



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de un sistema de control de inversor para máquina
de corriente continua sin escobillas de 200W.

Trabajo Fin de Grado

Grado en Ingeniería en Tecnologías Industriales

AUTOR/A: Rodríguez Bascuñana, David

Tutor/a: Martínez Román, Javier Andrés

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

ÍNDICE DE DOCUMENTOS

MEMORIA

PRESUPUESTO

PLANOS

ANEXOS



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Diseño de un sistema de control de inversor para máquina de corriente continua sin escobillas de 200W

MEMORIA

Trabajo Final de Grado
DAVID RODRÍGUEZ BASCUÑANA
Junio, 2022

RESUMEN

En la actualidad el uso de motores eléctricos de corriente continua sin escobillas (Brushless DC, BLDC) está generalizado tanto en elementos de uso cotidiano como en la industria. Por ello es una tarea fundamental el automatizar el funcionamiento de estos motores.

El presente trabajo se centra en el desarrollo de un control de velocidad para un motor de este tipo. El control deberá ser seguro, impidiendo alcanzar condiciones de funcionamiento inadmisibles tanto para el motor como para los elementos que compongan el sistema de control. Además, se contempla la implementación de una interfaz de usuario compatible con dispositivos móviles Android mediante la cual se pueda regular de manera inalámbrica mediante Bluetooth la velocidad del motor, así como obtener información relevante del sistema de control en tiempo real.

El sistema de control planteado se llevará cabo mediante un inversor y está basado en un microcontrolador tipo ESP32, sobre el que recaerán las tareas de control y comunicación con la interfaz de usuario. Para la operación del motor se empleará la modulación de tensión de las fases del mismo mediante el método de corriente rectangular típico de este tipo de máquinas junto con un algoritmo de modulación por ancho de pulso (PWM) por portadora triangular y control independiente de la parte positiva y negativa de cada rama del inversor.

Aparte de inversor y microcontrolador, serán necesarios otros elementos, tales como sensores de corriente o posición. El proyecto contempla la identificación de este tipo de elementos y su montaje mecánico y eléctrico en el sistema.

Palabras Clave: Control, Inversor, máquina de corriente continua sin escobillas, ESP32, aplicación Android

RESUM

En l'actualitat l'ús de motors elèctrics de corrent continu sense escobretes (Brushless DC, BLDC) està generalitzat tant en elements d'ús quotidià com en la indústria. Per això és una tasca fonamental l'automatitzar el funcionament d'aquests motors.

El present treball se centra en el desenvolupament d'un control de velocitat per a un motor d'aquest tipus. El control haurà de ser segur, per això no hauran permetre's condicions de funcionament inadmissibles tant per al motor com per als elements que formen el sistema de control. A més, es contempla la implementació d'una interfície d'usuari compatible amb dispositius mòbils Android mitjançant la qual es puga regular de manera sense fil mitjançant Bluetooth la velocitat del motor, així com obtindre informació rellevant del sistema de control a temps real.

Per al sistema de control plantejat s'usarà un inversor i està basat en un microcontrolador tipus ESP32, sobre el qual recauran les tasques de control i comunicació amb la interfície d'usuari. Per a l'operació del motor s'emprarà la modulació de tensió de les fases del mateix mitjançant el mètode de corrent rectangular típic d'aquesta mena de màquines juntament amb un algorisme de modulació per ample de pols (PWM) per portadora triangular i control independent de la part positiva i negativa de cada branca de l'inversor.

A part d'inversor i microcontrolador, seran necessaris altres elements, com ara sensors de corrent o posició. El projecte contempla la identificació d'aquesta mena d'elements i el seu muntatge mecànic i elèctric en el sistema.

Paraules clau: Control, Inversor, màquina de corrent continu sense escobretes, ESP32, aplicació Android

ABSTRACT

Nowadays, the use of brushless DC electric motors (BLDC) is widespread both in everyday items and in industry. For this reason, it is a fundamental task to automate the operation of these motors.

The present work focuses on the development of a speed control for a motor of this type. The control must be safe, preventing inadmissible operating conditions from being reached both for the motor and for the elements that make up the control system. In addition, the implementation of a human machine interface compatible with Android mobile devices is contemplated. This interface enables the motor speed to be regulated wirelessly via Bluetooth. Furthermore, it allows to obtain relevant information from the control system in real time.

The proposed control system will use an inverter and is based on an ESP32-type microcontroller. This microcontroller will be the responsible of the control and communication tasks with the user interface. For the operation of the motor it will be used the tension modulation of the phases of the motor with the rectangular current method typical of this kind of machines together with a pulse width modulation (PWM) algorithm by triangular carrier and independent control of the positive and negative branch of the inverter.

Apart from inverter and microcontroller, other elements will be necessary, such as current or position sensors. The project contemplates the identification of this type of elements and their mechanical and electrical assembly in the system.

Keywords: Control, Inverter, brushless DC motor, ESP32, Android application

ÍNDICE DE CONTENIDOS

| | | |
|-------------|--|----|
| CAPÍTULO 1: | INTRODUCCIÓN | 1 |
| 1.1 | ANTECEDENTES | 2 |
| 1.2 | OBJETIVOS | 4 |
| 1.3 | DESCRIPCIÓN Y ESTRUCTURA DE LA MEMORIA..... | 5 |
| CAPÍTULO 2: | CONTROL DEL MOTOR Y DESARROLLO EN SIMULINK..... | 6 |
| 2.1 | INTRODUCCIÓN | 6 |
| 2.2 | ESTRUCTURA | 6 |
| 2.3 | OBJETIVO DEL CONTROL | 6 |
| 2.4 | MODO DE CONTROL DEL MOTOR | 7 |
| 2.5 | ENSAYOS EN SIMULINK | 9 |
| 2.5.1 | Primer ensayo: f.e.m. inducida..... | 10 |
| 2.5.2 | Segundo ensayo: Corrientes en las fases a velocidad constante | 11 |
| 2.5.3 | Tercer ensayo: Alimentación de tensión con PWM | 13 |
| 2.5.4 | Cuarto ensayo: PWM con control proporcional..... | 14 |
| 2.5.5 | Quinto ensayo: Ajuste de corriente con control PI | 16 |
| 2.5.6 | Sexto ensayo: Ajuste independiente de velocidad con control PI | 18 |
| 2.5.7 | séptimo ensayo: Control en cascada..... | 20 |
| 2.5.8 | Octavo ensayo: Realimentación de corriente | 22 |
| 2.5.9 | Noveno ensayo: Comprobaciones finales | 25 |
| 2.6 | DESCRIPCION DEL SISTEMA REAL..... | 29 |
| CAPÍTULO 3: | PLACA DEL MICROCONTROLADOR..... | 31 |
| 3.1 | INTRODUCCIÓN | 31 |
| 3.2 | ESTRUCTURA | 31 |
| 3.3 | DESCRIPCIÓN DEL MICROCONTROLADOR | 31 |
| 3.4 | SENSORES DE CORRIENTE | 33 |
| 3.5 | SEÑALES DE CONTROL DE INVERSOR | 38 |
| 3.6 | CONEXIONES DE POTENCIA..... | 40 |
| 3.7 | MONTAJE DE LA PLACA DEL MICROCONTROLADOR..... | 41 |
| CAPÍTULO 4: | MEDIDAS Y MONTAJE DEL SENSOR DE POSICIÓN..... | 45 |
| 4.1 | INTRODUCCIÓN | 45 |

| | | |
|--|---|----|
| 4.2 | ESTRUCTURA | 45 |
| 4.3 | SENSOR DE POSICIÓN..... | 45 |
| 4.4 | MONTAJE MECÁNICO DEL ENCODER | 48 |
| 4.5 | LECTURA DE POSICIÓN EN EL MICROCONTROLADOR..... | 50 |
| 4.6 | OBTENCIÓN DEL SECTOR..... | 51 |
| 4.6.1 | Ensayos previos | 51 |
| 4.6.2 | implementación en el microcontrolador | 54 |
| 4.7 | MEDIDA DE VELOCIDAD EN EL MICROCONTROLADOR..... | 54 |
| CAPÍTULO 5: FUNCIONES DEL MICROCONTROLADOR | | 59 |
| 5.1 | INTRODUCCIÓN | 59 |
| 5.2 | ESTRUCTURA | 59 |
| 5.3 | REALIMENTACIÓN DE CORRIENTE Y VELOCIDAD | 59 |
| 5.4 | GENERACIÓN DE SEÑALES PWM Y HABILITACIÓN..... | 61 |
| 5.5 | BUCLE DE CONTROL | 62 |
| 5.5.1 | Lazo de corriente..... | 63 |
| 5.5.2 | Lazo de velocidad y control en cascada | 67 |
| CAPÍTULO 6: INTERFAZ DE USUARIO | | 72 |
| 6.1 | INTRODUCCIÓN | 72 |
| 6.2 | ESTRUCTURA | 72 |
| 6.3 | COMUNICACIÓN ENTRE DISPOSITIVOS..... | 72 |
| 6.4 | IMPLEMENTACIÓN DE LA COMUNICACIÓN | 73 |
| 6.4.1 | Implementación en el microcontrolador | 73 |
| 6.4.2 | Implementación en el dispositivo móvil..... | 75 |
| CAPÍTULO 7: PRUEBAS DE FUNCIONAMIENTO y CONCLUSIONES | | 81 |
| 7.1 | INTRODUCCIÓN | 81 |
| 7.2 | ESTRUCTURA | 81 |
| 7.3 | SECUENCIA DE FASES..... | 81 |
| 7.4 | CONTROL DE LA VELOCIDAD | 83 |
| 7.5 | CONCLUSIONES | 83 |
| REFERENCIAS..... | | 85 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 2.1 Elemento diferencial de capa de corriente en la cara interior del estator de una máquina de estator cilíndrico..... | 8 |
| Figura 2.2 Modelo Simulink primer ensayo | 10 |
| Figura 2.3 f.e.m. inducidas en las fases del motor BLDC y cambios de sector..... | 10 |
| Figura 2.4 Secuencia de corrientes alimentado el motor mediante inversor..... | 12 |
| Figura 2.5 Modelo Simulink segundo ensayo..... | 12 |
| Figura 2.6 Par del motor alimentando mediante inversor..... | 13 |
| Figura 2.7 Modelo Simulink tercer ensayo..... | 14 |
| Figura 2.8 Estabilización de las corrientes en las fases del motor ante un salto de duty cycle | 14 |
| Figura 2.9 Modelo Simulink cuarto ensayo | 15 |
| Figura 2.10 control de la corriente mediante PWM con acción proporcional..... | 16 |
| Figura 2.11 Modelo Simulink quinto ensayo..... | 17 |
| Figura 2.12 Corrientes en las fases con regulación PI y consigna 1 A..... | 18 |
| Figura 2.13 Modelo Simulink sexto ensayo..... | 19 |
| Figura 2.14 Velocidad y corrientes con control PI y salto de consigna de 0 a 1000 rpm | 20 |
| Figura 2.15 Modelo Simulink séptimo ensayo | 21 |
| Figura 2.16 Control en cascada según el modelo del séptimo ensayo | 22 |
| Figura 2.17 Control de la velocidad en cascada según el modelo del octavo ensayo | 23 |
| Figura 2.18 Ajuste PI velocidad, $K_p = 0,5$, $T_i = \infty$ | 23 |
| Figura 2.19 Ajuste PI velocidad, $K_p = 1$, $T_i = \infty$ | 24 |
| Figura 2.20 Ajuste PI velocidad, $K_p = 2$, $T_i = \infty$ | 24 |
| Figura 2.21 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.1$ | 24 |
| Figura 2.22 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.01$ | 25 |
| Figura 2.23 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.005$ | 25 |
| Figura 2.24 Resultados del control final en Simulink | 26 |
| Figura 2.25 Esquema de reguladores PI en Simulink | 27 |
| Figura 2.26 Esquema de inversor y realimentación de corriente en Simulink..... | 28 |
| Figura 2.27 Esquema de realimentación de velocidad, posición y sector | 29 |
| Figura 2.28 Esquema del sistema para el control de velocidad del motor BLDC (modificado de [3]) .. | 29 |
| Figura 3.1 Esquema y Pinout del microcontrolador ESP32 DEVKIT V1 [7] | 33 |

| | |
|---|----|
| Figura 3.2 función de transferencia corriente/voltaje del sensor de corriente LTSR 6-NP [8] | 34 |
| Figura 3.3 función de transferencia experimental para la lectura analógica con ESP32 [9]..... | 35 |
| Figura 3.4 relación corriente/dato digital del sensor para fase A..... | 36 |
| Figura 3.5 relación corriente/dato digital del sensor para fase B | 36 |
| Figura 3.6 Comparación entre funciones de transferencia experimental y teórica | 37 |
| Figura 3.7 Relación tensión/corriente en las fases del motor | 37 |
| Figura 3.8 Ejemplo de señales aportadas al inversor en el sector 1” “(elaboración propia) | 38 |
| Figura 3.9 Esquema del circuito para las ramas del inversor (elaboración propia) | 39 |
| Figura 3.10 Salida de puerta OR con PWM y habilitación a nivel bajo | 39 |
| Figura 3.11 Salida de puerta OR con PWM y habilitación a nivel alto | 40 |
| Figura 3.12 Esquema y conexiones del montaje en el laboratorio (elaboración propia) | 41 |
| Figura 3.13 Montaje de la placa del microcontrolador (elaboración propia) | 42 |
| Figura 3.14 Esquema eléctrico de la placa del microcontrolador (elaboración propia) | 43 |
| Figura 4.1 Esquema de un encoder óptico [10] | 46 |
| Figura 4.2 Comparativa entre discos de encoders ópticos [10]..... | 46 |
| Figura 4.3 Esquema de un encoder magnético (modificado de [11])..... | 47 |
| Figura 4.4 Encoder EMS22A50-M25-Ld6 [12] | 47 |
| Figura 4.5 modelos de acoplador (izquierda) y soporte (derecha) en Onshape..... | 49 |
| Figura 4.6 Corte de la pieza acoplador | 49 |
| Figura 4.7 Montaje del sensor de posición | 49 |
| Figura 4.8 Esquema del envío de datos del sensor de posición [13] | 50 |
| Figura 4.9 f.e.m. de las fases con el motor girando a 500 rpm | 52 |
| Figura 4.10 f.e.m. de las fases con el motor girando a 1000 rpm | 52 |
| Figura 4.11 Realimentación de velocidad, $K = 500$, $T_i = \infty$ | 56 |
| Figura 4.12 Realimentación de velocidad, $K = 250$, $T_i = \infty$ | 56 |
| Figura 4.13 Realimentación de velocidad, $K = 125$, $T_i = 0,1$ | 57 |
| Figura 4.14 Realimentación de velocidad, $K = 50$, $T_i = 0,05$ | 57 |
| Figura 5.1 Corriente realimentada ante salto en el duty cycle | 65 |
| Figura 5.2 Ensayo lazo corriente $K_c=0,0125$, $T_i=0,004$ | 66 |
| Figura 5.3 Ensayo lazo corriente $K_c=0,0250$, $T_i=0,004$ | 66 |
| Figura 5.4 Ensayo lazo corriente $K_c=0,050$, $T_i=0,004$ | 66 |
| Figura 5.5 Ensayo lazo corriente $K_c=0,035$, $T_i=0,004$ | 67 |
| Figura 5.6 Ensayo lazo velocidad $K_c = 0,005$, $T_i = \infty$ | 68 |

| | |
|---|----|
| Figura 5.7 Ensayo lazo velocidad $K_c = 0,01, T_i = \infty$ | 69 |
| Figura 5.8 Ensayo lazo velocidad $K_c = 0,05, T_i = \infty$ | 69 |
| Figura 5.9 Ensayo lazo velocidad $K_c = 0,01, T_i = 0,1$ | 70 |
| Figura 5.10 Ensayo lazo velocidad $K_c = 0,01, T_i = 0,05$ | 70 |
| Figura 5.11 Ensayo lazo velocidad $K_c = 0,025, T_i = 0,1$ | 71 |
| Figura 6.1 Esquema de comunicación entre interfaz y microcontrolador (elaboración propia) | 73 |
| Figura 6.2 Interfaz de usuario..... | 76 |
| Figura 7.1 Ensayo secuencia de fases con salto de 500 a 1000 rpm..... | 82 |
| Figura 7.2 Ensayo secuencia de fases con salto de 1000 a 1500 rpm..... | 82 |
| Figura 7.3 Ensayo validación control de la velocidad..... | 83 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 2.1 Estado de cada fase según el sector del motor | 11 |
| Tabla 3.1 Comparativa entre microcontroladores [4, 5, 6]..... | 33 |
| Tabla 4.1 Características encoder EMS22A50-M25-Ld6 [13]..... | 48 |
| Tabla 4.2 Desfases obtenidos para el ensayo a 500 rpm | 52 |
| Tabla 4.3 Desfases obtenidos para el ensayo a 1000 rpm | 52 |
| Tabla 4.4 Límites de tramos de conducción en ángulos y datos de 10 bits | 53 |
| Tabla 4.5 Límites de los sectores en ángulos y datos de 10 bits | 53 |
| Tabla 6.1 Elementos de la interfaz de usuario | 76 |

ÍNDICE DE CÓDIGOS

| | |
|---|----|
| Código 4.1 Función medida de la posición..... | 51 |
| Código 4.2 Función cálculo del sector | 54 |
| Código 4.3 Función medida de velocidad | 55 |
| Código 5.1 función realimentación de corriente..... | 60 |
| Código 5.2 Función para generar las señales del inversor | 61 |
| Código 5.3 Función para convertir el duty cycle de decimal a entero | 62 |
| Código 5.4 Implementación del regulador PI de corriente | 64 |

| | |
|---|----|
| Código 5.5 Implementación del regulador PI de velocidad | 68 |
| Código 6.1 definición de la tarea Task2code para comunicación Bluetooth en Arduino IDE..... | 74 |
| Código 6.2 Tarea Task2code encargada de la comunicación Bluetooth..... | 74 |
| Código 6.3 Envío de consigna de velocidad en la interfaz gráfica..... | 77 |
| Código 6.4 Recepción de datos en el dispositivo móvil | 79 |

CAPÍTULO 1: INTRODUCCIÓN

Hasta mediados del siglo XX los motores eléctricos de corriente continua predominantes eran los motores con escobillas. Este tipo de motores presentan ciertos inconvenientes fruto de su propio diseño. El motor con escobillas basa su funcionamiento en el cambio de polaridad de sus bobinas internas mediante el contacto continuo de los extremos libres de éstas con una fuente de tensión continua mientras se produce el giro del motor. Para ello se emplean escobillas y un colector de delgas. Estos elementos se desgastan con facilidad, producen rozamiento, lo cual disminuye el rendimiento y desprenden calor, ruido e incluso partículas de polvo conductoras que manchan el motor. Por este motivo este tipo de motores requieren un mantenimiento considerable. A principio de la década de 1960, gracias a los avances en la electrónica de potencia, surgió el motor eléctrico de corriente continua sin escobillas. Este tipo de motores no necesita del contacto físico móvil entre elementos conductores para su funcionamiento, para conocer la posición del motor y poder cambiar la polaridad de las fases se hace uso de sensores. Las ventajas de los motores sin escobillas son muchas, pero algunas de las más importantes son: presentan un mayor par por unidad de peso, tienen mayor eficiencia, menor mantenimiento y mayor tiempo de vida.

En paralelo al avance y desarrollo de mejores motores eléctricos también se ha observado una evolución en los computadores y microcontroladores. Esta evolución se hace especialmente notable al hablar del Internet de las cosas (Internet of things, IoT). Un concepto que engloba la interconexión entre dispositivos electrónicos presentes en elementos cotidianos. El internet de las cosas también tiene aplicación en la industria, puesto que permite la conexión de dispositivos como por ejemplo sensores o actuadores a computadores o interfaces desde las que se pueden realizar diversas tareas.

En la actualidad existen gran cantidad de microcontroladores enfocados a este propósito, como por ejemplo el microcontrolador ESP32 de Espressif Systems, el cual será empleado en este trabajo. Los microcontroladores de este tipo suelen presentar uno o varios módulos para comunicaciones inalámbricas, como pueden ser Bluetooth o Wifi, además de multitud de pines que permiten tanto la lectura de datos analógicos o digitales como el envío de señales lógicas, digitales, PWM, etc.

Estos microcontroladores son ideales para tareas de control de máquinas puesto que permiten la conexión de sensores, así como de actuadores mediante los cuales realizar las tareas requeridas. Además, gracias a sus módulos de comunicación Wifi y Bluetooth permiten que el control de la máquina se realice a distancia mediante la conexión a una interfaz, la cual puede implementarse en cualquier dispositivo móvil actual con capacidad de comunicación Bluetooth o Wifi.

El presente trabajo se desarrolla con respecto a lo explicado en los párrafos previos. Se pretende realizar el control de un motor BLDC mediante el uso de un microcontrolador con capacidad de comunicación Bluetooth para hacer posible la implementación de una interfaz gráfica ejecutable desde un dispositivo móvil desde la cual recibir información del sistema y recibir las órdenes de operación por parte del usuario.

1.1 ANTECEDENTES

El reciente uso de inversores con señales a las fases accesibles por parte del Departamento de Ingeniería Eléctrica de la Universidad Politécnica de Valencia ha propiciado una mejora en el aprendizaje en la asignatura de máster Análisis dinámico y control de accionamientos eléctricos. El empleo de estos inversores permite entender el funcionamiento en detalle del control sobre un motor eléctrico.

Anteriormente, el estudio de los sistemas de control de las diferentes tipologías de máquinas eléctricas y de las diferentes opciones de control se podía realizar en dos vertientes relativamente separadas. Por una parte, el estudio del régimen dinámico y de las diferentes variantes de control para cada máquina se podía realizar mediante las herramientas de análisis dinámico y control de Matlab y Simulink. Por otra parte, el ensayo de accionamientos industriales de diferentes fabricantes incorporando diferentes tipologías de máquinas eléctricas permitía comparar los resultados de los ensayos con el análisis dinámico realizado con Simulink y, así, establecer una correlación entre las soluciones industriales y la simulación del sistema.

Esta metodología presenta muchas limitaciones. Por una parte, a veces resulta muy complejo establecer comparaciones claras entre un sistema industrial y un sistema simulado debido a las fuertes limitaciones en la instrumentación disponible en el sistema industrial. Por ejemplo, como analizar si un sistema de control realiza el control de corriente utilizando una metodología concreta cuando es muy difícil acceder a las medidas internas del sistema de control como son la consigna de corriente o la corriente instantánea que se utiliza para el control. Por otra parte, la disponibilidad de sistemas de control hace que los laboratorios de accionamientos tengan un coste muy elevado para incorporar variadas tecnologías de control resueltas por diferentes fabricantes en equipos relativamente caros.

El Departamento de Ingeniería Eléctrica cuenta con múltiples bancos de ensayos de accionamientos que cubren variadas tipologías de máquinas de fabricantes y de metodologías de control, pero como se ha indicado previamente, hoy en día existen alternativas que permiten un aprendizaje más favorable de los alumnos al poder desarrollar e implementar y probar la efectividad por su cuenta de metodologías muy variadas de control de accionamientos.

Este trabajo se enmarca en una línea de trabajo que comenzó en el curso 2018-19 con la integración de una placa de control basada en un ESP32 con un inversor para prototipado, sistema en el que el microcontrolador genera las señales de operación de las tres ramas del inversor trifásico de acuerdo a algoritmos de control programados. Ese curso se desarrollaron tres trabajos de fin de grado orientados a resolver tres subsistemas de control típicos para una máquina de inducción: la modulación de ancho de impulso por vector espacial, la realimentación de posición y velocidad y la realimentación de corriente para el control por orientación por campo de la máquina de inducción. El curso 19-20, debido a la pandemia, requirió la reformulación de trabajos debido a la imposibilidad del trabajo presencial durante el confinamiento. Por último, el último curso 20-21 continuó con esos pasos iniciales con el desarrollo de un trabajo para el control por orientación de campo de rotor de una máquina síncrona de técnica senoidal.

Es en este contexto en el que se aborda el presente trabajo con el desarrollo de un banco de ensayos para una nueva tecnología de máquina eléctrica no trabajada todavía, la máquina de corriente continua sin escobillas o Brushless DC, con una sistemática de control completamente diferente a las dos tipologías previas y con un motor típico de imanes permanentes con sus especificidades típicas, a

saber, un tiempo característico eléctrico muy rápido que permite un control de la corriente rapidísimo, acompañado de una baja inercia que supone también una dinámica muy rápida en el subsistema mecánico.

1.2 OBJETIVOS

Los objetivos que se persiguen en este Trabajo de Fin de Grado son la implementación de un sistema de control de la velocidad de un motor BLDC integrado en un banco de ensayos de accionamientos electrónicos mediante el uso de un microcontrolador que coordine los elementos necesarios para el control: inversor, sensores, criterios de control, reguladores, etc. El control del motor debe permitir regular de forma precisa la velocidad impidiendo que se sobrepasen los límites de corriente admisibles en las fases del motor o en el inversor.

Además, el proyecto contempla el diseño y desarrollo de una interfaz gráfica que pueda ser ejecutada desde un dispositivo móvil Android. Desde esta interfaz, debe de ser posible obtener en tiempo real información del sistema de control, así como ordenar un cambio de velocidad del motor.

Con estos objetivos claros, se pueden establecer una serie de hitos a cumplir durante el desarrollo del trabajo. Estos son:

- Realizar simulaciones por software del control a implementar sobre el motor
- Identificar los elementos necesarios para el control real
- Realizar el montaje y conexiones de los elementos necesarios
- Desarrollar el código necesario para el control mediante microcontrolador
- Ajustar mediante ensayos los parámetros de los reguladores necesarios para obtener la mejor dinámica posible
- Diseñar y desarrollar la interfaz gráfica junto con el código necesario para la comunicación
- Probar el sistema final implementado y validarlo

1.3 DESCRIPCIÓN Y ESTRUCTURA DE LA MEMORIA

Con los objetivos anteriores en mente se puede establecer la forma y estructura del presente trabajo. Esta estructura está basada en capítulos, en los que se llevarán a cabo diferentes tareas permitiendo el cumplimiento progresivo de los objetivos anteriores.

En el segundo capítulo, se llevarán a cabo una serie de ensayos para simular mediante software el control a realizar y poner de manifiesto los elementos que son necesarios y algunos de los problemas que pueden aparecer durante la implementación real del banco. Además, estos ensayos permitirán conocer la respuesta a esperar del sistema.

En el tercer capítulo, conociendo los elementos necesarios para el control, se procederá a la selección, montaje, integración y conexión de estos elementos. Para llevar a cabo esta parte de manera correcta se deben tener en cuenta las propiedades de cada uno de los elementos, así como la forma de cada uno de enviar y recibir datos.

En los anteriores capítulos se pondrá de manifiesto la necesidad de un sensor de posición, que será el encargado de realimentar la información relativa a la posición y a la velocidad. Puesto que este es un elemento fundamental y requiere de un acoplamiento sobre el propio motor, el cuarto capítulo se dedicará enteramente al sensor de posición, sus funciones y su adaptación al banco de ensayos.

Una vez realizado el montaje con las conexiones necesarias entre elementos se procede, en el quinto capítulo, a desarrollar el código que permite al microcontrolador coordinar los diferentes elementos. En este capítulo también se implementarán y ajustarán los reguladores necesarios para el control de la velocidad del motor.

Una vez terminadas las tareas descritas en el anterior capítulo, el control sobre el motor queda finalizado, por lo que solo queda diseñar y desarrollar la interfaz de usuario. Esto se lleva a cabo en el sexto capítulo.

Para finalizar, en el séptimo y último capítulo, solo queda realizar pruebas para validar el sistema de control implementado. En este capítulo, además, se revisarán los objetivos para comprobar que se han cumplido a lo largo del desarrollo del trabajo.

CAPÍTULO 2: CONTROL DEL MOTOR Y DESARROLLO EN SIMULINK

2.1 INTRODUCCIÓN

En este capítulo se desarrolla el control a llevar a cabo sobre el motor BLDC, así como una serie de simulaciones en el software Simulink que permitirán poner de manifiesto tanto la respuesta a esperar de este tipo de control como los elementos necesarios para replicar el control en un sistema real. Este paso es fundamental hoy en día en el diseño de accionamientos eléctricos controlados electrónicamente ya que permite realizar la implementación en el sistema real con bastante seguridad de que la solución aportada para el control es viable y efectiva y, además, permite explorar alternativas y ajustes del sistema de control que luego se pondrán en práctica en la implementación del sistema real.

2.2 ESTRUCTURA

El capítulo se estructura en los siguientes apartados: OBJETIVO DEL CONTROL, en el cual se presentarán los distintos objetivos a cumplir con el control del motor, MODO DE CONTROL DEL MOTOR, apartado en el cual se presentará una descripción del modo seleccionado para controlar el motor, ENSAYOS EN SIMULINK, donde se desarrollarán una serie de ensayos previos de los cuales se obtendrán algunas nociones básicas del funcionamiento tanto del motor como del lazo de control y además se planteará un modelo válido para el control con la perspectiva de replicarlo con los elementos necesarios en el prototipo a desarrollar en el laboratorio, y, por último, DESCRIPCION DEL SISTEMA REAL, apartado en el cual se describirá el modelo real que permitirá el control del motor con todos sus elementos necesarios.

2.3 OBJETIVO DEL CONTROL

El objetivo a perseguir es el control de la velocidad del motor BLDC de manera segura, sin que en las fases del motor se sobrepase de forma permanente la corriente nominal y con la mejor dinámica posible en el control de corriente y de velocidad. Este control se llevará a cabo mediante un sistema en cascada con dos lazos, uno externo que será el encargado de controlar la velocidad y otro interno encargado del control de la corriente. En el siguiente apartado se desarrolla de manera detallada la forma de control aplicada al motor.

2.4 MODO DE CONTROL DEL MOTOR

Para el control del motor eléctrico sin escobillas se empleará el método de control de corriente rectangular, el cual se desarrolla en detalle en este apartado.

El fundamento para provocar el giro de un motor eléctrico se basa en generar dos campos magnéticos, uno en el estator y otro en el rotor, desfasados en el espacio, de tal forma que aparezca un par que tienda a alinearlos, este par electromagnético provocará el par que hará girar el motor. Para conseguir el campo magnético en el rotor se emplea un imán permanente, mientras que para el campo del estator se alimentan con corriente continua una combinación de fases situadas en el mismo para que en el momento adecuado produzcan un campo magnético determinado que hará que el imán permanente situado en el rotor quede sometido a un par de giro para intentar alinearse con el campo producido por el estator. Dado que se emplea corriente continua por tramos (rectangular) para alimentar las fases, se definirán seis sectores sobre los 360 grados de giro, estos sectores delimitarán las zonas de conducción positiva o negativa de cada fase. De esta manera, según el sector concreto en que se situó el motor, siempre se alimentaran las fases para producir el mayor par de giro del rotor. Una desventaja de este método de control es que al haber una serie de sectores definidos en los cuales la corriente que alimenta las fases es constante, durante el movimiento del rotor dentro de un mismo sector, el campo magnético formado por las fases del estator mantiene su orientación, mientras que el campo formado por el rotor, al moverse este, cambia su orientación. Dado que el par generado, depende en general de la orientación relativa entre los campos magnéticos, siendo este máximo a noventa grados, el par tendrá un cierto rizado. Los motores BLDC se diseñan, sin embargo, para que este rizado sea mínimo, lo que se consigue con una distribución espacial del campo producido por el imán permanente del rotor en el entrehierro que sea prácticamente plana y abarcando todo el paso polar. En una máquina de corriente alterna en general, sí que se cumple con bastante precisión la afirmación previa respecto del par (su dependencia de las orientaciones relativas de los campos producidos por estator y por rotor según una ley senoidal). Sin embargo, la constitución del motor BLDC consigue que el par que producen dos fases alimentadas con corriente constante sea aproximadamente constante para cualquier posición dentro del rango en que el campo producido por el imán permanente no cambia de sentido en todos los puntos del entrehierro situados bajo los conductores de una fase. En los siguientes párrafos se analizará el cálculo del par en una máquina eléctrica en general y la particularización a la máquina BLDC [1].

Consideremos una máquina rotativa con una de las configuraciones típicas para accionamientos controlados: estator de interior cilíndrico y rotor liso o con saliencia. El siguiente procedimiento nos permite calcular el par de origen electromagnético sobre el estator que, por el principio de acción y reacción será igual y contrario al que se ejerza sobre el rotor. Volviendo a la máquina de estator liso, si consideramos un elemento diferencial de la cara interior del estator, el esfuerzo que aparece sobre los conductores en él incluidos se puede determinar aplicando la ley de Biot y Savart teniendo en cuenta que la capa de corriente nos da información de los amperio-conductores alojados en ese elemento diferencial de la cara interior del estator, Figura 2.1:

$$dF_{est} = - B_{en\ est}(\alpha, t) \cdot l \cdot a_{est}(\alpha, t) \frac{D}{2} d\alpha$$

El esfuerzo diferencial anterior resulta en la dirección tangencial del entrehierro y se ha considerado positivo en la dirección de ángulos crecientes. En consecuencia, el par diferencial que aparece sobre ese elemento de la cara interior del estator es:

$$dT_{est} = \frac{D}{2} dF_{est} = - B(\alpha, t) \cdot l \cdot a_{est}(\alpha, t) \left(\frac{D}{2}\right)^2 d\alpha$$

En dónde consideramos positivo un par en sentido de ángulos crecientes. El par total sobre el estator se puede calcular integrando a lo largo de toda la periferia del estator:

$$T_{est} = -l \cdot \left(\frac{D}{2}\right)^2 \int_0^{2\pi} B_{en\ est}(\alpha, t) \cdot a_{est}(\alpha, t) d\alpha$$

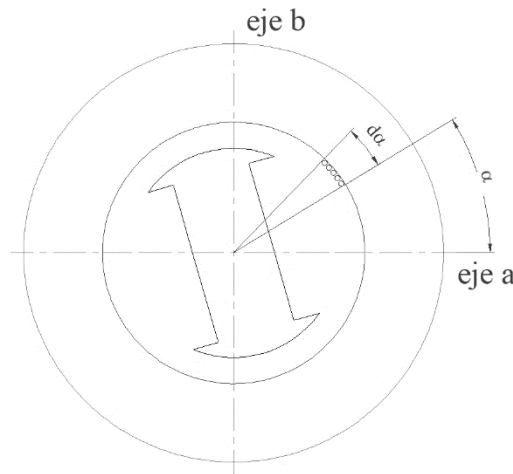


Figura 2.1 Elemento diferencial de capa de corriente en la cara interior del estator de una máquina de estator cilíndrico

Y el par sobre el rotor tendrá el mismo valor, pero sentido contrario:

$$T_{rot} = l \cdot \left(\frac{D}{2}\right)^2 \int_0^{2\pi} B_{en\ est}(\alpha, t) \cdot a_{est}(\alpha, t) d\alpha$$

Para iniciar el cálculo del par, dado que son máquinas con un circuito magnético principal no saturado por el importante entrehierro, asumiremos el objetivo por separado: con el par producido por la interacción de la capa de corriente del estator con el campo producido por el imán por un lado y con el campo autoinducido por otro. En la parte asociada al campo producido por el imán permanente, asumiremos la tarea también por partes, considerando cada fase del estator por separado. Con ese planteamiento, para una fase cualquiera del estator, la ecuación anterior se puede transformar teniendo en cuenta que la inducción debida al imán permanente en cada uno de los puntos del entrehierro en los que hay conductores de la fase coincide a una cierta escala con la f.e.m. (fuerza electromotriz) inducida en esos conductores debida al giro del imán respecto de ellos:

$$e_{pm,cond}(\alpha, t) = B_{pm,en\ est}(\alpha, t) \cdot l \cdot v = B_{pm,en\ est}(\alpha, t) \cdot l \cdot \Omega(t) \cdot \frac{D}{2}$$

Por otra parte, al tratarse de conductores de una fase, la corriente en todos ellos es la misma e igual a la corriente de fase, y la capa de corriente en cada punto de la cara interior del estator es igual a la densidad lineal de conductores en cada punto, $z_{est}(\alpha, t)$, por la corriente de fase:

$$a_{est}(\alpha, t) = z_{est}(\alpha, t) \cdot i_{fase}(t)$$

Con lo que la ecuación general del par puede reescribirse en este caso como:

$$\begin{aligned} T_{rot, pm-fase} &= \int_0^{2\pi} \frac{e_{pm,cond}(\alpha, t)}{\Omega(t)} z_{est}(\alpha, t) i_{fase}(t) \frac{D}{2} d\alpha = \\ &= \frac{i_{fase}(t)}{\Omega(t)} \int_0^{2\pi} e_{pm,cond}(\alpha, t) \cdot z_{est}(\alpha, t) \cdot \frac{D}{2} d\alpha \end{aligned}$$

Ahora bien, la integral en la expresión anterior coincide con la suma a lo largo de la periferia interior del estator de las ff.ee.mm. individuales de los conductores con su signo (si son de ida o de vuelta) y equivale a la f.e.m. inducida en la fase:

$$T_{rot, pm-fase} = \frac{e_{pm,fase}(t) i_{fase}(t)}{\Omega(t)}$$

Para el conjunto de las m_{est} fases del estator en general tendremos:

$$T_{rot, pm} = \frac{1}{\Omega(t)} \sum_{k=1}^{m_{est}} e_{pm,k}(t) i_k(t)$$

Esta ecuación permite reconocer la característica especial del sistema de control de par empleado en estas máquinas: dado que el rotor se diseña para que induzca una f.e.m. de forma trapezoidal en el rotor con tres tramos constantes positivos y negativos para cada fase distribuidos sobre un tercio de dos pasos polares, el objetivo para conseguir un par constante es alimentar en cada momento con corriente constante (proporcional al par deseado) las dos fases en las que se induce f.e.m. constante, cada una con su signo. De esta forma, con una corriente constante fácil de controlar se consigue un par motor constante, perfecto para conseguir una marcha suave con bajas vibraciones al no haber pulsaciones de par.

Respecto al par que se produce por interacción de la corriente del estator con el campo producido por el propio estator, en máquinas de imanes permanentes en general puede tener una cierta importancia debido al grado de saliencia, pero lo cierto es que las máquinas BLDC tienen comúnmente una saliencia muy reducida, lo que hace que el citado par, que es un par de reluctancia, sea prácticamente despreciable por comparación con el que produce la interacción con el imán.

2.5 ENSAYOS EN SIMULINK

Para desarrollar el sistema de control en un entorno de simulación se realizarán una serie de simulaciones en el programa Simulink, las cuales irán evolucionando para obtener un modelo final en el que se consigan los objetivos descritos anteriormente. Los modelos desarrollados utilizan el modelo base de la librería Simscape para motores BLDC [2].

