



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

TRABAJO FIN DE MÁSTER EN INGENIERÍA INDUSTRIAL

**PROYECTO DIGITALIZACIÓN DE UN
PUESTO DE PRÁCTICAS DE
LABORATORIO PARA ANÁLISIS DE
LA EFICIENCIA DE MOTORES Y SU
INTEGRACIÓN EN LA INDUSTRIA 4.0**

AUTOR: ANDREU CASTILLO PENADÉS

TUTOR: ÁNGEL SAPENA BAÑÓ

COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21

AGRADECIMIENTOS

Després de sis anys d'estudi, aquest projecte suposa la fi de la meua etapa com a estudiant a la UPV, per aquest motiu dedicaré unes línies com agraïment a totes aquelles persones que m'han acompanyat durant la meua etapa universitària i que m'han recolzat en la realització del present Treball Fi de Màster.

En primer lloc, agrair als meus pares, Rosa i Vicent, per haver-me recolzat durant tots els meus anys com a estudiant i, en definitiva, haver-me permés ser qui soc el dia de hui.

Al meu tutor Rubén, per l'ajuda aportada, les tutories i el temps dedicat aquests últims mesos, oferint consells i resolent dubtes que han donat lloc a l'elaboració del present projecte.

A Anna, pel seu suport i haver-me acompanyat tots aquests anys.

I als meus companys, per haver fet més agradable els anys d'estudi.

RESUMEN

El presente Trabajo de Fin de Máster aborda la digitalización de un banco de ensayos de accionamientos eléctricos disponible en el laboratorio del Departamento de Ingeniería Eléctrica de la UPV, donde se ha desarrollado íntegramente el proyecto. La citada digitalización consta de dos partes fundamentales: el diseño de un modelo digital del banco de ensayos y el diseño y desarrollo de un nuevo sistema basado en dispositivos de bajo coste que permita regular y monitorizar el puesto de prácticas de forma remota.

Por una parte, se ha desarrollado en Matlab/Simulink un modelo digital del puesto de prácticas compuesto por el motor de inducción y el variador de frecuencia que lo alimenta, integrando en este último las diferentes estrategias de control que se pueden implementar en el sistema real, es decir, el control escalar, vectorial y directo de par. El modelo digital del motor de inducción se implementará mediante el sistema de ecuaciones diferenciales que lo define, permitiendo su adaptabilidad a otros lenguajes de programación.

En cuanto a la parte experimental del proyecto, se ha desarrollado un sistema de bajo coste basado en un microcontrolador ESP32 que permite comandar los variadores de frecuencia del motor de inducción y de su carga resistente, un motor de imanes permanentes que actuará como generador. Además, el producto diseñado permite monitorizar diferentes magnitudes eléctricas y mecánicas del motor de inducción, tales como la velocidad, el par mecánico y el nivel de corriente consumida por el motor.

La regulación y monitorización del banco de ensayos se efectúa a través de una aplicación para dispositivos móviles, diseñada en Android Studio, que permite modificar las consignas de velocidad o frecuencia y par de los variadores de frecuencia. A través de los protocolos de comunicación *bluetooth* y MQTT se monitorizan las diferentes magnitudes asociadas al banco de ensayos.

Finalmente, la corriente consumida por el motor de inducción tiene su principal aplicabilidad en el análisis de la eficiencia de motores eléctricos, permitiendo comparar el nivel de corriente consumida por diferentes motores frente a las mismas condiciones de operación. Se realizarán un conjunto de ensayos alimentando diferentes motores directamente a red y a través del variador de frecuencia, realizando una comparativa del nivel de corriente consumida.

Palabras clave: Simulación, modelo digital, motor de inducción, control escalar, control vectorial, control directo de par, microcontrolador ESP32, Matlab, Simulink, Arduino IDE, Android Studio, sensores, Industria 4.0, Internet de las Cosas, accionamientos electromecánicos.

RESUM

El present Treball de Fi de Màster té el principal objectiu de digitalitzar un banc d'assajos d'accionaments elèctrics disponible en el laboratori del Departament d'Enginyeria Elèctrica de la UPV, on s'ha desenvolupat íntegrament el projecte. La citada digitalització consta de dos parts fonamentals: el disseny d'un model digital del banc d'assajos y el disseny y desenvolupament d'un nou sistema basat en dispositius de baix cost que permeta regular y monitoritzar el banc d'assajos.

D'una banda, s'ha desenvolupat en Matlab/Simulink un model digital dels diferents elements que componen el banc d'assajos, el qual està compost pel motor d'inducció i el variador de freqüència que l'alimenta, integrant en aquest últim les diferents estratègies de control que es poden implementar en el sistema real, és a dir, el control escalar, vectorial i directe de parell. El model digital del motor d'inducció s'implementarà mitjançant el sistema d'equacions diferencials que el defineix, permetent la seua adaptació a altres llenguatges de programació.

Quant a la part experimental del projecte, s'ha desenvolupat un sistema de baix cost basat en un microcontrolador ESP32 que permet comandar els variadors de freqüència del motor d'inducció i de la seua càrrega resistent, un motor d'imants permanents que actuarà com a generador. A més, el producte dissenyat permet monitoritzar diferents magnituds elèctriques i mecàniques del motor d'inducció, com ara la velocitat, el parell mecànic i el nivell de corrent consumit pel motor.

La regulació i monitorització del banc d'assajos s'efectua a través d'una aplicació per a dispositius mòbils, dissenyada en Android Studio, que permet modificar les consignes de velocitat o freqüència i parell dels variadors de freqüència. A través dels protocols de comunicació *bluetooth* i MQTT es monitoritzen les diferents magnituds associades al banc d'assajos.

Finalment, el corrent consumit pel motor d'inducció té la seua principal aplicabilitat en l'anàlisi de l'eficiència de motors elèctrics, permetent comparar el nivell de corrent consumit per diferents motors enfront de les mateixes condicions d'operació. Es realitzaran un conjunt d'assajos alimentant diferents motors directament a la xarxa elèctrica i a través del variador de freqüència, realitzant una comparativa del nivell de corrent consumit.

Paraules clau: Simulació, model digital, motor d'inducció, control escalar, control vectorial, control directe de parell, microcontrolador ESP32, Matlab, Simulink, Arduino IDE, Android Studio, sensors, Indústria 4.0, Internet de les Coses, accionaments electromecànics.

ABSTRACT

This Master's Thesis deals with the digitization of electric drive test bench available in the laboratory of the Department of Electrical Engineering of the UPV, where the project has been developed in its entirety. This digitization consists of two fundamental parts: the design of a digital model of the test bench and the design and development of a new system based on low-cost devices that allow remote regulation and monitoring of the test bench.

On the one hand, a digital model of the training station has been developed in Matlab/Simulink, consisting of the induction motor and the frequency inverter that feeds it, integrating in the latter the different control strategies that can be implemented in the real system, i.e., scalar, vector and direct torque control. The digital model of the induction motor will be implemented by means of the system of differential equations that defines it, allowing its adaptability to other programming languages.

As for the experimental part of the project, a low-cost system based on an ESP32 microcontroller has been developed to control the frequency variators of the induction motor and its resistant load, a permanent magnet motor that will act as a generator. In addition, the designed product allows the monitoring of different electrical and mechanical magnitudes of the induction motor, such as speed, mechanical torque and the level of current consumed by the motor.

The regulation and monitoring of the test bench are carried out through an application for mobile devices, designed in Android Studio, which allows the speed or frequency and torque setpoints of the frequency inverters to be modified. The different magnitudes associated with the test bench are monitored via Bluetooth and MQTT communication protocols.

Finally, the current consumed by the induction motor has its main applicability in the analysis of the efficiency of electric motors, making it possible to compare the level of current consumed by different motors under the same operating conditions. A series of tests will be carried out feeding the motors directly to the mains and through the frequency variator, comparing the level of current consumed.

Keywords: Simulation, digital model, induction motor, scalar control, vector control, direct torque control, ESP32 microcontroller, Matlab, Simulink, Arduino IDE, Android Studio, sensors, industry 4.0, Internet of Things, electromechanical drives.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

**PROYECTO DIGITALIZACIÓN DE UN PUESTO DE
PRÁCTICAS DE LABORATORIO PARA ANÁLISIS DE LA
EFICIENCIA DE MOTORES Y SU INTEGRACIÓN EN LA
INDUSTRIA 4.0**

ÍNDICE GENERAL DE DOCUMENTOS

MEMORIA
PRESUPUESTO
PLANOS
ANEXOS

AUTOR: ANDREU CASTILLO PENADÉS
TUTOR: ÁNGEL SAPENA BAÑÓ
COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

**PROYECTO DIGITALIZACIÓN DE UN PUESTO DE
PRÁCTICAS DE LABORATORIO PARA ANÁLISIS DE LA
EFICIENCIA DE MOTORES Y SU INTEGRACIÓN EN LA
INDUSTRIA 4.0**

MEMORIA

AUTOR: ANDREU CASTILLO PENADÉS

TUTOR: ÁNGEL SAPENA BAÑÓ

COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21

ÍNDICE DE CONTENIDOS

1	CAPÍTULO 1: INTRODUCCIÓN	1
1.1	ANTECEDENTES	1
1.2	DESCRIPCIÓN DEL PROYECTO Y OBJETIVOS	3
1.3	ESTRUCTURA DEL DOCUMENTO	7
2	CAPÍTULO 2: MODELADO Y SIMULACIÓN DEL MOTOR DE INDUCCIÓN	9
2.1	PRINCIPIO DE FUNCIONAMIENTO	9
2.1.1	CIRCUITO ELÉCTRICO EQUIVALENTE	10
2.1.2	CURVA PAR - VELOCIDAD	11
2.2	MODELO DEL MOTOR DE INDUCCIÓN	12
2.3	OBTENCIÓN DE PARÁMETROS	17
2.3.1	ENSAYO DE CONTINUA.....	17
2.3.2	ENSAYO DE CORTOCIRCUITO	17
2.3.3	ENSAYO DE VACÍO	18
2.4	VALIDACIÓN DEL MODELO.....	19
2.5	IMPLEMENTACIÓN DEL MODELO DEL ACIM EN MICROCONTROLADOR.....	20
2.5.1	SELECCIÓN DEL INTERVALO DE PASO Y ANÁLISIS DE ESTABILIDAD.....	21
2.6	RESOLUCIÓN DEL MODELO EN PYTHON	22
3	CAPÍTULO 3: MODELADO Y SIMULACIÓN DEL VARIADOR DE FRECUENCIA	24
3.1	INTRODUCCIÓN A LAS ESTRATEGIAS DE CONTROL	25
3.2	VARIADOR DE FRECUENCIA.....	26
3.2.1	INVERSOR TRIFÁSICO	27
3.2.2	SISTEMA DE CONTROL.....	34
3.3	CONTROL ESCALAR.....	34
3.4	CONTROL VECTORIAL.....	39
3.4.1	CONTROL VECTORIAL DIRECTO	41
3.4.2	CONTROL VECTORIAL INDIRECTO	41
3.5	CONTROL DIRECTO DE PAR	43
3.6	INTEGRACIÓN DE LOS MODELOS EN APP DESIGNER	46
4	CAPÍTULO 4: SENSORIZACIÓN E INSTRUMENTACIÓN ELECTRÓNICA	49
4.1	SENSORES DE POSICIÓN Y VELOCIDAD	50
4.1.1	ENCODER MAGNÉTICO.....	50
4.1.2	ENCODER ÓPTICO.....	52
4.1.3	SELECCIÓN DEL SENSOR DE POSICIÓN Y VELOCIDAD	55
4.2	SENSORES DE CORRIENTE	56

4.2.1	RESISTENCIA SERIE	56
4.2.2	TRANSFORMADOR DE CORRIENTE.....	56
4.2.3	SENSORES DE EFECTO HALL	58
4.2.4	SELECCIÓN DEL SENSOR DE CORRIENTE	59
4.3	SENSORES DE PAR	62
4.3.1	MÉTODO DIRECTO.....	62
4.3.2	MÉTODO INDIRECTO	63
4.3.3	SELECCIÓN DEL SENSOR DE PAR	63
4.4	ADAPTACIÓN DE LAS ENTRADAS DEL VARIADOR DE FRECUENCIA.....	65
4.5	ALIMENTACIÓN DE LOS COMPONENTES ELECTRÓNICOS.....	66
4.6	DISEÑO DEL PROTOTIPO	68
5	CAPÍTULO 5: PROGRAMACIÓN DEL MICROCONTROLADOR ESP32	70
5.1	PLACA DE DESARROLLO.....	70
5.2	ENTORNO DE DESARROLLO ARDUINO IDE.....	72
5.3	FILTRADO DIGITAL DE SEÑALES.....	73
5.3.1	FILTROS DE MEDIA Y MEDIANA MÓVIL.....	74
5.3.2	FILTRO HAMPEL.....	75
5.3.3	FILTRO BUTTERWORTH	76
5.4	GESTIÓN DE MÚLTIPLES TAREAS.....	78
5.5	ADQUISICIÓN DE DATOS	80
5.5.1	LECTURA DE POSICIÓN Y VELOCIDAD	81
5.5.2	LECTURA DE PAR	82
5.5.3	LECTURA DE CORRIENTES DE ESTATOR.....	84
5.6	TRANSMISIÓN DE DATOS MEDIANTE BLUETOOTH.....	87
5.6.1	ENVÍO DE UN REGISTRO DE DATOS.....	88
5.6.2	CONTROL DE LOS VARIADORES DE FRECUENCIA.....	88
5.6.3	CONFIGURACIÓN DE UN BANCO DE ENSAYOS	89
5.7	TRANSMISIÓN DE DATOS MEDIANTE MQTT.....	91
5.7.1	BROKER MQTT	91
5.7.2	CREACIÓN DE TOPICS	92
5.8	VISUALIZACIÓN LOCAL EN LA PANTALLA LCD	92
5.8.1	SISTEMA DE COORDENADAS Y DEFINICIÓN DE COLORES.....	92
5.8.2	DISEÑO DE LA INTERFAZ LOCAL DE VISUALIZACIÓN	94
6	CAPÍTULO 6: DISEÑO Y DESARROLLO DE LA APLICACIÓN ANDROID	97
6.1	INTRODUCCIÓN A ANDROID STUDIO	97
6.1.1	CREACIÓN DE INTERFACES GRÁFICAS	99

6.1.2	GENERACIÓN DE CÓDIGO	100
6.2	INTERFAZ DE USUARIO	101
6.2.1	INTERFAZ REGISTRO DE DATOS MQTT	101
6.2.2	INTERFAZ REGISTRO DE DATOS BLUETOOTH	101
6.2.3	INTERFAZ CONFIGURACIÓN DE PARÁMETROS	104
7	CAPÍTULO 7: RESULTADOS	106
7.1	ENSAYOS EXPERIMENTALES	106
7.1.1	CONEXIÓN DIRECTA A RED	109
7.1.2	CONEXIÓN MEDIANTE VARIADOR DE FRECUENCIA	110
7.2	ANÁLISIS DE EFICIENCIA	111
7.3	ANÁLISIS FILTRADO DIGITAL	112
7.4	VISUALIZACIÓN LOCAL EN LA PANTALLA LCD	114
7.5	COMPARATIVA MODELO DIGITAL Y SISTEMA REAL	115
8	CAPÍTULO 8: CONCLUSIONES Y LÍNEAS FUTURAS	116
9	BIBLIOGRAFÍA	118

ÍNDICE DE FIGURAS

Figura 1.1.	Banco de ensayos	3
Figura 1.2.	Conjunto de motores acoplados	4
Figura 1.3.	Caja externa de conexiones	4
Figura 1.4.	Variadores de frecuencia disponibles en el banco de ensayos	5
Figura 1.5.	Producto diseñado	6
Figura 2.1.	Circuito eléctrico equivalente	11
Figura 2.2.	Circuito eléctrico equivalente simplificado	11
Figura 2.3.	Curva par-velocidad [8]	12
Figura 2.4.	Motor de inducción en bucle abierto	13
Figura 2.5.	Cómputo de flujos de estator y rotor	13
Figura 2.6.	Selección de ecuaciones para el cómputo de las corrientes	14
Figura 2.7.	Cómputo de corrientes de estator y rotor - $R_{Fe} = \infty$	14
Figura 2.8.	Cómputo de corrientes de estator y rotor - $R_{Fe} < \infty$	15
Figura 2.9.	Ecuaciones mecánicas	16
Figura 2.10.	Esquema general del motor de inducción	16
Figura 2.11.	Fuente de tensión controlada	16
Figura 2.12.	Validación del modelo digital	19
Figura 2.13.	Resultados del modelo digital	20
Figura 2.14.	Validación de la resolución numérica del modelo	21
Figura 2.15.	Comparativa entre la resolución propuesta y ode45	22
Figura 2.16.	Análisis de estabilidad	22
Figura 2.17.	Resultados simulación en Python	23
Figura 3.1.	Modelo digital del motor de inducción y variador de frecuencia	24
Figura 3.2.	Conexión directa a red o variador de frecuencia	24

Figura 3.3. Modelo del variador de frecuencia.	26
Figura 3.4. Inversor trifásico [11].	27
Figura 3.5. Distribución espacial de los vectores en SVPWM [7].	28
Figura 3.6. Obtención del máximo voltaje de salida en SVPWM [7].	30
Figura 3.7. Patrones de conmutación SVPWM [7].	31
Figura 3.8. Implementación de la técnica SVPWM.	32
Figura 3.9. Validación de la técnica SVPWM.	32
Figura 3.10. Resultados motor en bucle abierto.	33
Figura 3.11. Selección del patrón de conmutación.	33
Figura 3.12. Selección de la estrategia de control.	34
Figura 3.13. Curva V-f [13].	35
Figura 3.14. Esquema general del control escalar [13].	36
Figura 3.15. Control escalar.	36
Figura 3.16. Efecto antiwindup.	37
Figura 3.17. Resultados control escalar.	38
Figura 3.18. Regulador de corrientes [14].	40
Figura 3.19. Cómputo de $i_{sd, ref}$	40
Figura 3.20. Cómputo de $wslip$ y θ	42
Figura 3.21. Cómputo de la magnitud y ángulo de la tensión de referencia.	42
Figura 3.22. Control vectorial indirecto.	42
Figura 3.23. Resultados control vectorial.	43
Figura 3.24. Esquema general del control DTC [7].	44
Figura 3.25. Estimador de flujo y par.	44
Figura 3.26. Cómputo de ψ_s, ref	44
Figura 3.27. Esquema control directo de par.	45
Figura 3.28. Reguladores de histéresis.	45
Figura 3.29. Patrón de conmutación - DTC.	46
Figura 3.30. Resultados control directo de par.	46
Figura 3.31. Interfaz App Designer - Configuración de parámetros.	47
Figura 3.32. Interfaz App Designer - Visualización de resultados.	47
Figura 4.1. Efecto Hall.	50
Figura 4.2. Señales producidas por un encoder magnético absoluto.	51
Figura 4.3. Transmisión de datos - Encoder magnético absoluto [22].	51
Figura 4.4. Encoder magnético incremental.	52
Figura 4.5. Codificación encoder óptico absoluto.	53
Figura 4.6. Encoder óptico incremental.	54
Figura 4.7. Encoder óptico incremental 1XP8001-1 de Siemens [23].	55
Figura 4.8. Divisor de tensión.	55
Figura 4.9. Transformador de corriente [25].	56
Figura 4.10. Principio de funcionamiento de un transformador de corriente [26].	57
Figura 4.11. Modificación de la relación de transformación [26].	57
Figura 4.12. Tipos de sensores de efecto Hall.	58
Figura 4.13. Hoja de características - Sensor de efecto Hall [29].	59
Figura 4.14. Configuración del sensor de efecto Hall [29].	61
Figura 4.15. Relación tensión de salida y corriente primaria [29].	62
Figura 4.16. Transductores de par [30].	63
Figura 4.17. Módulo de 8 relés.	65
Figura 4.18. Amplificador operacional OPA705 [31].	65

Figura 4.19. Esquema amplificador operacional.....	65
Figura 4.20. Fuente de tensión continua.	66
Figura 4.21. Divisor de tensión de dos niveles.....	67
Figura 4.22. Implementación del circuito electrónico.	68
Figura 4.23. Montaje final.	69
Figura 4.24. Producto final.	69
Figura 5.1. Microcontrolador ESP32.	71
Figura 5.2. Filtro analógico.	73
Figura 5.3. Efecto de la frecuencia de corte para un orden fijo - $f_s=100\text{Hz}$	77
Figura 5.4. Efecto del orden para una frecuencia de corte fija - $f_s=100\text{Hz}$	78
Figura 5.5. Diagrama de flujo general de las tareas definidas.	79
Figura 5.6. Lectura de velocidad.	82
Figura 5.7. Cálculos implicados en la lectura indirecta de par.....	82
Figura 5.8. Aplicación del filtro Hampel.	83
Figura 5.9. Lectura de par.	84
Figura 5.10. Lectura de corriente de fase mediante conexión a través de variador.	85
Figura 5.11. Lectura del módulo de la corriente.	86
Figura 5.12. Filtrado del módulo de la corriente.....	86
Figura 5.13. Lectura de la amplitud de la corriente.	87
Figura 5.14. Esquema broker MQTT.	91
Figura 5.15. Sistema de coordenadas de la pantalla LCD [37].	93
Figura 5.16. Interfaz para la visualización de magnitudes.	95
Figura 5.17. Interfaz para la visualización de parámetros de configuración.	96
Figura 6.1. Entorno de desarrollo Android Studio – Diseño de la interfaz.	98
Figura 6.2. Entorno de desarrollo Android Studio – Generación de código.	98
Figura 6.3. Comparativa programación gráfica y basada en texto.	99
Figura 6.4. Interfaz de usuario - MQTT.	101
Figura 6.5. Interfaz de usuario - bluetooth.	103
Figura 6.6. Monitorización del banco de ensayos.....	103
Figura 6.7. Configuración de parámetros del banco de ensayos.....	105
Figura 6.8. Configuración red Wifi y servidor MQTT.....	105
Figura 7.1. Tipos de conexión del motor de inducción.	107
Figura 7.2. Conexión de la placa.	108
Figura 7.3. Ensayo 2 - Conexión directa a red.....	109
Figura 7.4. Conexión de la placa.	110
Figura 7.5. Comparativa del nivel de corriente.....	111
Figura 7.6. Comparativa velocidad y par.....	112
Figura 7.7. Análisis filtrado digital - Nivel de corriente.....	113
Figura 7.8. Configuración de parámetros.....	114
Figura 7.9. Visualización local de magnitudes.....	114
Figura 7.10. Comparación de resultados de simulaciones y banco de ensayos real.	115

ÍNDICE DE TABLAS

Tabla 2.1. Resistencias estáticas.....	17
Tabla 2.2. Ensayo de cortocircuito.	17
Tabla 2.3. Ensayo de vacío.	18

Tabla 2.4. Cálculos auxiliares.....	19
Tabla 3.1. Posibles salidas del inversor trifásico.	28
Tabla 3.2. Vectores espaciales técnica SVPWM.....	29
Tabla 3.3. Patrones de conmutación – DTC.	45
Tabla 5.1. Pines reservados para la interfaz LCD.	71
Tabla 5.2. Pines de E/S empleados en el proyecto.	72
Tabla 5.3. Pines empleados en el muestreo de datos.....	80
Tabla 5.4. Modos de funcionamiento de una interrupción.	81
Tabla 5.5. Modos de apertura de archivos en la librería SPIFFS.	90
Tabla 5.6. Topics creados.	92
Tabla 5.7. Funciones empleadas para representar formas.....	93
Tabla 6.1. Elementos empleados en el proyecto de Android Studio.	100
Tabla 7.1. Placa de características de los motores ensayados.....	107
Tabla 7.2. Valores de carga aplicada en función del tiempo.	108

CAPÍTULO 1: INTRODUCCIÓN

1.1 ANTECEDENTES

Las máquinas eléctricas rotativas de corriente alterna, y en especial los motores de inducción, son ampliamente utilizadas en la industria actual. Para una misma potencia, los motores de inducción poseen un menor tamaño, peso y coste que las máquinas de continua. Además, la ausencia de escobillas, conmutadores y colectores hace posible el empleo de estos motores en cualquier entorno de trabajo.

Tal como ha ocurrido en diferentes inventos del ámbito de la ingeniería eléctrica, los primeros motores eléctricos de la historia basaban su principio de funcionamiento en la corriente continua. Los avances en la distribución y transporte de energía eléctrica generaron la necesidad de emplear redes de corriente alterna, dando al lugar al desarrollo de máquinas eléctricas trifásicas [1].

Hasta la década de los 80, el empleo de motores de corriente continua era predominante frente a los de corriente alterna en aplicaciones en las que se requería un control de velocidad y posición [2]. Esto es debido a la facilidad de implementar el control de motores de corriente continua a través del cambio en la tensión de alimentación de sus devanados.

Frente a estos, las técnicas de control de motores de inducción se basaban en la modificación del número de pares de polos alternando las conexiones de las bobinas, en la variación de la tensión de alimentación del motor o en el cambio de la frecuencia de dicha tensión [2]. Estas técnicas no sólo llevaban asociadas una complejidad constructiva elevada, sino que el rango de velocidades en el que era posible controlar la velocidad del motor era relativamente reducido.

El avance de la electrónica de potencia y el incremento en la capacidad de computación de los microcontroladores ha permitido implementar estrategias de control más precisas y eficientes, como el control escalar, ampliamente utilizado en la industria actual, el control vectorial o el control directo de par. En consecuencia, el motor de inducción fue sustituyendo al motor de corriente de continua de forma paulatina, convirtiéndose en el más utilizado en la actualidad cubriendo un amplio abanico de aplicaciones, desde bombas hidráulicas, cintas transportadoras, máquinas herramienta o sopladores, entre otras.

Debido a estos avances en los sistemas de control de motores de inducción, a lo largo de los años ha surgido la necesidad de diseñar simulaciones en las que sea posible analizar diversas estrategias de control de estos motores. El aumento de la capacidad de cálculo de los ordenadores actuales y la existencia de software que permita la simulación de procesos hacen posible el diseño de simulaciones dinámicas de motores de inducción, permitiendo validar cualquier sistema de control previamente a su implementación en el sistema real.

Además, el hecho de disponer de un modelo virtual del proceso industrial permite analizar su comportamiento frente a diferentes situaciones sin necesidad de detener el proceso productivo e incluso permite analizar el comportamiento real del proceso comparándolo con la respuesta estimada por el modelo virtual, lo que se conoce como gemelo digital [5].

Actualmente, las diferentes estrategias de control de motores eléctricos están integradas en los variadores de frecuencia. Estos dispositivos actúan como sistemas de control de la velocidad de

los motores eléctricos, permitiendo regular la velocidad mecánica del motor a través de las entradas digitales y analógicas del propio variador de frecuencia.

Mediante el empleo de estas entradas es posible modificar la consigna de frecuencia, velocidad o par del variador. En particular, las entradas digitales se pueden emplear para modificar estas consignas mediante cambios en escalón o bien para modificar parámetros del variador más genéricos tales como la selección del tipo de control aplicado. Por otra parte, las entradas analógicas se suelen emplear para modificar las consignas anteriores de una forma continua, es decir, sin la presencia de cambios bruscos.

En consecuencia, es necesario modificar el valor de las entradas digitales y analógicas del variador de frecuencia con el objetivo de modificar la consigna correspondiente. Existen diferentes alternativas que permiten controlar estas entradas, tales como el diseño de un panel físico que actúe sobre las entradas anteriores mediante el empleo de interruptores y potenciómetros o el empleo de PLCs de forma conjunta con sistemas SCADA.

La primera opción, es decir, el empleo de paneles físicos supone una alternativa de bajo coste cuya instalación y uso por parte del usuario es relativamente sencilla. Sin embargo, esta solución no integra la posibilidad de monitorizar el estado del motor de inducción mediante el registro de ciertas magnitudes tales como la velocidad, el par mecánico o la corriente consumida por el motor.

Por otra parte, la instalación de PLCs industriales y el diseño de sistemas SCADA permiten integrar de forma conjunta todas las funcionalidades del panel físico y la monitorización del estado de operación del motor [3]. El incremento en prestaciones respecto a los paneles físicos supone un notable incremento en los costes de adquisición de los diferentes productos.

Como alternativa a las dos soluciones propuestas anteriormente, resultaría de gran interés disponer de un producto de bajo coste basado, por ejemplo, en un microcontrolador que permita controlar las entradas del variador de frecuencia y, al mismo tiempo, monitorizar el estado de la máquina eléctrica.

De este modo sería posible aprovechar las características y funcionalidades derivadas del empleo de microcontroladores, como su fácil conexión con sensores y actuadores y su integración en sistemas IoT (*Internet of Things*), es decir, integrarlos en sistemas de comunicaciones inalámbricas, permitiendo controlar y monitorizar una máquina eléctrica de forma remota.

Adicionalmente, el diseño de un producto basado en microcontroladores ofrece otro tipo de ventajas. Es posible diseñar el código que integra el microcontrolador mediante software de código abierto, de modo que se tiene un fácil acceso a todo el material disponible por la comunidad de desarrolladores.

En definitiva, a nivel industrial el control de motores eléctricos trifásicos a través de variadores de frecuencia tiene un gran interés en multitud de aplicaciones, siendo necesario actuar sobre sus entradas digitales y analógicas y es en este ámbito donde el diseño y desarrollo de productos de bajo coste y basados en software libre permite obtener soluciones integrales que facilitan implementar de forma conjunta el control de variadores de frecuencia y la monitorización del estado de una máquina eléctrica de forma remota.

1.2 DESCRIPCIÓN DEL PROYECTO Y OBJETIVOS

En el presente Trabajo de Fin de Máster, elaborado en el Departamento de Ingeniería Eléctrica de la UPV y enmarcado en el programa de becas de la Fundación Elecnor, se pretende digitalizar un banco de ensayos de accionamientos eléctricos, el cual consiste en un sistema de control de velocidad y par asociado a un motor de inducción.

Dicha digitalización consta de dos partes diferenciadas: la obtención de un modelo virtual del montaje de laboratorio y la adaptación del banco de ensayos a la industria 4.0, permitiendo comandar los variadores de frecuencia del motor principal y de su carga, monitorizar un conjunto de variables asociadas al puesto de prácticas y analizar la eficiencia de diferentes motores a través de la corriente consumida por éstos, todo ello empleando una aplicación Android que realizará la función de interfaz de comunicaciones.

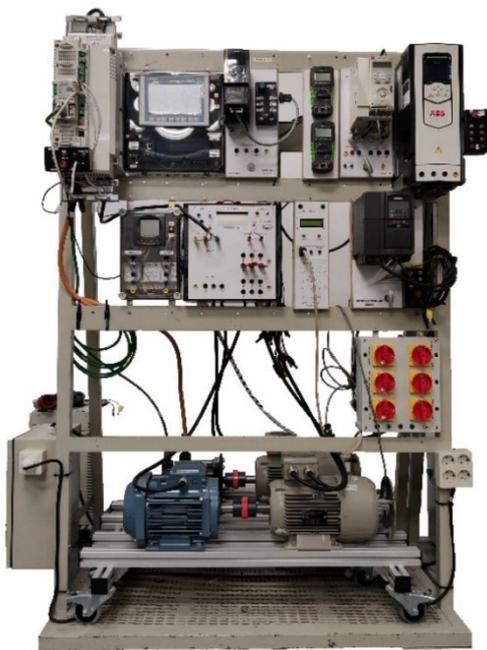


Figura 1.1. Banco de ensayos.

El banco de ensayos de accionamientos eléctricos está compuesto por una serie de elementos que permiten implementar la monitorización y control de un conjunto de máquinas eléctricas rotativas, de entre las cuales se emplearán dos motores acoplados: un motor de inducción que actuará como motor principal y un motor síncrono de imanes permanentes que realizará la función de carga resistente, actuando como generador.

En particular el puesto de prácticas dispone de una serie de variadores de frecuencia que incorporan diferentes tipos de controles de motores eléctricos: control escalar, control vectorial y control directo de par, además de poder ensayar los motores en bucle abierto o bucle cerrado mediante la realimentación de la medida de velocidad o par. Incluso mediante su conexión directa a red sin ningún tipo de control. Seguidamente, se detallan los diferentes elementos que componen el puesto de laboratorio

En cuanto a los motores acoplados, el motor de inducción es posible alimentarlo mediante diferentes métodos: mediante conexión directa a red o a través de un variador de frecuencia entre los disponibles. El servomotor, o motor de imanes permanentes, actuará como generador al estar acoplado al motor de inducción, por lo que al primero será necesario conectar una resistencia de frenado, cuya potencia de disipación se sitúa en torno 2kW, al bus de continua correspondiente con el fin de disipar en forma de calor la energía recuperada. El conjunto de motores acoplados, así como la resistencia de frenado, se exponen en la figura 1.2, donde el motor de reluctancia y el motor de inducción acoplado a este no son ámbito de este proyecto.

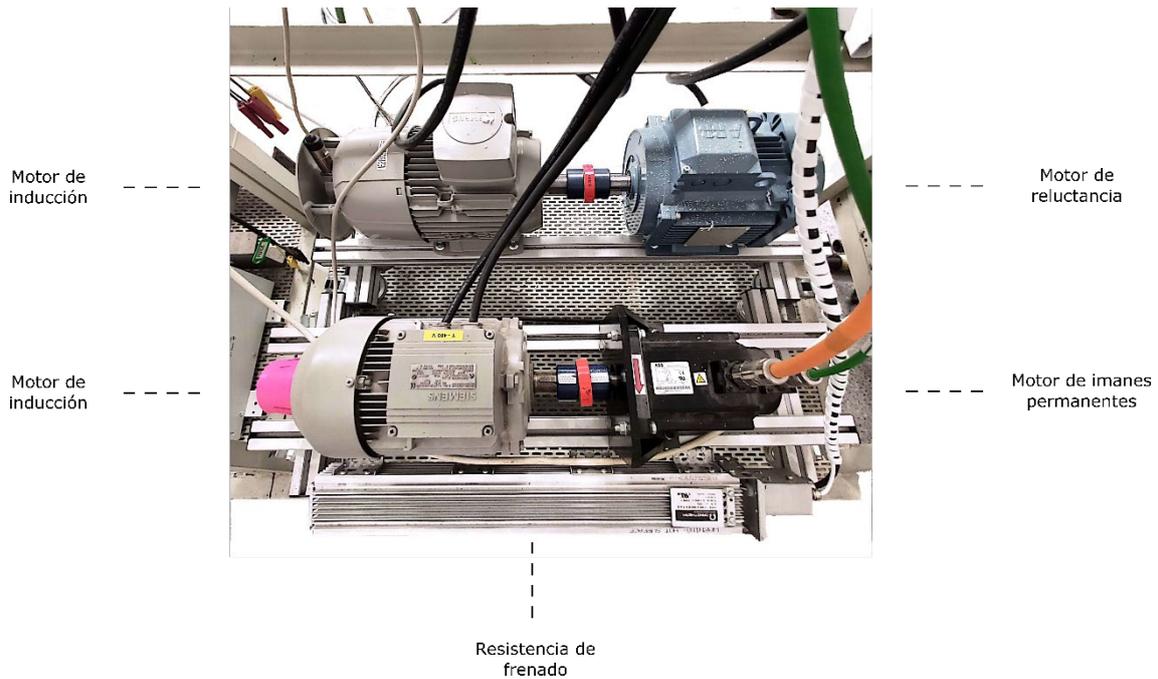


Figura 1.2. Conjunto de motores acoplados.

La conexión de los devanados del motor de inducción ya sea en estrella o triángulo, así como la conexión de la resistencia de frenado se realiza en una caja de conexiones externa, la cual se expone en la figura 1.3. Cabe destacar que, por seguridad, se dispone de un termopar conectado a la resistencia de frenado para que, en el caso de superar una temperatura umbral, se desconecte el sistema.

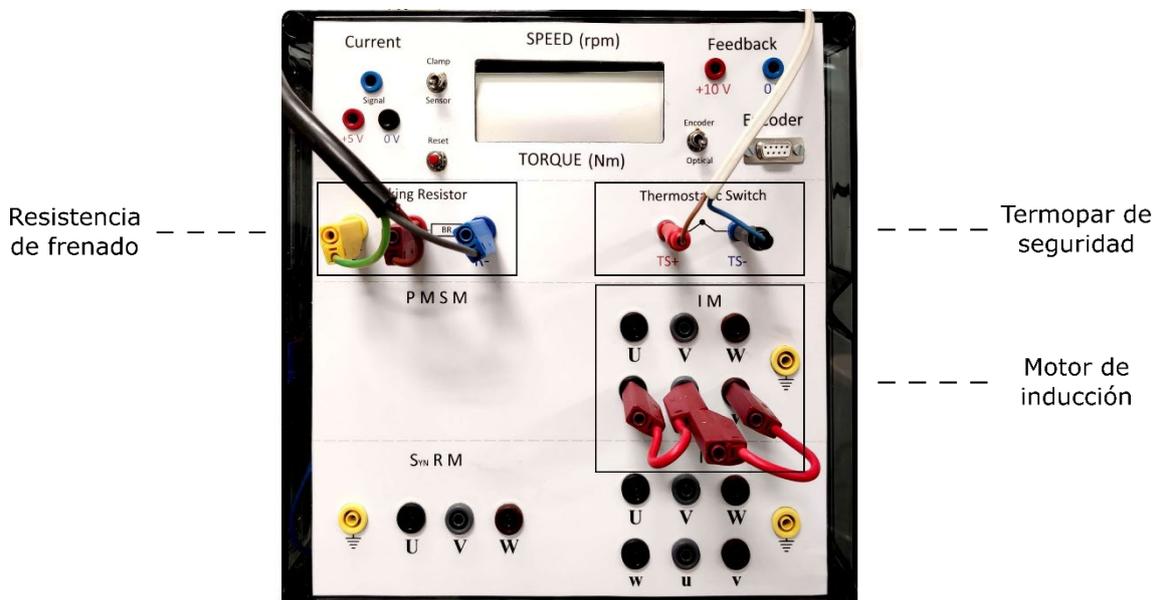


Figura 1.3. Caja externa de conexiones.

Por lo que respecta a los variadores de frecuencia del motor de inducción, el banco de ensayos dispone de diferentes tecnologías. En particular, el variador ABB ACS 150 presenta una tecnología basada en el control escalar. Tanto el variador Micromaster M 440, como el Sinamics G120 PN, ambos de Siemens, presentan una estrategia de control vectorial o de campo orientado sin necesidad de sensores. El variador ABB ACS 880 dispone de un control directo de par (DTC). Cabe resaltar que todos los variadores pueden actuar con un control escalar.

Finalmente, el variador asociado al motor de imanes permanentes es el ABB ACSM 1. El conjunto de variadores disponibles en el laboratorio se detallan en la figura 1.4.



Figura 1.4a. ABB ACSM 1.



Figura 1.4b. ABB ACS 800.



Figura 1.4c. ABB ACS 150.



Figura 1.4d. Sinamics G120 PN.



Figura 1.4e. Micromaster 440.

Figura 1.4. Variadores de frecuencia disponibles en el banco de ensayos.

Tal como se observa en las figuras anteriores, la regulación de los distintos variadores de frecuencia se puede realizar a través de una botonera mediante la cual es posible controlar las entradas digitales y analógicas de estos variadores, permitiendo poner en marcha o parar el motor correspondiente, introducir diferentes consignas discretas de frecuencia o velocidad, o modificar de forma continua la frecuencia, velocidad o par a través de un potenciómetro.

En cuanto al desarrollo del proyecto, en primer lugar, se diseñarán las simulaciones necesarias mediante Matlab/Simulink para obtener un modelo virtual del banco de ensayos presente en el laboratorio. La elaboración del modelo virtual permitirá diseñar todo el software del proyecto, es decir, el código asociado al microcontrolador y el correspondiente a la aplicación Android, sin necesidad de realizar el montaje del prototipo definitivo. Para ello, será necesario implementar el modelo virtual del motor de inducción en el microcontrolador, resolviendo el sistema de ecuaciones en tiempo real.

Con el objetivo de adaptar el puesto de laboratorio a la industria 4.0, se elaborará un sistema de comunicaciones inalámbrico que permita no sólo la visualización de magnitudes características del banco de ensayos, sino también su control a través de una aplicación para dispositivos móviles.

Se requiere visualizar la posición y velocidad angular, la corriente generada por una de las fases del motor de imanes permanentes, el par electromagnético y el nivel de corriente consumida por el motor de inducción mediante el cálculo del módulo asociado al fasor de corriente de estator y la amplitud de esta.

La adaptación del control del banco de ensayos permitirá sustituir las botoneras comentadas anteriormente y poder comandar los variadores de frecuencia a través de la aplicación Android. Como protocolos de transmisión de datos se empleará MQTT y *bluetooth*.

Por todo lo expuesto anteriormente, se requiere diseñar y desarrollar un nuevo producto que permita comandar los variadores de frecuencia del motor de inducción y el motor de imanes permanentes y poder monitorizar el estado del banco de ensayos a través de diferentes magnitudes eléctricas y mecánicas, permitiendo analizar la eficiencia de los diferentes puestos de prácticas disponibles en el laboratorio.

Dicho producto debe de poder emplearse en cualquier puesto de prácticas del laboratorio, por lo que deber se lo suficientemente versátil como para poder configurar parámetros característicos de cada banco de ensayos, tales como la resolución del encoder, número de par de polos de los motores acoplados o el factor de proporcionalidad que permite determinar el par electromagnético, entre otros El producto que se ha elaborado durante el desarrollo del presente Trabajo Fin de Máster se expone en la figura 1.5.

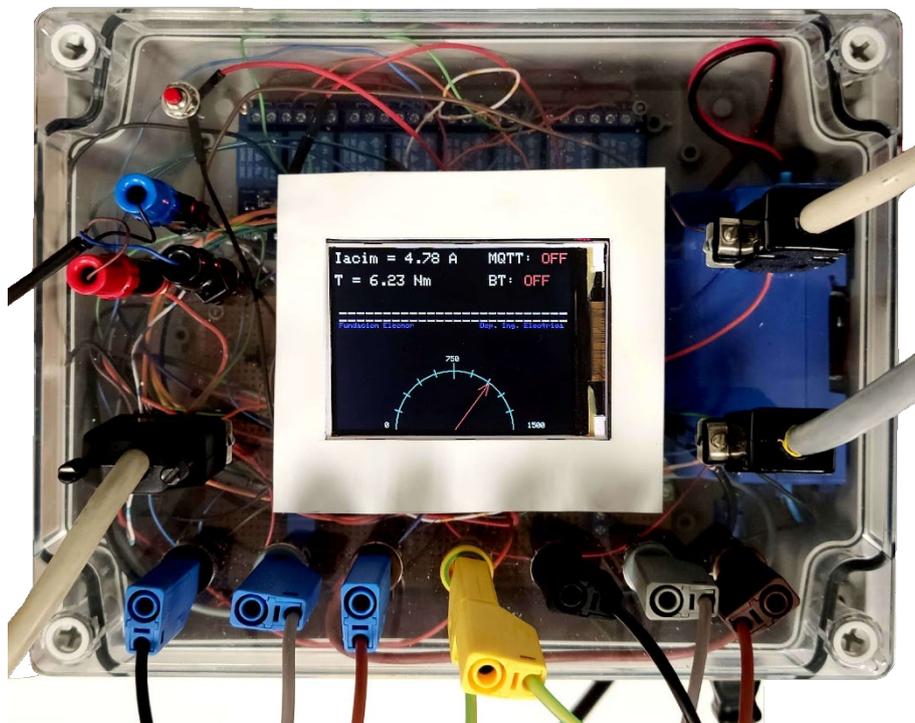


Figura 1.5. Producto diseñado.

