUNTER(&htim4);
    counter2 = __HAL_TIM_GET_COUNTER(&htim2);
    readCurrent = readADC;//El valor de la corriente está retrasado un periodo
    //3 micros
    printf("%s ", data);
    //11 micros
    fflush(stdout);//fflush es necesario
    //64 micros
    //En esta versión se aumentan los bytes de corriente y duty
    //ya que puede dar cambios muy bruscos, se va a enviar un número de 12 bits
    compresionASCII(accion, counter1, counter2, readCurrent, data);
    //67 micros
    controlCalc(MODO_ESCALON, 10000, 0, 1);// arg.2: 20000 --> 20000*150us = 3s
    //tercer arg. OFFSET (ejemplos -->220, 275, 300, 950, 1200, 3736(MAX)) E4(140)
    //cuarto arg. Atenuación (ejemplo: 8 es ocho veces más pequeño) E4(25)
    //THE END: el código temporizado se ejecuta en 140 micros en el peor caso
    //(Random Mix)
    //time = __HAL_TIM_GET_COUNTER(&htim8);
}
}

```

```

void controlCalc(uint8_t modo, uint32_t CP, int16_t offset, int16_t G){
    #define MODO_ESCALERA      1
    #define MODO_ESCALON      2
    #define MODO_RAMPA        3
    #define MODO_SENOIDAL     4
    #define MODO_RANDOM_E     5
    #define MODO_RANDOM_R     6
    #define MODO_RANDOM_S     7
    #define MODO_ARRANQUE     8
    #define MODO_HISTERESIS   9
    #define MODO_BARRIDO     10
    #define MODO_MIX          11
    #define MODO_RANDOM_M    12

```

```

    if ((cp == CP)&&(tabs>10000)) {
        HAL_RNG_GenerateRandomNumber(&hrng, &randomN);
        randomN = 360 + randomN/1272206; /*(12bits-360*2)/32bits
        switch (modo)
        {
        case MODO_ESCALERA://Aumenta poco a poco
            //(corriente de arranque umbral de 150 mA en vacio, +-360/480 mA en E1
            //E1:
            //ESCON->-360mA, -480mA, -370mA, -440mA
            //          390mA, 420mA, 380mA, 420mA

            if (macroetapa == 1){
                //inicios-->1800(-)/2220(+)
            }
        }
    }

```

```

        duty = 1840;
        macroetapa = 2;
    }
//    duty += 5;
//    if (duty >= max_duty) {
//        duty = middle_duty;
//    }
    duty -= 5;
    if (duty <= 0) {
        duty = middle_duty;
    }

    break;

case MODO_ESCALON://Señal cuadrada
    //CP de 600 para que no alcance v máx
    if (etapa) {
        duty = max_duty;//max_duty
        etapa = 0;
    } else {
        duty = min_duty;//min_duty
        etapa = 1;
    }
    //CP de 1 señal cuadrada de 0 al max
//    if(macroetapa>1200){
//        macroetapa = 0;
//    }else if(macroetapa>(tabs/6000)){//X para PULSOS (tabs/X para barrido)
//        etapa = 0;
//    }else{
//        etapa = 1;
//    }
//    if (etapa) {
//        duty = max_duty;
//    } else {
//        duty = middle_duty;
//    }
//    macroetapa++;
//    break;

case MODO_RAMPA://Señal triangular
    if (etapa) {
        duty+=4;
        if (duty >= max_duty){
            etapa = 0;
        }
    } else {
        duty-=4;
        if (duty <= min_duty) {
            etapa = 1;
        }
    }
//CP de unos 2
    break;

```

```

case MODO_SENOIDAL: //onda constante
    T = 500;//Periodo de onda en ms
    duty = A*sin(3*0.31415*t/T)+middle_duty;
    //CP debe estar a 1
    break;

case MODO_RANDOM_E: //Escalones aleatorios
    //CP de 600
    if (etapa) {
        duty = (randomN - 360)/2 + 360;
        etapa = 0;
    } else {
        duty = (randomN - 360)/2 + middle_duty;
        etapa = 1;
    }
    break;

case MODO_RANDOM_R: //Trayectoria con pendiente aleatoria
    if(macroetapa>=200){
        incremento = randomN/200;//CP en 3
        macroetapa = 0;
        //
        //
        //
        if(incremento%3!=0){
            etapa = !etapa;
        }
    }
    if (etapa) {
        duty+=incremento;
        if (duty >= max_duty) {
            duty = max_duty;
            etapa = 0;
        }
    } else {
        duty-=incremento;
        if (duty <= min_duty || duty > 10000) {
            duty = min_duty;
            etapa = 1;
        }
    }
    macroetapa++;

    break;

case MODO_RANDOM_S://Onda de amplitud y periodo variables aleatorios
    if(macroetapa>10000 && duty<(middle_duty + 5) && duty>(middle_duty - 5)){
        if(duty>middle_duty){
            etapa = 0;
        }else if(duty<(middle_duty)){
            etapa = 1;
        }
    }
    A = (randomN - 360)/2;
    HAL_RNG_GenerateRandomNumber(&hrng, &randomN);

```



```

case MODO_BARRIDO://Incremento progresivo de la frecuencia de onda
    if(macroetapa>3963 && duty<(middle_duty + 5) && duty>(middle_duty - 5)){
        if(duty>middle_duty){
            etapa = 0;
        }else if(duty<middle_duty){
            etapa = 1;
        }
        Tr = Tr*0.95;
        macroetapa = 0;
        t = 0;
        if(Tr <11){
            Tr = 1;
        }
    }
    A = (4096/2 - 360);
    if(etapa){
        duty = A*sin(3*0.31415*t/Tr) + middle_duty;
    }else{
        duty = middle_duty - A*sin(3*0.31415*t/Tr);
    }
    macroetapa++;
    //CP debe estar a 1
    break;

case MODO_MIX: //Superposición de 3 ondas
    T = 1000;//Periodo de onda en ms
    duty = A/2*sin(3*0.31415*t/T)+ A/4*sin(3*0.31415*t/T*20) +
A/4*sin(3*0.31415*t/T*5) + middle_duty;
    //CP debe estar a 1
    break;

case MODO_RANDOM_M: //Superposición de tres ondas aleatorias variables
    if(macroetapa>10000 && duty<(middle_duty + 5) && duty>(middle_duty - 5)){
        if(duty>(middle_duty)){
            etapa = 0;
        }else if(duty<(middle_duty)){
            etapa = 1;
        }
    }
    Ar1 = (randomN - 360)/6 + 281;//Para asegurar ondas excitantes
    //Ar1 = (randomN - 360)/8 + 422;
    HAL_RNG_GenerateRandomNumber(&hrrng, &randomN);
    Tr1 = 1 + randomN/4194304;//* 10bits/32bits
    HAL_RNG_GenerateRandomNumber(&hrrng, &randomN);
    randomN = randomN/1272206;//*(12bits-360*2)/32bits
    Ar2 = randomN/8;
    HAL_RNG_GenerateRandomNumber(&hrrng, &randomN);
    Tr2 = 1 + randomN/33554432;//* 7bits/32bits
    HAL_RNG_GenerateRandomNumber(&hrrng, &randomN);
    randomN = randomN/1272206;//*(12bits-360*2)/32bits
    Ar3 = randomN/8;
    HAL_RNG_GenerateRandomNumber(&hrrng, &randomN);

```

```

        Tr3 = 1 + randomN/33554432; /*7bits/32bits

        macroetapa = 0;
        t = 0;
    }
    if(etapa){
        duty = Ar1*sin(3*0.31415*t/Tr1) + Ar2*sin(3*0.31415*t/Tr2) +
Ar3*sin(3*0.31415*t/Tr3) + middle_duty;
    }else{
        duty = middle_duty - Ar1*sin(3*0.31415*t/Tr1) -
Ar2*sin(3*0.31415*t/Tr2) - Ar3*sin(3*0.31415*t/Tr3);
    }
    macroetapa++;
    //CP debe estar a 1
    break;

default:
    duty = middle_duty;
    break;
}
cp = 1;

//OFFSET
if(offset < -1688){
    offset = 0;
}
if(offset > 1688){
    offset = 0;
}

//accion = (duty - 2048)*(1688 - abs(offset))/1688 + 2048 + offset;
accion = (duty - 2048)/G + 2048 + offset;

}else if ((cp != CP)){
    cp++;
}
if(tabs>10000){
    t++;
}
tabs++;
}
}

```

```

void compresionASCII(uint16_t number1, uint16_t number2, uint16_t number3, uint16_t number4,
char *ascii) {

```

```

    // Dividir el número en (unidades con decenas) y (centenas con millares)
    //menos en el primer caso ya que va a ser un número más grande
    uint8_t units = number1 % 10;
    uint8_t tens = (number1 / 10) % 10;
    uint8_t hundreds = (number1 / 100)% 10;
    uint8_t thousands = number1 / 1000; //no va a ser mayor a 9999

```

```

// Generar los caracteres ASCII correspondientes a cada dígito
ascii[0] = thousands*10 + hundreds + 35;
ascii[1] = tens + 48;
ascii[2] = units + 48; // Añadir 48 para ignorar otros caracteres de ASCII

units = number2 % 10;
tens = (number2 / 10); //no va a ser mayor a 99

// Generar los caracteres ASCII correspondientes a cada dígito
ascii[3] = tens*10 + units + 35; //encoder 1 solo hay 90 valores

units = number3 % 10;
tens = (number3 / 10) % 10;
hundreds = number3 / 100; //no va a ser mayor a 999

// Generar los caracteres ASCII correspondientes a cada dígito
ascii[4] = hundreds*10 + tens + 35;
ascii[5] = units + 48; // Añadir 48 para ignorar otros caracteres de ASCII

units = number4 % 10;
tens = (number4 / 10) % 10;
hundreds = (number4 / 100)% 10;
thousands = number4 / 1000; //no va a ser mayor a 9999

ascii[6] = thousands*10 + hundreds + 35;
ascii[7] = tens + 48;
ascii[8] = units + 48; // Añadir 48 para ignorar otros caracteres de ASCII
}
/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
__disable_irq();
while (1)
{
}
/* USER CODE END Error_Handler_Debug */
}

```


ANEXO III. CÓDIGO POSPROCESO MATLAB

```
%%DATOS PSEUDO CODIFICADOS ASCII
clear all;
close all;
clc
FICHERO = 'Data_2024-07-01_210048.txt';
addpath(genpath('C:\Users\Asus\Desktop\TEFEEME\MATLABO'));
addpath(genpath('C:\Users\Asus\Desktop\BUENA DATA'));
addpath(genpath('C:\Users\Asus\Desktop\Datos Postproceso'));
opts = detectImportOptions(FICHERO);
opts = setvartype(opts, 'string');
opts.Delimiter = " ";
opts.VariableNamesLine = 0;
opts.DataLine = 1;
%opts.VariableNamingRule = 'preserve';
DATOS=rows2vars(readtable(FICHERO,opts));
Aascii = DATOS(:,2);
%%
A = descomprimirASCIIQuad(Aascii);
time = (1:size(A, 1))';
B = [time, A];
%%14 segundos de procesado para 10 s de exp (4 min para 80 s de exp)
%%
C=B;
rango3=91;
B3=B(:,3);
C(:,3) = deslimitarU(B3,rango3);
rango4=910;
B4=B(:,4);
C(:,4) = deslimitarU(B4,rango4);
%%
%Convertir unidades a segundos, amperios y grados
[filas, ~] = size(C);
SI = C(1:filas-1, :);
SI(:,1) = C(1:filas-1,1)*150/1000000;%Segundos
%%SI(:,2) = C(1:filas-1,2)*1.781/(2048-360);%A %(teoría +-1.780A) pero
%comprobar ESCON
SI(:,2) = (C(1:filas-1,2)+47)*0.545*3.5/1687*1.8;%A %(la ref. es un % de la
%I nominal) comprobar DIGITAX HD
SI(:,3) = C(1:filas-1,3)*360/8192;%grados
SI(:,4) = C(1:filas-1,4)*360/589824;%grados
%SI(:,5) = C(1:filas-1,5)*2.0/2048*0.975 + 0.014;%A %calibrar con ESCON
%también
SI(:,5) = C(1:filas-1,5)*0;%A %DIGITAX HD no devuelve corriente
SI(:,6) = SpeedCalc(SI(:,3),1);%RPM
d1 = designfilt("lowpassiir",FilterOrder=12, ...
    HalfPowerFrequency=0.15,DesignMethod="butter");
SI(:,7) = filtfilt(d1,SI(:,6));%RPM
```