2.5.1 Primer ensayo: f.e.m. inducida

Este primer ensayo se centra en observar lo que ocurre cuando, con las fases del motor en circuito abierto, giramos mecánicamente el rotor. Con esto, se pretende obtener el perfil de las fuerzas electromotrices inducidas en las fases del motor. Además, es interesante observar cómo cambian estas fuerzas electromotrices con el cambio de sector del rotor. Nótese que en el modelo los sectores están numerados de 0 a 5. Para conseguir este propósito, el modelo consistirá en el motor BLDC, una fuente externa de velocidad que permita girar el motor a una velocidad ajustable, sensores de voltaje en las fases, las cuales estarán en circuito abierto, y un sensor de la posición del motor para obtener el sector. El modelo puede observarse en la Figura 2.2

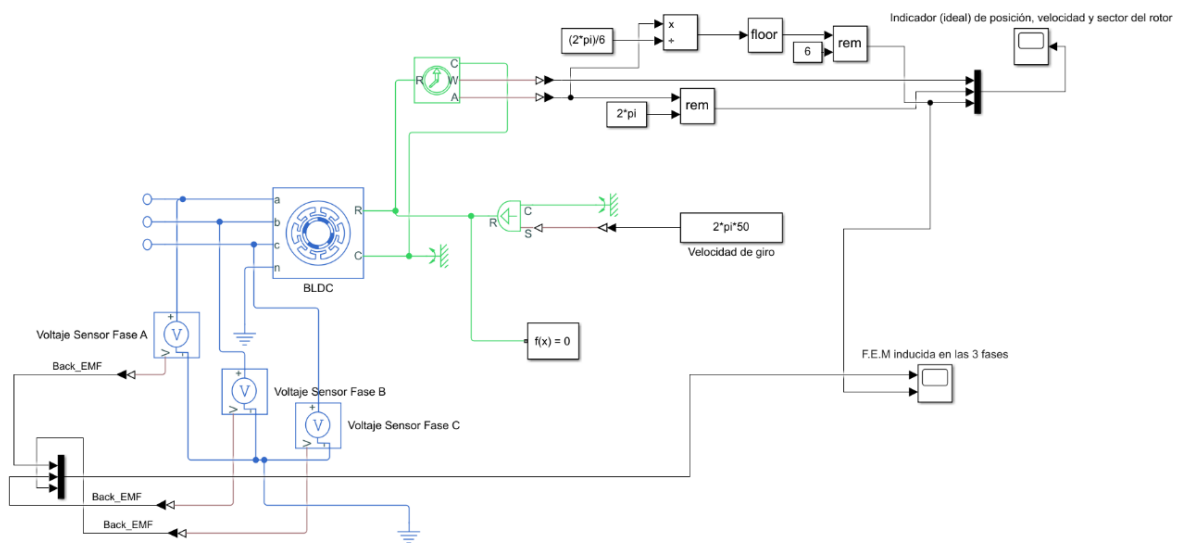


Figura 2.2 Modelo Simulink primer ensayo

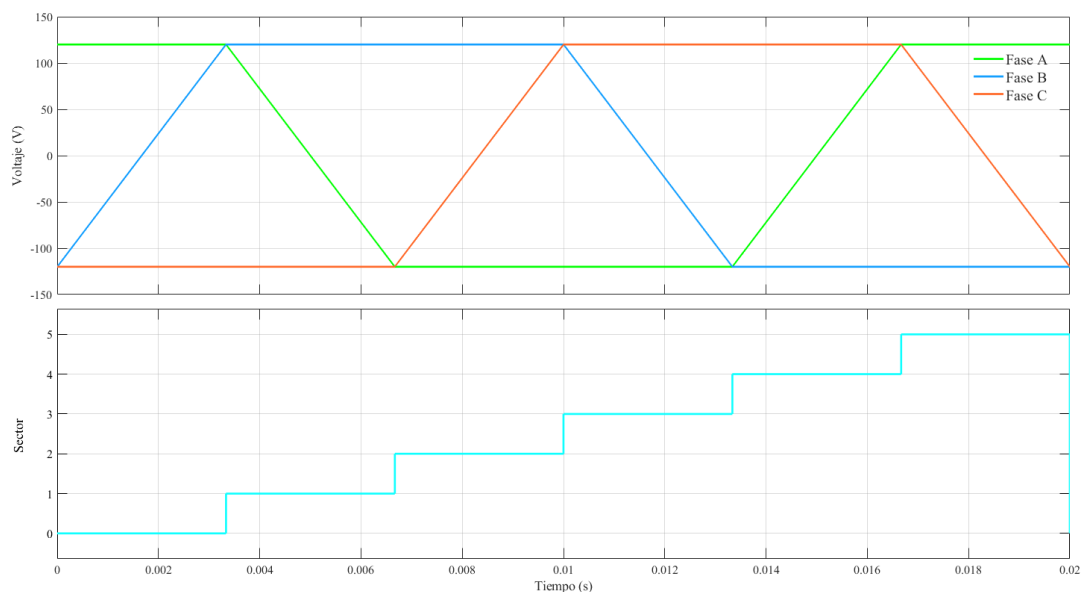


Figura 2.3 f.e.m. inducidas en las fases del motor BLDC y cambios de sector

De la Figura 2.33 se puede extraer que las tensiones en las fases son alternantes y tienen un perfil trapezoidal en el que en dos de los sectores aparece tensión positiva, en otros dos sectores tensión negativa y los dos restantes son transiciones entre estos estados. También es relevante resaltar que en todo momento nos encontramos una fase en estado positivo y otra en negativo. La información obtenida de este ensayo queda reflejada en la Tabla 2.1.

| Sectores: | 1 | 2 | 3 | 4 | 5 | 6 |
|---------------|------------|------------|------------|------------|------------|------------|
| Fase A | Positiva | Transición | Negativa | Negativa | Transición | Positiva |
| Fase B | Transición | Positiva | Positiva | Transición | Negativa | Negativa |
| Fase C | Negativa | Negativa | Transición | Positiva | Positiva | Transición |

Tabla 2.1 Estado de cada fase según el sector del motor

2.5.2 Segundo ensayo: Corrientes en las fases a velocidad constante

Después del primer ensayo, se pretende ahora alimentar el motor con corriente externa continua que habrá que conmutar en las diferentes fases para que tome valor positivo, negativo o nulo dependiendo del sector en que se encuentre el rotor. Para ello hemos de tener en cuenta la lógica de sectores. Como se ha visto, el funcionamiento del motor se basa en aplicar corriente en las fases adecuadas (corriente positiva en la fase que tiene f.e.m. inducida constante positiva, negativa en la que tiene f.e.m. negativa y corriente cero en la fase en transición) en cada momento para que el par interno sea siempre del mismo sentido e intensidad. Para ello, se incluye en el modelo un inversor que, atendiendo al sector en el que se encuentre el rotor aplique la corriente externa continua a las fases correspondientes. Para la lógica se sigue la secuencia de la Tabla 2.1 aplicando corrientes positivas negativas o nulas según corresponda. Este ensayo se realiza a velocidad constante y aplicada mecánicamente al rotor del motor para una mejor visualización de la secuencia de corrientes. El modelo puede observarse en la Figura 2.5.

Observando la Figura 2.4, se puede ver como la secuencia de corrientes es la esperada cambiando adecuadamente en cada cambio de sector.

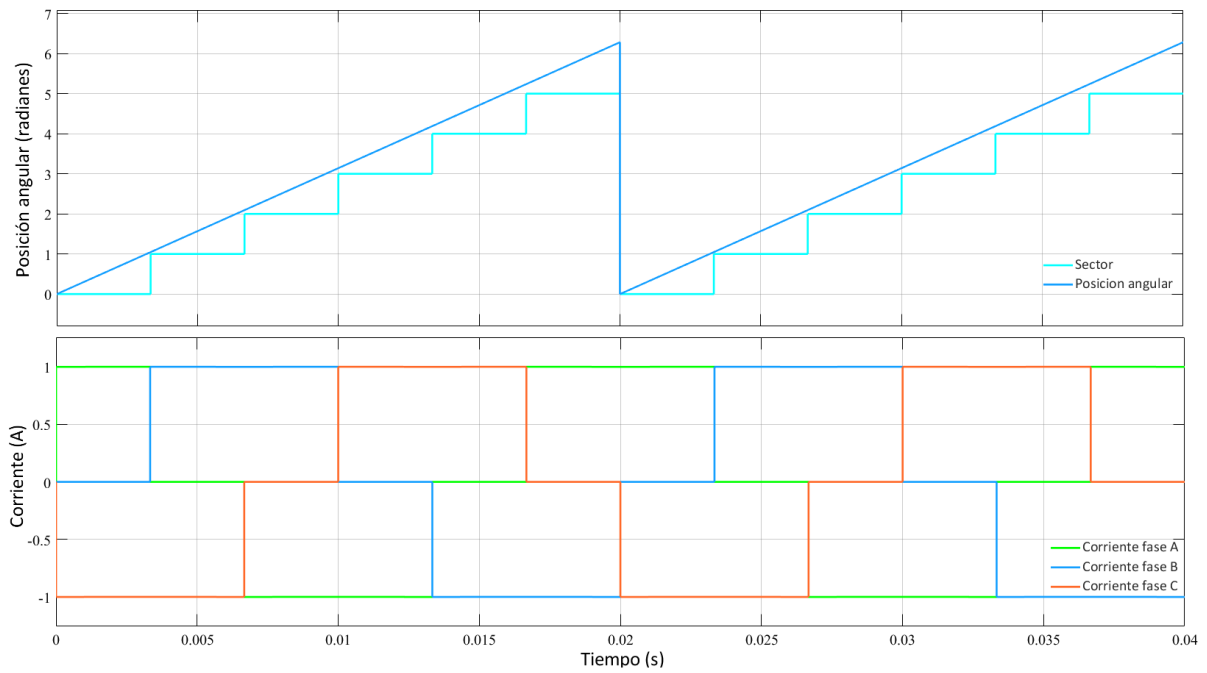


Figura 2.4 Secuencia de corrientes alimentado el motor mediante inversor

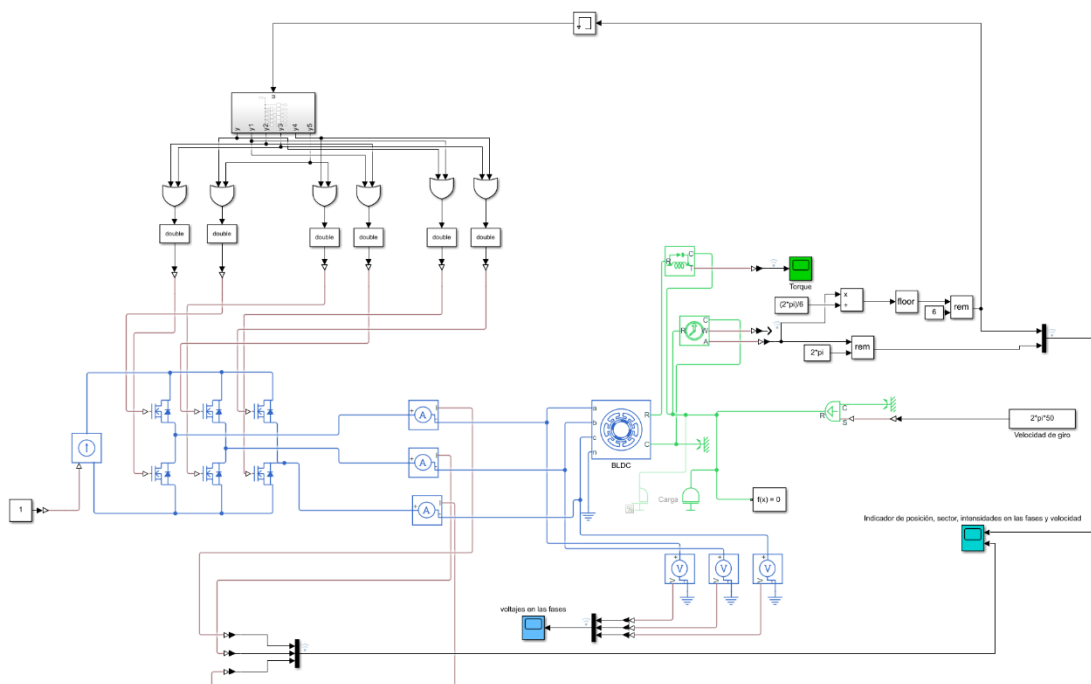


Figura 2.5 Modelo Simulink segundo ensayo

En la Figura 2.6 se muestra el par del motor en este ensayo. Como se puede observar, la secuencia de corriente rectangular en cada fase permite, mediante un control sencillo de la corriente (corriente rectangular por tramos) obtener un par prácticamente constante de origen electromagnético que permite una marcha suave con pocas vibraciones para el accionamiento.

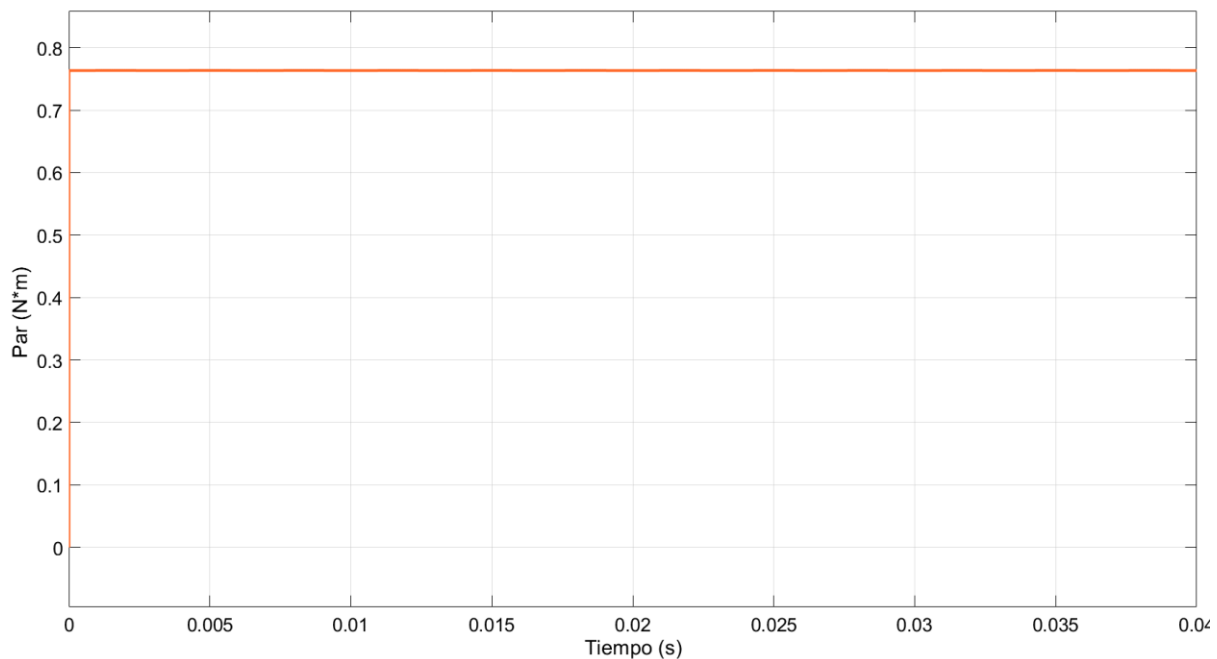


Figura 2.6 Par del motor alimentando mediante inversor

2.5.3 Tercer ensayo: Alimentación de tensión con PWM

A partir de este ensayo, el modelo se irá encaminando hacia el sistema que permitirá controlar el motor. Este ensayo se realiza a velocidad nula con el rotor bloqueado y se cambia la fuente de corriente por una fuente de tensión. Cabe esperar que el resultado sea muy parecido a alimentar con fuentes ideales de corriente salvo por un retraso de tipo exponencial amortiguado en las corrientes debido a las resistencias e inductancias propias del motor. Una vez cambiada la fuente de alimentación se procede a la implementación de una modulación por ancho de pulso que controle a través del duty cycle la tensión media que llega a las fases. De momento, se controla el duty cycle de forma manual, variándolo hasta conseguir una corriente en las fases de aproximadamente un amperio. Se ha de tener en cuenta que, para aplicar la modulación por ancho de pulso, se ha de pasar la PWM positiva, es decir aquella que tenga un duty cycle mayor de 0,5, a la fase que en ese momento deba tener tensión positiva y a la fase a la que le corresponda tensión negativa se la pasará un PWM con un duty cycle con un valor de uno menos el duty cycle asociado a la fase positiva, es decir, la parte complementaria. En este caso al estar el motor bloqueado no es necesario de momento aplicar la lógica de la asociación de la PWM a las diferentes fases según el sector. El modelo se muestra en la Figura 2.7.

En la Figura 2.8 se muestra cómo se estabiliza la corriente alrededor de un amperio con un salto de duty cycle de 0,5 a 0,51. Vemos como el resultado es el esperado, un establecimiento rápido de la corriente a un valor adecuado. Además, se hace notable el rizado de la corriente debido a la modulación por ancho de pulso, aunque su amplitud es moderada debido a la alta frecuencia de conmutación utilizada y al efecto amortiguador de las inductancias de fase del servomotor.

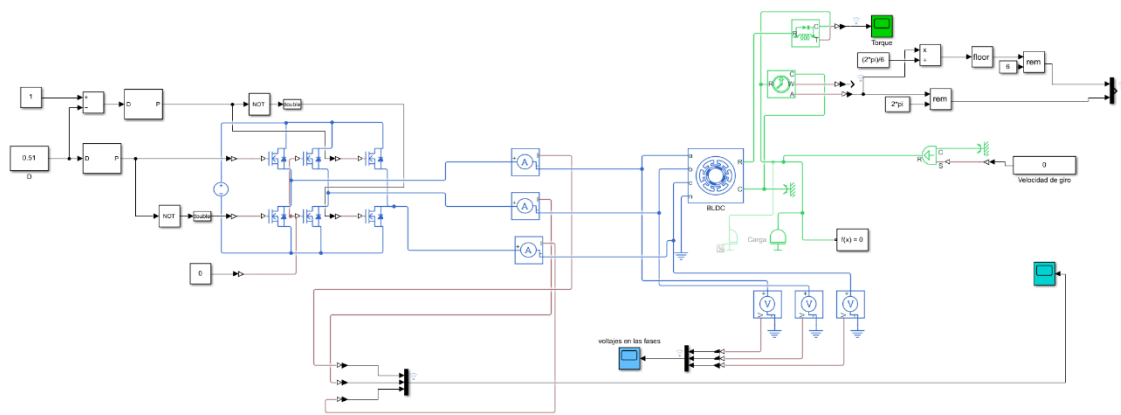


Figura 2.7 Modelo Simulink tercer ensayo

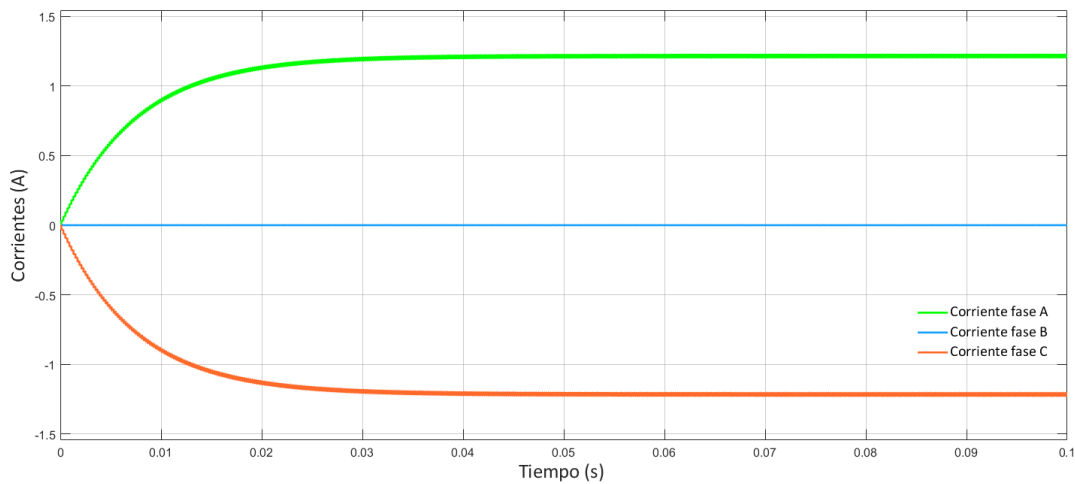


Figura 2.8 Estabilización de las corrientes en las fases del motor ante un salto de duty cycle

2.5.4 Cuarto ensayo: PWM con control proporcional

Este ensayo es evolución del anterior. Con el modelo actual, se cambia la forma en la que se proporciona el valor del duty cycle. Ahora este valor lo proporciona un control proporcional que tiene en cuenta el error entre la corriente de consigna que se quiera obtener y las que se tienen en las fases. El ensayo se realiza también a rotor bloqueado, por tanto, sigue sin ser necesario un control de la lógica del PWM. Para implementar el control se ha introducido un sistema de control de tiempo discreto para realizar un acercamiento a la implementación definitiva con el microcontrolador. El modelo queda reflejado en la Figura 2.9.

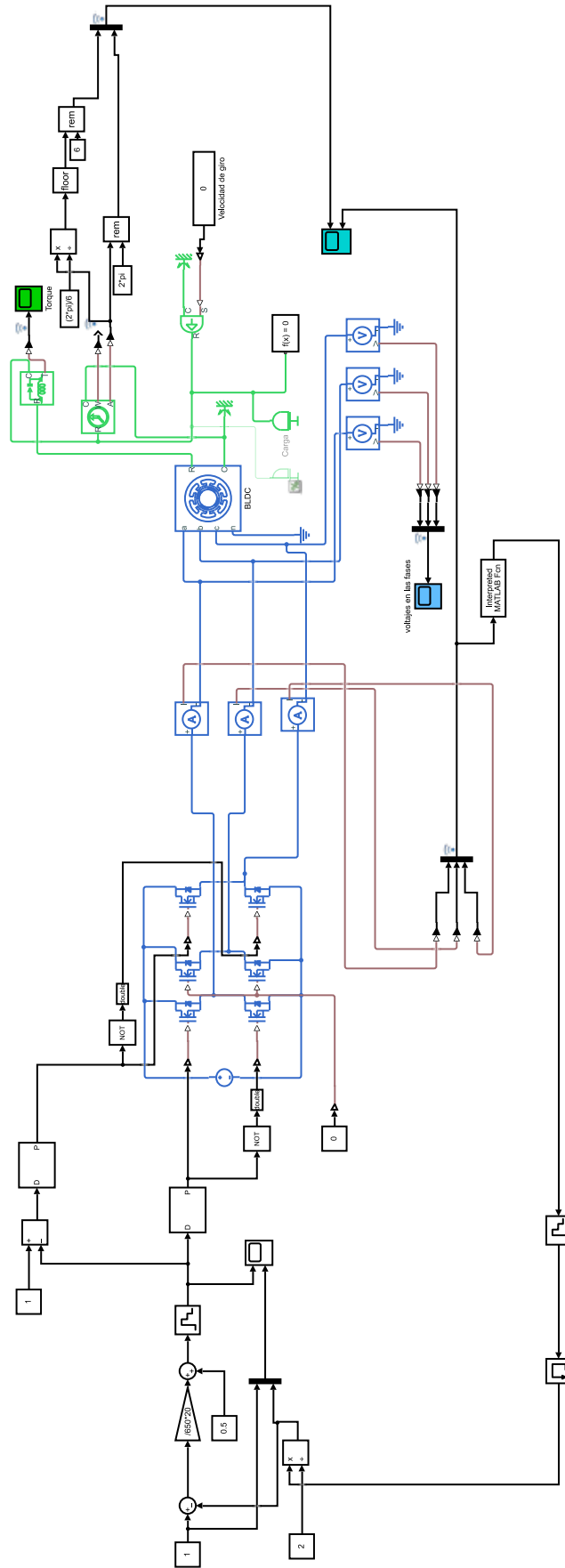


Figura 2.9 Modelo Simulink cuarto ensayo

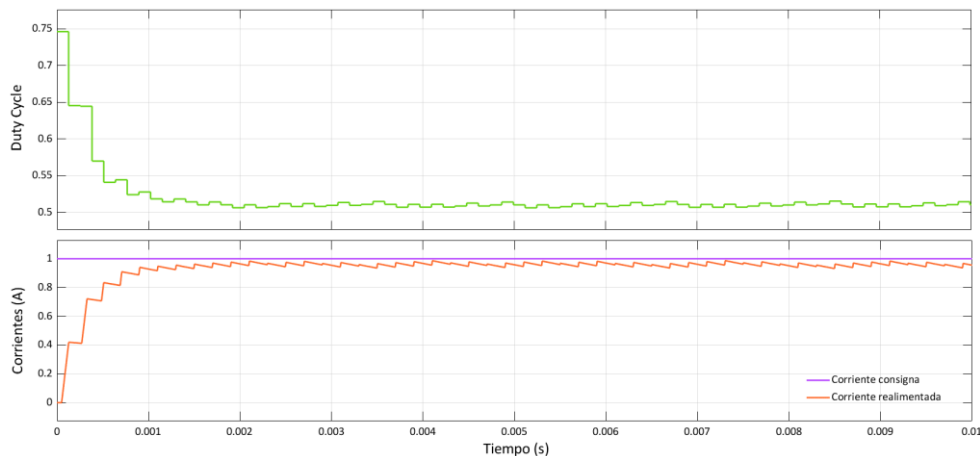


Figura 2.10 control de la corriente mediante PWM con acción proporcional

En la Figura 2.10 Se observa como el duty cycle va variando con el tiempo, descendiendo desde valores cercanos al 75% (el valor inicial que corresponde al instante inicial en el que el regulador de tipo P ve el máximo error en la corriente y debe actuar con la máxima acción de control) hasta valores ligeramente superiores al 50% una vez estabilizada la corriente en un valor cercano al de consigna, lo que produce un rápido aumento de la corriente hasta un establecimiento en el valor deseado, en este caso, un amperio. Este ensayo permite comprobar el funcionamiento adecuado del control P (discreto en el tiempo) de corriente en condiciones seguras del que se partirá para la implementación del control de corriente en el sistema real.

Cabe destacar en este apartado el modo en que se obtiene la corriente de realimentación. Dado que en cada instante siempre existe una fase en estado de conducción positiva, otra en conducción negativa y otra sin circulación de corriente, en estos ensayos se toma la corriente como la suma de las corrientes de las tres fases en valor absoluto y ese resultado se divide entre 2. Más adelante se pondrá de manifiesto el error asociado a realizar la realimentación de corriente siguiendo esta fórmula.

2.5.5 Quinto ensayo: Ajuste de corriente con control PI

Una vez visto el modo de ajustar la corriente al valor deseado, se procede a modificar el modelo para que tenga en cuenta la lógica de los sectores para que la PWM se aplique a las fases adecuadas en cada momento. Además, se implementa el control de tipo PI con el ajuste estándar consistente en establecer el tiempo integral del regulador PI igual al tiempo característico del devanado en el que se controla la corriente, es decir L_{dev} / R_{dev} . Para probar el funcionamiento del modelo se deja el rotor del motor girando a velocidad constante. En resumen, el modelo, mostrado en la Figura 2.11, consta de un regulador PI, que se realimenta de las corrientes en las fases y de un valor consigna, que en este caso es de 1A, y devuelve un valor de duty cycle. Este valor de duty cycle se envía a las ramas del inversor siguiendo siempre la lógica comentada anteriormente según el sector en el cual se encuentre el motor. Todo ello se realiza con el motor girando a velocidad constante. Se pretende medir la corriente en las fases y comprobar que llega al valor de consigna deseado y que, además, sigue la secuencia adecuada.

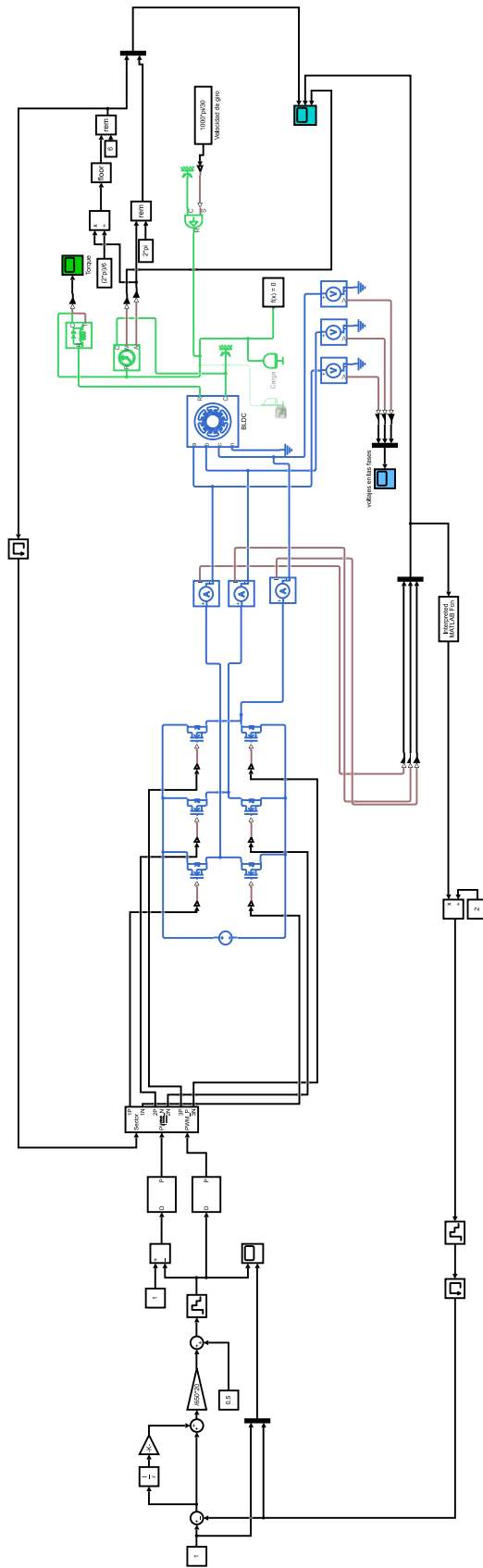


Figura 2.11 Modelo Simulink quinto ensayo

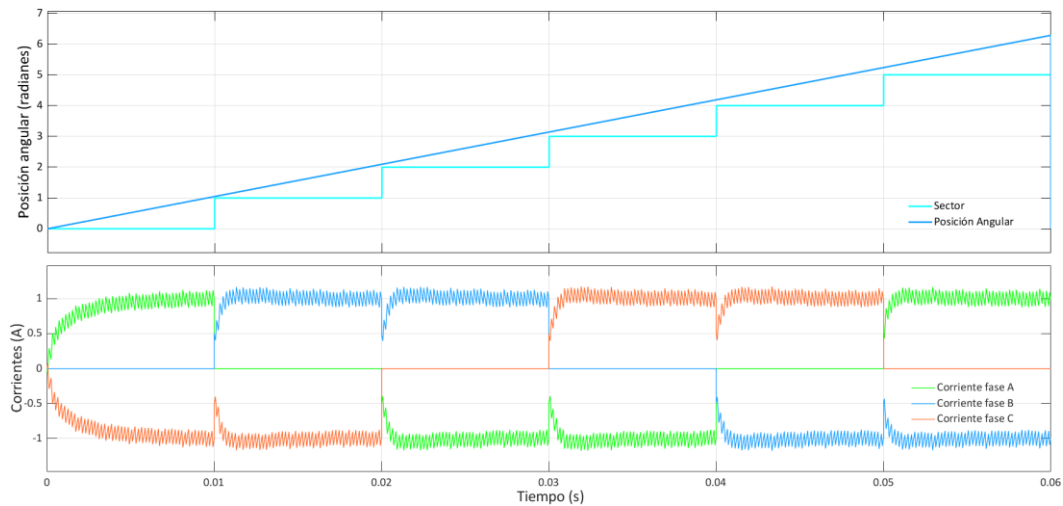


Figura 2.12 Corrientes en las fases con regulación PI y consigna 1 A

Se observa, con los resultados mostrados en la Figura 2.12, cómo se consigue el objetivo, establecer una corriente de un valor determinado y que circule de manera adecuada por las fases correspondientes según el sector en el que se encuentre el rotor. Se pueden apreciar las similitudes entre esta gráfica y la Figura 2.5, en la que se alimentaba al motor con una fuente de corriente ideal. Se puede observar también el patrón rectangular típico de la corriente en las máquinas BLDC con el pico de bajada (dip) en cada centro de la zona de conducción con corriente constante. Este pico se debe a la forma de realizar el control en el que se bloquea la rama que estaba previamente conduciendo lo que produce una rápida extinción de su corriente (más rápida que la construcción de la corriente de la rama que entra en conducción) que se traduce en una reducción breve y brusca de la corriente en la rama que en ese momento está conduciendo corriente constante. Este patrón es el origen de la variación temporal del par que proporciona la máquina BLDC en régimen permanente, que es origen de vibraciones y de marcha menos suave que la que se consigue en servos síncronos de imanes permanentes con tecnología de control por corriente senoidal.

Es necesario aclarar que, para el ajuste de la ganancia proporcional del regulador PI empleado se sigue un proceso consistente en comenzar con un valor de ganancia bajo e ir aumentándolo hasta que se aprecie una sobreoscilación moderada mientras que el tiempo integral se mantiene siempre igual al tiempo característico del devanado.

2.5.6 Sexto ensayo: Ajuste independiente de velocidad con control PI

Una vez resuelto el control de corrientes, se puede implementar un control similar para controlar la velocidad del motor. Para ello se desarrolla un control PI similar al que se ha implementado en corriente. Este control tiene en cuenta el error entre la velocidad de consigna y la velocidad medida. Para ello, se replica el modelo anterior cambiando la realimentación al PI por la velocidad medida del motor. La salida del PI seguirá siendo el duty cycle que entrará directamente al inversor. El modelo queda representado en la Figura 2.13 y los resultados obtenidos se muestran en la Figura 2.14.

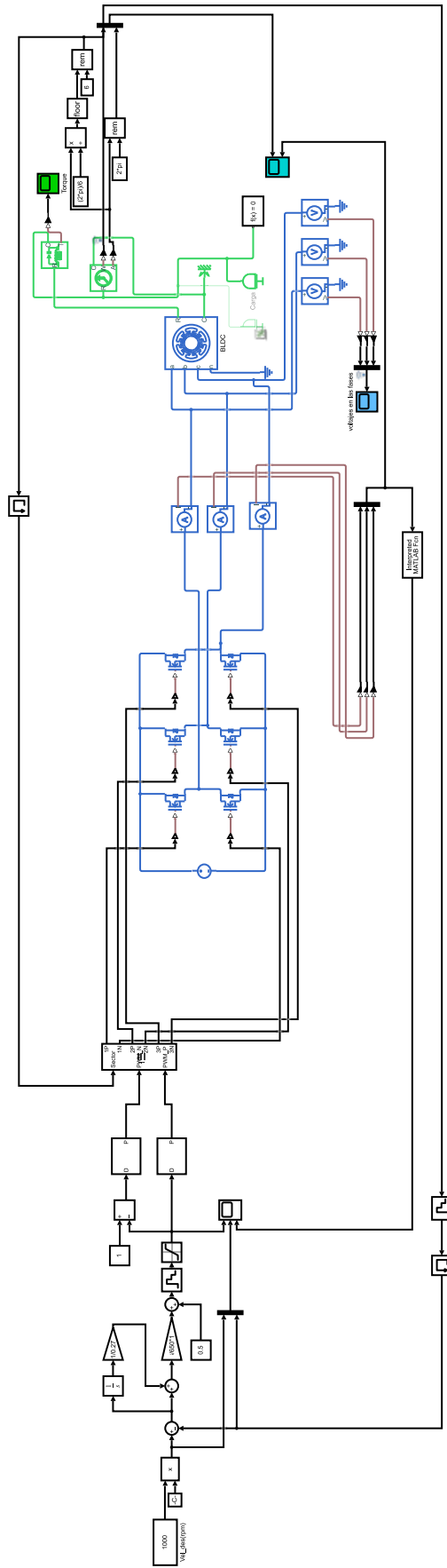


Figura 2.13 Modelo Simulink sexto ensayo

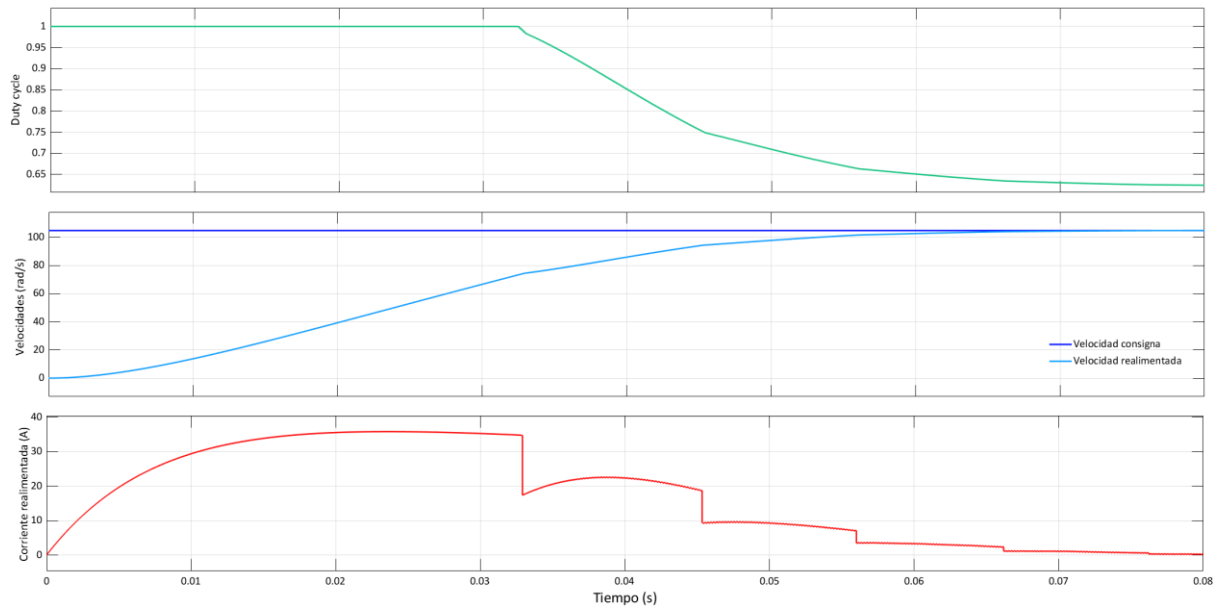


Figura 2.14 Velocidad y corrientes con control PI y salto de consigna de 0 a 1000 rpm

Se observa claramente como la velocidad consigna se alcanza rápidamente, pero a costa de aumentar la corriente a valores completamente inadmisibles para el inversor e incluso para las propias fases del motor. Por este motivo no puede implementarse directamente un control directo PI sobre la velocidad en sistemas industriales reales, ya que daría lugar a transitorios con valores inadmisibles de la corriente tanto para el inversor como para el motor.

2.5.7 séptimo ensayo: Control en cascada

Para paliar el problema de las elevadas corrientes que tiene asociado el control directo velocidad-duty cycle y que se ha puesto de manifiesto en el ensayo previo, en este ensayo se implementará un control en cascada. En una primera etapa se configura el control PI de la corriente velocidad igual que en el anterior ensayo, pero ahora la salida de esta etapa no es directamente el duty cycle, sino que es la consigna de corriente saturada a un rango de entre -2,5 y 2,5 amperios. Esta consigna entra en la segunda etapa, el control de corriente, que es el mismo que se pudo ver en el quinto ensayo. Con la saturación se impide que la corriente aumente a valores inadmisibles. Es decir, se utiliza la limitación a la salida del regulador de velocidad como una especie de protección interna del motor y del inversor ya que nunca se pedirá una corriente mayor que la nominal del mismo. En este modelo, además, se debe implementar un sistema anti-windup en ambos lazos, puesto que en ambos existe saturación y el regulador puede funcionar saturado durante periodos de tiempo significativos. El lazo de velocidad limita la salida a una corriente de consigna en el rango [-2,5 – 2,5] y el lazo de corriente limita su salida a un duty cycle en el rango [0 – 1]. Debido a esto, la acción integral tiende a aumentar durante el tiempo que la salida se encuentre saturada. En la Figura 2.15 se puede ver el modelo, que no es más que una combinación de los anteriores, con el PI de velocidad seguido del PI de corriente con las respectivas saturaciones y mecanismos anti-windup.

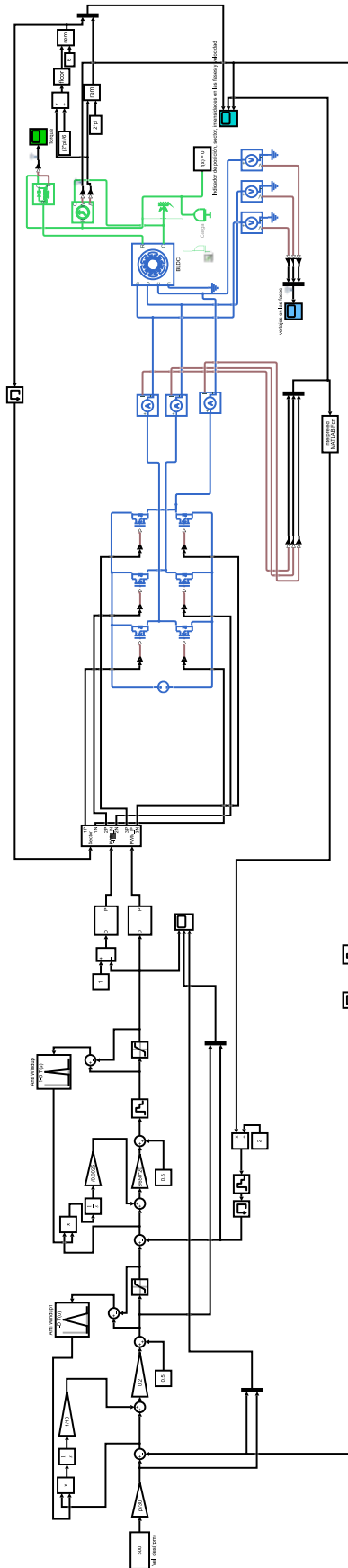


Figura 2.15 Modelo Simulink séptimo ensayo

En la Figura 2.16 se puede ver el resultado de este control en cascada. Se ha de tener en cuenta que a la corriente consigna que aparece en la gráfica no se le ha aplicado la saturación para comprobar que, aunque el lazo de velocidad pida más corriente, la real nunca sobrepasa los límites establecidos. Si bien la saturación de corriente se realiza de manera adecuada, se hace visible que el control no funciona de manera correcta. Cuando las corrientes toman valores cercanos a cero, el control no trabaja adecuadamente, provocando un salto tanto en las corrientes como en la velocidad. Estos saltos hacen que no se alcance nunca el valor de consigna deseado y aparezca una respuesta oscilante.