En definitiva, los objetivos a alcanzar en el presente proyecto son:

- Diseño de un modelo virtual del banco de ensayos que permita realizar simulaciones asociadas a diferentes tipos de control del motor de inducción, en particular el control escalar, vectorial y directo de par. El modelo digital se diseñará a partir de las ecuaciones diferenciales que rigen el comportamiento del motor. Además, se debe resolver el modelo de forma numérica con el fin de implementarlo en el microcontrolador, permitiendo simular en tiempo real el modelo del motor de inducción.
- Selección de los sensores e instrumentación asociada que permitan la monitorización y control del puesto de prácticas. Las magnitudes que se requieren monitorizar son: posición y velocidad angular del eje del motor, par electromagnético y corrientes de estator.
- Diseño del software embebido del microcontrolador, desarrollando el código necesario que permita implementar en tiempo real el control y monitorización de cualquier puesto de prácticas del laboratorio.
- Visualización local en una pantalla LCD de la velocidad, el par electromagnético y la corriente consumida por el motor de inducción.
- Diseño de una aplicación Android en la que se representen las diferentes magnitudes deseadas y que permita controlar los variadores de frecuencia del motor de inducción y su carga, un motor de imanes permanentes, posibilitando la sustitución de las diferentes botonas por una interfaz digital.

1.3 ESTRUCTURA DEL DOCUMENTO

El presente proyecto está dividido en dos partes bien diferenciadas. Por una parte, los diferentes modelos desarrollados, tanto los relativos al motor de inducción como a los sistemas de control, se elaboran en los capítulos 2 y 3, respectivamente, mientras que la parte experimental se elabora detalladamente en el resto de los capítulos. Más en concreto, las tareas desarrolladas en cada uno de los capítulos de los que está compuesto el documento se describen a continuación.

- Capítulo 1. Se realiza una introducción al proyecto con el fin de describir cada una de las partes que lo componen.
- Capítulo 2. En primer lugar, se introducen los fundamentos teóricos de las máquinas rotativas asíncronas para que el lector conozca su principio básico de funcionamiento. Seguidamente, se procede a desarrollar mediante Matlab/Simulink el modelo virtual del motor de inducción a través de sus ecuaciones características. Una vez validado el modelo en Simulink, se procede a resolver el modelo de forma numérica con el fin de implementar la resolución del sistema de ecuaciones diferenciales en el microcontrolador, permitiendo simular en tiempo real el modelo del motor de inducción. Además, con el objetivo de evidenciar el potencial de la simulación dinámica de procesos, se implementa el modelo diseñado en Python, demostrando que es posible modelar sistemas complejos empleando software de código abierto.

- Capítulo 3. Se implementa el modelo asociado al variador de frecuencia, incorporando el sistema de control del inversor, y se diseñan las estrategias de control escalar, control vectorial y control directo de par. Finalmente, se integran todas las simulaciones mediante el desarrollo de una aplicación gráfica mediante App Designer.
- Capítulo 4. Se diseña la instrumentación requerida para efectuar la lectura de las diferentes señales eléctricas. Se procede a analizar los fundamentos teóricos relativos a las diferentes tecnologías correspondientes a los sensores que se han empleado en el proyecto, analizando múltiples alternativas que implementan la lectura de cada variable para seguidamente seleccionar la opción más adecuada. Finalmente, se expone el hardware integrado en el producto desarrollado.
- Capítulo 5. Se expone el eje central de la parte experimental del proyecto: el microcontrolador ESP32. Se detallan las diferentes tareas que debe ejecutar el microcontrolador para poder controlar y monitorizar en tiempo real el banco de ensayos. Se expone cómo implementar la transmisión de datos mediante *bluetooth* y MQTT y, finalmente, se elabora una interfaz que permite visualizar de forma local la velocidad, el par y la corriente consumida del motor de inducción mediante el empleo de una pantalla LCD.
- Capítulo 6. Se desarrolla una aplicación para dispositivos móviles mediante Android Studio que permita obtener un registro histórico de las magnitudes asociadas al conjunto de motores acoplados y controlar los variadores de frecuencia. Se detalla el diseño del sistema de comunicaciones mediante *bluetooth* y el protocolo MQTT.
- Capítulo 7. Se exponen los resultados finales derivados del desarrollo del presente proyecto. Por una parte, se realizan diferentes ensayos alternando entre la alimentación del motor de inducción a través del variador de frecuencia y mediante conexión directa a red. Estos ensayos se realizarán sobre dos motores cuyas clases de eficiencia sean diferentes, en particular los estándares IE1 e IE2. Se compararán los datos extraídos, haciendo especial énfasis en la corriente consumida por estos frente a las mismas condiciones de operación, permitiendo analizar la eficiencia de estos motores. Seguidamente, se presentarán los resultados asociados a la implementación del proceso de filtrado digital, comparando los resultados obtenidos mediante aplicación del filtro digital con los resultados que se obtendrían desactivando dicho proceso de filtrado. Finalmente, se expondrá la interfaz local de comunicaciones diseñada e implementada de forma conjunta con la visualización de datos a través de la aplicación Android.
- Capítulo 8. El capítulo final de la memoria está dedicado a extraer las principales conclusiones derivadas de la elaboración del presente proyecto y a analizar posibles aplicaciones del producto desarrollado.

CAPÍTULO 2: MODELADO Y SIMULACIÓN DEL MOTOR DE INDUCCIÓN

En este capítulo se procede a implementar un modelo dinámico del motor de inducción. En esta fase del proyecto no se tendrá en cuenta el modelo del variador de frecuencia, excluyendo las diferentes estrategias de control asociadas, las cuales serán descritas en el capítulo 3.

El apartado 2.1 está dedicado a la introducción del principio básico de funcionamiento del motor de inducción, detallando características importantes tales como el circuito eléctrico equivalente o la curva de par – velocidad con el fin de que el lector se familiarice con diferentes conceptos que se tratarán a lo largo del capítulo. En el apartado 2.2 se desarrolla la implementación del modelo digital en Simulink. Los parámetros del motor se han obtenido de forma experimental tal como se detalla en el epígrafe 2.3. La validación de la correcta implementación del sistema de ecuaciones diferenciales se realiza en el apartado 2.4. En el apartado 2.5 se detallan los pasos a seguir para implementar el modelo digital en un microcontrolador. Finalmente, en el apartado 2.6 se efectúa la adaptación del modelo al lenguaje de programación Python.

Cabe resaltar que el modelo digital del banco de ensayos formado por el motor de inducción y el variador de frecuencia y desarrollado en Matlab/Simulink se ha publicado en un repositorio online en GitHub [42]. La estructura de todo el código disponible se detalla en el anexo 4.

2.1 PRINCIPIO DE FUNCIONAMIENTO

En este apartado se pretende dotar al lector de las nociones básicas relativas al principio de funcionamiento de las máquinas eléctricas rotativas asíncronas o motores de inducción.

Un motor de inducción está compuesto por dos partes bien diferenciadas: el estator y el rotor. El primero de estos se encuentra fijo y es donde se ubican las bobinas estacionarias mientras que en el segundo se sitúan las bobinas móviles, cuya velocidad de giro se denota por Ω_r . Cada una de las bobinas, tanto del estator como del rotor, se encuentran desfasadas 120° entre ellas.

El estator no presenta grandes diferencias respecto del resto de motores eléctricos, encontrando las principales diferencias en el rotor. En el caso de los motores de inducción, las bobinas rotóricas se encuentran cortocircuitadas. Sin embargo, según la forma en la que estén cortocircuitadas estas bobinas, existen dos tipos de motores de inducción [6]:

- **Jaula de ardilla.** El cortocircuito de las bobinas rotóricas se efectúa en el interior de la máquina. Su uso es el más generalizado en la industria debido a que su empleo presenta diversas ventajas: es una máquina de construcción simple, peso reducido y mínimo volumen, además de llevar asociado un coste bajo y un mantenimiento inferior al del resto de máquinas eléctricas rotativas.
- **Rotor bobinado.** El cortocircuito de las bobinas del rotor se realiza en el exterior de la máquina. Sus principales ventajas respecto de los motores de inducción de jaula de ardilla son: la tensión e intensidad del rotor son magnitudes medibles y las características del circuito eléctrico del rotor son accesibles desde el exterior.

Respecto a su funcionamiento, el estator es alimentado mediante un sistema trifásico de tensiones cuya frecuencia se denomina frecuencia de sincronismo. Dicha alimentación se efectúa a través de la red eléctrica si se desea que el sistema trifásico tenga una frecuencia de 50Hz (en Europa) o bien mediante un variador de frecuencia si la aplicación industrial requiere el empleo de una frecuencia diferente.

Este sistema trifásico de tensiones genera un sistema trifásico de corrientes que, a su vez, generará un campo magnético giratorio del estator. Cuando éste gira, se produce una variación del flujo que atraviesa las espiras del rotor dando lugar, según la ley de Lenz, a una fuerza electromotriz cuyo sentido es tal que el flujo generado por ésta se opone a la variación del flujo generador, es decir, tratará de minimizar su variación, por lo que el flujo del rotor tratará de seguir al flujo del estator.

En consecuencia, por lo comentado anteriormente, se induce una fuerza electromotriz en el circuito eléctrico del rotor de tal modo que, al encontrarse cortocircuitado, se generará un sistema de corrientes inducidas en el propio rotor que dará lugar al campo magnético del rotor cuya superposición con el del estator producirá el campo magnético resultante del motor de inducción.

Como resultado, se genera un par electromagnético que producirá el giro del rotor a una velocidad inferior a la de sincronismo ya que, si su velocidad fuera la de sincronismo no existiría una variación de flujo, desaparecería la fuerza electromotriz inducida y por tanto no se generaría el par electromagnético que produce el giro del rotor.

Por ello, estas máquinas rotativas también se conocen como motores asíncronos, ya que el rotor no gira a la misma velocidad que el campo magnético generado por el estator.

Al cociente de la diferencia entre ambas velocidades y la velocidad de sincronismo se le conoce por deslizamiento. Además, este cociente también es la relación entre las frecuencias de las corrientes de rotor y estator. En definitiva, el deslizamiento es la diferencia relativa entre la velocidad de sincronismo, asociada al sistema trifásico de tensiones de alimentación del motor, y la velocidad mecánica del eje, cuya expresión matemática se expone a continuación:

$$(2.1) \quad s = \frac{\Omega_s - \Omega_r}{\Omega_s} = \frac{w_r}{w_s} = \frac{f_r}{f_s}$$

La velocidad de sincronismo y la frecuencia de las tensiones de alimentación del motor están relacionadas a través del número de par de polos según la ecuación 2.2.

$$(2.2) \quad \Omega_s = f_s \cdot \frac{60}{p}$$

2.1.1 CIRCUITO ELÉCTRICO EQUIVALENTE

Un motor de inducción es un sistema cuyo análisis es relativamente complejo. En consecuencia, la obtención de un circuito eléctrico equivalente que permita la aplicación de las leyes que rigen la teoría de circuitos posibilita la obtención de magnitudes tales como potencia, par o corrientes, entre otras.

El circuito exacto equivalente de un motor asíncrono se expone en la figura 2.1.

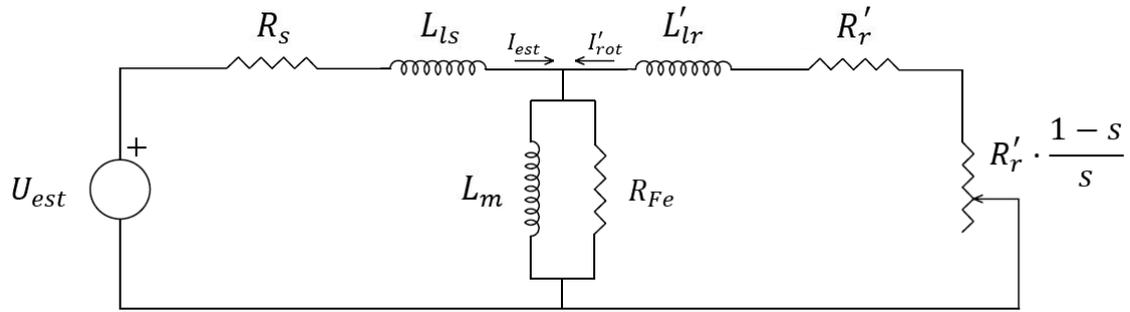


Figura 2.1. Circuito eléctrico equivalente.

Sin embargo, a nivel industrial es habitual trabajar con una versión simplificada del circuito eléctrico equivalente de la figura 2.1. La modificación del circuito eléctrico consiste en desplazar la rama paralelo y agrupar los términos de resistencias de pérdidas en el cobre e inductancias de dispersión, tal como se expone en la figura 2.2.

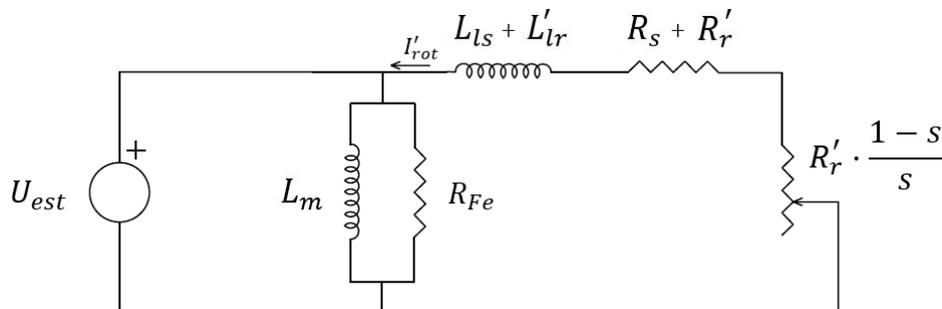


Figura 2.2. Circuito eléctrico equivalente simplificado.

2.1.2 CURVA PAR - VELOCIDAD

Una de las características más importantes de los motores de inducción consiste en la curva de par en función del deslizamiento. La intersección entre la curva de par motor y la de par resistente proporciona el punto de operación de la máquina.

El par del motor de inducción se puede obtener a partir de la potencia mecánica transmitida al eje y la velocidad de giro de este. Además, la velocidad de giro del rotor y la frecuencia de sincronismo están relacionadas según la ecuación 2.1. La potencia mecánica se puede obtener a partir del circuito eléctrico equivalente tal como se detalla en 2.3.

$$(2.3) \quad T_{mi} = \frac{P_{mi}}{\Omega_r} = \frac{P_{mi}}{\Omega_s \cdot (1-s)}, \quad P_{mi} = 3 \cdot I'_{rot}{}^2 \cdot \frac{R'_r \cdot (1-s)}{s}$$

$$T_{mi} = \frac{P_{mi}}{\Omega_s} \cdot 3 \cdot I'_{rot}{}^2 \cdot \frac{R'_r}{s}$$

Es posible expresar el valor de I'_{rot} en función de la tensión del estator, las resistencias de pérdidas en el hierro y las reactancias de dispersión

$$(2.4) \quad I'_{rot} = \frac{U_{est}}{Z_{eq}} = \frac{U_{est}}{\sqrt{\left(R_s + \frac{R'_r}{s}\right)^2 + (X_{ls} + X'_{lr})^2}}$$

Sustituyendo 2.4 en 2.3 se obtiene la ecuación que permite representar gráficamente el par del motor de inducción en función del desplazamiento y los parámetros del circuito equivalente.

$$(2.5) \quad T_{mi} = \frac{3 \cdot R_r'}{\Omega_s \cdot s} \cdot \frac{U_{est}^2}{\left(R_s + \frac{R_r'}{s}\right)^2 + (X_{ls} + X_{lr}')^2}$$

La ecuación 2.5 tiene su principal aplicación en el control de los motores de inducción ya que permite analizar la variación de la curva de par en función de variaciones en la tensión de alimentación y la frecuencia de sincronismo. La representación gráfica del par frente a cambios de tensión y frecuencia se efectúa en la figura 2.3.

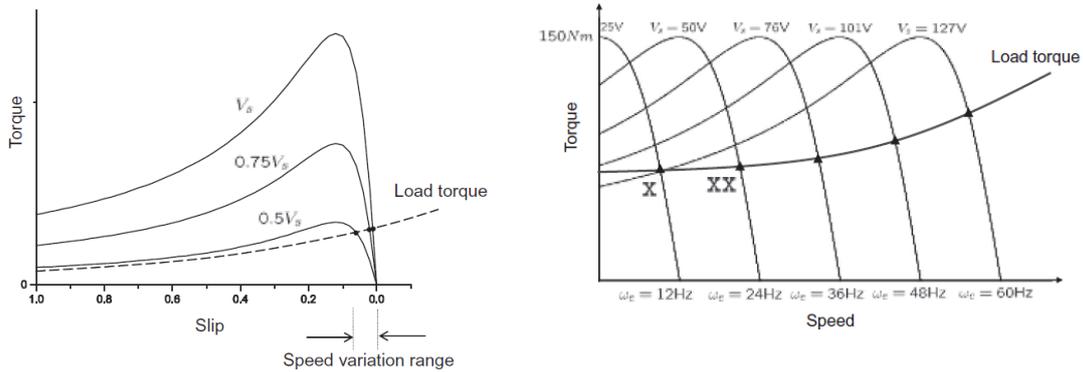


Figura 2.3a. Cambios en la tensión de alimentación. Figura 2.3b. Cambios en la frecuencia de sincronismo.

Figura 2.3. Curva par-velocidad [8].

2.2 MODELO DEL MOTOR DE INDUCCIÓN

Al inicio del desarrollo del proyecto, se ha elaborado un modelo del motor de inducción en ejes dq , cuyo código asociado se expone en el anexo 1. El principal inconveniente de este modelo consiste en que requiere conocer la frecuencia de sincronismo correspondiente a las tensiones de alimentación del motor debido a que el sistema de referencia respecto del cual se desarrolla el modelo gira a la velocidad de sincronismo del sistema de tensiones trifásico. Este dato es proporcionado directamente por el control escalar, pero en otros sistemas de control tales como el control vectorial, dicha variable no es accesible directamente.

En consecuencia, se ha optado por desarrollar un modelo del motor de inducción en ejes $\alpha\beta$, es decir, dicho modelo emplea un sistema de referencia fijo, no siendo necesario el conocimiento de la frecuencia de sincronismo de las tensiones de alimentación.

El modelo del motor de inducción basado en las componentes $\alpha\beta$ ha sido elaborado tanto en Simulink como en Python, demostrando que es posible implementar la simulación de sistemas dinámicos en herramientas de código libre. Los resultados de la simulación realizada en Python se detallan en el apartado 2.6.

En la figura 2.4 se exponen los diferentes elementos que componen la simulación del motor en bucle abierto.

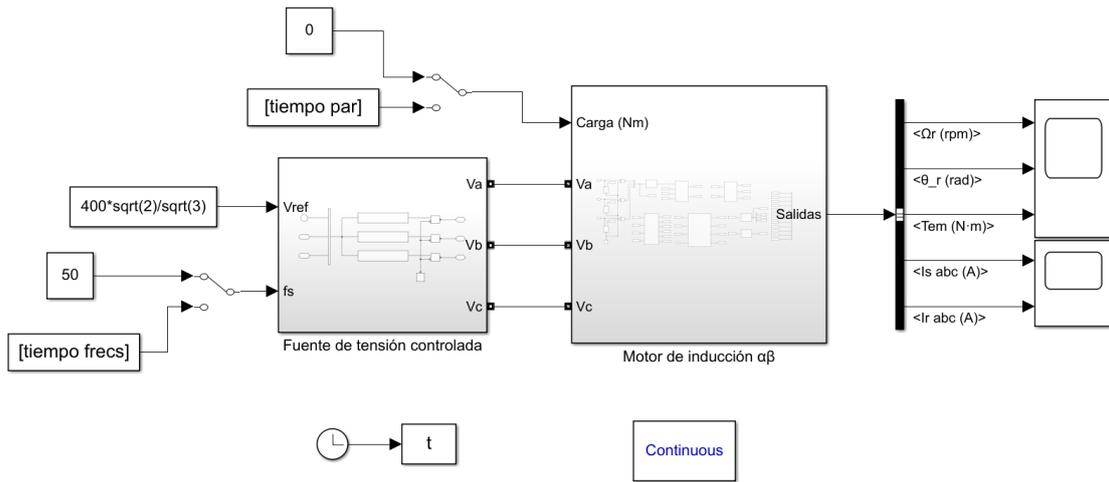


Figura 2.4. Motor de inducción en bucle abierto.

El modelo del motor de inducción ha sido desarrollado en ejes $\alpha\beta$, cuyas ecuaciones dinámicas se exponen a continuación. Dichas ecuaciones son un caso particular de las relativas al modelo en ejes dq . Las diferentes ecuaciones han sido agrupadas en diferentes bloques de Simulink, principalmente en ecuaciones de flujo, de corrientes y mecánicas,

De este modo, particularizando las ecuaciones para un eje fijo se obtienen, en primer lugar, las ecuaciones de flujo que se detallan en 2.6 cuya implementación se realiza en la figura 2.5.

$$(2.6) \quad \begin{aligned} \frac{d\psi_{s\alpha}}{dt} &= v_{s\alpha} - i_{s\alpha} \cdot R_s, & \frac{d\psi_{s\beta}}{dt} &= v_{s\beta} - i_{s\beta} \cdot R_s \\ \frac{d\psi_{r\alpha}}{dt} &= -i_{r\alpha} \cdot R_r - \omega_r \cdot \psi_{r\beta}, & \frac{d\psi_{r\beta}}{dt} &= -i_{r\beta} \cdot R_r + \omega_r \cdot \psi_{r\alpha} \end{aligned}$$

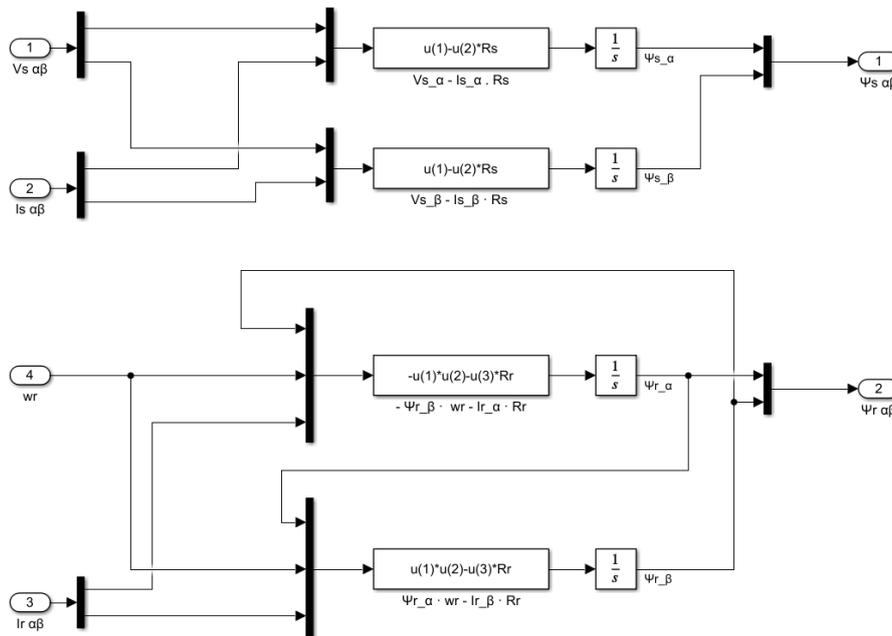


Figura 2.5. Cómputo de flujos de estator y rotor.

Respecto al cómputo de las corrientes de estator y rotor, es necesario tener en cuenta que el sistema de ecuaciones a resolver depende de la inclusión en el modelado del motor de inducción

de las pérdidas en el hierro. En la figura 2.6 se observa que, en función del valor de R_{Fe} , se ejecuta un sistema de ecuaciones u otro.

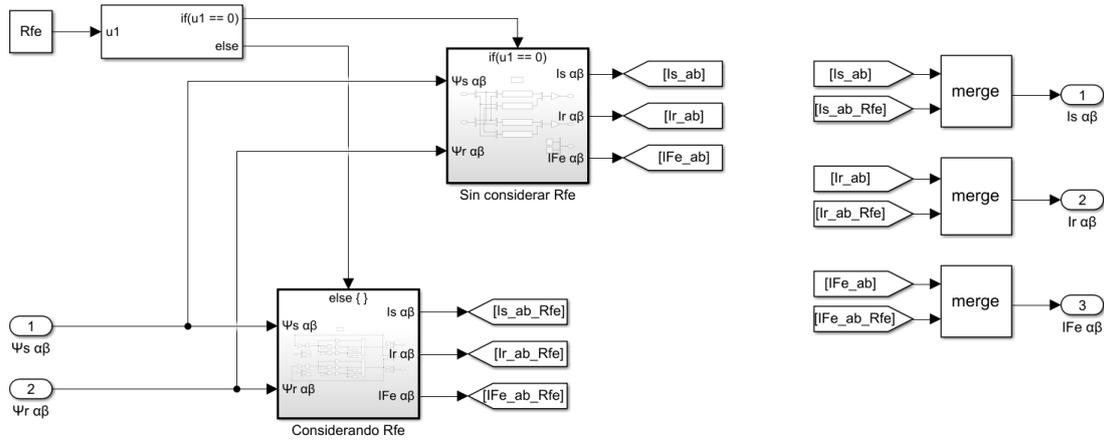


Figura 2.6. Selección de ecuaciones para el cómputo de las corrientes.

En particular, sin incluir dichas pérdidas, las ecuaciones relativas a las corrientes de estator y rotor viene dadas por 2.7 y han sido implementadas en la figura 2.7.

$$(2.7) \quad \begin{aligned} i_{s\alpha} &= \frac{L_r \cdot \psi_{s\alpha} - L_m \cdot \psi_{r\alpha}}{L_x}, & i_{s\beta} &= \frac{L_r \cdot \psi_{s\beta} - L_m \cdot \psi_{r\beta}}{L_x} \\ i_{r\alpha} &= \frac{L_s \cdot \psi_{r\alpha} - L_m \cdot \psi_{s\alpha}}{L_x}, & i_{r\beta} &= \frac{L_s \cdot \psi_{r\beta} - L_m \cdot \psi_{s\beta}}{L_x} \\ L_x &= L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m \end{aligned}$$

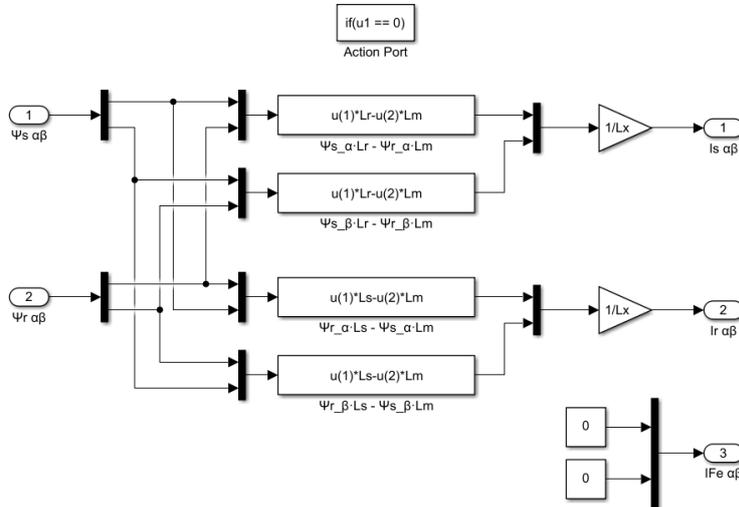


Figura 2.7. Cómputo de corrientes de estator y rotor - $R_{Fe} = \infty$.

Si se consideran las pérdidas en el hierro es necesario modificar el sistema de ecuaciones anterior, ya que aparece una nueva corriente en el circuito equivalente: la corriente de pérdidas en el hierro, i_{Fe} . En consecuencia, la corriente magnetizante no será directamente la suma de las corrientes de estator y rotor.

$$(2.8) \quad \frac{d\psi_{m\alpha}}{dt} = R_{Fe} \cdot i_{Fe\alpha}, \quad \frac{d\psi_{m\beta}}{dt} = R_{Fe} \cdot i_{Fe\beta}$$

Teniendo en cuenta la propia definición de la inductancia magnetizante, es decir, la inductancia es igual al cociente entre el flujo magnético y la corriente que genera dicho flujo, se obtiene:

$$(2.9) \quad L_m = \frac{\psi_{m\alpha}}{i_{m\alpha}} \rightarrow i_{m\alpha} = \frac{\psi_{m\alpha}}{L_m}, \quad L_m = \frac{\psi_{m\beta}}{i_{m\beta}} \rightarrow i_{m\beta} = \frac{\psi_{m\beta}}{L_m}$$

Realizando un balance de flujos tanto para estator como para el rotor, es posible determinar las componentes asociadas a las corrientes de estator y rotor.

$$(2.10) \quad \begin{aligned} \psi_{s\alpha} &= L_{ls} \cdot i_{s\alpha} + \psi_{m\alpha} \rightarrow i_{s\alpha} = \frac{\psi_{s\alpha} - \psi_{m\alpha}}{L_{ls}} \\ \psi_{s\beta} &= L_{ls} \cdot i_{s\beta} + \psi_{m\beta} \rightarrow i_{s\beta} = \frac{\psi_{s\beta} - \psi_{m\beta}}{L_{ls}} \\ \psi_{r\alpha} &= L_{lr} \cdot i_{r\alpha} + \psi_{m\alpha} \rightarrow i_{r\alpha} = \frac{\psi_{r\alpha} - \psi_{m\alpha}}{L_{lr}} \\ \psi_{r\beta} &= L_{lr} \cdot i_{r\beta} + \psi_{m\beta} \rightarrow i_{r\beta} = \frac{\psi_{r\beta} - \psi_{m\beta}}{L_{lr}} \end{aligned}$$

Finalmente, a partir de un balance de corrientes aplicado sobre la rama central del circuito equivalente se obtiene la corriente de pérdidas en el hierro.

$$(2.11) \quad i_{Fe\alpha} = i_{s\alpha} + i_{r\alpha} - i_{m\alpha}, \quad i_{Fe\beta} = i_{s\beta} + i_{r\beta} - i_{m\beta}$$

De este modo se tiene un sistema de 10 ecuaciones con 10 incógnitas, cuya implementación en Simulink se expone en la figura 2.8.

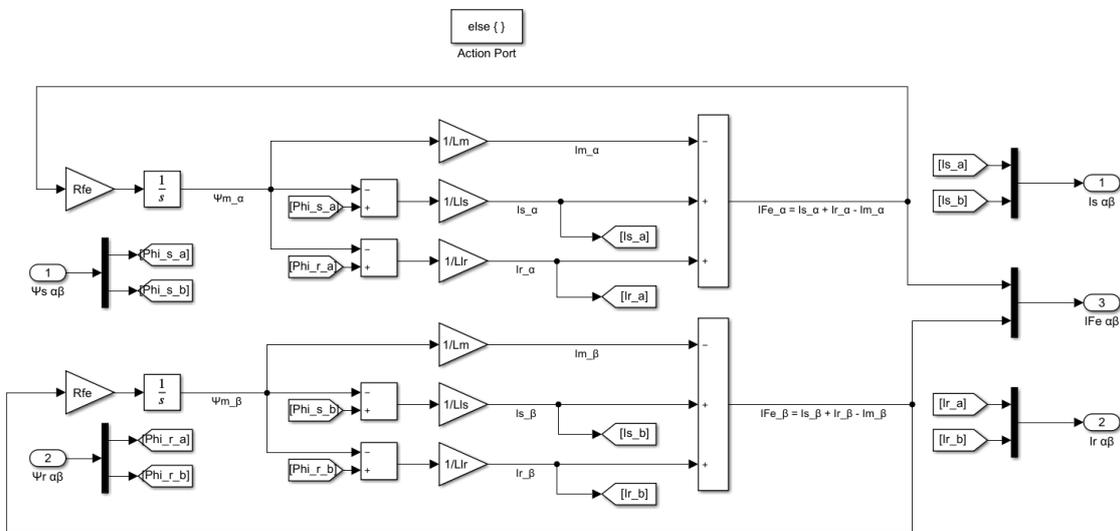


Figura 2.8. Cómputo de corrientes de estator y rotor - $R_{Fe} < \infty$.

Finalmente, se realiza el cálculo del par electromagnético y, a partir de un balance de momentos aplicado sobre el eje del motor eléctrico, se determina la velocidad y, a partir de esta, la posición angular del mismo. El conjunto de ecuaciones mecánicas se detalla en 2.12.

$$(2.12) \quad \begin{aligned} T_{em} &= \frac{3}{2} \cdot p_{ACIM} \cdot \frac{L_m}{L_{lr} + L_m} \cdot ((i_{s\beta} - i_{Fe\beta}) \cdot \psi_{r\alpha} - (i_{s\alpha} - i_{Fe\alpha}) \cdot \psi_{r\beta}) \\ \frac{d\Omega_r}{dt} &= \frac{T_{em} - F \cdot \Omega_r - T_{carga}}{J}, \quad \theta = \int \Omega_r \cdot dt \end{aligned}$$

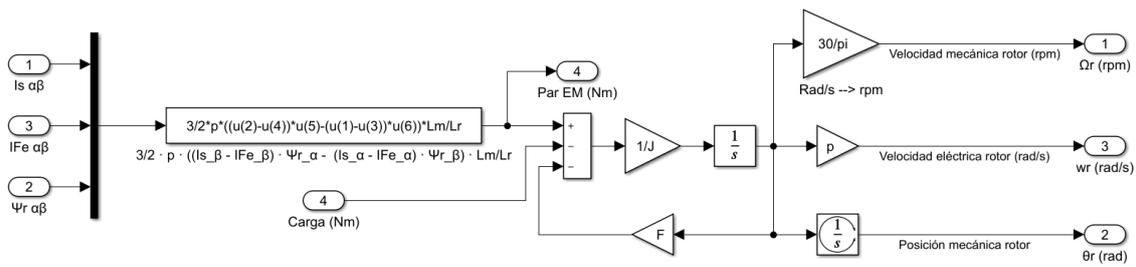


Figura 2.9. Ecuaciones mecánicas.

Todas las ecuaciones anteriores han sido implementadas según el esquema de la figura 2.10. En el bloque de medidas auxiliares simplemente se efectúa una transformación de algunas variables a ejes d, q con el fin de poder representar un diagrama fasorial de estas magnitudes, además de calcular el módulo del fasor de corriente de estator y el ángulo del flujo del rotor. Estas medidas no son esenciales para el correcto funcionamiento del modelo.

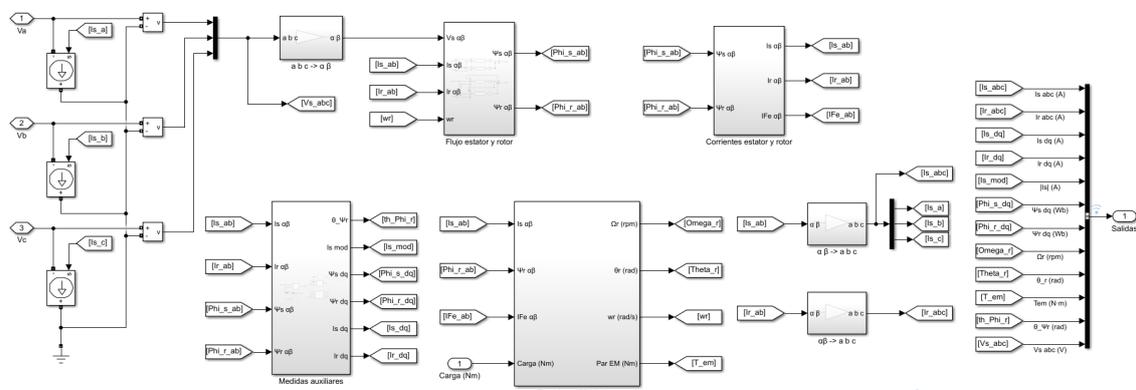


Figura 2.10. Esquema general del motor de inducción.

Tal como se ha comentado anteriormente, la simulación de las diferentes estrategias de control se realiza en el capítulo 3, de tal modo que en el presente capítulo la alimentación del motor se efectuará a través de una fuente de tensión controlada con el objetivo de validar el modelo del motor.

El modelo de la fuente de tensión controlada se expone en la figura 2.11, donde se observa que es posible modificar tanto la tensión eficaz como la frecuencia asociada a las tensiones de alimentación del motor.

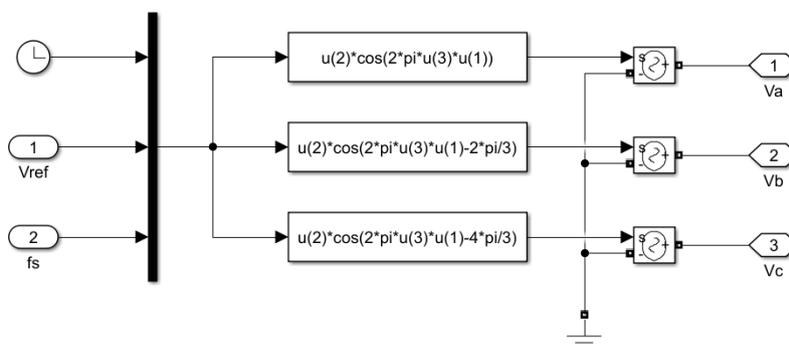


Figura 2.11. Fuente de tensión controlada.

2.3 OBTENCIÓN DE PARÁMETROS

Tal como se ha podido evidenciar anteriormente, el modelo del ACIM requiere la obtención de los parámetros del circuito equivalente. Estos se obtienen a partir de los ensayos de vacío, cortocircuito y de continua. En particular, los parámetros que se calculan en cada uno de los ensayos se detalla a continuación:

- Ensayo de continua: resistencias estáticas.
- Ensayo de cortocircuito: resistencias rotóricas e inductancias de la rama serie.
- Ensayo de vacío: resistencia de pérdidas en el hierro e inductancia magnetizante.

2.3.1 ENSAYO DE CONTINUA

Con el objetivo de determinar el valor de la resistencia estática, R_s , se alimenta una de las fases del estator mediante corriente continua, midiendo la tensión aplicada y la corriente consumida es posible calcular la resistencia del estator.

$R_{uv} (\Omega)$	$R_{uw} (\Omega)$	$R_{vw} (\Omega)$	$R_{med} (\Omega)$	$R_s = \frac{R_{med}}{2} (\Omega)$
6.6	6.5	6.7	6.6	3.3

Tabla 2.1. Resistencias estáticas.

2.3.2 ENSAYO DE CORTOCIRCUITO

También conocido como ensayo a rotor bloqueado, consiste en someter el motor a su corriente nominal manteniendo su rotor bloqueado. Permite obtener R_r' , L_{ls} y L_{lr} . En la tabla 2.2 se presentan los resultados del ensayo.

$U_{cc} (V)$	$I_{cc,L} (A)$	$P_{cc} (W)$	$Q_{cc} (VAr)$	$U_1 (V)$	$U_2 (V)$	$U_3 (V)$	$I_1 (A)$	$I_2 (A)$	$I_3 (A)$
88.03	4.75	420	586	88.2	88.4	87.5	4.75	4.7	4.8

Tabla 2.2. Ensayo de cortocircuito.

Despreciando la corriente de vacío por presentar un valor muy inferior a la corriente de cortocircuito, se cumplen las expresiones $R_{cc} = R_s + R_r'$ y $X_{cc} = X_s + X_r'$

Mediante de las potencias definidas a partir de los valores de fase, es posible determinar la resistencia y reactancia de cortocircuito según 2.13 y 2.14.

$$(2.13) \quad P_{cc} = 3 \cdot R_{cc} \cdot I_{cc}^2 \rightarrow R_{cc} = \frac{P_{cc}}{3 \cdot I_{cc}^2} = 6.205 \Omega$$

$$(2.14) \quad Q_{cc} = 3 \cdot X_{cc} \cdot I_{cc}^2 \rightarrow X_{cc} = \frac{Q_{cc}}{3 \cdot I_{cc}^2} = 8.657 \Omega$$

Finalmente, empleando los valores de la resistencia estática hallada en el epígrafe 2.3.1, es posible calcular la resistencia rotórica y las inductancias de la rama serie, asumiendo que se cumple que $X_s \approx X_r'$.

$$(2.15) \quad R_{cc} = R_s + R_r' \rightarrow R_r' = R_{cc} - R_s = 2.905 \Omega$$

$$(2.16) \quad X_{cc} = X_s + X_r' \rightarrow X_s = X_r' = \frac{X_{cc}}{2} = 4.329 \Omega \rightarrow L_{ls} = L_{lr} = \frac{X_s}{2 \cdot \pi \cdot 50} = 0.0138 H$$

2.3.3 ENSAYO DE VACÍO

Se ensaya el motor a tensión nominal y a tensión reducida, dejando que gire libremente. Esto permite obtener los valores de R_{Fe} y L_m .

$U_0 (V)$	$I_{0L} (A)$	$P_0 (W)$	$Q_0 (VAr)$	$U_1 (V)$	$U_2 (V)$	$U_3 (V)$	$I_1 (A)$	$I_2 (A)$	$I_3 (A)$
400.33	3.43	312	2354	401	401	399	3.4	3.38	3.51
201.23	0.993	110	326	201	201.7	201	0.984	0.986	1.01

Tabla 2.3. Ensayo de vacío.

En primer lugar, el cálculo más inmediato es el de la inductancia magnetizante. A partir de un balance de potencias reactivas se concluye lo siguiente:

$$(2.17) \quad Q_0 = Q_{Lm} + Q_{Ls} + Q_{Lr}$$

Empleando las reactancias obtenidas en el apartado 2.3.1 se obtiene la potencia reactiva asociada a la inductancia magnetizante:

$$(2.18) \quad Q_{Ls} = Q_{Lr} = 3 \cdot X_s \cdot I_{0F}^2 = 3 \cdot X_s \cdot I_{0L}^2 = 152.791 VAr$$

$$(2.19) \quad Q_{Lm} = Q_0 - Q_{Ls} - Q_{Lr} = 2048.418 VAr$$

$$(2.20) \quad Q_{Lm} = 3 \cdot \frac{U_f^2}{X_m} \rightarrow X_m = \frac{U_L^2}{Q_{Lm}} = 78.238 \Omega \rightarrow L_m = \frac{X_m}{2 \cdot \pi \cdot 50} = 0.249 H$$

Seguidamente, se procede a calcular la resistencia de pérdidas en el hierro.

El balance de potencias válido para los dos puntos de operación de la tabla 2.3 se expone en la ecuación 2.21, en la que la potencia total se divide en tres términos: potencia de pérdidas en el cobre (P_c), potencia de pérdidas en el hierro (P_{Fe}) y potencias de pérdidas mecánicas debido a la fricción (P_{mec}).

$$(2.21) \quad P_0 = P_c + P_{Fe} + P_{mec}$$

El término que se puede determinar de forma inmediata son las pérdidas en el cobre mediante la ecuación 2.22.

$$(2.22) \quad P_c = 3 \cdot R_s \cdot I_{0F}^2 = 3 \cdot R_s \cdot I_{0L}^2$$

Se cumple que la suma de las pérdidas en el hierro y las pérdidas mecánicas es proporcional al cuadrado de la tensión siguiendo una relación lineal tal como expresa la ecuación 2.23.

$$(2.23) \quad P_{Fe} + P_{mec} = P_0 - P_c = k \cdot U^2 + n$$

Las pérdidas mecánicas son constantes e independientes de la tensión aplicada. Además, cuando la tensión es nula se cumple que $P_{Fe} = 0$, por lo que extrapolando los resultados de la tabla 2.3 para tensión nula es posible cuantificar las pérdidas mecánicas, es decir, la ordenada en el origen serán las pérdidas mecánicas, $n = P_{mec}$. En la tabla 2.4 se exponen resultados derivados de las expresiones anteriores.

$U_0 (V)$	$I_{0L}(A)$	$P_0(W)$	$P_c(W)$	$P_0 - P_c(W)$
400.33	3.43	312	102.531	209.469
201.23	0.993	110	8.593	101.407

Tabla 2.4. Cálculos auxiliares.

A partir de los datos anteriores es posible obtener el factor de proporcionalidad k como la pendiente de la recta que relaciona $P_0 - P_c$ y U^2 y la ordenada en el origen n .

$$(2.26) \quad k = \frac{(P_{01} - P_{c1}) - (P_{02} - P_{c2})}{U_1^2 - U_2^2} = 0.0009 \frac{W}{V^2}$$

$$(2.27) \quad n = P_{mec} = (P_{01} - P_{c1}) - k \cdot U_1^2 = 64.872 W$$

Finalmente, se calcula la potencia de pérdidas en el hierro para la tensión nominal:

$$(2.28) \quad P_{Fe} = P_0 - P_c - P_{mec} = 144.597 W$$

$$(2.29) \quad P_{Fe} = 3 \cdot \frac{U_F^2}{R_{Fe}} \rightarrow R_{Fe} = \frac{U_L^2}{P_{Fe}} = 1108.35 \Omega$$

2.4 VALIDACIÓN DEL MODELO

Para poder analizar la correcta implementación del sistema de ecuaciones diferenciales se procede a comparar la respuesta proporcionada por el modelo digital diseñado con el modelo implementado en la librería Simscape de Simulink, tal como se expone en la figura 2.12.