```

function A = descomprimirASCIIQuad(ascii)
A00 = table2array(ascii(:,1));
[n_filas, ~] = size(A00); % Obtenemos el número de filas de la matriz
Aclean = A00(1:n_filas-3, :); %se quitan posibles valores a trozos
valor = zeros(1, 9);
cadena = Aclean{2,1};
valor(1) = double(cadena(1))-35;
valor(2) = double(cadena(2))-48;
valor(3) = double(cadena(3))-48;
valor(4) = double(cadena(4))-35;
valor(5) = double(cadena(5))-35;
valor(6) = double(cadena(6))-48;
valor(7) = double(cadena(7))-35;
valor(8) = double(cadena(8))-48;
valor(9) = double(cadena(9))-48;
A(1,1) = valor(1)*100 + valor(2)*10 + valor(3) - 2048;
A(1,2) = valor(4);
A(1,3) = valor(5)*10 + valor(6);
A(1,4) = valor(7)*100 + valor(8)*10 + valor(9) - 2048;
for i = 2:(n_filas-3)
    cadena = Aclean{i,1};
    valor(1) = double(cadena(1))-35;
    valor(2) = double(cadena(2))-48;
    valor(3) = double(cadena(3))-48;
    valor(4) = double(cadena(4))-35;
    valor(5) = double(cadena(5))-35;
    valor(6) = double(cadena(6))-48;
    valor(7) = double(cadena(7))-35;
    valor(8) = double(cadena(8))-48;
    valor(9) = double(cadena(9))-48;

    A(i,1) = valor(1)*100 + valor(2)*10 + valor(3) - 2048;
    A(i,2) = valor(4);
    A(i,3) = valor(5)*10 + valor(6);
    A(i-1,4) = valor(7)*100 + valor(8)*10 + valor(9) - 2048;
end
end

```

```

function C = deslimitarU(B, rango)
[long, ~] = size(B);
diff = 0;
big = B(1);
C(1) = 0;
half = rango/2;
for i = 2:long
    if abs(B(i)-B(i-1))>half && B(i)>half
        diff = rango-abs(B(i)-B(i-1));
        big = big - diff;
    elseif abs(B(i)-B(i-1))>half && B(i)<half
        diff = rango-abs(B(i)-B(i-1));
        big = big + diff;
    else
        big = big + (B(i)-B(i-1));
        diff = 0;
    end
    C(i) = big;
end
end

```

```

%% Guardar Datos
NombreArchivo = 'PosDataE2';
E2_Barrido_2 = SI;
cd('C:\Users\Asus\Desktop\Datos Postproceso');
save(NombreArchivo, 'E2_Barrido_2', '-append');%, '-append' para no
%sobreescribir
cd('C:\Users\Asus\Desktop');

```

ANEXO IV. CÓDIGO IDENTIFICACIÓN MATLAB

```
%%IDENTIFICACIÓN
clear all;
close all;
clc
addpath(genpath('C:\Users\Asus\Desktop\TEFEEME\MATLABO'));
addpath(genpath('C:\Users\Asus\Desktop\BUENA DATA'));
addpath(genpath('C:\Users\Asus\Desktop\Datos Postproceso'));
load("PosDataEl_Ali.mat");
load("PosDataEl_Desali.mat");
%%
MainNo0=EscalonesOff_1;% (9500:end,:) (1:15000,:)      %E1A_
%%(T,U,th,q,I,thpmla,thp,thpp,qp,qqp)
% Añadir una fila de ceros al principio
fila_de_ceros = zeros(1, size(MainNo0, 2));
Main = [fila_de_ceros; MainNo0];
T=Main(:,1);% s
U=Main(:,2);% A
th=Main(:,3)*(-1)*2*pi/360;% rad
q=Main(:,4)*(-1)*2*pi/360;% rad
I=Main(:,5);% A
th_0 = th;
q_0 = q;
%FiltFilt
d1 = designfilt("lowpassiir",FilterOrder=12, ...
    HalfPowerFrequency=0.15,DesignMethod="butter");
thff = filtfilt(d1,th);
%Filtro BUTT
[b, a] = butter(3, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
thbutt = filter(b, a, th); % x es tu señal de entrada
[b, a] = butter(2, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
thbutfff = filtfilt(b, a, th); % x es tu señal de entrada
[b, a] = butter(4, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
qbutt = filter(b, a, q); % x es tu señal de entrada
%Filtro FIR
b = fir1(18, 0.01);
a = 1; % Filtros FIR tienen a = 1
thfir = filter(b, a, th);
%Suavizar
ths = movmean(th, 3); % Suavizado
[b, a] = butter(2, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
ths = filter(b, a, ths); % x es tu señal de entrada
%interpolación
% Nueva escala de tiempo con el doble de puntos
% IP=20;
% [long, ~] = size(T);
% T_il = linspace(1, long, long*IP) '*0.00015;
% % Interpolación lineal
% th_il = interp1(T, th,T_il, 'linear');
% %filtrado adicional
% % d1 = designfilt("lowpassiir",FilterOrder=12, ...
% %     HalfPowerFrequency=0.15/IP,DesignMethod="butter");
```

```

% % thff_il = filtfilt(dl,th_il);% °
%
% [b, a] = butter(3, 0.1/IP, 'low'); % Filtro Butterworth de orden 3
%(fn=2*fc/fm)
% thbutt_il = filter(b, a, th_il); % x es tu señal de entrada
%
% b = fir1(200, 0.1/IP);
% a = 1; % Filtros FIR tienen a = 1
% thfir_il = filter(b, a, th_il);
%VELOCIDAD
thp = SpeedCalc(th,1)';%           250 periodos límite superior
qp = SpeedCalc(q,1)';
thpf = SpeedCalc(th,10);
thpfe = SpeedCalc(th,100);%       250 periodos límite superior
qpf = SpeedCalc(q,10);%
thpff = SpeedCalc(thff,1);
%thpff = MiddleSpeedCalc(thff);
thpbutt = SpeedCalc(thbutt,1);
qpbutt = SpeedCalc(qbutt,1);
[b, a] = butter(1, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
thbutt = filter(b, a, th); % x es tu señal de entrada
thpP = SpeedCalc(thbutt,1);
[b, a] = butter(1, 0.1); % Filtro Butterworth de orden 3 (fn=2*fc/fm)
qbutt = filter(b, a, q); % x es tu señal de entrada
qpP = SpeedCalc(qbutt,1);
thpfir = SpeedCalc(thfir,1);
thpFM = MiddleSpeedCalc(th,10);
thps = SpeedCalc(th,10);
thpFMBUT = MiddleSpeedCalc(thbutt,1);
thp_0 = thp;
qp_0 = qp;
%thpbutttff = SpeedCalc(thbutttff,1);
% thpbutttff = gradient(thbutt,1)/0.00015/360*60;
%thpbutttff = SpeedCalc(th,1);
%thpbutttff = SpeedCalc(thSG,1);
%thpbutttK = SpeedCalc(thbutt,1)*1.011;
% thpbutttK = POSTFilt(thpbutt,6);
% Diseño del filtro notch
%vamos a interpolar
% IP=4;
% [long, ~] = size(T);
% T_il = linspace(1, long, long*IP) '*0.00015;
% % Interpolación lineal
% thp_il = interp1(T, thp,T_il, 'linear');
% [b, a] = iirnotch(0.6, 0.0001);
% thpbutttN = filter(b, a, thp_il);
% thp_il = SpeedCalc_il(th_il,10,IP);
% %thpff_il = SpeedCalc_il(thff_il,10,IP);
% thpbutt_il = SpeedCalc_il(thbutt_il,1,IP);
% thpfir_il = SpeedCalc_il(thfir_il,30,IP);
% Compensar el desfase (el desfase es aproximadamente la mitad del orden del
filtro)
%delay = 4*2;

```

```

%thpbuttc = circshift(thpbutt, -delay);
% Eliminar los valores iniciales y finales afectados por la compensación
%thpbuttc(1:delay) = NaN;
%thpbuttc(end-delay+1:end) = NaN;
%ACELERACIÓN
thppP=AccelCalc(thp,100);%
thpp=AccelCalcFine(th,15);%
[b, a] = butter(3, 0.1); % Filtro Butterworth
thppbutt = filter(b, a, thpp);
qpp=AccelCalcFine(q,30);%
[b, a] = butter(3, 0.1); % Filtro Butterworth
qppbutt = filter(b, a, qpp);
thpp_0=thpp;
qpp_0=qpp;
%Decisión:
th = th_0;%      rad
thp = thp_0;%    rad/s
thpp = thpp_0;%  rad/s^2
thpF = thpf';%   rad/s
thppF = thppbutt;% rad/s^2
q = q_0;%        rad
qp = qp_0;%      rad/s
qpp = qpp_0;%    rad/s^2
qpF = qpf';%     rad/s
qppF = qppbutt;% rad/s^2
%Representación
ploteo = 20;%%SEGUNDOS DE REPRESENTACIÓN
zero0 = 0;
figure;
plot(T,qpbutt*(-1),'r', LineWidth=2)
hold on
plot(T,thpbutt/30,'b', LineWidth=3)
xlim([zero0 ploteo+zero0]);
hold off
legend('Velocidad Secundario filtro Butt(rad/s)', 'Velocidad Primario filtro Butt(rad/s)/(-30)')
xlabel('Tiempo(s)');
ylabel('Según Leyenda');
% title('Representación temporal de datos de corriente y primario del %Harmonic Drive');
title('Representación temporal de las velocidades del Harmonic Drive');
grid on
%%
%%%%MODELO
%NUMERADOR[A + Bz-1 + Cz-2 ...] DENOMINADOR[Ad + Bdz-1 + Cdz-2
%...]
%EN DISCRETO ES AL REVÉS QUE EN CONTINUO!!
%MODEL = idtf([0, 154.2, -153.7], [1, -1.882, 0.3318, 1.09, -0.534],
%0.00015);
% bode(MODEL);
% step(MODEL);
% impulse(MODEL);
Kt = 0.217;%      Constante de par(Nm/A)

```

```

K = 3400;%           Rigidez torsional T1->2 Nm, T2->6.9 Nm
%[3400,2400,1900] (Nm/rad)
tm = 0.00015;%      Periodo de muestreo (s)
N = -30;%           Reducción(-)
Jm = 0.000306;%     Momento de inercia motor (Kgm^2)
bm = 0.02044817256;% Coeficiente de fricción viscosa motor (Nm/(rad/s))
Je = 0.0000033;%    Momento de inercia harmonic drive (Kgm^2)
%?????????????No se saben. Rango->[0.01-0.1]
be = 0.1;%          Coeficiente de fricción viscosa harmonic
%drive (Nm/(rad/s))
B = 0.11;%          Coeficiente de amortiguamiento elástico del harmonic
%drive (Nm/(rad/s))
%help polyfit
a0 = 1;  b0 = 1;  c0 = 1;
d0 = 1;  e0 = 1;  f0 = 1;  g0 = 1;
Gth = tf([a0*(Kt*Je), b0*(Kt*(B+be)), c0*(Kt*K)], [d0*(Je*Jm),
e0*(Je*(B/N^2+bm)+Jm*(B+be)) ...
, f0*(K*(Je/N^2+Jm)), g0*(K*(be/N^2+bm)), 0]);
Gq = tf([Kt*B/N, Kt*K/N], [Je*Jm, Je*(B/N^2+bm)+Jm*(B+be) ...
, K*(Je/N^2+Jm), K*(be/N^2+bm), 0]);
Gsimp = tf([Kt],[Jm, bm, 0]);
GZth = c2d(Gth, tm);
GZq = c2d(Gq, tm);
GZsimp = c2d(Gsimp, tm);
% Simula la respuesta del sistema a la entrada
output = lsim(GZth, I, T);
% Grafica la entrada y la salida
plot(T, I, 'b', T, output, 'r');
xlabel('Tiempo');
ylabel('Amplitud');
hold on
plot(T,th)
legend('Entrada', 'thCalc','thReal');
title('Respuesta del sistema a la entrada');
hold off
y = th;
u = I;
z1E = iddata(y, u, 0.00015, 'Name' , 'Harmonic Hell1E' , 'TimeUnit' , 's' );
%%
FuncVisualTp(Gth);
%%
% Encontrar las raíces (polos) del denominador original
poles = roots([a, b, c]);
% Las raíces serán de la forma -1/Tp1 y -1/Tp2
Tp1 = -1 / poles(1);
Tp2 = -1 / poles(2);
% Definir la constante de la función de transferencia original
K = 1; % Define tu valor de K
% Crear la función de transferencia en el nuevo formato
numerator = K;
denominator = conv([1, Tp1], [1, Tp2]);
G = tf(numerator, denominator);
%%

```