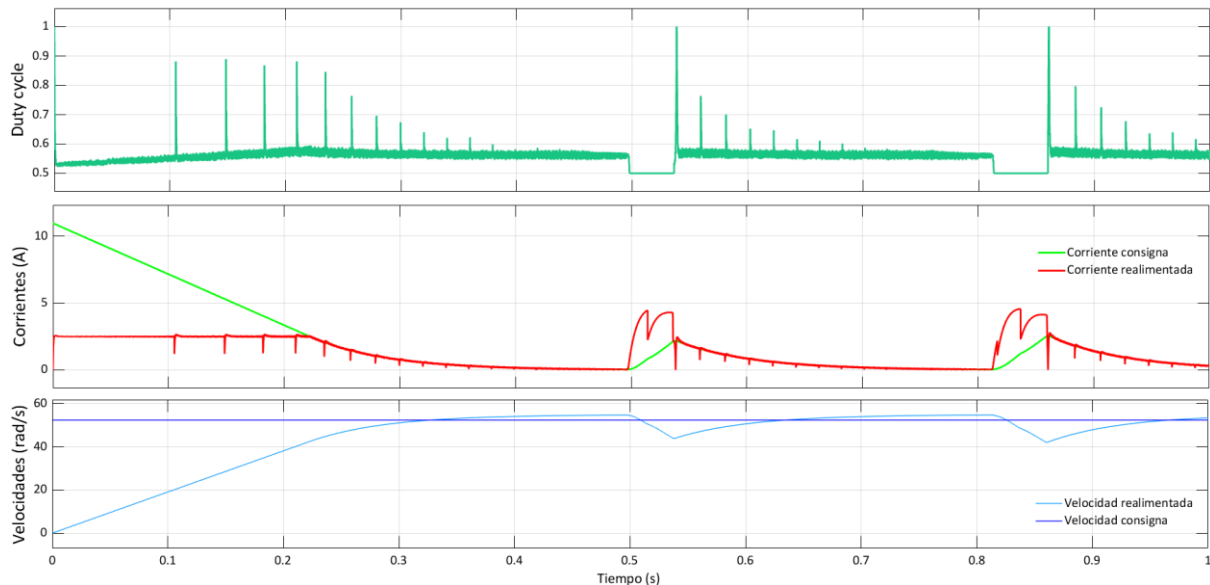


Figura 2.16 Control en cascada según el modelo del séptimo ensayo

2.5.8 Octavo ensayo: Realimentación de corriente

Después de revisar las distintas etapas del modelo, se comprueba que el error observado reside en la forma en que se toma la realimentación de corrientes. La corriente realimentada se calcula a partir de las corrientes en las tres fases del motor. Para que el control obtenga el valor de la corriente se ha de tener en cuenta que no se puede utilizar solo el valor de corriente de una fase ya que dependiendo del sector esta será positiva, negativa o nula así que la solución que se implementó en un principio fue tomar la suma de los valores absolutos de las tres corrientes y dividirlo entre dos. De este modo, dado que siempre habrá una fase en conducción positiva y una en conducción negativa, se obtiene el valor de la corriente que se aplica a la fase que en ese momento sea positiva. Esto, sin embargo, constituye un error, dado que no se podrán tener en cuenta corrientes negativas. Es por eso por lo que el modelo no se comportaba de manera adecuada en el ensayo anterior.

Para enmendar este error se debe emplear una nueva forma de realimentar la corriente. Esta vez se realiza una suma de la corriente de cada fase multiplicada por un cociente dependiendo del sector en el que se encuentre el motor. En caso de que la corriente de una fase concreta deba ser positiva según la Figura 2.4 en un sector en concreto, se multiplicara por uno, en caso de que deba ser negativa se multiplicara por menos uno y en caso de que deba ser nula se multiplicará por el mismo coeficiente que tenía en el sector anterior. Después, la suma de las tres fases deberá dividirse entre dos. De esta

manera se tienen en cuenta corrientes positivas y negativas. El modelo de este ensayo es igual que el empleado en el ensayo anterior con la única diferencia del cambio en la realimentación de corriente. En la Figura 2.17 se comprueba cómo, una vez cambiado el modelo se pueden alcanzar corrientes negativas y se consigue estabilizar la corriente y velocidad al valor de consigna.

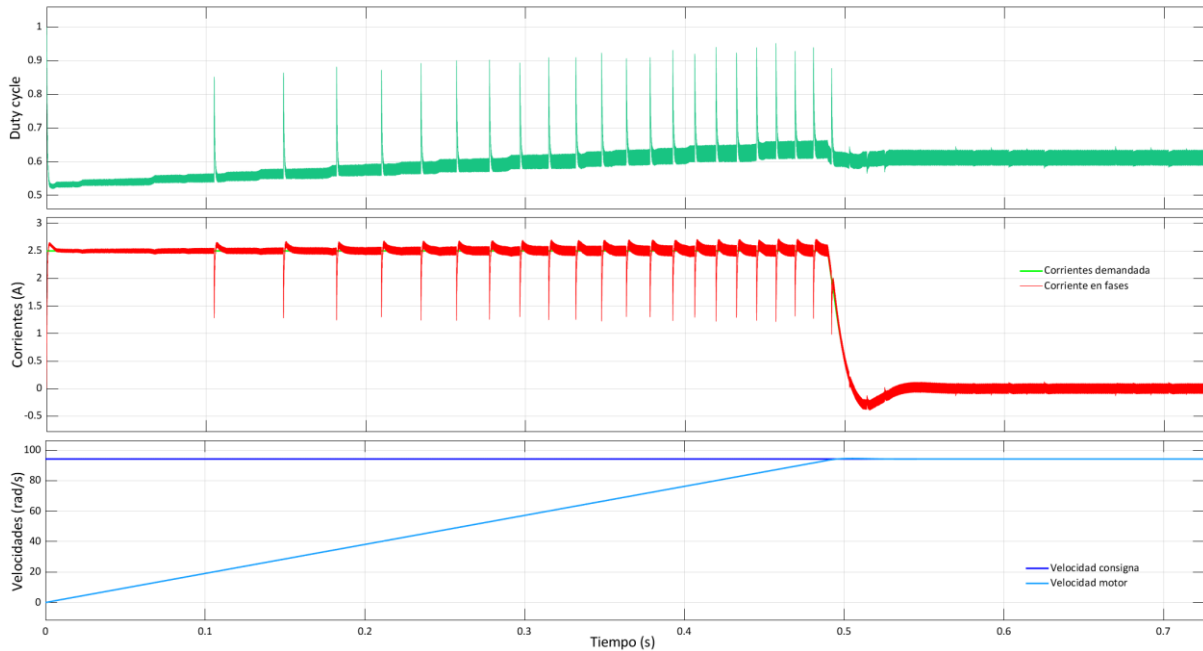


Figura 2.17 Control de la velocidad en cascada según el modelo del octavo ensayo

El ajuste del sistema PI en el lazo de velocidad se ha realizado siguiendo un método experimental. Primero se comienza con valores conservadores y se va aumentando la ganancia proporcional y disminuyendo la acción integral hasta obtener resultados aceptables, con buenos tiempos de establecimiento y poca sobreoscilación.

A continuación, se muestran algunos ensayos realizados para obtener el ajuste del regulador PI de velocidad con el mejor comportamiento dinámico posible. Todos los ensayos se realizan para una velocidad de consigna de 100 rpm. Primero se comienza solo con acción proporcional. En este caso se empieza con un valor de ganancia proporcional de valor 0,5. El resultado de este primer ensayo se muestra en la Figura 2.18.

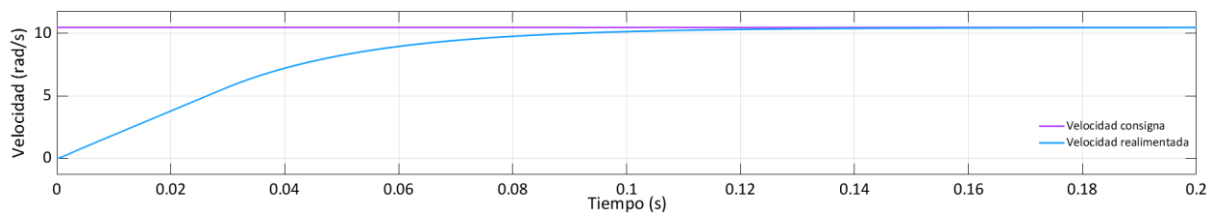


Figura 2.18 Ajuste PI velocidad, $K_p = 0,5$, $T_i = \infty$

Se puede observar como la velocidad realimentada alcanza el valor de consigna, sin embargo, se debe seguir aumentando la ganancia para comprobar si es posible reducir el tiempo de establecimiento. El nuevo valor de la ganancia proporcional es 1. Con esta nueva ganancia el comportamiento es el mostrado a continuación en la Figura 2.19.

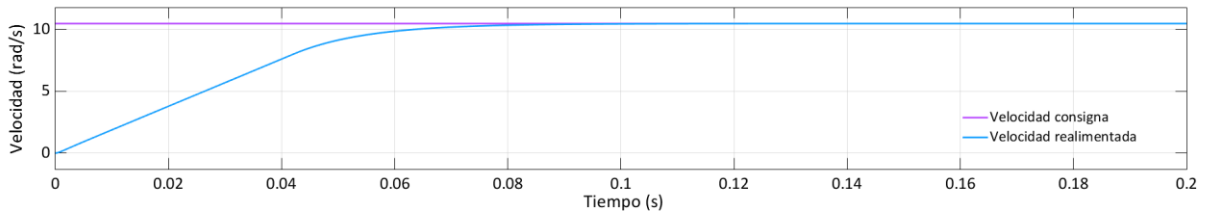


Figura 2.19 Ajuste PI velocidad, $K_p = 1$, $T_i = \infty$

Se hace notable en este ensayo una mejoría en el tiempo de establecimiento. Por tanto, se sigue aumentando la ganancia, esta vez hasta un valor de 2.

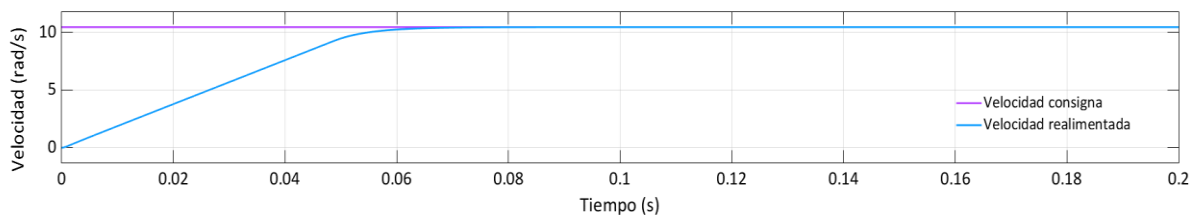


Figura 2.20 Ajuste PI velocidad, $K_p = 2$, $T_i = \infty$

La Figura 2.20 pone de manifiesto una leve mejoría en el tiempo de establecimiento. También se hace notar que la pendiente en ambos ensayos es la misma. Teniendo en cuenta esto, se procede ahora a disminuir progresivamente el tiempo integral hasta que aparezca una leve sobreoscilación. El siguiente ensayo se toma con los valores $K_p = 2$ y $T_i = 0,1$ y puede verse en la Figura 2.21. Se puede observar que no se aprecia ninguna diferencia notable con el anterior ensayo.

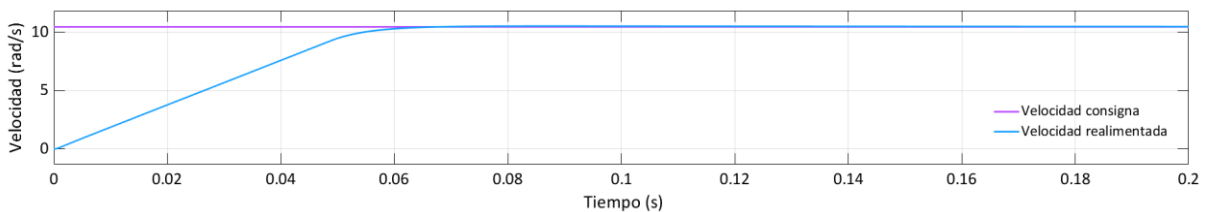


Figura 2.21 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.1$

Puesto que la acción integral introducida no ha mejorado la dinámica, se sigue disminuyendo el tiempo integral. Para el siguiente ensayo los valores son: $K_p = 2$ y $T_i = 0,01$. El resultado se muestra en la Figura 2.22.

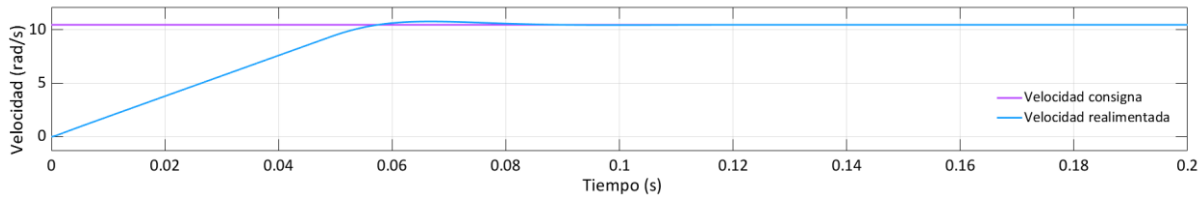


Figura 2.22 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.01$

Se observa un mejor comportamiento con la sobreoscilación buscada. De momento es el mejor ajuste. Sin embargo, se sigue reduciendo el tiempo integral hasta 0,005 para comprobar que no mejora la respuesta del sistema. El nuevo ensayo se puede ver en la Figura 2.23. Se observa en este último ajuste que la sobreoscilación es demasiado grande y no se mejora el tiempo de establecimiento.

Con todos los ensayos anteriores, el ajuste que proporciona la mejor respuesta tiene como parámetros: $K_p = 2$ y $T_i = 0,01$.

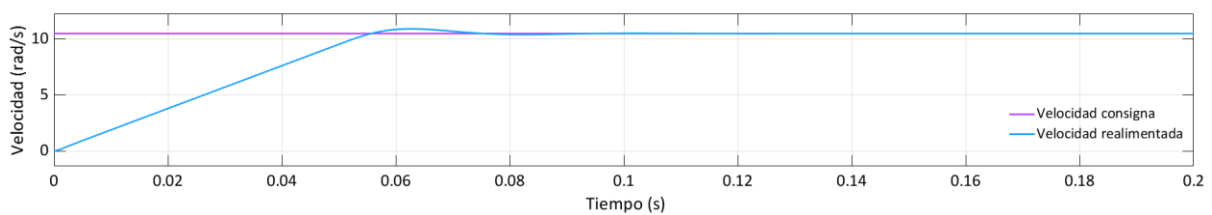


Figura 2.23 Ajuste PI velocidad, $K_p = 2$, $T_i = 0.005$

2.5.9 Noveno ensayo: Comprobaciones finales

En este último ensayo solo queda validar el modelo. Para ello, se piden al motor diversas consignas de velocidad, tanto positivas como negativas y se comprueba que se alcanzan en un tiempo aceptable, que el motor es capaz de estabilizarse en esos valores de velocidad, y, por último, que en ningún momento se supera el valor límite de corriente, 2,5A. Los resultados se muestran en la Figura 2.24.

Como era de esperar, se cumplen todos los objetivos del control de la velocidad del motor. Al pedir al sistema un salto de consigna de velocidad, el primer lazo devuelve un valor de consigna de corriente saturado que provoca en el lazo de corriente un salto de duty cycle. El sistema se mantiene con la corriente saturada hasta que la velocidad del motor se acerca a la velocidad consigna. En este momento, el lazo de velocidad disminuye la salida de consigna de corriente llegando a hacerla negativa para posteriormente estabilizarse en valores cercanos a cero. Esto se traduce en un establecimiento correcto de la velocidad de consigna tanto para valores positivos como negativos.

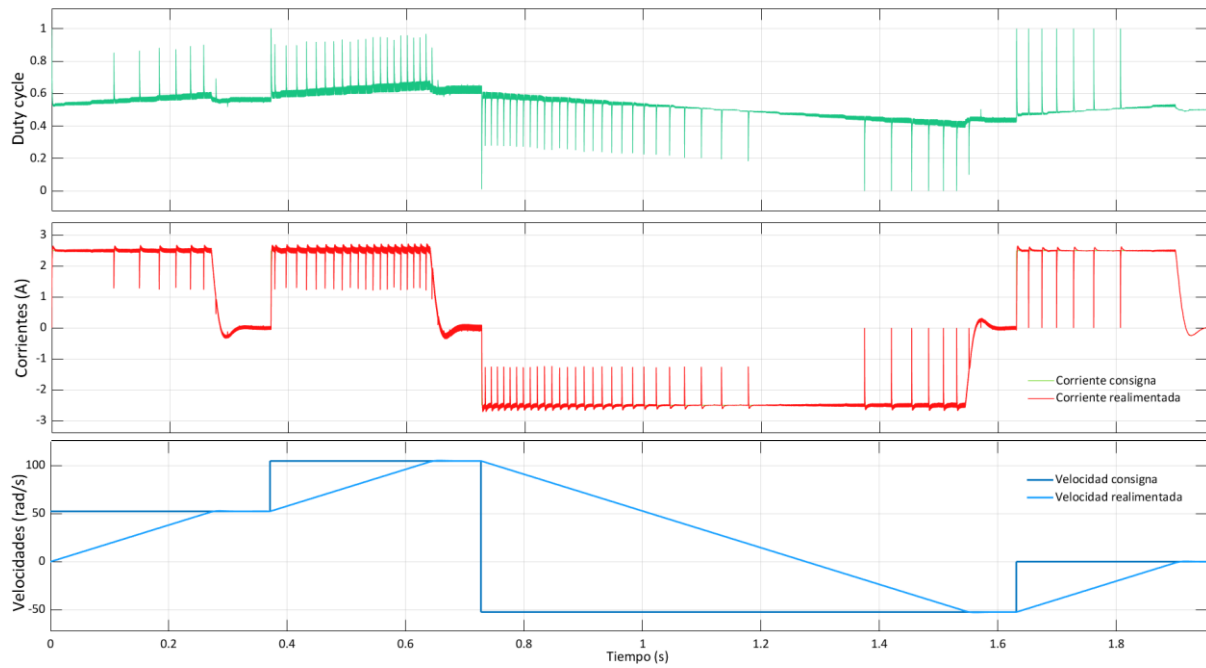


Figura 2.24 Resultados del control final en Simulink

Para ilustrar el modelo final del control, se realiza una descripción del mismo basado en los sistemas de los que se compone¹.

El primer sistema es el conformado por los reguladores PI junto con los correspondientes mecanismos de saturación y antiwindup. El esquema se muestra en la Figura 2.25.

¹ Se exponen solo los sistemas externos, los subsistemas y ecuaciones internas se muestran en los anexos

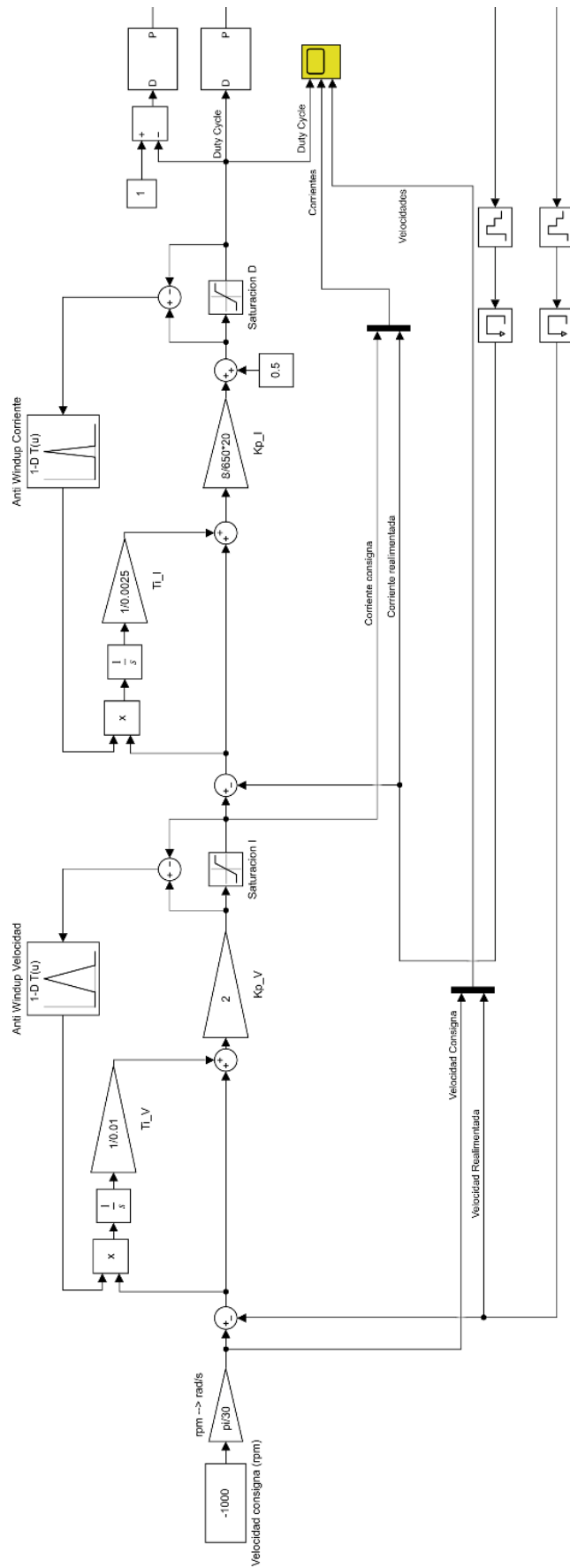


Figura 2.25 Esquema de reguladores PI en Simulink

Las conexiones que llegan desde la parte inferior son las señales de velocidad y corriente realimentada. Puede observarse como tras ambos lazos de control, la salida, es decir, el duty cycle se lleva a los generadores de PWM. Cabe comentar en este apartado que el control se realiza siempre sobre la velocidad en radianes por segundo. Por ese motivo, esas eran las unidades mostradas para la velocidad en las anteriores gráficas.

El segundo sistema se muestra en la Figura 2.26. Se trata del inversor, que recibe las señales para la apertura de sus interruptores tras hacer pasar las señales PWM por un subsistema encargado de evaluar el sector y asociar a cada rama del inversor la señal adecuada. En este sistema se pueden ver además los amperímetros junto al subsistema encargado de la realimentación de corriente.

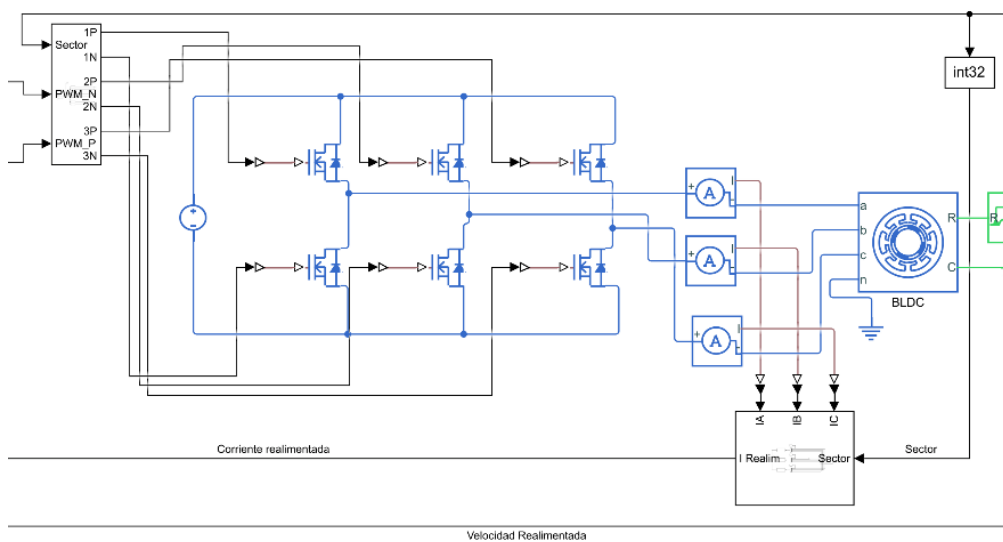


Figura 2.26 Esquema de inversor y realimentación de corriente en Simulink

El siguiente sistema, Figura 2.26, muestra la realimentación de velocidad, posición y sector. Este sistema está formado por bloques matemáticos que permiten obtener una medida ideal de estos parámetros. Sin embargo, en la realidad, el sensor de posición tiene una cierta precisión, lo que hace que los datos siempre presenten un cierto rango de tolerancia.

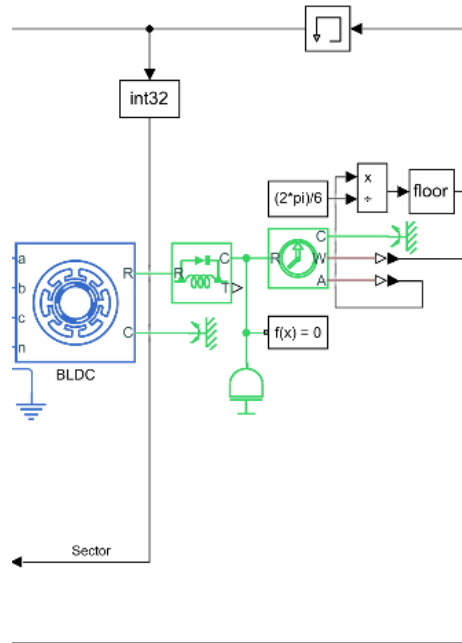


Figura 2.27 Esquema de realimentación de velocidad, posición y sector

En los anexos del trabajo se incluye una imagen del esquema completo junto con la información detallada del funcionamiento de los subsistemas.

2.6 DESCRIPCION DEL SISTEMA REAL

Con todos los ensayos anteriores se ha podido observar cómo, a través del uso de ciertos elementos como sensores, inversores o reguladores PI es posible realizar el control de velocidad de un motor BLDC. El funcionamiento del sistema empleado se puede ver resumido en el esquema de la Figura 2.28.

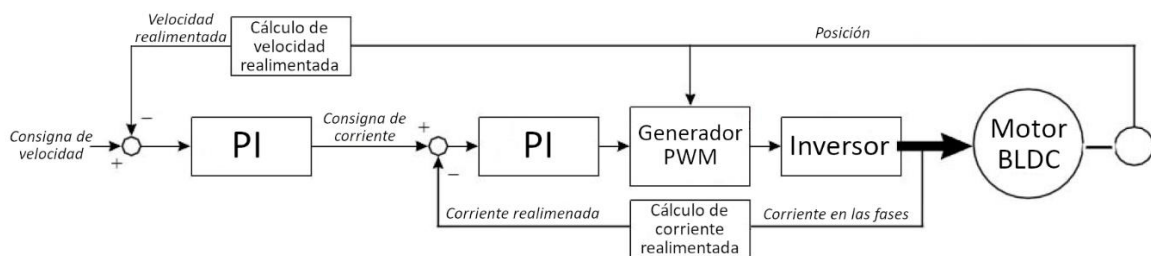


Figura 2.28 Esquema del sistema para el control de velocidad del motor BLDC (modificado de [3])

Para llevar este sistema a la práctica se ha de hacer una revisión de las medidas necesarias sobre el motor. Estas medidas son: medida de la corriente de al menos dos fases, medida de la posición para

poder obtener la información sobre el sector y, por último, medida de la velocidad. Además, en este sistema es necesario un inversor, un motor BLDC, dos reguladores PI y un generador de PWM.

Para la medida de las corrientes en las fases son necesarios dos sensores de efecto Hall, para la medida de posición y velocidad se necesita un sensor de posición angular, y, para la generación de señales PWM, reguladores PI y la coordinación entre los diferentes elementos se puede emplear un microcontrolador.

En resumen, los elementos necesarios para realizar el control mostrado en este apartado son:

- Un motor BLDC
- Un inversor de tres fases
- Un sensor de posición angular
- Dos sensores de efecto Hall
- Un microcontrolador

A pesar de ser estos los elementos estrictamente necesarios, en los próximos capítulos se verá cómo, para realizar ciertas tareas, como por ejemplo la comunicación entre microcontrolador e inversor se introducen elementos extra como puertas lógicas.

CAPÍTULO 3: PLACA DEL MICROCONTROLADOR

3.1 INTRODUCCIÓN

Tras analizar los ensayos realizados en Simulink y determinar los elementos necesarios para llevar a cabo el control del motor, en este capítulo se describirán los distintos componentes empleados y se mostrará la forma de conexión entre ellos para garantizar el funcionamiento del sistema de control del motor BLDC.

3.2 ESTRUCTURA

Primero se realizará una breve descripción de las funciones que debe realizar el microcontrolador, así como las características que debe presentar para, posteriormente, realizar la selección del más adecuado. Esto se desarrolla en el apartado DESCRIPCIÓN DEL MICROCONTROLADOR. Después, en el apartado SENSORES DE CORRIENTE, se tratarán los Sensores de efecto Hall para la realimentación de corriente, y se comentará como son capaces de enviar la información de las corrientes al microcontrolador. Tras esto, se desarrolla el apartado SEÑALES DE CONTROL DE INVERSOR, en el cual se pone de manifiesto el modo de conexión de la placa del microcontrolador con el inversor y como se transmite la información entre ellos. Más adelante, se comentará el modo en el que se hace posible la alimentación de las fases del motor, describiendo la conexión de la placa del microcontrolador con la etapa de potencia en el apartado CONEXIONES DE POTENCIA. Por último, en el apartado MONTAJE DE LA PLACA DEL MICROCONTROLADOR se mostrará la placa del microcontrolador en su totalidad indicando todos sus elementos y la conexión entre ellos.

3.3 DESCRIPCIÓN DEL MICROCONTROLADOR

El microcontrolador es el centro del sistema, es el encargado de recibir señales de entrada de datos como las corrientes, posición y velocidad y enviar señales de salida al inversor. Estas señales de salida son señales de control que, en algunos casos, incluyen una modulación por ancho de pulsos. Además, el microcontrolador debe permitir recibir y enviar información de manera inalámbrica a un dispositivo Android desde el cual se controlará el motor. Esta conexión se realizará por Bluetooth. Por estos motivos, se debe realizar una elección del microcontrolador que permita realizar estas tareas con la mayor precisión y fiabilidad posible.

Los microcontroladores disponibles para estos propósitos se pueden clasificar según las marcas que los fabrican. Por una parte, encontramos los microcontroladores Arduino, los cuales están enfocados a educación o bien tareas simples. La ventaja de estos microcontroladores es la gran información y soporte que existe sobre ellos y la compatibilidad nativa con su propio software, el Arduino IDE. Otra opción son los microcontroladores fabricados por Espressif, los cuales están orientados a aplicaciones industriales y del internet de las cosas. Dentro de este grupo, encontramos los kits basados en el ESP8266 [4] y los basados en el ESP32 [5], el cual es una evolución del anterior. Para seleccionar el microcontrolador, se han escogido tres modelos con distintas prestaciones y situados en el mismo

rango de precios. A continuación, se muestra una comparativa de las características más relevantes para nuestro sistema que presentan estos tres microcontroladores, los cuales son: un nRF52840 [6], que implementan los kits Arduino Nano 33 BLE, un ESP32 y un ESP8266.

En la Tabla 3.1 se ponen de manifiesto diversas diferencias entre controladores. La primera de ellas es que el microcontrolador ESP8266 no presenta módulo Bluetooth, por tanto, queda descartado de las opciones, quedando solo el ESP32 y el nRF52840. Se puede observar que el ESP32 presenta una mayor velocidad de reloj y mayor resolución en los canales asociados a la PWM. Además, también es compatible con el software Arduino IDE mediante el uso de librerías específicas proporcionadas por el fabricante. Por estos motivos se elige el microcontrolador ESP32 para el control del sistema. Para emplear el microcontrolador en nuestro sistema, se utiliza la placa ESP32 DEVKIT V1, de la cual se muestra en la Figura 3.1 un esquema junto con el Pinout.

Para asegurar que la resolución de 16 bits para los canales de PWM es adecuada, se puede hacer una estimación de la precisión en corriente que proporciona. Asumiendo un rango de alimentación de las fases del motor con 650 voltios, dado que el duty cycle permite elegir una tensión de la fase entre menos la tensión de bus continua y la tensión de bus, para un valor entre 0 y 1, y una resistencia por fase de 2Ω , lo cual, como se verá más adelante es aceptable, se puede obtener la precisión en corriente al trabajar con una resolución determinada para los canales PWM.

Dado que con el número de niveles que sea capaz de proporcionar la regulación PWM lo que cambia es el nivel de tensión que se aporta a las fases, la resolución en voltaje para 16 bits de resolución en la señal PWM se calcula como:

$$\text{Resolucion Voltaje (16 bits)} = \frac{650V}{2^{16}} = 0,00992 V$$

Para el cálculo de la resolución en corriente solo hace falta tener en cuenta la resistencia por fase.

$$\text{Resolucion Corriente (16 bits)} = \frac{0,00992}{2\Omega} = 0,00496 A = 4,95 mA$$

Es lógico aceptar esta resolución para el control a realizar. También se puede justificar con estos cálculos el descarte del microcontrolador nRF52840, ya que con 8 bits de resolución para la PWM solo se podría obtener una resolución en corriente de más de 1 amperio, lo cual no es aceptable para un control fino en una máquina con una corriente nominal de unos 2 A. El cálculo se muestra en la siguiente ecuación:

$$\text{Resolucion Corriente (8 bits)} = \frac{650V}{2^8 * 2\Omega} = 1,26 A$$

| Microcontrolador: | nRF52840 (Arduino) | ESP 32 | ESP 8266 |
|---|-----------------------|--------|----------|
| Velocidad de reloj máxima (MHz) | 64 | 240 | 160 |
| Modulo Bluetooth | Si | Si | No |
| Resolución ACD (convertidor analógico digital) (bits) | 12 | 12 | 10 |
| Resolución PWM (bits) | 8 | 16 | 10 |
| Precio | 17,50€ | 10€ | 8,29€ |

Tabla 3.1 Comparativa entre microcontroladores [4, 5, 6]

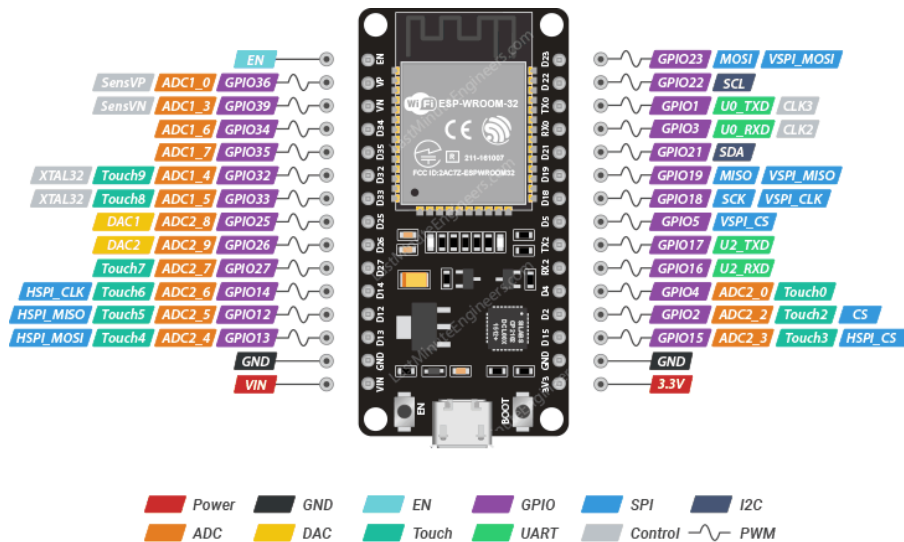


Figura 3.1 Esquema y Pinout del microcontrolador ESP32 DEVKIT V1 [7]

3.4 SENSORES DE CORRIENTE

Para realizar la realimentación de corriente, son necesarios al menos dos sensores que midan las corrientes en dos de las fases, puesto que la corriente circulante por la tercera fase puede obtenerse conocidas las otras dos al utilizar una conexión a tres hilos con los terminales del motor. Para estos sensores, se opta por utilizar dos sensores de efecto Hall LISR 6-NP [8]. Puesto que estos sensores presentan una corriente nominal de 6A eficaces, con un rango de medida extendido hasta 2,25 veces el valor máximo de la corriente nominal, y la corriente nominal de las fases es 2,1A, los cables deberán enrollarse para dar tres vueltas sobre el sensor. Estos sensores proporcionan una tensión proporcional a la corriente circulante a través de ellos. Este modelo en concreto presenta una sensibilidad de 104,16mV/A. La tensión que devuelve el sensor es transferida a los pines de lectura analógica del

microcontrolador, los cuales transforman la información de tensión en un dato de 12 bits. Para obtener la relación entre la corriente que circula por las fases y el dato digital que proporciona el microcontrolador primero se realizará un cálculo teórico, con el cual se mostrará el modo en que se transfiere la información desde el sensor hasta el microcontrolador y posteriormente se realizará un ensayo para comprobar la validez del modelo teórico.

Para empezar con el modelo teórico es necesario conocer la función de transferencia corriente/voltaje proporcionada por el fabricante, la cual se muestra en la Figura 3.2. Aparte, el fabricante nos indica que la corriente nominal I_{pn} , corresponde a un valor de seis amperios. Sin embargo, como se ha comentado, en este caso los cables dan tres vueltas al sensor, por lo que esta corriente nominal será igual a dos amperios. La función de transferencia se muestra en la siguiente ecuación:

$$V = 2,5 \pm \left(0,625 \times \frac{I}{I_{pn}} \right)$$

Considerando $I_{pn} = 2A$

$$V = 2,5 \pm (0,3125 \times I)$$

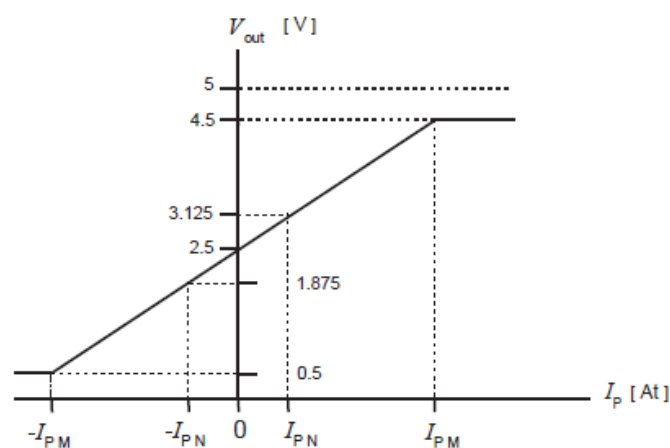


Figura 3.2 función de transferencia corriente/voltaje del sensor de corriente LTSR 6-NP [8]

Para obtener la función de transferencia asociada al microcontrolador hemos de conocer su funcionamiento en cuanto a lectura de datos analógicos se refiere. Los puertos de lectura analógica del microcontrolador pueden leer tensiones en un rango de 0 a 3,3V y devuelven valores digitales de 12 bits, es decir, en un rango de 0 a 4095. Cabría esperar que la ley de transferencia fuera lineal entre voltaje y dato digital, siguiendo la siguiente ecuación:

$$Dato (12 bits) = V \times \frac{4095}{3,3}$$

Y en general es cierto, pero se ha de tener en cuenta que, en la práctica, la función de transferencia no es completamente lineal en todo el rango de tensiones. La Figura 3.3 pone de manifiesto el comportamiento real de la lectura de datos, el cual se aleja de la linealidad conforme las tensiones se acercan al extremo superior del rango.