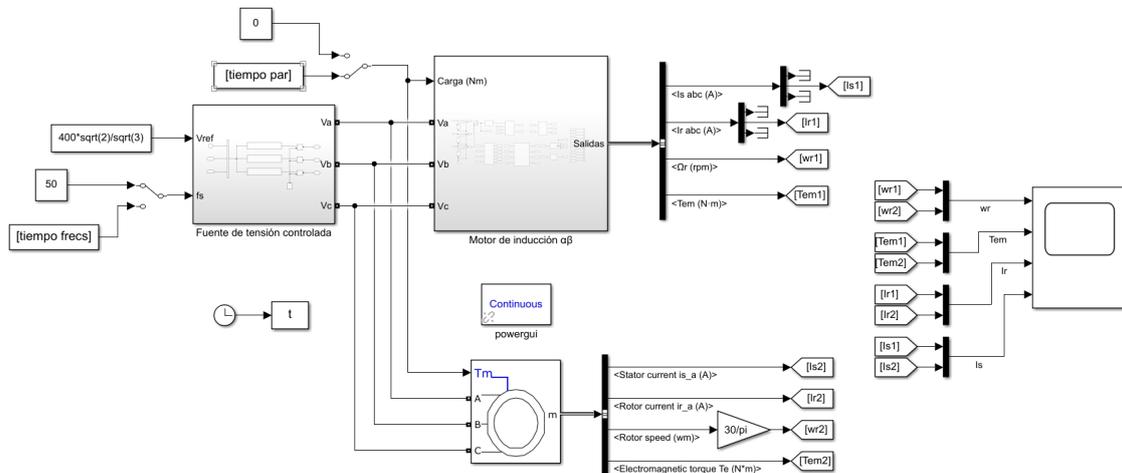


Figura 2.12. Validación del modelo digital.

En particular, se compara la evolución temporal de una de las fases de la corriente de estator y rotor, la velocidad mecánica y el par electromagnético. Se han obtenido los mismos resultados en ambos modelos, validando de forma satisfactoria la implementación del sistema de ecuaciones diferenciales.

Finalmente, se ensaya el motor en bucle abierto sometiéndolo a una frecuencia fija y modificando el par de la carga en diferentes intervalos. Este mismo ensayo se ha realizado sobre el motor real permitiendo comparar su respuesta frente a la del modelo digital, cuyos resultados asociados se exponen en el capítulo 7.

Concretamente, las variables que se van a representar y que también se pueden extraer del sistema real son la velocidad mecánica, el par y el módulo de la corriente de estator. Estos resultados se presentan en la figura 2.13.

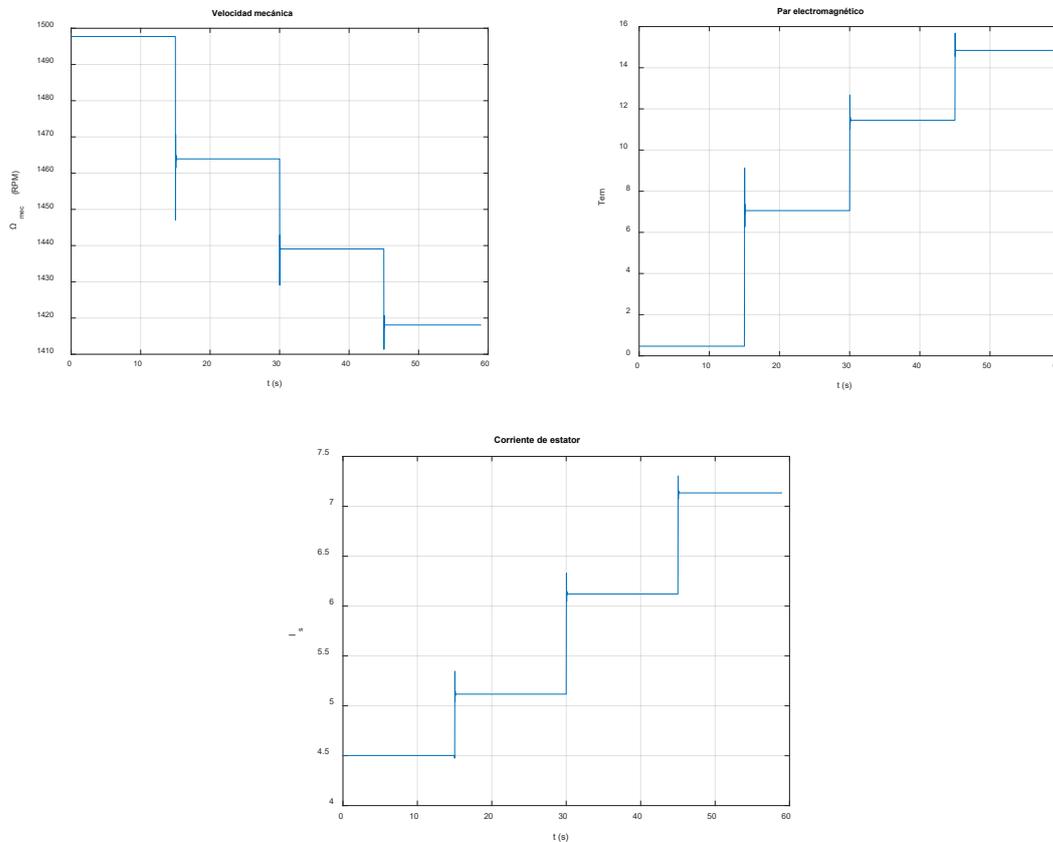


Figura 2.13. Resultados del modelo digital.

2.5 IMPLEMENTACIÓN DEL MODELO DEL ACIM EN MICROCONTROLADOR

En el presente epígrafe se detallarán los pasos a seguir para implementar un modelo digital en un microcontrolador. Esto permite desarrollar un gemelo digital [5], concepto empleado en la industria 4.0 y que podría ser empleado para comparar la respuesta real del banco de ensayos con la correspondiente al modelo digital en tiempo real.

Con el objetivo de implementar el modelo matemático del motor de inducción en el microcontrolador, es necesario resolver de forma numérica el sistema de ecuaciones diferenciales que se ha expuesto en el presente capítulo.

La resolución de forma numérica de dicho sistema de ecuaciones diferenciales se expone en el anexo 2, de modo que en el presente epígrafe únicamente se expondrán los resultados que se derivan de dicho anexo.

Tal como se detallará en el capítulo 4, el microcontrolador elegido en el proyecto es el ESP32. Si se incluían las pérdidas en el hierro en el modelo del ACIM, el tiempo de cómputo que requería la resolución del modelo en el ESP32 se incrementaba notablemente, por lo que finalmente no se han tenido en cuenta estas pérdidas.

2.5.1 SELECCIÓN DEL INTERVALO DE PASO Y ANÁLISIS DE ESTABILIDAD

Para resolver de forma numérica el modelo matemático es necesario seleccionar un intervalo de paso adecuado. Cabe recordar que dicho parámetro, denotado por h , se corresponde con el intervalo de tiempo que separa dos iteraciones consecutivas del algoritmo de cálculo.

Cuanto menor sea el intervalo de paso menor será el error cometido en las aproximaciones realizadas en el valor de las derivadas, pero se incrementará el tiempo de cómputo pudiendo ser irrealizable por el ESP32.

Se ha desarrollado un código de Matlab que permite validar la resolución numérica del modelo del motor de inducción. Se ha comparado la respuesta de la solución aportada con la resolución del modelo mediante el método ode45 ya implementado en Matlab. El código correspondiente se expone en el anexo 2.

Mediante el empleo de un intervalo de paso de $10\mu\text{s}$ se conseguían resultados aparentemente idénticos a los obtenidos mediante el método ode45 de Matlab. En la figura 2.14 se expone una comparativa entre las respuestas correspondientes a la resolución aportada y al método ode45.

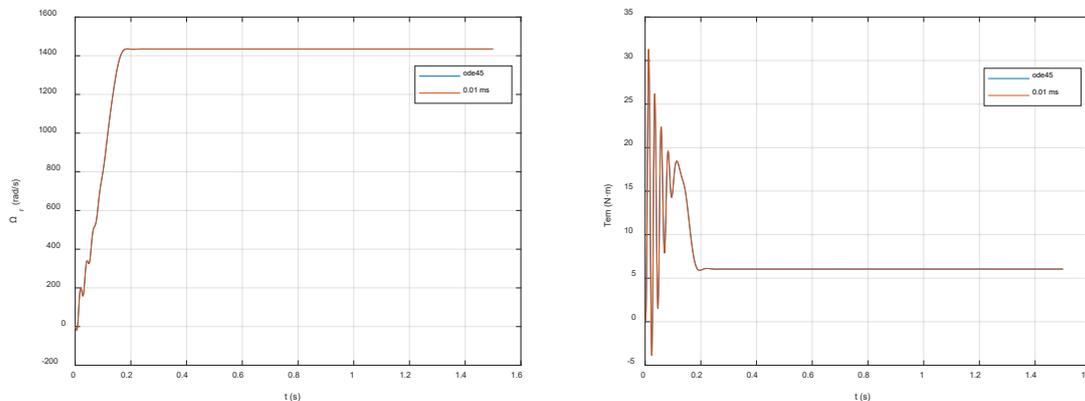


Figura 2.14. Validación de la resolución numérica del modelo.

Por otra parte, un intervalo de paso demasiado grande da lugar a un mayor error en la resolución del modelo y, además, puede inestabilizar el sistema.

El modelo ha sido implementado con éxito en el microcontrolador, requiriendo un tiempo de cómputo de aproximadamente $150\mu\text{s}$. Su resolución ha sido implementada en el *loop*, de modo que el coste computacional total empleado en la resolución del modelo y otras tareas como la conversión a *string* de diferentes variables y la transmisión de datos entre núcleos se eleva a unos $250\mu\text{s}$, por lo que ha sido éste el intervalo de paso empleado en la implementación del modelo en el ESP32. En este caso se produce un pequeño error entre la respuesta obtenida y la correspondiente a la resolución mediante ode45, tal como se puede observar en la figura 2.15.

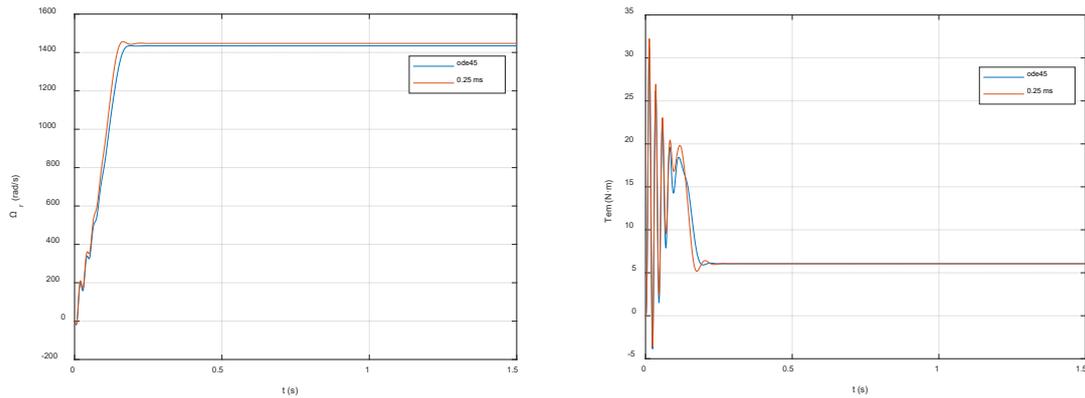


Figura 2.15. Comparativa entre la resolución propuesta y ode45.

En cuanto a la estabilidad del modelo, si se emplea un intervalo de paso excesivo puede inestabilizar el sistema. Para valores en torno de 1ms se producían oscilaciones en la respuesta del modelo, pero finalmente se alcanzaba un valor estable. Sin embargo, a partir del valor 1.1 ya no se lograba estabilizar el sistema tal como se observa en la figura 2.16, por lo que no se debería de emplear un intervalo de paso superior a ese umbral.

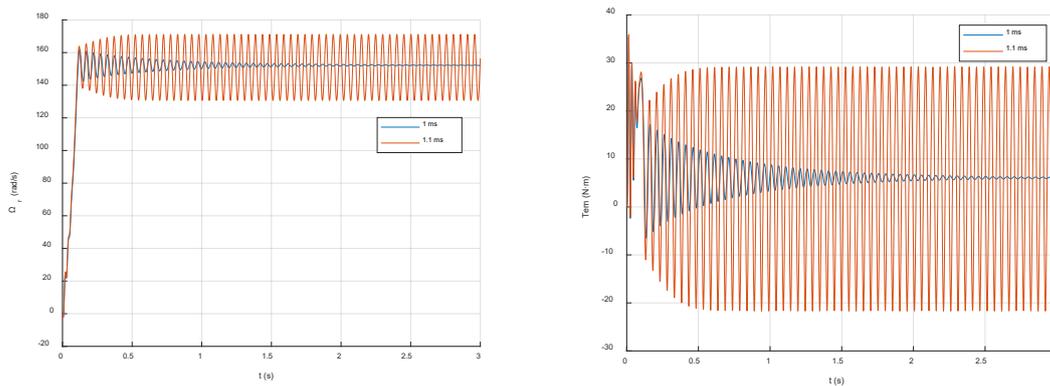


Figura 2.16. Análisis de estabilidad

El modelo virtual presenta características muy similares al sistema real. Se permite modificar la consigna de frecuencia y de par desde la propia aplicación Android, emulando el control de ambos variadores de frecuencia.

2.6 RESOLUCIÓN DEL MODELO EN PYTHON

El hecho de disponer del sistema de ecuaciones diferenciales del modelo digital permite adaptar la resolución del modelo a cualquier lenguaje de programación que permita resolver estos sistemas de ecuaciones de forma numérica.

Para facilitar esta adaptación, se ha propuesto resolver el modelo del motor de inducción en Python. Por una parte, es posible resolver sistemas de ecuaciones mediante métodos numéricos disponibles en la librería *scipy*. Sin embargo, existe una librería alternativa, denotada por *control*, que permite implementar simulaciones de sistemas dinámicos, tanto lineales como no lineales, y ofrece una característica interesante: permite agrupar cada sistema en un bloque de forma

equivalente a Simulink, realizando la interconexión entre los diferentes bloques no de forma gráfica, sino mediante código basado en texto.

En particular, se han definido dos bloques: uno asociado al modelo del motor de inducción y otro que implementará la fuente de tensión controlada.

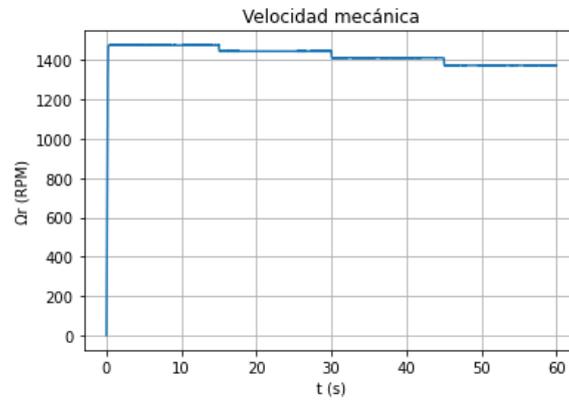


Figura 2.17. Resultados simulación en Python.

CAPÍTULO 3: MODELADO Y SIMULACIÓN DEL VARIADOR DE FRECUENCIA

En el presente capítulo se realiza la simulación de los sistemas de control que se encuentran implementados en el banco de ensayos del laboratorio a través de diferentes variadores de frecuencia. En el epígrafe 3.1 se realiza una introducción a los sistemas de control de motores de inducción. En el apartado 3.2 se describen todos los elementos que componen un variador de frecuencia. Finalmente, en los apartados 3.3 a 3.5 se detallan los esquemas del control escalar, vectorial y directo de campo.

El esquema formado por el variador de frecuencia y el motor eléctrico se exponen en la figura 3.1, donde se representa la propia red eléctrica, el variador de frecuencia y el motor de inducción cuyo modelo se ha desarrollado en el capítulo anterior.

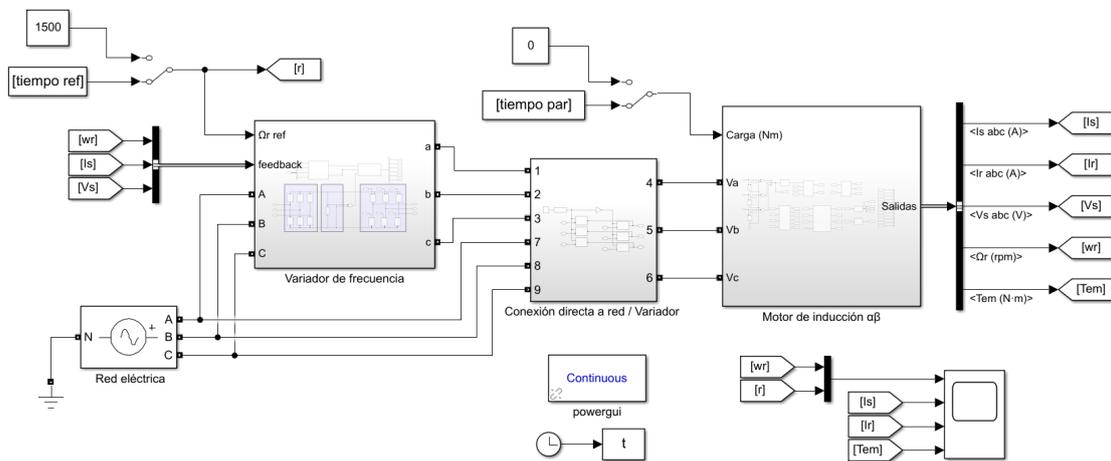


Figura 3.1. Modelo digital del motor de inducción y variador de frecuencia.

Se observa que es posible alimentar el motor mediante conexión directa a red o bien a través del variador de frecuencia del mismo modo que en el sistema real. La conexión de un modo u otro se efectúa mediante el esquema detallado en la figura 3.2.

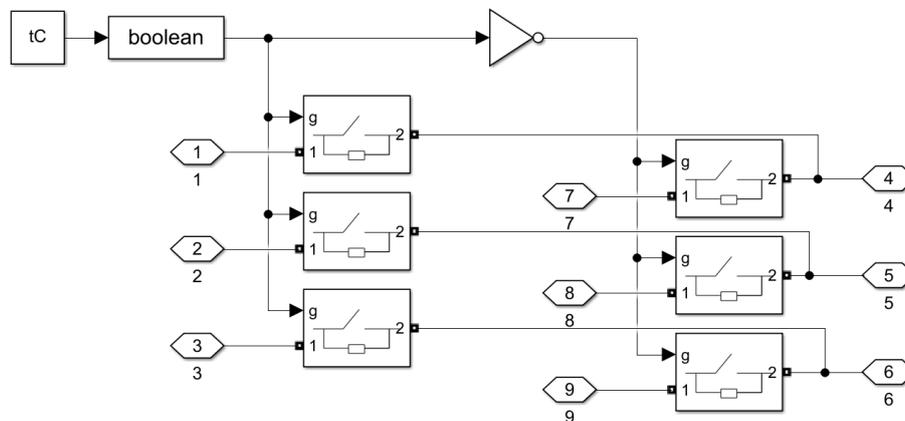


Figura 3.2. Conexión directa a red o variador de frecuencia.

3.1 INTRODUCCIÓN A LAS ESTRATEGIAS DE CONTROL

Combinando las ecuaciones 2.1 y 2.2 se puede llegar a la expresión 3.1 que relaciona la velocidad de giro del rotor con los parámetros que permiten modificar esta velocidad. A partir de esta ecuación se deducen los diferentes métodos mediante los cuales es posible modificar la velocidad en los sistemas de control.

$$(3.1) \quad \Omega_r = \Omega_s \cdot (1 - s) = f_s \cdot \frac{60}{p} \cdot (1 - s)$$

Según la ecuación 3.1, la velocidad mecánica del motor se puede modificar a través de cualquiera de los siguientes tres parámetros: número de par de polos, deslizamiento y frecuencia de sincronismo.

La modificación del número de par de polos lleva asociada una elevada complejidad técnica debido al requerimiento de relés y elementos de corte que permitan modificar la conexión de los devanados de la máquina [2]. Actualmente, esta estrategia ha caído en desuso debido a que únicamente permite ajustar la velocidad a un reducido número de valores discretos de velocidad, generalmente 2 o 3 niveles.

También es posible modificar la velocidad mecánica a través del deslizamiento, lo que implica producir una variación en la curva par-velocidad. Tal como se ha expuesto en el apartado 2.1.2, una opción para conseguir dicha variación, atendiendo a la dependencia entre el par y el cuadrado de la tensión, consiste en incrementar o decrementar la tensión de alimentación del motor, produciendo un desplazamiento vertical en la curva par-velocidad. El principal inconveniente de esta metodología radica en el reducido rango de velocidades respecto del cual podría operar el motor.

Además, es posible implementar una regulación por resistencias externas mediante el conexionado de una carga resistiva trifásica en los devanados del rotor junto con un sistema de conmutación. Esta estrategia tiene dos efectos diferenciados: permite modificar la pendiente de la curva par-velocidad en el rango de velocidades cercanas a la de sincronismo y produce un aumento en el par de arranque del motor. Las principales limitaciones en el empleo de esta estrategia consisten, por una parte, en la accesibilidad a los devanados del rotor, únicamente posible en los motores de rotor bobinado y, además, se produce un aumento en el volumen, peso y pérdidas de potencia debido a la inclusión del sistema trifásico de resistencias y al sistema de conmutación.

El último parámetro modificable en la ecuación 3.1 es la frecuencia de sincronismo. Tal como se ha comentado en el capítulo anterior, el motor de inducción presenta puntos de funcionamiento estables en velocidades de giro cercanas a la velocidad de sincronismo. Es por ello por lo que las estrategias citadas anteriormente únicamente consiguen regular la velocidad angular en un reducido rango de valores. Sin embargo, la modificación en la velocidad de sincronismo proporciona una gran versatilidad y flexibilidad en el control de velocidad.

Los sistemas de control modernos basan su funcionamiento en una variación conjunta de la amplitud y frecuencia de la tensión de alimentación del motor. El componente que se encarga de realizar dicha función es el variador de frecuencia.

3.2 VARIADOR DE FRECUENCIA

Tal como se ha comentado anteriormente, las estrategias actuales de control de motores eléctricos se basan en la variación de la frecuencia asociada a las tensiones de alimentación del motor. Sin embargo, la frecuencia de la tensión trifásica de la red eléctrica es fija, correspondiente a un valor de 50Hz en Europa, por lo que es necesario emplear un dispositivo que sea capaz de controlar tanto el valor eficaz como la frecuencia de las tensiones de entrada del motor. Estas funcionalidades son llevadas a cabo por el variador de frecuencia.

El variador de frecuencia está compuesto por los siguientes elementos:

- Rectificador trifásico no controlado. Un rectificador es un dispositivo electrónico que convierte corriente alterna en corriente continua. Debido al rizado asociado a la señal de salida, es inviable conectar directamente la salida del rectificador a la entrada del inversor, por ello es necesario conectar en paralelo un condensador de alta capacidad que reduce en gran medida el rizado de salida, dando lugar a una onda prácticamente plana.
- Inversor trifásico. Partiendo de la tensión continua proporcionada por el rectificador, el inversor obtiene a su salida un sistema trifásico de tensiones cuyo valor eficaz y frecuencia son controladas por el sistema de control.
- Sistemas de control. Se encarga de generar las consignas de amplitud y frecuencia del inversor trifásico. Se modelarán las estrategias de control escalar, control vectorial y control directo de par.

El modelo del variador de frecuencia compuesto por los elementos citados anteriormente se expone en la figura 3.3.

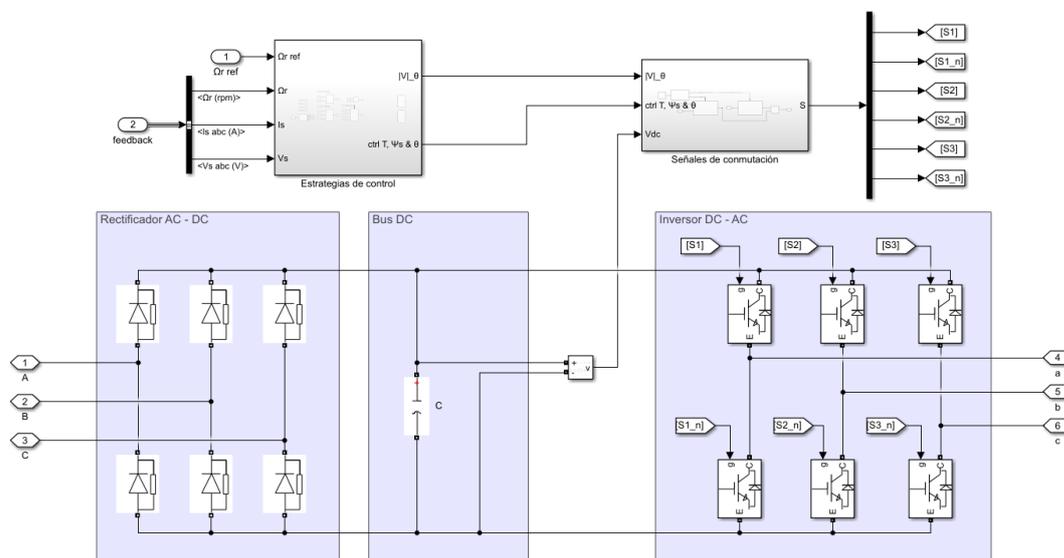


Figura 3.3. Modelo del variador de frecuencia.

Seguidamente, se describirá el elemento principal del hardware del variador de frecuencia: el inversor trifásico.

3.2.1 INVERSOR TRIFÁSICO

El inversor está formado por tres ramas de transistores independientes entre sí a partir de las cuales se genera el sistema trifásico de tensiones que alimentan el motor. La configuración típica de este componente se expone en la figura 3.4.

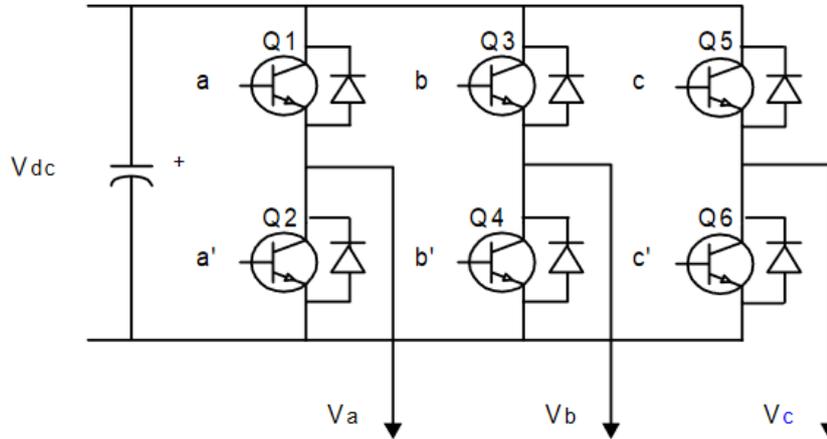


Figura 3.4. Inversor trifásico [11].

En la figura anterior, V_{dc} es la tensión continua proporcionada por el rectificador trifásico. Los seis transistores que componen el inversor se denotan por Q_1 a Q_6 , los cuales están controlados por las señales de conmutación a, a', b, b', c y c' . Finalmente, la salida del inversor se puede expresar en términos de las tensiones de fase V_a, V_b, V_c o bien mediante las tensiones de línea V_{ab}, V_{bc}, V_{ca} .

Las señales de conmutación únicamente pueden tomar dos valores lógicos, de modo que el transistor asociado estará abierto cuando la señal correspondiente se encuentre a nivel bajo, mientras que estará cerrado cuando dicha señal esté a nivel alto.

Además, las señales de conmutación de los transistores Q_2, Q_4 y Q_6 , es decir a', b', c' , no son independientes, sino que tienen una lógica negada respecto de las señales a, b, c . En consecuencia, los estados de las tres señales independientes únicamente pueden tomar $2^3 = 8$ combinaciones diferentes a partir de las cuales se obtiene todas las posibles tensiones de salida del inversor [11].

La relación entre las señales de conmutación independientes y las tensiones de salida de fase y de línea se exponen en la ecuación 3.2.

$$(3.2) \quad \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \begin{bmatrix} V_{ab} \\ V_{bc} \\ V_{ca} \end{bmatrix} = V_{dc} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

De este modo es posible determinar los valores particulares de cada posible salida conocidas las señales de conmutación de los transistores Q_1, Q_3 y Q_5 . Las 8 combinaciones posibles de estos transistores se exponen en la tabla 3.1, en la que cabe destacar que dos del conjunto de combinaciones producen una tensión nula.

a	b	c	V_a/V_{dc}	V_b/V_{dc}	V_c/V_{dc}	V_{ab}/V_{dc}	V_{bc}/V_{dc}	V_{ca}/V_{dc}
0	0	0	0	0	0	0	0	0
1	0	0	2/3	-1/3	-1/3	1	0	-1
1	1	0	1/3	1/3	-2/3	0	1	-1
0	1	0	-1/3	2/3	-1/3	-1	1	0
0	1	1	-2/3	1/3	1/3	-1	0	1
0	0	1	-1/3	-1/3	2/3	0	-1	1
1	0	1	1/3	-2/3	1/3	1	-1	0
1	1	1	0	0	0	0	0	0

Tabla 3.1. Posibles salidas del inversor trifásico.

Sin embargo, las tensiones de la tabla anterior son tensiones continuas. Para poder generar el sistema trifásico de tensiones alternas es necesario emplear técnicas PWM (*Pulse with Modulation*) que permiten obtener tensiones alternas conmutando las tensiones continuas a una elevada frecuencia. Mediante estas técnicas es posible modificar tanto la magnitud de la tensión como su frecuencia, de modo que este tipo de técnicas son ampliamente empleadas en sistemas de control de motores eléctricos.

Existen diferentes técnicas PWM que se emplean a nivel industrial, tales como PWM senoidal, SVPWM, histéresis PWM o inyección de armónicos [12], siendo las más empleadas las dos primeras. Todas estas técnicas tienen un objetivo común: la reducción del contenido de armónicos introducido al motor.

En el presente proyecto, se empleará la técnica SVPWM (*Space Vector Pulse with Modulation*) que, en contraste con la técnica PWM senoidal, proporciona una menor distorsión armónica total en las tensiones de salida y, además, ofrece un uso más eficiente del voltaje del bus de continua V_{dc} .

El objetivo consiste en obtener un vector espacial cuyo módulo y velocidad de giro sean la tensión de referencia y la frecuencia proporcionadas por el sistema de control.

Esta metodología se basa en la definición de 6 vectores espaciales distribuidos de forma equidistante y que están asociados a los 6 estados de la tabla 3.1 cuya salida V_a, V_b, V_c no es nula. La distribución espacial de estos vectores forman un hexágono tal como se presenta en la figura 3.5, donde cabe destacar que las señales de conmutación 100 y 000 proporcionan un vector nulo de salida que se ubica en el propio centro del hexágono.

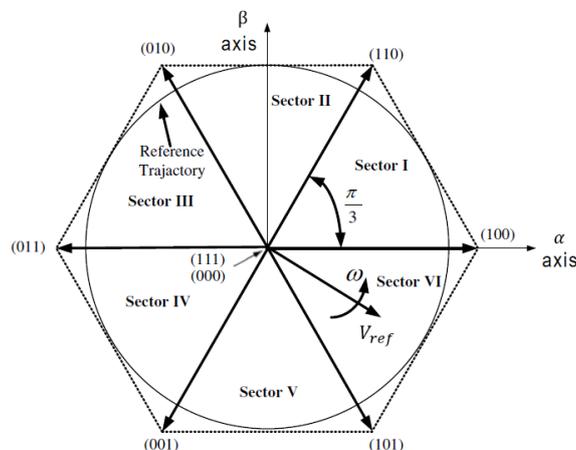


Figura 3.5. Distribución espacial de los vectores en SVPWM [7].

Los valores particulares de los diferentes vectores espaciales que componen el hexágono de la figura anterior, incluyendo los vectores nulos, se exponen en la tabla 3.2. En esta tabla se detallan tanto los valores de línea como de fase, pero en las diferentes simulaciones y debido a su estructura se emplearán los valores de fase.

a	b	c	Vector	Tensión de fase	Tensión de línea
0	0	0	\vec{V}_0	0	0
1	0	0	\vec{V}_1	$2/3 V_{dc} e^{-j0}$	$2/\sqrt{3} V_{dc} e^{j0}$
1	1	0	\vec{V}_2	$2/3 V_{dc} e^{j\pi/3}$	$2/\sqrt{3} V_{dc} e^{j\pi/3}$
0	1	0	\vec{V}_3	$2/3 V_{dc} e^{j2\pi/3}$	$2/\sqrt{3} V_{dc} e^{j2\pi/3}$
0	1	1	\vec{V}_4	$2/3 V_{dc} e^{j\pi}$	$2/\sqrt{3} V_{dc} e^{j\pi}$
0	0	1	\vec{V}_5	$2/3 V_{dc} e^{j4\pi/3}$	$2/\sqrt{3} V_{dc} e^{j4\pi/3}$
1	0	1	\vec{V}_6	$2/3 V_{dc} e^{j5\pi/3}$	$2/\sqrt{3} V_{dc} e^{j5\pi/3}$
1	1	1	\vec{V}_7	0	0

Tabla 3.2. Vectores espaciales técnica SVPWM.

En cada uno de los seis sectores, el vector de referencia \vec{V}_{ref} se obtiene a partir de la combinación lineal de los vectores adyacentes al sector en el que se encuentre en ese instante y del vector nulo conmutando los transistores correspondientes. El valor medio de dicha conmutación es \vec{V}_{ref} y, para el caso del primer sector, viene dado por la ecuación 3.3.

$$(3.3) \quad \vec{V}_{ref} \cdot t_z = \vec{V}_1 \cdot t_a + \vec{V}_2 \cdot t_b + \vec{V}_0 \cdot t_0$$

En la ecuación anterior se tienen los siguientes términos:

- t_a : tiempo de aplicación del vector $\vec{V}_1 = 2/3 V_{dc} e^{j0}$
- t_b : tiempo de aplicación del vector $\vec{V}_2 = 2/3 V_{dc} e^{j\pi/3}$
- t_0 : tiempo de aplicación del vector nulo $\vec{V}_0 = 0$.
- $\vec{V}_{ref} = V_{ref} e^{j\theta}$. Vector de referencia proporcionado por el sistema de control.
- t_z : paso de conmutación. Es el tiempo total dentro del cual se calcula la tensión de referencia y verifica que $t_z = t_a + t_b + t_0$.

El valor de t_z es un parámetro de diseño, mientras que los tiempos t_a y t_b se obtienen de la ecuación 3.3. Sustituyendo en dicha ecuación los vectores espaciales adyacentes al primer sector y proyectando todos los vectores sobre los ejes real e imaginario se obtienen los tiempos de aplicación de cada vector. El desarrollo que permite obtener los tiempos de aplicación de los vectores del primer sector se detalla en 3.4.

Este desarrollo puede tener dos enfoques: es posible obtener t_a y t_b en función del módulo y ángulo de \vec{V}_{ref} o bien a partir de sus componentes real e imaginaria, es decir V_α y V_β . En el presente proyecto se desarrollará el primer enfoque debido a que es más inmediato comparar la tensión de referencia con la tensión máxima que es capaz de proporcionar la técnica SVPWM, tal como se detallará más adelante.

$$\begin{aligned}
V_{ref} e^{j\theta} \cdot t_z &= 2/3 V_{dc} e^{j0} \cdot t_a + 2/3 V_{dc} e^{j\frac{\pi}{3}} \cdot t_b \\
(3.4) \quad V_{ref} \cdot \cos(\theta) \cdot t_z &= \frac{2}{3} V_{dc} \cdot t_b \cdot \sin\left(\frac{\pi}{3}\right) \rightarrow t_b = \sqrt{3} \cdot \frac{V_{ref} \cdot \cos(\theta)}{V_{dc}} \cdot t_z \\
V_{ref} \cdot \sin(\theta) \cdot t_z &= \frac{2}{3} V_{dc} \cdot \left(t_a + t_b \cdot \frac{1}{2}\right) \rightarrow t_a = \frac{V_{ref} \cdot t_z}{2 \cdot V_{dc}} \cdot (3 \cdot \cos(\theta) - \sqrt{3} \cdot \sin(\theta))
\end{aligned}$$

Procediendo de forma análoga para todos los sectores es posible obtener una expresión general que permita determinar los tiempos de aplicación t_a y t_b en función del sector en el que se encuentre el vector de referencia. La expresión 3.5 permite obtener estas variables, además del tiempo de aplicación del vector nulo, en función del sector $k = [1, \dots, 6]$. Hay que tener en cuenta que estas dos ecuaciones son válidas para los sectores impares, de modo que en el caso de los sectores pares es necesario alternarlas.

$$\begin{aligned}
(3.5) \quad t_a &= \frac{V_{ref} \cdot \sqrt{3}}{V_{dc}} \cdot t_z \cdot \sin\left(k \cdot \frac{\pi}{3} - \theta\right) \\
t_b &= \frac{V_{ref} \cdot \sqrt{3}}{V_{dc}} \cdot t_z \cdot \sin\left(\theta - (k-1) \cdot \frac{\pi}{3}\right) \\
t_z &= t_a + t_b + t_0 \rightarrow t_0 = t_z - t_a - t_b
\end{aligned}$$

Es posible determinar el valor máximo del voltaje que la técnica SVPWM es capaz de conseguir [7]. Este voltaje es el radio de la circunferencia inscrita en el hexágono de la figura 3.5, tal como se detalla en la figura 3.6.

$$(3.5) \quad V_{max} = \frac{2}{3} \cdot V_{dc} \cdot \cos\left(\frac{\pi}{6}\right) = \frac{V_{dc}}{\sqrt{3}}$$

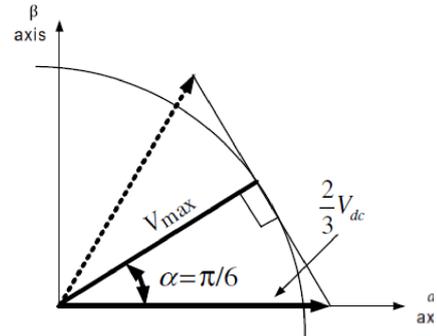


Figura 3.6. Obtención del máximo voltaje de salida en SVPWM [7].

Cabe destacar que el máximo voltaje de salida que es capaz de proporcionar la técnica PWM senoidal (SPWM) es de $\frac{V_{dc}}{2}$, por lo que el empleo del SVPWM supone un incremento del 15.47% en el voltaje de salida. Esto justifica el empleo de esta técnica en el presente proyecto debido a su mejor aprovechamiento de la tensión disponible en el bus de continua.

Además, en el diseño del algoritmo SVPWM y su implementación en Simulink es necesario tener en cuenta esta tensión máxima ya que el módulo de la tensión de referencia no debe sobrepasar ese valor. Habitualmente, en la determinación de los tiempos de aplicación de cada vector se suele definir una variable auxiliar denotada por $mag = V_{ref} \cdot \sqrt{3}/V_{dc}$. Empleando la ecuación 3.5 se evidencia que el máximo valor que puede tomar esta variable auxiliar es 1.

Finalmente, una vez calculados los tiempos anteriores, el último paso consiste en obtener las señales de conmutación a , b y c que controlan los transistores del inversor. Para ello, dentro de cada paso de conmutación t_z se deben de aplicar una serie de secuencias que permiten obtener el vector de referencia. Estos patrones de conmutación se detallan en la figura 3.7.

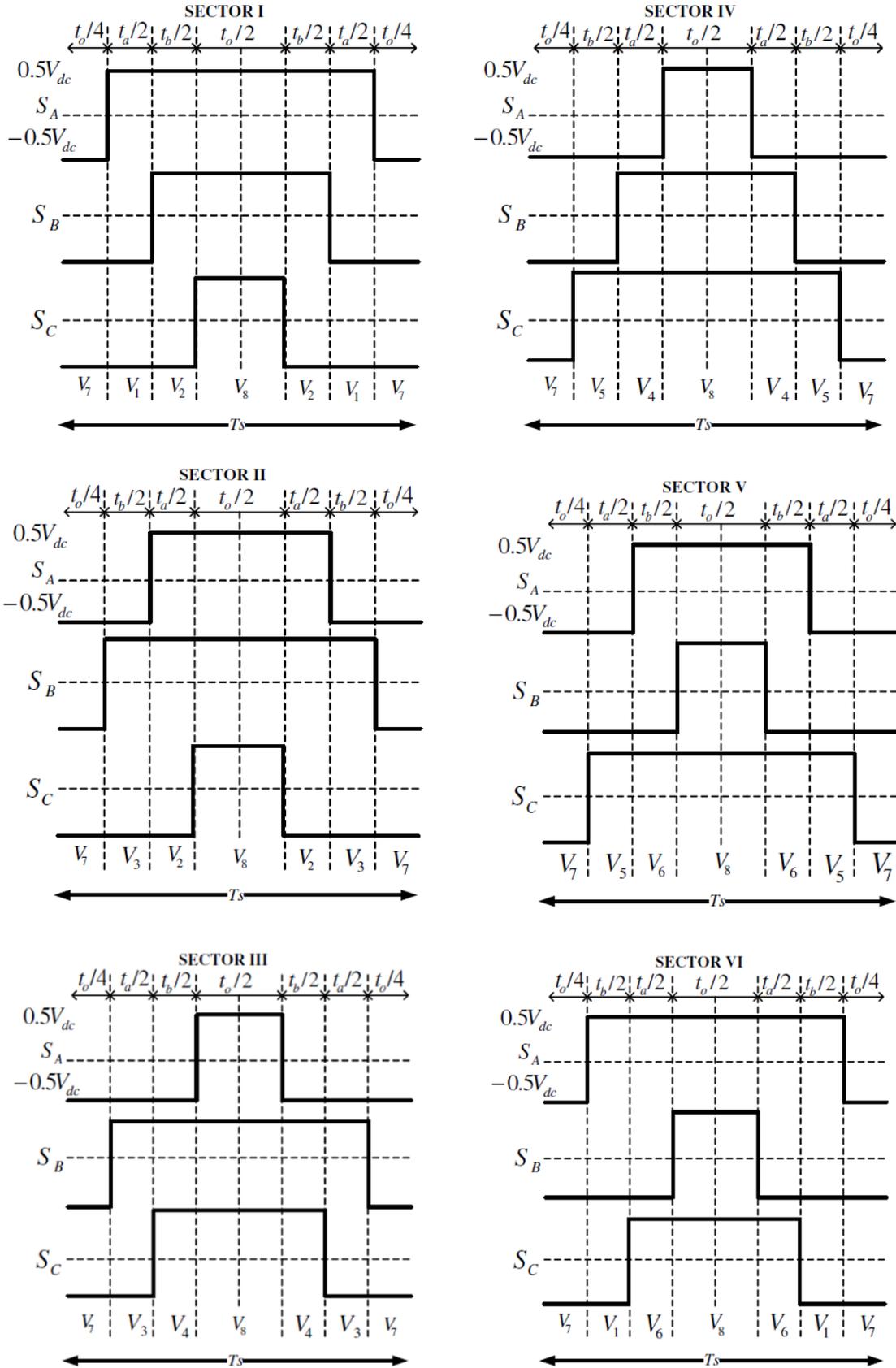


Figura 3.7. Patrones de conmutación SVPWM [7].

En cuanto a la implementación de esta técnica en Simulink, se ha desarrollado un conjunto de esquemas que, a partir del módulo y ángulo de la tensión de referencia proporcionada por el sistema de control, calculan las señales de conmutación necesarias.