```

% Grafica la entrada y la salida
% Simula la respuesta del sistema a la entrada
output = lsim(tf8_1, I, T);
plot(T, I, 'b', T, output, 'r');
xlabel('Tiempo');
ylabel('Amplitud');
hold on
plot(T,th)
legend('Entrada', 'thCalc','thReal');
title('Respuesta del sistema a la entrada');
hold off
%%
% Modelo ARX no lineal (IDNLARX)
y = [th q];
u = I;
z = iddata(y, u, 0.00015, 'Name' , 'Harmonic Hell' , 'TimeUnit' , 's' );
NN3 = [2 2 1 3;2 2 1 3];
Fcn1 = idWaveletNetwork(73); % output function for the first output
Fcn2 = idWaveletNetwork(107); % output function for the second output
mw3 = nlarx(z, NN3, [Fcn1; Fcn2]);
compare(z,mw3)

```



```

function vel = SpeedCalc(pos, TM)
[long, ~] = size(pos);
for i = 1:TM
    vel(i) = 0;
end
for i = (TM+1):long
    vel(i) = (pos(i) - pos(i-TM))/(0.00015*TM); %%rad/s
end
end

```

```

function vel = MiddleSpeedCalc(pos, TM)
[long, ~] = size(pos);
for i = 1:TM
    vel(i) = 0;
    vel(long-i+1) = 0;
end
for i = (TM+1):(long-TM)
    vel(i) = (pos(i+TM) - pos(i-TM))/(0.00015*TM^2);
end
end

```

```

function accel = AccelCalcFine(pos, TM)
[long, ~] = size(pos);
for i = 1:TM*2
    accel(i,1) = 0;
end
for i = (TM*2+1):(long)
    accel(i,1) = (pos(i) - 2*pos(i-TM) + pos(i-TM*2))/(0.00015*TM)^2; %%rad/s^2
end
end

```

```

DELAYIth = 1;

```

```

DELAYthq = 3 + DELAYith;%21,32,10,7
NL = length(T);
Phi = zeros(NL, 6); % Matriz de regresores
Phi(1:(end-DELAYith), 1) = th((DELAYith+1):end);
Phi(2:(end-DELAYith), 2) = th((DELAYith+1):end-1);
Phi(1:(end-DELAYthq), 3) = q((DELAYthq+1):end);
Phi(2:(end-DELAYthq), 4) = q((DELAYthq+1):end-1);
Phi(:, 5) = I;
Phi(:, 6) = ones(NL,1);
Y = zeros(NL, 2);
Y(3:(end-DELAYith), 1) = th((DELAYith+1):end-2);
Y(3:(end-DELAYthq), 2) = q((DELAYthq+1):end-2);
PhiT=Phi.';
format long
% XE = inv(PhiT*Phi)*PhiT*Y
XE = PhiT*Phi\PhiT*Y
%XE2=pinv(Phi)*Y
Kt = 0.217;%           Constante de par(Nm/A)
K = 3400;%           Rigidez torsional T1->2 Nm, T2->6.9 Nm
%[3400,2400,1900](Nm/rad)
tm = 0.00015;%       Periodo de muestreo(s)
N = -30;%           Reducción(-)
Jm = 0.000306+0.0000022;%       Momento de inercia motor y eje E134 con
%tornillos(Kgm^2)
bm = 0.02044817256;%       % Coeficiente de fricción viscosa motor(Nm/(rad/s))
Je = 0.0000033+0.0000175;%       Momento de inercia harmonic drive y
%eslabón(Kgm^2)
%?????????????No se saben. Rango->[0.01-0.1]
be = 1;% 12.4224       Coeficiente de fricción viscosa harmonic
%drive(Nm/(rad/s))
B = 1;% 1283          Coeficiente de amortiguamiento elástico del harmonic
%drive(Nm/(rad/s))
Troz = 0.064;%       Par de rozamiento simplificado (Nm)
a = -K*tm^2/(N^2*Jm)-tm*(bm+B/N^2)/Jm-1;
b = tm*(bm+B/N^2)/Jm+2;
c = tm*(K*tm+B)/(N*Jm);
d = tm*B/(N*Jm);
e = tm^2*Kt/Jm;
e2 = Troz;
f = K*tm^2/(N*Je)+tm*B/(N*Je);
g = -tm*B/(N*Je);
h = -tm^2*K/Je-1-tm*(be+B)/Je;
i = 2+tm*(be+B)/Je;
XE_teorica = [a b c d e e2;f g h i 0 0]'
Yteo=Phi*XE_teorica;
Yest=Phi*XE;
plot(T,Y(:,1),'x')
hold on
plot(T,Y(:,2)*30,'x')
plot(T,Yteo(:,1),'-.k')
plot(T,Yteo(:,2)*30,'-.b')
plot(T,Yest(:,1),'r','LineWidth',2)
plot(T,Yest(:,2)*30,'g','LineWidth',2)

```