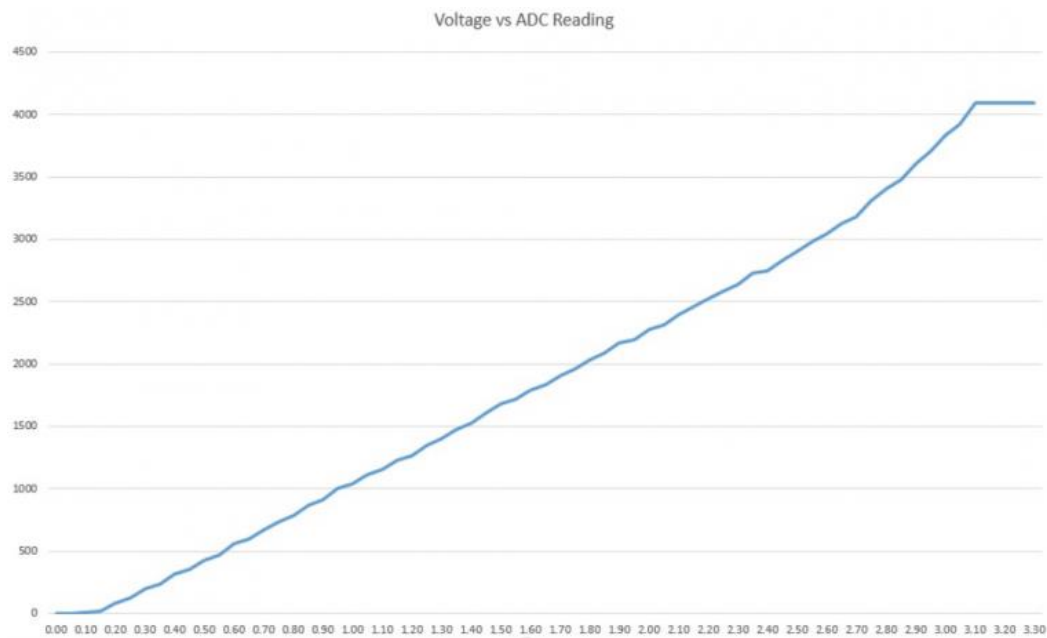


Figura 3.3 función de transferencia experimental para la lectura analógica con ESP32 [9]

Si bien es cierto que el comportamiento no es totalmente lineal, para la ecuación teórica se asumirá que el error al considerarlo lineal es mínimo. Operando las anteriores expresiones, la función de transferencia que relaciona directamente la corriente que circula por las fases del motor con el dato enviado por el microcontrolador es la siguiente:

$$\text{Dato (12 bits)} = 3102,2727 \pm 387,784 \times I(A)$$

Además, para adaptar el rango de salida del transductor de corriente (0 – 5 V) al de las entradas analógicas del ESP32 (0 – 3,3 V) se ha integrado un divisor de tensión de relación 2/3 entre el transductor y el ESP32. Como consecuencia de ello, la función transferencia teórica es:

$$\text{Dato (12 bits)} = 2068,2 \pm 258,5 \times I(A)$$

Una vez obtenida la ecuación teórica se debe contrastar mediante un ensayo que nos permita obtener de forma experimental la curva de calibración entre corriente y dato digital de 12 bits.

Para ello, el ensayo a realizar consistirá en alimentar con una fuente de alimentación las fases del motor y medir simultáneamente el amperaje introducido y el valor enviado por el microcontrolador de los sensores de corriente. Los resultados obtenidos en este ensayo para los sensores de la fase A y la fase B se muestran en la Figura 3.4 y la Figura 3.5 respectivamente. Combinando las ecuaciones de los dos sensores se puede obtener una ecuación única con la media de ambas:

$$\text{Dato (12 bits)} = 1929 \pm 347,21 * I(A)$$

Despejando de la ecuación anterior se puede obtener la ecuación que permite conocer la intensidad a partir del dato proporcionado por el microcontrolador.

$$I(A) = 0,00288 * \text{Dato}(12 \text{ bits}) - 5,55572$$

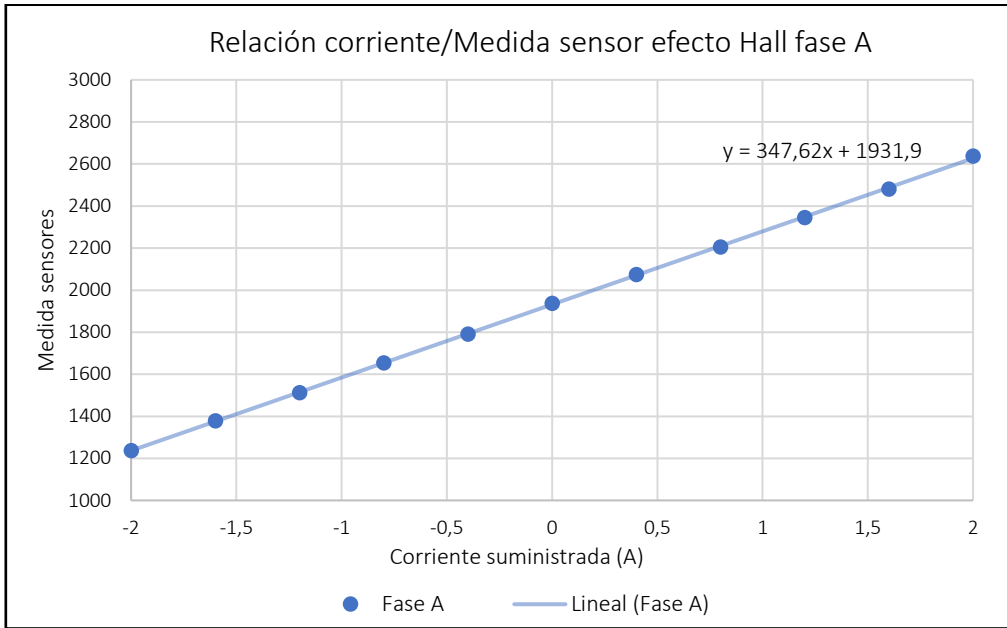


Figura 3.4 relación corriente/dato digital del sensor para fase A

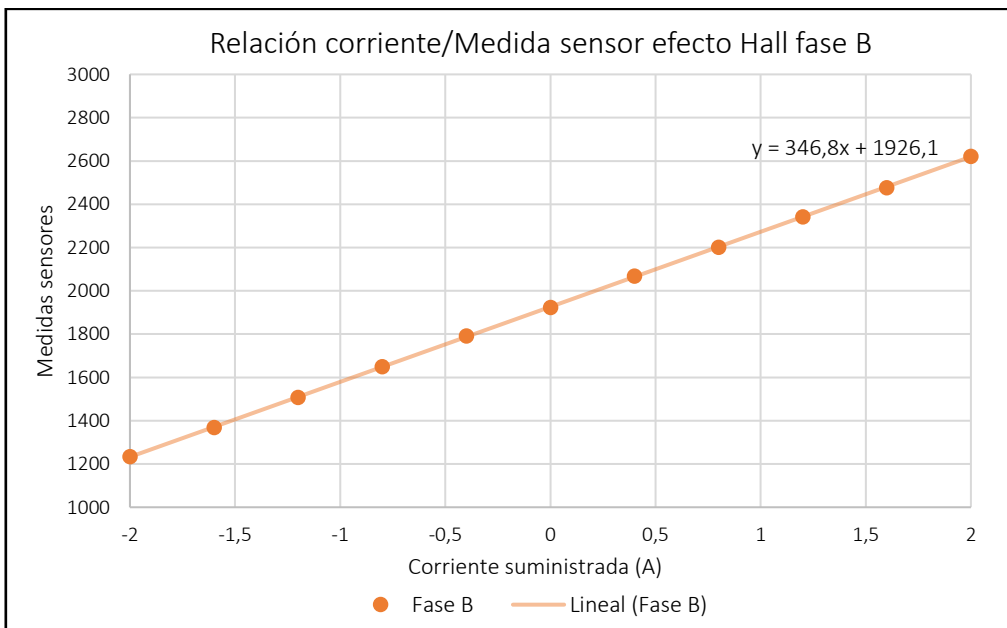


Figura 3.5 relación corriente/dato digital del sensor para fase B

Aunque la ecuación a emplear en el código sea la experimental, pues es la que tiene en cuenta todas las particularidades del montaje, es interesante comparar ambas funciones, experimental y teórica para comprobar que el funcionamiento de los sensores es el esperado. Esta comparación se muestra en la Figura 3.6. Como es de esperar, la variación entre función de transferencia experimental y teórica es mínima.

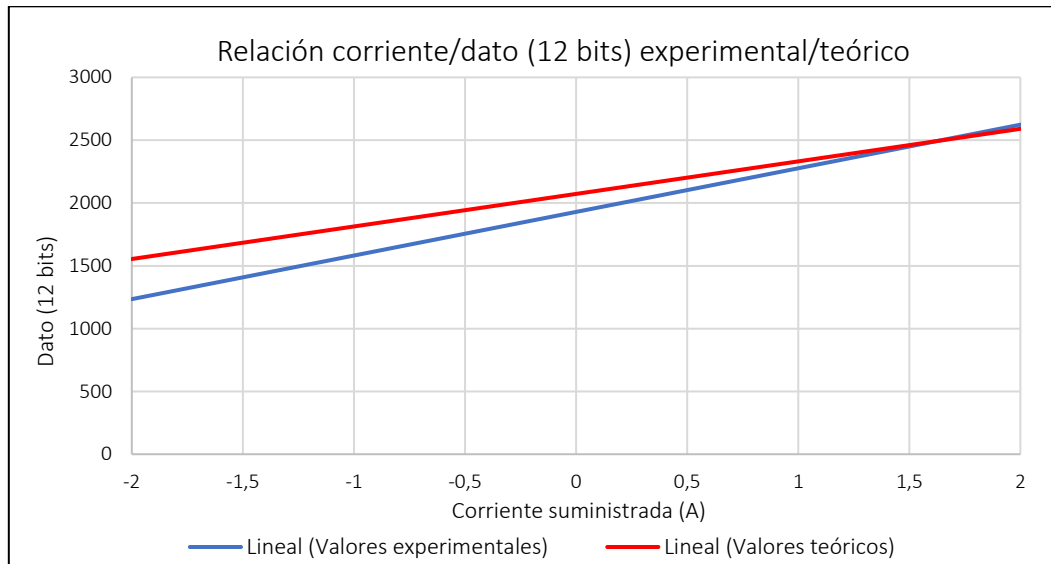


Figura 3.6 Comparación entre funciones de transferencia experimental y teórica

Es necesario comentar que, con este montaje, también se puede hallar el valor de la resistencia de las fases midiendo, además de los parámetros anteriores, la tensión proporcionada por la fuente de alimentación. Los resultados de este ensayo se muestran en la Figura 3.7, de la cual puede extraerse el valor de dos resistencias de fase en serie: 2,85 Ω .

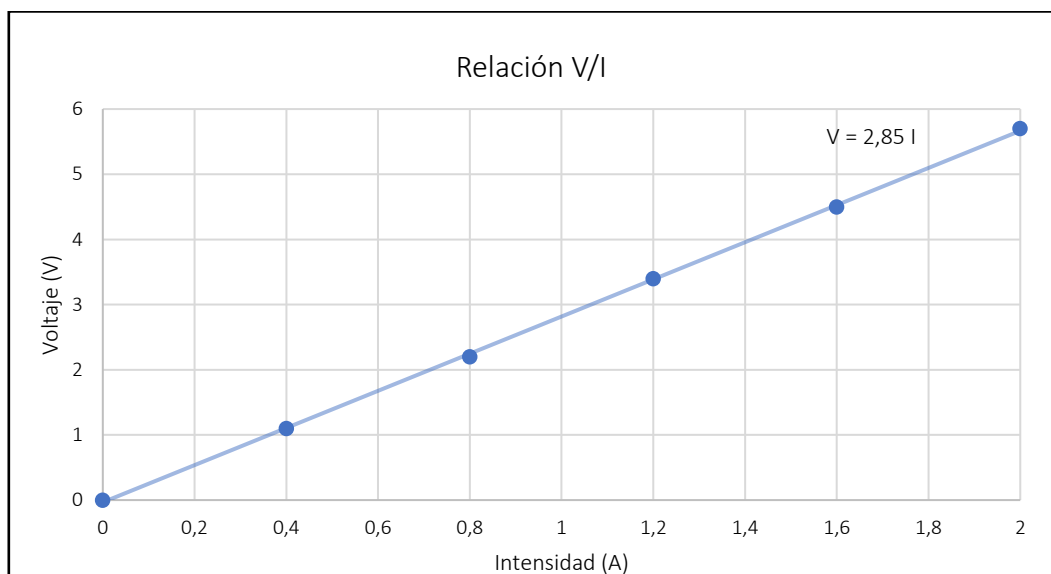


Figura 3.7 Relación tensión/corriente en las fases del motor

3.5 SEÑALES DE CONTROL DE INVERSOR

Para que el inversor funcione de manera correcta requiere de las señales de activación para sus seis interruptores MOSFET. Esta información será enviada en forma de señales PWM directamente desde el microcontrolador, el cual tendrá en cuenta el duty cycle actual, así como el sector en que se encuentre el rotor en cada momento. Desde el microcontrolador se generarán dos señales PWM, una basada en el duty cycle requerido y otra obtenida como la complementaria de la primera, es decir restando a la constante con valor unitario el valor del duty cycle de la primera señal PWM. El funcionamiento del motor Brushless DC requiere que en todo momento exista una fase a la cual se le aporta tensión positiva, la cual estará asociada con el duty cycle requerido, una fase con tensión negativa, asociada con el complementario del duty cycle requerido, y la última fase que debe quedar sin corriente y requerirá tensión nula en régimen permanente (fase del motor abierta, rama del inversor en bloqueo). En la Figura 3.8 se muestra un ejemplo de las señales que deben aportarse a cada MOSFET en el caso concreto del primer sector con un duty cycle mayor que 0,5. Para el resto de los sectores el funcionamiento es homólogo.

Para poder suministrar tensión nula a la fase correspondiente se emplea un mecanismo, que, haciendo uso de puertas lógicas tipo OR y señales de habilitación es capaz de bloquear las señales de encendido en los MOSFET de cada una de las ramas del inversor. El mecanismo se basa en hacer pasar todas las señales PWM dirigidas a los MOSFET negativos de cada rama por una puerta OR de dos pines. Al otro pin de la puerta lógica llega una señal de habilitación para esa parte negativa de la rama, la cual será nula en caso de conducción positiva o negativa de la fase (parte negativa de la rama habilitada) y tendrá un valor de 3,3 V, es decir, un uno lógico, cuando por esa rama se requiera bloquear la corriente (ambos lados + y - de la rama bloqueados). De esta manera, la puerta lógica saturará haciendo que el valor digital de la señal negada sea precisamente cero. En la Figura 3.9 se puede ver un esquema del sistema descrito, En concreto para la fase A.

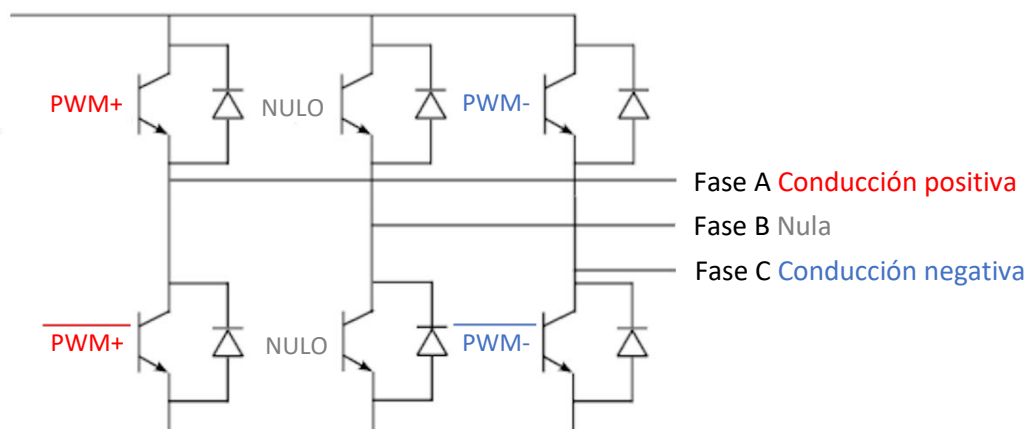


Figura 3.8 Ejemplo de señales aportadas al inversor en el sector 1 (elaboración propia)

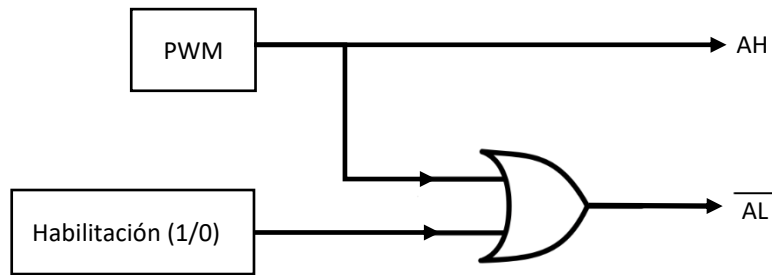


Figura 3.9 Esquema del circuito para las ramas del inversor (elaboración propia)

Para comprobar el correcto funcionamiento de este sistema de control para las ramas del inversor se realiza un ensayo en el cual se observa la salida de la puerta lógica OR en dos situaciones. En la primera las entradas a la puerta lógica son la propia señal PWM y la señal de habilitación a nivel bajo. En la segunda se cambia la habilitación y se pone a nivel alto. Los resultados se muestran en las siguientes figuras, la señal registrada en azul se corresponde a la salida de la puerta lógica y la registrada en rojo a la señal de habilitación. Se puede observar como el comportamiento es el esperado.

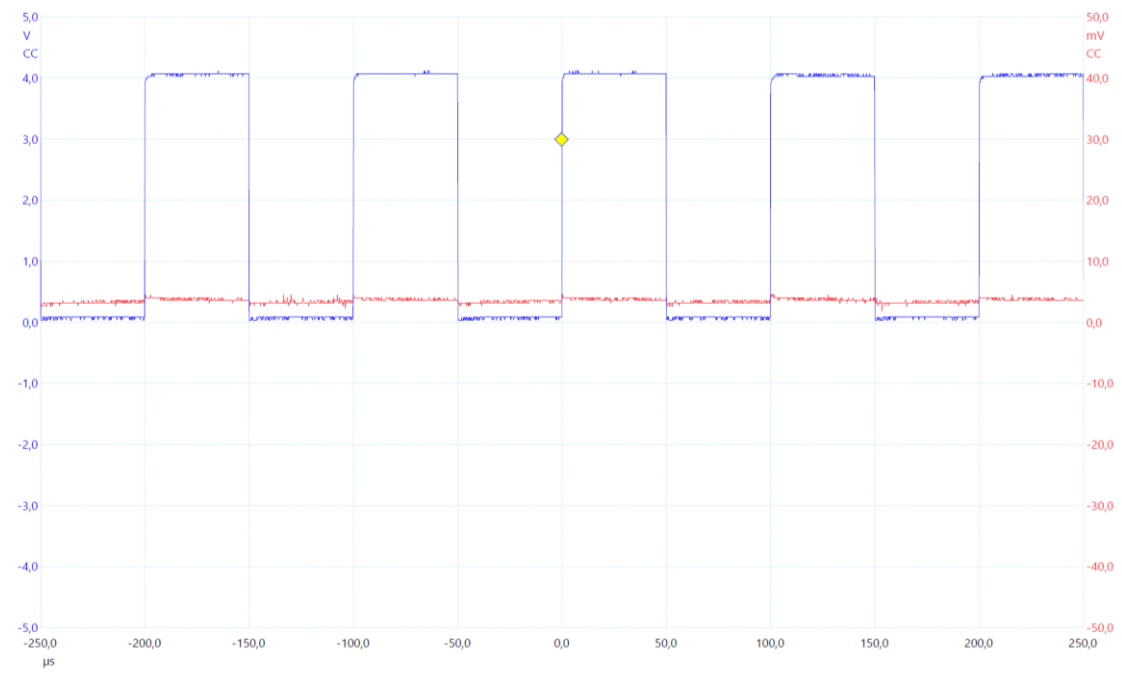


Figura 3.10 Salida de puerta OR con PWM y habilitación a nivel bajo

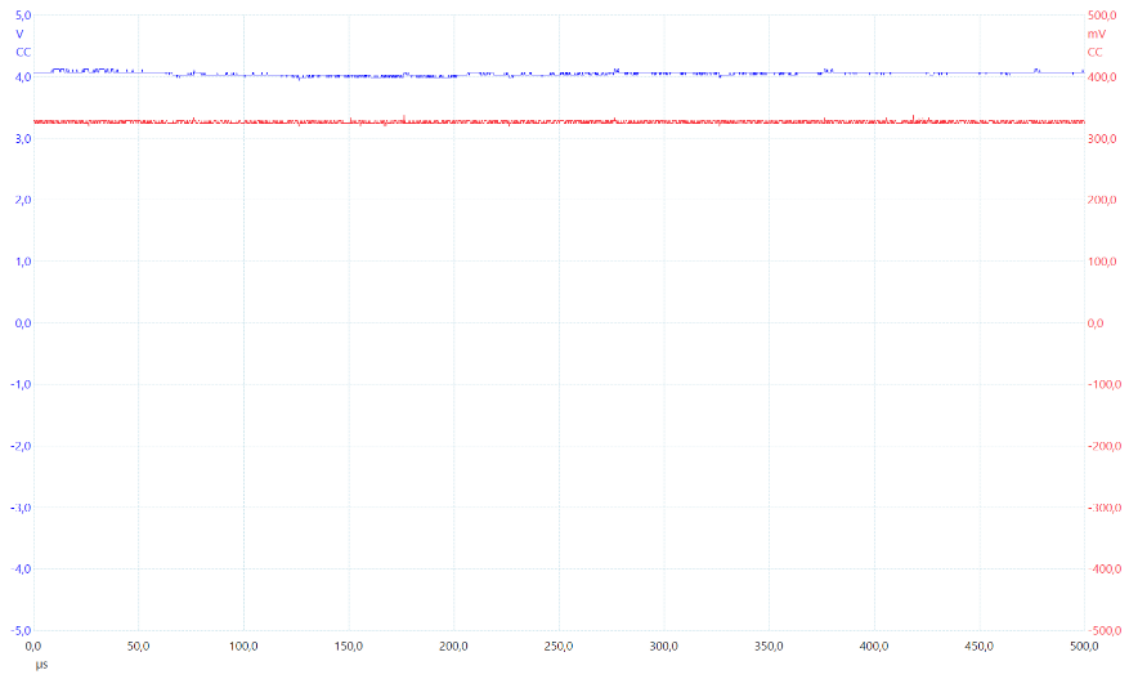


Figura 3.11 Salida de puerta OR con PWM y habilitación a nivel alto

3.6 CONEXIONES DE POTENCIA

Como se ha comentado en el apartado anterior, el microcontrolador envía las señales de habilitación y de PWM al inversor, y es el inversor el que suministra a las fases del motor la tensión necesaria. En la Figura 3.12 se puede observar el montaje físico del sistema con los diferentes elementos, así como las conexiones entre ellos.

Para ilustrar las conexiones existentes que permiten ofrecer la tensión necesaria a las fases del motor, se recorre el montaje desde la tensión obtenida directamente de la red eléctrica, hasta llegar a la tensión proporcionada al motor.

La tensión de la red eléctrica se toma como alimentación para el inversor, el cual tiene una conexión mediante un cable plano de 34 pines con la placa del microcontrolador. Esta conexión es la encargada de transmitirle al inversor la información aportada por el microcontrolador relativa al estado de cada rama de las fases del motor mediante las señales PWM y de habilitación. Con esta información, el inversor es capaz de proporcionar las tensiones necesarias a las fases del motor. Antes de llegar al motor, los cables de las fases se llevan a la placa del microcontrolador para poder obtener la realimentación de corriente. Una vez tomadas las medidas mediante los sensores de efecto Hall, la placa del microcontrolador alimenta las fases del motor desde su conexión de potencia.

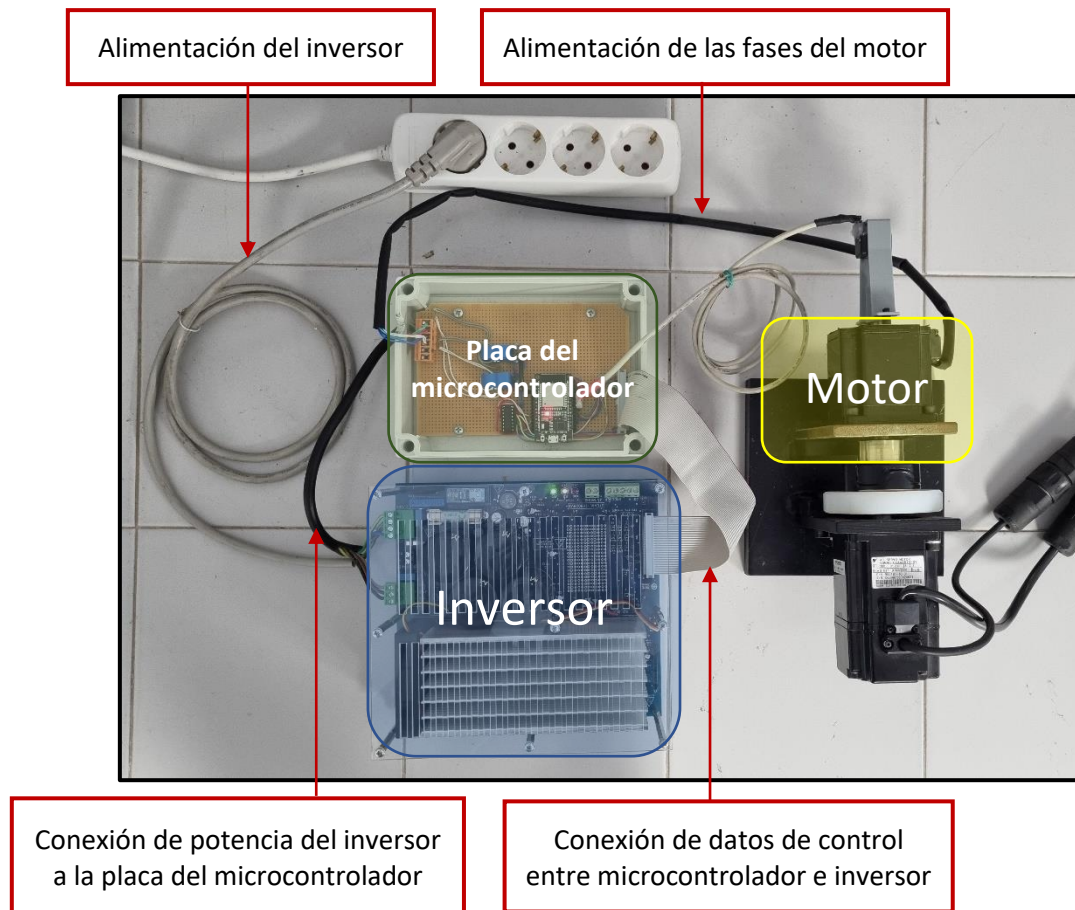


Figura 3.12 Esquema y conexiones del montaje en el laboratorio (elaboración propia)

3.7 MONTAJE DE LA PLACA DEL MICROCONTROLADOR

Conocidos todos los componentes, así como sus funciones y el modo de comunicación del microcontrolador con el inversor para transmitir las señales PWM y de habilitación, en este apartado se describe de forma detallada la forma de conexión de todos los elementos entre sí y con el microcontrolador. Para ilustrar el sistema de la placa del microcontrolador, se muestra en la Figura 3.13 una imagen del montaje real realizado en el laboratorio, en la cual se identifican todos los componentes.

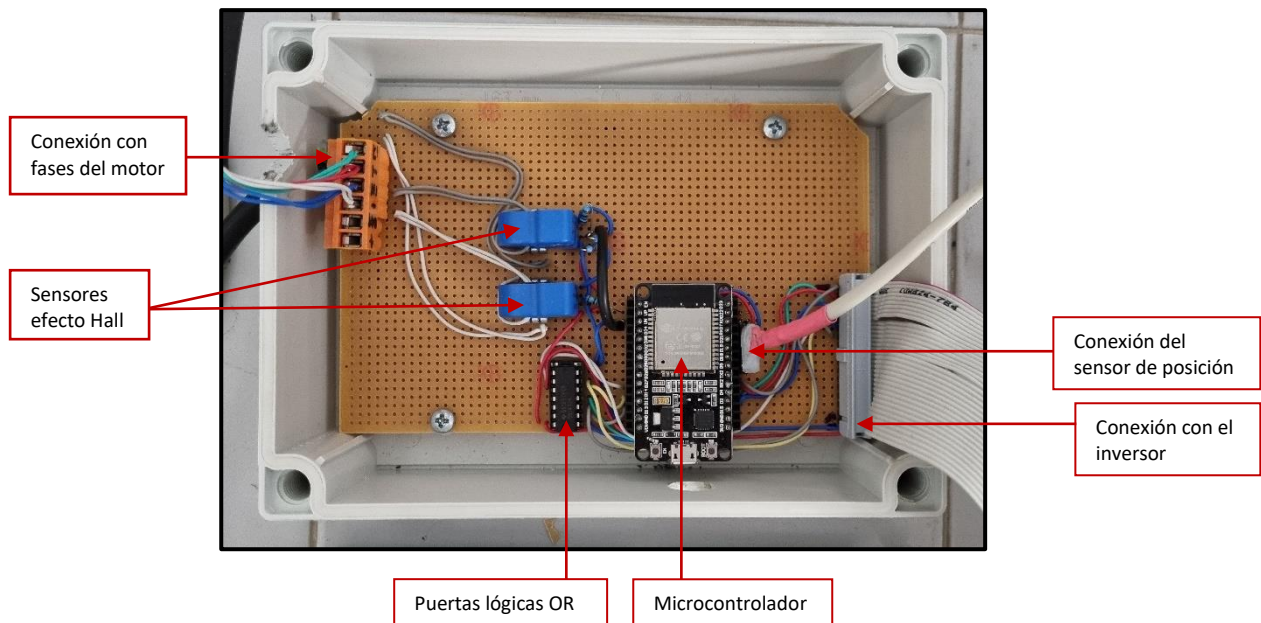


Figura 3.13 Montaje de la placa del microcontrolador (elaboración propia)

Para completar este apartado, se muestra a continuación un esquema eléctrico con todas las conexiones existentes entre componentes, Figura 3.14. Cabe mencionar que las posiciones relativas entre elementos no se corresponden a las posiciones que se pueden ver en el montaje real mostrado en la Figura 3.13. Esto es así para facilitar la visualización de las conexiones. Además, para favorecer aún más el entendimiento del esquema, se ha optado por asociar un código de colores a cada fase. El color azul representa todas las conexiones referentes a la fase A, el verde a la fase B y el violeta la fase C. Nótese además que todas las conexiones a la tensión de alimentación de 5V provienen de la salida proporcionada por el inversor, correspondiente al pin Vdd_micro.

Para complementar el esquema anterior, las conexiones de los distintos elementos con el microcontrolador se describen a continuación mostrando para cada sistema los pines del microcontrolador empleados. La numeración de los pines del microcontrolador se puede ver en la Figura 3.1.

- *Realimentación de corriente*
 - Lectura de corriente fase A: GPIO 39
 - Lectura de corriente fase B: GPIO 36
- *Realimentación de posición*
 - Clock (CLK): GPIO 18
 - Digital Output (DO): GPIO 19
 - Chip Select (CS): GPIO 5
- *Señales PWM*
 - PWM fase A: GPIO 4
 - PWM fase B: GPIO 2
 - PWM fase C: GPIO 15
- *Señales de habilitación*
 - Habilitación fase A: GPIO 14
 - Habilitación fase B: GPIO 12
 - Habilitación fase C: GPIO 13

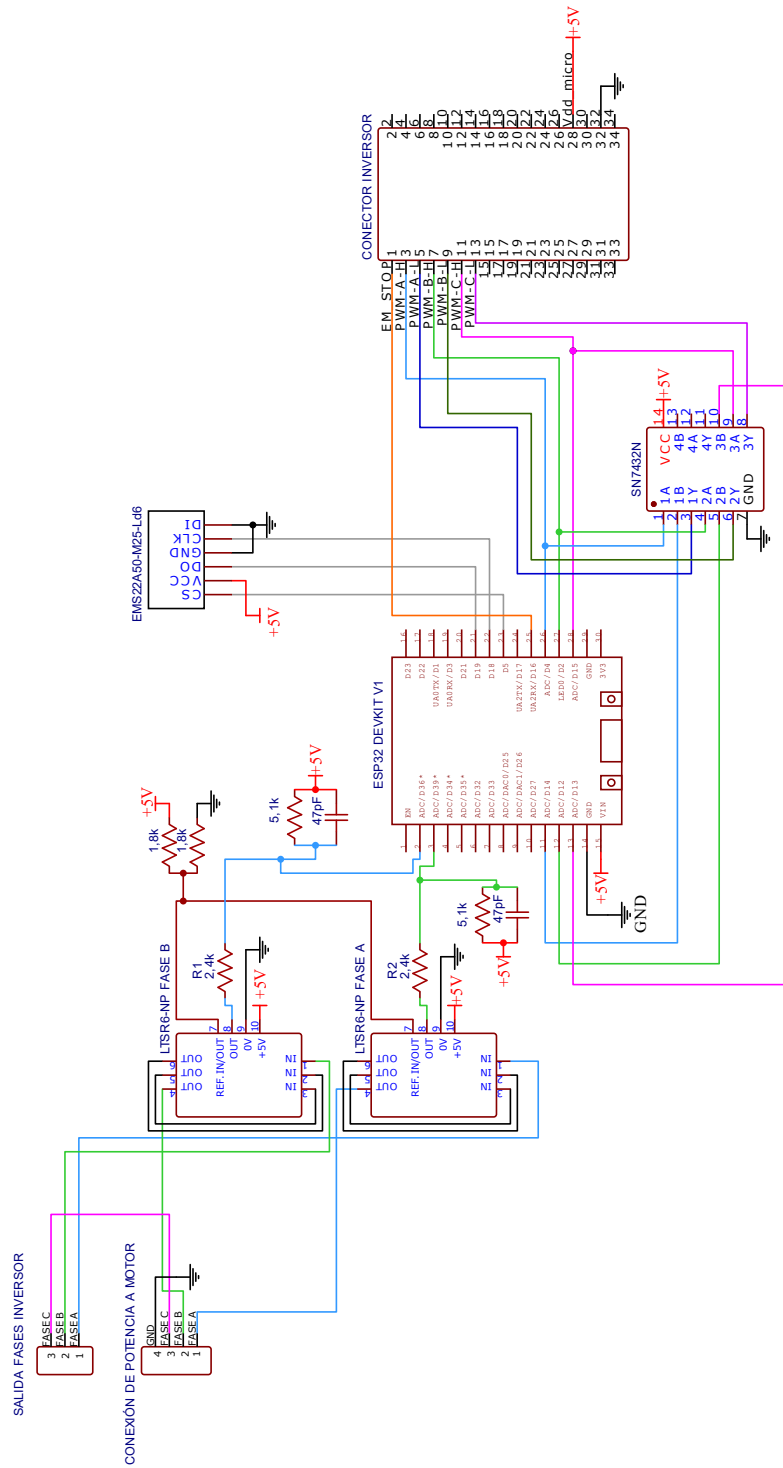


Figura 3.14 Esquema eléctrico de la placa del microcontrolador (elaboración propia)

CAPÍTULO 4: MEDIDAS Y MONTAJE DEL SENSOR DE POSICIÓN

4.1 INTRODUCCIÓN

Para poder realizar un control adecuado del motor, es necesario un dispositivo capaz de medir de forma fiable la posición angular del rotor de éste. Con esta información, es posible establecer el sector en que se encuentra en cada momento el rotor. La información del sector, como se ha podido comprobar en el segundo capítulo, es fundamental para la correcta generación de las señales de control y PWM enviadas al inversor y para la realimentación de corriente. Además, gracias a este elemento, también es posible obtener la velocidad angular del motor y con este dato realimentar el lazo de control de velocidad.

4.2 ESTRUCTURA

En el primer apartado, SENSOR DE POSICIÓN, se presentarán los tipos principales de sensores de posición disponibles en el mercado para finalmente elegir el más adecuado. Posteriormente se desarrollará el montaje mecánico del sensor al eje del motor, apartado MONTAJE MECÁNICO DEL ENCODER. Después, en el apartado LECTURA DE POSICIÓN EN EL MICROCONTROLADOR se describirá la forma en la que se podrá obtener la lectura de posición mediante el uso de este sensor. Una vez es posible obtener la medida de la posición angular del motor, en el apartado OBTENCIÓN DEL SECTOR, se desarrollará el método para obtener a partir de la lectura de posición el sector en que se encuentra el motor. Por último, se mostrará cómo, a través del sensor de posición, se puede obtener la velocidad de giro del motor en el apartado MEDIDA DE VELOCIDAD EN EL MICROCONTROLADOR.

4.3 SENSOR DE POSICIÓN

Como se ha comentado, el sensor debe permitir una medida fiable de la posición y la velocidad. Es por ello por lo que se ha de escoger aquel sensor que garantice una buena fiabilidad y precisión. A pesar de que el motor a controlar presenta su propio encoder óptico para la medida de posición, se ha optado por el empleo de un sensor externo. Esta decisión tiene su base en el hecho de que el encoder óptico del propio motor no es fácilmente accesible, dado que no conocemos sus prestaciones ni características técnicas al utilizar tecnología propietaria del fabricante y, además, debido a que utiliza un protocolo de transmisión serie que dificulta mucho su medida directa. Por tanto, aun siendo capaces de obtener medidas de este sensor a través del servoamplificador, se empleará un sensor externo de características conocidas.

Para la medida de posiciones y velocidades angulares, en el mercado nos encontramos dos grandes grupos de sensores: Los encoders ópticos y los encoders magnéticos. A continuación, se detallan las diferencias más notables entre estos dos grupos.

La tecnología de los encoders ópticos se basa en un fotoemisor, un receptor óptico y un disco opaco con ventanas que permiten el paso de la luz emitida por el fotoemisor hasta llegar al receptor, tal como podemos ver de forma simplificada en la Figura 4.1. Según la señal que reciba en cada momento el receptor, el sensor es capaz de establecer la posición y la velocidad angular del motor.

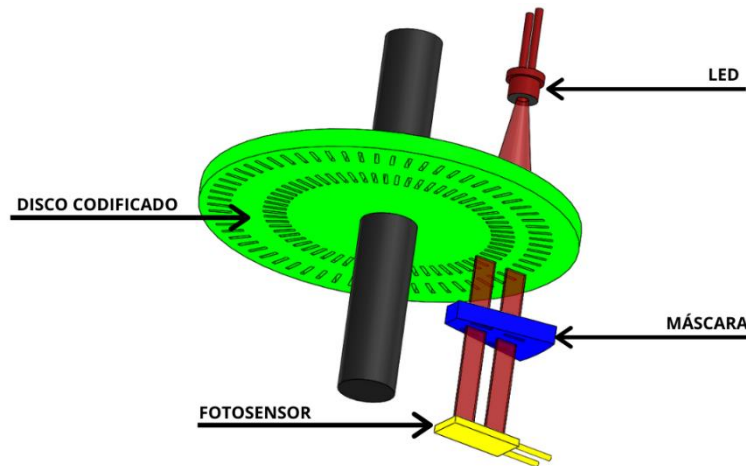


Figura 4.1 Esquema de un encoder óptico [10]

Cabe destacar, que dentro de los encoders ópticos existen dos tipos, los absolutos y los incrementales. Mientras que los absolutos son capaces de medir la posición en todo momento, los incrementales son capaces de medir incrementos de la posición. De esta forma, un encoder incremental no será capaz de proporcionar la posición una vez encendido, puesto que lo que mide son pulsos y a través de ellos calcula el desplazamiento. En la Figura 4.2, se observa la diferencia entre los discos de un encoder absoluto y otro incremental. Se ve claramente como a través de las ranuras del disco asociado al encoder absoluto, el sensor es capaz de obtener una posición única mientras que con el disco del encoder incremental solo es posible medir los incrementos de la posición.

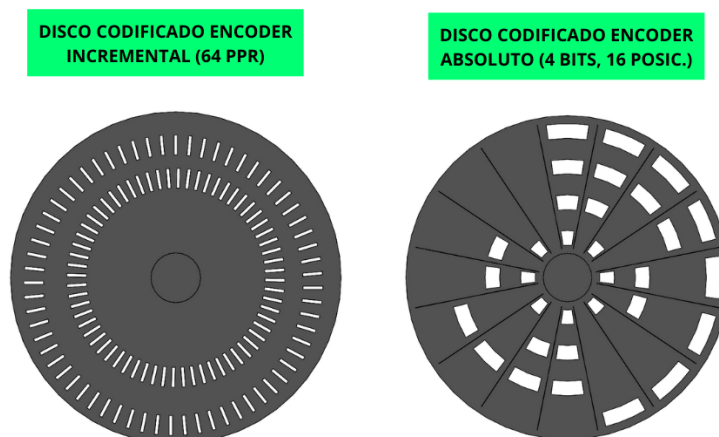


Figura 4.2 Comparativa entre discos de encoders ópticos [10]

El mecanismo de los encoders magnéticos es similar al de los ópticos. Estos sensores están formados por un disco unido al eje motor con uno o varios imanes permanentes en su periferia y un sensor, que detecta el campo magnético producido por el movimiento del disco. Con las medidas que el sensor toma del campo se obtiene la medida de la posición. En la Figura 4.3 se aprecia un esquema del funcionamiento de este tipo de sensores.