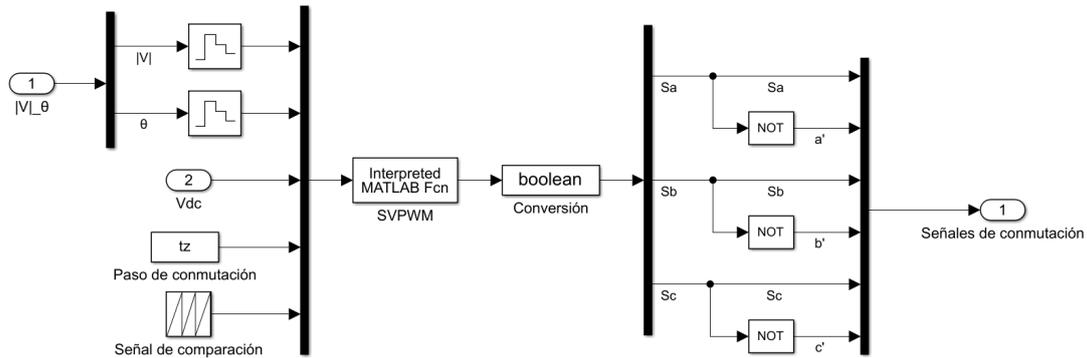


Figura 3.8. Implementación de la técnica SVPWM.

El cálculo de las señales a' , b' , c' se han obtenido mediante el diseño del diagrama de Simulink detallado en la figura 3.8, mientras que el cálculo de los tiempos de aplicación de cada vector y la selección del patrón de conmutación adecuado, es decir, las señales a, b, c , se han implementado mediante una función de Matlab, cuyo código desarrollado se expone en [42]. En el anexo 1, además, se detallan las ecuaciones que se deberían de emplear en el caso de que se desee diseñar el algoritmo a partir de las componentes α, β del vector de referencia.

El algoritmo implementado ha sido validado comparando la salida proporcionada por el conjunto de señales de conmutación mediante el esquema de la figura 3.9 y empleando el bloque SVPWM disponible en la librería Simscape. Para poder comparar de forma correcta las señales obtenidas en ambos métodos, hay que tener en cuenta que la entrada de este bloque SVPWM no es directamente el módulo de la tensión de referencia, sino que previamente hay que multiplicarlo por $\sqrt{3}/V_{dc}$, es decir, la entrada de este bloque es la variable mag , permitiendo al algoritmo verificar si la magnitud de la tensión de referencia supera el valor umbral simplemente comprobando si la variable mag es superior a 1. Cabe destacar que, de hecho, el bloque SVPWM de la librería Simscape no toma como entrada la tensión V_{dc} . La validación del algoritmo implementado se expone en la figura 3.9.

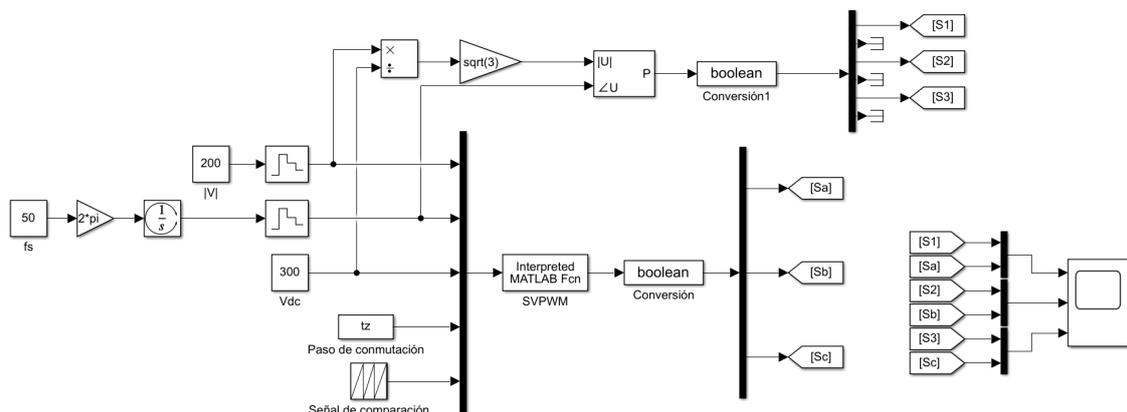


Figura 3.9. Validación de la técnica SVPWM.

La estrategia SVPWM basada en las componentes α, β también ha sido diseñada y validada en Simulink de forma análoga al caso anterior tal como se detalla en el anexo 1.

A partir de todos los modelos digitales realizados hasta ahora, es decir, el motor de inducción y el variador de frecuencia, es posible realizar simulaciones del motor en bucle abierto, permitiendo validar la correcta implementación de los diferentes diseños. En particular, se ensaya el motor a unas frecuencias de 50Hz y 33.33Hz y se aplica un par mecánico de 10Nm en el instante 2s. Los resultados se exponen en la figura 3.10.

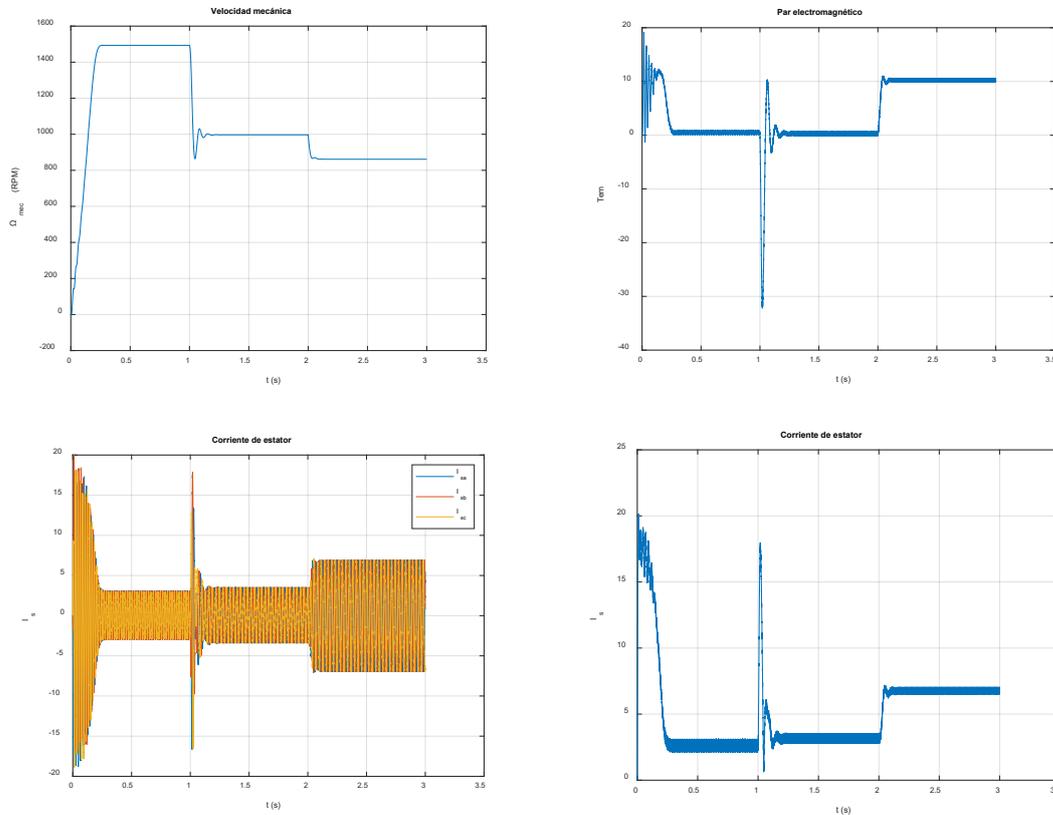


Figura 3.10. Resultados motor en bucle abierto.

En efecto, los resultados son los esperados. Al reducir la frecuencia del sistema trifásico de tensiones de alimentación del motor eléctrico se reduce la velocidad mecánica del motor y apenas se ven modificados el nivel de corriente de estator y el par electromagnético. Por otra parte, cuando se incrementa el par de la carga se reduce la velocidad de giro y aumenta notablemente el nivel de corriente consumida.

Cabe destacar que en el caso del control DTC se emplea un patrón diferente al definido por la estrategia SVPWM. Por ello, en función de qué estrategia de control se seleccione, la forma de comandar los transistores debe ser diferente. La selección de una técnica u otra se efectúa tal como se expone en la figura 3.11. En el apartado 3.5 se detalla el comportamiento interno del bloque que computa las secuencias de conmutación del control DTC.

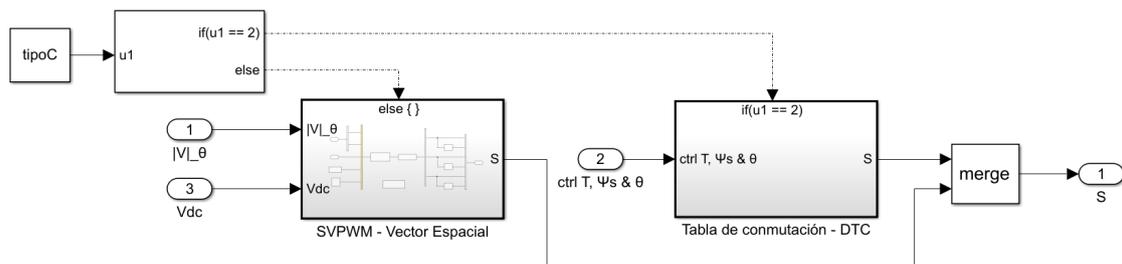


Figura 3.11. Selección del patrón de conmutación.

3.2.2 SISTEMA DE CONTROL

El objetivo de este capítulo consiste en integrar todos los sistemas de control, es decir, los controles escalar, vectorial y directo de par, en un mismo diagrama de Simulink. Para ello se ha definido una variable auxiliar mediante la cual se ejecuta una estrategia de control u otra en función del valor que tome esta. La selección del sistema de control deseado se detalla en la figura 3.12.

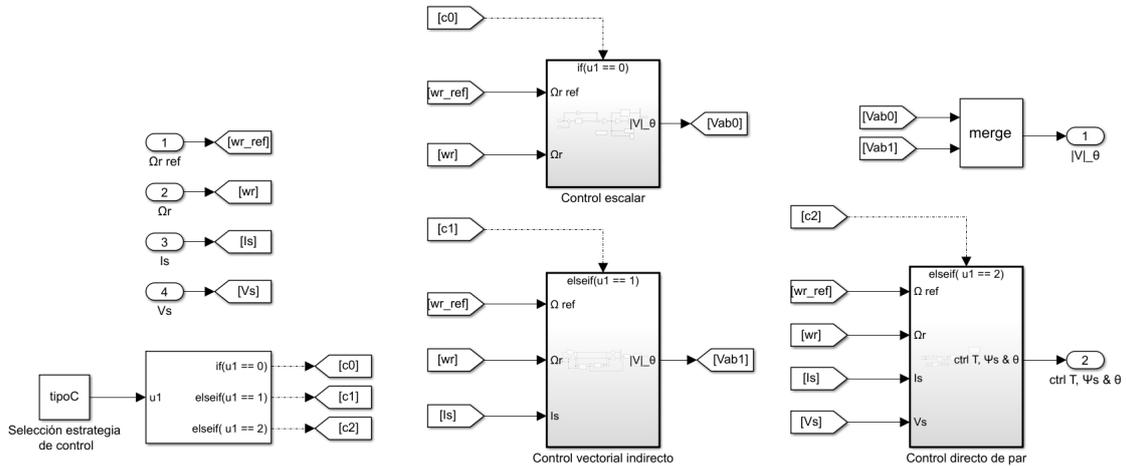


Figura 3.12. Selección de la estrategia de control.

3.3 CONTROL ESCALAR

También es conocido como control V/f y es la estrategia de control que más extendido tiene su uso en la industria. Consiste en proponer una frecuencia de alimentación de las fases del motor mediante un controlador P o PI y, a partir del valor de frecuencia propuesto, obtener un valor de tensión que consiga mantener la ratio V/f constante.

Permite que el motor opere en dos zonas de funcionamiento: si se mantiene la ratio V/f constante el motor opera a flujo constante, mientras que, si se incrementa la frecuencia manteniendo el voltaje a un valor fijo, entonces opera a potencia constante.

Seguidamente, se procede a exponer el principio de funcionamiento del control escalar, justificando matemáticamente cómo es posible mantener constante el flujo magnético de entrehierro a partir del ratio tensión-frecuencia.

A partir de la expresión temporal del flujo magnético, es posible obtener la relación entre la amplitud o valor eficaz de la fuerza electromotriz y el flujo magnético.

$$(3.6) \quad E = \frac{2\pi}{\sqrt{2}} \cdot f_{est} \cdot \psi = k \cdot f_{est} \cdot \psi, \quad k = cte$$

Si se simplifica el circuito eléctrico equivalente tal como se ha expuesto en el capítulo anterior, entonces se puede afirmar que la tensión de alimentación del estator es aproximadamente igual a la fuerza electromotriz inducida, verificándose la expresión 3.7.

$$(3.7) \quad V \approx E = k \cdot f_{est} \cdot \psi \rightarrow \psi \approx \frac{V}{k \cdot f_{est}}$$

En consecuencia, si se mantiene constante la ratio V/f , se mantendrá constante el flujo magnético y, además, es posible demostrar que también se mantiene invariable el par magnético.

La aproximación citada anteriormente es válida en un amplio rango excepto a bajas velocidades, donde se produce una reducción tanto del flujo como del par electromagnético. Es por ello por lo que se emplea una tensión adicional a bajas velocidades con el objetivo de compensar dicha caída de flujo.

Por otra parte, no es posible incrementar la tensión de forma indefinida ya que no se debe sobrepasar la tensión nominal, dando lugar a que no es posible mantener la ratio V/f constante a frecuencias superiores a la nominal. En consecuencia, a frecuencias superiores a la nominal el motor opera en lo que se conoce por zona de debilitamiento de campo, en la que se produce una reducción del par electromagnético proporcionado por el motor y se mantiene constante su potencia generada.

Los dos efectos citados anteriormente, es decir, el incremento del voltaje a bajas frecuencias y la inclusión de la zona de debilitamiento de campo se exponen en la figura 3.13.

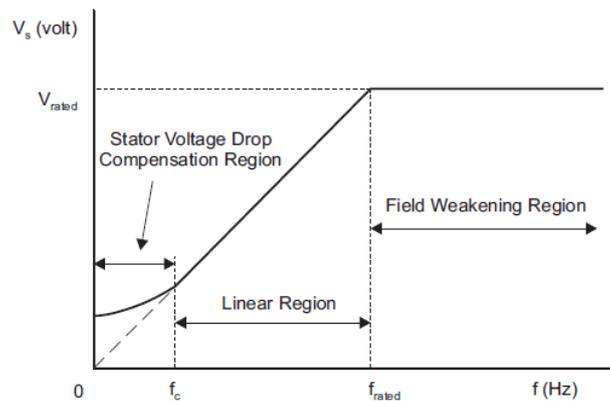


Figura 3.13. Curva V-f [13].

La construcción del esquema de control escalar parte de la relación existente entre la velocidad mecánica y la frecuencia de sincronismo cuando el motor se encuentra en vacío, donde el deslizamiento es nulo. De este modo, a partir de la ecuación 3.1 se deduce la ecuación 3.7 válida exclusivamente cuando el motor opera en vacío.

$$(3.7) \quad \Omega_r = \Omega_s = f_s \cdot \frac{60}{p} \rightarrow f_s = \Omega_r \cdot \frac{p}{60} = \Omega_{ref} \cdot \frac{p}{60}$$

En consecuencia, es posible establecer un sistema de control en bucle abierto que permite controlar la velocidad de giro del motor de inducción cuando éste se encuentra en vacío. En este caso, a partir de la velocidad de referencia, Ω_{ref} , es posible determinar cuál debe ser la frecuencia de las tensiones de alimentación del motor.

Sin embargo, la ecuación anterior no es válida cuando se conecta una carga al motor. En efecto, si se alimenta el motor con la frecuencia propuesta por la ecuación 3.7, la velocidad real sería inferior a la de referencia, por lo que es necesario incrementar la frecuencia propuesta por la ecuación anterior. Esto se puede lograr mediante el empleo de un regulador P o PI que se encargue de estimar el incremento entre la frecuencia de sincronismo necesaria y la propuesta por la ecuación 3.7 a partir del error entre la velocidad de referencia y la real.

En la figura 3.14 se representa de forma esquemática los diferentes elementos que componen el sistema de control escalar. En primer lugar, se efectúa la lectura de la velocidad mecánica del motor eléctrico a través de un encoder con el objetivo de calcular el error entre la referencia y la velocidad real. A partir de dicho error y la frecuencia de sincronismo que se debería aplicar en bucle abierto se determina la frecuencia de alimentación del sistema trifásico de tensiones.

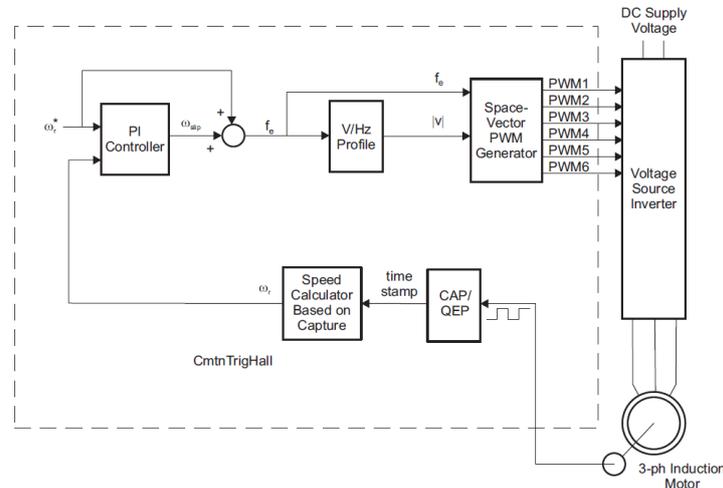


Figura 3.14. Esquema general del control escalar [13].

Tal como se ha comentado en el apartado anterior, la técnica de modulación basada en el vector espacial requiere como entradas el módulo y el ángulo de la tensión de referencia, por ello el esquema de Simulink que se ha propuesto para diseñar el control escalar obtiene como salidas las citadas componentes, tal como se expone en la figura 3.15.

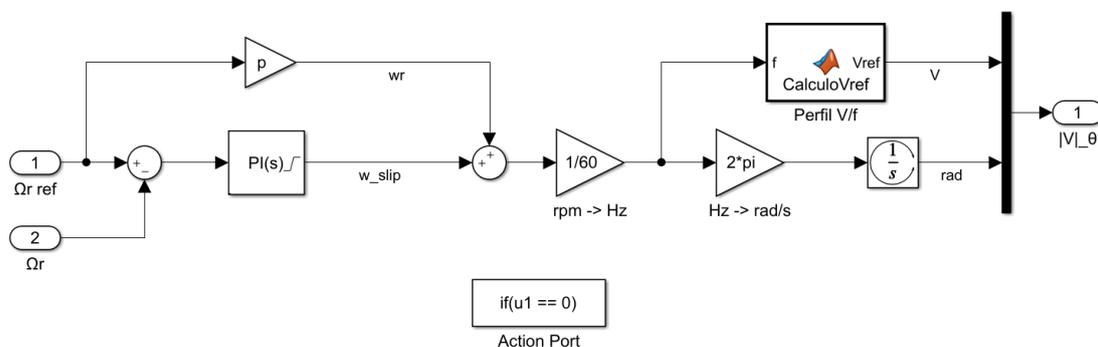


Figura 3.15. Control escalar.

La sintonización del PID consiste en determinar los valores numéricos de la parte proporcional e integral del regulador, es decir, las variables k_p y k_i . En este caso existen dos efectos contrapuestos:

- Valores pequeños de k_p y k_i presentan un buen seguimiento de la referencia cuando el motor opera en vacío y se producen cambios en la consigna de velocidad. Sin embargo, cuando se producen cambios en la carga, la compensación de la caída de velocidad que experimenta el motor presenta una dinámica relativamente lenta.
- Valores grandes de k_p y k_i implican la aparición de ciertas oscilaciones en la respuesta de la velocidad cuando se producen cambios en la referencia, pero cuando se mantiene

la consigna de velocidad fija y se modifica la carga, el rechazo de esta perturbación lleva asociado un transitorio más rápido.

En cuanto a su implementación, es necesario tener en cuenta ciertas consideraciones prácticas. Ha sido necesario saturar la salida del PID con el objetivo de evitar que este proponga un valor de w_{slip} excesivamente grande especialmente cuando se producen cambios bruscos de velocidad. Se ha fijado una saturación en w_{slip} de $\pm 150 \cdot p \text{ rpm}$. Cabe destacar que la variable w_{slip} está expresada en revoluciones eléctricas, no mecánicas, a diferencia de como ocurre en la variable Ω_{slip} que satura en $\pm 150 \text{ rpm}$.

El valor anterior se puede considerar un parámetro de diseño, cuyo valor depende del nivel de carga en el que va a operar el motor. Si se emplea un valor pequeño puede dar lugar a un incorrecto seguimiento de la referencia a grandes cargas, mientras que un valor elevado puede dar lugar a oscilaciones en la evolución de la velocidad.

Durante las diferentes pruebas se ha evidenciado que, cuando se produce un cambio en escalón de la velocidad de referencia entre valores relativamente distantes entre sí, la variable w_{slip} satura rápidamente al valor correspondiente. Esto produce que la acción integral del PID continúe integrando el error aunque su salida esté saturada, generando un efecto indeseado. Debido a esto es necesario activar el *antiwindup* para evitar la integración del error cuando la salida del PID esté saturada. En la figura 3.16 se analiza el seguimiento de la referencia mediante el empleo del *antiwindup*.

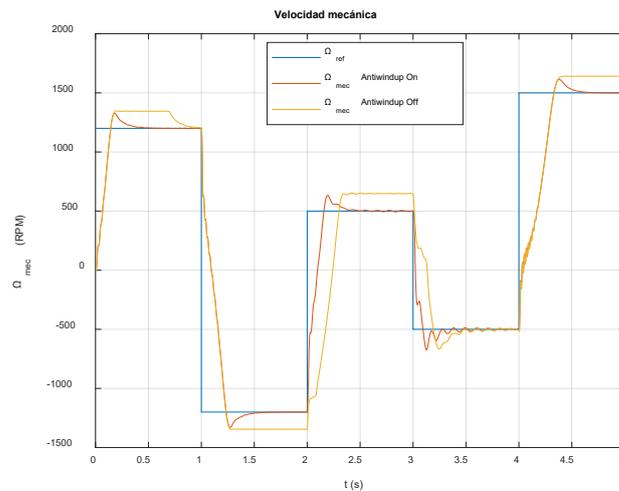


Figura 3.16. Efecto antiwindup.

En la figura anterior se han realizado cambios en la referencia cada segundo, aplicando el conjunto de valores formado por $[1200, -1200, 500, -500, 1500] \text{ rpm}$. Por una parte, se observa que la variable Ω_{slip} satura en $\pm 150 \text{ rpm}$ según el tramo que se analice, manteniéndose durante un cierto intervalo de tiempo en el valor de saturación cuando no está activado el *antiwindup*. Esto puede llegar a producir un incorrecto seguimiento de la referencia en el tiempo deseado. Sin embargo, cuando está activado, la acción integral deja de actuar cuando w_{slip} satura, dando lugar a una reducción inmediata de su magnitud cuando el error entre la velocidad mecánica y la de referencia cambia de signo. Otra posibilidad para mitigar este efecto consiste en realizar cambios en la referencia mediante rampas, evitando los cambios en escalón.

Con el fin de validar el sistema de control escalar se han realizado dos simulaciones: aplicando cambios en la referencia en escalón manteniendo el motor en vacío y efectuando cambios en la carga manteniendo el motor a una referencia fija. Estos resultados se exponen en la figura 3.17.

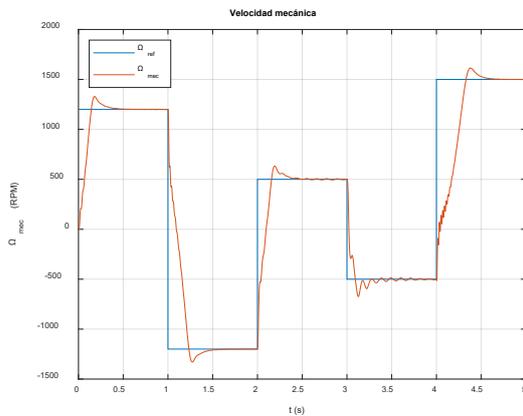


Figura 3.17a. Prueba de velocidad - Ω_r .

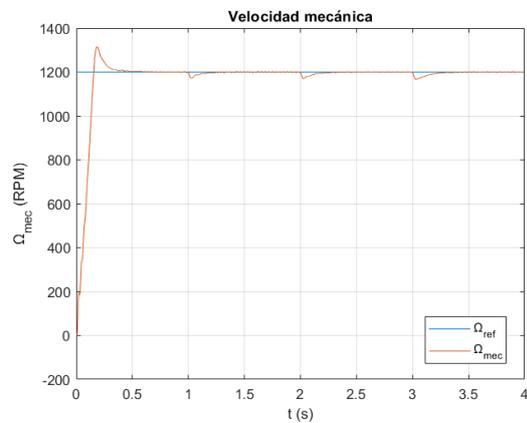


Figura 3.17d. Prueba de par - Ω_r .

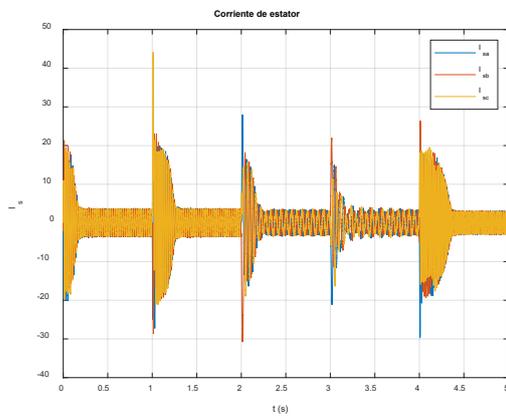


Figura 3.17b. Prueba de velocidad - I_s .

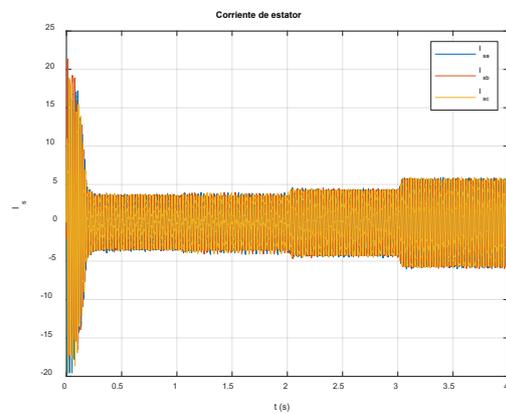


Figura 3.17e. Prueba de par - I_s .

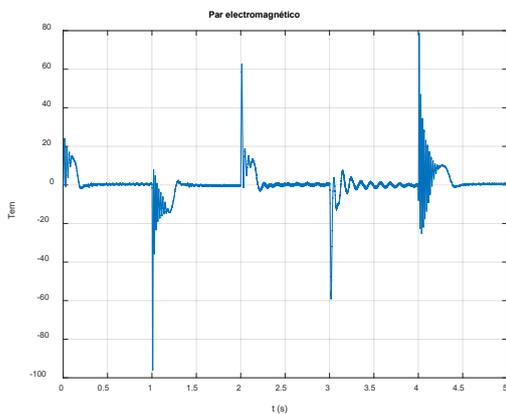


Figura 3.17c. Prueba de velocidad - T_{em} .

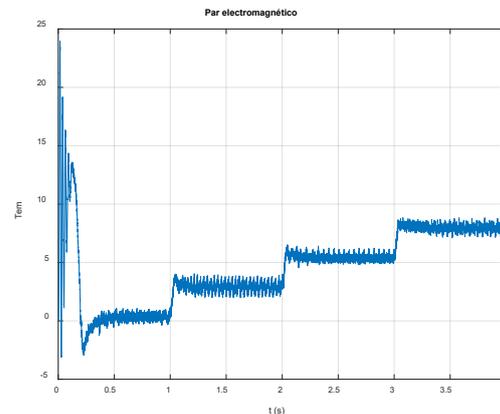


Figura 3.17f. Prueba de par - T_{em} .

Figura 3.17. Resultados control escalar.

En primer lugar, en la figura 3.17a se evidencia el correcto seguimiento de las referencias, verificando la correcta inversión de giro cuando se introduce una referencia negativa. Además, en la figura 3.17b se observa que, en régimen permanente, el par electromagnético se mantiene en un valor prácticamente nulo, sin tomar un valor de 0Nm debido a la fricción mecánica. Finalmente, debido a que la carga se mantiene inalterada, el nivel de corriente consumida por el motor se mantiene constante, tal como se detalla en la figura 3.17c.

En el último ensayo se ha mantenido una referencia fija de 1200 rpm, modificando la carga entre los valores [0, 2.5, 5, 7.5] Nm a intervalos regulares. Se observa un correcto seguimiento de la referencia en la figura 3.17d, compensando la caída de velocidad producida por el aumento de la carga. Además, se observa que a medida que se incrementa la carga, aumenta el par electromagnético y el nivel de corriente consumida según las figuras 3.17e y 3.17f.

3.4 CONTROL VECTORIAL

Empleando las transformaciones de Clarke y Park para definir un sistema de referencia d, q que gire a la velocidad de sincronismo dada por las tensiones de alimentación y que además el eje d esté alineado con el flujo magnético del rotor. De este modo se cumple que $\psi_r = \psi_{rd}$ y $\psi_{rq} = 0$, por lo que la ecuación del par electromagnético, despreciando las pérdidas en el hierro, se puede expresar según 3.8. Cabe destacar que el modelo matemático del motor en ejes d, q se detalla en el anexo 1.

$$(3.8) \quad \begin{aligned} T_{em} &= \frac{3}{2} \cdot p \cdot \frac{L_m}{L_{lr} + L_m} \cdot (i_{sq} \cdot \psi_{rd} - i_{sd} \cdot \psi_{rq}) = \\ &= \frac{3}{2} \cdot p \cdot \frac{L_m}{L_{lr} + L_m} \cdot i_{sq} \cdot \psi_{rd} = k \cdot \psi_r \cdot i_{sq} \end{aligned}$$

En la ecuación anterior se puede observar que, si el flujo del rotor se mantiene constante, el par electromagnético es proporcional a la componente q de la corriente del estator.

Seguidamente, se desarrolla la relación existente entre la componente d de la corriente del estator, i_{sd} , y el flujo del rotor, ψ_r . Empleando el operador diferencial $s = d/dt$, la ecuación correspondiente a la caída de tensión del rotor se puede expresar como:

$$(3.9) \quad \frac{d\psi_{rd}}{dt} = s \cdot \psi_{rd} = -i_{rd} \cdot R_r - (w_s - w_r) \cdot \psi_{rq}$$

Teniendo en cuenta que, orientando el eje d del sistema de referencia giratorio con el flujo del rotor, entonces se cumple $\psi_r = \psi_{rd}$ y $\psi_{rq} = 0$ y, por tanto:

$$(3.10) \quad s \cdot \psi_r = -i_{rd} \cdot R_r \rightarrow i_{rd} = -s \cdot \frac{\psi_r}{R_r}$$

Sustituyendo la expresión anterior en la ecuación 1.2 del anexo 1, que relaciona la componente d de enlaces de flujo del rotor en función de las corrientes, se obtiene:

$$(3.11) \quad \begin{aligned} \psi_{rd} = \psi_r &= i_{rd} \cdot (L_m + L_{lr}) + i_{sd} \cdot L_m = -s \cdot \frac{\psi_r}{R_r} \cdot L_r + i_{sd} \cdot L_m \\ \psi_r &= \frac{L_m}{1 + s \cdot \frac{L_r}{R_r}} \cdot i_{sd} = \frac{L_m}{1 + s \cdot \tau_r} \cdot i_{sd}, \quad \tau_r = \frac{L_r}{R_r} = \frac{L_m + L_{lr}}{R_r} \end{aligned}$$

Mediante la ecuación anterior, es posible observar que el flujo magnético del rotor es proporcional a la componente i_{sd} con un cierto retardo dado por la constante de tiempo τ_r asociada a un sistema de primer orden.

De este modo, es posible controlar ψ_r a través de i_{sd} . Si se mantiene constante i_{sd} , entonces el flujo del rotor vendrá dado por la ecuación 3.12 dado que el operador de Laplace es nulo en régimen permanente.

$$(3.12) \quad \psi_r = L_m \cdot i_{sd}$$

En consecuencia, mediante la regulación de las corrientes i_{sd} e i_{sq} en un sistema de referencia giratorio cuya velocidad es la de sincronismo y esté alineado con el flujo del rotor, es posible controlar de forma independiente tanto el par como el flujo. Es por ello por lo que es necesario mantener las componentes dq de las corrientes de estator en unos valores deseados. De esta idea parte el regulador de corrientes, cuyo esquema se muestra en la figura 3.18.

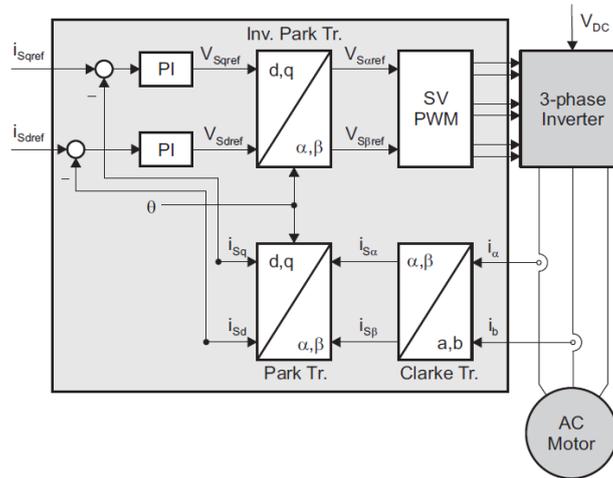


Figura 3.18. Regulador de corrientes [14].

Habitualmente, la referencia de flujo ψ_r , y por tanto, la referencia de i_{sd} se mantiene a un valor constante excepto a velocidades superiores a la nominal, en cuyo caso la referencia de flujo se reduce de forma inversamente proporcional a la velocidad. En consecuencia, es necesario obtener el valor nominal de ψ_r a partir del cual se calcula el valor nominal de i_{sd} empleando la ecuación 3.12. Como alternativa, es posible simular el modelo digital del motor de inducción implementado en el capítulo 1 operando bajo condiciones nominales y analizar los resultados de la simulación con el objetivo de determinar los valores nominales de dichas variables.

Se ha optado por esta segunda opción, por lo que los valores nominales de ψ_r y i_{sd} que, a su vez, son el flujo del rotor y la componente d de la corriente de referencia del control vectorial son: $\psi_{r,ref} = 0.975 \text{ Wb}$ y $i_{sd,ref} = 4.5 \text{ A}$. Se observa que, en efecto, la relación entre ambos valores es el parámetro L_m identificado en el capítulo 1. El cómputo de $i_{sd,ref}$ se efectúa en función de la velocidad mecánica del rotor tal como se expone en la figura 3.19.

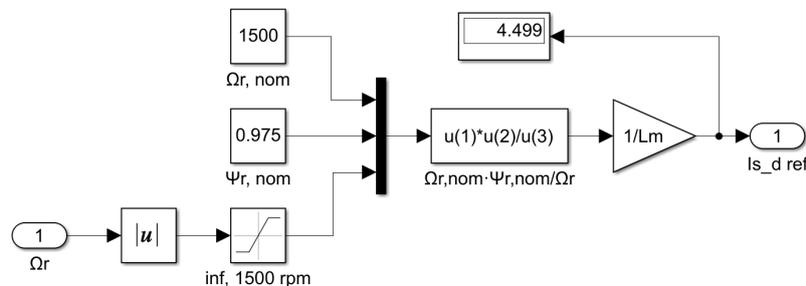


Figura 3.19. Cómputo de $i_{sd,ref}$.

Para realizar la transformación de Park de tal modo que el eje d esté alineado con el flujo del rotor, es necesario conocer la posición angular del mismo. En función de cómo se obtenga la información asociada a dicho ángulo, el control vectorial se divide en dos estrategias: control

vectorial directo e indirecto. En el control vectorial directo, la posición del flujo magnético se obtiene a partir de la lectura o estimación de este, mientras que el control vectorial indirecto emplea la velocidad mecánica y de deslizamiento para determinar la posición del flujo.

3.4.1 CONTROL VECTORIAL DIRECTO

Tal como se ha expuesto anteriormente, en el control vectorial directo, la posición angular del flujo se obtiene directamente mediante la lectura o estimación de este, tal como se expone a continuación:

$$(3.13) \quad \theta_{\psi_r} = \text{atan2} \left(\frac{\psi_{r\beta}}{\psi_{r\alpha}} \right)$$

Debido a que en el banco de ensayos no se encuentra implementado el control vectorial directo, únicamente se integrará en las simulaciones el control vectorial indirecto.

3.4.2 CONTROL VECTORIAL INDIRECTO

Debido a la dificultad asociada a la lectura del flujo magnético, se implementa el control vectorial indirecto que, a partir de las corrientes de estator, es capaz de obtener la posición del flujo magnético del rotor.

Teniendo en cuenta que $\psi_{rq} = 0$, se cumple que:

$$(3.14) \quad \frac{d\psi_{rq}}{dt} = -i_{rq} \cdot R_r + (w_s - w_r) \cdot \psi_{rd} = 0$$

Seguidamente, empleando la ecuación 3.14 se define la velocidad angular de deslizamiento tal como se expone a continuación:

$$(3.15) \quad w_{sl} = w_s - w_r = R_r \cdot \frac{i_{rq}}{\psi_{rd}}$$

A partir de la ecuación 3.16 se establece una relación entre las componentes q de las corrientes de estator y rotor.

$$(3.16) \quad \psi_{rq} = (i_{sq} + i_{rq}) \cdot L_m + i_{rq} \cdot L_{lr} = i_{sq} \cdot L_m + i_{rq} \cdot L_r = 0 \rightarrow i_{rq} = -i_{sq} \cdot \frac{L_m}{L_r}$$

Sustituyendo 3.16 en la ecuación 3.15 y empleando la ecuación 3.11, se obtiene:

$$(3.17) \quad w_{sl} = -\frac{R_r \cdot L_m}{L_r} \cdot \frac{i_{sq}}{\psi_{rd}}$$

$$\psi_{rd} = \psi_r = \frac{L_m}{1 + s \cdot \frac{L_r}{R_r}} \cdot i_{sd}$$

$$w_{slip} = -\frac{R_r}{L_r} \cdot \frac{i_{sq}}{i_{sd}} \cdot \left(1 + s \cdot \frac{L_r}{R_r} \right) = -\frac{i_{sq}}{i_{sd}} \cdot \left(\frac{R_r}{L_r} + s \right)$$

El cómputo del ángulo asociado al flujo del rotor se efectúa tal como se detalla en la figura 3.20, donde w_{slip} se ha calculado en régimen permanente, es decir, sustituyendo $s = 0$.

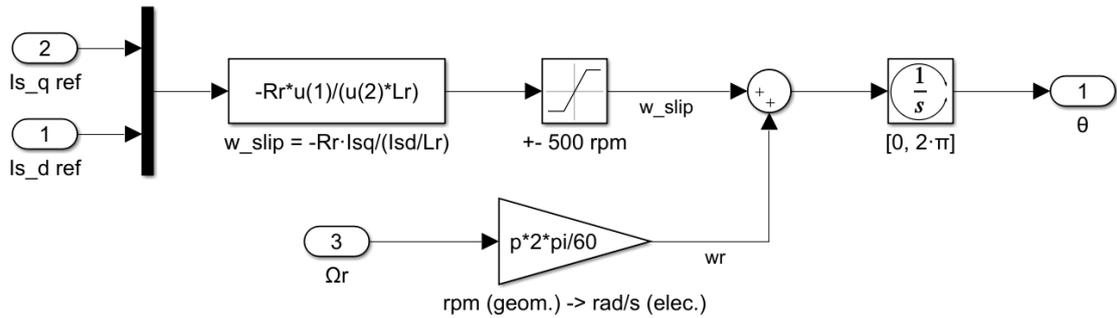


Figura 3.20. Cómputo de w_{slip} y θ .

Tal como se ha comentado en el epígrafe 3.2.1, la técnica SVPWM implementada parte de la magnitud y el ángulo de la tensión de referencia, por lo que será necesario realizar la conversión necesaria para obtener estas variables a partir de las componentes α, β de dicha tensión. La magnitud y ángulo del vector espacial de referencia se efectúan tal como se detalla en la figura 3.21.

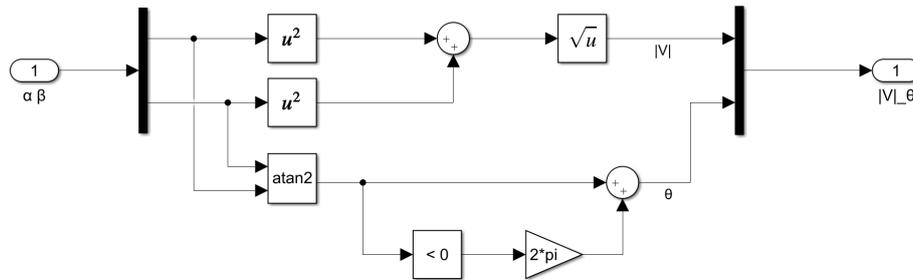


Figura 3.21. Cómputo de la magnitud y ángulo de la tensión de referencia.

Finalmente, los diferentes bloques comentados anteriormente se han integrado en el esquema de la figura 3.22.

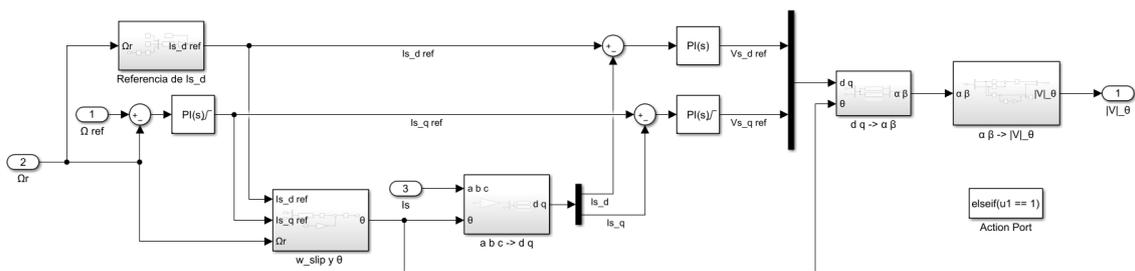


Figura 3.22. Control vectorial indirecto.

Los resultados del control vectorial se exponen en la figura 3.23, donde se evidencia un correcto seguimiento de las referencias y compensación del incremento del par resistente. El análisis de la corriente consumida y el par electromagnético es análogo al realizado en el control escalar, por lo que únicamente se representará la velocidad mecánica.

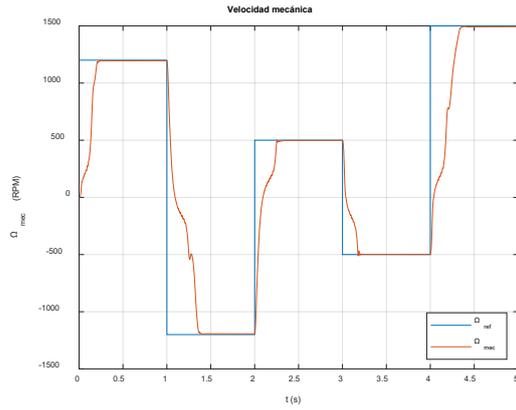


Figura 3.23a. Prueba de velocidad.