```
hold off
legend('Posición Primario Real', 'Posición Secundario Real x30', 'Posición
Primario Teórica', 'Posición Secundario Teórica x30', 'Posición Primario
Estimada', 'Posición Secundario Estimada x30')
xlabel('Tiempo(s)');
ylabel('Posición(rad)');
title('Comparación de modelos polinómicos de parámetros del Harmonic Drive');
grid on
```



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

**DESARROLLO DE EXPERIMENTOS PARA
LA IDENTIFICACIÓN DE UNA ARTICULACIÓN
CON HARMONIC DRIVE**

2. PLANOS

TRABAJO FINAL DEL

Máster Universitario en Ingeniería Mecatrónica

REALIZADO POR

Rafael Luis Saneleuterio Temporal

TUTORIZADO POR

Ranko Zotovic Stanisic

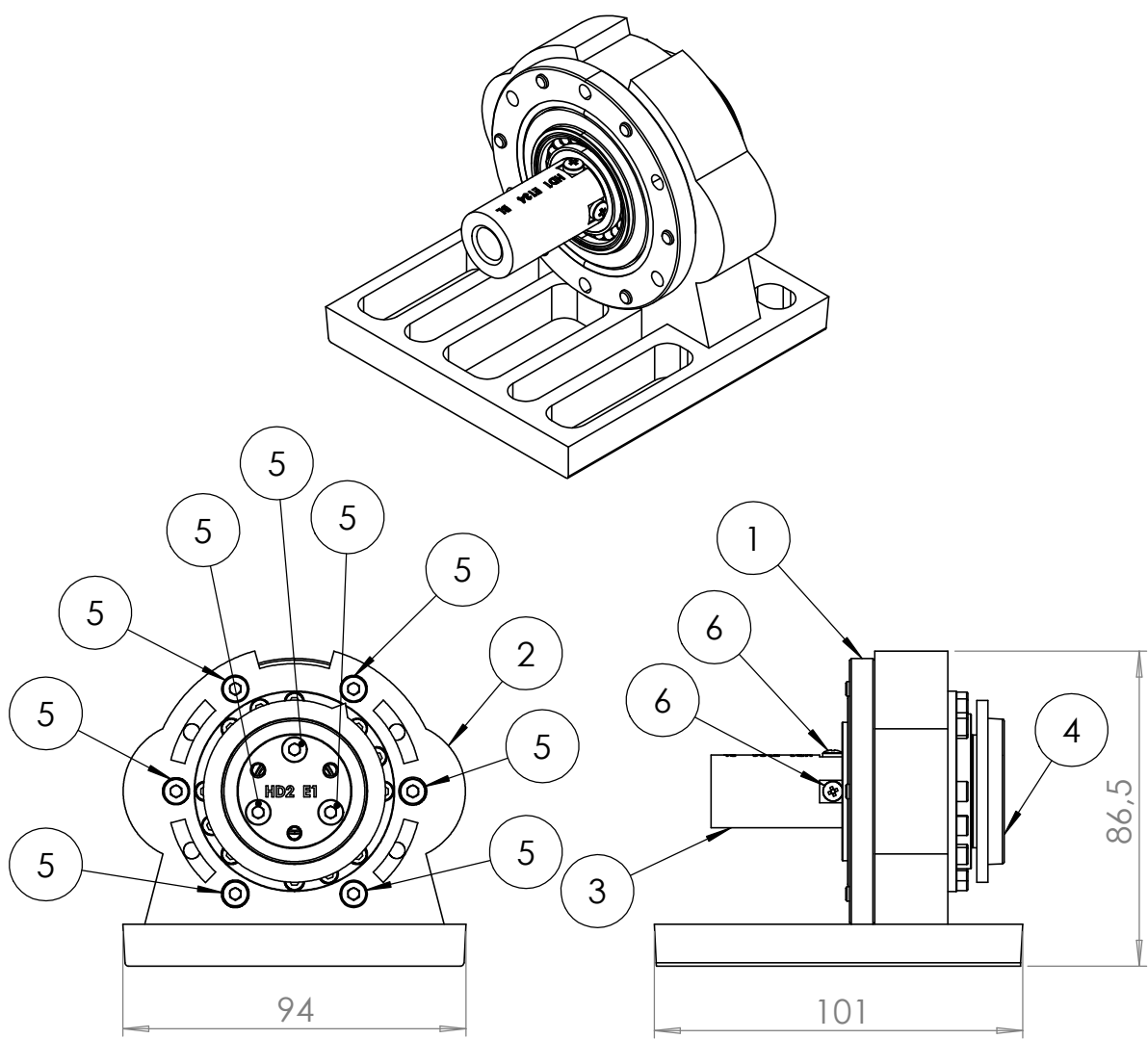
CODIRECTOR:

Santiago Carlos Gutiérrez Rubert

CURSO ACADÉMICO: 2023/2024

Índice planos

1. Conjunto E1: Plano 1	133
1.1. Base E1: Plano 2	134
1.2. Eje E134: Plano 3	135
1.3. Soporte E1: Plano 4	136
2. Conjunto E2: Plano 5	137
2.1. Eje E23: Plano 6	138
2.2. Soporte E2: Plano 7	139
3. Conjunto E3: Plano 8	140
4. Conjunto E4: Plano 9	141
4.1. Base E4: Plano 10	142



6	Tornillo de cabeza ranurada	2	Acero ISO 7045 M3x4
5	Tornillo de cabeza hex. hueca	9	Acero ISO 4762 M4x12
4	Soporte encoder E1: Plano N° 4	1	Nylon ϕ 50x13,5 mm
3	Eje E134: Plano N° 3	1	Nylon ϕ 20x41 mm
2	Base E123: Plano N° 2	1	Nylon 94x86,5x101 mm
1	Harmonic Drive HFUC_14_2UH	1	Según ficha técnica
Marca	Descripción	Cantidad	Material y dimensiones

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

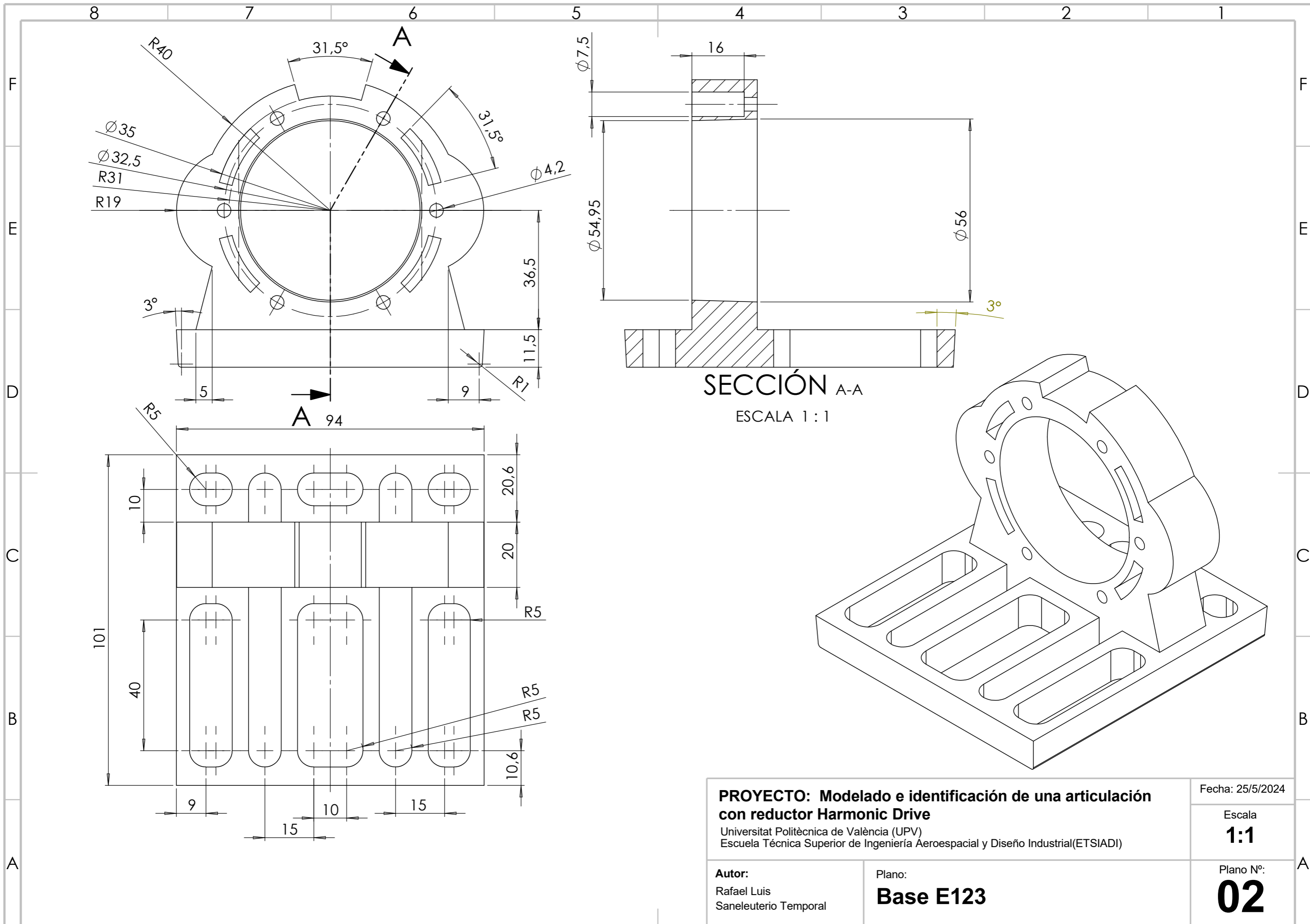
Fecha: 25/5/2024

Escala
1:2

Autor:
Rafael Luis
Saneleuterio Temporal

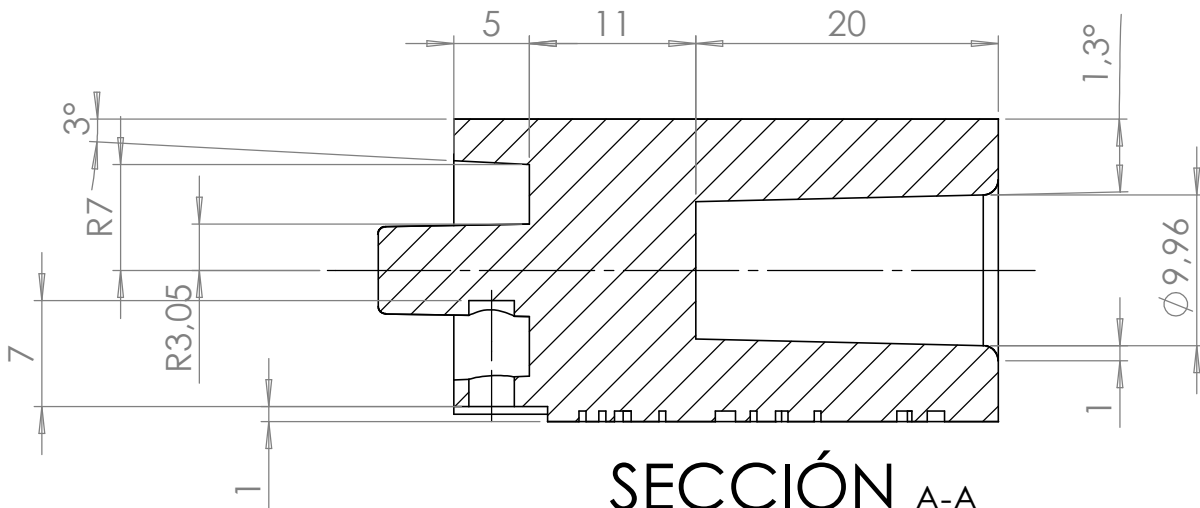
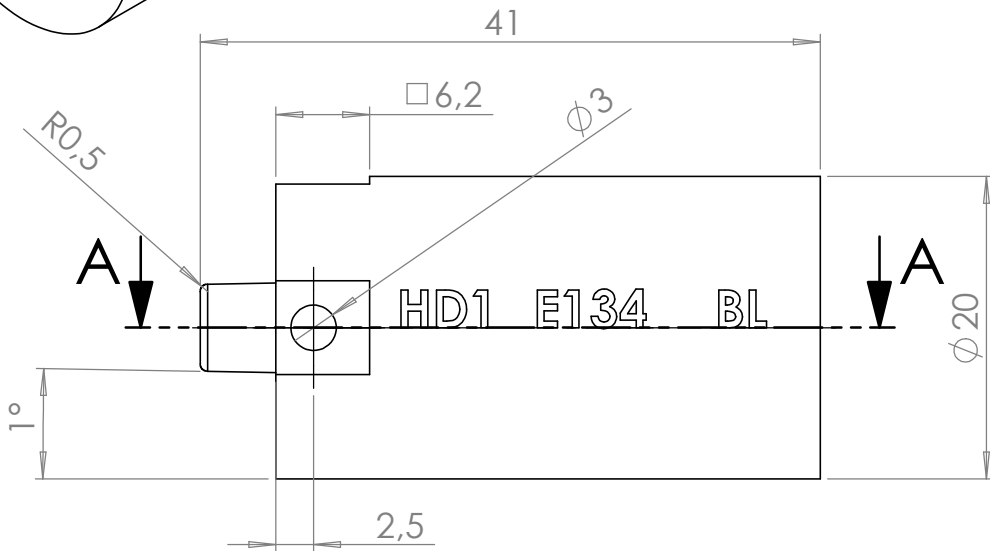
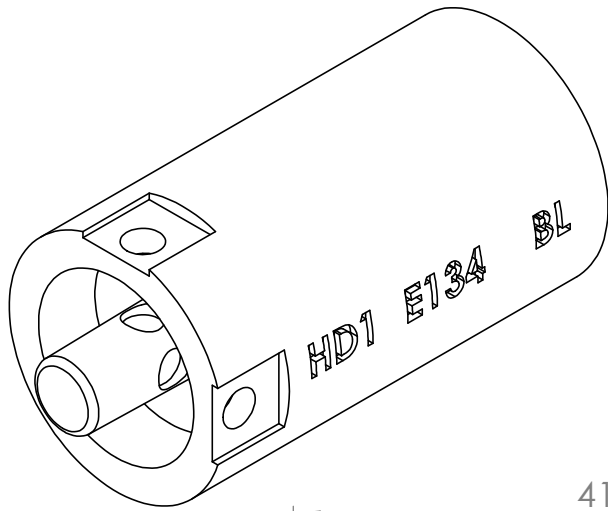
Plano:
Conjunto E1

Plano N°:
01



SECCIÓN A-A
ESCALA 1 : 1

<p>PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive Universitat Politècnica de València (UPV) Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)</p>		Fecha: 25/5/2024
		Escala 1:1
Autor: Rafael Luis Saneleuterio Temporal	Plano: Base E123	Plano Nº: 02



SECCIÓN A-A

ESCALA 2 : 1

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

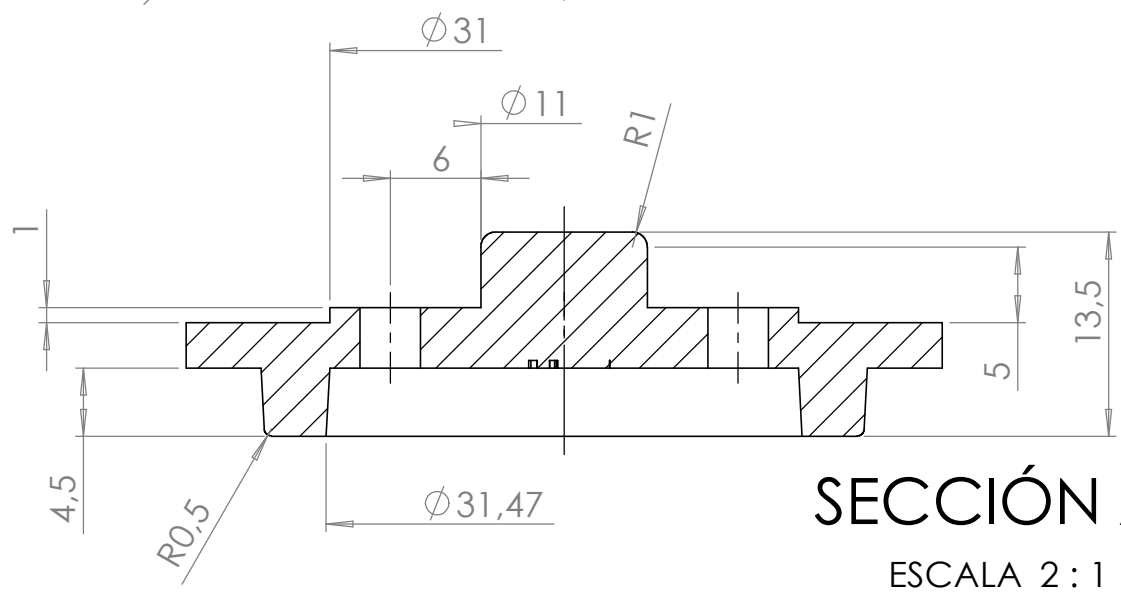
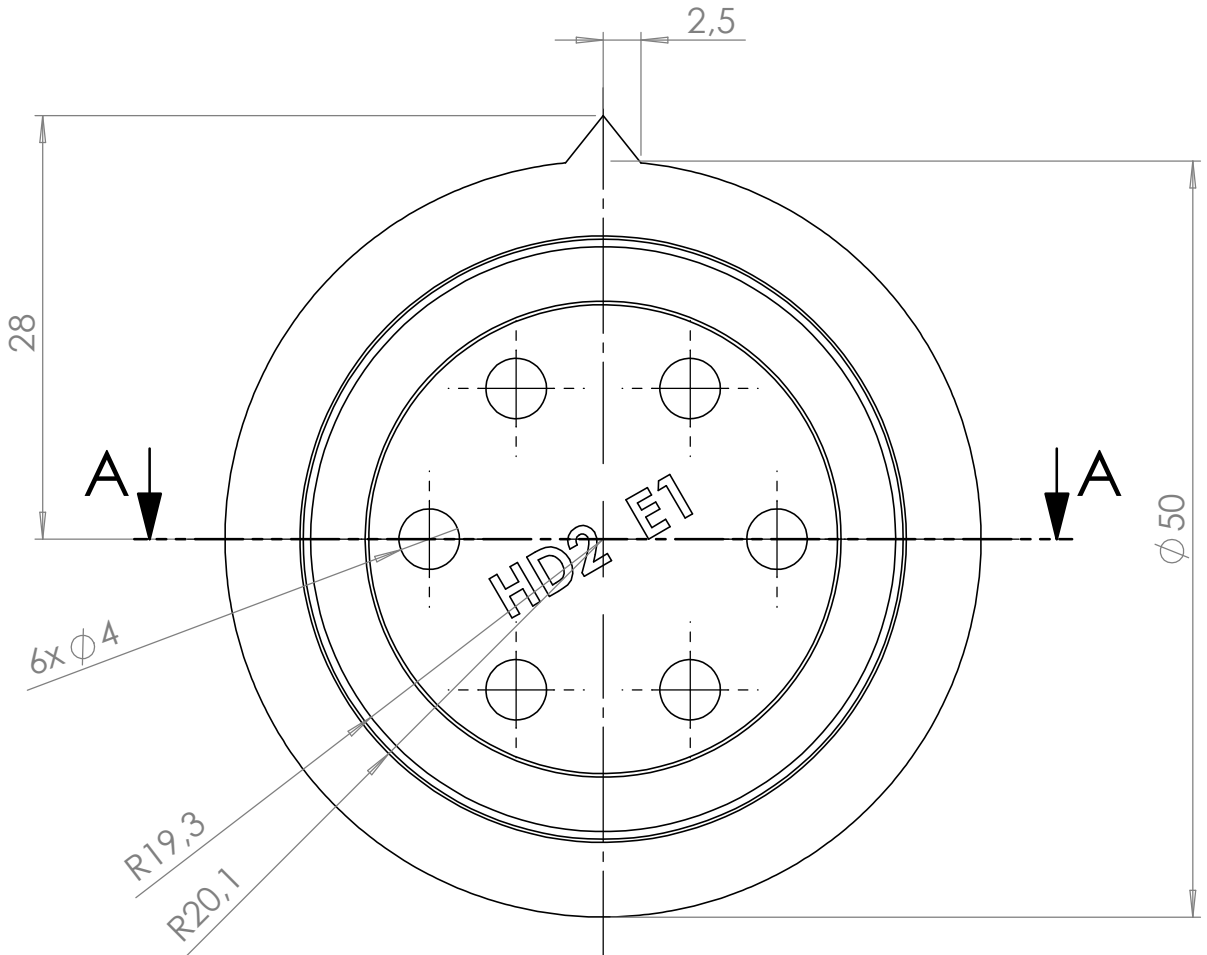
Fecha: 25/5/2024

Escala
2:1

Autor:
Rafael Luis
Saneleuterio Temporal

Plano:
Eje E134

Plano N°:
03



SECCIÓN A-A
ESCALA 2 : 1

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

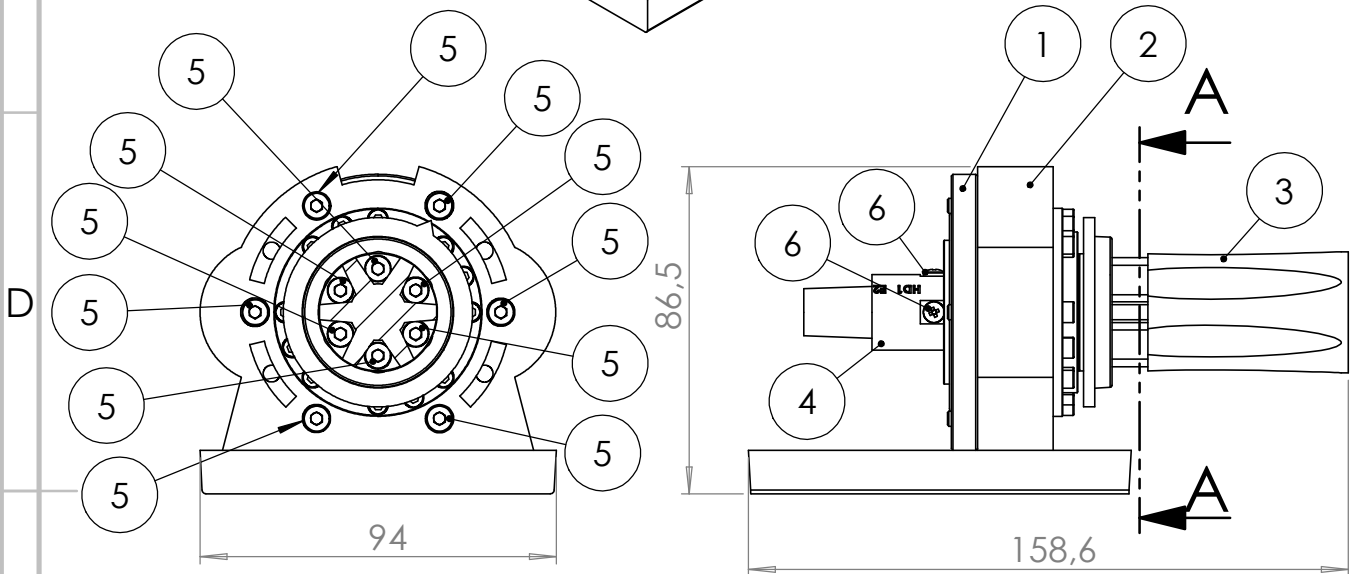
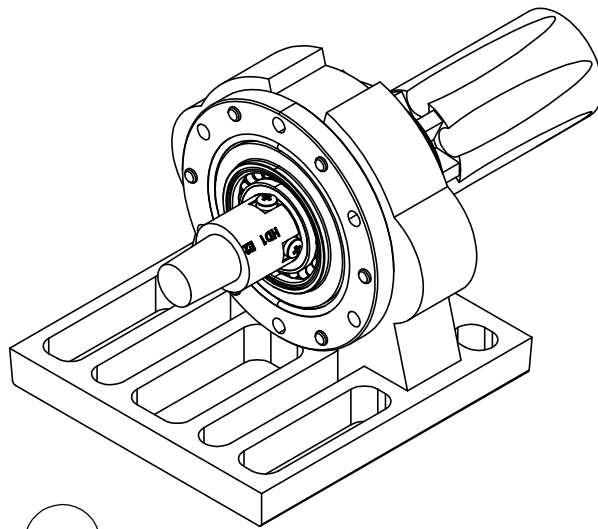
Fecha: 25/5/2024

Escala
2:1

Autor:
Rafael Luis
Saneleuterio Temporal

Plano:
Soporte encoder E1

Plano N°:
04



SECCIÓN A-A

ESCALA 1 : 2

6	Tornillo de cabeza ranurada	2	Acero ISO 7045 M3x4
5	Tornillo de cabeza hex. hueca	12	Acero ISO 4762 M4x12
4	Soporte encoder E2: Plano N° 7	1	Nylon ϕ 20x42 mm
3	Eje E23: Plano N° 7	1	Nylon ϕ 50x76 mm
2	Base E123: Plano N° 2	1	Nylon 94x86,5x101 mm
1	Harmonic Drive HFUC_14_2UH	1	Según ficha técnica
Marca	Descripción	Cantidad	Material y dimensiones

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

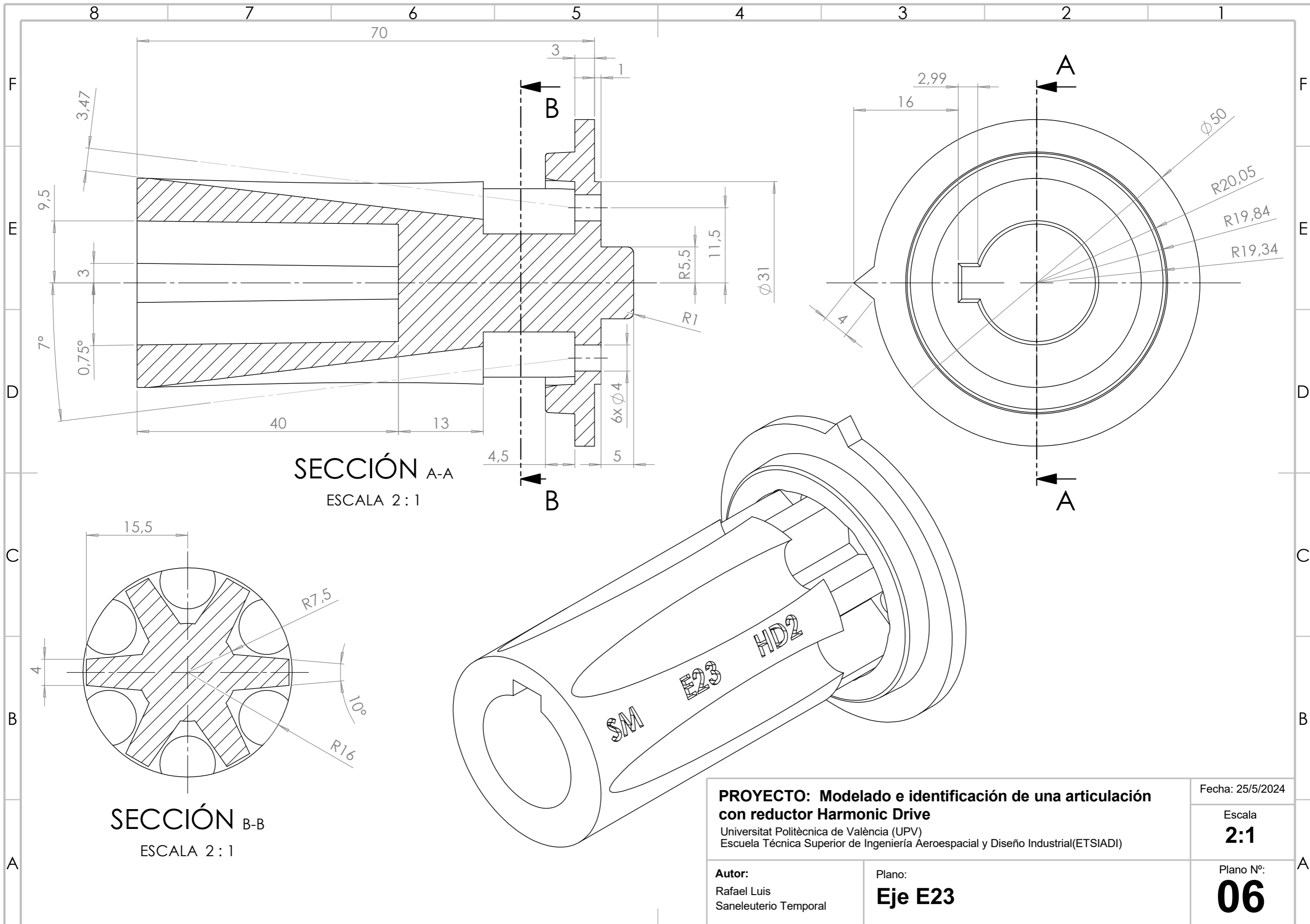
Fecha: 25/5/2024

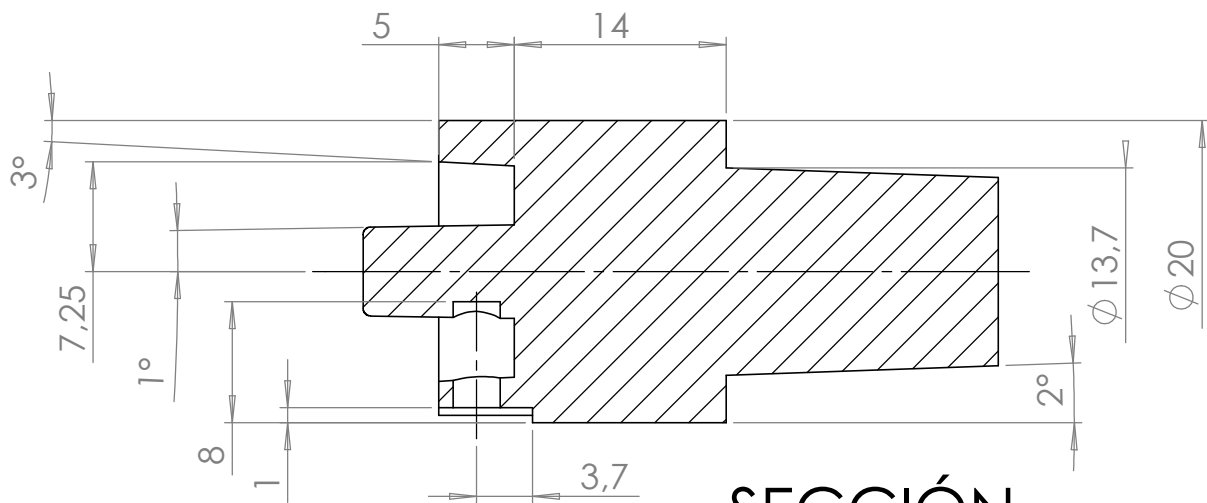
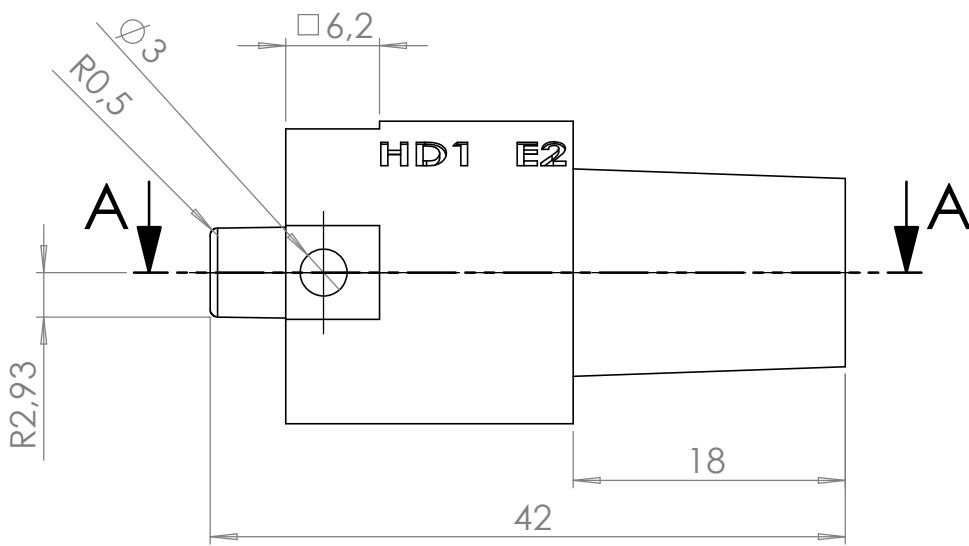
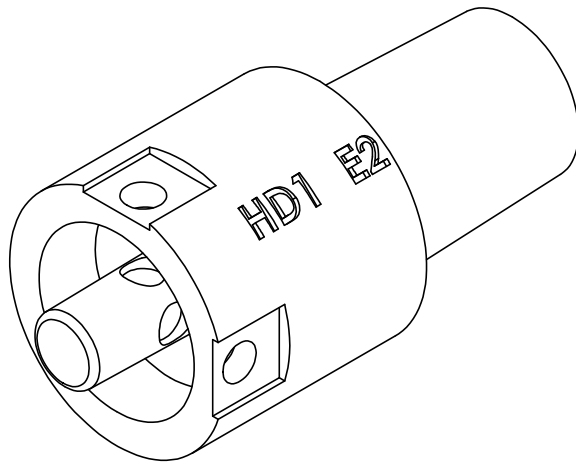
Escala
1:2

Autor:
Rafael Luis
Saneleuterio Temporal

Plano:
Conjunto E2

Plano N°:
05





SECCIÓN A-A

ESCALA 2 : 1

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

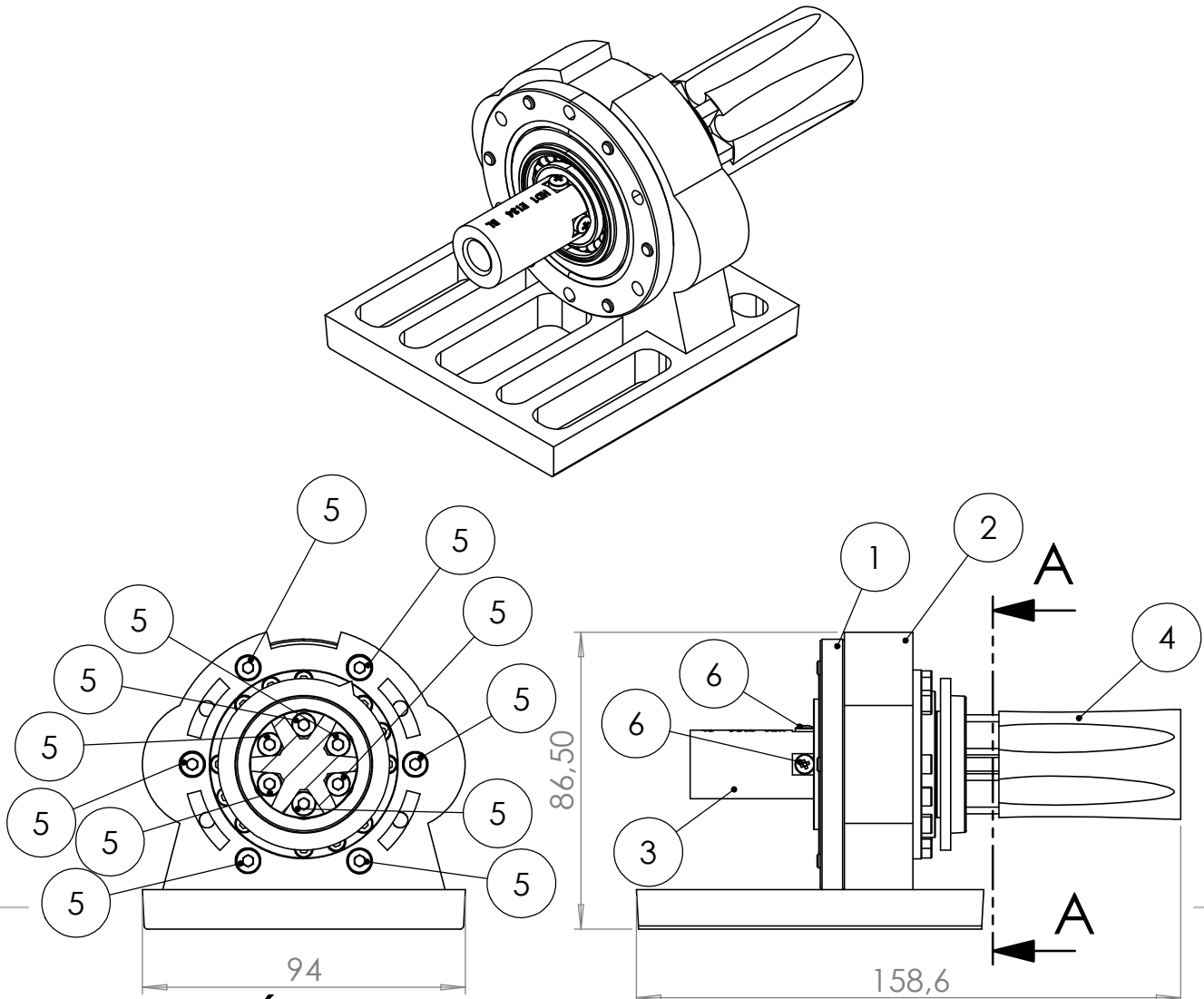
Fecha: 25/5/2024

Escala
2:1

Autor:
Rafael Luis
Saneleuterio Temporal

Plano:
Soporte encoder E2

Plano N°:
07



SECCIÓN A-A
ESCALA 1 : 2

6	Tornillo de cabeza ranurada	2	Acero ISO 7045 M3x4
5	Tornillo de cabeza hex. hueca	12	Acero ISO 4762 M4x12
4	Eje E23: Plano N° 7	1	Nylon ϕ 20x42 mm
3	Eje E134: Plano N° 3	1	Nylon ϕ 20x41 mm
2	Base E123: Plano N° 2	1	Nylon 94x86,5x101 mm
1	Harmonic Drive HFUC_14_2UH	1	Según ficha técnica
Marca	Descripción	Cantidad	Material y dimensiones

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

Fecha: 25/5/2024

Escala
1:2

Autor:
Rafael Luis
Saneleuterio Temporal

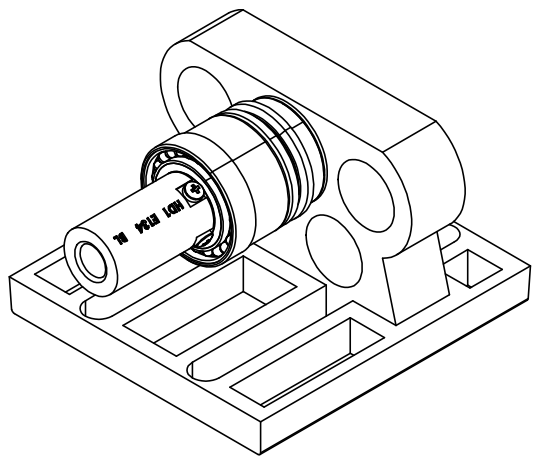
Plano:
Conjunto E3

Plano N°:
08

4 3 2 1

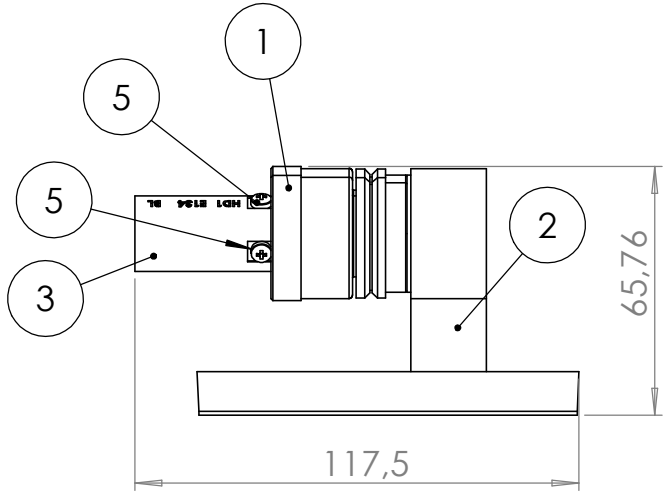
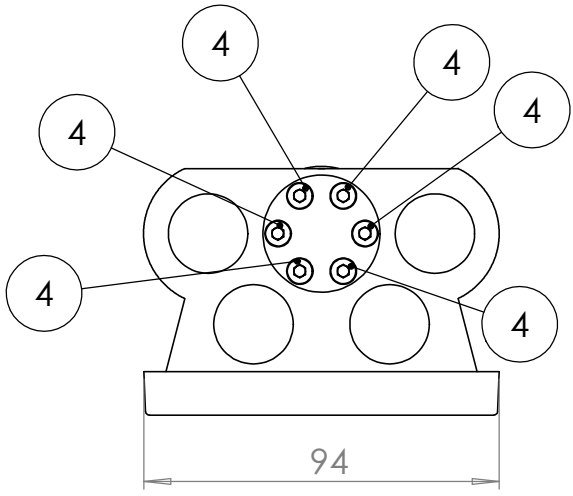
F

F



E

E



D

D

C

C

5	Tornillo de cabeza ranurada	2	Acero ISO 7045 M3x4
4	Tornillo de cabeza hex. hueca	6	Acero ISO 4762 M4x12
3	Eje E134: Plano N° 3	1	Nylon ϕ 20x41 mm
2	Base E4: Plano N° 10	1	Nylon 94x65x101 mm
1	HFUC_14_2UH simplificado	1	Wave gen. y Flex spline según esp.
Marca	Descripción	Cantidad	Material y dimensiones

B

B

PROYECTO: Modelado e identificación de una articulación con reductor Harmonic Drive.

Universitat Politècnica de València (UPV)
Escuela Técnica Superior de Ingeniería Aeroespacial y Diseño Industrial(ETSIADI)

Fecha: 25/5/2024

Escala
1:2

A

A

Autor:
Rafael Luis
Saneleuterio Temporal

Plano:
Conjunto E4

Plano N°:
09

4 3 2 1

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DESARROLLO DE EXPERIMENTOS PARA LA IDENTIFICACIÓN DE UNA ARTICULACIÓN CON HARMONIC DRIVE

3. Pliego de condiciones

TRABAJO FINAL DEL

Máster Universitario en Ingeniería Mecatrónica

REALIZADO POR

Rafael Luis Saneleuterio Temporal

TUTORIZADO POR

Ranko Zotovic Stanisic

CODIRECTOR:

Santiago Carlos Gutiérrez Rubert

CURSO ACADÉMICO: 2023/2024

Índice pliego de condiciones

1. Definición y alcance del pliego	145
2. Especificaciones técnicas	145
2.1. Condiciones de los materiales	145
2.1.1. Subsistema electrónico	145
2.1.2. Subsistema de programación	146
2.1.3. Subsistema mecánico	147
2.2. Condiciones de la ejecución	148
2.2.1. Conexiones eléctricas	148
2.2.2. Montaje de la mesa de experimentación	149
2.2.3. Adquisición de datos	150