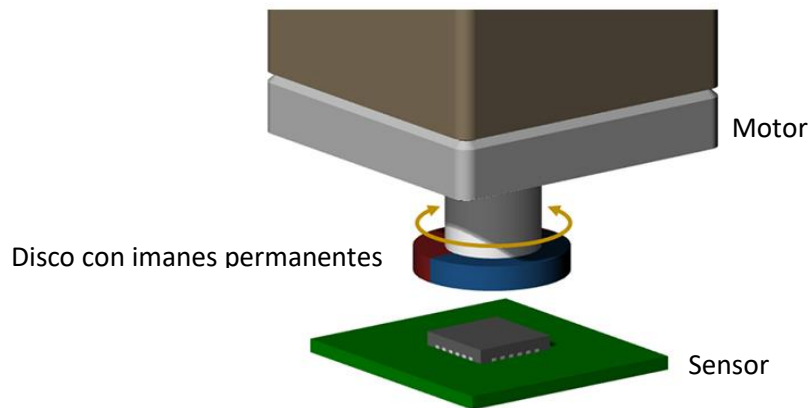


Figura 4.3 Esquema de un encoder magnético (modificado de [11])

A pesar de que ambos tipos de sensores proporcionan buenas medidas de posición, cada uno de ellos está enfocado a un ámbito de uso diferente. Los encoders ópticos son, en general, sensores más precisos y con una mayor resolución mientras que los encoders magnéticos a pesar de ser menos precisos son los más indicados en entornos en los que prima la fiabilidad puesto que son más robustos y sus principales ventajas es ser más pequeños y tener una gran fiabilidad incluso en entornos con suciedad o vibraciones. Un inconveniente de este tipo de encoder es su incapacidad de trabajar en entornos con presencia de campos magnéticos.

Dado que los encoders magnéticos presentan grandes ventajas en cuanto a fiabilidad, son más pequeños y, además, notoriamente más baratos, se elegirá este tipo de sensor para la medida de posición del motor. En concreto el sensor elegido es el encoder EMS22A50-M25-Ld6, el cual puede verse en la Figura 4.4. Las características básicas del sensor se muestran en la Tabla 4.1.



Figura 4.4 Encoder EMS22A50-M25-Ld6 [12]

| Características encoder EMS22A50-M25-Ld6 | |
|--|-----------------|
| Resolución | 1024 posiciones |
| Modo de transmisión | Serie (SPI) |
| Voltaje de suministro | 5V |
| Velocidad nominal | 1000 rpm |
| Nº de vueltas admisible | 100.000.000 |
| Protección IP | IP65 |

Tabla 4.1 Características encoder EMS22A50-M25-Ld6 [13]

Con los datos proporcionados es posible realizar el cálculo aproximado de las horas de vida sin fallo del sensor funcionando a su velocidad nominal.

$$100.000.000rev * \frac{1 \text{ min}}{1000 \text{ rev}} * \frac{1h}{60 \text{ min}} = 1.600,66 \text{ horas}$$

4.4 MONTAJE MECÁNICO DEL ENCODER

Una vez elegido el sensor con el cual se obtendrán las medidas de posición, es necesario unirlo al eje del motor para su correcto funcionamiento. Para ello, primero se ha de quitar la tapa trasera del motor donde está situado el encoder propio del motor dejando el eje al descubierto y posteriormente establecer una unión mecánica sólida entre el sensor y el eje del motor de modo que los ejes permanezcan coaxiales en todo momento. Para llevar a cabo este cometido son necesarias dos piezas. La primera es una que sea capaz de acoplar los ejes de sensor y motor, y la segunda, una que permita mantener la base del sensor de posición fija a la base del motor para que el movimiento de los ejes no altere la posición del sensor ni produzca vibraciones ni deslizamientos.

Dada la naturaleza de las piezas, piezas pequeñas y con un diseño concreto que se adecue a las medidas del motor y el sensor, la mejor opción para obtenerlas es mediante técnicas de impresión 3D (fabricación aditiva), debido a que de esta forma se pueden obtener piezas con un diseño a medida de forma más económica y rápida que mediante técnicas tradicionales de fabricación.

El proceso a seguir para imprimir las piezas es el siguiente: Primero se debe diseñar la pieza con el software adecuado, en este caso, Onshape. Una vez la pieza está diseñada, se ha de transferir al software encargado de generar las ordenes en formato G-code, en este caso, el programa encargado de realizar esta función es PrusaSlicer. El formato G-code es un formato de archivo que contiene las ordenes que debe ejecutar una máquina de control numérico. Es el formato empleado en corte por láser, fresadoras CNC o impresión 3D. Tras la obtención del G-code se procede a la impresión de la pieza. La técnica empleada es extrusión de material con plástico PLA. Esta técnica se basa en la

deposición de capas de material caliente sobre una superficie que, al solidificar se unen entre si formando la pieza deseada. La Figura 4.5 muestra los modelos de las piezas acoplador y soporte, realizados en el programa Onshape². Además, se muestra en la Figura 4.6 un corte de la pieza acoplador para ilustrar el acoplamiento entre los ejes de sensor y motor. La Figura 4.7 muestra el montaje final realizado en el laboratorio.

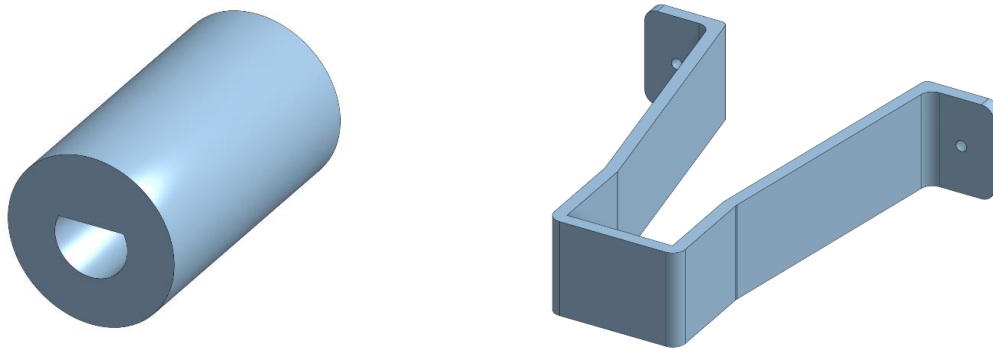


Figura 4.5 modelos de acoplador (izquierda) y soporte (derecha) en Onshape

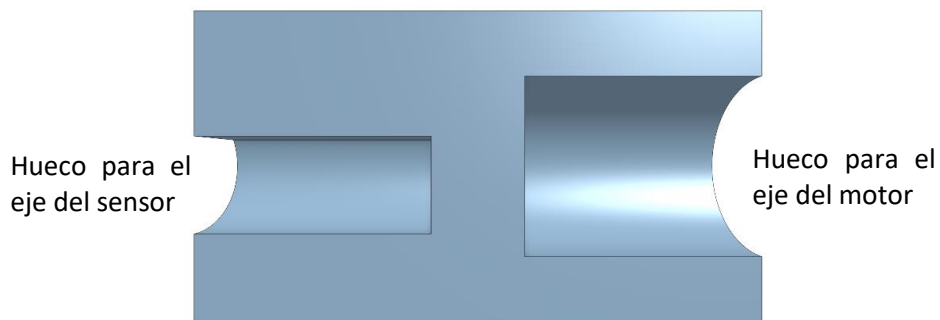


Figura 4.6 Corte de la pieza acoplador

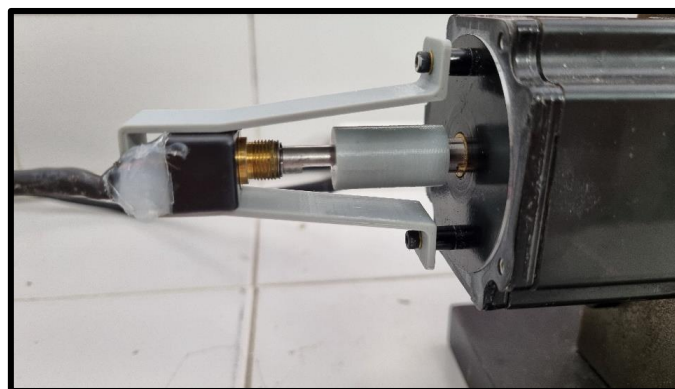


Figura 4.7 Montaje del sensor de posición

² En el documento planos de este trabajo se presentan los planos acotados de ambas piezas

4.5 LECTURA DE POSICIÓN EN EL MICROCONTROLADOR

Una vez realizado el montaje mecánico, se ha de conectar el sensor al microcontrolador para poder recoger los datos de posición del motor. El sensor de posición presenta tres pines digitales que serán los que se emplearán para la medida de la posición mediante un protocolo tipo serie (Serial Parallel Interface, SPI), las señales referentes a estos pines son: Clock (CLK), Digital output (DO) y Chip select (CS). Estas señales se conectan a diferentes pines del microcontrolador. El formato de dato que maneja el sensor de posición es un dato de diez bits, El cual se entrega de manera secuencial bit a bit por cada flanco de subida de la señal CLK. A parte de estos diez bits, el dispositivo entrega además cinco bits extra que codifican el estado y un bit de paridad como método de detección de errores. En la Figura 4.8 se puede ver un fragmento de la ficha técnica del sensor de posición donde se muestra el método utilizado para enviar los datos.

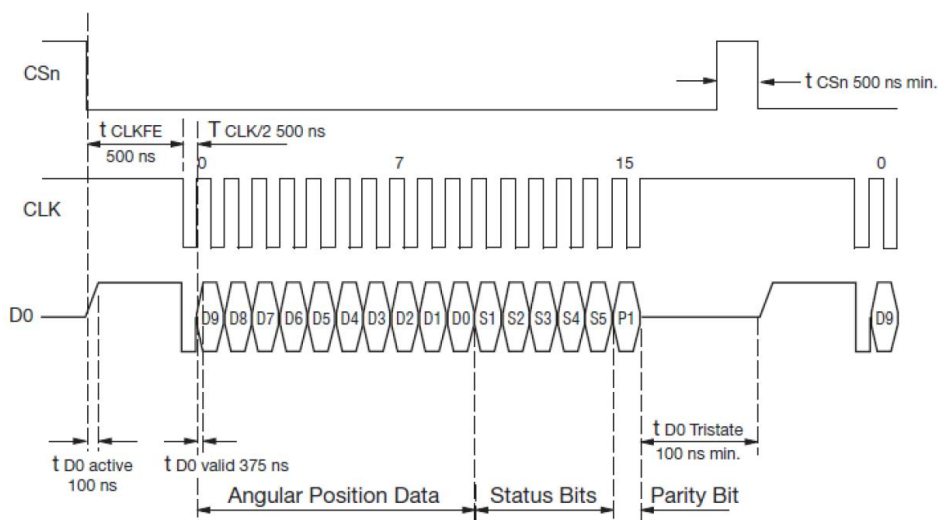


Figura 4.8 Esquema del envío de datos del sensor de posición [13]

Conociendo el método del sensor de posición para enviar datos, se ha de realizar un código para que el microcontrolador sea capaz de leer el valor enviado por el sensor. El código deberá mandar señales de reloj y por cada una registrar el valor del bit enviado por el sensor en el canal digital output. Una vez registrados todos los bits, el resultado será un valor entero entre 0 y 1023. En el Código 4.1 se puede ver el fragmento de código asociado a la lectura de posición.

Como se puede observar, primero se ponen a nivel bajo la señal chip select y la señal de reloj durante 44 ciclos de reloj del microcontrolador. Posteriormente, hay un bucle de diez iteraciones, en el cual, la primera instrucción a realizar es activar la señal de reloj durante 44 ciclos de reloj. A continuación, se guarda en el bit correspondiente a la iteración en que se encuentre el bucle, teniendo en cuenta que se empieza desde el bit 9 y se acaba en el bit 0, el valor correspondiente a la lectura del puerto asociado a digital output. Y, por último, se desactiva la señal de reloj durante 22 ciclos de reloj. De esta forma, en la variable "pos" se obtendrá el valor de la posición codificado en un formato de 10 bits. Tras finalizar el bucle, las señales de reloj y chip select se dejan a nivel alto para informar al sensor de posición de que puede pasar a estado de reposo y la función devuelve el valor de la posición.

```

uint16_t medirPos(){
    uint16_t pos=0;
    digitalWrite(ENC_CS, LOW);
    digitalWrite(ENC_CLK, LOW); delayshort(44);
    for (int i=9; i>=0; i--) {
        digitalWrite(ENC_CLK, HIGH); delayshort(44);
        pos=pos|(digitalRead(ENC_DATA)<<i);
        digitalWrite(ENC_CLK, LOW); delayshort(22);
    }
    digitalWrite(ENC_CLK, HIGH);
    digitalWrite(ENC_CS, HIGH);
    return pos;
}

```

Código 4.1 Función medida de la posición

4.6 OBTENCIÓN DEL SECTOR

Mediante el empleo del sensor de posición se debe de extraer además de la propia posición y velocidad la información relativa al sector en el que se encuentra el motor para el control del mismo, así como para la correcta realimentación de corriente. Para ello primero se deben realizar una serie de ensayos de los cuales se pueda obtener el perfil de las fuerzas electromotrices inducidas de las tres fases del motor con sus correspondientes desfases con respecto a las medidas de posición del sensor. Con esta información se procederá a definir los rangos de posición que constituyen cada sector. Este apartado se divide en, por una parte, los ensayos previos con los que obtener la relación entre posición y sector, y, por otra parte, en la implementación en el microcontrolador de la obtención del sector.

4.6.1 Ensayos previos

El ensayo consiste en registrar mediante el uso de un osciloscopio digital la fuerza electromotriz inducida en cada una de las fases, en circuito abierto, al girar el motor a velocidad constante. De manera simultánea, se registra, también con el osciloscopio, una señal lógica definida en el microcontrolador, la cual toma valor 0 en la primera mitad de valores de la posición, es decir, mientras la lectura de posición se sitúe en el rango [0 – 511], y toma valor 1, en la segunda mitad, lo que constituye el rango [512 – 1023]. Dado que las lecturas con el osciloscopio relativas a las tensiones en las fases presentan gran cantidad de ruido, estas señales se filtran empleando el software Matlab³. Puesto que las señales son senoidales⁴, el filtrado consiste en realizar la transformada de Fourier a la señal y representar únicamente el primer armónico. El ensayo se realiza para 500 y 1000 rpm. Los ángulos que se comentan en este apartado son ángulos referidos al primer periodo de las tres ff.ee.mm. de fase medido desde el comienzo de una vuelta (de los tres que tienen, al ser una máquina hexapolar) y, por tanto, son ángulos de onda o eléctricos. En el siguiente apartado, para la obtención del sector de funcionamiento a partir de la medida del sensor de posición (ángulo mecánico) habrá que realizar la conversión de ángulo mecánico a ángulo eléctrico o de onda de la f.e.m.-corriente.

³ Las funciones empleadas pueden consultarse en los anexos

⁴ En este motor en concreto

- *Ensayo a 500 rpm*

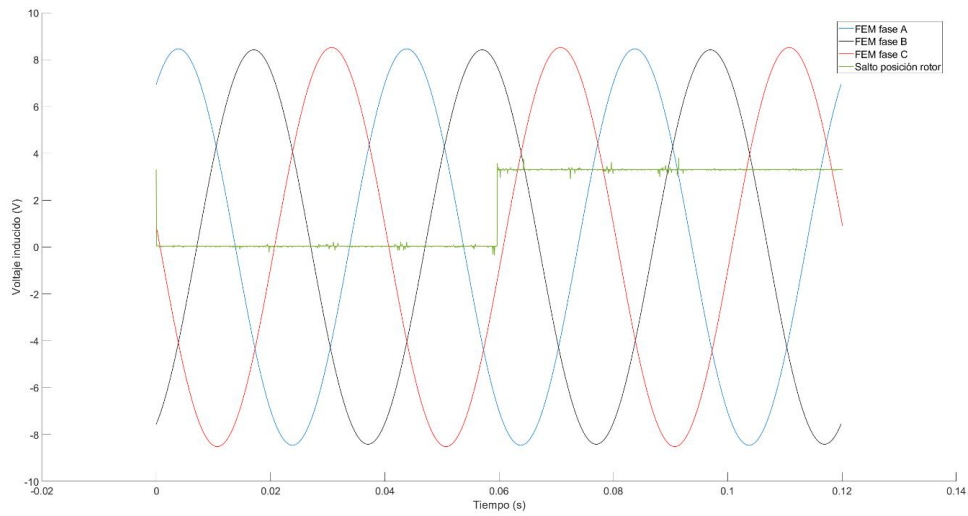


Figura 4.9 f.e.m. de las fases con el motor girando a 500 rpm

| Fase | Fase A | Fase B | Fase C |
|-------------|---------|----------|--------|
| Desfase (°) | -35,48° | -155,22° | 85,20° |

Tabla 4.2 Desfases obtenidos para el ensayo a 500 rpm

- *Ensayo a 1000 rpm*

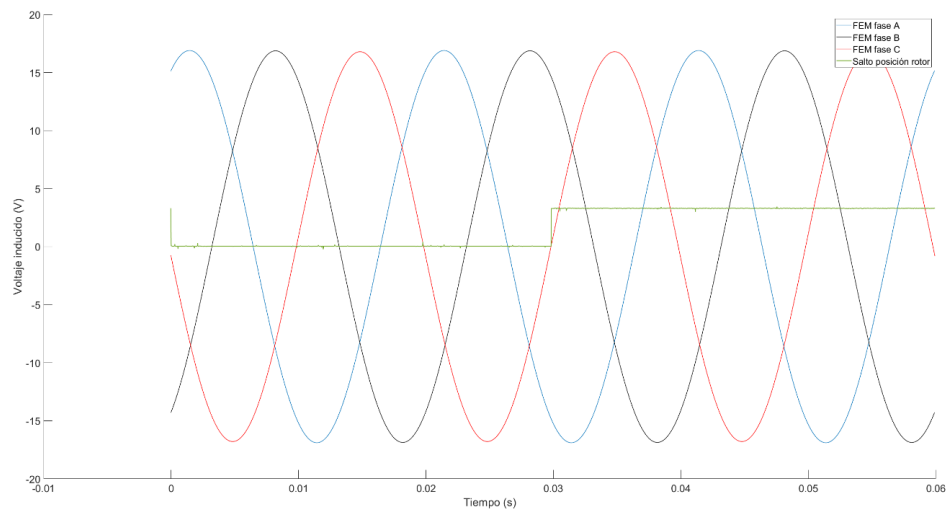


Figura 4.10 f.e.m. de las fases con el motor girando a 1000 rpm

| Fase | Fase A | Fase B | Fase C |
|-------------|---------|----------|--------|
| Desfase (°) | -27,90° | -147,00° | 92,56° |

Tabla 4.3 Desfases obtenidos para el ensayo a 1000 rpm

De los resultados mostrados en las figuras y tablas anteriores, los desfases mostrados hacen referencia al desfase en grados de cada una de las ondas senoidales referentes a cada fase con respecto al salto de posición medido, salto que se produce cuando el sensor de posición detecta una vuelta completa del motor.

Es necesario comentar en este apartado que, a pesar de que el motor no presenta un perfil de fuerzas electromotrices trapezoidal esto no implica ningún problema a la hora de obtener los sectores. Simplemente se debe tomar el pico más alto de cada fase como punto medio de la conducción positiva y el punto más bajo como punto medio de la conducción negativa.

Con todos estos datos, se puede proceder a asignar los sectores a los ángulos correspondientes y posteriormente convertir los datos de ángulos a posiciones registradas en datos de 10 bits. Estos pasos se pueden observar en la Tabla 4.4.

| Conducción | Fase A | | Fase B | | Fase C | |
|---------------------------|----------|----------|----------|----------|----------|----------|
| | Positiva | Negativa | Positiva | Negativa | Positiva | Negativa |
| Límite inferior (°) | -28,69 | 151,31 | 91,31 | 271,31 | -148,69 | 31,31 |
| Límite superior (°) | 91,31 | 271,31 | 211,31 | 391,31 | -28,69 | 151,31 |
| Límite inferior (10 Bits) | 942 | 430 | 260 | 772 | 601 | 89 |
| Límite superior (10 Bits) | 260 | 772 | 601 | 89 | 942 | 430 |

Tabla 4.4 Límites de tramos de conducción en ángulos y datos de 10 bits

Por último, solo queda asociar los tramos de conducción positiva, negativa y transición a los sectores adecuados. El resultado de este proceso se representa en la Tabla 4.5, donde se puede ver el inicio y fin de cada sector representado con el valor digital de 10 bits que devuelve el sensor de posición.

| Sector inicio/fin | Angulo (°) | Valor digital 10 bits |
|-------------------|------------|-----------------------|
| 1 inicio | 148,69 | 423 |
| 1 fin | 208,69 | 593 |
| 2 inicio | 208,69 | 593 |
| 2 fin | 268,69 | 764 |
| 3 inicio | 268,69 | 764 |
| 3 fin | 328,69 | 934 |
| 4 inicio | 328,69 | 934 |
| 4 fin | 28,69 | 82 |
| 5 inicio | 28,69 | 82 |
| 5 fin | 88,69 | 252 |
| 6 inicio | 88,69 | 252 |
| 6 fin | 148,69 | 423 |

Tabla 4.5 Límites de los sectores en ángulos y datos de 10 bits

4.6.2 implementación en el microcontrolador

Antes de implementar la obtención del sector en el microcontrolador se ha de tener en cuenta que los límites de cada sector obtenidos con los ensayos previos son límites eléctricos, es decir, no se corresponden con los ángulos y posiciones reales (mecánicos, según los proporciona el sensor de posición) en los cuales se encuentran los sectores en el motor. Esto se debe a que para obtener los datos de conducción positiva y negativa se ha tomado el punto medio y sumado y restado 30 grados a cada lado, operaciones realizadas sobre los ángulos de onda o eléctricos, para de esta forma obtener solo seis sectores. Para tener esto en cuenta solo se ha de transformar la posición, multiplicándola por 3 y realizando el resto con 1023 para asegurar que no sobrepasa el límite. De esta manera se consigue que el control tenga en cuenta los tres pares de polos del motor haciendo que haya en realidad 18 sectores. El código relativo a la función para obtener el sector se muestra a continuación.

```
void CalcSector(uint16_t posicion) {  
  
    posicionSect = (posicion*3) & 1023;  
  
    if(posicionSect<89) sector=6;  
    else if(posicionSect<260) sector=1;  
    else if(posicionSect<430) sector=2;  
    else if(posicionSect<601) sector=3;  
    else if(posicionSect<772) sector=4;  
    else if(posicionSect<942) sector=5;  
    else sector=6;  
  
}
```

Código 4.2 Función cálculo del sector

4.7 MEDIDA DE VELOCIDAD EN EL MICROCONTROLADOR

Una vez desarrollada la función que permite la medida de la posición en un instante concreto, se debe implementar otra función que permita realimentar el lazo de control con la velocidad del motor. Para ello, existen diversos métodos. El más simple consiste en medir la posición continuamente con un cierto tiempo de muestreo, conocidos dos valores de posición y el tiempo de muestreo, obtener la velocidad se reduce a realizar una operación matemática: dividir ambos valores y aplicar un factor de escala para pasar de pulsos por segundo a revoluciones por minuto. Este método sin embargo presenta ciertas limitaciones. Por ejemplo, si el tiempo de muestreo es demasiado pequeño, existirá una velocidad mínima, la cual será tomada como cero puesto que el sensor medirá la misma posición en dos instantes consecutivos. Por el contrario, si el tiempo de muestreo es demasiado grande, existirá una velocidad máxima en la cual el ángulo recorrido en ese tiempo será mayor que una revolución completa y no podrá ser interpretada de manera correcta.

El método empleado para evitar estos inconvenientes consiste en medir la posición en cada iteración del bucle de control, calcular una velocidad estimada y realizar una aproximación de la posición que deberá devolver el sensor en la siguiente iteración. Con la diferencia entre la posición estimada y la real, se obtendrá un error. A partir de aplicar sobre este error una dinámica proporcional e integral

con los parámetros adecuados, se obtendrá la velocidad estimada con la cual se procederá a estimar la siguiente posición. Podríamos decir que se trata de una versión básica de un observador de velocidad empleando para esa observación la diferencia entre la posición real y la estimada a partir de la velocidad observada.

El Código 4.3 muestra la función relativa a la medida de velocidad, a continuación, se explica línea a línea.

```
float CalcV(){  
  
    float Velocidad;  
  
    pos[1] = posicion;  
  
    posEst = posEst+Vest*dt;  
  
    if(posEst>=1024.0) posEst -= 1024.0;  
    if(posEst<0.0) posEst = 1024.0-posEst;  
  
    error = (pos[1]-posEst);  
  
    if(error<-500) error += 1024;  
    if(error>500) error -= 1024;  
  
    V_Ui_realim = V_Ui_realim+((V_K*dt)/V_Ti_realim)*error;  
  
    Vest = V_K*error + V_Ui_realim;  
  
    pos[0] = pos[1];  
  
    Velocidad = Vest*(1/1024.00)*60.00;  
  
    return Velocidad;  
}
```

Código 4.3 Función medida de velocidad

Se comienza definiendo una variable tipo float correspondiente a la velocidad. El vector pos[], almacena en la posición 0, el valor correspondiente a la posición de la iteración anterior y en la posición 1, el valor correspondiente a la posición actual. Una vez recogido el valor de la posición actual, se calcula la estimación de dicha posición a través de la velocidad estimada y se almacena en la variable posEst. Además, se fuerza a la posición estimada a que se mantenga en el rango [0-1023]. Con los valores real y estimado de la posición actual se calcula el error y se limita, teniendo en cuenta los casos en los que las posiciones se encuentren en el límite inferior o superior del rango. Con el error calculado, se procede a obtener la acción integral y proporcional mediante los parámetros V_K, ganancia proporcional y V_Ti_realim, Tiempo integral. Por último, se calcula la velocidad estimada (Vest) y se realiza la conversión de pulsos/segundo a rpm.

Para hallar los parámetros adecuados se realizan una serie de ensayos consistentes en girar el motor a una velocidad conocida mediante un generador acoplado al mismo y observar la respuesta del sistema para diferentes parámetros. La lectura se realiza a través del envío de datos por Bluetooth. Tras la realización de estos ensayos se determina que el mejor ajuste es el obtenido con una ganancia proporcional igual a 250 y sin acción integral. A continuación, se muestran algunos de estos ensayos. Para todos ellos, se gira el motor a 500 rpm en ambos sentidos alternativamente.

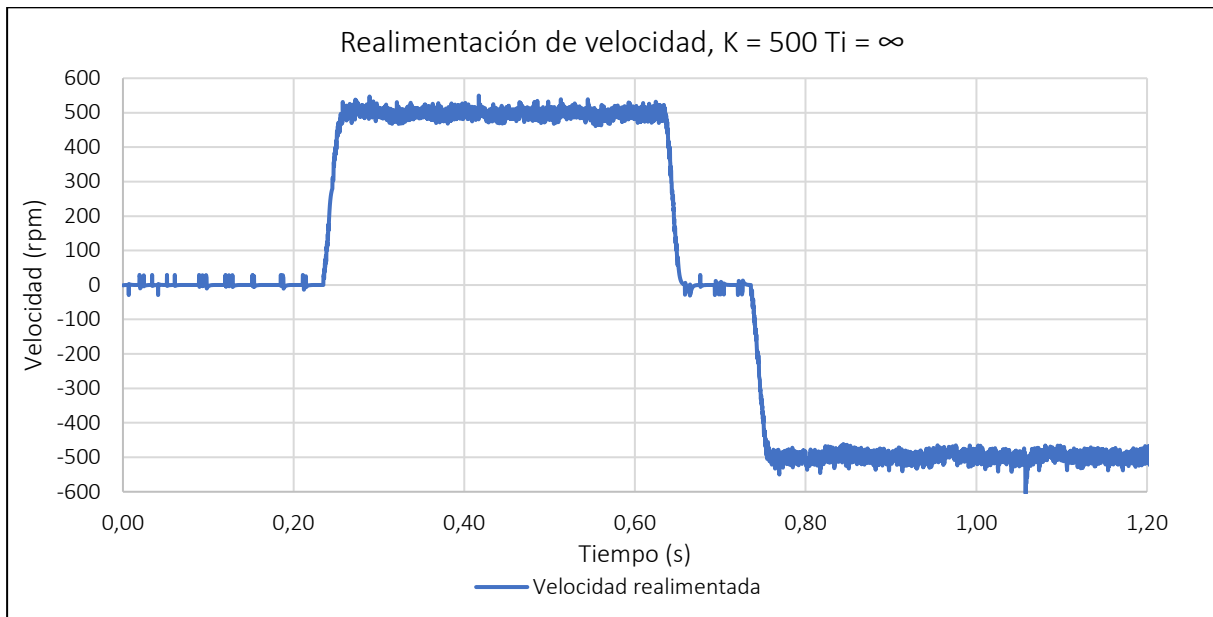


Figura 4.11 Realimentación de velocidad, $K = 500$, $T_i = \infty$

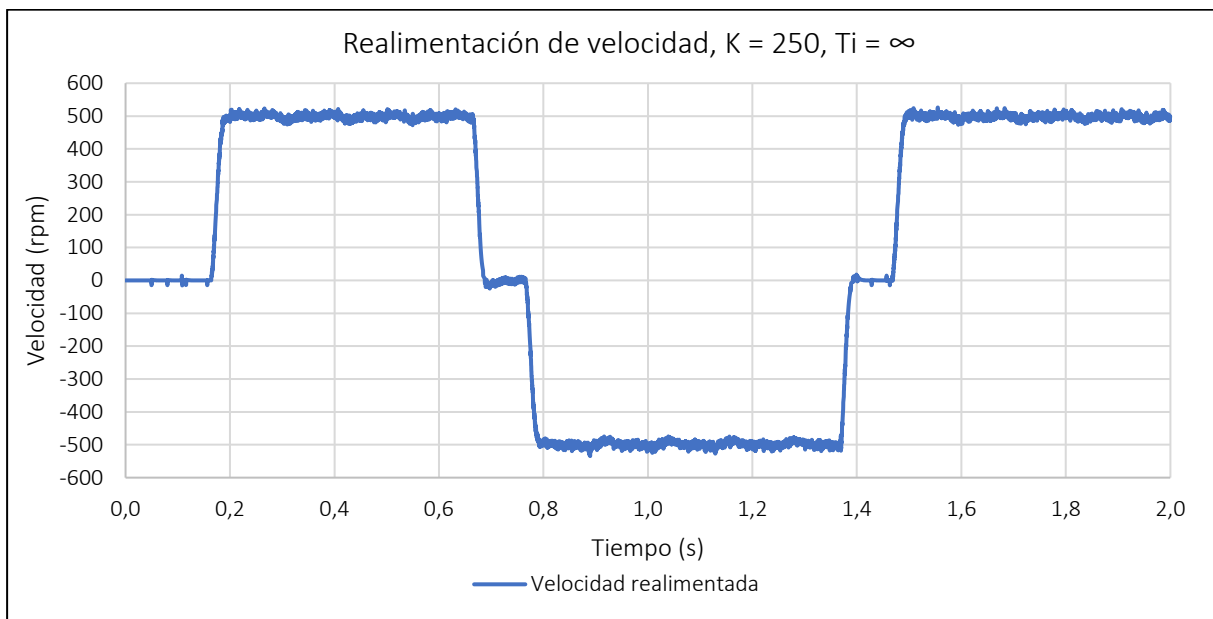


Figura 4.12 Realimentación de velocidad, $K = 250$, $T_i = \infty$

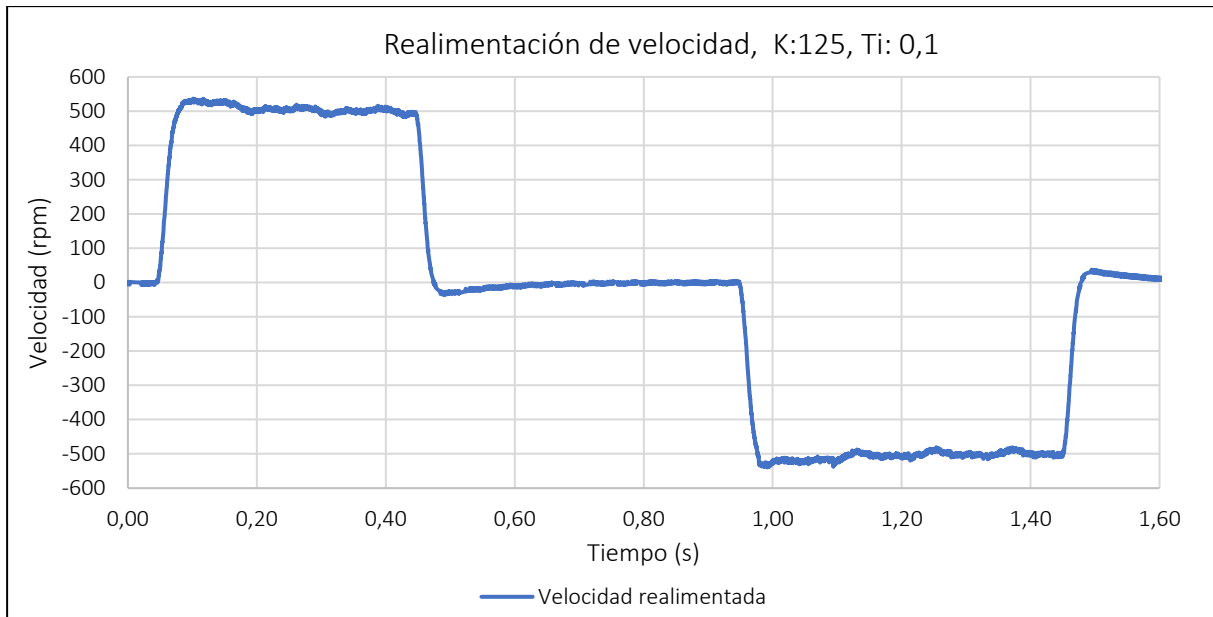


Figura 4.13 Realimentación de velocidad, K = 125, Ti = 0,1

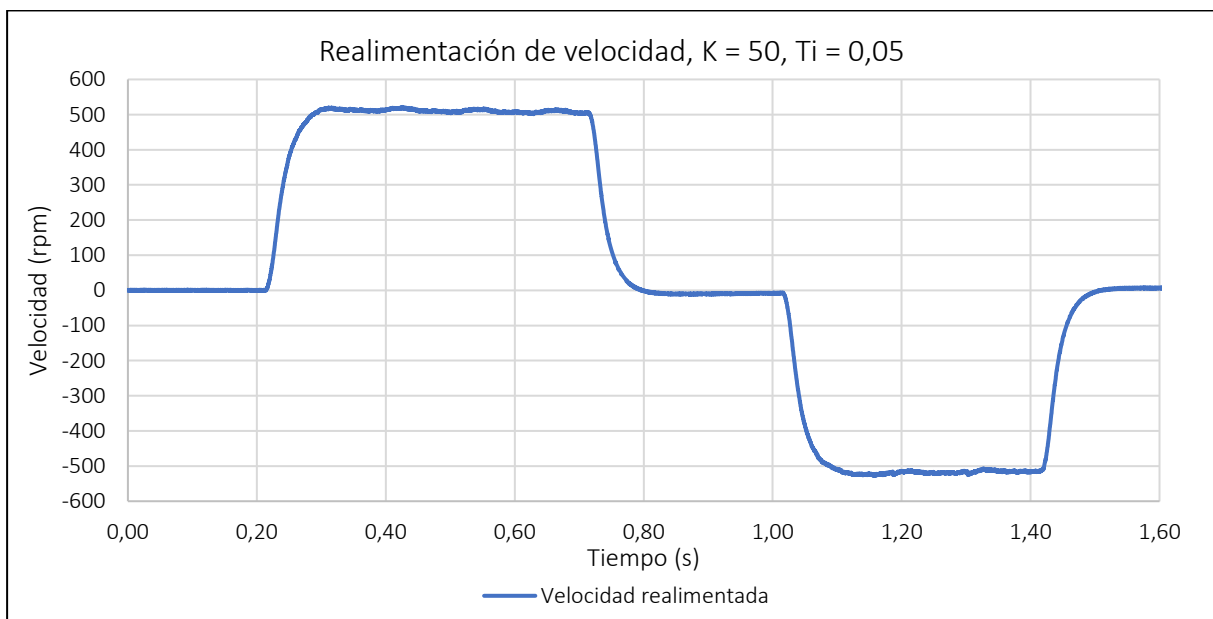


Figura 4.14 Realimentación de velocidad, K = 50, Ti = 0,05

Se observa en las figuras anteriores una tendencia clara. Al disminuir la ganancia proporcional se reduce el ruido, pero también se hace necesario un menor tiempo integral, que suele propiciar mayor sobreoscilación, además de incrementar el tiempo de establecimiento. Dado que lo que se busca es el menor tiempo de establecimiento para evitar interferir con el regulador PI, se escoge el ajuste referente a la Figura 4.12, que, a pesar de presentar rizado, es muy ligero y es un ajuste con buena dinámica y un tiempo de establecimiento muy bajo.

CAPÍTULO 5: FUNCIONES DEL MICROCONTROLADOR

5.1 INTRODUCCIÓN

Tras el montaje de la placa del microcontrolador y haber obtenido la información necesaria de todos los componentes en los capítulos anteriores, en este capítulo se desarrolla la parte correspondiente a las tareas que recaen sobre el microcontrolador a la hora de realizar el control del motor. Las funcionalidades a realizar por el microcontrolador son: Implementar el lazo de control en cascada con los correspondientes reguladores PI y mecanismos de saturación y anti-windup, realizar la realimentación de corriente y velocidad, y enviar al inversor las señales adecuadas de PWM y habilitación. EL microcontrolador, además, deberá establecer una comunicación Bluetooth con un dispositivo móvil, el cual realizará la función de interfaz de usuario, sin embargo, esta funcionalidad se tratará en un capítulo independiente. Esto se debe a que el ámbito de este capítulo se centra exclusivamente en las funciones necesarias para el control del motor.

5.2 ESTRUCTURA

Este capítulo se divide en las diferentes funciones a realizar por el microcontrolador. El primer apartado, REALIMENTACIÓN DE CORRIENTE Y VELOCIDAD mostrará el modo en que el microcontrolador realiza la realimentación de corriente y velocidad al lazo de control. Después, se mostrará en el apartado GENERACIÓN DE SEÑALES PWM Y HABILITACIÓN la forma en la que el microcontrolador envía las señales de control al inversor. El último apartado, BUCLE DE CONTROL, describe la forma en la que se ha desarrollado el lazo de control con los reguladores PI así como el modo de ajuste de cada uno hasta llegar al control en cascada.

5.3 REALIMENTACIÓN DE CORRIENTE Y VELOCIDAD

Dado que en el capítulo 4 ya se definió la función mediante la cual se realiza la lectura de velocidad, este apartado se encuentra más enfocado en la realimentación de corriente. Como se vio en el segundo capítulo, para la realimentación de corriente es esencial disponer de la información del sector en el que se encuentra el motor. Con esta información, el sistema es capaz de conocer el estado de conducción de cada fase, según la Tabla 2.1. La corriente realimentada se obtiene como el resultado de multiplicar la corriente de cada fase por su estado de conducción teórico según el sector, sumar los tres productos y, por último, dividir el resultado entre 2, puesto que siempre hay dos fases en estado de conducción. Con este cálculo es posible registrar la corriente realimentada, tanto si es positiva como si es negativa, en cuyo caso las corrientes presentarán el estado de conducción inverso al teórico mostrado en la Tabla 2.1. Para finalizar el cálculo, solo queda resolver como se toman las corrientes de las fases que se encuentren en un sector en el que les corresponda conducción nula. Se podrían multiplicar por cero y no tener en cuenta, sin embargo, dado que los cambios de conducción no son ideales ni instantáneos, siempre hay un tiempo en el que por dicha fase sigue circulando corriente. Por estos motivos, la solución más adecuada es multiplicar la corriente de estas fases por el mismo

coeficiente por el que se multiplicó en el sector anterior. Como este coeficiente depende del sentido de giro del motor, se debe definir una variable que almacene el sentido de giro registrado por el sensor de posición en su cálculo de la velocidad de giro del motor.