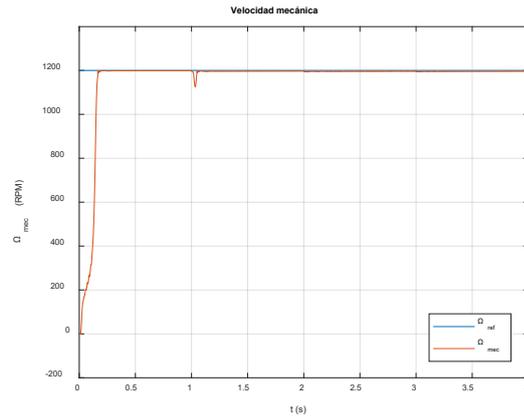


Figura 3.23b. Prueba de par.

Figura 3.23. Resultados control vectorial.

3.5 CONTROL DIRECTO DE PAR

Este tipo de control basa su principio de funcionamiento en la posibilidad de controlar de forma independiente el par y el flujo, pero a diferencia del control vectorial, el control directo de par o DTC no requiere de controladores de corriente en su esquema de control ya que efectúan un control directo del flujo y del par [7] y, además, no emplea ninguna transformación en ejes d, q .

Como alternativa a los reguladores de corriente, se emplean controladores de histéresis cuyas salidas se emplean para determinar qué estados del inversor deben estar activos. El flujo se puede estimar a partir de las ecuaciones 2.6, tal como se describe en las ecuaciones siguientes:

$$(3.18) \quad \psi_{s\alpha} = \int v_{s\alpha} - i_{s\alpha} \cdot R_s, \quad \psi_{s\beta} = \int v_{s\beta} - i_{s\beta} \cdot R_s$$

Si se desprecia la caída de tensión resistiva, se evidencia que la variación del flujo es proporcional al nivel de tensión de entrada al motor. Además, es posible demostrar que la constante de tiempo del flujo de estator es notablemente inferior a la del rotor, por lo que cambios en la tensión producirán una rápida variación en el flujo de estator, dando lugar a una rápida variación del ángulo entre los flujos de estator y rotor. Al modificar instantáneamente este ángulo, se controla instantáneamente el par electromagnético, siempre que el error entre la consigna del flujo de estator y su valor real se encuentren dentro de la banda de histéresis que se defina.

De entre las diferentes ecuaciones a partir de las que es posible calcular el par [8], la ecuación 3.20 expresa esta magnitud en función de los flujos de estator y rotor y el ángulo que forman estos. Se evidencia que, si se mantiene constante el flujo de estator a través del correspondiente regulador de histéresis, se puede controlar instantáneamente el par debido a la elevada constante de tiempo del flujo de rotor.

$$(3.20) \quad T_{em} = \frac{3}{2} \cdot p \cdot \frac{L_m}{\sigma \cdot L_s \cdot L_r} \cdot |\psi_s| \cdot |\psi_r| \cdot \sin(\alpha)$$

Partiendo de estas ideas, se construye el esquema de control del DTC, el cual se expone en la figura 3.24. Los controladores de histéresis de flujo y par requieren el conocimiento del valor real de estas dos variables. Por este motivo, el primer paso consiste en estimar estas magnitudes.

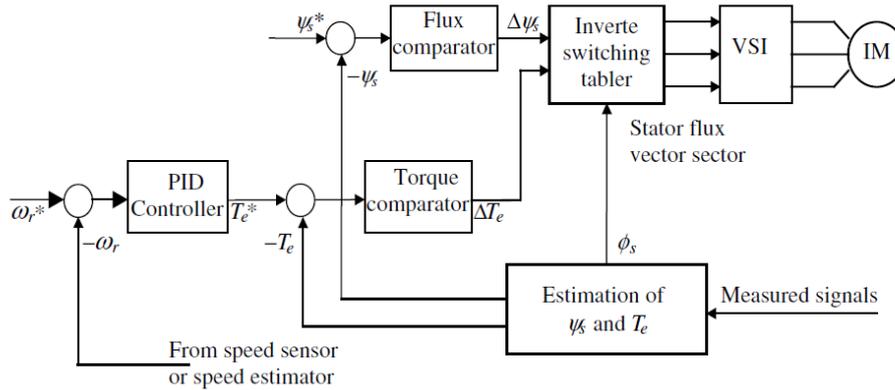


Figura 3.24. Esquema general del control DTC [7].

La estimación del flujo se efectúa a partir de las expresiones 3.18 y 3.19, mientras que el par es estimado a partir de la ecuación 3.21.

$$(3.21) \quad T_{em} = \frac{3}{2} \cdot p \cdot (\psi_{s\alpha} \cdot i_{s\beta} - \psi_{s\beta} \cdot i_{s\alpha})$$

El estimador de flujo y par diseñado se expone en la figura 3.25. El bloque que computa el módulo y ángulo a partir de las componentes α, β es el mismo que el expuesto en la figura 3.21.

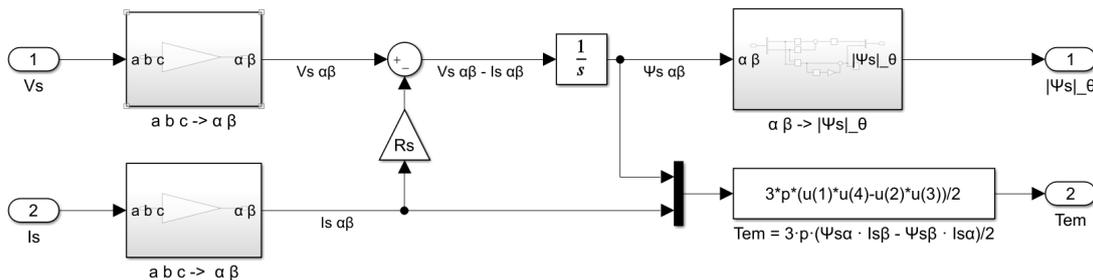


Figura 3.25. Estimador de flujo y par.

Seguidamente, es necesario calcular las referencias de flujo y par. La referencia del flujo de estator se mantendrá a su valor nominal de una forma similar al control vectorial. A velocidades inferiores a la nominal, la referencia de ψ_s se mantendrá a su valor nominal, mientras que a velocidades superiores este valor se reducirá de forma inversamente proporcional. El valor nominal del flujo de estator se ha determinado de forma similar al apartado anterior, sometiendo el motor de inducción a sus condiciones nominales y analizando el valor del flujo mediante simulación. Se ha obtenido $\psi_{s,nom} = 1.037 \text{ Wb}$.

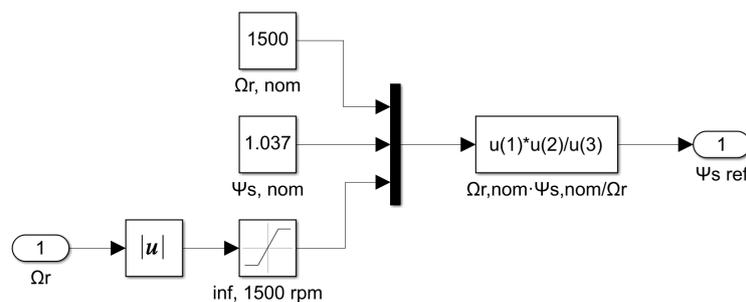


Figura 3.26. Cómputo de $\psi_{s,ref}$.

La referencia de par se ha determinado a partir de un regulador PI cuya entrada es el error de velocidad.

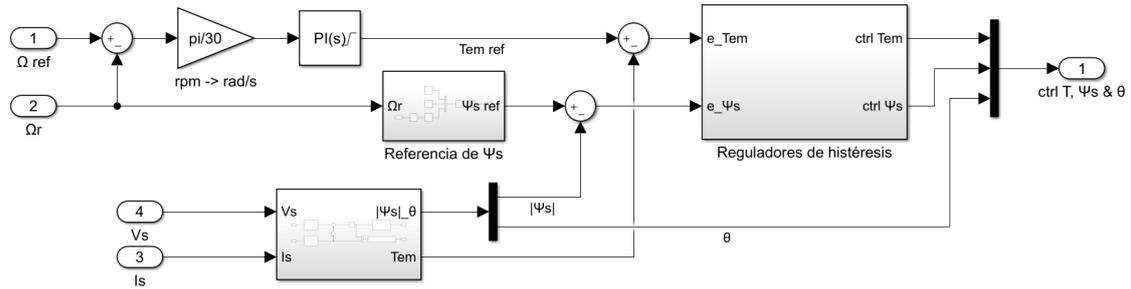


Figura 3.27. Esquema control directo de par.

Los errores de par y flujo son la entrada de los reguladores de histéresis cuyo esquema se detalla en la figura 3.28.

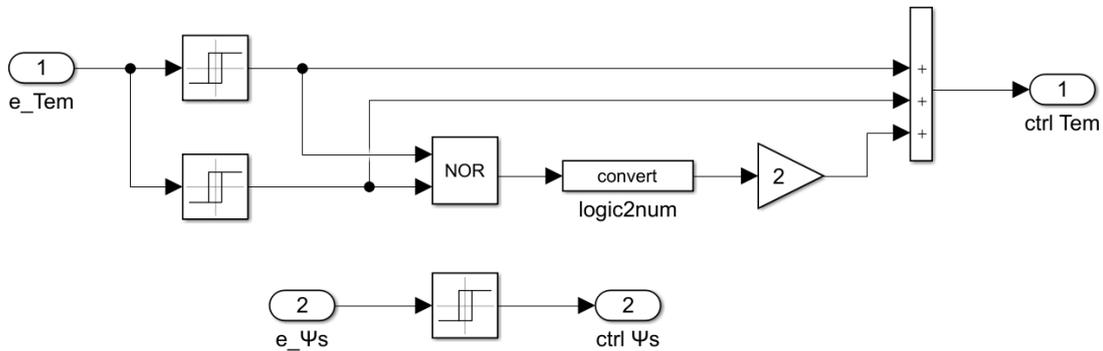


Figura 3.28. Reguladores de histéresis.

Los patrones de conmutación se eligen a partir de las señales proporcionadas por el regulador de histéresis y el ángulo del flujo de estator a partir del cual se determina el sector correspondiente según la tabla 3.3, donde se detalla qué vector espacial de los descritos en la tabla 3.2 es necesario aplicar en función de las variables $ctrl_{T_{em}}$ y $ctrl_{\psi_s}$ y el sector de Ψ_s .

$ctrl_{\psi_s}$	$ctrl_{T_{em}}$	Sector 1	Sector 2	Sector 3	Sector 4	Sector 5	Sector 6
1	1	\vec{V}_2	\vec{V}_3	\vec{V}_4	\vec{V}_5	\vec{V}_6	\vec{V}_1
	2	\vec{V}_0	\vec{V}_7	\vec{V}_0	\vec{V}_7	\vec{V}_0	\vec{V}_7
	3	\vec{V}_6	\vec{V}_1	\vec{V}_2	\vec{V}_3	\vec{V}_4	\vec{V}_5
2	1	\vec{V}_3	\vec{V}_4	\vec{V}_5	\vec{V}_6	\vec{V}_1	\vec{V}_2
	2	\vec{V}_7	\vec{V}_0	\vec{V}_7	\vec{V}_0	\vec{V}_7	\vec{V}_0
	3	\vec{V}_5	\vec{V}_6	\vec{V}_1	\vec{V}_2	\vec{V}_3	\vec{V}_4

Tabla 3.3. Patrones de conmutación – DTC.

En la figura 3.11 se evidenciaba la existencia de dos formas de generar los patrones de conmutación. El segundo de los bloques se corresponde con el patrón que se expone en la tabla 3.3, cuya implementación se efectúa tal como se detalla en la figura 3.29. El código de la selección de patrón de conmutación se expone en [42].

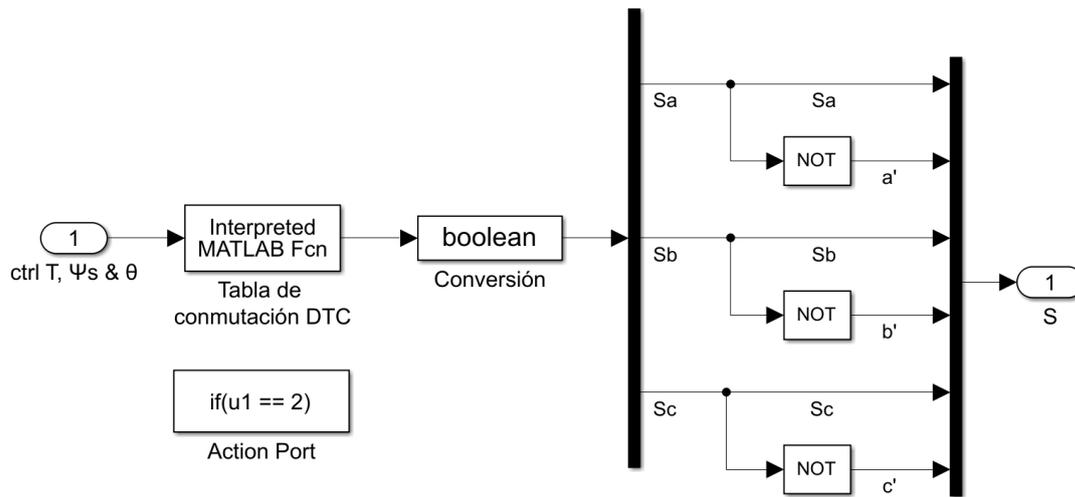


Figura 3.29. Patrón de conmutación - DTC.

Finalmente, los resultados se detallan en la figura 3.30, donde se corrobora un correcto seguimiento de la referencia de velocidad. En el caso de la prueba de par, se observa cierto error de posición debido a la definición de la banda de histéresis de ambos reguladores.

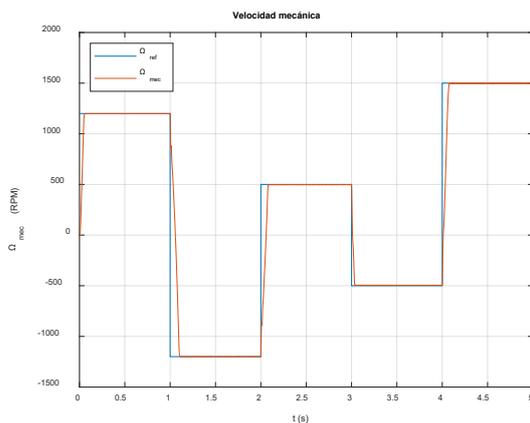


Figura 3.30a. Prueba de velocidad.

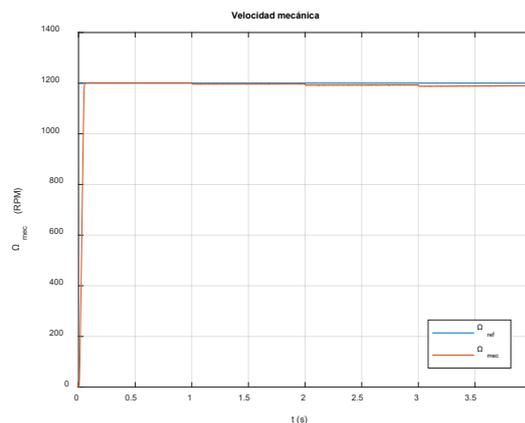


Figura 3.30b. Prueba de par.

Figura 3.30. Resultados control directo de par.

3.6 INTEGRACIÓN DE LOS MODELOS EN APP DESIGNER

Con el objetivo de facilitar al usuario la configuración de los principales parámetros que intervienen en los modelos digitales diseñados, se ha elaborado una interfaz gráfica en App Designer de Matlab en la que se han implementado las siguientes funciones:

- Modificación de los parámetros del modelo del motor de inducción.
- Elección del tiempo de simulación.
- Selección de la estrategia de control: escalar, vectorial o directo de par.
- Selección del tipo de conexión: directa a red o a través del variador.
- Introducción de la referencia de velocidad y del par resistente.

- Visualización de resultados, representando gráficamente la posición mecánica del rotor, su velocidad angular, el par electromagnético y la corriente de estator. Además, se ha elaborado un diagrama fasorial que permite representar cualquier magnitud a partir de sus componentes. En particular, en el presente proyecto únicamente se representan las corrientes de estator y rotor.

La aplicación diseñada se expone en las figuras 3.31 y 3.32, donde se expone la configuración de la simulación y la representación de los datos correspondientes, respectivamente.

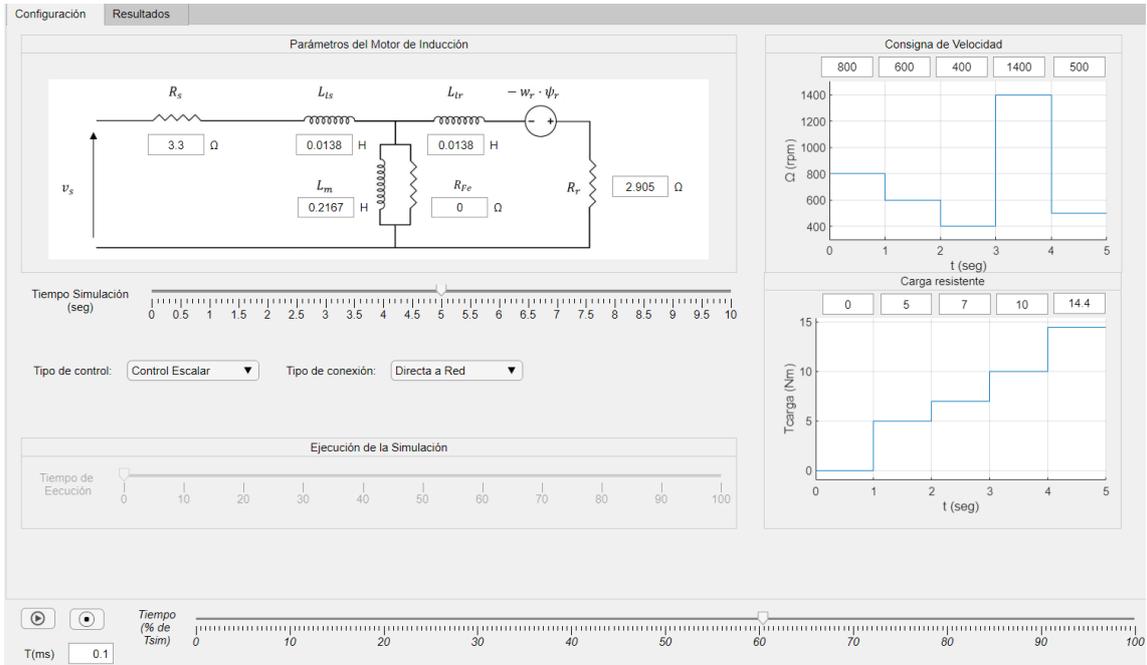


Figura 3.31. Interfaz App Designer - Configuración de parámetros.

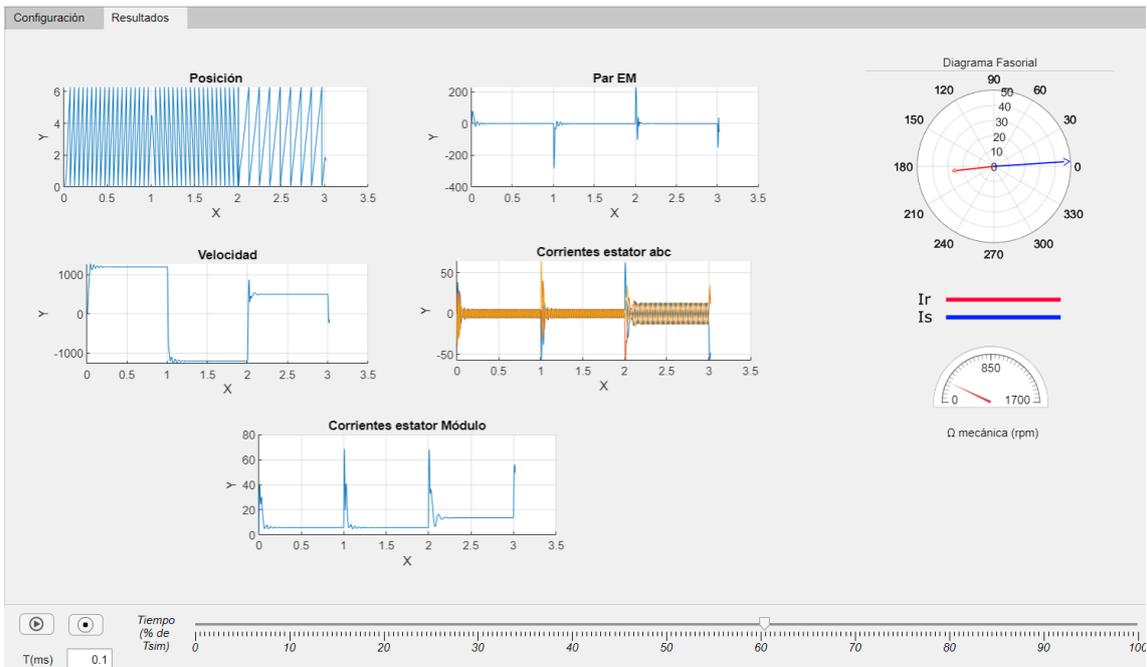


Figura 3.32. Interfaz App Designer - Visualización de resultados.

En las figuras anteriores se observa que la aplicación diseñada se ha dividido en dos ventanas. Una de estas está orientada a la introducción de los parámetros del motor de inducción, consignas de velocidad o par, selección del tiempo de simulación y, además, dispone de una barra de progreso que informa al usuario del avance de la simulación ejecutada. En la siguiente ventana se ha dispuesto la visualización de las diferentes magnitudes citadas anteriormente. En esta segunda ventana se representan los datos hasta el instante que el usuario haya seleccionado en la barra inferior.

Inicialmente, se planteó la posibilidad de representar en tiempo real las magnitudes citadas anteriormente, permitiendo modificar la referencia de velocidad y par también en tiempo real. Sin embargo, debido al elevado tiempo de computación que se deriva de la introducción del variador de frecuencia, finalmente se ha descartado esta opción, por lo que todos los resultados se visualizan una vez finaliza la simulación.

Como alternativa, se permite al usuario analizar los resultados hasta el instante deseado a través de una barra deslizable que indica hasta qué instante de tiempo se van a representar los datos. Esto permite, por ejemplo, visualizar el diagrama fasorial en un determinado instante de tiempo.

CAPÍTULO 4: SENSORIZACIÓN E INSTRUMENTACIÓN ELECTRÓNICA

Hasta ahora se ha logrado desarrollar un modelo digital del banco de ensayos del laboratorio, permitiendo realizar los mismos ensayos que en el sistema real, es decir, alimentando el motor directamente a través de la red eléctrica o bien mediante el variador de frecuencia, integrando las diferentes tecnologías de control que implementa éste.

En el presente capítulo se procede a describir la instrumentación electrónica que se ha empleado en el presente proyecto para diseñar un producto basado en un dispositivo de bajo coste que permite controlar y monitorizar el puesto de prácticas. Este producto integra unas funcionalidades muy similares a las implementadas en la interfaz de App Designer desarrollada en el capítulo 3, es decir, la solución diseñada debe permitir modificar las consignas de velocidad, o frecuencia en el caso de operar a través del variador en bucle abierto, y par resistente, además de registrar los valores particulares de diferentes magnitudes eléctricas y mecánicas.

Las magnitudes que se requieren medir son las siguientes:

- Posición y velocidad angular.
- Par de la carga.
- Corrientes de estator consumidas.

En los epígrafes 4.1 a 4.3 se describen las diferentes tecnologías de sensores que permiten realizar las lecturas correspondientes y la justificación de la selección de la alternativa más adecuada. Se analizarán los sensores de posición y velocidad, sensores de corriente y sensores de par. Se describirá la instrumentación que se ha empleado para acondicionar las diferentes señales a los rangos de entrada y salida necesarios.

El microprocesador seleccionado, el ESP32, no presenta los mismos niveles de tensión de entrada y salida que los correspondientes a los sensores implementados ni a los asociados a las entradas digitales y analógicas de los variadores. Además, cada sensor requiere una alimentación a un nivel de tensión diferente, por lo que será necesario emplear instrumentación electrónica adicional.

En particular, en el apartado 4.4 se detalla como comandar los variadores de frecuencia del ACIM y PMSM a través del microprocesador empleando un conjunto de relés externos. En el apartado 4.5 se expone como se ha implementado la alimentación de los diferentes sensores y, finalmente, en el epígrafe 4.6 se expone el producto diseñado.

4.1 SENSORES DE POSICIÓN Y VELOCIDAD

En este apartado se realiza un estudio de las tecnologías asociadas a los sensores de posición y velocidad que se emplean en la industria, analizando sus diferencias con el objetivo de justificar la elección de uno de ellos. Se analizarán los encoders magnéticos y ópticos, tanto incrementales como absolutos. Mediante estos sensores se obtiene una lectura directa de la posición angular a partir de la cual es posible calcular la velocidad angular.

Finalmente, se describirá como efectuar la lectura de posición y velocidad mediante el sensor seleccionado.

4.1.1 ENCODER MAGNÉTICO

Consisten en un conjunto de imanes permanentes o electroimanes cuyo campo magnético generado es detectado por sensores magnéticos, habitualmente sensores de efecto Hall o resistencias magnéticas.

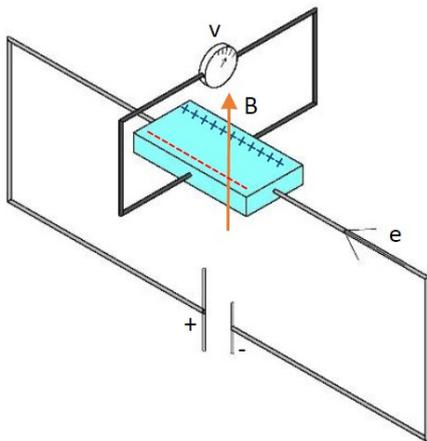


Figura 4.1. Efecto Hall.

El efecto Hall [18] consiste en la generación de una diferencia de potencial debido a la aplicación de un campo magnético sobre un flujo de cargas eléctricas. En particular, si un material conductor por el que circula una corriente eléctrica es sometido a un campo magnético perpendicular a dicha corriente, entonces sobre las cargas actúa una fuerza dada por la ley de Lorentz, perpendicular a la corriente y el campo magnético. Esta fuerza genera una desviación de las cargas respecto su sentido natural, dando lugar a una diferencia de tensión proporcional al campo magnético y cuyo valor es posible medir.

- Encoder magnético absoluto

Codifican cada posición angular mediante una secuencia única, generalmente expresada en código binario natural. En este tipo de sensores, el valor asociado a la posición angular no se pierde al desconectar la alimentación del sensor, a diferencia de lo que ocurre en los encoders incrementales [20], tal como se expone posteriormente.

La principal desventaja que se halla en la implementación de este tipo de sensores radica en la dificultad de diseñar un controlador específico que sea capaz de interpretar la señal producida por estos encoders. Un ejemplo de aplicación y diseño de una interfaz de comunicación entre ESP32 y encoder se puede hallar en [22], donde se ha empleado el sensor EMS22A50, el cual está compuesto por 4 sensores de efecto Hall desfasados 90° en el espacio, dando lugar a 4 señales senoidales desfasadas entre ellas 90° en el tiempo. A partir de estas cuatro señales, es posible obtener la posición angular a partir de las ecuaciones 4.1. La generación de las diferentes señales se expone en la figura 4.2.

$$(4.1) \quad \begin{aligned} S_1 &= V_{HS1} + V_{HS2} - V_{HS3} - V_{HS4} \\ S_2 &= V_{HS1} - V_{HS2} - V_{HS3} + V_{HS4} \\ \theta &= \text{atan2}(S_2, S_1) \end{aligned}$$

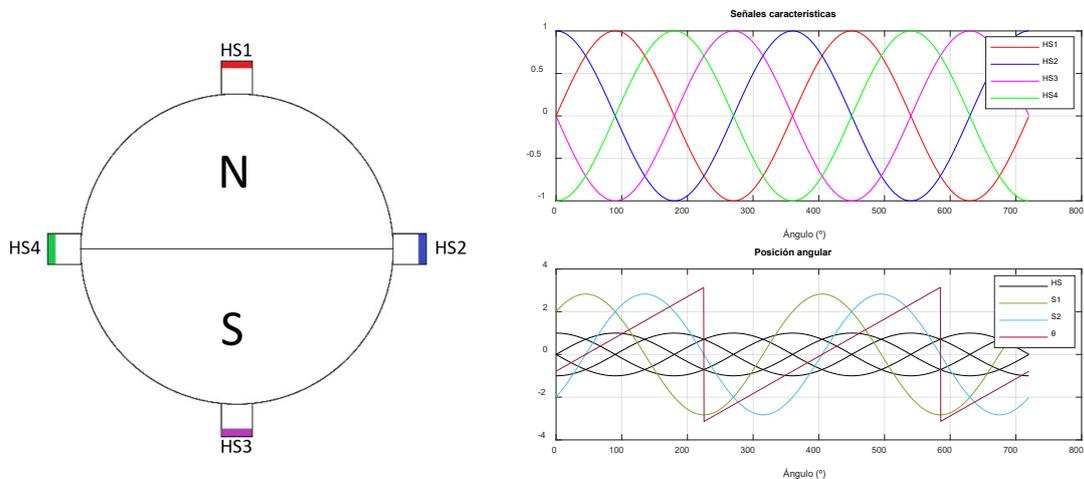


Figura 4.2. Señales producidas por un encoder magnético absoluto.

Normalmente, la posición angular está codificada en binario natural, cuya secuencia se obtiene a partir de un registro de datos característico de, por ejemplo, el bus SPI. En el caso del encoder EMS22A50 citado anteriormente, la información de salida del sensor se obtiene a partir de tres señales eléctricas tal como se expone en la figura 4.3.

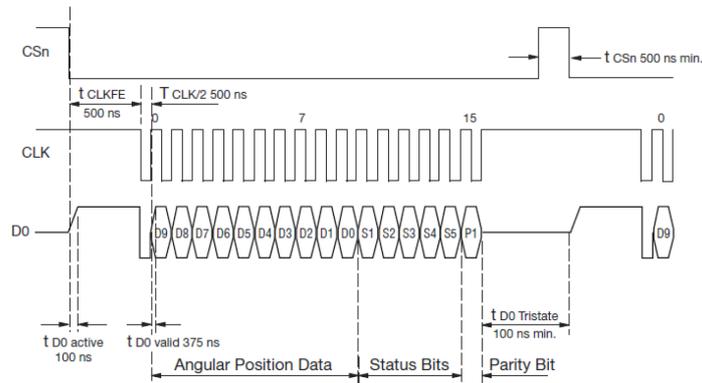


Figura 4.3. Transmisión de datos - Encoder magnético absoluto [22].

En definitiva, con el fin de efectuar la lectura de la posición angular, es necesario emplear tres señales principales, además de la referencia de tensión y la alimentación. Por una parte, la señal CS_n (Chip Select) se encarga de indicar la transmisión de una nueva trama de datos, la señal CLK (señal de reloj) indica cuando se debe de producir la transmisión de un nuevo bit dentro de cada trama de datos, y finalmente, la señal D_0 almacena el conjunto de bits que componen una trama de datos, dentro de la cual se encuentra la posición expresada en binario natural y otra información de interés como podría ser el sentido de giro o el bit de paridad, el cual indica si se ha producido algún error en la transmisión de una trama de datos.

Por lo citado anteriormente, se evidencia cierta dificultad en la implementación de la lectura de posición angular a través de un encoder absoluto.

- Encoder magnético incremental

Generalmente están compuestos por una serie de imanes distribuidos uniformemente a lo largo de la periferia de un disco de tal modo que, gracias a un sensor de efecto Hall, por cada revolución se genera un número de pulsos igual al número de pares de polos que forman el disco magnético. A diferencia de los sensores absolutos, el valor de la posición actual se pierde al desconectar la alimentación del sensor [20].

Además, si se dispone otro sensor de efecto Hall desfasado 90° en el espacio respecto al anterior, se generan dos señales cuadradas, denotadas por canales A y B, desfasadas 90° en el tiempo a partir de las cuales es posible determinar el sentido de giro del eje al que esté acoplado el encoder, tal como se expone en la figura 4.4.

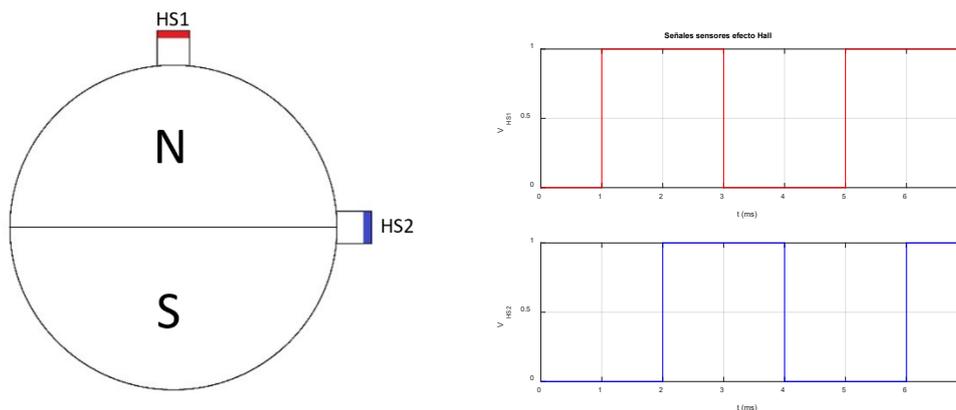


Figura 4.4. Encoder magnético incremental.

En la figura anterior, el encoder presenta un par de polos y 2 sensores de efecto Hall, dando lugar a una resolución correspondiente a 2 bits ya que es posible determinar 4 posiciones en cada sector. Para incrementar la resolución del encoder se debe aumentar el número de sensores de efecto Hall o bien el número de pares de polos. Si, por ejemplo, se disponen de 64 pares de polos y 4 sensores de efecto Hall, se generarían 1024 pulsos por revolución (PPR), dando lugar a una resolución asociada a 10 bits [19]. Además, se suele disponer de otro canal, habitualmente denotado por canal Z, que se emplea como referencia con el objetivo de indicar cuándo se ha efectuado una revolución completa.

El desarrollo de un algoritmo que sea capaz de efectuar la lectura de posición en los encoders incrementales es relativamente más sencillo que el correspondiente a los absolutos, debido a la mayor facilidad en la lectura de la posición angular y un menor uso de pines del microcontrolador.

4.1.2 ENCODER ÓPTICO

Su principio de funcionamiento se basa en la emisión de luz hacia un disco formado por un conjunto de sectores transparentes y opacos. En la parte opuesta del disco se halla un fotorreceptor que se encarga de determinar si el haz de luz ha sido bloqueado por los sectores opacos citados anteriormente [21].

El disco puede estar formado por una o varias pistas circulares respecto de las cuales se distribuyen los sectores transparentes y opacos. Habitualmente, cada pista dispone de un

fotoemisor y fotorreceptor de modo que se generan señales cuadradas a partir de las cuales se obtiene la posición angular asociada, a excepción de los encoders incrementales, que están formados por una única pista principal en la cual se ubican dos fotorreceptores que permiten generar las señales de cuadratura tal como se expondrá posteriormente.

- Encoder óptico absoluto

De forma análoga a los sensores magnéticos, los encoders ópticos absolutos generan una trama de datos de tal modo que cada posición se encuentra vinculada a una codificación única. Además, no pierden el valor actual de posición cuando cesa la alimentación. En este tipo de encoders, cada fotorreceptor emite un 1 o 0 lógico en función de si el haz de luz ha sido bloqueado o no por los sectores.

La resolución de los encoders absolutos es dependiente del número de pistas que componen al disco óptico. En particular, el número de bits de salida es igual al número de pistas. En la figura 4.5 se expone el esquema asociado a un disco óptico formado por 4 pistas, dando lugar a un total de 16 posiciones diferentes. Además, en dicha figura se exponen los dos principales tipos de codificación en la que está expresado el valor de posición angular proporcionado por el sensor: binario natural o código Gray.

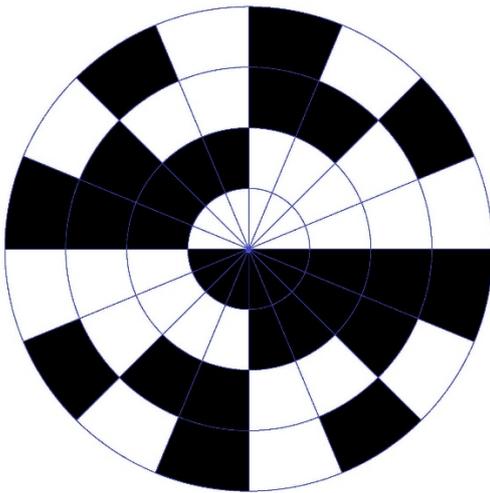


Figura 4.5a. Binario natural.

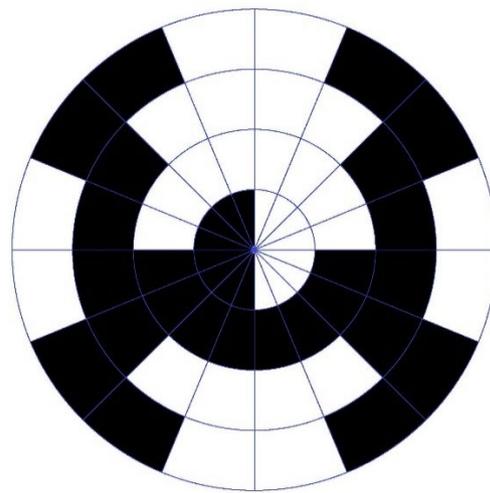


Figura 4.5b. Código Gray.

Figura 4.5. Codificación encoder óptico absoluto.

Generalmente, los encoders ópticos absolutos están codificados en código Gray debido a sus ventajas en el caso de que se produzca un error de lectura. En definitiva, las dos codificaciones posibles se basan en la lectura de bits, de modo que en la transición de una posición angular a otra se producen cambios en los bits desde nivel alto a nivel bajo o viceversa.

La ventaja del código Gray frente al código binario consiste en que de una posición angular a otra, es decir, de un sector de circunferencia a otro, se producen cambios en un único bit de todos los que componen un sector, tal como se puede evidenciar de la figura 4.5b.

Un error de lectura se produce cuando se trata de efectuar una lectura de posición en la transición de un sector a otro, de modo que es posible que no se haya terminado de producir la transición de un bit de un nivel a otro.

En el peor de los casos, en el código Gray se leerá la posición inmediatamente anterior al sector actual en el caso de que se produzca un error de lectura. Sin embargo, en el código binario, al efectuarse cambios en dos bits diferentes, puede ocurrir que al realizar la lectura únicamente se haya producido la transición de nivel de un único bit, en cuyo caso la posición muestreada estará muy alejada del sector real, produciendo cambios bruscos en la posición angular.

- Encoder óptico incremental

Los encoders incrementales están formados por una pista principal donde se distribuyen de manera uniforme los diferentes sectores a partir de los cuales se generan las señales de cuadratura de forma análoga a los encoders magnéticos incrementales. Además, es habitual disponer de una pista secundaria formada por un único sector transparente que actúa como referencia, con el objetivo de reiniciar el contador de pulsos e indicar si se ha completado una revolución completa.

La señal de salida de estos sensores es una señal cuadrada, de tal modo que cuando el haz de luz atraviesa un sector transparente, dicha señal se encuentra a nivel alto, mientras que si la luz es bloqueada por un sector opaco, la señal se encontrará a nivel bajo. Una configuración típica de los encoders ópticos incrementales se expone en la figura 4.6.

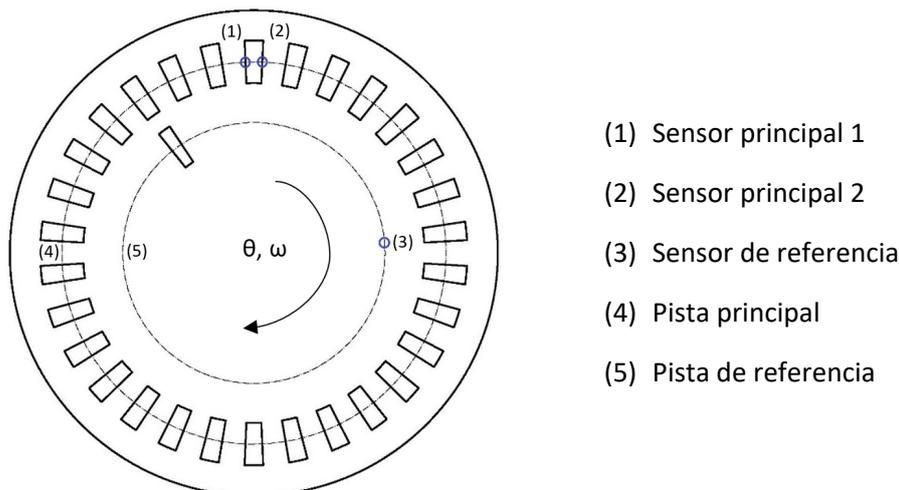


Figura 4.6. Encoder óptico incremental.

Existen dos alternativas para obtener la posición angular en los encoders incrementales. Por una parte, se evidencia que a medida que se reduce la velocidad de giro del eje respecto del que esté acoplado el encoder, el ancho de la señal cuadrada cuando ésta se encuentra a nivel alto aumenta, de modo que existe una proporcionalidad inversa entre el ancho de pulso y la velocidad de giro.

En consecuencia, mediante el empleo de un contador rápido sería posible leer el intervalo de tiempo en el que la señal se encuentra a nivel alto, proporcionando una lectura indirecta de la velocidad. Alternativamente, empleando una interrupción por hardware se puede contabilizar el número de pulsos generados durante un determinado periodo de tiempo, obteniendo la velocidad angular media en dicho intervalo.

4.1.3 SELECCIÓN DEL SENSOR DE POSICIÓN Y VELOCIDAD

La principal ventaja de los encoders incrementales radica en la facilidad de obtención de la posición angular ya que simplemente contabilizando el número de pulsos emitidos en un determinado intervalo de tiempo es posible obtener un registro de posición y velocidad angular. Además, en los encoders absolutos es necesario muestrear dos posiciones consecutivas que difieran en menos de una revolución con el objetivo de calcular la velocidad angular de forma correcta, mientras que en los incrementales no es necesario. Por otra parte, los encoders incrementales emplean un menor número de pines. De hecho, si se emplease un encoder absoluto sería necesario integrar un conversor adicional para incrementar el número de pines disponibles ya que los disponibles en la placa de desarrollo son insuficientes.



Figura 4.7. Encoder óptico incremental 1XP8001-1 de Siemens [23].

En consecuencia, se ha optado por emplear un encoder óptico incremental para realizar las lecturas de posición y velocidad, en particular el modelo 1XP8001-1 de Siemens.

Este modelo dispone de 1024 PPR (Pulsos por Revolución), es decir, lleva asociada una resolución de 10 bits.

Es necesario incidir en que la tensión de salida del sensor depende de la tensión de alimentación, cuyo valor mínimo es de 10V.

En el presente proyecto se ha empleado una alimentación de 12V, por lo que la salida del encoder no será posible conectarla directamente a una entrada digital del ESP32 ya que su valor a nivel alto está asociado a 3.3V. Por ello, es necesario emplear un divisor de tensión para adaptar ambos niveles de tensión. El diseño de un divisor de tensión se realiza empleando el teorema de Millman tal como se expone a continuación, donde V_1 es la tensión de salida del sensor y V_2 se corresponde con la entrada del microprocesador.

$$(4.2) \quad V_2 = \frac{\frac{V_1}{R_1}}{\frac{1}{R_1} + \frac{1}{R_2}} = \frac{R_2}{R_1 + R_2} \cdot V_1$$

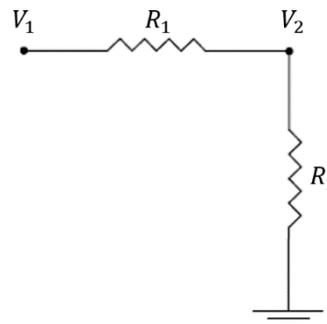


Figura 4.8. Divisor de tensión.

Mediante unos valores normalizados de resistencias de $R_1 = 4.7 \text{ k}\Omega$ y $R_2 = 1.5 \text{ k}\Omega$ se obtiene una ganancia de 0.242, obteniendo un valor a nivel alto de 2.9V a la entrada del ESP32, tensión suficiente para detectar los flancos de subida del encoder.