2.2.4. Control de ejecución	150
2.3. Pruebas y ajustes de servicio	150
3. Normativa	151

Índice de tablas del pliego

Tabla 1: Lista de material electrónico	145
Tabla 2: Lista de material mecánico	147
Tabla 3: Conexiones de los componentes electrónicos	148

1. Definición y alcance del pliego

En el presente documento se especifican los materiales, servicios, ejecuciones, pruebas y ajustes para la correcta implementación de los experimentos para la identificación del harmonic drive. El ámbito de aplicación de este apartado se extiende a los subsistemas electrónico, de programación y mecánico que forman parte del proyecto.

Quedan excluidos todos los trabajos y condiciones en cuanto al modo de adquisición o compra de los elementos de las pruebas así como la configuración detallada del *software* la cual se manifiesta en la memoria del proyecto.

2. Especificaciones técnicas

2.1. Condiciones de los materiales

2.1.1. Subsistema electrónico

En la siguiente tabla (tabla 1) se listan los componentes electrónicos utilizados en todos los montajes experimentales junto con el modelo concreto y condiciones de utilización.

Componente	Cantidad	Modelo	Observaciones
Placa de desarrollo	1	STM32F429 Discovery kit DISC-1	Lógica a 3,3 V
Controlador Motor BLDC	1	ESCON 50/5	Máximo: 56 V, 5 A
Motor BLDC	1	Maxon 346970 EC 90 flat brushless, 60 W y encoder	Rango: 3 V - 6 V Máximo: 130 mA
Cabezal de encoder magnético	1	RLC2HD13BK00C00	Lógica a 5 V
Disco magnético axial incremental	1	MR050C040A072B00	Resolución: 72 x 8192 cuentas
Amplificador operacional	1	LM358	Dual AO
Resistencia 1 K Ω	2	1000 Ohmios E12	Error máximo: 5 %
Resistencia 470 Ω	3	470 Ohmios E12	Error máximo: 5 %
Resistencia 4.7 K Ω	1	4700 Ohmios E12	Error máximo: 5 %
Resistencia 22 K Ω	1	22000 Ohmios E12	Error máximo: 5 %
Resistencia 47 K Ω	1	47000 Ohmios E12	Error máximo: 5 %
Resistencia 10 K Ω	1	10000 Ohmios E12	Error máximo: 5 %
Resistencia 220 Ω	12	220 Ohmios E12	Error máximo: 5 %

Componente	Cantidad	Modelo	Observaciones
Placa de prototipos	3	<i>Protoboard</i> 170 puntos	Paso de <i>sockets</i> : 2,54 mm
PCB de prototipos	1	PCB de soldadura de 13 x 40 puntos	Paso de <i>sockets</i> : 2,54 mm
Cable macho a hembra	>13	Cables <i>jumpers</i> hembra/macho	Máximo: 3 A De 10 cm a 20 cm
Cable macho a macho	>16	Cables <i>jumpers</i> macho/macho	Máximo: 3 A De 10 cm a 20 cm
Cable potencia motor BLDC	1	Cable 275934 Maxon	Máximo: 15 A
Cable USB Datos	1	Cable de USB a mini USB	Transmisión de carga y datos
Cable USB Driver	1	Cable de USB a micro USB	Configuración
Controlador servomotor y cables de potencia	1	Digitax HD M751 01400042A10101AB110	Alimentación de potencia: trifásica
Teclado Remoto	1	Remote-Keypad RTC	LDC keypad option
Cable ethernet	1	Cat 5E	Longitud: 0.5 m
Servomotor	1	Nidec 095U3C300BACAA100190	Par nominal: 5.6 Nm
Fuente DC del driver del motor BLDC	1	QS20	ESCON: 36 v
Fuente DC de la amplificación	1	Ruzizao	Vcc del AO: 36 V
Fuente DC del driver del servomotor	1	GVD305SF	Parte lógica del driver: 24 V
Interruptor	1	Pulsador enclavado	Estados: 2
Alambre de soldar	10 g	Alambre de estaño de 0,8 mm de grosor	Sn mínimo: 63 %

Tabla 1: Lista de material electrónico

2.1.2. Subsistema de programación

Se requiere de una computadora con puerto USB. En concreto, se han adquirido los datos en un portátil con un procesador Ryzen 9 (5000 series) con 16 GB de memoria RAM, y Windows 10 Pro. La tarjeta gráfica que se ha explotado es la Geforce RTX 3050 Ti.

2.1.3. Subsistema mecánico

En la tabla 2 se listan los materiales utilizados para el subsistema mecánico junto con sus cantidades.

Componente	Cantidad	Detalle
Harmonic Drive	1	HFUC-14-2UH (ratio de 30 y lubricación grasa)
Nylon	200 g	200 g para impresión SLS
Tornillo 1	2	Norelem M10 x 45, con 27 mm de rosca
Arandela 1	4	Norelem M10 x 1.5, D22 mm
Tornillo 2	4	Norelem M10 x 35
Arandela 2	2	Norelem M12 x 4, D21 mm
Tornillo 3	2	Norelem M10 x 40, con 26 mm de rosca
Tornillo 4	2	Norelem M10 x 80, con 27 mm de rosca
Arandela apoyo	2	Norelem NLM 08910
Tuerca	2	Norelem M10 x 6, D117 mm
Tornillo 5	2	Norelem M10 x 65, con 33 mm de rosca
Piezas sujeción	2	Norelem NLM 2III 010 005
Piezas apoyo	4	Norelem NLM 22I6 010 006
Escuadra auxiliar	1	Acero 3 mm, 90 x 105 x 105 mm
Lego y Meccano	1	Estructura de 55 mm de alto
Lámina madera	1	Madera 3 mm, 60 x 80 mm
Papel de aluminio	20	Aluminio 0.2 mm, 60 x 60 mm
Tapa de cartón	1	200 x 160 mm
Tornillo cabeza hex. hueca	12	Acero ISO 4762 M4x12
Tornillo de cabeza ranurada	4	Acero ISO 7045 M3x4
Mesa de trabajo	1	Norelem Modulaire 2004 010 012 1908

Tabla 2: Lista de material mecánico

2.2. Condiciones de la ejecución

2.2.1. Conexiones eléctricas

Para los experimentos E1 y E1' las conexiones son las siguientes:

Interconexiones		Tipo de unión
Fuente DC Ruzizao: GND	GND	Cable M/M
Fuente DC Ruzizao: 36V	Circuito amplificador: +Vcc	Cable M/M
Fuente DC Ruzizao: GND	Circuito amplificador: GND	Cable M/M
STM32F429: PA4	Circuito amplificador: Entrada	Cable H/M
STM32F429: USB ST-LINK	Computadora	USB Mini
STM32F429: PD12	DT encoder motor: Canal A (3.3V)	Cable H/M
STM32F429: PD13	DT encoder motor: Canal B (3.3V)	Cable H/M
STM32F429: PB3	DT encoder mag.: Canal A (3.3V)	Cable H/M
STM32F429: PA5	DT encoder mag.: Canal B (3.3V)	Cable H/M
STM32F429: 5 V	Interruptor: Borne 1	Cable H/M
STM32F429: GND	GND	Cable H/M
STM32F429: PC5	ESCON 50/5: J6, Pin 5	Cable H/M
Computadora	ESCON 50/5: J7, USB	USB Micro
Encoder motor: Canal A	ESCON 50/5: J4, Pin 6	Cable M/H
Encoder motor: Canal B	ESCON 50/5: J4, Pin 8	Cable M/H
Encoder motor: Vcc	ESCON 50/5: J5, Pin 6	Cable M/M
Encoder motor: GND	ESCON 50/5: J5, Pin 5	Cable M/M
Encoder magnético: +Vcc	ESCON 50/5: J5, Pin 6	Cable M/M