Todo lo explicado hasta ahora con respecto a la realimentación de corriente se puede visualizar en el extracto de código de la función “CalcI”, utilizado para la realimentación de corriente.

```
float CalcI(uint16_t Ia,uint16_t Ib,int sector){

    ia = 0.00288*Ia-5.4506;
    ib = 0.00288*Ib-5.4327;
    ic = -ia-ib;

    switch(sector){

        case 1: //A:+, B:0, C:-
            return (ia-ib*(sentido)-ic)/2;
            break;

        case 2: //A:0, B:+, C:-
            return (+ia*(sentido)+ib-ic)/2;
            break;

        case 3: //A:-, B:+, C:0
            return (ib-ia-ic*(sentido))/2;
            break;

        case 4: //A:-, B:0, C:+
            return (+ib*(sentido)-ia+ic)/2;
            break;

        case 5: //A:0, B:-, C:+
            return (-ib-ia*(sentido)+ic)/2;
            break;

        case 6: //A:+, B:-, C:0
            return (ia-ib+ic*(sentido))/2;
            break;

    }
}
```

Código 5.1 función realimentación de corriente

Se observa como los parámetros de la función son las lecturas de los dos sensores de efecto Hall y el sector. Con las funciones de transformación obtenidas en el capítulo 3 se convierte la lectura analógica de los sensores en valores de corriente en amperios y se halla la corriente restante como la complementaria de las dos anteriores. Con las corrientes ya en amperios solo queda evaluar el sector en el que se encuentra el motor y aplicar las operaciones comentadas anteriormente. Nótese como se utiliza la variable global “sentido” para calcular el modo de conducción a tomar para las corrientes de la fase a la que le corresponde conducción nula

En cuanto a la realimentación de velocidad, simplemente se realiza una llamada a la función mostrada en el Código 4.3.

La realimentación de corriente y velocidad es lo primero que se lleva a cabo dentro del bucle en el cual se implementa el lazo de control.

5.4 GENERACIÓN DE SEÑALES PWM Y HABILITACIÓN

En el capítulo 3 se explicó el modo de comunicación con el inversor que lleva a cabo el microcontrolador. Esta comunicación se basa en señales de PWM con un duty cycle que viene dado por el lazo de control de corriente y señales de Habilitación que permiten mantener en bloqueo las ramas del inversor que correspondan con aquella fase que en cada sector deba mantenerse en conducción nula.

En este apartado se muestra la función que se ejecuta al final del lazo de control que permite enviar al inversor las señales antes comentadas. Un extracto de la función implementada en el Arduino IDE ⁵se puede ver en el Código 5.2.

```
void GeneraPWM(int sector, float dc){
switch(sector){

case 1:/* Fase A: +; Fase B: 0; Fase C: - */
digitalWrite(GPIO_HABA, LOW);
digitalWrite(GPIO_HABB, HIGH);
digitalWrite(GPIO_HABC, LOW);

dA=dc;
dB=0;
dC=1-dc;

dcAc =CalcD(dA);
dcBc =CalcD(dB);
dcCc =CalcD(dC);

ledcWrite(pwmA,dcAc);
ledcWrite(pwmB,dcBc);
ledcWrite(pwmC,dcCc);

break;

case 2:/* Fase A: 0; Fase B: +; Fase C: - */
digitalWrite(GPIO_HABA, HIGH);
digitalWrite(GPIO_HABB, LOW);
digitalWrite(GPIO_HABC, LOW);

dA=0;
dB=dc;
dC=1-dc;

dcAc =CalcD(dA);
dcBc =CalcD(dB);
dcCc =CalcD(dC);

ledcWrite(pwmA,dcAc);
ledcWrite(pwmB,dcBc);
ledcWrite(pwmC,dcCc);

break;
```

Código 5.2 Función para generar las señales del inversor

⁵ El código completo se encuentra en los anexos

Por economía del espacio se han representado solo los casos referentes a los dos primeros sectores dado que el resto son homólogos. Se observa en el código como se entregan como parámetros el duty cycle (dc) y el sector. Al evaluar el sector, primero se envían las señales de habilitación adecuadas según corresponda a las dos ramas del inversor que deben tener conducción de corriente, luego se asocian a las tres fases el duty cycle positivo y el negativo. Al estar la señal de habilitación correspondiente a la fase a la que le corresponda conducción nula en nivel alto, el valor de duty cycle asociado a la parte positiva de esa rama es nulo, para forzar su bloqueo. Posteriormente se ejecuta la función “CalcD”, la cual se desarrolla en el Código 5.3 y cuyo único cometido es transformar el duty cycle como valor decimal, en un valor entero dentro del rango de valores que establece la propia resolución asociada a los canales PWM, en este caso, la resolución es 12 bits. Y, por último, se envía en los pines asociados a tal propósito la señal PWM para las tres fases.

```

short CalcD(float dc){
/*Calcula el duty cycle como valor entero a partir del valor decimal [0,1] ----> [0,2^res-1]*/
    short DC;
    DC = dc*(pow(2, rpwm)-1);
    return DC;
}

```

Código 5.3 Función para convertir el duty cycle de decimal a entero

5.5 BUCLE DE CONTROL

Para desarrollar este apartado, primero se tratará de forma independiente el lazo de control de corriente y una vez ajustado correctamente se añadirá el lazo de velocidad para formar el sistema final en cascada. En ambos subapartados se mostrará el método para ajustar los reguladores PI de cada lazo en el sistema real, así como los resultados finales tras el ajuste.

El bucle de control a implementar es discreto, por ello, los reguladores a desarrollar deben ser también discretos. Mientras que, en un control continuo, la ecuación referente a la acción proporcional del controlador es:

$$U_p(t) = K_c \cdot e(t)$$

Donde U_p corresponde a la acción proporcional, K_c a la ganancia proporcional y e al error entre consigna y valor realimentado. En un control discreto, al no tener un registro continuo del error la ecuación se transforma en:

$$U_p(k) = K_c \cdot e(k)$$

El error ahora es el medido en una iteración (k) del bucle de control. La acción integral también cambia, mientras que, en el regulador continuo, la ecuación es:

$$U_i(t) = K_c \cdot \frac{1}{T_i} \cdot \int_0^t e(\tau) d\tau$$

Donde U_i corresponde a la acción integral y T_i al tiempo integral. Igual que antes, dado que el regulador no permite integrar de manera continua, la ecuación para el cálculo de la acción integral en un regulador discreto queda:

$$U_i(k) = U_i(k-1) + \frac{K_c \cdot T_s}{T_i} \cdot e(k)$$

Donde T_s representa el paso de tiempo de cada iteración. Se observa como en cada iteración sumamos al valor de acción integral de la iteración anterior la aproximación discreta a la integral del error. A pesar de no emplear en este caso la acción derivada, esta también presenta diferencias. Mientras la acción derivada de un control continuo es:

$$U_d(t) = K_c \cdot T_d \cdot \frac{de(t)}{dt}$$

Para obtener la acción derivada en un control discreto se sustituye la derivada por su aproximación:

$$U_d(k) = K_c \cdot T_d \cdot \frac{e(k) - e(k-1)}{T_s}$$

La acción de control resultante del conjunto PID en ambos casos será la suma de todas las acciones:

$$U_{pid} = U_p + U_i + U_d$$

Una vez realizada esta breve introducción sobre los controladores PID continuos y discretos es necesario mencionar que, en este caso, los controladores son todos PI, discretos y con un paso de tiempo de 200 microsegundos. En los siguientes apartados se mostrará la implementación y ajuste de los reguladores.

5.5.1 Lazo de corriente

El lazo de corriente es el primero en implementarse. De esta forma se podrá ajustar de manera independiente y comprobar que su funcionamiento es adecuado antes de seguir con el regulador de velocidad y el control en cascada. El Código 5.4 muestra el extracto del bucle de control asociado al lazo de corriente. Lo primero que realiza el código es obtener el error entre consigna y realimentación. Después, calcula la acción proporcional empleando la variable I_Kc , la cual está asociada a la ganancia proporcional del regulador de corriente. Tras el cálculo de la acción proporcional, se lleva a cabo el mecanismo anti-windup, el cual comprueba si el regulador se encuentra saturado, en cuyo caso no integra. Si no está saturado, se calcula la acción integral. Tras esto, se calcula la acción de control y se satura para que no aparezcan valores de duty cycle menores que 0 ni mayores que 1. Tras la saturación, se asocia a la variable que almacena el duty cycle la acción de control. Es necesario comentar que en realidad la saturación se realiza en el rango $[-0,5 - 0,5]$, y posteriormente se suma un offset de valor 0,5. Esta forma de operar se debe a que el regulador debe tener un rango simétrico para poder controlar partiendo del estado de reposo corrientes positivas y negativas. Dado que el estado de reposo del inversor es un duty cycle de 0,5 (el valor que aplicado a las dos fases que deben conducir produce una tensión media nula entre sus extremos), se hace necesario añadir este valor de reposo

(0,5) a la salida del regulador (simétrica y comprendida en el intervalo [-0,5 – 0,5]) para conseguir un valor centrado en el intervalo aceptable por las funciones de modulación PWM: [0 – 1].

```

//LAZO DE CORRIENTE

//Se obtiene el error
I_err=I_consigna-Irealim;

//Se calculan las acciones proporcional e integral
I_Up = I_Kc*I_err;

//Antiwindup
if((I_Upid-I_Upid_Sat)>0.1 or (I_Upid-I_Upid_Sat)<-0.1){

    // No se integra

}else{

    I_Ui = I_Ui+((I_Kc*Ts)/I_Ti)*I_err;

}

//Calculo de la accion de control
I_Upid = I_Up + I_Ui;

//Saturamos en el rango [DC_lim_sup; DC_lim_inf]

if(I_Upid>DC_lim_sup) I_Upid_Sat=DC_lim_sup;

else if(I_Upid<DC_lim_inf)I_Upid_Sat=DC_lim_inf;

else I_Upid_Sat=I_Upid;

//se asocia la accion de control despues de la saturación al duty cycle

dc=I_Upid_Sat+0.5;

```

Código 5.4 Implementación del regulador PI de corriente

Para la obtención de los parámetros I_{Kc} y I_{Ti} , la ganancia proporcional y el tiempo integral respectivamente, se realiza un ensayo consistente en medir la corriente realimentada ante un salto de duty cycle de 0,50 a 0,52. De esta manera se puede obtener una primera aproximación de los parámetros. El resultado de este ensayo es el mostrado en la Figura 5.1. Se ha de tener en cuenta que este proceso se asocia con la función de transferencia de un sistema de primer orden de la forma:

$$G(s) = \frac{K_p}{\tau s + 1}$$

Donde K_p es la ganancia del proceso y τ la constante de tiempo. Mientras que el control PI viene dado por la expresión:

$$PI(s) = \frac{K_c(T_i s + 1)}{T_i s}$$

Para la primera aproximación se pretende realizar un control por cancelación de polos, es decir, igualando el polo de la función de transferencia del proceso al cero del controlador PI. Para conseguir este propósito, el tiempo integral debe ser igual a la constante de tiempo del proceso.

Para la obtención de la ganancia se debe tener en cuenta que el salto de duty cycle ha sido de 0,02 unidades, mientras que el salto experimentado por la corriente ha sido de 1,6 A. con estos datos, el cálculo a realizar para la primera aproximación de la ganancia proporcional es:

$$K_p = \frac{U}{Y} = \frac{0,02}{1,6} = 0,0125$$

Para el cálculo de la constante de tiempo se ha de tener en cuenta que el tiempo en el que la respuesta del sistema llega a un valor de $\pm 5\%$ el valor de establecimiento, en este caso corresponde con 0,0155s, corresponde con tres veces la constante de tiempo. Por tanto:

$$\tau = Ti = \frac{0,014}{3} = 0,0046$$

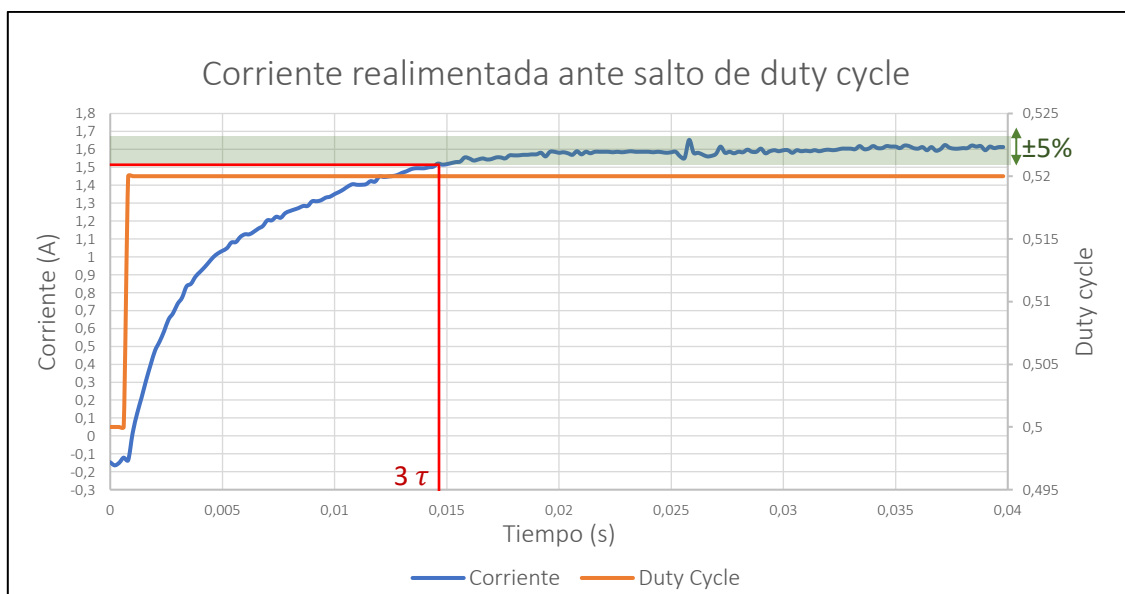


Figura 5.1 Corriente realimentada ante salto en el duty cycle

Los parámetros iniciales, por tanto, son: $K_c = 0,0125$ y $T_i = 0,0046$. Sin embargo, se deben realizar ensayos variando estos parámetros para observar la respuesta del sistema y escoger aquellos parámetros que proporcionen las mejores características posibles. La primera prueba se realiza con los siguientes valores: $K_c = 0,0125$ y $T_i = 0,004$ y se pide una consigna de 1A. La respuesta del sistema se muestra en la Figura 5.2. La conclusión que se obtiene de este ensayo es que la ganancia proporcional es insuficiente. El siguiente ensayo, Figura 5.3, se realiza manteniendo el tiempo integral y aumentando la ganancia hasta un valor de 0,0250. Se observa como la ganancia sigue siendo insuficiente. En el siguiente ensayo se prueba con una ganancia $K_c = 0,05$. El resultado, visible en la Figura 5.4 pone de manifiesto una fuerte sobreoscilación y una atenuación demasiado suave del comportamiento oscilante que hace evidente que la ganancia ahora es excesiva. El último ensayo, Figura 5.5, se realiza con una ganancia de valor 0,035 y puede verse que es el mejor ajuste, pues el sistema llega a la consigna en un tiempo de establecimiento muy breve y con una sobreoscilación adecuada, del orden del 20% del salto, que hace indicar que nos encontramos con un ajuste muy cercano al óptimo dinámico.

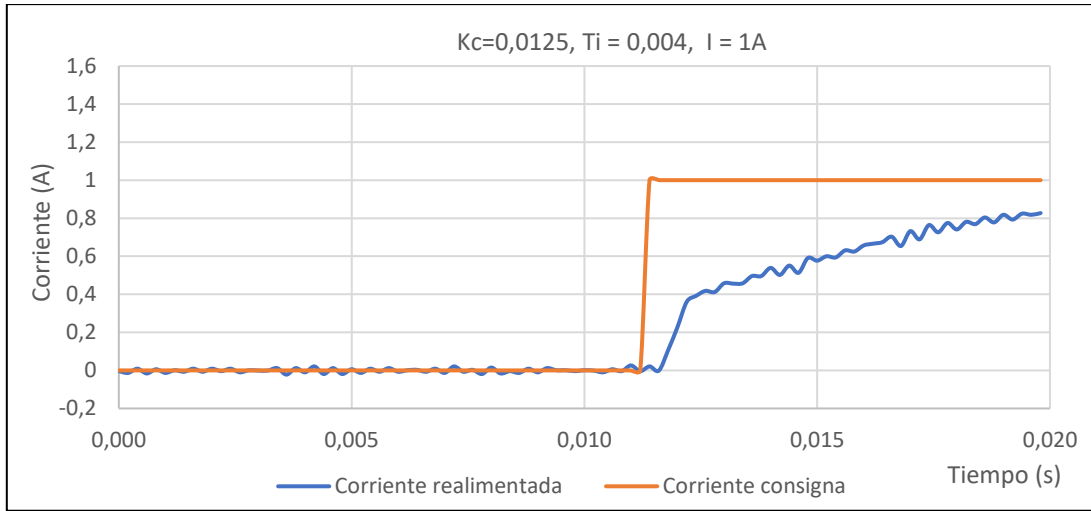


Figura 5.2 Ensayo lazo corriente $K_c=0,0125, T_i=0,004$

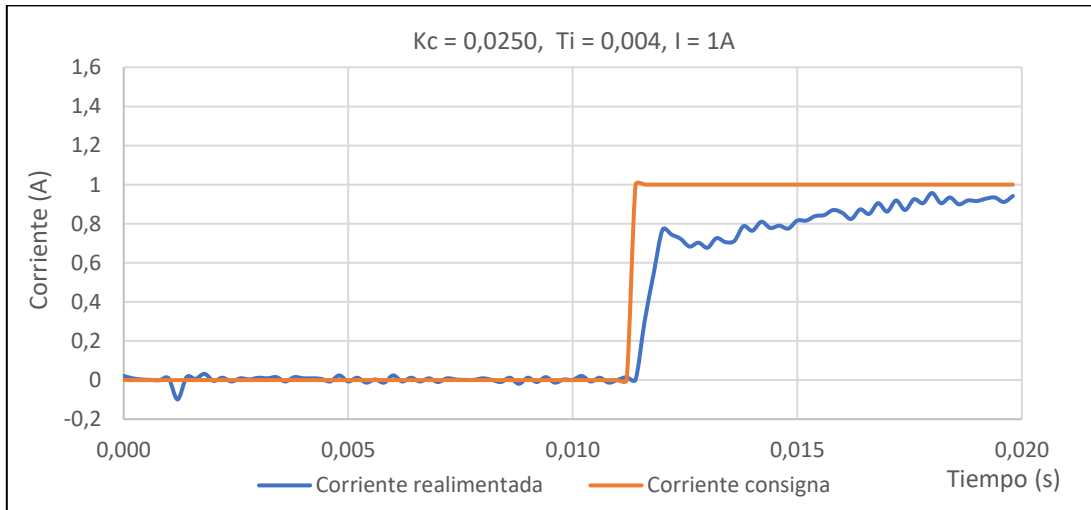


Figura 5.3 Ensayo lazo corriente $K_c=0,0250, T_i=0,004$

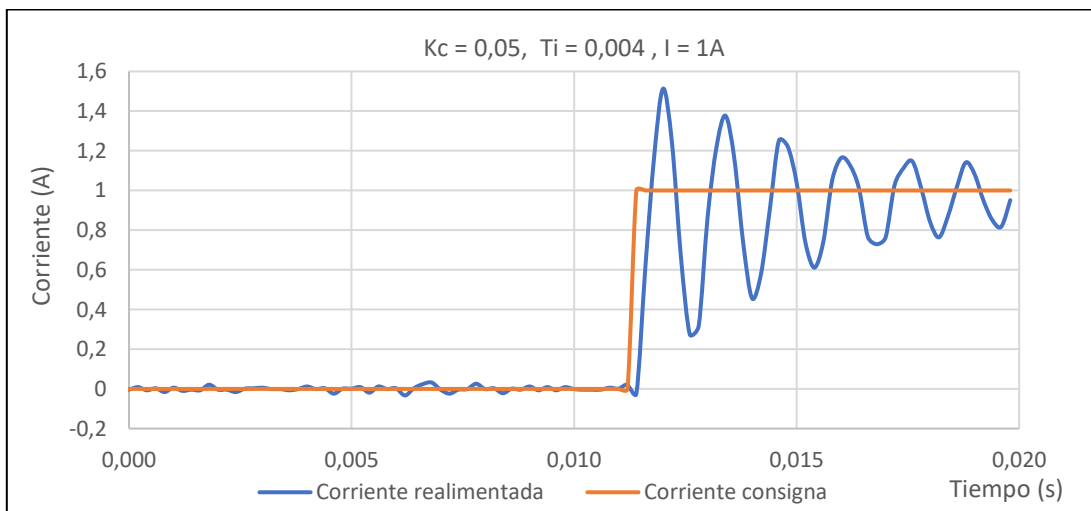


Figura 5.4 Ensayo lazo corriente $K_c=0,050, T_i=0,004$

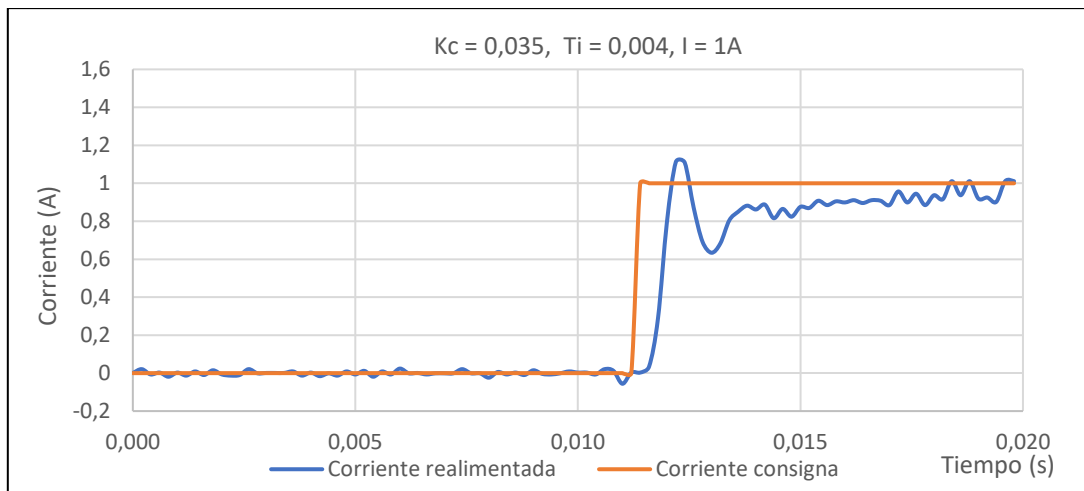


Figura 5.5 Ensayo lazo corriente $K_c=0,035$, $T_i=0,004$

Por tanto, los parámetros finales para el PI asociado al lazo de corriente son $K_c = 0,035$ y $T_i = 0,04$. Se ha de tener en cuenta que, en los ensayos realizados, siempre se observa un rizado debido a la propia tolerancia de los elementos empleados.

Para estos ensayos, así como para los que se muestran en los siguientes apartados los datos se obtienen a través de un registro realizado por el microcontrolador y posteriormente se envían por Bluetooth para su análisis.

5.5.2 Lazo de velocidad y control en cascada

El proceso a seguir para el lazo de velocidad es similar al realizado para el lazo de corriente, con la única diferencia de que el regulador PI se implementa dentro del control en cascada. Es por esto por lo que el ajuste de este regulador se realiza de manera completamente experimental siguiendo los criterios ya comentados en el diseño del sistema por simulación que permiten obtener un ajuste cercano al óptimo dinámico. Además, se debe tener en cuenta que la propia medida de velocidad realimentada, al regirse según un sistema con acción proporcional e integral, también presenta su propia dinámica. El código empleado para el regulador PI asociado a la velocidad se muestra en el Código 5.5. Puede observarse como los pasos realizados son iguales a los del regulador PI de corriente con la única diferencia de que ahora, la acción de control se asocia a la consigna de corriente tras saturarse entre los valores admisibles.

```

//LAZO VELOCIDAD

//se obtiene el error
V_err = V_consigna-Vrealim;

//se calculan las acciones proporcional e integral
V_Up = V_Kc*V_err;

//Antiwindup
if(abs(V_Upid - V_Upid_Sat)>0.1){

    //no se integra

}else{

    V_Ui = V_Ui+((V_Kc*Ts)/V_Ti)*V_err;

}

//Calculo de la acción de control
V_Upid = V_Up + V_Ui;

//Saturamos en el rango [I_lim_sup; I_lim_inf]
if(V_Upid>I_lim_sup) V_Upid_Sat=I_lim_sup;
else if(V_Upid<I_lim_inf)V_Upid_Sat=I_lim_inf;
else V_Upid_Sat=V_Upid;

//Asociamos la accion de control tras la saturación a la consigna de corriente
I_consigna = V_Upid_Sat;

//LAZO DE CORRIENTE

```

Código 5.5 Implementación del regulador PI de velocidad

El método para obtener los parámetros adecuados consiste en aumentar la ganancia proporcional hasta donde sea posible sin que aparezca sobreoscilación y una vez ajustada la ganancia, ir disminuyendo el tiempo integral hasta obtener una respuesta adecuada con una ligera sobreoscilación. En todos los ensayos se pide una consigna de 500 rpm a los 0,02 segundos.

Se comienza con una ganancia $K_c = 0,005$ y sin activar la acción integral, es decir, como si fuera un regulador tipo P. La Figura 5.6 pone de manifiesto que, a pesar de acercarse a la consigna, el ajuste no consigue que el motor sea capaz de alcanzarla.

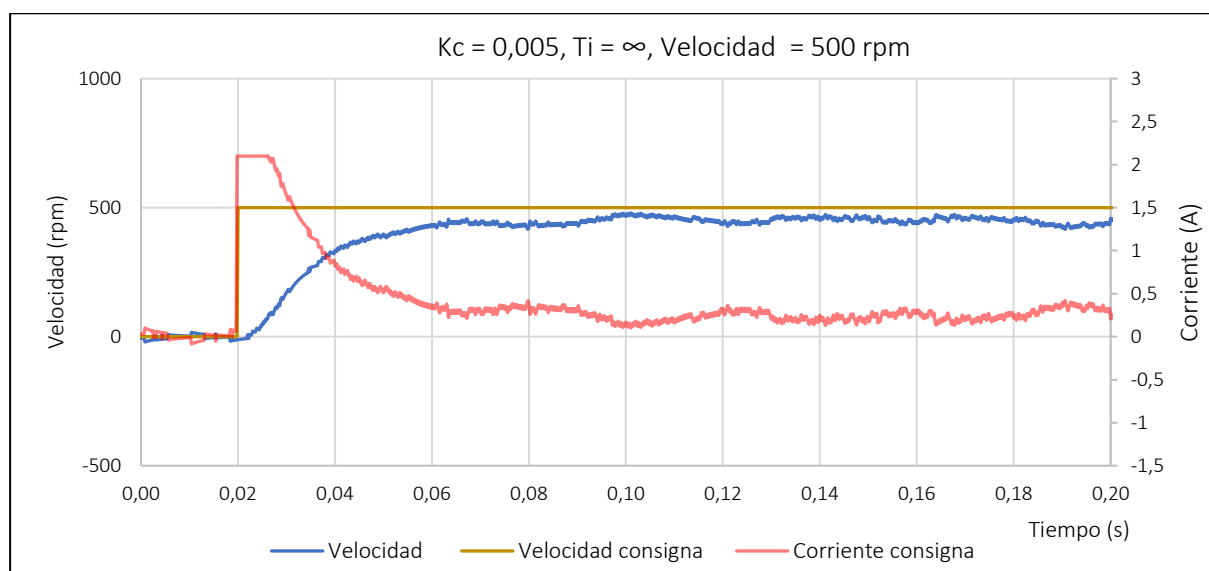


Figura 5.6 Ensayo lazo velocidad $K_c = 0,005$, $T_i = \infty$

En el siguiente ensayo se aumenta la ganancia proporcional hasta un valor de 0,01. En la Figura 5.7 se puede ver como la consigna de corriente se mantiene saturada durante más tiempo hasta que la velocidad se acerca lo suficiente a la consigna. Además, se puede ver como la respuesta con este ajuste es mejor que en el ensayo anterior. Sin embargo, se procede a aumentar la ganancia hasta 0,05 para evaluar si mejora aún más la respuesta del sistema.

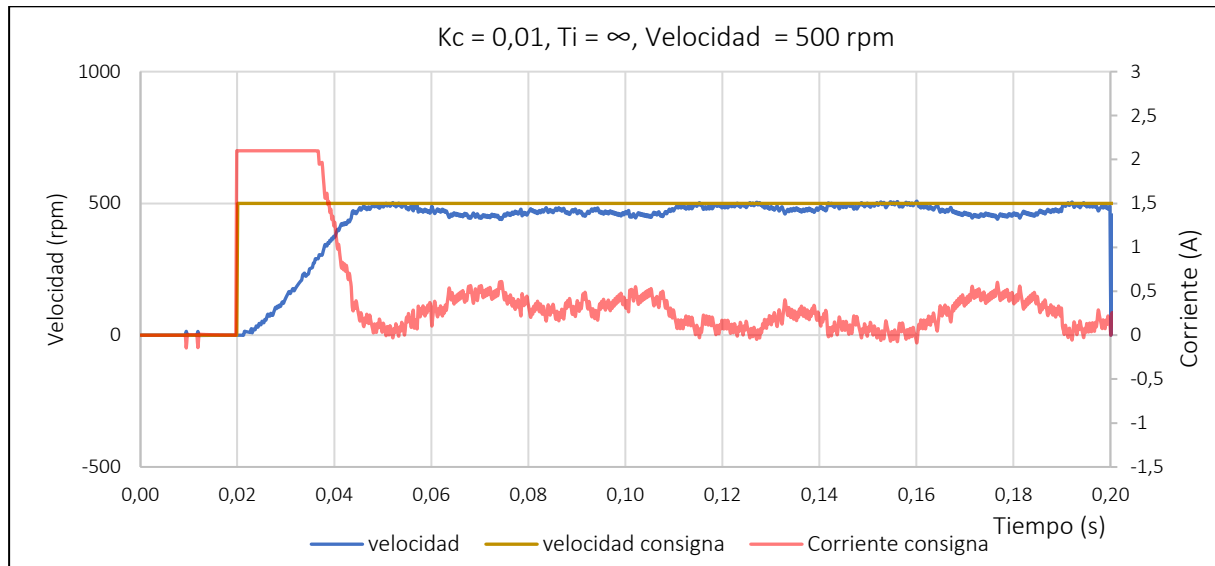


Figura 5.7 Ensayo lazo velocidad $K_c = 0,01$, $T_i = \infty$

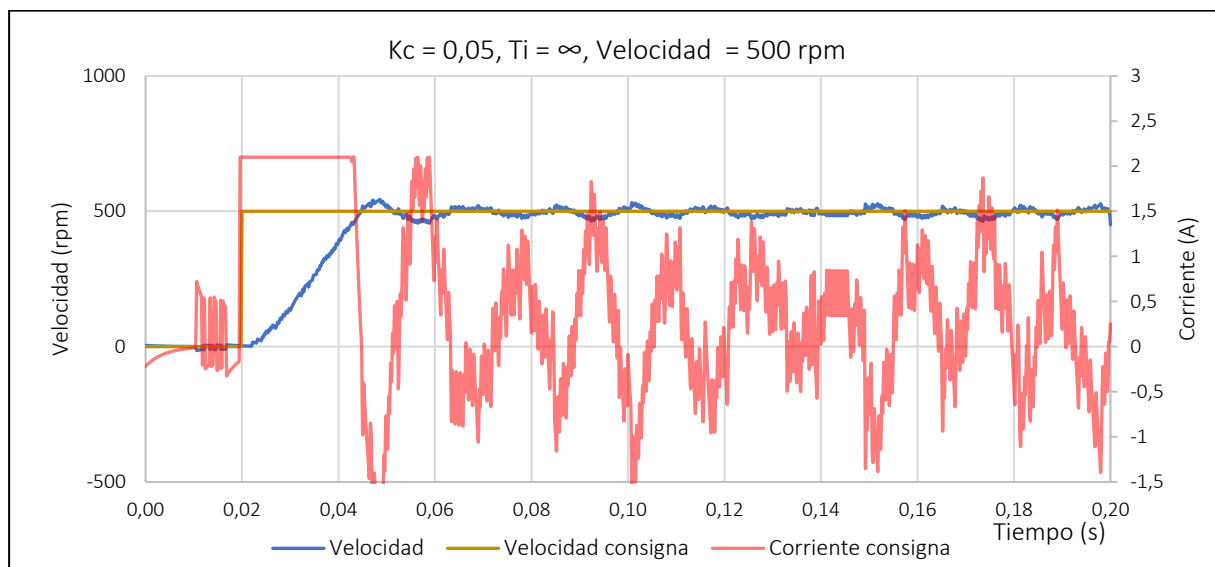


Figura 5.8 Ensayo lazo velocidad $K_c = 0,05$, $T_i = \infty$

La respuesta mostrada en la Figura 5.8 tras aumentar la ganancia no ofrece mejoras con respecto al ajuste anterior. Puede observarse una gran oscilación en la corriente de consigna que a su vez se traduce en un rizado en la velocidad. Además, debido a la saturación de la corriente, el tiempo de establecimiento tampoco mejora con respecto a la anterior ganancia. Por todos estos motivos, se

escoge la ganancia de 0,01, y, a partir de ella, se disminuye el tiempo integral. El primer ajuste disminuyendo el tiempo integral tiene como valores $K_c = 0,01$ y $T_i = 0,1$. El resultado se puede observar en la Figura 5.9. Se observa como la velocidad se ajusta mejor a la consigna respecto a lo visto en la Figura 5.7. Para el siguiente ensayo se toma un tiempo integral de valor 0,05 y se procede a evaluar si la respuesta sigue mejorando, o por el contrario empeora.

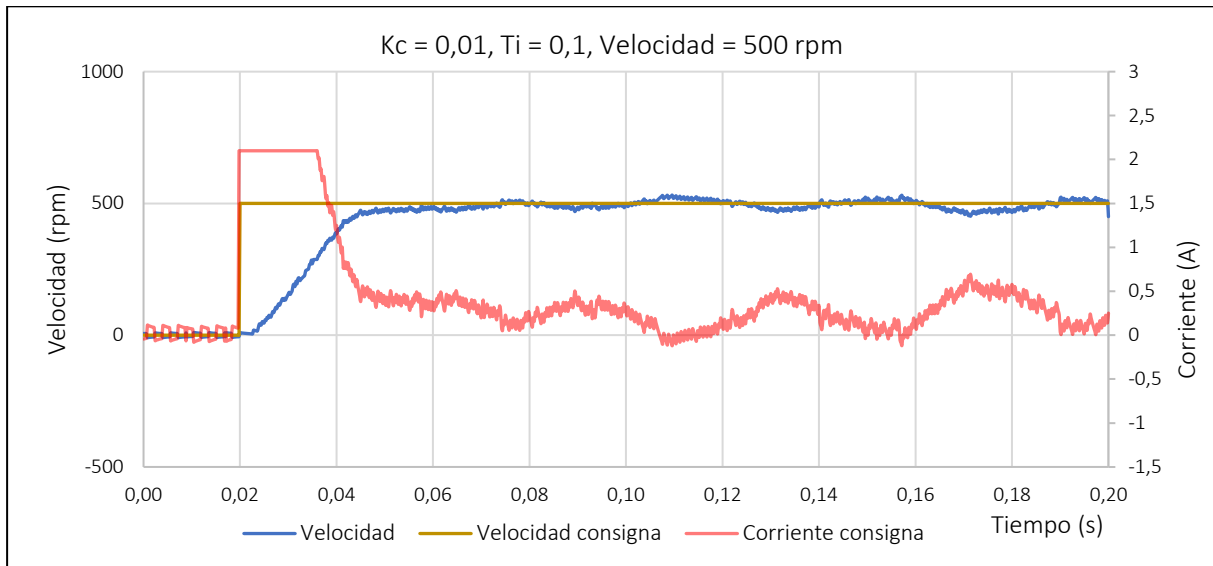


Figura 5.9 Ensayo lazo velocidad $K_c = 0,01$, $T_i = 0,1$

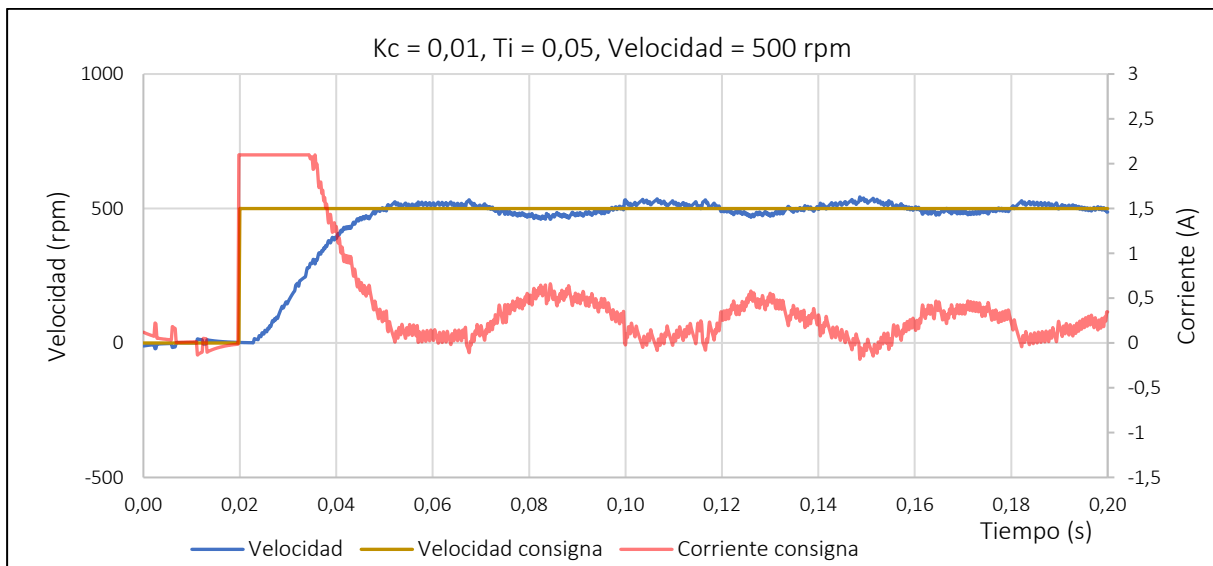


Figura 5.10 Ensayo lazo velocidad $K_c = 0,01$, $T_i = 0,05$

Al disminuir el tiempo integral a 0,05 se hace visible, según la Figura 5.10 que la respuesta no mejora, sino que presenta una oscilación mayor. Por tanto, el mejor ajuste obtenido hasta ahora es el consistente en los parámetros $K_c = 0,01$ y $T_i = 0,1$. Sin embargo, al realizar el ajuste de la ganancia proporcional, se han omitido ciertos valores puesto que se ha observado una buena respuesta con una ganancia de 0,01 y se han descartado ganancias mayores al observar que con una ganancia de 0,05 el sistema presentaba grandes oscilaciones. Para realizar un ajuste más fino, se procede a probar una ganancia intermedia entre ambos valores, $K_c = 0,025$. La Figura 5.11 muestra el resultado de este ensayo. Se puede observar que este ajuste ha mejorado la respuesta de todos los vistos anteriormente, por tanto, serán estos valores los que se utilicen para el regulador PI del lazo de velocidad.