4.2 SENSORES DE CORRIENTE

Existen diferentes tecnologías que permiten la realizar la lectura de corriente [24], entre las que destacan las resistencias serie, los transformadores de corriente y los sensores de efecto Hall. En el presente epígrafe se describirán estas tecnologías.

Cabe destacar que se requiere el empleo de unos sensores que sean capaces de medir corrientes de bajas frecuencias y que presenten una implementación sencilla en el ESP32.

4.2.1 RESISTENCIA SERIE

Consiste en introducir una resistencia en serie que permite realizar una lectura de la corriente mediante la aplicación de la ley de Ohm. Para ello es necesario conocer de forma precisa el valor de la resistencia en serie, también conocida como resistencia *shunt*, y determinar el voltaje en bornes de esta resistencia. El valor de la corriente se obtendría simplemente dividiendo el voltaje medido entre el valor conocido de la resistencia. Esta metodología es sencilla de implementar y presenta un coste reducido.

El principal factor que afecta al correcto desempeño de este tipo de sensores es la temperatura, ya que en función de ésta el valor de la resistencia varía sustancialmente. Además, el hecho de incluir una resistencia en el circuito electrónico añade pérdidas de potencia adicionales por disipación térmica.

En consecuencia, el empleo de este tipo de sensores queda restringido a aplicaciones en las que la precisión y las pérdidas por disipación térmica no sean requisitos determinantes.

4.2.2 TRANSFORMADOR DE CORRIENTE

Es un tipo particular de transformador que produce una corriente en el devanado secundario proporcional a la corriente que se desea medir, la cual circula por el devanado primario. Permite reducir el nivel de corriente a valores que sean fácilmente medibles. Habitualmente se reduce la corriente hasta valores de 5A o 1A, y ofrece aislamiento galvánico entre las magnitudes eléctricas de alta y baja tensión [25].

Debido a que su principio de funcionamiento se basa en el empleo de transformadores eléctricos, únicamente se pueden emplear para realizar la lectura de corriente alterna, por lo que no se podrían emplear para leer corrientes continuas o alternas a bajas frecuencias.

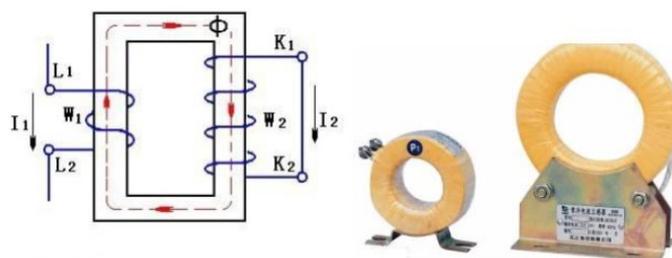


Figura 4.9. Transformador de corriente [25].

El devanado primario está compuesto por un número reducido de espiras que atraviesan el núcleo magnético que, habitualmente, presenta una forma toroidal sobre la cual se arrollan las espiras del devanado secundario, tal como se exponen en las figuras 4.9 y 4.10. Cabe destacar que el número de espiras del secundario es mucho más elevado que el correspondiente al primario [26].

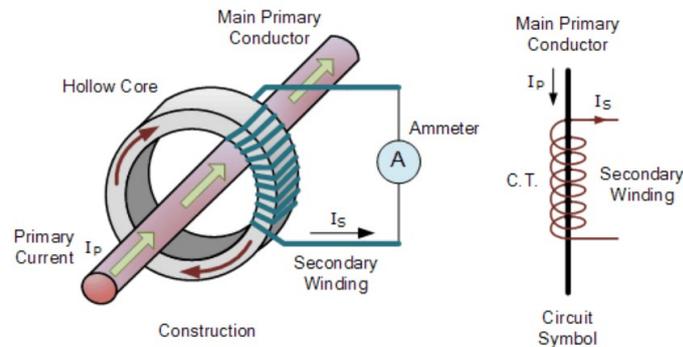


Figura 4.10. Principio de funcionamiento de un transformador de corriente [26].

Al cerrar el devanado secundario mediante una resistencia de bajo valor con el objetivo de medir la corriente ya sea a través de un amperímetro o de una resistencia serie, el transformador opera prácticamente en cortocircuito ya que la resistencia interna del secundario suele ser de un valor reducido. En consecuencia, se cumple que la relación de transformación es igual a la ratio de voltajes entre primario y secundario y éste aproximadamente proporcional a la ratio entre sendas corrientes según la expresión 4.3.

$$(4.3) \quad r_t = \frac{N_p}{N_s} = \frac{V_p}{V_s} \approx \frac{I_s}{I_p}$$

En consecuencia, es posible determinar la corriente del primario a partir de la lectura del secundario tal como sigue:

$$(4.4) \quad I_p \approx \frac{I_s}{r_t}$$

Tal como se ha comentado anteriormente, el número de espiras del primario es significativamente reducido mientras que el secundario puede estar formado por decenas o cientos de espiras. Esto permite modificar sustancialmente la relación de transformación simplemente incrementando el número de espiras del primario tal como se expone en la figura 4.11.

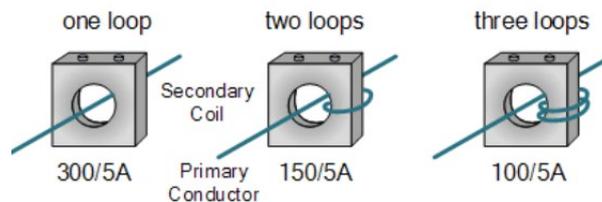


Figura 4.11. Modificación de la relación de transformación [26].

4.2.3 SENSORES DE EFECTO HALL

Su principio de funcionamiento es el mismo en el que se basan los encoders magnéticos: el efecto Hall. A diferencia de los transformadores de corriente, los sensores de efecto Hall son capaces de medir corrientes a bajas frecuencias e incluso corriente continua.

Este tipo de sensores se clasifican en dos categorías: sensores de lazo abierto y cerrado. Ambos tipos de sensores están formados por tres componentes principales: un núcleo magnético de forma toroidal que se encarga de concentrar las líneas de flujo magnético a través del cual circula la corriente que se desea medir, el sensor Hall propiamente dicho y la electrónica necesaria para acondicionar la señal [28].

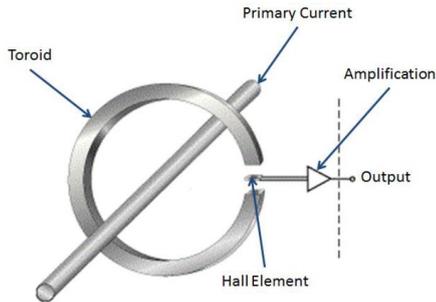


Figura 4.12a. Bucle abierto [27].

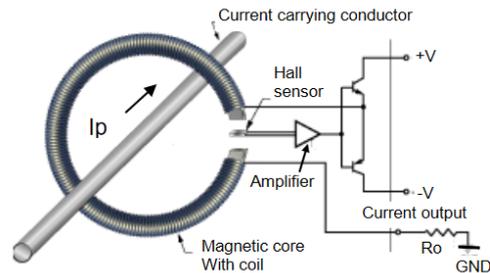


Figura 4.12b. Bucle cerrado [28].

Figura 4.12. Tipos de sensores de efecto Hall.

En los sensores de lazo abierto se hace pasar la corriente que se desea medir, conocida por corriente principal o primaria y denotada por I_p , a través del núcleo magnético, de tal modo que se generará un campo magnético proporcional a dicha corriente. En la sección del entrehierro se sitúa un sensor Hall de forma similar a los encoders magnéticos de forma que el campo magnético en el entrehierro será perpendicular a dicho sensor Hall. En consecuencia, a su salida se generará un voltaje, debido a la fuerza de Lorentz, proporcional al campo magnético que lo atraviesa, el cual a su vez es proporcional a la corriente que atraviesa el núcleo magnético. De este modo, realizando la lectura del voltaje de salida, es posible medir la corriente principal.

Por otra parte, los sensores de lazo cerrado presentan un componente adicional que consiste en un devanado alrededor del núcleo magnético, tal como es posible observar en la figura 4.12b, de forma que el flujo magnético generado por esta corriente secundaria se opone al generado por la corriente principal. La tensión de salida del sensor Hall se emplea para regular un amplificador de corriente, cuya corriente de salida es la que circula por el citado devanado. El objetivo es conseguir que el campo magnético resultante en el entrehierro sea nulo, condición que se cumple cuando se verifica la ecuación 4.4.

$$(4.4) \quad N_p \cdot I_p = N_s \cdot I_s$$

En la ecuación anterior, I_s es la corriente que circula por el devanado del núcleo magnético, también conocida por corriente de compensación, y N_s es el número de vueltas o espiras alrededor de éste. Conocido el valor de I_s , es posible determinar el valor asociado a la corriente principal. Para ello se sitúa una resistencia entre la corriente de salida del devanado y masa, tal como se observa en la figura 4.12b.

Frente a los sensores de lazo abierto, los de lazo cerrado presentan una rápida respuesta, gran linealidad, una reducida deriva térmica y son menos sensibles al ruido. Sin embargo, el hecho de introducir un elemento adicional encarece la adquisición de este tipo de sensores.

4.2.4 SELECCIÓN DEL SENSOR DE CORRIENTE

Debido a su economía, tamaño y facilidad de montaje, se ha optado por emplear los sensores de efecto Hall de lazo cerrado, en particular el modelo LTSR-6np del fabricante LEM, cuyas especificaciones se detallan en la figura 4.13.

Electrical data			
I_{PN}	Primary nominal RMS current	6	At
I_{PM}	Primary current, measuring range	0 ... ± 19.2 ¹⁾	At
I_P	Overload capability	250	At
V_{out}	Output voltage (analog) @ I_P	$2.5 \pm (0.625 \times I_P / I_{PN})$	V
	@ $I_P = 0$	2.5 ²⁾	V
V_{ref}	Reference voltage (internal reference), Ref _{OUT} mode	2.5 ³⁾	V
	Reference voltage (external reference), Ref _{IN} mode	1.9 ... 2.7 ⁴⁾	V
G	Sensitivity	104.16	mV/A
N_S	Number of secondary turns (± 0.1 %)	2000	
R_L	Load resistance	≥ 2	k Ω
C_{Lmax}	Maximum capacitive loading	500	pF
R_{IM}	Internal measuring resistance (± 0.5 %)	208.33	Ω
TCR_{IM}	Temperature coefficient of R_{IM}	< 50	ppm/K
U_C	Supply voltage (± 5 %)	5	V
I_C	Current consumption @ $U_C = 5$ V	Typical	$28 + I_S$ ⁵⁾ (V_{out}/R_L) mA

Figura 4.13. Hoja de características - Sensor de efecto Hall [29].

La salida proporcionada por el sensor está expresada en voltios y presenta un valor de 2.5V cuando la corriente primaria es de 0A. En consecuencia, debido a que el valor medio de las corrientes que se desean medir es de 0A, se cumple que el valor medio de la tensión de salida sea de 2.5V. Además, de dicha figura se puede deducir que la ganancia entre la tensión de salida y la corriente total que atraviesa el sensor es de:

$$(4.5) \quad G = \frac{1}{104.16 \frac{mV}{A}} = 9.6 \frac{A}{V}$$

Seguidamente, se procede a obtener la relación existente entre el valor digital medido por el convertidor ADC del microcontrolador, el cual está comprendido entre 0 y 4095, y la corriente que atraviesa el sensor de efecto Hall. El objetivo es obtener la ganancia del sensor, denotada por G_{Hall} , cuyo cálculo permitirá reducir el número de operaciones realizadas por el sensor en cada paso de muestreo.

Existe una relación matemática entre la tensión de salida, V_{out} , y la corriente primaria, I_p . A partir de la figura 4.13, se puede extraer la expresión 4.6, donde I_{PN} representa la corriente primaria nominal y tiene un valor de 6A.

$$(4.6) \quad V_{out} = 2.5 + 0.625 \cdot \frac{I_p}{I_{PN}}, \quad I_{PN} = 6A$$

Si se tiene en cuenta el número de espiras asociadas a la corriente primaria, es necesario modificar la ecuación anterior, dando lugar a la ecuación 4.7, donde N representa el número de vueltas de la corriente que se desea medir.

$$(4.7) \quad V_{out} = 2.5 + 0.625 \cdot \frac{N_p \cdot I_p}{I_{PN}}$$

De este modo, realizando la lectura de la tensión de salida es posible obtener el valor de la corriente primaria tal como se expone a continuación. Cabe destacar que se obtiene la ganancia de 9.6A/V calculada anteriormente.

$$(4.8) \quad I_p = \frac{I_{PN}}{0.625 \cdot N_p} \cdot (V_{out} - 2.5) = \frac{9.6}{N_p} \cdot (V_{out} - 2.5)$$

En efecto, se obtiene la misma ganancia calculada en 4.5. Cabe destacar que, tal como se ha derivado en la ecuación 4.8, para poder aplicar dicha ganancia es necesario restar previamente el valor medio de la tensión de salida del sensor.

Sin embargo, del mismo modo que ocurría en el caso del encoder, es necesario emplear un divisor de tensión para adaptar el rango de salida del sensor al rango de las entradas analógicas del ESP32. En concreto, se requiere reducir el valor máximo de salida del sensor, correspondiente a 5V, hasta los 3.3V de entrada del microcontrolador.

A partir de la ecuación 4.9, se define la ganancia entre la tensión de entrada y salida del divisor de tensión como:

$$(4.9) \quad G = \frac{R_2}{R_1 + R_2} = \frac{V_2}{V_1} = \frac{V_2}{V_{out}} \rightarrow V_{out} = \frac{V_2}{G}$$

Sustituyendo 4.9 en 4.8, se obtiene la siguiente expresión, donde V_2 es la tensión de entrada del ESP32.

$$(4.10) \quad I_p = \frac{I_{PN}}{0.625 \cdot N_p} \cdot \left(\frac{V_2}{G} - 2.5 \right) = \frac{I_{PN}}{0.625 \cdot N_p \cdot G} \cdot (V_2 - 2.5 \cdot G)$$

El valor de tensión V_2 que llega a una de las entradas analógicas no es directamente medible por el ESP32, sino que el microcontrolador lee el valor digital de salida proporcionado por el ADC. Por tanto, la relación entre el valor analógico y el valor digital leído por el ESP32 se expone en la ecuación 4.11, donde n representa el número de bits del convertidor ADC.

$$(4.11) \quad V_2 = \frac{V_{ADC}}{2^n - 1} \cdot 3.3, \quad n = 12$$

Sustituyendo en 4.10 se obtiene la relación matemática final que relaciona la corriente que se desea medir y el valor digital proporcionado por el convertidor analógico a digital.

$$(4.12) \quad I_p = \frac{I_{PN} \cdot 3.3}{0.625 \cdot N_p \cdot G \cdot (2^n - 1)} \cdot \left(V_{ADC} - 2.5 \cdot \frac{G \cdot (2^n - 1)}{3.3} \right) = G_{Hall} \cdot (V_{ADC} - V_{med})$$

$$G_{Hall} = \frac{I_{PN} \cdot 3.3}{0.625 \cdot N_p \cdot G \cdot (2^n - 1)}, \quad V_{med} = 2.5 \cdot \frac{G \cdot (2^n - 1)}{3.3}$$

Cabe destacar que, a priori, no es posible conocer el valor exacto de G_{Hall} y V_{med} en la ecuación anterior ya que existe cierta incertidumbre en el cálculo del parámetro G debido a las tolerancias de las resistencias. En consecuencia, se ha optado por obtener el valor de V_{med} de forma experimental mediante el cálculo del valor medio de un conjunto de lecturas.

Se han empleado unos valores normalizados de resistencias de $R_1 = 2.2 \text{ k}\Omega$ y $R_2 = 3.9 \text{ k}\Omega$, por lo que la ganancia nominal del divisor de tensión es $G = 0.64$, dando lugar a la siguiente recta de transferencia para un valor de $N_p = 1$:

$$(4.19) \quad I_p = 0.0121 \cdot (V_{ADC} - 1983)$$

Tal como se ha comentado, los valores anteriores diferirán de un sensor a otro, por lo que se podría requerir de un calibrado previo del sensor. Es posible analizar el efecto de las tolerancias del divisor de tensión teniendo en cuenta que las resistencias R_1 y R_2 tienen una tolerancia del 10% para ambos casos. Denotando por α y β las tolerancias de R_1 y R_2 , respectivamente, se deduce la ganancia del divisor de tensión en función de dichas tolerancias.

$$(4.13) \quad G = \frac{R_2 \cdot (1 \pm \beta)}{R_1 \cdot (1 \pm \alpha) + R_2 \cdot (1 \pm \beta)} = \frac{1}{1 + \frac{R_1 \cdot (1 \pm \alpha)}{R_2 \cdot (1 \pm \beta)}}$$

$$(4.14) \quad G_{\min} = \frac{1}{1 + \frac{R_1 \cdot (1 + \alpha)}{R_2 \cdot (1 - \beta)}} = 0.59$$

$$G_{\max} = \frac{1}{1 + \frac{R_1 \cdot (1 - \alpha)}{R_2 \cdot (1 + \beta)}} = 0.68$$

G_{nom}	$G_{Hall,nom}$	$V_{med,nom}$
0.6	0.0121	1983
G_{\min}	$G_{Hall,\min}$	$V_{med,\min}$
0.59	0.0113	1836
G_{\max}	$G_{Hall,\max}$	$V_{med,\max}$
0.64	0.0131	2123

Tabla 4.1. Análisis del efecto de las tolerancias de R_1 y R_2 .

En la tabla 4.1 se detalla el rango de valores que pueden tomar las ganancias del sensor de efecto Hall, así como el valor medio de la tensión digital muestreada, debido a las variaciones introducidas por las tolerancias de las resistencias. Cabe destacar que los valores nominales llevan asociados unas tolerancias nulas.

El valor medio real de la tensión digital es posible determinarlo experimentalmente de manera sencilla simplemente calculando el valor medio de un conjunto de muestras cuando el nivel de corriente es nulo. El valor de la ganancia G_{Hall} varía ligeramente debido al efecto de las tolerancias, por lo que se ha optado por emplear su valor nominal. Como alternativa se podría haber realizado un proceso de calibración previo, haciendo pasar una corriente de valor conocido por el sensor y muestrear el voltaje de salida. El principal inconveniente de esta metodología consiste en que el valor de G_{Hall} se efectúa a partir de datos experimentales contaminados por el ruido de medida.

En función del número de espiras de la corriente primaria, existen diferentes conexiones asociadas a los terminales del sensor de efecto Hall, tal como se expone en 4.14. En el proyecto se ha empleado la primera configuración ya que únicamente se ha dispuesto de una espira.

Number of primary turns	Primary nominal RMS current	Nominal ¹⁾ output voltage V_{out}	Primary resistance R_p [$m\Omega$]	Primary insertion inductance L_p [μH]	Recommended connections
1	± 6	2.5 ± 0.625	0.18	0.013	
2	± 3	2.5 ± 0.625	0.81	0.05	
3	± 2	2.5 ± 0.625	1.62	0.12	

Figura 4.14. Configuración del sensor de efecto Hall [29].

En la figura 4.15 es posible analizar la relación existente entre la tensión de salida del sensor y la corriente primaria que lo atraviesa. Se observa que el voltaje de salida satura a partir de tensiones superiores a 4.5V. De hecho, este valor de tensión lleva asociado una corriente de $I_p = 9.6 \cdot (4.5 - 2.5) = 19.2 A$, es decir, el fondo de escala del sensor, cuyo valor coincide con el expuesto en la hoja de características de la figura 4.13.

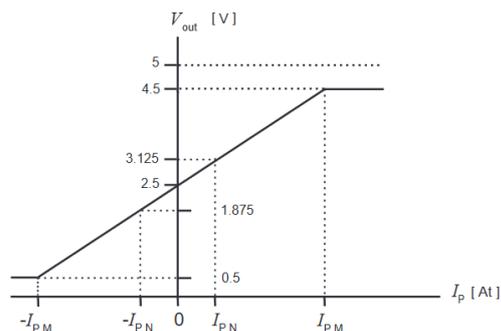


Figura 4.15. Relación tensión de salida y corriente primaria [29].

Respecto al muestreo de la corriente, hay que tener en cuenta que es necesario tomar un número de muestras suficiente dentro de cada periodo con el objetivo de conseguir que la onda muestreada se parezca en la medida de lo posible a una senoidal.

El caso más crítico se corresponde con una frecuencia asociada a las tensiones de alimentación del motor de 50Hz, en cuyo caso, según el teorema de Nyquist, el paso de muestreo máximo de las corrientes de estator del motor de inducción debe ser de:

$$(4.15) \quad T_s < T_{s\text{m}\acute{a}x} = \frac{1}{2 \cdot 50} = 10 \text{ ms}$$

En el presente proyecto se ha empleado un paso de muestreo de 2ms, valor suficiente para realizar las lecturas de posición y corrientes.

4.3 SENSORES DE PAR

Existen dos principales metodologías que permiten implementar una lectura de par: los métodos directos e indirectos. Mientras que los métodos directos consisten en medir el par a través del propio eje de transmisión de potencia, los métodos indirectos se basan en la obtención del par a partir de una medida auxiliar, en general a partir de magnitudes eléctricas auxiliares [30].

4.3.1 MÉTODO DIRECTO

Por otra parte, es posible implementar lecturas de par basadas en el efecto piezoeléctrico. Habitualmente, para tal fin se emplean galgas extensométricas instaladas en un puente de Wheatstone.

Cuando sobre el eje de transmisión actúa un par, se produce una torsión y flexión en el propio eje, lo cual se traduce en una deformación de la galga extensométrica, dando lugar a una variación en su resistencia. A partir de un puente de Wheatstone es posible traducir esta variación de resistencia en una variación de voltaje [30].



Figura 4.16. Transductores de par [30].

4.3.2 MÉTODO INDIRECTO

Permiten obtener una medida del par sin necesidad de emplear ninguna intervención mecánica en el eje de transmisión de potencia. Existen diferentes alternativas que permiten implementar lecturas de par basadas en métodos indirectos.

Por ejemplo, si se determina la potencia mecánica transmitida al eje, así como su velocidad angular, es posible calcular el par mediante la expresión 4.16. Debido al desconocimiento de la potencia mecánica transmitida al eje del motor, la lectura del par se puede efectuar a partir de la potencia eléctrica consumida por el motor, lo que conlleva ciertas imprecisiones en el cálculo del par debido a las pérdidas de potencia y a los diferentes puntos de operación del motor [30].

$$(4.16) \quad T_{em} = \frac{P_{mec}}{\Omega_{mec}}$$

Alternativamente, se puede realizar la lectura de par a partir del valor medio cuadrático de la corriente consumida por la carga conectada al motor ya que, en el presente proyecto, se trata de un motor de imanes permanentes.

$$(4.17) \quad T_{em} = K \cdot I_{RMS}$$

La principal ventaja del método indirecto, tal como se ha comentado anteriormente, consiste en que no se realiza ninguna intervención en el eje de transmisión y, además, emplean componentes relativamente más económicos. Sin embargo, la calibración de estos sensores, así como la implementación del cálculo del par en microcontroladores, son procesos más complejos de llevar a cabo en comparación con los métodos directos.

4.3.3 SELECCIÓN DEL SENSOR DE PAR

Se procede a efectuar una lectura indirecta del par. Para ello, hay que tener en cuenta que el par es proporcional al valor medio cuadrático de la corriente generada por la carga, es decir, el motor de imanes permanentes.

En primer lugar, es necesario calcular el valor medio cuadrático de la corriente generada por el motor síncrono, denotado por I_{RMS} . Su valor se calcula teniendo en cuenta que la señal obtenida es una señal discreta y se determina a partir de la ecuación 4.18, en la que es necesario tomar N muestras de la corriente a lo largo de un periodo completo.

$$(4.18) \quad I_{RMS} = \sqrt{\frac{\sum_{i=1}^N I_i^2}{N}}$$

En consecuencia, para poder obtener el valor medio cuadrático es necesario determinar previamente el periodo asociado a la onda senoidal de la corriente generada por el motor de imanes permanentes. Para calcular dicho periodo, hay que tener en cuenta que los ejes de ambos motores están acoplados mecánicamente, por lo que la velocidad de ambos motores es la misma, cumpliéndose la siguiente expresión:

$$(4.19) \quad \Omega_r = 60 \cdot \frac{f_{PMSM}}{p_{PMSM}}$$

El periodo de la señal de interés, teniendo en cuenta que el motor de imanes permanentes tiene 4 pares de polos, será, por tanto:

$$(4.20) \quad T_{PMSM} = \frac{1}{f_{PMSM}} = \frac{60}{\Omega_r \cdot p_{PMSM}} = \frac{15}{\Omega_r}$$

A partir de este valor se calcula el número de muestras necesarias para completar un periodo. Para esto hay que tener en cuenta el paso de muestreo empleado, el cual es de 2ms.

$$(4.21) \quad N_{m\acute{a}x} = \frac{T_{PMSM}}{T_s}, \quad T_s = 2ms$$

De este modo es posible implementar la medida indirecta del par. Hay que tener en cuenta que mediante esta metodología se muestrean dos señales: la corriente generada por el motor de imanes permanentes y el par. En el caso de la corriente, se ha empleado un paso de muestreo de 2ms, del mismo modo que las corrientes del motor de inducción y la posición angular. En este caso, la máxima frecuencia a la que opera el motor de imanes permanentes es 100Hz, por lo que el paso de muestreo debería ser inferior a 5ms según el teorema de Nyquist.

Por otra parte, el par se calcula dentro de un paso de muestreo variable dado por T_{PMSM} . Sin embargo, tal como se describirá en el capítulo 6, la aplicación Android permite registrar señales con pasos de muestreo diferentes, pero que deben ser constantes para cada señal. Debido a esto, se ha optado por emplear un paso de muestreo de 100ms, el mismo que el correspondiente a la velocidad. Entonces, en cada paso de muestreo se obtendrán las N muestras calculadas en la ecuación 4.18 y no se enviarán hasta transcurridos los 100ms.

Esto introduce una limitación en el cálculo del par ya que, si la corriente presenta un periodo superior a 100ms, el cálculo del valor medio cuadrático será erróneo. En particular, teniendo en cuenta que la frecuencia del motor de imanes permanentes es aproximadamente el doble de la correspondiente al motor de inducción, el variador de frecuencia del ACIM no debería de operar a frecuencias inferiores a 5Hz.

Finalmente, se aplica la relación de proporcionalidad directa entre el par y el valor medio cuadrático de la corriente.

$$(4.22) \quad T_{em} = K \cdot I_{RMS}$$

Es posible determinar el factor de proporcionalidad entre el par y el valor medio cuadrático de la corriente a partir de la placa de características del motor síncrono:

$$(4.23) \quad K = \frac{15.5}{14.4} = 1.11 \frac{Nm}{A}$$

4.4 ADAPTACIÓN DE LAS ENTRADAS DEL VARIADOR DE FRECUENCIA

Tal como se ha descrito al inicio del presente capítulo, las entradas digitales del variador presentan un valor a nivel alto de 24V, mientras que las analógicas poseen un rango de entrada entre 0V y 10V.

Todo ello, sumado al hecho de que las salidas de los sensores superan el valor máximo de entrada del ESP32, situado en 3.3V, evidencia la necesidad de emplear instrumentación electrónica adicional que permita adaptar los rangos de salida del microprocesador y de entrada del variador de frecuencia.

En cuanto a las entradas digitales de los variadores, se ha empleado un conjunto de 8 microrelés que permiten actuar sobre el variador del motor de inducción, así como sobre el motor de imanes permanentes. Respecto el motor de inducción, se emplean cuatro entradas digitales asociadas a la marcha y paro, inversión de giro y consignas de frecuencia o velocidad. Se pueden emplear hasta 4 consignas diferentes ya que se disponen de 2^2 combinaciones distintas.

Por otra parte, el variador del motor de imanes permanentes emplea tres de las cuatro entradas digitales restantes, que se corresponden con marcha y paro, reset y el selector del modo de control (velocidad o par).



Figura 4.17. Módulo de 8 relés.

En cuanto a sus entradas analógicas, se han empleado dos amplificadores operacionales, en particular el modelo OPA705PA, permitiendo amplificar la salida analógica del ESP32 desde 3.3V a 10V.

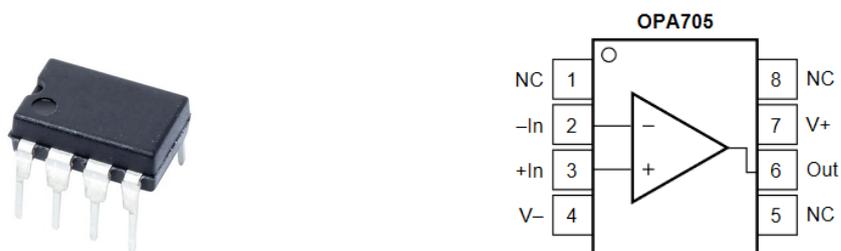


Figura 4.18. Amplificador operacional OPA705 [31].

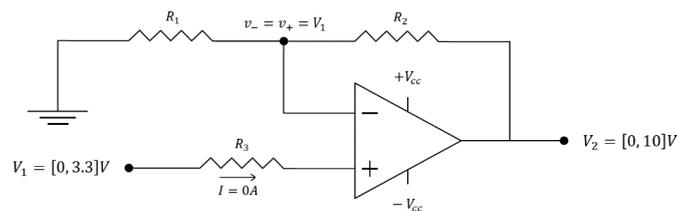


Figura 4.19. Esquema amplificador operacional.

El esquema de conexión de cada amplificador se representa en la figura 4.19. Teniendo en cuenta que, en el caso ideal, las corrientes de entrada del amplificador son nulas y se cumple que las tensiones de los terminales de entrada son iguales, entonces mediante la aplicación del teorema de Millman es posible obtener la relación entre la tensión de salida, que será la conectada al variador, y la tensión de entrada, que será a su vez, la tensión de salida del ESP32.

$$(4.24) \quad v_- = \frac{\frac{V_2}{R_2}}{\frac{1}{R_1} + \frac{1}{R_2}} = \frac{R_1}{R_1 + R_2} \cdot V_2 = v_+ = V_1 \rightarrow V_2 = \left(1 + \frac{R_2}{R_1}\right) \cdot V_1$$

Si se emplea un valor de $R_2 = 2 \cdot R_1$, se obtiene una ganancia de 3, por lo que la tensión de salida sería de 9.9V cuando la entrada sea de 3.3V. En consecuencia, se han empleado dos resistencias en serie de valor 4.7k para R_2 y una resistencia de 4.7k para R_1 .

En cuanto a la alimentación del amplificador operacional, se tiene que cumplir que su valor sea como mínimo el valor máximo de la tensión de salida del amplificador. En este proyecto se ha empleado una tensión de alimentación de 12V.

4.5 ALIMENTACIÓN DE LOS COMPONENTES ELECTRÓNICOS

La alimentación del encoder requiere de valor mínimo una tensión de 11V, mientras que la tensión de alimentación de los sensores de corriente y los microrrelés es de 5V. Por otra parte, la alimentación de los amplificadores operacionales debe ser superior a 10V.

Cabe destacar que el sensor de efecto Hall asociado a la lectura de par ya se encuentra instalado en el puesto de prácticas e incorpora un divisor de tensión para reducir el nivel de tensión de 24V a 5V.

Esto permite agrupar los diferentes componentes en tres niveles de tensión:

- Alimentación a 24V. Se corresponde con la tensión de entrada del divisor de tensión asociado al sensor de efecto Hall que permite implementar la lectura indirecta del par.
- Alimentación a 12V. Este nivel de tensión es requerido por el encoder óptico y por los dos amplificadores operacionales.
- Alimentación a 5V. Es requerida por los 3 sensores de efecto Hall y por el conjunto formado por los 8 microrrelés.

Se ha dispuesto de una fuente de tensión continua que permite pasar de la tensión de red a una tensión de 24V, permitiendo alimentar al primer sensor de efecto Hall, el cual se encuentra previamente instalado en cada uno de los bancos de ensayos disponibles en el laboratorio.

En particular, el modelo empleado es el DRL30-24-1 del fabricante TDK-Lambda, el cual se expone en la figura 4.20.



Figura 4.20. Fuente de tensión continua.

Seguidamente, es necesario reducir el voltaje de salida de la fuente de continua a los niveles de tensión de 12V y 5V, por lo que inicialmente se ha planteado un divisor de tensión de dos niveles, representado en la figura 4.21, que permitiera obtener dichos voltajes.

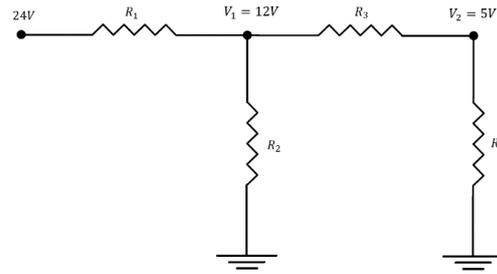


Figura 4.21. Divisor de tensión de dos niveles.

En este caso, en cada uno de los nodos se produce un consumo de corriente por parte de los sensores, amplificadores operacionales y microrrelés, por lo que de forma estricta la aplicación del teorema de Millman se debería de realizar tal como sigue, donde I_c representa la corriente consumida por cada componente conectado al nodo correspondiente.

$$(4.25) \quad V_j = \frac{\sum_i \frac{V_i}{R_i} - \sum_i I_{c_i}}{\sum_i \frac{1}{R_i}}$$

Como primera aproximación, y frente al desconocimiento de la totalidad de las corrientes consumidas por todos los componentes, se ha diseñado el divisor de tensión sin tener en cuenta el consumo de corriente. Posteriormente, se comprobará si la conexión de los diferentes componentes supone una caída de tensión aceptable.

Los niveles de tensión, en función de las diferentes resistencias vienen dados por las ecuaciones 4.26 y 4.27.

$$(4.26) \quad V_2 = \frac{R_4}{R_3 + R_4} \cdot V_1$$

Asumiendo un valor de $V_1 = 12V$, es posible dimensionar las resistencias R_3 y R_4 . En particular, se han empleado unos valores de 4.7k y 3.3k, respectivamente. A partir de estos datos, es posible determinar los valores de las resistencias restantes a partir de la siguiente expresión:

$$(4.27) \quad V_1 = \frac{\frac{24}{R_1}}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3 + R_4}} = \frac{(R_3 + R_4) \cdot R_2}{(R_3 + R_4) \cdot R_2 + (R_3 + R_4) \cdot R_1 + R_1 \cdot R_2} \cdot 24$$

Si se emplean unos valores de $R_1 = 2.7k$ y $R_2 = 4.7k$ se obtienen unas tensiones en los nodos de $V_1 = 12.55V$ y $V_2 = 5.17V$. Estos niveles de tensión han sido constatados de forma experimental sin tener en cuenta la conexión de los diferentes componentes. Sin embargo, cuando éstos son conectados a los diferentes nodos, se produce una caída de tensión excesiva debido al consumo de corriente que imposibilita la correcta alimentación de todos los componentes.

Como alternativa, se han empleado dos convertidores DC/DC, en particular los modelos B2412S-1W y B2405S-1W del fabricante EVISUN, que permiten reducir de 24V a 12V y de 24V a 5V, respectivamente. Mediante el empleo de estos sensores ha sido posible implementar de forma correcta la alimentación de todos los componentes.

4.6 DISEÑO DEL PROTOTIPO

Respecto al diseño y montaje del prototipo, hay que tener en cuenta que es necesario implementar los siguientes elementos:

- Dos sensores de efecto Hall. Cabe destacar que el tercer sensor restante, asociado a la lectura indirecta de par, ya se encuentra instalado en el banco de ensayos, por lo que no será necesario instalarlo en el prototipo a desarrollar.
- Cuatro divisores de tensión. Tres de ellos se emplean para reducir el nivel de tensión de salida de los sensores de efecto Hall y el restante, en la salida del encoder.
- Dos amplificadores operaciones.
- Módulo de 8 relés.
- Fuente de tensión continua de 24V.
- Dos convertidores DC/DC, de 24V a 12V y de 24V a 5V.

El esquema de conexión de todos los componentes, es decir, el diseño del circuito electrónico se expone en el plano 1 del presente documento. Por otra parte, la conexión de las entradas digitales y analógicas de los variadores de frecuencia del motor de inducción y el motor de imanes permanentes se expone en el plano 2.

Una vez diseñado el circuito electrónico, se procede a realizar el montaje de este. Para ello se han instalado todos los componentes en una *perfboard*, efectuando las conexiones eléctricas mediante soldadura en su parte trasera. El montaje del circuito se expone en la figura 4.22.

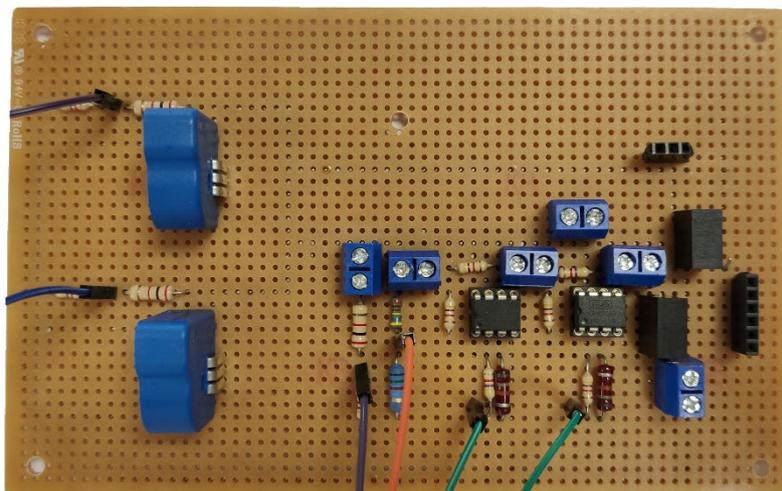


Figura 4.22. Implementación del circuito electrónico.

Seguidamente, el circuito diseñado anteriormente se ha integrado de forma conjunta con la fuente de tensión continua y el módulo de 8 relés tal como se detalla en la figura 4.23.

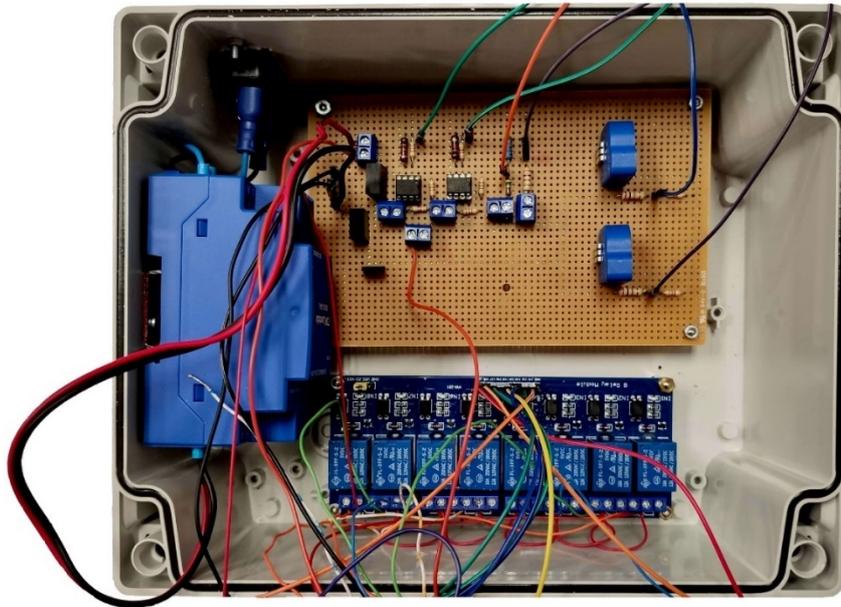


Figura 4.23. Montaje final.

Finalmente, el último paso consiste en implementar el núcleo central del circuito, es decir, el microprocesador ESP32. Además, se han dispuesto de los conectores necesarios que permiten realizar el conexionado de los sensores y variadores de frecuencia de cualquier banco de ensayos, tal como se detalla en la figura 4.24.

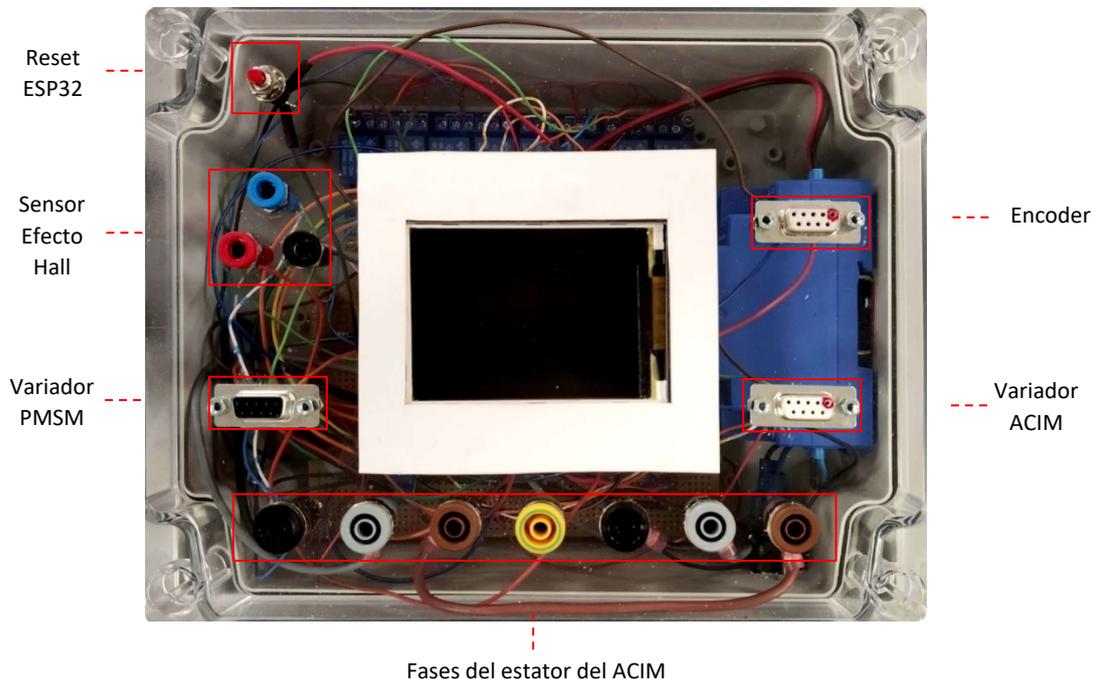


Figura 4.24. Producto final.

Se ha implementado de forma satisfactoria todo el hardware del proyecto, El siguiente paso consiste en diseñar el software embebido del microcontrolador y la aplicación Android que se comunicará con este. Esto se realiza en los capítulos 5 y 6.

CAPÍTULO 5: PROGRAMACIÓN DEL MICROCONTROLADOR ESP32

En el presente capítulo se describirán, en primer lugar, las principales características del microprocesador ESP32, núcleo central del proyecto.

En el apartado 5.2, se expondrá el diseño de un nuevo proyecto en el entorno de desarrollo Arduino IDE, detallando los pasos iniciales que se deben realizar para empezar a programar el microprocesador seleccionado.

Debido al ruido de medida que presentan algunas señales, tales como la velocidad angular y el par, es necesario emplear un proceso de filtrado digital. Por ello, en el epígrafe 5.3 se describe el procedimiento a seguir para implementar filtros discretos en un microcontrolador, analizando múltiples alternativas per focalizando en un filtro en particular ampliamente utilizado, el filtro Butterworth. Además, se describirá un método de identificación y eliminación de *outliers* en tiempo real conocido por filtro Hampel, cuya principal aplicación en el proyecto se halla en el filtrado de la corriente generada por el motor de imanes permanentes.

Finalmente, se detallarán las diferentes tareas que ejecuta el microprocesador, desde el proceso de muestreo de datos, pasando por la transmisión de estos mediante *bluetooth* y MQTT y la visualización local de magnitudes en la pantalla LCD.

El objetivo consiste en implementar en un mismo programa todas las siguientes funcionalidades:

- Muestreo de señales que presenten diferentes pasos de muestreo.
- Envío de datos a través de bluetooth e internet a mediante el protocolo MQTT.
- Visualización de diferentes magnitudes en una pantalla LCD.
- Configuración de cualquier banco de ensayos a través de la aplicación Android. El programa debe almacenar los datos recibidos en la memoria interna para evitar la pérdida de información cada vez que se reinicie el dispositivo.
- Recepción de órdenes desde la aplicación Android. Se debe de poder comandar los variadores de frecuencia desde la propia aplicación.