Encoder magnético: GND	ESCON 50/5: J5, Pin 5	Cable M/M
Interruptor: Borne 2	ESCON 50/5: J5, Pin 2	Cable M/M
Fuente DC PLUS: 36V	ESCON 50/5: J1, +Vcc	Cable BT
Fuente DC PLUS: GND	ESCON 50/5: J1, GND	Cable BT
GND	ESCON 50/5: J5, Pin 5	Cable M/M
Cable 275934: Apantallamiento	ESCON 50/5: J3, Pin 5	Cable 275934
Cable 275934: Apantallamiento	ESCON 50/5: J2, Pin 4	Cable 275934

Interconexiones		Tipo de unión
Motor BLDC: pin 6	ESCON 50/5: J3, Pin 5	Cable 275934
Motor BLDC: pin 3	ESCON 50/5: J3, Pin 4	Cable 275934
Motor BLDC: pin 5	ESCON 50/5: J3, Pin 3	Cable 275934
Motor BLDC: pin 2	ESCON 50/5: J3, Pin 2	Cable 275934
Motor BLDC: pin 1	ESCON 50/5: J3, Pin 1	Cable 275934
Motor BLDC: pin 4	ESCON 50/5: J2, Pin 3	Cable 275934
Motor BLDC: pin 8	ESCON 50/5: J2, Pin 2	Cable 275934
Motor BLDC: pin 7	ESCON 50/5: J2, Pin 1	Cable 275934
Circuito amplificador: Salida	ESCON 50/5: J6, Pin 3	Cable M/M
Circuito amplificador: 10V cte.	ESCON 50/5: J6, Pin 4	Cable M/M
Encoder motor: Canal A	DT encoder motor: Canal A (5V)	Cable H/M
Encoder motor: Canal B	DT encoder motor: Canal B (5V)	Cable H/M
Encoder magnético: Canal A	DT encoder mag.: Canal A (5V)	Soldadura PCB
Encoder magnético: Canal B	DT encoder mag.: Canal B (5V)	Soldadura PCB

Tabla 3: Conexiones de los componentes electrónicos

En resto de experimentos siguen la especificación anterior excepto en ciertos casos en los que se tienen unas conexiones particularmente condicionadas que se explican a continuación:

- En E2' se da uso del controlador del servomotor Digitax HD en lugar del ESCON 50/5 y cuya etapa de potencia está alimentada a través de la red trifásica del laboratorio (etapa de control a 24 V). A este se le transfiere la señal de referencia amplificada al pin 10 y la referencia fija al pin 9 (la señal diferencial debe reducirse a un rango de 13 V). A través de los cables de potencia y feedback, se conecta el controlador al servomotor. Es necesario hacer un puente de las entradas digitales de habilitación y marcha para el funcionamiento del motor. Para configurar los parámetros se utiliza el Remote-Keypad RTC conectado por cable ethernet.
- En E3' se combinan las conexiones de E1 y E2' a excepción de la referencia del Digitax HD, que es un valor constante de 5 V.
- En E4 las conexiones son análogas a E1 a excepción de que no se hace uso del encoder magnético.

2.2.2. Montaje de la mesa de experimentación

Paso 1: Encajar sobre la mesa de ensayos las piezas de apoyo de Norelem en los huecos 3C, 3D, 3E y 3F en forma de "u" y atornillar las bases con 2 tornillos del número 3.

Paso 2: Encajar el eje E134 al eje del motor BLDC (ya atornillado a la escuadra) y luego al wave generator mediante los 2 tornillos ranurados.

Paso 3: Encajar el circular spline a la base E123 (flexspline a la base E4 en el caso de E4) y colocar 6 tornillos de cabezas hexagonales.

Paso 4: Encajar el wave generator en el circular spline y depositar el ensamblaje actual entre los soportes del norelem con el papel de aluminio dejando la lámina de madera debajo de la escuadra hasta que esté alineado. En el caso de E1, no colocar el papel de aluminio.

Paso 5: Afianzar la escuadra del motor con las piezas de sujeción de Norelem y los tornillos y arandelas número 1 en las posiciones 1F y 1C.

Paso 6: Afianzar la respectiva base con los soportes de Norelem mediante los tornillos y arandelas número 2.

Paso 7: Asegurar los soportes de Norelem mediante tornillos del número 4 y la arandela de apoyo.

Paso 8: En el caso de E1 y E1', montar el soporte de encoder E1 sobre la salida del reductor mediante tres tornillos de cabeza hexagonal y luego fijar el disco magnético del encoder. En el caso de E2' y E3', encajar primero el disco magnético en el eje E23 y luego introducir este en el eje del servomotor. Luego colocar 2 soportes de Norelem de forma análoga al paso 1 pero en la columna 9. Seguidamente, depositar el servomotor en los soportes sobre papel de aluminio hasta conseguir la alineación correcta. Encajar con la salida del reductor y colocar 6 tornillos de cabeza hexagonal. Finalmente, afianzar los soportes con dos tornillos del número 5.

Paso 9: A excepción del E4, colocar la estructura de Lego y Mecano necesaria para conseguir una altura de 55 mm y luego atornillar en la parte superior la PCB sobre la que se suelda el lector del encoder magnético.

Paso 10: Aproximar la estructura con el lector entre 0.3 mm a 1 mm del disco magnético y fijarla en ese punto mediante 2 tornillos de Norelem número 5, tuercas y arandelas del número 2.

2.2.3. Adquisición de datos

Se ha de conectar la placa de desarrollo a través del cable mini USB al ordenador. Seguidamente, se ha de reiniciar la placa mientras se abre el programa del PuTTY dos veces. En la segunda, se abre el terminal con la configuración pertinente y se inicia el programa de la placa activando el interruptor de habilitación del motor. Cuando se hayan recolectado los datos, se cierra el PuTTY y se apaga el interruptor.

2.2.4. Control de ejecución

A través del programa de ESCON Studio se puede comprobar que la corriente del motor sigue la referencia en tiempo de ejecución.

2.3. Pruebas y ajustes de servicio

Al obtenerse los datos, en el posproceso de Matlab se ha de comprobar la ausencia de errores en las matrices de datos recibidas. Finalmente, se comprueba que, tras numerosas vueltas en un largo periodo de tiempo, la ejecución en segundos coincide con el número de

muestreos multiplicado por 0.00015 y que el número de vueltas detectadas coinciden con las reales.

Limpiar con alcohol las piezas de Norelem antes de usarlas y engrasarlas una vez terminadas las tareas.

3. Normativa

Normativa sobre instrumentación y medición:

- ISO/IEC 17025: Estándar para la competencia de laboratorios de ensayo y calibración, relevante para asegurar la precisión de las mediciones.
- Normas IEC 61557: Estándares para equipos de medición eléctrica.

Normativa sobre componentes electrónicos y sistemas de control:

- IEC 61131: Normas para controladores programables industriales.
- Normas de la IEEE (Institute of Electrical and Electronics Engineers): Relevantes para el diseño y desarrollo de sistemas electrónicos y de control.

Normativa específicas para el sector robótico:

- ISO/TS 15066: Especificación técnica para robots colaborativos, particularmente relevante para ensayos relacionados con la reversibilidad y la interacción segura con humanos.

Normativa de compatibilidad electromagnética:

- Directiva 2014/30/UE de Compatibilidad Electromagnética: Asegura que los equipos eléctricos y electrónicos no causen ni se vean afectados por interferencias electromagnéticas.

Normativa de seguridad:

- Directiva 2006/42/CE de Máquinas: Regula los requisitos esenciales de seguridad y salud para el diseño y fabricación de máquinas.
- Reglamento (UE) 2016/425 sobre Equipos de Protección Individual: Establece las normas para los equipos de protección personal que se puedan necesitar durante la experimentación.
- Directiva 2014/35/UE de Baja Tensión: Regula la seguridad de los equipos eléctricos dentro de ciertos rangos de voltaje.
- UNE-EN ISO 12100:2012 Seguridad Máquinas: Evaluación Riesgos
- UNE-EN 60204-1:2007 Seguridad Máquinas. Equipo Eléctrico. Requisitos Generales
- Normativa de seguridad de laboratorios V.5E.0.013 y V.5E.0.066 de la Universitat Politècnica de València, campus de Vera.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DESARROLLO DE EXPERIMENTOS PARA LA IDENTIFICACIÓN DE UNA ARTICULACIÓN CON HARMONIC DRIVE

4. Presupuesto

TRABAJO FINAL DEL

Máster Universitario en Ingeniería Mecatrónica

REALIZADO POR

Rafael Luis Saneleuterio Temporal

TUTORIZADO POR

Ranko Zotovic Stanisic

CODIRECTOR:

Santiago Carlos Gutiérrez Rubert

CURSO ACADÉMICO: 2023/2024

Índice presupuesto

1. Cuadro de precios elementales	154
2. Cuadro de precios descompuestos	156
3. Valoración	159

Índice de tablas del presupuesto

Tabla 1: Precios elementales de los materiales	154
Tabla 2: Precios elementales de mano de obra, equipo, <i>software</i> y medios auxiliares	155
Tabla 3: Precios descompuestos del subsistema electrónico	156
Tabla 4: Precios descompuestos de los subsistemas de programación y control	157
Tabla 5: Precios descompuestos del subsistema mecánico	158
Tabla 6: Valoración final del proyecto	159

1. Cuadro de precios elementales

Para obtener el listado de todos los elementos que intervienen en el presupuesto del desarrollo de los experimentos, se ha optado por el método de precios descompuestos. Es por ello que, primero conviene inventariar todos los costes de material (tabla 1), mano de obra, equipos y *software* (tabla 2). En este apartado solo se tienen en cuenta los precios elementales, es decir, el coste unitario.

1. Cuadro de precios elementales I			
Ref.	Unidad	Descripción	Precio (€)
Materiales			
m1	ud.	Placa de desarrollo, STM32F429 Discovery kit DISC-1	28,63
m2	ud.	Controlador Motor BLDC, ESCON 50/5	362,50
m3	ud.	Motor BLDC, Maxon EC 90 flat brushless, 60 W y encoder	109,81
m4	ud.	Cabezal de encoder magnético, RLC2HD13BK00C00	24,79
m5	ud.	Disco magnético axial incremental, MR050C040A072B00	46,93
m6	ud.	Amplificador operacional, LM358	0,18
m7	ud.	Resistencias, E12	0,02
m8	ud.	Placa de prototipos, protoboard 170 puntos	0,99
m9	ud.	PCB de prototipos, soldadura de 13 x 40 puntos	0,70
m10	ud.	Cables jumpers	0,06
m11	ud.	Cable potencia motor BLDC, Cable 275934 Maxon	38,55
m12	ud.	Cable USB Datos, mini USB	3,15
m13	ud.	Cable USB Driver, Cable de USB a micro USB	2,88
m14	ud.	Controlador servomotor y cables de potencia, Digitax HD M751	809,59
m15	ud.	Remote-Keypad RTC	79,00
m16	ud.	Cable ethernet, Cat 5E	2,11
m17	ud.	Servomotor, Nidec 095U3C300BACAA100190	1509,45
m18	ud.	Fuente DC, Ruzizao	37,91
m19	ud.	Interruptor, pulsador enclavado	0,79
m20	g	Alambre de estaño de 0,8 mm de grosor	0,05
m21	ud.	Harmonic Drive, HFUC-14-2UH (ratio de 30 y lubricación grasa)	1106,00
m22	g	Nylon, para impresión SLS	0,03
m23	ud.	Escuadra auxiliar, Acero 3 mm, 90 x 105 x 105 mm	12,16
m24	g	Lego y Meccano, Estructura de 55 mm de alto	0,05
m25	ud.	Tornillo cabeza hex. hueca, Acero ISO 4762 M4x12	0,06
m26	ud.	Tornillo de cabeza ranurada, Acero ISO 7045 M3x4	0,35
m27	ud.	Lámina madera, 3 x 60 x 80 mm	0,79
m28	ud.	Papel de aluminio 0.2 mm, 60 x 60 mm	0,40

Tabla 1: Precios elementales de los materiales

1. Cuadro de precios elementales II			
Ref.	Unidad	Descripción	Precio (€)
<u>Mano de obra</u>			
h1	h.	Ingeniero desarrollador *[1]	25
<u>Equipos</u>			
ei1	h.	Impresión 3D SLS	7,90
ei2	h.	Estación soldadora	0,02
ei3	h.	Fuente DC, QS20	0,01
ei4	h.	Fuente DC, GVD305SF	0,01
ei5	h.	Material Norelem	0,01
ei6	h.	Ordenador	0,32
<u>Software</u>			
sw1	lic. mensual	STM32CubeIDE	0,00
sw2	lic. mensual	ESCON Studio	0,00
sw3	lic. mensual	PuTTY	0,00
sw4	lic. mensual	Matlab	0,00
sw5	lic. mensual	Solidworks	0,00
<u>Medios auxiliares</u>			
	%	Medios auxiliares sobre costes directos *[2]	10%

Tabla 2: Precios elementales de mano de obra, equipo, *software* y medios auxiliares

*[1] El coste del ingeniero desarrollador se ha extrapolado del salario medio de los ingenieros industriales según las estadísticas de Jobted, [en línea]. Disponible en: <https://www.jobted.es/salario/ingeniero-industrial>