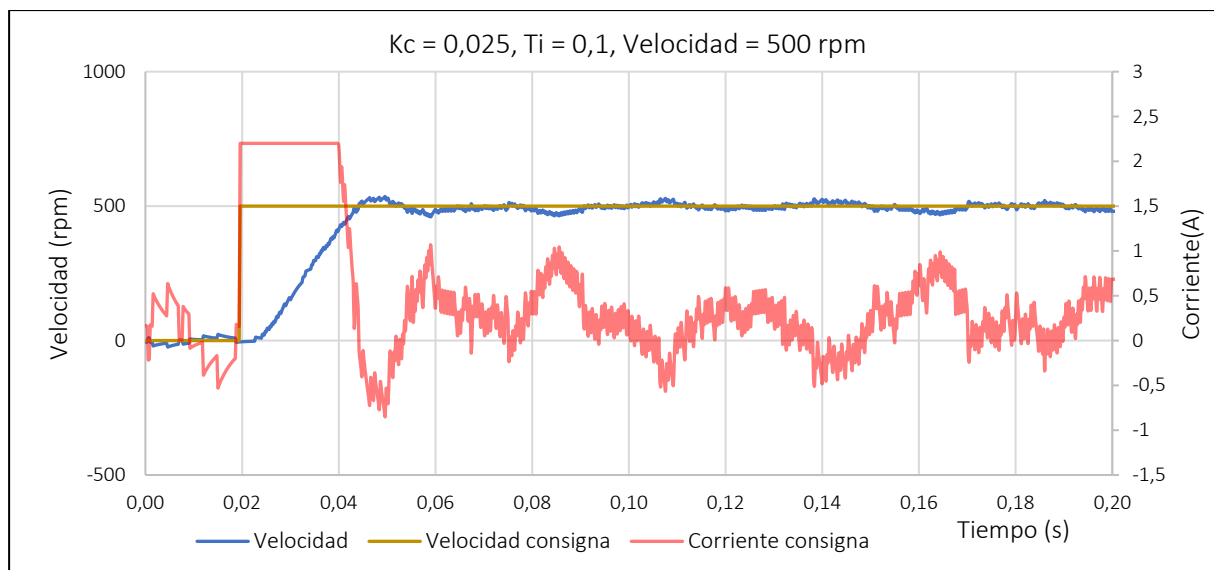


Figura 5.11 Ensayo lazo velocidad $K_c = 0,025, T_i = 0,1$

Con esto quedan completamente desarrolladas las funciones implementadas en el microcontrolador necesarias para el control del motor. Además, se han ajustado correctamente los reguladores PI referentes a los dos lazos presentes en el control en cascada.

CAPÍTULO 6: INTERFAZ DE USUARIO

6.1 INTRODUCCIÓN

En este capítulo se trata la comunicación entre el dispositivo móvil, que será el encargado de representar la información más relevante del sistema, así como de recibir las órdenes de control por parte del usuario y asignar así la velocidad de consigna al motor y el microcontrolador. Esta comunicación se lleva a cabo mediante Bluetooth por su relativa facilidad de funcionamiento. Sin embargo, es necesario comentar que, en un entorno industrial, la comunicación se llevaría a cabo o bien cableada o bien mediante Wi-fi, puesto que uno de los problemas que presenta la comunicación Bluetooth es su latencia relativamente alta.

6.2 ESTRUCTURA

El capítulo se estructurará en los siguientes apartados: Primero se tratará de manera general el modo de comunicación entre dispositivos en el apartado COMUNICACIÓN ENTRE DISPOSITIVOS. Después se tratará en detalle la implementación de esta comunicación tanto en el microcontrolador como en el dispositivo móvil en el apartado IMPLEMENTACIÓN DE LA COMUNICACIÓN. Este apartado se divide en la implementación en el microcontrolador y la implementación en el dispositivo móvil, subapartado en el cual se describirá la interfaz de usuario con todas sus funcionalidades.

6.3 COMUNICACIÓN ENTRE DISPOSITIVOS

Para realizar la comunicación el microcontrolador deberá enviar al dispositivo móvil información sobre el tiempo de funcionamiento del sistema, la velocidad del motor, la corriente de realimentación y el duty cycle, para que el dispositivo móvil represente y grafique esta información en tiempo real. El dispositivo móvil deberá enviar al microcontrolador información sobre el encendido y apagado del inversor y sobre la consigna de velocidad. En la Figura 6.1 se muestra un esquema de la comunicación entre dispositivos.

Para llevar a cabo esta comunicación, se establecerá un canal Bluetooth en el cual el microcontrolador realizará la función de servidor, y el dispositivo móvil la función de cliente. Una vez establecido el canal de comunicación, para que los dispositivos sean capaces de detectar la información que reciben, se establecerá un sistema consistente en enviar los datos junto con un identificador ASCII al inicio, el cual, consistirá en una letra mayúscula, y otro identificador al final del dato, que corresponde con la letra minúscula "f". De esta manera ambos dispositivos serán capaces de identificar qué tipo de dato están leyendo y qué longitud tiene.

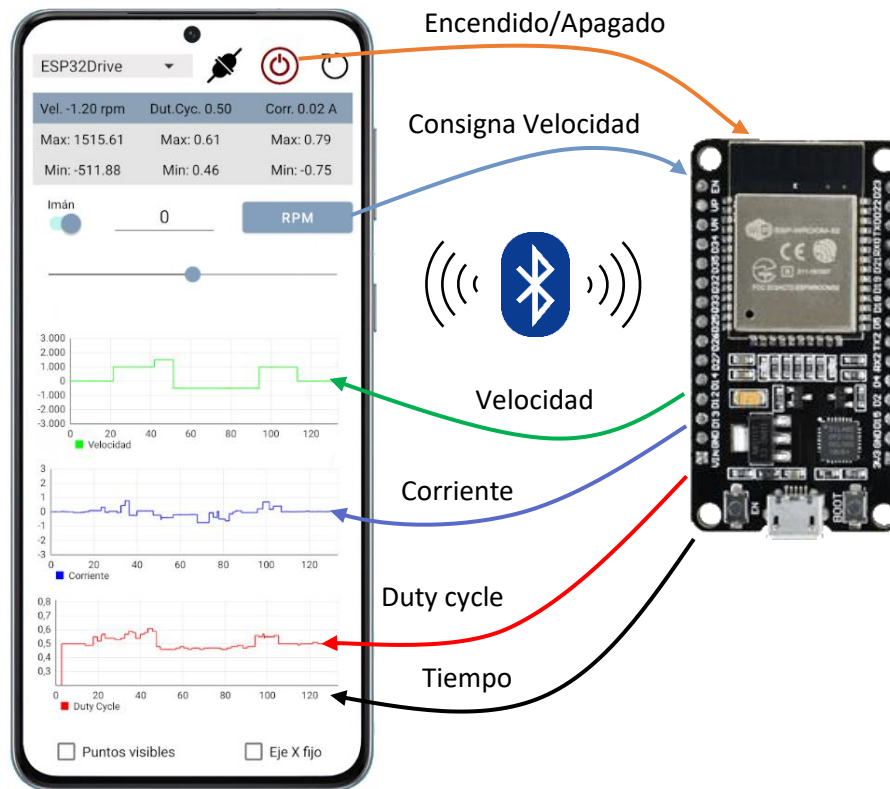


Figura 6.1 Esquema de comunicación entre interfaz y microcontrolador (elaboración propia)

6.4 IMPLEMENTACIÓN DE LA COMUNICACIÓN

Para realizar una comunicación efectiva, tanto el programa que ejecute el microcontrolador, como el que ejecute la aplicación móvil, deberán ser capaces de establecer un canal de comunicación Bluetooth, y transmitir los datos mediante el mismo, para ello deben sincronizarse para la lectura y envío de datos. En este apartado se tratará, por un lado, la implementación de la comunicación Bluetooth en el microcontrolador, y, por otro, la implementación en el dispositivo móvil.

6.4.1 Implementación en el microcontrolador

Para que las tareas de comunicación Bluetooth puedan realizarse de manera independiente y no interfieran con el resto del programa, se realizarán en un hilo de ejecución independiente, desde el cual se leerán y enviarán los datos necesarios. En el Código 6.1 se puede observar la declaración de la tarea "Task2" asociada al hilo de ejecución 0 (núcleo 0 del ESP32), donde se llevan a cabo las comunicaciones Bluetooth. El hecho de que el modelo utilizado del ESP32 sea de doble núcleo supone una ventaja fundamental ya que se puede reservar un núcleo (el 1) para realizar las tareas de control, especialmente las de control de corriente que requieren pasos de control y tiempos de respuesta muy breves sin ser interrumpido y sin correr el riesgo, por tanto, de que no se pueda atender a los rápidos cambios de la corriente en un tiempo muy corto por la ejecución de las tareas de comunicaciones que

se ejecutan en el otro núcleo (el 0) de forma independiente y sin interferir con el control a paso de control constante . Como se ha comentado anteriormente, para la comunicación, se han de establecer unos criterios para que tanto el dispositivo móvil como el microcontrolador puedan identificar si llegan datos nuevos y de qué tipo son esos datos. Es por eso por lo que, para el envío de datos desde el controlador, cada dato se envía comenzando por su identificador y terminando con la letra "f". Esto puede verse claramente en el Código 6.2.

```
// Comienza la comunicación Bluetooth
ESP_BT.begin("ESP32Drive"); //Nombre de la señal Bluetooth

//Se crea una tarea que será ejecutada en la función task2code() con prioridad 1 en el nucleo 0

xTaskCreatePinnedToCore(
    Task2code, /* función de la tarea */
    "Task2", /* nombre de la tarea */
    10000, /* tamaño de la tarea */
    NULL, /* parametro de la tarea */
    1, /* prioridad de la tarea*/
    &Task2, /* identificador de la tarea */
    0); /* nucleo en que se ejecuta la tarea */
}
```

Código 6.1 definición de la tarea Task2code para comunicación Bluetooth en Arduino IDE

```
void Task2code( void * pvParameters ){
    short posBT = 0;

    while(true){

        while(ESP_BT.available()){ //Check if we receive anything from Bluetooth

            BT_data[posBT]=ESP_BT.read();
            posBT++;
            Serial.println(BT_data[posBT]);

        }

        if(posBT){
            //Lectura de datos por Bluetooth
            /* ENCENDER*/
            if(BT_data[0] == 49 /*1*/){
                swon=true;
                ESP_BT.println("Encender");
            }
            /*APAGAR */
            else if(BT_data[0] == 48 /*0*/){
                swoff=true;
                ESP_BT.println("Apagar");
            }
            /*CONSIGNA VELOCIDAD */
            else if(BT_data[0] == 86 /*V*/){

                String Vcon="";

                for( int pos=1 ; pos<32 ; pos++ ){

                    if(BT_data[0]==102/*f*/){
                        break;
                    }
                    Vcon=Vcon+BT_data[pos];
                }
                if(abs(Vcon.toFloat())<VConslim){
                    I_U1=0;
                    V_U1=0;
                    V_consigna=Vcon.toFloat();
                }
            }
        }
        posBT = 0;

        //Envio de datos por bluetooth
        ESP_BT.print("T"+String(millis())+"f"); //Tiempo
        ESP_BT.println(";");
        ESP_BT.print("D"+String(dc)+"f"); //Duty Cycle
        ESP_BT.println(";");
        ESP_BT.print("I"+String(Irealim)+"f"); //Corriente
        ESP_BT.println(";");
        ESP_BT.print("V"+String(Vrealim)+"f"); //Velocidad
        ESP_BT.println(";");
        delay(100);
    }
}
```

Código 6.2 Tarea Task2code encargada de la comunicación Bluetooth

En el Código 6.2 se pueden observar dos partes diferenciadas. Por un lado, al iniciar la tarea se comprueba si hay datos disponibles mediante la llamada a la función “ESP_BT.avaible”. En caso de haber datos disponibles, estos se almacenan en un vector de caracteres, “BT_data”, el cual admite un máximo de 32 caracteres. Una vez almacenados en el vector, se procede a comprobar si se trata de datos útiles evaluando el carácter de la posición inicial. Si el carácter es un “1”, se ejecuta la orden de encender el inversor y poner en marcha el control, si se trata de un “0”, se apaga el inversor, y, si se trata de una “V”, se recorre el vector hasta llegar a una posición en la que el carácter almacenado sea una “f”. Después se evalúa el contenido desde la “V” hasta la “f”, y si se trata de un dato decimal válido entre -3000 y 3000, se asocia ese valor como nueva consigna de velocidad. Tras la comprobación del recibo de datos, se puede ver la parte destinada al envío. Como puede observarse, todos los datos se envían entre su identificador y la letra “f” indicando el final del dato.

6.4.2 Implementación en el dispositivo móvil

El dispositivo sobre el cual se implementa la interfaz gráfica es un dispositivo con sistema operativo Android. Para poder crear la aplicación se emplea Android Studio [14], un software dedicado para este propósito. El software permite programar en Java o en lenguaje Kotlin, en cualquier caso, se trata de lenguajes orientados a objetos. Para la creación de la aplicación de control del motor se ha empleado Java como lenguaje de programación.

La estructura básica de un proyecto en Android Studio se divide en dos partes. Por un lado, se desarrolla la interfaz gráfica mediante un archivo con extensión .xml, y, por otra parte, se programa la lógica de la aplicación en uno o varios archivos con extensión .java.

Primero, se comentará la parte correspondiente a la interfaz gráfica, explicando cada una de sus partes, así como las funcionalidades que ofrecen.

El desarrollo de la interfaz gráfica mediante el archivo .xml se basa en insertar, ordenar y editar los objetos que mejor se adecuen a la funcionalidad que se requiera. Estos objetos, posteriormente se identificarán en el fichero java para poder realizar las tareas necesarias y seguir la lógica adecuada. Cada objeto está enfocado a un uso concreto. Sin embargo, muchos de ellos comparten funcionalidades, por ejemplo, una imagen, (ImageView) puede realizar la función de botón (Button) o un interruptor (Switch) realiza acciones similares a una casilla de verificación (CheckBox).

Las principales funcionalidades que se buscan en la interfaz de usuario es que sea capaz de representar la información más relevante del sistema: Velocidad, corriente y duty cycle, y que, además, envíe señales de control, que permitan modificar el comportamiento del sistema mediante el cambio de la consigna de velocidad. Para cambiar la consigna se puede emplear una casilla editable o una barra deslizante. Además, mediante el botón “imán” se facilita el establecimiento de la consigna restringiendo el movimiento de la barra deslizante a múltiplos de 100. Para el envío de la consigna se utiliza el botón “rpm”.

Para poder entender de manera clara las diferentes funcionalidades y cómo se aplican, se muestra en la Figura 6.2 una imagen de la interfaz de usuario con sus diferentes elementos, los cuales se explican en la Tabla 6.1.

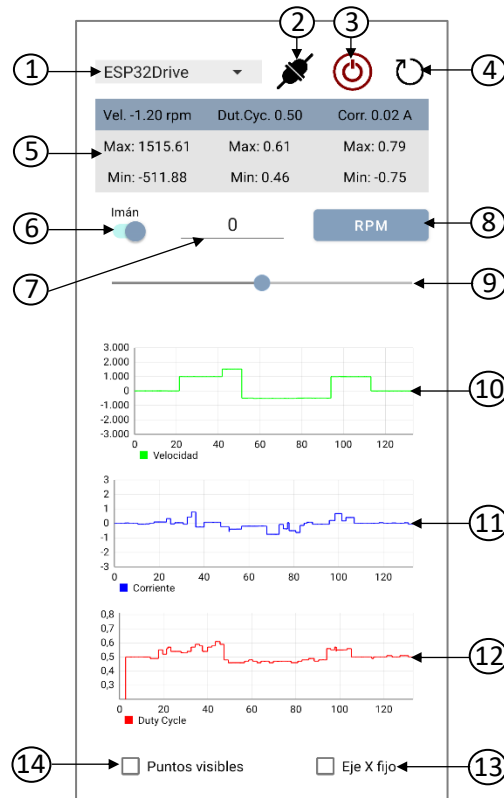


Figura 6.2 Interfaz de usuario

| Número | Nombre | Tipo | Función |
|--------|------------------------|----------------------|--|
| 1 | spDispositivos | Spinner | Permite seleccionar el dispositivo con el que conectar entre los emparejados |
| 2 | ivConectar | ImageView | Botón que permite conectar con el dispositivo seleccionado y crear un canal de comunicación |
| 3 | ivEncender | ImageView | Botón para encender y apagar el inversor y poner en marcha el control |
| 4 | ivRestart | ImageView | Botón que restablece los máximos y mínimos y limpia las gráficas |
| 5 | txVel, txCorr, txDc... | Conjunto de TextView | Tabla que muestra los valores de velocidad, corriente y Duty cycle con sus máximos y mínimos |
| 6 | swIman | Switch | Interruptor que, activado, limita el movimiento de la barra deslizante (elemento 9) a múltiplos de 100 |
| 7 | etVelConsig | EditText | Cuadro de texto editable para introducir manualmente el valor de la consigna de velocidad |
| 8 | btEnvio | Button | botón para enviar la consigna de velocidad mostrada en el elemento número 7 |
| 9 | sbBarraVel | SeekBar | Barra deslizante que permite establecer el valor de la consigna de velocidad entre -3000 y 3000 |
| 10 | graficaVel | LineChart | Gráfica que muestra en tiempo real los valores recibidos de velocidad |
| 11 | graficaCorr | LineChart | Gráfica que muestra en tiempo real los valores recibidos de corriente |
| 12 | graficaDC | LineChart | Gráfica que muestra en tiempo real los valores recibidos de duty cycle |
| 13 | chEjeFijo | CheckBox | Casilla de verificación que inhabilita el autoajuste del eje x y lo limita a 10 segundos |
| 14 | chPuntos | CheckBox | Casilla de verificación que permite mostrar los puntos concretos del envío en las gráficas |

Tabla 6.1 Elementos de la interfaz de usuario

En cuanto a la lógica implementada, al igual que en el código mostrado anteriormente para el envío y recibo de datos con el microcontrolador, en la aplicación móvil también es necesario un hilo secundario encargado exclusivamente del recibo de datos. Dado que el envío de datos es puntual y se lleva a cabo solo tras pulsar un botón, esta tarea puede mantenerse en el hilo de ejecución principal. Para la creación de varios hilos de ejecución en Android Studio, solo es necesario crear dos clases independientes. La clase MainActivity.java, es la encargada de representar todos los elementos en pantalla exceptuando las gráficas y de enviar cuando sea necesario las ordenes por Bluetooth, mientras que la clase ReciboDatos.java, es la encargada recibir datos, evaluarlos y actualizar las gráficas. Para la comunicación entre clases se ha decidido crear un objeto de tipo ReciboDatos en la clase principal cuando se haya creado el canal de comunicación Bluetooth, pasar como parámetros al objeto las gráficas y otros datos de interés y, a través de este objeto, ejecutar los métodos necesarios desde la clase MainActivity.java. Esto hace posible sincronizar las acciones relativas a mostrar los puntos, mantener el eje x de las gráficas fijo o limpiar el contenido de las gráficas.

A continuación, se mostrarán las partes más significativas del código implementado en Android Studio, las cuales son el envío y recibo de datos junto con la actualización de las gráficas.

En el Código 6.3 se muestra el extracto de código dentro del hilo principal que se ejecuta al pulsar el botón BtEnvío, con el cual se envía la consigna de velocidad.

```

btEnvio.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (conectado) {

            if (etValorEnvio.getText().length() > 0) {

                //Lee el valor en el EditText
                VelConsig = Double.parseDouble(String.valueOf(etValorEnvio.getText()));

                //se comprueba que no excede los límites
                if (VelConsig >= -3000 && VelConsig <= 3000) {

                    //Se actualiza la barra deslizable
                    sbBarraDeslizante.setProgress((int)(Math.round(VelConsig*resolucion)));
                    sbConsigna.setProgress((int)(Math.round(VelConsig*resolucionConsigna)));

                    String textoEnvio;
                    int caracter;
                    //Se crea la cadena a enviar
                    textoEnvio = "V"+ etValorEnvio.getText()+"f";

                    try {

                        //se envia la cadena
                        for (caracter = 0; caracter < textoEnvio.length(); caracter++) {
                            salidas.write(textoEnvio.charAt(caracter));
                        }

                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                }

            }

        }else{

            VelConsig = Double.parseDouble(String.valueOf(etValorEnvio.getText()));

            if (VelConsig >= -3000 && VelConsig <= 3000) {

                sbBarraDeslizante.setProgress((int) (Math.round(VelConsig * resolucion)));
                sbConsigna.setProgress((int)(Math.round(VelConsig*resolucionConsigna)));
            }

            MensajesPantalla( codigo: 0);

        }

    }
});

```

Código 6.3 Envío de consigna de velocidad en la interfaz gráfica

Se observa como lo primero que realiza el código es comprobar que existe una conexión con el microcontrolador. Si existe, lee el contenido de la casilla de texto editable donde se halla la consigna de velocidad. Si el dato leído es un dato válido entre -3000 y 3000, se enviará por bluetooth encabezado por su identificador, "V", y con la letra "F" al final, indicando el final del dato. Si aún no existe comunicación con el microcontrolador, simplemente cambia la posición de la barra deslizante.

En el Código 6.4 se muestra un extracto de la tarea secundaria que es realizada en el hilo de lectura de datos por Bluetooth

Este código empieza a ejecutarse en cuanto el microcontrolador y el dispositivo móvil establecen un canal de comunicación. El modo de funcionamiento del recibo de datos es el siguiente: Para empezar, se actualizan las gráficas con los datos disponibles, posteriormente se leen los caracteres disponibles hasta llegar a una letra correspondiente a un identificador. El carácter identificador se almacena y se procede a leer la información disponible hasta llegar a la letra "F", indicador de final de dato, y se guarda en la cadena "cadenaRecibida". Una vez almacenado tanto el identificador como el dato leído, se evalúa el identificador mediante la estructura switch(identificador) y, dependiendo del resultado, se actualizan los valores de velocidad, corriente, duty cycle o tiempo. Este código se repite en bucle con un tiempo de espera por iteración de 50ms para evitar el posible bloqueo de la aplicación por carga excesiva del procesador.

A lo largo de la memoria se han mostrado códigos relativos a la implementación de determinadas funciones y reguladores en el microcontrolador en el Arduino IDE y al desarrollo de la interfaz gráfica en Android Studio. En el anexo de este trabajo puede encontrarse un repositorio online en el que se encuentra el código del microcontrolador completo, todo el código de la aplicación en Android Studio y el ejecutable de la propia aplicación junto con su manual de usuario.


```

public void run(){
    while(true) {

        while (!pausa) {
            try {
                Thread.sleep( millis: 50);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            ActualizaGrafica();
            char caracter = '#';
            char identificador = '#';
            String cadenaRecibida = "";

            try {
                entradas.skip(entradas.available());
                //Lee caracteres hasta llegar a una V, I o D//
                while (caracter != 'V' && caracter != 'I' && caracter != 'D' && caracter != 'T') {
                    caracter = (char) entradas.read();
                }
                identificador = caracter; //guardamos el caracter con el que ha llegado//

                //lee hasta encontrar una f y se guarda en una cadena//
                while (caracter != 'f') {
                    caracter = (char) entradas.read();
                    if (caracter != 'f') {
                        cadenaRecibida = cadenaRecibida + caracter;
                    }
                }

                switch (identificador) {

                    case 'V':

                        tvVelocidad.setText("Vel. " + cadenaRecibida + " rpm");

                        vel = Double.parseDouble(cadenaRecibida);

                        if (vel > velMax) {
                            velMax = vel;
                        }
                        if (vel < velMin) {
                            velMin = vel;
                        }

                        tvVelocidadMax.setText("Max: " + Double.toString(velMax));
                        tvVelocidadMin.setText("Min: " + Double.toString(velMin));

                        break;

                    case 'D':

```

Código 6.4 Recepción de datos en el dispositivo móvil

CAPÍTULO 7: PRUEBAS DE FUNCIONAMIENTO y CONCLUSIONES

7.1 INTRODUCCIÓN

Una vez se ha montado el sistema de control con todos sus elementos, se ha desarrollado el código que permite realizar el control deseado y se han ajustado los parámetros correspondientes a los lazos de corriente y velocidad, solo queda validar el modelo final con ensayos que permitan poner de manifiesto el correcto funcionamiento del control diseñado para el motor BLDC.

7.2 ESTRUCTURA

Para la validación del control se realizarán dos ensayos. Por una parte, en el apartado SECUENCIA DE FASES, se comprobará que, ante un salto en la consigna de velocidad, existe un salto de la consigna de corriente, que provoca que las corrientes en las tres fases del motor se ajusten a dicha consigna siguiendo siempre la secuencia correspondiente. En el apartado CONTROL DE LA VELOCIDAD se comprobará que, ante el cambio de consigna de velocidad, el motor es capaz de establecer esa velocidad y mantenerse estable. Y, por último, en el apartado CONCLUSIONES se tendrán en cuenta estos resultados para validar el sistema diseñado para el control de velocidad del motor en relación al cumplimiento de cada uno de los objetivos específicos establecidos para este Trabajo Fin de Grado.

7.3 SECUENCIA DE FASES

Para la realización de estos ensayos se envía primero al motor una consigna de velocidad, y, una vez la velocidad del motor se estabilice, se pide un salto a una velocidad mayor. En estos ensayos se registra la corriente de consigna, así como las corrientes circulantes en las fases. Los ensayos son dos, en el primero el salto es de 500 a 1000 rpm y el segundo de 1000 a 1500 rpm. Los ensayos se muestran en la Figura 7.1 y la Figura 7.2.

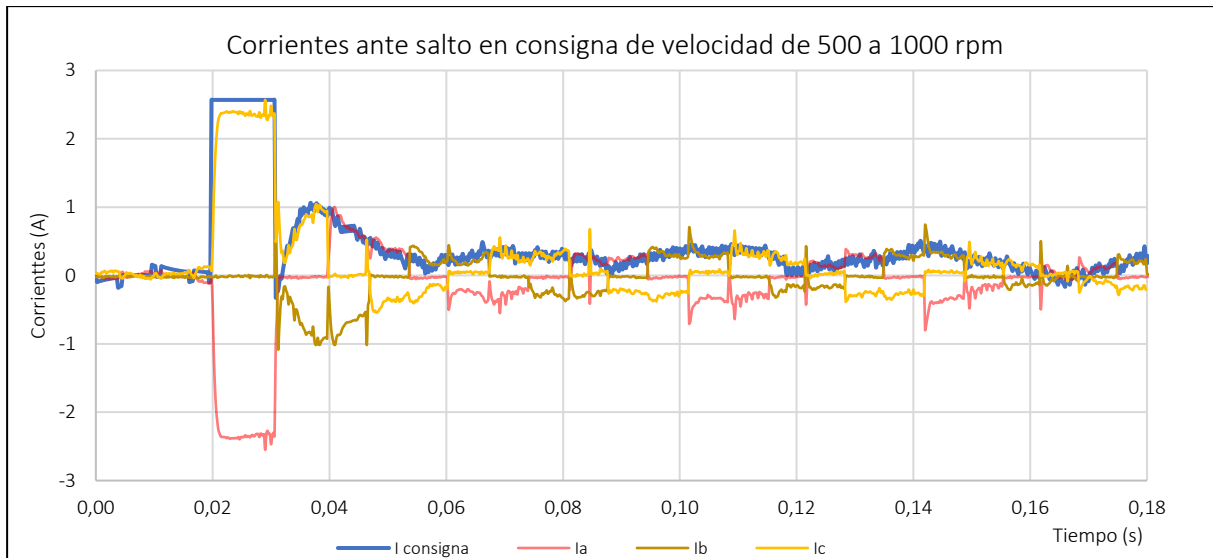


Figura 7.1 Ensayo secuencia de fases con salto de 500 a 1000 rpm

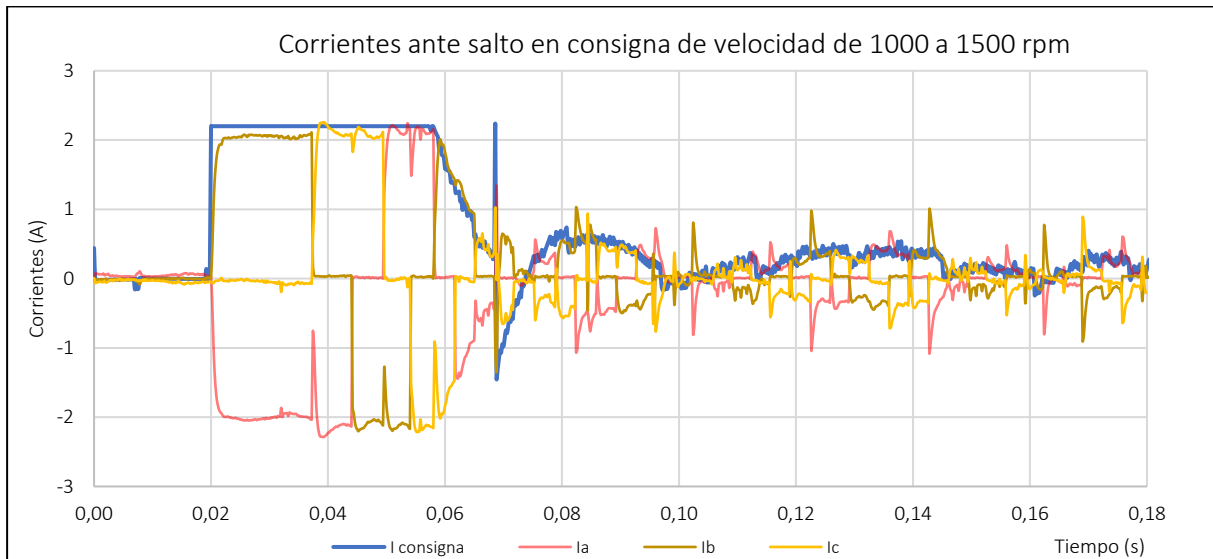


Figura 7.2 Ensayo secuencia de fases con salto de 1000 a 1500 rpm

Se pone de manifiesto en ambas gráficas como, ante el salto en la consigna de velocidad, la corriente consigna aumenta hasta saturar impidiendo que alcance valores inadmisibles para el inversor o las propias fases del motor. También se puede ver como la secuencia de corrientes es la adecuada, en cada fase se observa como la corriente evoluciona pasando siempre por dos sectores de conducción positiva, un sector de no conducción y dos sectores de conducción negativa.

7.4 CONTROL DE LA VELOCIDAD

En el apartado Lazo de velocidad y control en cascada del capítulo 5 se pudo comprobar que el motor alcanzaba de manera completamente satisfactoria las consignas de velocidad que se pedían y con una respuesta dinámica excelente. Sin embargo, todos los ensayos se realizaron para una velocidad de consigna de 500 rpm. Para este ensayo se pedirá una consigna mayor, de 1000 rpm para comprobar que el resultado es el adecuado.

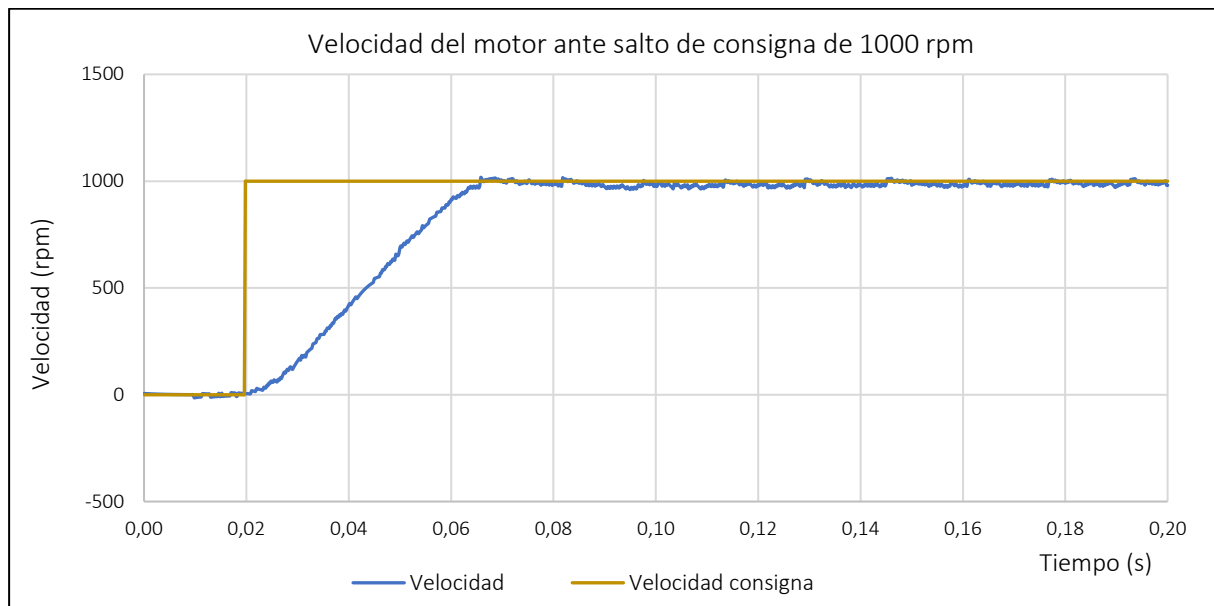


Figura 7.3 Ensayo validación control de la velocidad

Se muestra en la Figura 7.3 el comportamiento del motor en cuanto a velocidad tras solicitar una consigna de 1000 rpm desde el reposo. Se observa cómo tras un tiempo muy breve de sólo 40 ms, el cual viene dado por aceleración máxima aceptable que se consigue manteniendo durante todo el proceso una corriente prácticamente constante e igual a su valor nominal, la velocidad se ajusta a la consigna solicitada mostrando un comportamiento dinámico excelente con muy buena precisión y muy baja sobreoscilación.

7.5 CONCLUSIONES

Tras los apartados anteriores en los cuales se ha podido comprobar cómo los objetivos planteados para el control de la velocidad se cumplen se puede asumir que el modelo queda validado.

A continuación, se repasan los objetivos definidos al principio del trabajo y se comprobará que se han cumplido indicando en que capítulos se han alcanzado cada uno de estos hitos.

El primer objetivo consistía en el desarrollo de simulaciones del control a implementar, este hito se cumplió en el segundo capítulo. En este capítulo se utilizó el software Simulink para la realización de los ensayos. Hacia el final del capítulo, se pudo extraer la información necesaria para conocer los

elementos necesarios para construir el sistema de control en la realidad, con lo cual se cumple el segundo objetivo: identificar los elementos necesarios para el control real. El tercer objetivo es el relativo al montaje del sistema. Este hito se completa tras los capítulos 3 y 4. En el capítulo 3 se indicaba el montaje tanto del circuito electrónico correspondiente a la placa del microcontrolador como al circuito de potencia, mientras que en el capítulo 4 se exponía el montaje del sensor de posición mediante técnicas de impresión 3d para la realización de un acoplamiento al motor. El cuarto objetivo es el referente al desarrollo del código necesario para el control del motor. En este caso, este objetivo se ha cumplido entre el capítulo 4, en el cual se mostraban los códigos referentes a la realimentación del sector y velocidad y el capítulo 5, capítulo en el que se muestran todas las funciones a realizar por el microcontrolador incluido el ajuste de los reguladores PI. Por tanto, el quinto objetivo, que es el ajuste de los reguladores, queda también cumplido. El desarrollo de la interfaz gráfica junto con el código necesario constituye el sexto objetivo. Este se cumple tras el capítulo 6, dedicado enteramente a este cometido. Por último, el objetivo consistente en probar y validar el sistema queda completamente cumplido en este último capítulo.

REFERENCIAS

- [1] J. Martinez-Roman, A. Sapena-Baño y J. Burriel-Valencia, *Modelado de Máquinas de Alterna para Control Electrónico de Accionamientos Eléctricos*, Valencia: Editorial UPV, 2021.
- [2] Mathworks, «BLDC Three-winding brushless DC motor with trapezoidal flux distribution,» 29 05 2022. [En línea]. Available: <https://es.mathworks.com/help/physmod/sps/ref/bldc.html>.
- [3] Texas Instruments, «Trapezoidal Control of BLDC Motors,» 15 4 2011. [En línea]. Available: <https://www.ti.com/lit/an/sprabz4/sprabz4.pdf?ts=1654733975266>. [Último acceso: 9 6 2022].
- [4] Espressif, «ESP8266 A cost-effective and highly integrated Wi-Fi MCU for IoT applications,» 29 05 2022. [En línea]. Available: <https://www.espressif.com/en/products/socs/esp8266>.
- [5] Espressif, «ESP32 A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications,» 29 05 2022. [En línea]. Available: <https://www.espressif.com/en/products/socs/esp32>.
- [6] Arduino, «Nano 33 BLE,» [En línea]. Available: <https://docs.arduino.cc/hardware/nano-33-ble>.
- [7] Last Minute Engineers, «ESP32 Pinout Reference,» 6 2 2022. [En línea]. Available: <https://lastminuteengineers.com/esp32-pinout-reference/>. [Último acceso: 5 6 2022].
- [8] LEM, «Current Transducer LTSR 6-NP,» 22 9 2017. [En línea]. Available: https://www.lem.com/sites/default/files/products_datasheets/ltsr_6-np.pdf. [Último acceso: 5 6 2022].
- [9] R. Santos y S. Santos, «ESP32 ADC – Read Analog Values with Arduino IDE,» 31 5 2019. [En línea]. Available: <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>. [Último acceso: 5 6 2022].
- [10] EuroEncoder, «Encoders ópticos,» 5 7 2021. [En línea]. Available: <https://euroencoder.com/encoders-opticos/>. [Último acceso: 2 6 2022].
- [11] AKM, «Principle and advantages of magnetic encoder,» 7 9 2020. [En línea]. Available: <https://www.akm.com/eu/en/products/rotation-angle-sensor/tutorial/magnetic-encoder/>. [Último acceso: 5 6 2022].
- [12] MOUSER, «EMS22A50-M25-LD6,» 16 10 2014. [En línea]. Available: <https://www.mouser.es/ProductDetail/Bourns/EMS22A50-M25-LD6?qqs=%2FxFxQVPCMPNzh1DOMfMt1xZg%3D%3D>. [Último acceso: 5 6 2022].

- [13] BOURNS, « EMS22A - Non-Contacting Absolute Encoder,» 16 9 2014. [En línea]. Available: <https://www.mouser.es/datasheet/2/54/EMS22A-50229.pdf>. [Último acceso: 6 6 2022].
- [14] Android, «Introducción a Android Studio,» 17 5 2021. [En línea]. Available: <https://developer.android.com/studio/intro?hl=es-419>. [Último acceso: 7 6 2022].



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Diseño de un sistema de control de inversor para máquina de corriente continua sin escobillas de 200W

PRESUPUESTO

Trabajo Final de Grado

DAVID RODRÍGUEZ BASCUÑANA

Junio, 2022

ÍNDICE DE CONTENIDOS

| | |
|---|---|
| INTRODUCCIÓN | 1 |
| UNIDAD DE OBRA 1. ENSAYOS EN SIMULINK Y CÓDIGO EN MATLAB | 1 |
| UNIDAD DE OBRA 2. IMPRESIÓN 3D | 1 |
| UNIDAD DE OBRA 3. CIRCUITO ELECTRÓNICO | 2 |
| UNIDAD DE OBRA 4. CIRCUITO DE POTENCIA..... | 3 |
| UNIDAD DE OBRA 5. CÓDIGO DEL MICROCONTROLADOR E INTERFAZ GRÁFICA..... | 3 |
| UNIDAD DE OBRA 6. ENSAYOS Y AJUSTE DE REGULADORES | 4 |
| UNIDAD DE OBRA 7. REDACCIÓN DE LA MEMORIA..... | 4 |
| DIVISIÓN POR TIPO DE RECURSO | 4 |
| PRESUPUESTO | 5 |

INTRODUCCIÓN

El presupuesto es una parte fundamental de cualquier proyecto, puesto que en él se refleja el coste económico asociado a la realización del mismo. El presupuesto de este trabajo integra todos los costes divididos en siete unidades de obra.

Es necesario comentar que existe un único presupuesto, que es el referente a la realización del trabajo en su conjunto formado por una serie de unidades de obra. Por tanto, el cálculo del importe de cada unidad se puede obviar puesto que la medición de cada unidad de obra es unitaria y el coste unitario es el relativo al coste mostrado al final de cada tabla correspondiente a las distintas unidades de obra.