Con el fin de no extender en exceso la redacción del documento, se ha optado por exponer todo el código desarrollado mediante Arduino IDE en un repositorio de GitHub de libre acceso [42].

5.1 PLACA DE DESARROLLO

En el presente proyecto se ha utilizado la placa de desarrollo ESP-WROVER-KIT del fabricante Espressif Systems, la cual lleva incorporado un microprocesador ESP32, pantalla LCD, múltiples pines de entrada/salida, alimentación vía 5V o USB, transmisión de datos a través de *bluetooth* o Wifi y diferentes interfaces de comunicación, como I2C o UART, entre otras características.

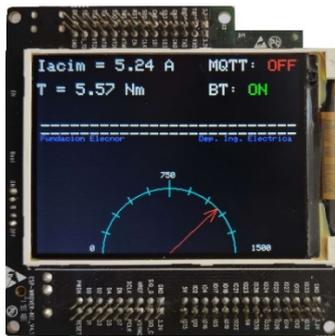


Figura 5.1a. Pantalla LCD.

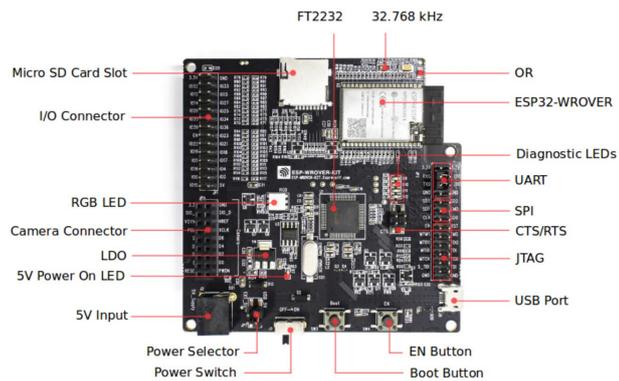


Figura 5.1b. Características [32].

Figura 5.1. Microcontrolador ESP32.

La pantalla LCD se ha empleado para desarrollar una visualización local de diferentes variables asociadas al banco de ensayos, cuya interfaz se expone en la figura 5.1a y cuyo desarrollo se expone en el apartado 5.8.

Esta placa presenta un total de 34 pines de entrada/salida, sin embargo, existen algunos de ellos que únicamente se pueden emplear como entradas. Además, si se desea emplear la pantalla LCD, 7 de los 34 pines se reservan para tal uso. Por otra parte, dos de los pines tienen reservado su uso para la gestión de la memoria ram, por lo que únicamente se disponen de 25 pines efectivos. Los pines reservados para interactuar con la pantalla LCD se exponen en la tabla 5.1.

Pin	Funcionalidad
GPIO 18	RESET
GPIO 19	SCL
GPIO 21	D/C
GPIO 22	CS
GPIO 23	SDA
GPIO 25	SDO
GPIO 5	Backlight

Tabla 5.1. Pines reservados para la interfaz LCD.

En el desarrollo del presente proyecto se han empleado un total de 3 entradas analógicas para efectuar la lectura de corrientes, en particular 2 corrientes de estátor y la corriente generada por el motor de imanes permanentes empleada en la lectura indirecta de par, 1 entrada digital empleada como interrupción con el fin de calcular la velocidad mecánica del eje y, finalmente, 8 salidas digitales y 2 salidas analógicas para comandar los variadores de frecuencia. El conjunto de entradas/salidas empleado en el proyecto se detallan en la tabla 5.2.

Función		Pin
Lectura de velocidad		GPIO 2
Lectura de corrientes de estator	Fase a	GPIO 36
	Fase b	GPIO 39
Medida indirecta de par		GPIO 35
Salidas digitales variador PMSM	Marcha/Paro	GPIO 4
	Reset	GPIO 27
	Selector Control	GPIO 15
Salidas digitales variador ACIM	Marcha/Paro	GPIO 0
	Inversión de giro	GPIO 13
	Consigna 1	GPIO 14
	Consigna 2	GPIO 12
Salida analógica PMSM		GPIO 25
Salida analógica ACIM		GPIO 26

Tabla 5.2. Pines de E/S empleados en el proyecto.

5.2 ENTORNO DE DESARROLLO ARDUINO IDE

Arduino IDE es un software de código abierto ampliamente utilizado para desarrollar y cargar código en cualquier placa de desarrollo de Arduino. Además, permite generar código para placas de otros fabricantes, siempre que existan las librerías correspondientes.

Previamente al desarrollo de cualquier código para la placa ESP-WROVER-KIT, es necesario configurar el entorno de desarrollo para que sea posible cargar programas en la citada placa. Para ello, dentro del gestor de URLs adicionales de tarjetas es necesario añadir el enlace siguiente:

- https://dl.espressif.com/dl/package_esp32_index.json

Finalmente, dentro de la ventana herramientas se selecciona la placa deseada, en este caso es Adafruit ESP32 Feather. A partir de este punto ya es posible desarrollar código para la tarjeta empleada en el proyecto.

Cualquier programa desarrollado en Arduino IDE presenta, por defecto, dos funciones principales:

- Setup. Únicamente se ejecuta una vez cuando se enciende la placa y permite configurar ciertas características del código. En el presente proyecto, el setup se emplea para definir los coeficientes del filtro digital, inicializar la pantalla LCD, configurar la transmisión de datos a través de *bluetooth*, crear las colas de transmisión de datos entre núcleos, asignar tareas a diferentes núcleos, indicar qué pines son de salida, leer datos de la memoria interna e inicializar variables.

- Loop. Se ejecuta indefinidamente durante el funcionamiento de la tarjeta. Permite implementar instrucciones que se ejecutan de forma periódica, como podría ser un proceso de muestreo de datos o la transmisión de estos a través de *bluetooth* o MQTT. En la solución propuesta únicamente se realiza el proceso de muestreo de datos.

5.3 FILTRADO DIGITAL DE SEÑALES

El proceso de filtrado consiste en procesar una determinada señal con el objetivo de atenuar el efecto del ruido de medida o bien para extraer ciertas componentes en frecuencia de una determinada señal [33].

Los filtros se clasifican en dos categorías principales:

- Filtros analógicos. Se diseñan mediante elementos físicos, principalmente resistencias y condensadores. El principal inconveniente de emplear este tipo de filtros radica en que es necesario modificar el circuito electrónico siempre que se desee cambiar alguna característica del filtro. Un caso típico de filtro analógico paso bajo se expone en la figura 5.2, cuya función de transferencia asociada se expone en la ecuación 5.1.

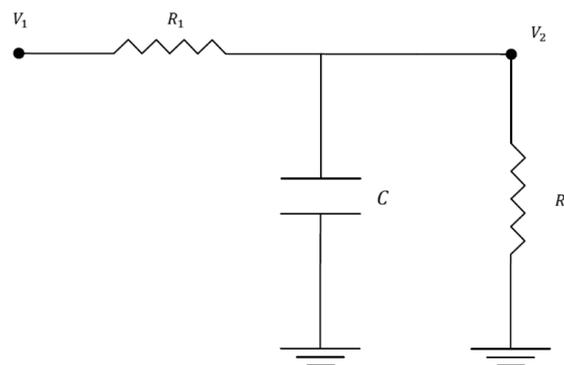


Figura 5.2. Filtro analógico.

Es posible observar que la construcción de este tipo de filtros es posible realizarla a partir de un divisor de tensión conectando un condensador entre la tensión de salida y masa.

$$(5.1) \quad v_{out} = \frac{R_2}{R_1 + R_2} \cdot \frac{1}{1 + \frac{R_1 \cdot R_2 \cdot C \cdot \omega j}{R_1 + R_2}}$$

- Filtros digitales. Se diseñan e implementan vía software en un microprocesador que se encarga de muestrear una determinada señal y aplicar el correspondiente proceso de filtrado. Se distinguen dos subcategorías:
 1. Filtros recursivos, también denominados en la literatura como filtros IIR (Infinite Impulse Response). En este tipo de filtros, el valor actual de la señal filtrada, denotado por $y(k)$, depende únicamente del valor actual de la señal no filtrada, denotada por $x(k)$, y de las muestras anteriores, es decir, los valores de salida de la función de transferencia discreta únicamente dependen de los valores de

entrada. La denominación de filtro IIR proviene del hecho que la función de transferencia discreta asociada al filtro digital presenta una respuesta finita frente a una entrada impulso

2. Filtros no recursivos, también conocidos como filtros FIR (Finite Impulse Response). En los filtros no recursivos, a diferencia de los anteriores, la salida actual depende no únicamente de valores actuales y pasados de la señal de entrada, sino que también depende de los valores anteriores de la propia señal de salida.

Se ha optado por emplear filtros digitales con el fin de diseñar un producto versátil, permitiendo modificar los diferentes filtros de una forma sencilla en el caso de que la solución propuesta se emplee en equipos relativamente distintos y sea necesario modificar los parámetros del filtro.

En el presente proyecto se empleará principalmente un filtro digital recursivo de tipo paso bajo ampliamente conocido, el filtro Butterworth y, adicionalmente, se aplicará el filtro Hampel como método de identificación y eliminación de *outliers* en tiempo real. Además, se han analizado alternativas como la aplicación de un filtro de media o mediana móvil.

La expresión general asociada a la función de transferencia discreta de un filtro digital [33], que relaciona la variable filtrada y sin filtrar se expone a continuación, donde hay que tener en cuenta que siempre se cumple que el primer coeficiente del vector a , es decir a_0 , es unitario.

$$(5.2) \quad G(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^n b_i \cdot z^{-i}}{\sum_{j=0}^m a_j \cdot z^{-j}} = \frac{\sum_{i=0}^n b_i \cdot z^{-i}}{1 + \sum_{j=1}^m a_j \cdot z^{-j}} = \frac{b_0 + b_1 \cdot z^{-1} + \dots + b_n \cdot z^{-n}}{1 + a_1 \cdot z^{-1} + \dots + a_m \cdot z^{-m}}$$

En el caso más general se cumple que $m \neq n$, de tal modo que el orden del filtro es el valor máximo entre m y n .

Mediante la expresión 5.2 y aplicando las propiedades de la transformada en Z , es posible determinar la ecuación en diferencias que permite calcular para cada instante de tiempo la señal discreta filtrada, detallada en la ecuación 5.3.

$$(5.3) \quad y(k) = \sum_{i=0}^n b_i \cdot x(k-i) - \sum_{j=1}^m a_j \cdot y(k-j)$$

5.3.1 FILTROS DE MEDIA Y MEDIANA MÓVIL

Son los filtros más sencillos de diseñar e implementar debido, principalmente, a que son filtros no recursivos. Consisten en calcular la media o mediana sobre una ventana de datos de un determinado tamaño [34]. Al ser procesos de filtrado en tiempo real, la ventana será móvil, es decir, se debe desplazar en cada recepción de un nuevo dato.

- Filtro de media móvil

Teniendo en cuenta que se trata de un filtro no recursivo, la muestra filtrada tomada en el instante actual únicamente dependerá de los datos no filtrados. En consecuencia, todos los coeficientes del vector a serán nulos, excepto el coeficiente a_0 que será unitario. Por otra parte, todos los coeficientes del vector b tienen un valor de $1/L$, siendo L la longitud de la ventana móvil cuyo valor es $L = n + 1$. La implementación del filtro se logra mediante la ecuación 5.4.

$$(5.4) \quad y(k) = \sum_{i=0}^n b_i \cdot x(k-i) = \sum_{i=0}^{L-1} \frac{1}{L} \cdot x(k-i) = \frac{1}{L} \sum_{i=0}^{L-1} x(k-i)$$

- Filtro de mediana móvil

Este tipo de filtro es relativamente similar al caso anterior. Se debe definir una ventana móvil respecto de la cual se aplicará el filtro no recursivo. En este caso se aplica la mediana sobre el conjunto de datos que componen dicha ventana. Cabe destacar que el cálculo de la mediana móvil en cada instante de tiempo no es inmediato a diferencia de como ocurre en el filtro de media móvil.

$$(5.5) \quad y(k) = \text{median}(\{x(k)\})$$

En este caso es necesario ordenar de forma ascendente o descendiente todos los datos que forman la ventana móvil, de modo que la mediana será aquel valor que se encuentre en el percentil 50%.

En cuanto a las diferencias entre estos dos filtros, se evidencia una mayor dificultad en la implementación de los filtros de mediana móvil, ya que el reordenamiento de la ventana se debe de realizar en tiempo real con el fin de optimizar el algoritmo, es decir, no se debe de reordenar toda la ventana móvil en cada iteración ya que se incrementaría en gran medida el coste computacional. Sin embargo, la mediana es un estimador de tendencia robusto ya que es menos sensible a *outliers* que la media.

La principal desventaja común a ambos filtros consiste en que es necesario aplicar el proceso de filtrado a señales senoidales que tienen una frecuencia variable. Por una parte, se debe de muestrear la corriente consumida por dos de las fases del ACIM y, por otro lado, se realiza la lectura de la corriente generada por el PMSM. En ambos casos la frecuencia de las señales es variable debido a la alimentación de los motores a través del variador de frecuencia correspondiente.

Hay que tener en cuenta que la media y mediana de una señal senoidal tomada a lo largo de un periodo completo son nulas. Al operar a frecuencia variable, no es posible definir un tamaño de la ventana móvil adecuado ya que puede ocurrir que una longitud proporcione buenos resultados a una frecuencia fija, pero en el momento en el que se opere a una frecuencia distinta puede ocurrir que la longitud definida se corresponda a datos muestreados a lo largo de un periodo completo, ofreciendo constantemente un valor nulo.

5.3.2 FILTRO HAMPEL

Se trata de un tipo de filtro muy particular cuyo objetivo consiste en identificar y eliminar *outliers* presentes en el muestreo de datos de forma *online* [35][36]. Se evidencian dos procesos distintivos: la identificación de un *outlier* y su eliminación.

- Identificación de un *outlier*

Este proceso se implementa mediante el conocido por identificador Hampel, el cual permite identificar un outlier si difiere en cierta medida de la tendencia habitual de la señal. Dicha tendencia se mide a partir de la mediana y la desviación mediana absoluta (MAD) escalada. La desviación mediana absoluta se define como sigue.

$$(5.6) \quad MAD = \text{median}(\{|x(k) - X_{med}|\}), \quad X_{med} = \text{median}(\{x(k)\})$$

Seguidamente, se calcula la desviación mediana absoluta escalada, σ , que se corresponde con una estimación de la desviación estándar de una distribución normal. Cabe destacar que este valor, al calcularse de forma indirecta a través del parámetro MAD, se ve menos afectado por la presencia de *outliers* que la desviación típica estándar calculada a partir de la media.

$$(5.7) \quad \sigma = K \cdot MAD, \quad K = 1.4826$$

En cuanto a la identificación propiamente dicha de un *outlier*, si la diferencia entre el valor actual y la mediana difiere en un determinado número de veces, denotado por n , la desviación mediana absoluta escalada, entonces dicha muestra se identifica como un *outlier*. Esta identificación se efectúa según la ecuación 5.8.

$$(5.8) \quad \text{outlier}(x(k)) = \begin{cases} \text{false}, & \text{si } |x(k) - X_{med}| \leq n \cdot \sigma \\ \text{true}, & \text{si } |x(k) - X_{med}| > n \cdot \sigma \end{cases}$$

Como alternativa, se podrían emplear la media y desviación típica, pero tal como se ha expuesto en el apartado 5.3.1, estos valores se ven afectados por los propios *outliers*, por lo que no serían adecuados para identificarlos.

- Eliminación de un *outlier*

El filtro Hampel por definición implementa la eliminación de un *outlier* mediante su sustitución por la mediana de la ventana de datos tal como se detalla en la ecuación 5.9. Sin embargo, existen variaciones del filtro que implementan esta eliminación de *outliers* de forma diferente. Por ejemplo, en el caso de señales senoidales de frecuencia variable sería más útil sustituir el *outlier* identificado por el valor medio de sus dos muestras adyacentes.

$$(5.9) \quad y(k) = \begin{cases} x(k), & \text{si } |x(k) - X_{med}| \leq n \cdot \sigma \\ X_{med}, & \text{si } |x(k) - X_{med}| > n \cdot \sigma \end{cases}$$

La ventaja de este tipo de filtro frente a los de media y mediana móvil consiste en que no sustituyen todos los datos de la ventana móvil, sino que únicamente afecta a aquellos valores que son identificados como *outliers*. Por este motivo, el filtro Hampel podría tener aplicación en señales de frecuencia variable. La validación del filtro Hampel previa a su implementación en el microcontrolador se detalla en el anexo 3.

5.3.3 FILTRO BUTTERWORTH

Por los motivos citados en el apartado anterior y debido a las múltiples ventajas que ofrece el filtro Butterworth, no se han implementado los filtros de media y mediana móvil, centrandos los esfuerzos en el diseño de una función que permita aplicar el filtro Butterworth a cualquier señal de entrada.

A partir de la expresión 5.3, teniendo en cuenta que en el filtro Butterworth se cumple que $m = n$, es posible determinar la ecuación en diferencias que permite calcular para cada instante de tiempo la señal discreta en un filtro Butterworth. En consecuencia, la ecuación 5.10 es la que se debe de implementar en el microprocesador.

$$(5.10) \quad y(k) = \sum_{i=0}^n b_i \cdot x(k - i) - \sum_{j=1}^n a_j \cdot y(k - j)$$

En primer lugar, se ha diseñado y validado el filtro digital en Matlab, cuyo código se expone en el anexo 3, obteniendo los coeficientes del numerador y denominador asociados a la función de transferencia discreta. Para ello se han empleado las siguientes funciones:

- *butter*($n, fc/(fs/2)$). Devuelve los coeficientes a y b que permiten la implementación del filtro digital. Requiere como parámetros de entrada el orden del filtro, la frecuencia de corte deseada y la frecuencia de muestreo de la señal sobre la que se aplica el proceso de filtrado.
- *filter*(b, a, x). Implementa de forma recursiva la ecuación 5.3, es decir, devuelve como variable de salida la secuencia de valores asociados a la señal filtrada. Cabe destacar que esta función no limita su uso únicamente al filtro *Butterworth*.

Mediante el empleo de Matlab es posible realizar un análisis en frecuencia de la función de transferencia discreta (ecuación 5.2) frente a diferentes frecuencias de corte y diferente orden del filtro digital. Este análisis se expone en las figuras 5.3 y 5.4.

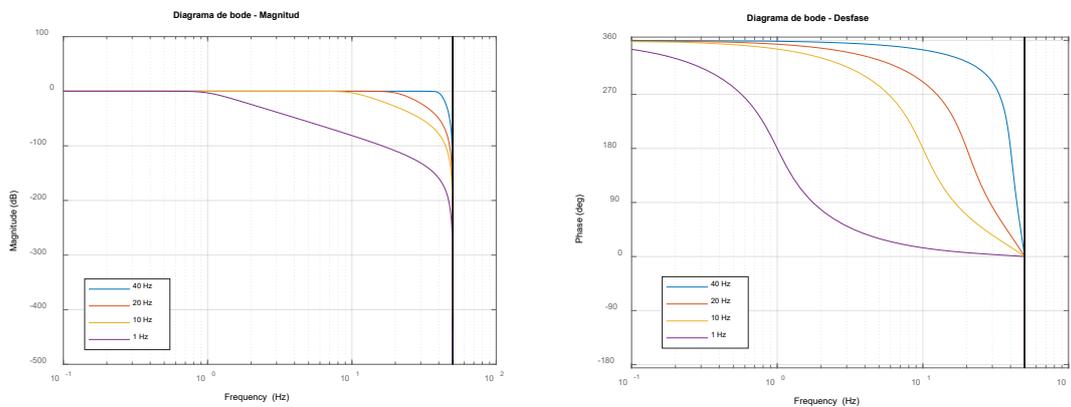


Figura 5.3. Efecto de la frecuencia de corte para un orden fijo - $fs=100Hz$.

En primer lugar, en la figura anterior se evidencia una característica inherente a cualquier sistema discreto: la respuesta en frecuencia de su función de transferencia discreta está limitada a la mitad de la frecuencia de muestreo de la señal. En cuanto al efecto de la frecuencia de corte, se observa que cuanto menor sea el valor de f_c mayor es la atenuación producida sobre las diferentes componentes en frecuencia de la señal, es decir, se producirá una mayor atenuación del ruido de medida, pero mayor será el desfase entre las señales de entrada y de salida.

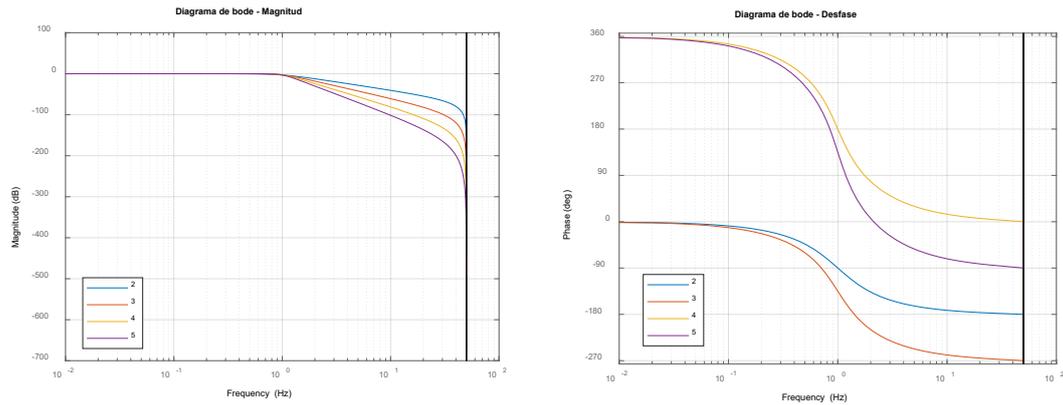


Figura 5.4. Efecto del orden para una frecuencia de corte fija - $f_s=100\text{Hz}$.

En la figura 5.4 es posible observar que un incremento en el orden del filtro se traduce en un aumento en la pendiente del diagrama de bode asociado, es decir, una determinada componente en frecuencia será atenuada en mayor medida cuanto mayor sea el orden del filtro. Sin embargo, un orden alto implica un mayor retraso de la señal de salida.

Para la obtención de los resultados anteriores, se ha desarrollado un *script* de Matlab que permite realizar estos análisis de forma automática simplemente a partir de las frecuencias de corte y el orden del filtro dados por el usuario. Además, dicho *script* permite aplicar y validar el proceso de filtrado digital a cualquier señal y analizar el efecto de emplear diferentes frecuencias de corte y orden del filtro de forma totalmente automática.

En cuanto a la implementación del filtrado digital en el microprocesador y frente a la ausencia de librerías que permitan aplicar el filtro de tipo Butterworth, se ha desarrollado en Arduino IDE una función que permite aplicar un filtro Butterworth a cualquier señal discreta que se desee. Para ello, dicha función recibe como parámetros los coeficientes a y b del filtro y los valores asociados a las señales filtrada y original.

5.4 GESTIÓN DE MÚLTIPLES TAREAS

En los restantes epígrafes, se procede a describir las diferentes funcionalidades implementadas en el microprocesador ESP32.

Además del propio *loop*, se han definido tres tareas adicionales que permiten implementar el conjunto de funciones necesarias. Más en concreto, las tareas e instrucciones asociadas que deben ser ejecutadas se detallan mediante un diagrama de flujo en la figura 5.5.

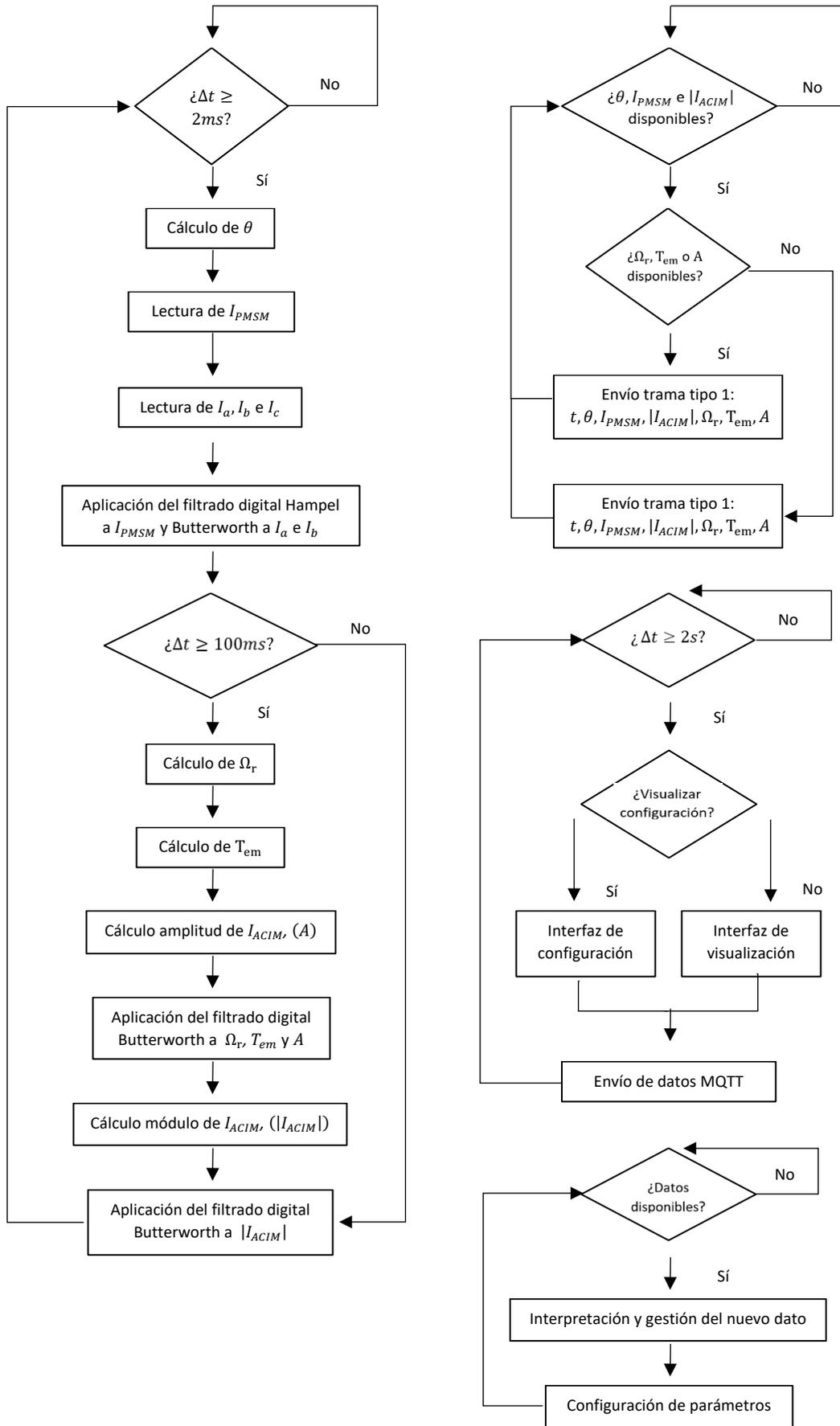


Figura 5.5. Diagrama de flujo general de las tareas definidas.

En resumen, las tareas implementadas son las siguientes:

- Proceso de muestreo de datos. Se ejecuta en el núcleo 1 y, en particular, se encarga de muestrear las corrientes de dos de las fases del estator del motor de inducción, la corriente generada por el motor de imanes permanentes, el número de pulsos emitidos por el encoder y calcular la velocidad mecánica y el par.
- Transmisión de datos mediante *bluetooth*. Se ejecuta en el núcleo 0 y se encarga de enviar a la aplicación Android los diferentes datos muestreados. Cabe destacar que este proceso de envío de datos debe permitir la transmisión de señales que tienen diferentes pasos de muestreo. Por una parte, las corrientes y posición angular presentan un paso de muestreo de 2ms, mientras que en el del par y la velocidad es de 100ms en ambos casos.
- Visualización en la pantalla LCD y envío de datos mediante el protocolo MQTT. Se ha implementado en el núcleo 0. Es necesario visualizar en la pantalla diferentes magnitudes, como la corriente consumida por el motor de inducción, el par y la velocidad a través de un indicador semicircular. Además, se visualiza el estado de la transmisión de datos mediante *bluetooth* y Wifi. Finalmente, también es posible visualizar parte de los parámetros de configuración almacenados en la memoria interna. La tasa de refresco de la pantalla LCD, así como la tasa de envío de datos a través de MQTT, se ha fijado en 2 segundos.
- Recepción e interpretación de datos recibidos mediante *bluetooth*. Permite, en primer lugar, recibir órdenes a través de la aplicación Android, tales como el inicio del proceso de envío de datos mediante *bluetooth* y el control de los variadores de frecuencia. Además, se encarga de interpretar todos los datos asociados a la configuración del banco de ensayos.

Para asignar una tarea a un núcleo concreto, previamente es necesario definir la función que va a ser ejecutada por el núcleo deseado que, habitualmente, debe ser un bucle infinito. Una vez programada la citada función, ésta es asignada a un determinado núcleo mediante la instrucción `xTaskCreatePinnedToCore`.

5.5 ADQUISICIÓN DE DATOS

El proceso de muestreo de señales se ha implementado en el núcleo 1 y, en particular, se ejecuta en el *loop* principal. Los pines empleados en la adquisición de datos, así como su lectura asociada, se exponen en la tabla 5.3.

Funcionalidad		Pin
Interrupción por hardware		GPIO 2
Lectura de corriente PMSM		GPIO 35
Lectura de corrientes ACIM	Fase a	GPIO 36
	Fase b	GPIO 39

Tabla 5.3. Pines empleados en el muestreo de datos.

5.5.1 LECTURA DE POSICIÓN Y VELOCIDAD

Las lecturas de posición y velocidad se han realizado a partir de la contabilización de los pulsos generados por el encoder. Dicho conteo de pulsos se determina mediante interrupción por hardware, de tal modo que cada vez que la señal digital de entrada pase de nivel bajo a nivel alto se incrementará una variable que contabilizará el número de pulsos emitidos.

En cuanto a la definición de interrupciones por hardware en el microcontrolador, se emplea la instrucción `attachInterrupt(GPIO,ISR,Modo)`, donde sus parámetros de entrada son el pin en el que se desea configurar la interrupción, el nombre de la función que se debe ejecutar cada vez que se produzca la interrupción y, finalmente, el modo de funcionamiento, es decir, bajo qué condiciones que debe de ejecutar la interrupción, que pueden ser los siguientes:

Modo de funcionamiento	Condición
HIGH	La interrupción se produce cuando la señal de entrada se encuentre a nivel alto.
LOW	La interrupción se produce cuando la señal de entrada se encuentre a nivel bajo.
RISING	La interrupción se produce cuando la señal de entrada pasa de nivel bajo a nivel alto.
FALLING	La interrupción se produce cuando la señal de entrada pasa de nivel alto a nivel bajo.
CHANGE	La interrupción se produce cuando la señal de entrada cambia de nivel alto a nivel bajo o viceversa.

Tabla 5.4. Modos de funcionamiento de una interrupción.

En particular, se han definido dos variables, $cont_p$ y $cont_v$, que actúan como contadores de pulsos de modo que la primera de ellas permite obtener el registro de posición mientras que la segunda se emplea para calcular la velocidad. De este modo, la variable $cont_p$ únicamente toma valores comprendidos en el intervalo $[0, 2^n - 1]$, mientras que la variable $cont_v$ puede incluir varias revoluciones consecutivas.

La posición angular se determina mediante la expresión 5.11. Cabe destacar que las posiciones angulares asociadas a 0° y 360° son físicamente equivalentes pese a que matemáticamente sean valores distintos, por lo que se requiere que la variable θ_{mec} no pueda tomar ambos valores durante la ejecución del programa. Por ello, se ha diseñado el código de forma que el valor máximo de la variable $cont_p$ sea $2^n - 1$, por lo que la posición angular nunca tomará el valor de 360° .

$$(5.11) \quad \theta_{mec} = \frac{cont_p}{2^n} \cdot 360, \quad \theta_{mec,máx} = \frac{2^n - 1}{2^n} \cdot 360$$

El cálculo de la velocidad se realiza contabilizando el número de pulsos a lo largo de un determinado intervalo de tiempo, denotado por Δt , a partir de la ecuación 5.12.

$$(5.12) \quad \Omega_r = 60 \cdot \frac{cont_v}{2^n \cdot \Delta t}$$

Si se optase por emplear una única variable de tipo contador, calculando la velocidad a partir de dos posiciones diferentes, habría que tener en cuenta que la lectura de velocidad se debería de realizar a partir de dos posiciones que difieran menos de una revolución.

Por otra parte, cuanto menor es el paso de muestreo asociado a la velocidad, más notorio es el efecto de la resolución del encoder y más se evidencia la necesidad de aplicar un proceso de filtrado digital. Empleando, por ejemplo, un paso de muestreo de 2ms, se obtienen los resultados que se exponen en la figura 5.6, en la que, además, se ha aplicado un filtro Butterworth de cuarto orden comparando el efecto de la reducción de la frecuencia de corte.

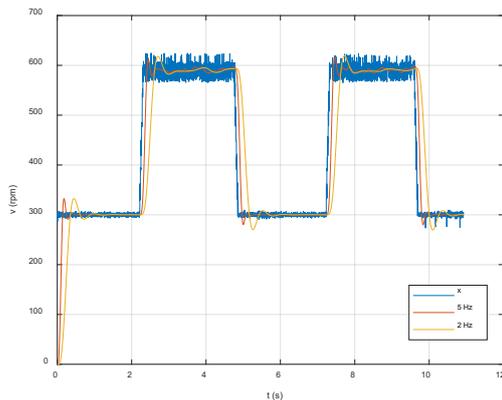


Figura 5.6a. Comparación señal original y filtrada.

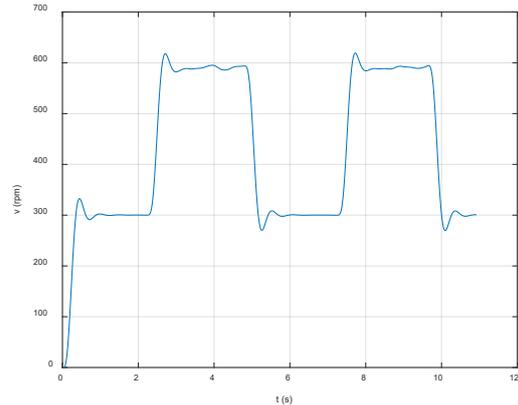


Figura 5.6b. Señal filtrada

Figura 5.6. Lectura de velocidad.

Tal como se había expuesto en el apartado 5.3, una reducción en la frecuencia de corte implica realizar un diseño del filtro más agresivo, obteniendo una señal con menos rizado, pero retrasando la señal de salida. Finalmente, se ha optado por un paso de muestreo de 100ms y un filtro Butterworth de frecuencia de corte 1Hz y orden 4.

5.5.2 LECTURA DE PAR

Tal como se ha expuesto en el capítulo anterior, se ha implementado una lectura indirecta del par a partir del cálculo del valor medio cuadrático asociado a la corriente generada por el motor de imanes permanentes.

Previamente al cálculo del par propiamente dicho, es necesario obtener una serie de variables implicadas en dicho cálculo y que han sido descritas en el epígrafe 4.3.2. El conjunto de cálculos que se debe realizar se detalla en la figura 5.7.

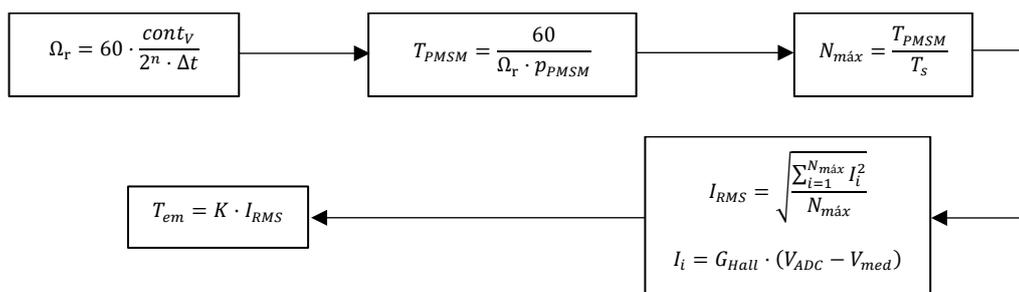


Figura 5.7. Cálculos implicados en la lectura indirecta de par.

Cada 2ms se necesario muestrear la corriente del PMSM con el objetivo de calcular el numerador de I_{RMS} de forma recursiva. Se ha analizado la posibilidad de filtrar únicamente la corriente o aplicar un posterior filtrado al par si la señal obtenida no ofrece una calidad suficiente.

Durante las pruebas experimentales, debido a que la frecuencia de la corriente es variable y que ha sido necesario emplear una frecuencia de corte relativamente cercana a la máxima frecuencia de operación del PMSM (100Hz), se producía una atenuación indeseada cuando el motor operaba a frecuencias cercanas a 100Hz. En definitiva, no es posible aplicar un filtro de tipo Butterworth ni tampoco los filtros de media y mediana móvil ya que la frecuencia de la corriente es variable.

Como alternativa se ha aplicado el filtro Hampel detallado en el apartado 5.3.2, pero modificando la sustitución de un *outlier* por el valor medio de sus muestras adyacentes en lugar de la mediana. Los resultados se exponen en la figura 5.8.

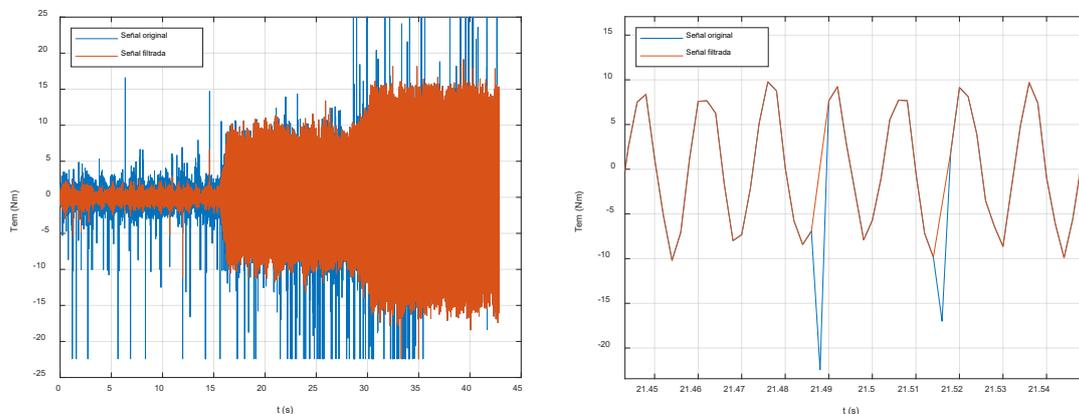


Figura 5.8. Aplicación del filtro Hampel.

Los resultados anteriores se han obtenido alimentando el motor a través el variador de frecuencia. Si se alimenta el motor mediante conexión directa a red la relación señal/ruido se incrementa notablemente e incluso no sería necesario aplicar un proceso de filtrado digital, tal como se detallará en el capítulo 7. Sin embargo, tal como se observa en la figura 5.8, la cantidad de ruido introducida por el variador empeora en gran medida la calidad de la lectura, por lo que se ha aplicado el filtrado digital en cualquier caso.

Seguidamente, cada 100ms, se calcula la velocidad mecánica a partir de la cual se determina el periodo de la corriente del PMSM que, a su vez, permite obtener el número de muestras de corriente que se deben tomar para completar un periodo de la señal.

Finalmente, cabe resaltar que el empleo de un paso de muestreo de 100ms en el cálculo del par introduce una limitación a tener en cuenta: no será posible calcular el valor medio cuadrático de corrientes cuyo periodo sea superior a los 100ms.

Además, sobre el par se ha aplicado el filtro Butterworth con una frecuencia de corte de 0.3Hz, obteniendo los resultados que se detallan en la figura 5.9. La señal filtrada parte de 0 ya que el algoritmo inicializa a 0 las muestras anteriores al instante inicial.

Tal como ocurría en el caso de la figura 5.8, la conexión mediante el variador de frecuencia empeora la relación señal ruido en comparación con la conexión directa a red, siendo necesario un posterior filtrado aunque la señal de la corriente haya sido previamente filtrada.

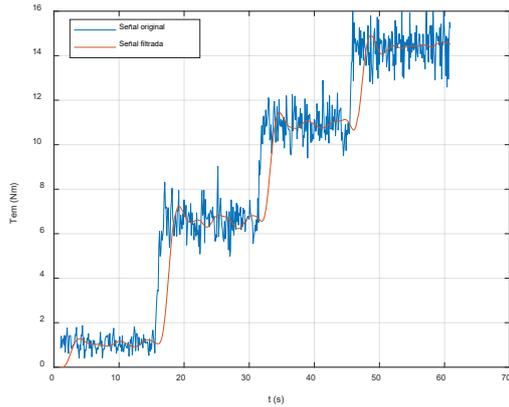


Figura 5.9a. Par – Conexión directa a red.

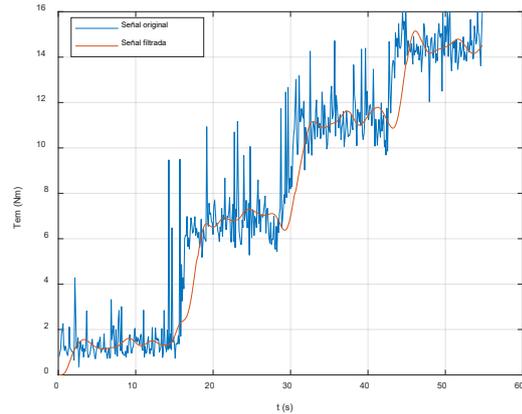


Figura 5.9b. Par – Conexión a través de variador.

Figura 5.9. Lectura de par.

5.5.3 LECTURA DE CORRIENTES DE ESTATOR

Se requiere obtener las corrientes de las tres fases del estator del motor de inducción. Sin embargo, únicamente es necesario muestrear dos de ellas ya que es posible determinar la tercera mediante el empleo de la ecuación 5.13.

$$(5.13) \quad I_a + I_b + I_c = 0 \rightarrow I_c = -I_a - I_b$$

En un primer planteamiento, se propuso enviar las tres corrientes a la aplicación Android. El principal inconveniente de esta metodología consiste en que no es posible comparar de una forma efectiva el nivel de corriente consumida por diferentes motores y además el hecho de enviar tres variables adicionales a través de *bluetooth* dificulta lograr un paso de muestreo de 2ms.

Como alternativa se ha optado por representar el nivel de corriente a través del módulo y la amplitud de la corriente de estator consumida por el ACIM. Sin embargo, en cualquier caso, es necesario realizar la lectura de la corriente de las fases del estator del ACIM, requiriendo la aplicación de un filtrado sobre éstas.

- Lectura de corrientes de fase

Tal como se ha expuesto, únicamente es necesario efectuar la lectura de dos de las fases del estator, siendo necesario aplicar un proceso de filtrado sobre ambas debido al ruido de medida que presentan estas, especialmente cuando se alimenta el motor a través del variador.

Se ha propuesto aplicar el filtro Butterworth con un paso de muestreo de 2ms y una frecuencia de corte de 60Hz, siendo este valor suficiente para atenuar el ruido de medida. Para validar esta elección, se ha obtenido un registro de datos experimentales operando el motor en un rango amplio de frecuencias, desde el reposo hasta los 50Hz. Esto permite comprobar si se produce una indeseada atenuación de la señal, tal como ocurría con la lectura de la corriente generada por el PMSM.

En la figura 5.10 se comparan resultados experimentales obtenidos mediante conexión del motor a través del variador, aplicando el proceso de filtrado digital a las frecuencias de corte de 51Hz y 60Hz. Se observa que en el primer caso se produce una atenuación de la señal a medida que su frecuencia se aproxima a los 50Hz. Sin embargo, empleando una frecuencia de corte de 60Hz ofrece buenos resultados sin afectar a la señal.