*[2] Los medios auxiliares hacen referencia a aquellos costes de difícil cuantificación incluyendo conceptos como las pequeñas herramientas, mermas de material, consumo eléctrico, etc. Estos gastos se integran en el presupuesto como un único porcentaje característico en proyectos de ingeniería (10 %) sobre el subtotal de materiales, mano de obra, equipos y *software*.

2. Cuadro de precios descompuestos

A continuación, se desglosa el proyecto en tres principales componentes que vendrían a ser el subsistema electrónico, el de programación junto con el de control, y el mecánico. De manera que, se listan y cuantifican los materiales, equipos, mano de obra y *software* que intervienen en cada uno de los subsistemas, como se puede ver en las tablas 3, 4 y 5.

2. Cuadro de precios descompuestos I					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
d1	ud.	Subsistema electrónico formado por los componentes homónimos y el uso del material del laboratorio incluyendo mano de obra			
Materiales					
m1	ud.	Placa de desarrollo, STM32F429 Discovery kit DISC-1	28,63	1	28,63
m2	ud.	Controlador Motor BLDC, ESCON 50/5	362,50	1	362,50
m3	ud.	Motor BLDC, Maxon EC 90 flat brushless, 60 W y encoder	109,81	1	109,81
m4	ud.	Cabezal de encoder magnético, RLC2HD13BK00C00	24,79	1	24,79
m5	ud.	Disco magnético axial incremental, MR050C040A072B00	46,93	1	46,93
m6	ud.	Amplificador operacional, LM358	0,18	1	0,18
m7	ud.	Resistencias, E12	0,02	22	0,41
m8	ud.	Placa de prototipos, protoboard 170 puntos	0,99	3	2,96
m9	ud.	PCB de prototipos, soldadura de 13 x 40 puntos	0,70	1	0,70
m10	ud.	Cables jumpers	0,06	30	1,65
m11	ud.	Cable potencia motor BLDC, Cable 275934 Maxon	38,55	1	38,55
m12	ud.	Cable USB Datos, mini USB	3,15	1	3,15
m13	ud.	Cable USB Driver, Cable de USB a micro USB	2,88	1	2,88
m14	ud.	Controlador servomotor y cables de potencia, Digitax HD M751	809,59	1	809,59
m15	ud.	Remote-Keypad RTC	79,00	1	79,00
m16	ud.	Cable ethernet, Cat 5E	2,11	1	2,11
m17	ud.	Servomotor, Nidec 095U3C300BACAA100190	1509,45	1	1509,45
m18	ud.	Fuente DC, Ruzizao	37,91	1	37,91
m19	ud.	Interruptor, pulsador enclavado	0,79	1	0,79
m20	g	Alambre de estaño de 0,8 mm de grosor	0,05	10	0,46
Mano de obra					
h1	h.	Ingeniero desarrollador	25	120	3000,00

2. Cuadro de precios descompuestos I					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
<u>Equipos</u>					
ei2	h.	Estación soldadora	0,02	1	0,02
ei3	h.	Fuente DC, QS20	0,01	336	2,65
ei4	h.	Fuente DC, GVD305SF	0,01	336	2,65
ei6	h.	Ordenador	0,32	120	37,92
<u>Medios auxiliares</u>					
	%	Medios auxiliares sobre costes directos	10%	6077,09	607,71
Precio de desarrollo y ejecución material					6713,43

Tabla 3: Precios descompuestos del subsistema electrónico

2. Cuadro de precios descompuestos II					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
d2	ud.	Subsistemas de programación y control formado por la configuración en los diferentes programas incluyendo mano de obra			
<u>Mano de obra</u>					
h1	h.	Ingeniero desarrollador	25	110	2750,00
<u>Equipos</u>					
ei6	h.	Ordenador	0,32	110	34,76
<u>Software</u>					
sw1	lic. mensual	STM32CubeIDE	0,00	4	0,00
sw2	lic. mensual	ESCON Studio	0,00	3	0,00
sw3	lic. mensual	PuTTY	0,00	3	0,00
sw4	lic. mensual	Matlab	0,00	3	0,00
<u>Medios auxiliares</u>					
	%	Medios auxiliares sobre costes directos	10%	2784,76	278,48
Precio de desarrollo y ejecución material					3063,24

Tabla 4: Precios descompuestos de los subsistemas de programación y control

2. Cuadro de precios descompuestos III					
Ref.	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
d3	ud.	Subsistema mecánico formado por el diseño asistido por ordenador (CAD), la fabricación de las piezas, el ensamblaje y el uso de material del laboratorio incluyendo mano de obra			
<u>Materiales</u>					
m21	ud.	Harmonic Drive, HFUC-14-2UH (ratio de 30 y lubricación grasa)	1106,00	1	1106,00
m22	g	Nylon, para impresión SLS	0,03	200	6,30
m23	ud.	Escuadra auxiliar, Acero 3 mm, 90 x 105 x 105 mm	12,16	1	12,16
m24	g	Legó y Meccano, Estructura de 55 mm de alto	0,05	50	2,32
m25	ud.	Tornillo cabeza hex. hueca, Acero ISO 4762 M4x12	0,06	12	0,72
m26	ud.	Tornillo de cabeza ranurada, Acero ISO 7045 M3x4	0,35	4	1,41
m27	ud.	Lámina madera, 3 x 60 x 80 mm	0,79	1	0,79
m28	ud.	Papel de aluminio 0.2 mm, 60 x 60 mm	0,40	20	7,90
<u>Mano de obra</u>					
h1	h.	Ingeniero desarrollador	25	70	1750,00
<u>Equipos</u>					
ei1	h.	Impresión 3D SLS	7,90	12	94,80
ei5	h.	Material Norelem	0,01	1	0,01
ei6	h.	Ordenador	0,32	70	22,12
<u>Software</u>					
sw4	lic. mensual	Programa Solidworks	0,00	2	0,00
<u>Medios auxiliares</u>					
	%	Medios auxiliares sobre costes directos	10%	3004,53	300,45
Precio de desarrollo y ejecución material					3304,99

Tabla 5: Precios descompuestos del subsistema mecánico

3. Valoración

El resumen del presupuesto lleva a la suma de todos los costes descompuestos (tabla 6). Finalmente, la valoración del proyecto Desarrollo de experimentos para la identificación de una articulación con Harmonic Drive, asciende a 6 908,02 € sin IVA y 8 358,71 € con IVA.

3. Valoración					
Ref	Unidad	Descripción	Precio (€)	Cantidad	Parcial (€)
d1	ud.	Subsistema electrónico formado por los componentes homónimos y el uso del material del laboratorio incluyendo mano de obra	6713,43	1	6713,43
d2	ud.	Subsistemas de programación y control formado por la configuración en los diferentes programas incluyendo mano de obra	3063,24	1	3063,24
d3	ud.	Subsistema mecánico formado por el diseño asistido por ordenador (CAD), la fabricación de las piezas, el ensamblaje y el uso de material del laboratorio incluyendo mano de obra	3304,99	1	3304,99
Total presupuesto de desarrollo y ejecución material antes de impuestos					13081,65
Impuestos					
	%	IVA	21%	13081,65	2747,15
Total presupuesto del proyecto					15828,80

Tabla 6: Valoración final del proyecto