En cada unidad de obra existe una división entre recursos materiales y humanos. En los recursos humanos, las tareas mostradas son realizadas por un ingeniero en Tecnologías Industriales exceptuando las de supervisión, realizadas por el director de proyecto. Para el cálculo de los precios unitarios de los recursos materiales inventariables se emplea la fórmula:

$$\left(\frac{\text{Tiempo de uso del equipo}}{\text{Periodo de amortización}} \right) \cdot \text{Coste del equipo}$$

| UNIDAD DE OBRA 1. ENSAYOS EN SIMULINK Y CÓDIGO EN MATLAB | | | | |
|--|------------------------------------|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Ordenador personal | 15,00 €/mes | 3 meses | 45,00 € |
| | Software Matlab | 67,00 €/mes | 3 meses | 201,00 € |
| | Software Simulink | 100,00 €/mes | 3 meses | 300,00 € |
| | Software Simscape | 67,00 €/mes | 3 meses | 201,00 € |
| HUMANOS | Estudios previos | 27,00 €/h | 12 h | 324,00 € |
| | Ensayos y simulaciones en Simulink | 27,00 €/h | 40 h | 1.080,00 € |
| | Desarrollo de código en Matlab | 27,00 €/h | 12 h | 324,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 10 h | 1.000,00 € |
| COSTE UNIDAD DE OBRA 1 | | | | 3.475,00 € |

| UNIDAD DE OBRA 2. IMPRESIÓN 3D | | | | |
|--------------------------------|------------------------------------|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Impresora Prusa i3 MK3s+kit | 0,25 €/h | 3 h | 0,75 € |
| | Plástico Prusament PLA | 0,03 €/g | 12 g | 0,37 € |
| | Licencia Onshape | 125,00 €/mes | 1 mes | 125,00 € |
| | Software PrusaSlicer | 0,00 €/u | 1 u | 0,00 € |
| HUMANOS | Estudios previos | 27,00 €/h | 10 h | 270,00 € |
| | Diseño de piezas | 27,00 €/h | 12 h | 324,00 € |
| | Impresión de piezas | 27,00 €/h | 3 h | 81,00 € |
| | Ensamblaje de piezas en el sistema | 27,00 €/h | 5 h | 135,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 2 h | 200,00 € |
| COSTE UNIDAD DE OBRA 2 | | | | 1.136,12 € |

| UNIDAD DE OBRA 3. CIRCUITO ELECTRÓNICO | | | | |
|---|---|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | ESP32 DevKit V1 | 10,59 €/u | 1 u | 10,59 € |
| | Sensor efecto hall LTSR 6-NP | 18,37 €/u | 2 u | 36,74 € |
| | Encoder EMS22A50-M25-Ld6 | 33,74 €/u | 1 u | 33,74 € |
| | Puertas lógicas SN7432N | 2,82 €/u | 1 u | 2,82 € |
| | Resistencias 5k1 Ω | 0,18 €/u | 2 u | 0,36 € |
| | Resistencias 2k4 Ω | 0,22 €/u | 2 u | 0,44 € |
| | Resistencias 1k8 Ω | 0,20 €/u | 2 u | 0,40 € |
| | Condensadores cerámicos 47pF | 0,24 €/u | 2 u | 0,48 € |
| | Conectores pin hembra | 0,01 €/u | 34 u | 0,34 € |
| | Conectores pin macho | 0,01 €/u | 34 u | 0,34 € |
| | Cable plano de 34 pines | 6,80 €/u | 1 u | 6,80 € |
| | Placa de soldadura de 15x20 cm | 5,20 €/u | 1 u | 5,20 € |
| | Caja de conexiones | 3,00 €/u | 1 u | 3,00 € |
| | Cables de cobre 6x0,02mm ² | 0,37 €/m | 1 m | 0,37 € |
| | Conectores atornillados macho hembra | 0,50 €/u | 4 u | 2,00 € |
| | Estaño para soldadura | 0,05 €/g | 100 g | 5,00 € |
| | Soldador eléctrico tipo F | 20,29 €/u | 1 u | 20,29 € |
| | Funda termorretráctil | 0,50 €/m | 0,2 m | 0,10 € |
| | Cargador USB 5V | 4,50 €/u | 1 u | 4,50 € |
| | Tornillos | 0,07 €/u | 3 u | 0,21 € |
| Bloque de terminales enchufables hembra de 4 vías | 1,10 €/u | 1 u | 1,10 € | |
| HUMANOS | Estudios previos | 27,00 €/h | 12 h | 324,00 € |
| | Montaje y soldaduras | 27,00 €/h | 10 h | 270,00 € |
| | Pruebas de funcionamiento y circuitería | 27,00 €/h | 2 h | 54,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 5 h | 500,00 € |
| COSTE UNIDAD DE OBRA 3 | | | | 1.282,82 € |

| UNIDAD DE OBRA 4. CIRCUITO DE POTENCIA | | | | |
|--|---|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Inversor STEVAL-IHM028V2 | 345,80 €/u | 1 u | 345,80 € |
| | Regleta eléctrica de 4 enchufes | 5,86 €/u | 1 u | 5,86 € |
| | Bloque de terminales enchufables macho de 4 vías | 3,78 €/u | 2 u | 7,56 € |
| | Bloque de terminales enchufables macho de 3 vías | 2,91 €/u | 1 u | 2,91 € |
| | Manguera cable cobre 3x(1,5)mm ² | 1,20 €/m | 1,00 m | 1,20 € |
| | Cable cobre 1,5mm ² | 0,60 €/m | 1,00 m | 0,60 € |
| | Cable de alimentación con conector Schuko | 7,65 €/u | 1 u | 7,65 € |
| | Motor SGMAH-02AAA61D-OY | 350,00 €/u | 2 u | 700,00 € |
| | Bancada de ensayos | 500,00 €/u | 1 u | 500,00 € |
| | Magneto Térmico unipolar Merlin Gerin Multi9 K32a | 4,95 €/u | 2 u | 9,90 € |
| | Acople elástico entre motores | 35,40 €/u | 1 u | 35,40 € |
| HUMANOS | Estudios previos | 27,00 €/h | 2 h | 54,00 € |
| | Montaje y conexionado | 27,00 €/h | 6 h | 162,00 € |
| | Pruebas de funcionamiento y circuitería | 27,00 €/h | 2 h | 54,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 6 h | 600,00 € |
| COSTE UNIDAD DE OBRA 4 | | | | 2.486,88 € |

| UNIDAD DE OBRA 5. CÓDIGO DEL MICROCONTROLADOR E INTERFAZ GRÁFICA | | | | |
|--|------------------------------------|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Ordenador personal | 15,00 €/mes | 6 meses | 90,00 € |
| | Software Arduino IDE | 0,00 €/u | 1 u | 0,00 € |
| | Software Android Studio | 0,00 €/u | 1 u | 0,00 € |
| | Dispositivo móvil personal | 29,13 €/m | 6 meses | 174,78 € |
| | Cable USB A - Micro USB B | 5,99 €/u | 1 u | 5,99 € |
| | Cable USB A - USB C | 6,99 €/u | 1 u | 6,99 € |
| HUMANOS | Estudios previos | 27,00 €/h | 5 h | 135,00 € |
| | Desarrollo del código Arduino | 27,00 €/h | 20 h | 540,00 € |
| | Desarrollo de la interfaz gráfica | 27,00 €/h | 20 h | 540,00 € |
| | Pruebas de funcionamiento | 27,00 €/h | 10 h | 270,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 2 h | 200,00 € |
| COSTE UNIDAD DE OBRA 5 | | | | 1.962,76 € |

| UNIDAD DE OBRA 6. ENSAYOS Y AJUSTE DE REGULADORES | | | | |
|---|---|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Osciloscopio PicoScope serie 2000 | 139,00 €/u | 1 u | 139,00 € |
| | Sonda de osciloscopio | 26,00 €/u | 2 u | 52,00 € |
| | Ordenador personal | 15,00 €/m | 3 meses | 45,00 € |
| | Resistencias 10kΩ | 0,04 €/u | 3 u | 0,12 € |
| | Fuente de alimentación corriente continua | 158,50 €/u | 1 u | 158,50 € |
| | Cables de cobre 6x0,02mm ² | 0,37 €/m | 1 m | 0,37 € |
| | Dispositivo movil | 29,13 €/m | 3 m | 87,39 € |
| | Software PicoScope 6 | 0,00 €/u | 1 u | 0,00 € |
| HUMANOS | Estudios previos | 27,00 €/h | 5 h | 135,00 € |
| | Ensayos corriente realimentada | 27,00 €/h | 12 h | 324,00 € |
| | Ensayos posición y velocidad realimentada | 27,00 €/h | 10 h | 270,00 € |
| | Ajuste de reguladores | 27,00 €/h | 20 h | 540,00 € |
| | Ensayos para validación del modelo | 27,00 €/h | 10 h | 270,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 5 h | 500,00 € |
| COSTE UNIDAD DE OBRA 6 | | | | 2.521,38 € |

| UNIDAD DE OBRA 7. REDACCIÓN DE LA MEMORIA | | | | |
|---|------------------------------------|-----------------|----------|-------------------|
| RECURSOS | Concepto | Precio unitario | Cantidad | Coste |
| MATERIALES | Ordenador personal | 15,00 €/m | 2 meses | 30,00 € |
| HUMANOS | Redacción de la memoria | 27,00 €/h | 70 h | 1.890,00 € |
| | Supervisión (director de proyecto) | 100,00 €/h | 20 h | 2.000,00 € |
| COSTE UNIDAD DE OBRA 7 | | | | 3.920,00 € |

DIVISIÓN POR TIPO DE RECURSO

Los costes mostrados de todas las unidades de obra pueden dividirse en costes materiales y costes humanos. La división de los costes queda de la siguiente forma:

| Tipo de recurso | Coste |
|-----------------|--------------------|
| MATERIAL | 3.414,96 € |
| HUMANO | 13.370,00 € |
| TOTAL | 16.784,96 € |

PRESUPUESTO

A continuación, se muestra el presupuesto total a partir de las distintas unidades de obra y con los correspondientes gravámenes.

| PRESUPUESTO | |
|--|--------------------|
| Concepto | Coste |
| UNIDAD DE OBRA 01. | 3.475,00 € |
| UNIDAD DE OBRA 02. | 1.136,12 € |
| UNIDAD DE OBRA 03. | 1.282,82 € |
| UNIDAD DE OBRA 04. | 2.486,88 € |
| UNIDAD DE OBRA 05. | 1.962,76 € |
| UNIDAD DE OBRA 06. | 2.521,38 € |
| UNIDAD DE OBRA 07. | 3.920,00 € |
| Presupuesto de ejecución material | 16.784,96 € |
| Gastos generales (12%) | 2.014,20 € |
| Beneficio industrial (6%) | 1.007,10 € |
| Presupuesto de ejecución por contrata | 19.806,26 € |
| IVA (21%) | 4.159,31 € |
| Presupuesto base licitación | 23.965,57 € |

El presupuesto asciende a un total de **VEINTITRÉS MIL NOVECIENTOS SESENTA Y CINCO EUROS CON CINCUENTA Y SIETE CÉNTIMOS.**



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

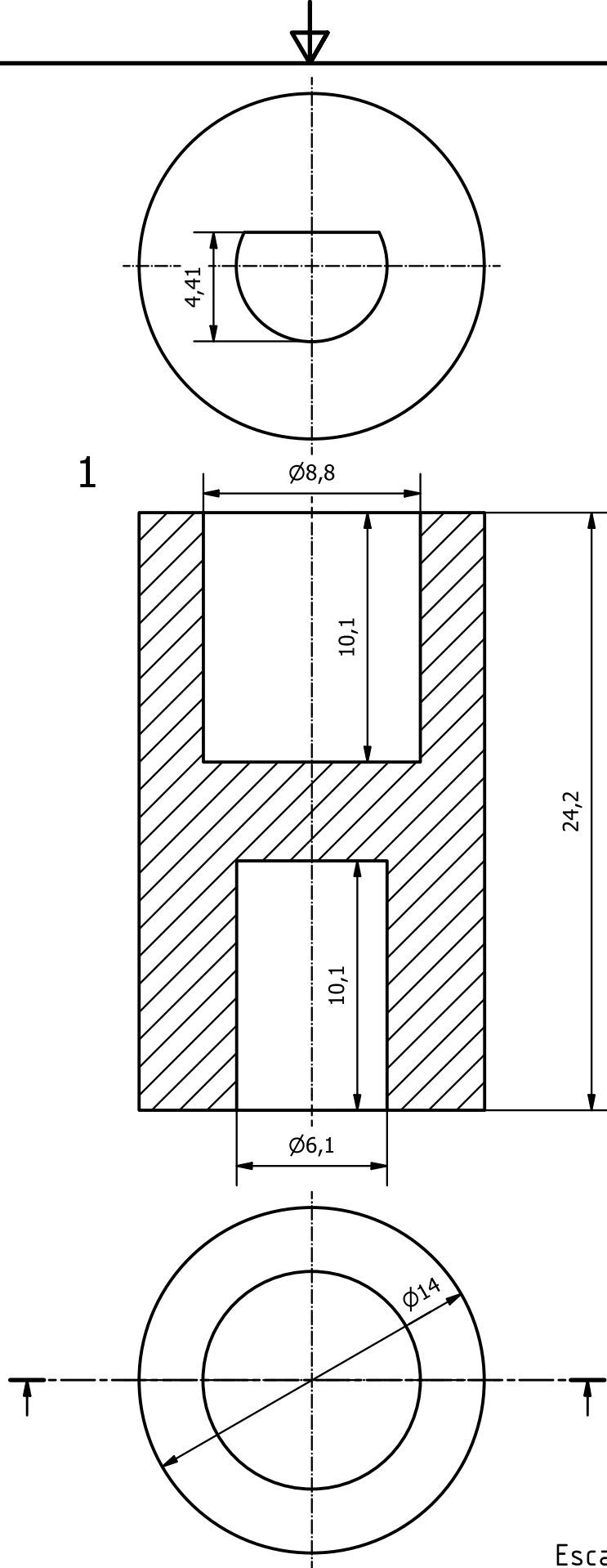
Diseño de un sistema de control de inversor para máquina de corriente continua sin escobillas de 200W

PLANOS

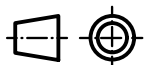
Trabajo Final de Grado
DAVID RODRÍGUEZ BASCUÑANA
Junio, 2022


ÍNDICE DE CONTENIDOS

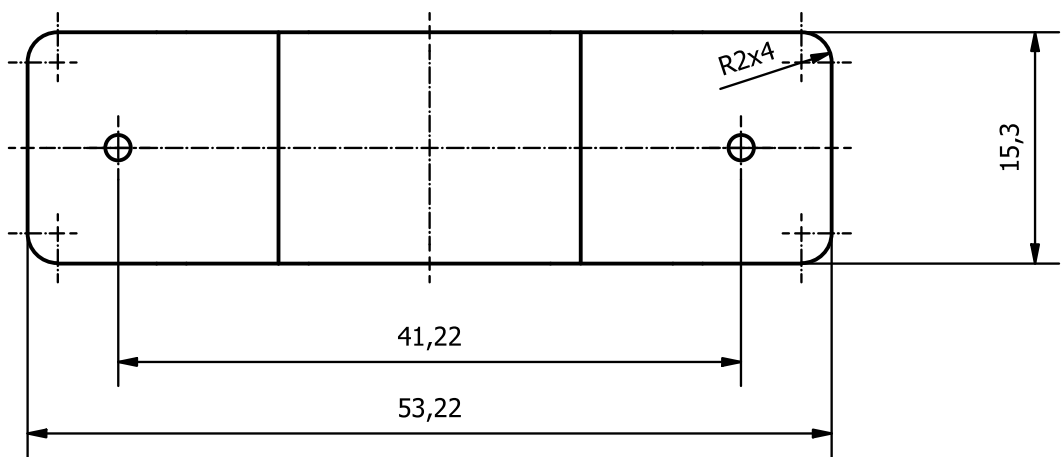
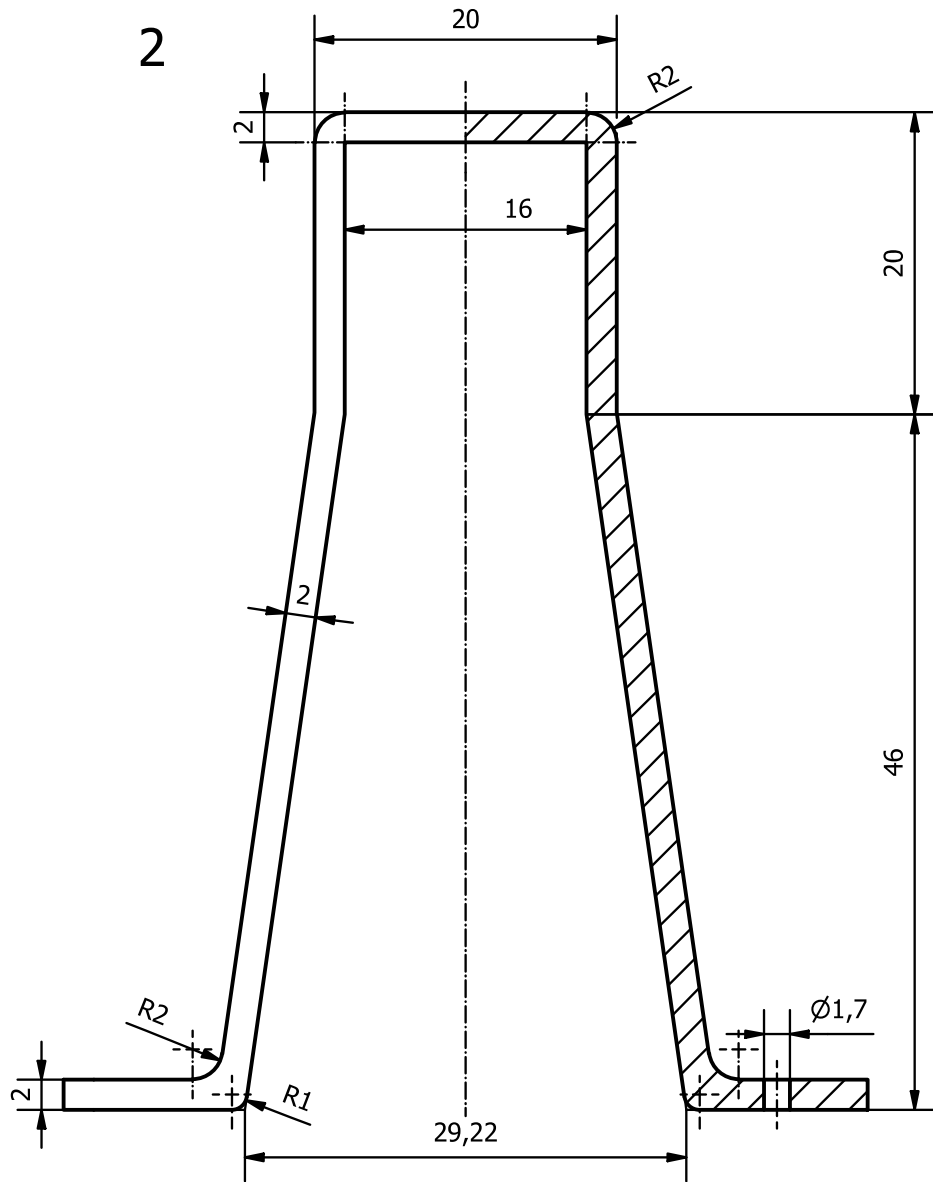
| | |
|----------------------|---|
| P-01 Acoplador | 1 |
| P-02 Soporte | 2 |



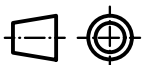
Escala: 4:1




| | | |
|--|---|----------------------------|
|  UNIVERSITAT POLITÈCNICA DE VALÈNCIA | Título: Acoplador | |
| | Tipo de documento: Dibujo de detalle | Nº identificación: P-01 |
| | Fecha edición: 06/05/22 | |
| Creado por: David Rodríguez Bascuñana | Aprobado por: David Rodríguez Bascuñana | Hoja: 1/2 |



Escala: 2:1



| | | |
|--|--|----------------------------|
|  UNIVERSITAT POLITÈCNICA DE VALÈNCIA | Título: Soporte | |
| | Tipo de documento: Dibujo de detalle | Nº identificación: P-02 |
| | Fecha edición: 06/05/22 | |
| Creado por: David Rodríguez Bascuñana | Aprobado por: David Rodríguez Bascuñana | Hoja: 2/2 |



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Diseño de un sistema de control de inversor para máquina de corriente continua sin escobillas de 200W

ANEXOS

Trabajo Final de Grado

DAVID RODRÍGUEZ BASCUÑANA

Junio, 2022

ÍNDICE DE CONTENIDOS

| | | |
|-----|---|---|
| 1. | CÓDIGO Y SISTEMAS MATLAB Y SIMULINK | 1 |
| 1.1 | Código en Matlab | 1 |
| 1.2 | Diseño en Simulink | 2 |
| 2. | CÓDIGO ARDUINO IDE..... | 6 |
| 3. | APLICACIÓN ANDROID..... | 6 |
| 3.1 | Código fuente..... | 6 |
| 3.2 | Manual de usuario..... | 6 |

ÍNDICE DE FIGURAS

| | | |
|----------------|--|---|
| Figura Anexo 1 | Subsistema lógica de conmutación | 2 |
| Figura Anexo 2 | Subsistema Realimentación de corriente..... | 3 |
| Figura Anexo 3 | Contenido de la tabla de la verdad en Simulink..... | 4 |
| Figura Anexo 4 | Modelo completo Simulink | 5 |
| Figura Anexo 5 | Inicio de la aplicación Android | 7 |
| Figura Anexo 6 | Aplicación Android tras conexión con microcontrolador..... | 8 |

ÍNDICE DE CÓDIGOS

| | | |
|----------------|--|---|
| Código Anexo 1 | Procesado de los ensayos en vacío | 1 |
| Código Anexo 2 | Representación de las transformadas de Fourier con salto de posición | 2 |

ÍNDICE DE TABLAS

| | | |
|---------------|---|---|
| Tabla Anexo 1 | Descripción de los elementos de la interfaz gráfica | 8 |
|---------------|---|---|

1. CÓDIGO Y SISTEMAS MATLAB Y SIMULINK

En este primer apartado del anexo se muestran tanto el modelo completo empleado en Simulink con todos sus subsistemas y funciones y el código empleado en Matlab para la obtención de las gráficas y valores mostrados en el apartado 4.6.1. Ensayos previos, con los cuales se permitía caracterizar al motor para la lectura correcta de la posición.

1.1 Código en Matlab

En Matlab se han empleado dos códigos para la extracción de los datos previamente obtenidos con el osciloscopio digital. El primero de ellos simplemente prepara los datos, realiza la transformada de Fourier y almacena la señal estimada en una serie de variables según el valor del Id, variable que indica sobre que fase se ha realizado el ensayo.

```
time=(0:3905)*Tinterval+Tstart;
dA=A(2:3906)-A(1:3905); % Derivada de la señal A (salto de posición)
mfin=max(find(dA<-2)); % se localizan los flancos de bajada
mini=min(find(dA<-2));
Arec=A(mini:mfin); % se recortan las señales para dejar un periodo
Brec=B(mini:mfin);
timerec=time(mini:mfin); % se recorta el tiempo
fftB=fft(Brec); % Transformada de fourier de B
Bmax=abs(fftB(4))/max(size(Brec))*2; % Amplitud de B
Bangle=angle(fftB(4))*(180/pi); % Ángulo de B
f0=1/max(timerec); % Frecuencia
Best=Bmax*cos(2*pi*f0*3*timerec+Bangle*(pi/180)); % Estimación de B filtrada

if(Id==0)

Aaz=A;
Baz=B;
ArecAz=Arec;
BrecAz=Brec;
BestAz=Best;
timerecAz=timerec;
end

if(Id==1)

Abl=A;
Bbl=B;
ArecBl=Arec;
BrecBl=Brec;
BestBl=Best;
timerecBl=timerec;
end

if(Id==2)

Aro=A;
Bro=B;
ArecRo=Arec;
BrecRo=Brec;
BestRo=Best;
timerecRo=timerec;
end
```

Código Anexo 1 Procesado de los ensayos en vacío

Una vez se han procesado las tres fases y, por tanto, se ha almacenado la información correspondiente a todas ellas se ejecuta el segundo código. Este código simplemente se encarga de visualizar las tres señales junto con el salto de posición.

```
figure(1)
hold on
plot(timerecAz,BestAz);
plot(timerecBl,BestBl);
plot(timerecRo,BestRo);
plot(timerec,Arec);
```

Código Anexo 2 Representación de las transformadas de Fourier con salto de posición

1.2 Diseño en Simulink

En este apartado se presentará en su totalidad el modelo final empleado para la realización de las simulaciones mostradas en el capítulo 2 con todos los subsistemas y funciones que lo componen. Para empezar, se presentan los subsistemas diseñados para el modelo. El primer subsistema es el encargado de la lógica de conmutación, mediante el cual se hace llegar a las ramas del inversor las señales adecuadas dependiendo del sector del motor.

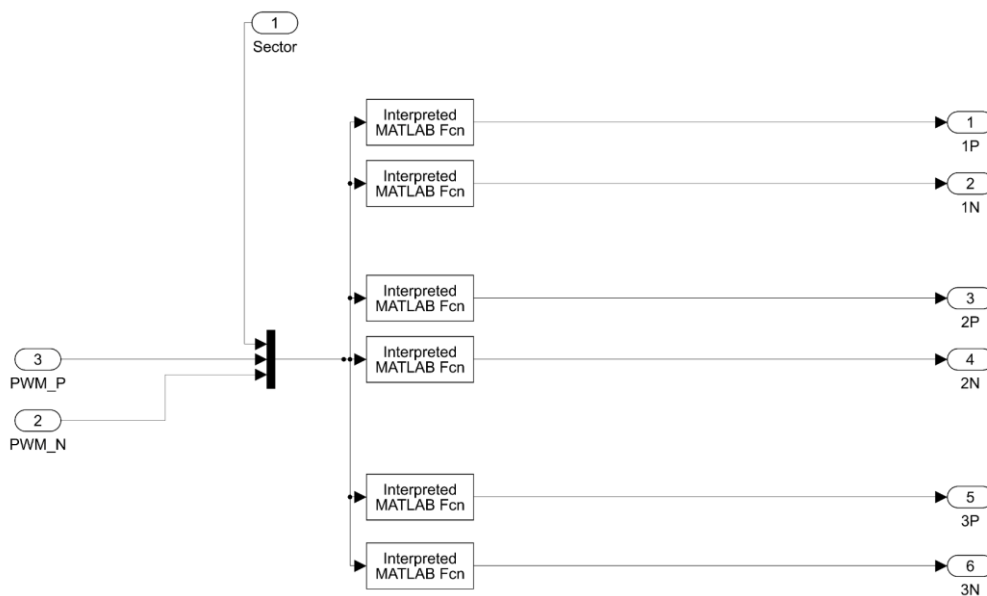


Figura Anexo 1 Subsistema lógica de conmutación

Los bloques Interpreted MATLAB Fcn permiten la inclusión de fórmulas que actúan sobre las señales de entrada para devolver una determinada señal de salida. Las funciones empleadas en cada bloque, en orden descendente desde el bloque superior son:

$$\begin{aligned} & ((u(1)>4)+(u(1)<1))*u(2)+(u(1)<4)*(u(1)>1)*u(3) \\ & ((u(1)>4)+(u(1)<1))*(1-u(2))+(u(1)<4)*(u(1)>1)*(1-u(3)) \\ & (u(1)>0)*(u(1)<3)*u(2)+(u(1)>3)*u(3) \\ & (u(1)>0)*(u(1)<3)*(1-u(2))+(u(1)>3)*(1-u(3)) \\ & (u(1)>2)*(u(1)<5)*u(2)+(u(1)<2)*u(3) \\ & (u(1)>2)*(u(1)<5)*(1-u(2))+(u(1)<2)*(1-u(3)) \end{aligned}$$

El segundo subsistema es el relativo a la realimentación de corriente. El mecanismo de realimentación se expone en la memoria de este trabajo en el apartado 2.5.8 Octavo ensayo: Realimentación de corriente.

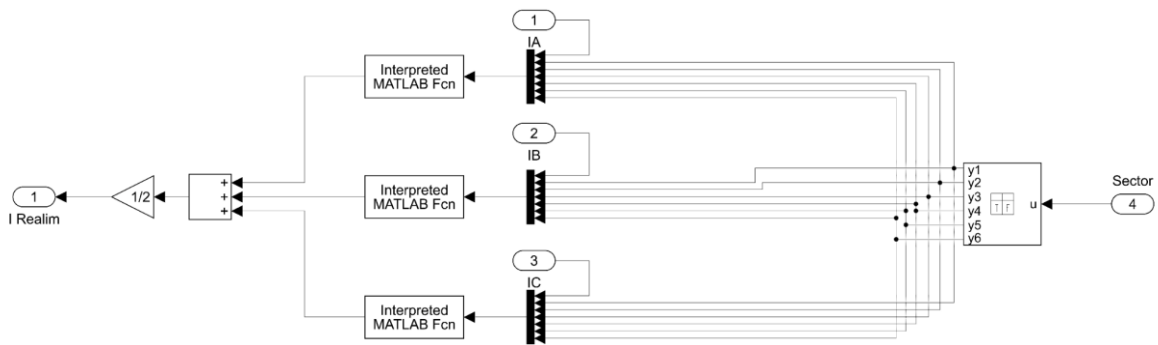


Figura Anexo 2 Subsistema Realimentación de corriente

En este subsistema se utiliza una tabla de verdad para realizar la función de multiplexor. Este recibe como señal de entrada un valor entero correspondiente al sector y devuelve seis señales binarias de forma que asigna el valor lógico uno al canal con el número correspondiente al sector.

Las fórmulas referentes a los bloques Interpreted MATLAB Fcn de este sistema son en orden descendente:

$$\begin{aligned} & u(1)*((u(2)+u(3)+u(7))-(u(4)+u(5)+u(6))) \\ & u(1)*((u(3)+u(4)+u(5))-(u(2)+u(6)+u(7))) \\ & u(1)*((u(5)+u(6)+u(7))-(u(2)+u(3)+u(4))) \end{aligned}$$

La codificación de la tabla de verdad se puede ver a continuación:

| Condition Table: | | | | | | | |
|-------------------------|-----------|---|---|---|---|---|---|
| 1: Description | Condition | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 Sector 1 | u == 0 | T | F | F | F | F | - |
| 2 Sector 2 | u == 1 | F | T | F | F | F | - |
| 3 Sector 3 | u == 2 | F | F | T | F | F | - |
| 4 Sector 4 | u == 3 | F | F | F | T | F | - |
| 5 Sector 5 | u == 4 | F | F | F | F | T | - |
| 6 Sector 6 | u == 5 | F | F | F | F | T | - |
| 7 Condición por defecto | false | - | - | - | - | - | - |
| | Actions | 1 | 2 | 3 | 4 | 5 | 6 |

| Action Table: | |
|----------------|--|
| 1: Description | Action |
| 1 | y1 = 1; y2=0; y3=0; y4=0; y5=0; y6=0; |
| 2 | y1 = 0; y2=1; y3=0; y4=0; y5=0; y6=0; |
| 3 | y1 = 0; y2=0; y3=1; y4=0; y5=0; y6=0; |
| 4 | y1 = 0; y2=0; y3=0; y4=1; y5=0; y6=0; |
| 5 | y1 = 0; y2=0; y3=0; y4=0; y5=1; y6=0; |
| 6 | y1 = 0; y2=0; y3=0; y4=0; y5=0; y6=1; |
| 7 | y1 = 0; y2=0; y3=0; y4=0; y5=0; y6=0; |

Figura Anexo 3 Contenido de la tabla de la verdad en Simulink

Por último, en relación al modelo en Simulink, se presenta en la página siguiente una imagen del modelo completo final. Este modelo se corresponde al empleado en el último ensayo del capítulo 2. Se trata de la simulación final únicamente con las funcionalidades que debe presentar el modelo real.

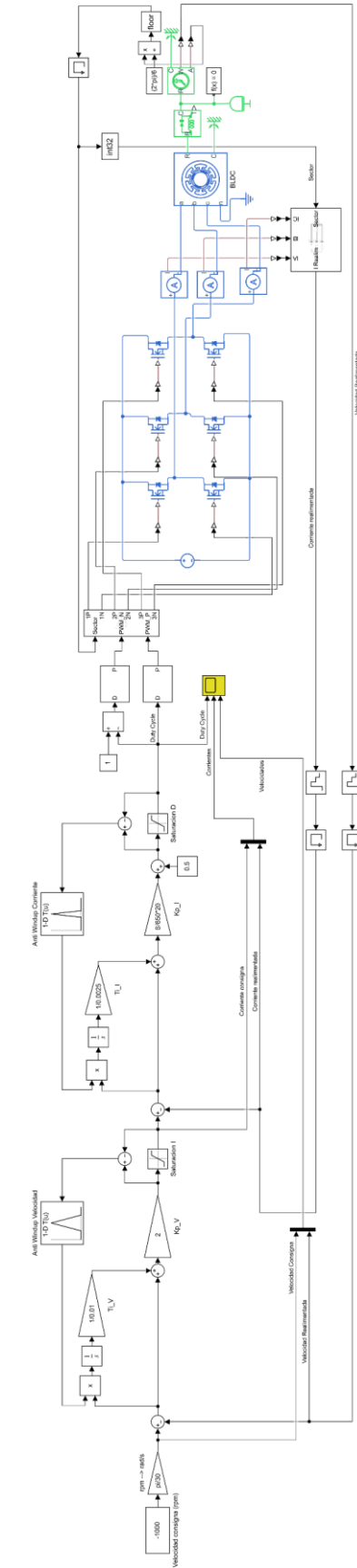


Figura Anexo 4 Modelo completo Simulink

2. CÓDIGO ARDUINO IDE

El código completo empleado para el control del sistema mediante el microcontrolador ESP32 se puede encontrar en un repositorio online disponible desde el enlace: https://github.com/Darodbas/Control_BLDC/. El archivo referente al código en Arduino IDE tiene el nombre "Código_Microcontrolador.ino".

3. APLICACIÓN ANDROID

En este apartado se muestra el código completo de toda la aplicación y un manual de usuario explicativo de la App.

3.1 Código fuente

Todo el contenido y código de la aplicación se encuentra en el mismo repositorio online comentado anteriormente. Este puede consultarse en el enlace https://github.com/Darodbas/Control_BLDC/. En el repositorio pueden encontrarse todos los archivos necesarios para la ejecución del programa en Android Studio. Además, se ha publicado bajo la licencia GNU General Public License v3.0. La cual permite la copia, modificación y distribución libre del mismo.

3.2 Manual de usuario

A continuación, se detalla el modo de uso de la aplicación Control BLDC.

- Descarga de la aplicación

La aplicación puede descargarse directamente desde el repositorio anteriormente nombrado. Sin embargo, antes de proceder a la instalación, se han de tener en cuenta varias consideraciones. Al ser una aplicación fuera de la tienda oficial de Android se deberá permitir al dispositivo la instalación de aplicaciones desde orígenes desconocidos. Esta opción se puede encontrar en los ajustes del dispositivo, normalmente desde el apartado de seguridad, aunque depende de cada fabricante. Una vez habilitada la opción, se procede a instalar la app desde el archivo ejecutable con extensión .apk. Puesto que el dispositivo no reconocerá al editor de la aplicación es probable que muestre un mensaje de advertencia. En algunos casos se deberá aceptar explícitamente la instalación de la app.

- Permisos de la aplicación

Una vez instalada, antes de abrirla, se deberá habilitar el permiso de dispositivos cercanos para la comunicación Bluetooth. Esto se realiza desde los propios ajustes de la aplicación, en el apartado permisos. Este es el único permiso necesario para la ejecución de la app. Aparte, si el Bluetooth del

dispositivo esta desactivado, la aplicación pedirá permiso al usuario para activarlo. Y, si el dispositivo móvil no es compatible con Bluetooth el programa avisará de ello.

- Funcionamiento

Con todos los pasos anteriores realizados se puede abrir ya la aplicación. Lo primero que aparecerá será una pantalla con el siguiente aspecto

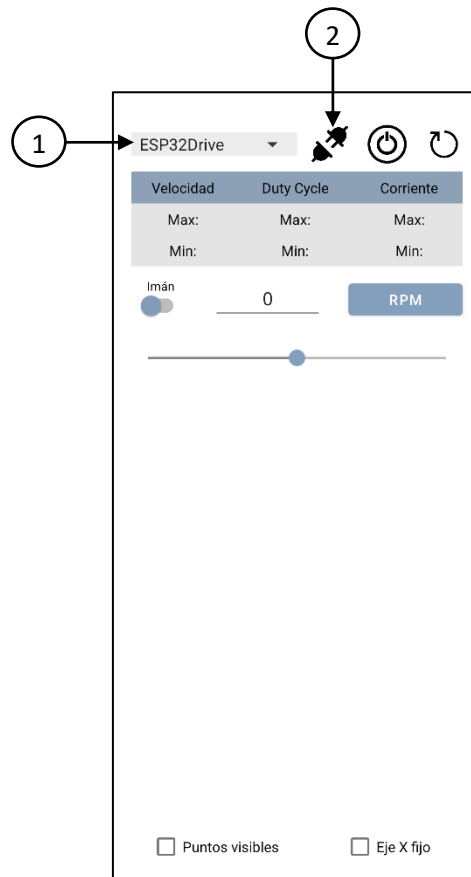


Figura Anexo 5 Inicio de la aplicación Android

Lo primero que se deberá hacer será seleccionar en el desplegable (número 1) el dispositivo que actúe como microcontrolador. En este caso, según se muestra en el código en Arduino, corresponde a “ESP32Drive”. Se ha de tener en cuenta que este desplegable solo muestra los dispositivos emparejados, por tanto, previamente, desde los ajustes de bluetooth del dispositivo móvil se deberá haber realizado el emparejamiento con el microcontrolador. Una vez seleccionado, para crear un canal de comunicación se pulsará el botón de conexión (número 2). La vinculación puede tardar unos segundos. Si se ha establecido correctamente el canal de comunicación Bluetooth el icono del botón de conexión cambiará y se mostrarán las gráficas. Si no se ha conseguido establecer la comunicación se mostrará un mensaje de error. Una vez estén conectados el dispositivo móvil con el microcontrolador, empezará la transmisión de información y la aplicación tendrá el aspecto visible en la Figura Anexo 6. Para cambiar la consigna se puede emplear una casilla editable o una barra deslizante. Para el envío de la consigna se utiliza el botón “rpm”. Para entender el funcionamiento de

la aplicación se muestra una descripción de cada elemento presente en la interfaz en la Tabla Anexo 1, la cual puede verse también en la memoria como Tabla 6.1.

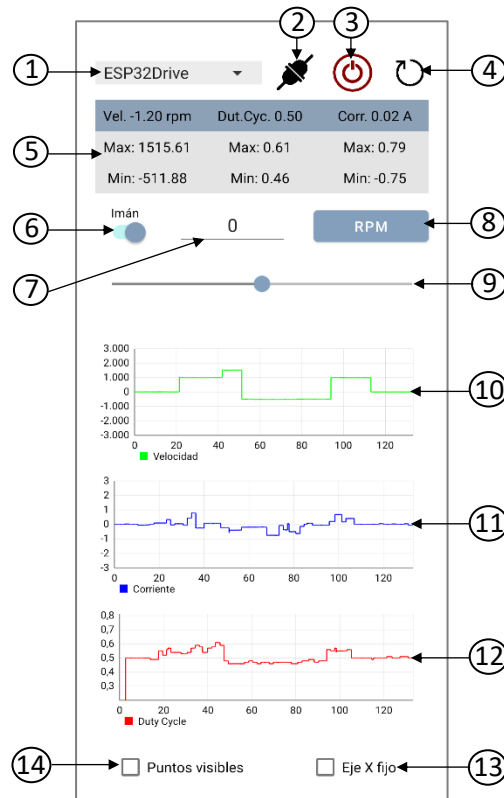


Figura Anexo 6 Aplicación Android tras conexión con microcontrolador

| Número | Nombre | Tipo | Función |
|--------|------------------------|----------------------|--|
| 1 | spDispositivos | Spinner | Permite seleccionar el dispositivo con el que conectar entre los emparejados |
| 2 | ivConectar | ImageView | Botón que permite conectar con el dispositivo seleccionado y crear un canal de comunicación |
| 3 | ivEncender | ImageView | Botón para encender y apagar el inversor y poner en marcha el control |
| 4 | ivRestart | ImageView | Botón que restablece los máximos y mínimos y limpia las gráficas |
| 5 | txVel, txCorr, txDc... | Conjunto de TextView | Tabla que muestra los valores de velocidad, corriente y Duty cycle con sus máximos y mínimos |
| 6 | swlman | Switch | Interruptor que, activado, limita el movimiento de la barra deslizando (elemento 9) a múltiplos de 100 |
| 7 | etVelConsig | EditText | Cuadro de texto editable para introducir manualmente el valor de la consigna de velocidad |
| 8 | btEnvio | Button | botón para enviar la consigna de velocidad mostrada en el elemento número 7 |
| 9 | sbBarraVel | SeekBar | Barra deslizando que permite establecer el valor de la consigna de velocidad entre -3000 y 3000 rpm |
| 10 | graficaVel | LineChart | Gráfica que muestra en tiempo real los valores recibidos de velocidad |
| 11 | graficaCorr | LineChart | Gráfica que muestra en tiempo real los valores recibidos de corriente |
| 12 | graficaDC | LineChart | Gráfica que muestra en tiempo real los valores recibidos de duty cycle |
| 13 | chEjeFijo | CheckBox | Casilla de verificación que inhabilita el autoajuste del eje x y lo limita a 10 segundos |
| 14 | chPuntos | CheckBox | Casilla de verificación que permite mostrar los puntos concretos del envío en las gráficas |

Tabla Anexo 1 Descripción de los elementos de la interfaz gráfica