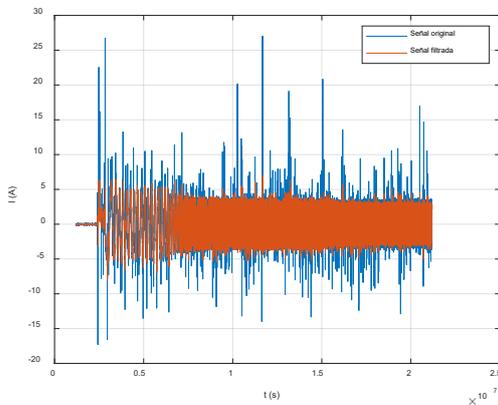


Figura 5.10a. Corriente de fase – $f_c = 51\text{Hz}$.

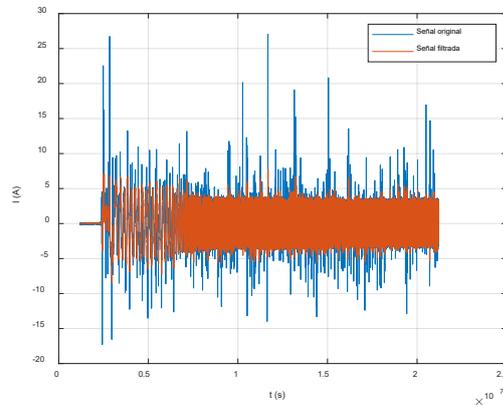


Figura 5.10b. Corriente de fase – $f_c = 60\text{Hz}$.

Figura 5.10. Lectura de corriente de fase mediante conexión a través de variador.

- Cálculo del módulo del fasor de corriente

En cuanto al módulo, es necesario aplicar la transformada directa de Clarke y calcular el módulo de la corriente a partir de sus componentes α, β de modo que será este único valor el enviado a la aplicación Android y no el valor concreto de corriente de cada una de las fases. El cálculo del módulo de la corriente del ACIM se efectúa a partir de las siguientes expresiones:

$$(5.14) \quad I_{ACIM} = \sqrt{I_{\alpha}^2 + I_{\beta}^2}, \quad \begin{aligned} I_{\alpha} &= \frac{2 \cdot I_a - I_b - I_c}{3} \\ I_{\beta} &= \frac{\sqrt{3}}{3} \cdot (I_b - I_c) \end{aligned}$$

Cabe destacar que en la lectura de las corrientes de estator el efecto del ruido de medida era muy evidente, especialmente en el caso de la conexión a través del variador de frecuencia, de tal modo que no era posible visualizar de forma correcta el módulo de la corriente. En consecuencia, se hace evidente la necesidad de aplicar un proceso de filtrado digital. En este caso el filtro Butterworth ha presentado buenos resultados, permitiendo una correcta representación del módulo de la corriente.

En la figura 5.11 se exponen datos experimentales relativos al cálculo del módulo de la corriente, obtenidos mediante conexión directa a red y a través del variador de frecuencia. Cabe destacar que estos cálculos se han realizado sobre las corrientes de cada fase filtradas. Se evidencia que la cantidad de ruido introducido al alimentar el motor a través del variador es mucho más evidente que cuando se conecta directamente a la red, pese a que en ambos casos se ha aplicado el filtro sobre las corrientes I_a e I_b . En definitiva, es necesario aplicar un filtrado adicional sobre el módulo de la corriente de estator.

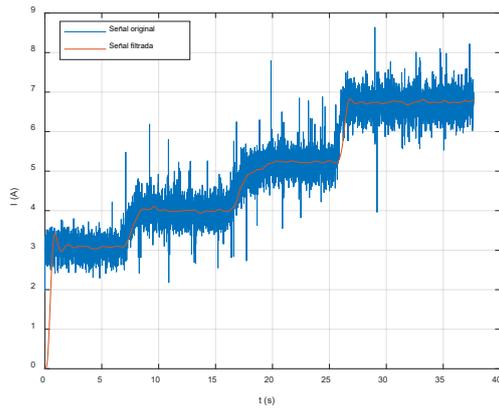


Figura 5.11a. Conexión directa a red.

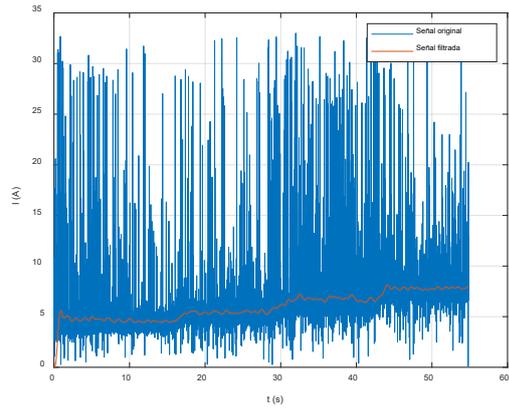


Figura 5.11b. Conexión mediante variador.

Figura 5.11. Lectura del módulo de la corriente.

El filtro Butterworth se ha diseñado empleando un paso de muestreo de 2ms y una frecuencia de corte de 1Hz, posibilitando una lectura aceptable del módulo de la corriente en ambos casos, tal como se expone en la figura 5.12.

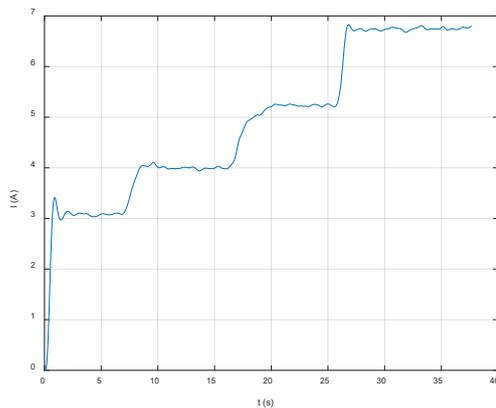


Figura 5.12a. Conexión directa a red.

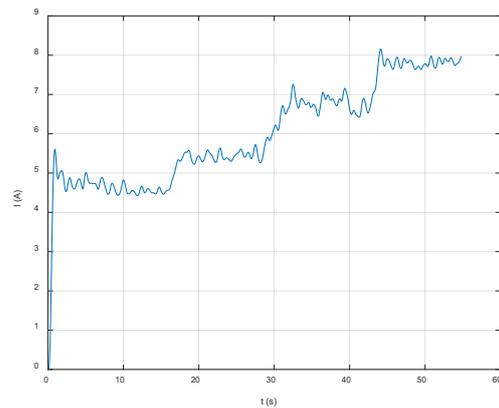


Figura 5.12b. Conexión mediante variador.

Figura 5.12. Filtrado del módulo de la corriente.

- Cálculo de la amplitud de la corriente

Respecto a la amplitud de la corriente, se han empleado las señales I_a e I_b para su cálculo ya que son las variables medidas directamente mediante sensores e I_c no ofrece información adicional. El algoritmo que permite obtener la amplitud de una señal senoidal consiste en obtener el máximo valor de la señal a lo largo de un periodo.

Sin embargo, el ruido presente en las corrientes de fase afecta a la obtención del valor máximo de la señal. Para mitigar este efecto, se ha optado por dividir la señal en el conjunto de periodos que se pueden integrar en un paso de muestreo de 100ms. Por ejemplo, si el periodo de la señal es de 20ms, el algoritmo determina el valor máximo de cada uno de los 5 periodos disponibles en dichos 100ms. Seguidamente, se obtiene el valor mínimo de estos máximos, logrando evitar que el ruido de medida afecte a la lectura de amplitud. Este proceso, se aplica sobre las corrientes I_a e I_b , calculando finalmente el mínimo de los dos valores obtenidos.

El periodo de la señal se calcula de forma aproximada mediante la ecuación 5.15. Esta expresión no es exacta ya que se está trabajando con un motor asíncrono, de modo que dicha ecuación estrictamente sólo es válida cuando opera en vacío, en cuyo caso la velocidad mecánica y de sincronismo son prácticamente iguales. Al operar en carga, el periodo calculado es ligeramente superior al real, por lo que se muestrearía un periodo completo. Sin embargo, con el objetivo de calcular el valor máximo sería suficiente con capturar únicamente un semiperiodo.

$$(5.15) \quad T_{ACIM} \approx \frac{60}{\Omega_r \cdot p_{ACIM}}$$

Cabe destacar que, si el periodo de la señal senoidal es superior al paso de muestreo de 100ms, únicamente se calcula el valor máximo de la fracción de periodo que se haya capturado. Esto introduce una limitación en la mínima frecuencia a la que puede operar el variador con el fin de poder muestrear al menos un semiperiodo y calcular correctamente la amplitud. Concretamente, la mínima frecuencia de la señal debe ser de 5Hz.

El algoritmo presentado ha ofrecido buenos resultados operando tanto en carga como en vacío. El hecho de calcular el valor mínimo de una serie de valores máximos, empleando ambas corrientes I_a e I_b , permite realizar una lectura aceptable de la amplitud incluso sin realizar ningún proceso de filtrado sobre las corrientes de fase ni sobre la propia amplitud. En la figura 5.13 se exponen los resultados asociados a la lectura de amplitud.

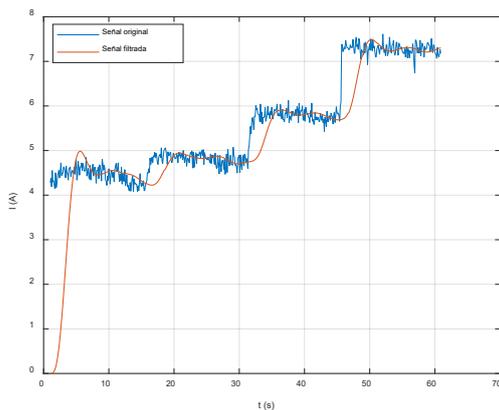


Figura 5.13a. Conexión directa a red.

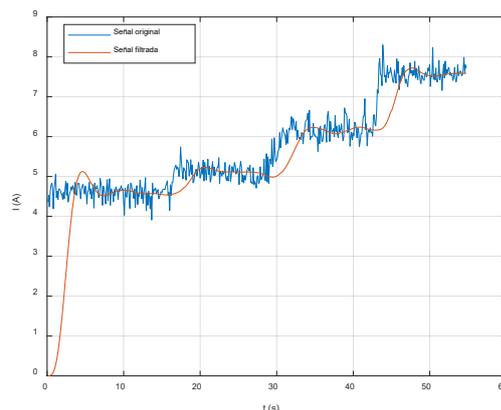


Figura 5.13b. Conexión mediante variador.

Figura 5.13. Lectura de la amplitud de la corriente.

Finalmente, se ha empleado un filtro Butterworth con una frecuencia de corte de 10Hz, ofreciendo una lectura de amplitud de mayor calidad.

La implementación en el microcontrolador de la lectura de todas las variables implicadas en el proceso de adquisición de datos se detalla en [42].

5.6 TRANSMISIÓN DE DATOS MEDIANTE BLUETOOTH

A continuación, se describe el proceso de transmisión de datos a través de *bluetooth* entre el microprocesador y la aplicación Android. Dicha transmisión debe permitir enviar un registro de datos correspondiente a la posición, velocidad angular, corriente generada por el motor de imanes permanentes, corriente consumida por el motor de inducción y el par transmitido al eje.

Por otra parte, se debe permitir enviar órdenes desde la aplicación al ESP32, tales como comandar los variadores de frecuencia y configurar un conjunto de parámetros internos del código Arduino y que dependen del banco de ensayos.

5.6.1 ENVÍO DE UN REGISTRO DE DATOS

Tal como se ha comentado anteriormente, cada registro de datos está formado por un conjunto de cinco variables: posición y velocidad angular, corrientes consumidas por ambos motores y el par transmitido al eje.

Además, no todas las variables presentan el mismo paso de muestreo. Por una parte, la posición angular y las corrientes tienen un paso de 2ms, mientras que el par y la velocidad, un paso de 100ms. En consecuencia, la transmisión de datos debe ser lo suficientemente robusta para permitir registrar señales de diferentes pasos de muestreo.

El envío de variables se realiza en tramas o paquetes de datos. Se evidencia que las variables de posición y corrientes deben estar presentes en cualquier paquete de datos, mientras que la velocidad y el par únicamente estarán disponibles en aquellos paquetes que se envíen cada 100ms. Por tanto, se definen dos tipos de paquetes de datos, cuya estructura se representa a continuación:

- Paquete 1: *, tiempo posición I_{ACIM} I_{IM} velocidad par#*
- Paquete 2: *, tiempo posición I_{ACIM} I_{IM} #*

Se observa que el inicio de cada paquete de datos viene indicado por el carácter “,”, mientras que el final de cada paquete viene definido por “#”.

Cabe recordar que el proceso de muestreo de datos y su envío mediante *bluetooth* se realiza en núcleos diferentes. Como método de transmisión de variables entre núcleos se han empleado colas de transmisión de datos. En particular, se han definido 6 colas, una por cada variable incluyendo el instante de tiempo actual. En cuanto a las colas asociadas a la posición y corrientes, se ha empleado el parámetro *portMAX_DELAY*, con el objetivo de que el programa quede en espera hasta que haya algún dato disponible en la cola. Sin embargo, las colas de la velocidad y el par no deben permanecer en espera ya que no todos los paquetes disponen de estas dos variables. Seguidamente, en función de si las colas de velocidad y par disponen de un dato se define un paquete de datos u otro.

El inicio de un nuevo registro de datos, así como la finalización de este, se realiza a través de la recepción de un 1 o un 0, respectivamente.

5.6.2 CONTROL DE LOS VARIADORES DE FRECUENCIA

Mediante el empleo de *bluetooth* se debe poder comandar los variadores de frecuencia del motor de inducción y del motor de imanes permanentes. Las órdenes asociadas a cada variador son las siguientes:

- Variador motor de inducción: marcha/paro, inversión de giro y consignas de frecuencia mediante dos entradas digitales y a través de una entrada analógica.

- Variador motor de imanes permanentes: marcha/paro, reset, selector del tipo de control y consigna de par mediante entrada analógica.

El control de las salidas digitales del ESP32 es relativamente sencillo. Todas las órdenes vienen precedidas por un carácter que indica qué salida o conjunto de salidas se va a controlar. Seguidamente, se recibe un 1 o 0 lógicos indicando si la salida se debe de poner a nivel alto o bajo, respectivamente.

El control de las salidas analógicas es un proceso más complejo ya que es necesario enviar un valor digital comprendido entre 0 y 255. Dicho valor se traducirá en una salida de voltaje comprendida entre 0 y 3.3V gracias al convertidor DAC del microcontrolador. Hay que tener en cuenta que la transmisión de datos mediante *bluetooth* se realiza carácter a carácter, por lo que, en función del número de cifras que tenga el valor digital recibido, el tratamiento de los datos debe ser diferente en cada caso tal como se detalla a continuación:

- Valor de tres cifras. El primer dato recibido debe ser multiplicado por 100 y el segundo por 10, mientras que el tercero se mantiene inalterado. Finalmente, se suman los tres valores.
- Valores de dos cifras. El primero se multiplica por 10 y se suman ambos números.
- Valor de una cifra. Es enviado directamente al convertidor DAC del ESP32.

El proceso descrito se aplica del mismo para controlar ambas salidas analógicas.

5.6.3 CONFIGURACIÓN DE UN BANCO DE ENSAYOS

El producto diseñado se puede emplear sobre cualquier banco de ensayos del laboratorio. Hay que tener en cuenta que pueden existir ciertas variaciones entre un banco de ensayos y otro. Por una parte, en cada banco de ensayos dispone de un sensor de efecto Hall empleado para medir la corriente del motor de imanes permanentes, por lo que el valor real de cada resistencia del divisor diferirá de un banco a otro debido a las tolerancias, afectando a la lectura del par. Además, cada puesto de prácticas puede tener unos motores con características diferentes o emplear un encoder con una resolución diferente.

En consecuencia, se ha optado por diseñar un programa lo suficientemente versátil para poder configurar cualquier banco de ensayos sin necesidad de modificar el código de Arduino. Los parámetros modificables desde la aplicación Android son los siguientes:

- Resolución del encoder.
- Factor de proporcionalidad entre el par y el valor medio cuadrático de la corriente generada por el motor de imanes permanentes.
- Número de par de polos del motor de inducción.
- Número de par de polos del motor de imanes permanentes.

La resolución del encoder afecta directamente a las lecturas de posición y velocidad, mientras que el factor de proporcionalidad incide en el cálculo del par. Por otra parte, el número de par de polos del motor de inducción se emplea en el cálculo de la máxima velocidad a la que operará dicho motor, mientras que el par de polos del motor de imanes permanentes se emplea en el

cálculo del periodo de la corriente generada por este motor, es decir, afecta al intervalo de tiempo en el cual se calcula el valor medio cuadrático de la corriente.

El envío de la resolución del encoder, así como el número de par de polos de ambos motores, se realiza de una forma análoga al valor digital correspondiente a las salidas analógicas.

Sin embargo, el factor de escala entre el par y el valor medio de la corriente es un valor decimal, por lo que su parte entera será tratada del mismo modo que los valores digitales anteriores, mientras que en la parte decimal se ha propuesto considerar únicamente dos decimales de tal modo que el primer decimal recibido se debe dividir entre 10 y el segundo, entre 100 para finalmente sumar todos los datos resultantes.

Para finalizar, es posible configurar a qué red Wifi se desea conectar, así como a qué *broker* MQTT e introducir las contraseñas asociadas.

Una vez modificados los parámetros requeridos, es necesario almacenarlos en la memoria interna del microcontrolador para evitar la pérdida de información cada vez que se reinicie el dispositivo. La lectura y escritura de ficheros se ha llevado a cabo mediante la librería *SPIFFS*.

En primer lugar, se debe cargar la librería y, posteriormente, abrir el archivo almacenado en la memoria interna. Por defecto, cualquier archivo es abierto en modo de sólo lectura. Además, es posible abrirlo según las opciones que se detallan en la tabla 5.5 [38].

Parámetro	Modo de acceso
r	Sólo lectura
r +	Lectura y escritura.
w	Sólo escritura. El contenido previo que exista en el fichero será eliminado y sobrescrito por el nuevo contenido. Si el fichero no existe, se crea uno nuevo.
w +	Lectura y escritura.
a	Sólo escritura. Permite añadir nuevo contenido sin eliminar la información que exista previamente a la apertura del fichero.
a +	Lectura y escritura. Permite leer el contenido previo del archivo y añadir información adicional.

Tabla 5.5. Modos de apertura de archivos en la librería *SPIFFS*.

Inicialmente, cuando se enciende el dispositivo, es necesario leer todos los parámetros almacenados en el fichero. En particular, se almacenan un total de 9 parámetros: resolución del encoder (expresada en número de bits), el factor de escala entre el par y el valor medio cuadrático de la corriente generada por el PMSM, la opción de aplicar el filtro digital a las diferentes señales, el número de par de polos tanto del ACIM como del PMSM, el nombre de la red Wifi y bróker MQTT, así como las contraseñas asociadas.

Todas estas variables están separadas por un espacio, por lo que será posible leer los parámetros del archivo mediante el método `readStringUntil(" ")`. Además, es necesario emplear los métodos `toInt` y `toFloat` en el caso de que el parámetro leído sea un entero o un valor decimal, respectivamente. La lectura de los parámetros se realiza en un bucle hasta que no haya contenido disponible, tal como se expone en [42].

5.7 TRANSMISIÓN DE DATOS MEDIANTE MQTT

MQTT, de sus siglas en inglés *Message Queue Telemetry Transport*, es un protocolo de transmisión de datos entre diferentes dispositivos. En particular, es un protocolo M2M (*Machine to Machine*) que requiere un menor consumo de recursos que otros protocolos disponibles, por lo que es ampliamente utilizado en el IoT debido a su fácil implementación en microcontroladores [4].

Se basa en un protocolo de publicación/suscripción, de tal modo que cualquier dispositivo puede enviar y recibir mensajes a través de lo que se conoce por *topic*, es decir, un canal de transmisión de datos a través del cual se comparte información.

5.7.1 BROKER MQTT

El componente dentro de la arquitectura MQTT que se encarga de gestionar la transmisión de mensajes a través de este protocolo se conoce por *broker* o servidor MQTT. Cuando se publica un determinado mensaje en un *topic*, el *broker* comprueba qué dispositivos se encuentran suscritos a dicho *topic* para seguidamente enviarles el mensaje correspondiente.

Existen diferentes alternativas que permiten implementar un *broker* MQTT en cualquier proyecto. Por una parte, es posible configurar un servidor MQTT empleando una Raspberry pi, para lo que sería necesario instalar un determinado servidor en este dispositivo, por ejemplo Eclipse Mosquitto, servidor ampliamente empleado en proyectos IoT. Alternativamente, es posible emplear *brokers* gratuitos que permiten implementar un prototipado rápido de la tecnología MQTT. Finalmente, se ha optado por esta última opción, empleando el servidor conocido por HiveMQ.

En este proyecto en particular, el ESP32 actuará como publicador de mensajes, enviando los datos de tiempo, velocidad y corriente consumida por el ACIM, mientras que los clientes suscritos a los *topics* correspondientes son los dispositivos Android tal como se expone en la figura 5.14.

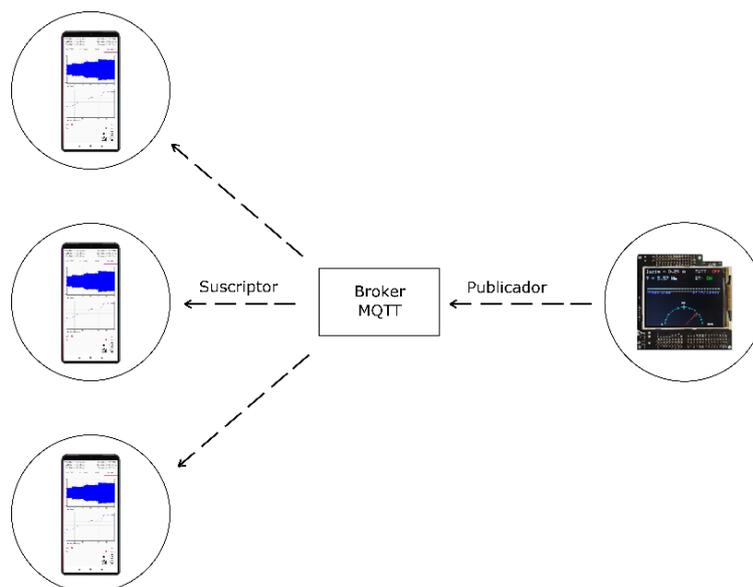


Figura 5.14. Esquema broker MQTT.

5.7.2 CREACIÓN DE TOPICS

Se ha optado por monitorizar vía internet la velocidad angular y la corriente consumida por el motor de inducción. Para ello se han definido un total de tres *topics* correspondientes al tiempo actual, la velocidad y la amplitud de la corriente.

La definición de un *topic* se realiza mediante una sintaxis particular, empleando el operador / para separar niveles. En este caso, se tiene un nivel principal llamado `esp32_Andreu`, el cual agrupa un total de 3 subniveles correspondientes a cada una de las tres variables citadas anteriormente.

Topic	Descripción
<code>esp32_Andreu/</code>	Canal principal que incluye los subniveles de tiempo, velocidad y corriente.
<code>esp32_Andreu/tiempo</code>	Subnivel empleado para monitorizar el instante de tiempo actual.
<code>esp32_Andreu/velocidad</code>	Subnivel empleado para transmitir la velocidad mecánica.
<code>esp32_Andreu/corriente</code>	Subnivel empleado para enviar la amplitud de la corriente consumida por el motor.

Tabla 5.6. *Topics* creados.

El ESP32 se encargará de enviar los datos correspondientes a través de estos *topics*, mientras que el dispositivo Android, o cualquier otro componente suscrito a estos *topics*, recibirá las variables correspondientes.

En cuanto a la suscripción a un conjunto de *topics*, cabe destacar que es posible suscribirse uno a uno o bien se puede emplear una suscripción multinivel en el caso de que se desee suscribirse a una serie de subniveles de forma simultánea. Por ejemplo, si se desea realizar una suscripción a los tres *topics* definidos anteriormente, sería suficiente con suscribirse al *topic* `esp32_Andreu/#`, donde el carácter # indica que se procede a suscribirse a todos los Subniveles pertenecientes al canal principal `esp32_Andreu`.

5.8 VISUALIZACIÓN LOCAL EN LA PANTALLA LCD

En primer lugar, con el objetivo de poder interactuar con la pantalla LCD, hay que tener en cuenta que es necesario incluir previamente las librerías *WROVER_KIT_LCD* y *Adafruit_GFX*. Además, cabe destacar que, tal como se ha comentado anteriormente, los pines de la tabla 5.1 se reservan exclusivamente para la interacción con la pantalla LCD. Mediante el empleo de estas dos librerías es posible representar casi cualquier forma, variable o información que se desee en la pantalla de una forma relativamente sencilla.

5.8.1 SISTEMA DE COORDENADAS Y DEFINICIÓN DE COLORES

Para representar gráficamente puntos, líneas y formas hay que tener en cuenta cómo está definido el sistema de coordenadas de la pantalla. Por una parte, el origen de dicho sistema de referencia se sitúa en la esquina superior izquierda según la figura 5.15. Además, la unidad en la que se define cualquier coordenada es el píxel, de modo que habrá que tener esto en cuenta cada vez que se apliquen funciones en las que intervengan coordenadas [37].

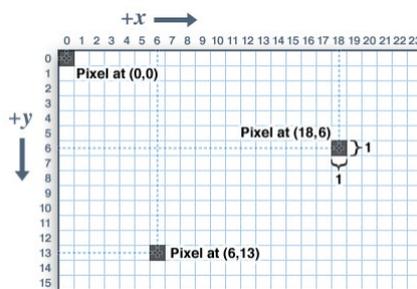


Figura 5.15. Sistema de coordenadas de la pantalla LCD [37].

Mediante el empleo de estas coordenadas es posible representar gráficamente casi cualquier forma que se desee. De entre el conjunto de posibilidades que ofrece la librería para representar formas, en el proyecto se han empleado únicamente las funciones que se detallan en la tabla 5.7.

Función	Descripción
width()	Devuelve el ancho de la pantalla expresado en número de píxeles.
heigh()	Devuelve el alto de la pantalla expresado en número de píxeles.
drawLine(uint16_t x0, uint16_t y0, uint16_t x1, uint16_t y1, uint16_t color)	Permite representar una línea recta dadas las coordenadas de dos puntos.
drawCircle(uint16_t x0, uint16_t y0, uint16_t r, uint16_t color)	Permite dibujar un círculo dados su centro y su radio.
setCursor(uint16_t x0, uint16_t y0)	Sitúa el cursor en una determinada posición sobre la que se desee escribir texto.

Tabla 5.7. Funciones empleadas para representar formas.

En cuanto a la definición de colores, éstos se representan mediante un valor sin signo de 16 bits, independientemente de si la pantalla física posee una mayor o menor representación de colores, es decir, la librería siempre opera con valores de 16 bits. Estos 16 bits representan un color expresado en RGB, de tal modo que los 5 primeros bits se asocian al rojo, los 6 bits intermedios se relacionan con el verde y, finalmente, los 5 bits restantes se corresponden con el azul. El hecho de que el color verde disponga de un bit adicional radica en que el ojo humano es más sensible al color verde [37].

Em consecuencia, es posible representar un total de 65536 colores diferentes. Habitualmente, estos colores se definen en hexadecimal, pero si se dispone de una pantalla monocromática únicamente se trabaja con el valor 0 o 1.

Además, es posible rotar la pantalla mediante la función `setRotation`, cuyo único parámetro de entrada es un valor entero comprendido entre 0 y 3, de modo que las únicas posiciones angulares posibles son los ángulos 0°, 90°, 180° y 270°.

Finalmente, la librería `Adafruit_GFX` presenta otras características diferentes a las empleadas en el proyecto, tales como la creación de formas adicionales o la representación de imágenes cargadas en memoria.

5.8.2 DISEÑO DE LA INTERFAZ LOCAL DE VISUALIZACIÓN

En la pantalla LCD es necesario representar las variables asociadas a la corriente consumida por el motor de inducción, el par transmitido al eje, la velocidad angular mediante un indicador semicircular y el estado de conexión de la transmisión de datos mediante *bluetooth* y MQTT.

Además, cuando el usuario acceda al modo de configuración de un banco de ensayos, se visualizarán en la pantalla LCD los parámetros almacenados en la memoria interna. De esta forma se permitirá visualizar en tiempo real cambios en los parámetros realizados desde la aplicación Android.

En definitiva, se han definido dos modos de visualización de datos y, por tanto, dos interfaces gráficas, cuyo diseño se detallará a continuación. Cabe destacar que ambas interfaces han sido definidas dentro de una misma tarea junto con la transmisión de datos mediante MQTT.

- Visualización de magnitudes eléctricas y estado de conexión

En primer lugar, es necesario que el núcleo en el que se ejecuta esta tarea acceda a información correspondiente al núcleo que se encarga de realizar el proceso de muestreo de datos. En este caso se han empleado variables globales como método de transmisión de datos entre núcleos debido a que no es una tarea tan crítica como la asociada a la transmisión de datos mediante *bluetooth*.

En cuanto a los estados de conexión de la transmisión mediante *bluetooth* y MQTT, en el primer caso se ha empleado una variable booleana auxiliar, `registrando_BT`, de modo que se encuentra a nivel alto cuando el usuario inicia un nuevo registro de datos y, el segundo caso, se ha empleado la instrucción `client.connected()`.

Respecto a la visualización de la velocidad, se ha diseñado un indicador semicircular que recorra todos los valores de velocidad desde 0rpm hasta la velocidad nominal del motor que depende del número de par de polos del ACIM.

Inicialmente, la idea asociada a la visualización de la velocidad a través de un indicador semicircular surgió del diseño de una librería que permitiera representar gráficos vectoriales. Durante la fase de diseño de simulaciones se desarrolló una serie de funciones que permitían representar gráficamente en la aplicación Android los fasores de corriente de estator y rotor tomando como entradas las coordenadas d, q de los mismos. La representación fasorial fue validada mediante el empleo de datos de simulación cargados en el microcontrolador, aunque finalmente la idea original se desechó y se reutilizaron las funciones diseñadas para representar gráficamente la velocidad mediante un indicador semicircular.

Para poder visualizar de forma correcta el indicador de velocidad, se requiere representar de forma gráfica una semicircunferencia con diferentes posiciones de velocidad, las velocidades máximas y mínima y una flecha que permita determinar la velocidad actual del motor.

El diseño se ha realizado de tal forma que la velocidad actual se corresponda con un segmento del arco de una semicircunferencia, cuya longitud total es $\pi \cdot R$, siendo R el radio de la semicircunferencia. Denotando por L el segmento de ese arco se puede comprobar que su valor viene dado por la ecuación 5.15.

$$(5.15) \quad L = \frac{\Omega_r}{\Omega_{nominal}} \cdot \pi \cdot R, \quad \Omega_{nominal} = 60 \cdot \frac{f_{nom}}{p_{ACIM}}$$

Una vez determinado el arco correspondiente a la velocidad actual, se procede a obtener el ángulo recorrido por dicho arco.

$$(5.16) \quad L = \theta \cdot R \rightarrow \theta = \frac{L}{R} = \frac{\Omega_r}{\Omega_{nominal}} \cdot \pi$$

Finalmente, el último paso consiste en representar gráficamente la flecha que indica la velocidad actual. El punto de inicio será la posición inferior central de la pantalla, cuyas coordenadas se denotan por $(\frac{c_x}{2}, c_y)$, siendo c_x el ancho y c_y el alto de la pantalla, respectivamente, mientras que el punto final vendrá dado por las siguientes coordenadas:

$$(5.17) \quad \begin{aligned} x &= R \cdot \cos(\theta) + \frac{c_x}{2} \\ y &= R \cdot \sin(\theta) + c_y \end{aligned}$$

Con el fin de representar gráficamente el extremo de la flecha se requiere emplear un total de tres puntos adicionales: la base y las dos esquinas correspondientes. La base del extremo se ha definido a unas coordenadas dadas por:

$$(5.18) \quad \begin{aligned} x_b &= 0.8 \cdot R \cdot \cos(\theta) + \frac{c_x}{2} \\ y_b &= 0.8 \cdot R \cdot \sin(\theta) + c_y \end{aligned}$$

En cuanto a los dos puntos restantes, es posible realizar un cambio de coordenadas aplicando una rotación sobre la recta dada por los puntos $(\frac{c_x}{2}, c_y)$ y (x_b, y_b) . Para ello se emplea la matriz de rotación, cuya definición viene dada por la ecuación 5.19.

$$(5.19) \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\text{sen}(\alpha) \\ \text{sen}(\alpha) & \cos(\alpha) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Mediante los ángulos $\alpha = \pm 0.1$ rad se han definido los dos puntos restantes asociados al extremo de la flecha, por lo que el diseño de la interfaz estaría resuelto.

Los resultados asociados a la implementación del diseño de la interfaz en el ESP32 se exponen en la figura 5.16.

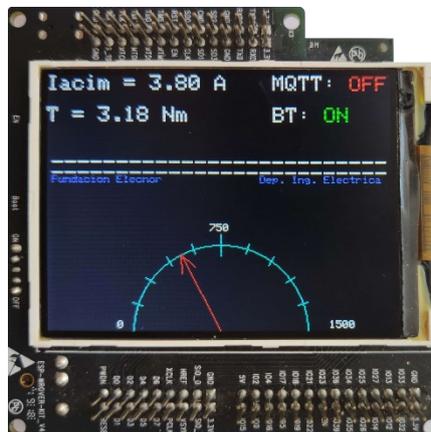


Figura 5.16. Interfaz para la visualización de magnitudes.

- Visualización de parámetros

Esta interfaz se habilita cuando el usuario accede a la configuración de parámetros desde la aplicación Android. Esto permite representar en la pantalla LCD la configuración de parámetros almacenados en la memoria interna.

En particular, se representa la resolución del encoder expresada en número de bits, el número de par de polos del ACIM y PMSM, el factor de proporcionalidad entre el par y la corriente generada por el PMSM, el estado de aplicación del proceso de filtrado digital y la red Wifi y servidor MQTT configurados. La interfaz diseñada se expone en la figura 5.17.

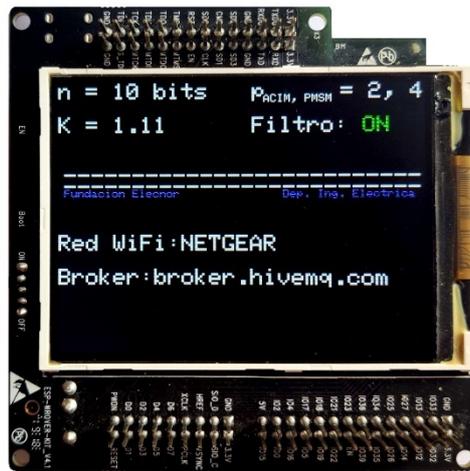


Figura 5.17. Interfaz para la visualización de parámetros de configuración.

CAPÍTULO 6: DISEÑO Y DESARROLLO DE LA APLICACIÓN ANDROID

En el presente capítulo que describirán todas las características que incorpora la aplicación Android y se detallará el procedimiento que se ha seguido para diseñarla. En primer lugar, en el apartado 6.1 se realizará una introducción al entorno de desarrollo Android Studio, exponiendo las posibilidades de programación que ofrece este software. Seguidamente, en el epígrafe 6.2 se describirá la interfaz de usuario diseñada y, finalmente, se analizarán todas las funcionalidades que implementa la aplicación Android diseñada.

La aplicación Android que se desea diseñar debe satisfacer una serie de hitos que se resumen a continuación:

- Visualización mediante *bluetooth* de un registro histórico de datos asociado a las siguientes magnitudes: posición y velocidad angular, corriente generada por una de las fases del PMSM, par electromagnético, módulo del fasor de corriente de estator y la amplitud de la señal temporal asociada.
- Visualización a través del protocolo MQTT de la velocidad angular y la amplitud de la corriente de estator del ACIM.
- Control de los variadores de frecuencia del ACIM y el PMSM.
- Almacenamiento en la nube de los datos registrados mediante *bluetooth*, permitiendo su posterior análisis mediante Matlab o similares.
- Configuración de cualquier puesto de prácticas mediante la modificación de sus parámetros característicos, tales como la resolución del encoder, el número de par de polos del ACIM y el PMSM y el factor de proporcionalidad entre el par electromagnético y el valor medio cuadrático asociado a la corriente generada por una de las fases del PMSM. También se debe de poder configurar a qué red Wifi y servidor MQTT se desea conectar, así como introducir las contraseñas correspondientes, además de poder activar o desactivar la aplicación del proceso de filtrado digital, permitiendo almacenar un conjunto de datos no filtrados para su posterior análisis.

De forma análoga al capítulo 5, el código desarrollado en Arduino IDE no se ha incluido en el documento para no extender su longitud. Como alternativa el código se encuentra disponible en un repositorio de GitHub [42].

6.1 INTRODUCCIÓN A ANDROID STUDIO

El software Android Studio es un entorno de desarrollo multiplataforma que permite diseñar aplicaciones para dispositivos móviles basadas en el sistema operativo Android mediante el lenguaje de programación Java o Kotlin. En el presente proyecto se ha empleado java como lenguaje de programación.

Mediante el empleo de Android Studio no únicamente es posible diseñar todo el código que permite ejecutar el correcto funcionamiento de cualquier aplicación Android, sino que además permite desarrollar la interfaz de usuario asociada desde el propio software. La interfaz característica de cualquier proyecto creado con Android Studio se expone en las figuras 6.1 y 6.2. En la primera de estas figuras es posible observar el diseño de una interfaz gráfica mientras que en la segunda se expone la generación de nuevo código basado en el lenguaje de programación java.

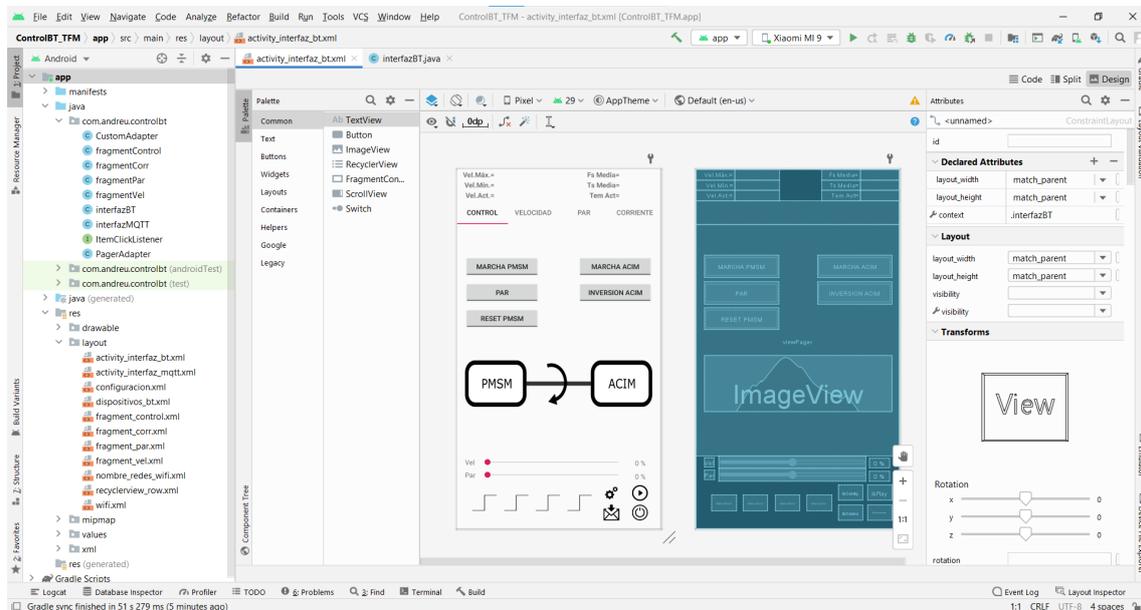


Figura 6.1. Entorno de desarrollo Android Studio – Diseño de la interfaz.

Tal como se observa en la figura 6.1, en el lateral izquierdo se encuentran dos ventanas verticales: en la primera se ubican todos los archivos que componen un proyecto, mientras que la segunda permite añadir nuevos elementos a la interfaz gráfica. En la parte central se halla la interfaz gráfica que se está desarrollando y, finalmente, en el lateral derecho es posible modificar diferentes propiedades características de los elementos creados en la interfaz.

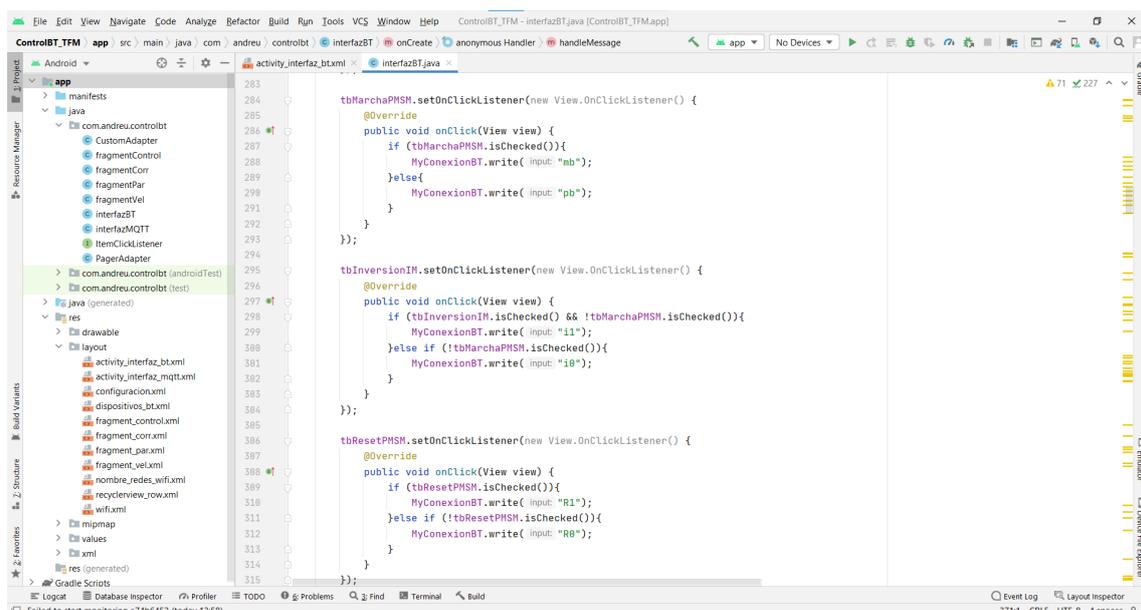


Figura 6.2. Entorno de desarrollo Android Studio – Generación de código.

En la figura 6.2 se expone el modo de generación de nuevo código. A diferencia del diseño de interfaces gráficas, la creación de código no permite una programación gráfica, sino que se trata de una programación basada en texto.

Tal como se ha podido evidenciar anteriormente, para diseñar una nueva aplicación en Android Studio es necesario emplear principalmente dos tipos de archivos, definidos a continuación:

- Archivo XML. Este tipo de fichero incluye el diseño de la interfaz gráfica que visualizará el usuario final. Para programar una interfaz es posible emplear programación gráfica o bien utilizar una programación basada en texto.
- Archivo java. Incluye la programación del funcionamiento interno de la aplicación. Este fichero está estrechamente relacionado con la interfaz gráfica, ya que es necesario hacer referencia a los objetos creados en el archivo XML con el objetivo de ejecutar ciertas instrucciones cuando el usuario interactúe con la interfaz.

6.1.1 CREACIÓN DE INTERFACES GRÁFICAS

El diseño de una nueva interfaz de usuario se realiza, habitualmente, mediante programación gráfica, introduciendo los elementos necesarios y creando lo que se conoce por restricciones entre estos componentes. Una restricción es una imposición espacial entre dos elementos de una interfaz gráfica, permitiendo ubicar todos estos elementos de una forma relativamente sencilla.

Cabe destacar que también es posible crear una interfaz mediante programación basada en texto, facilitando la creación de nuevos objetos y las restricciones ente estos de una forma más accesible cuando el número de elementos de la interfaz es considerable.

Una comparativa entre la forma de crear una interfaz de forma gráfica o mediante programación basada en texto se expone en la figura 6.3.

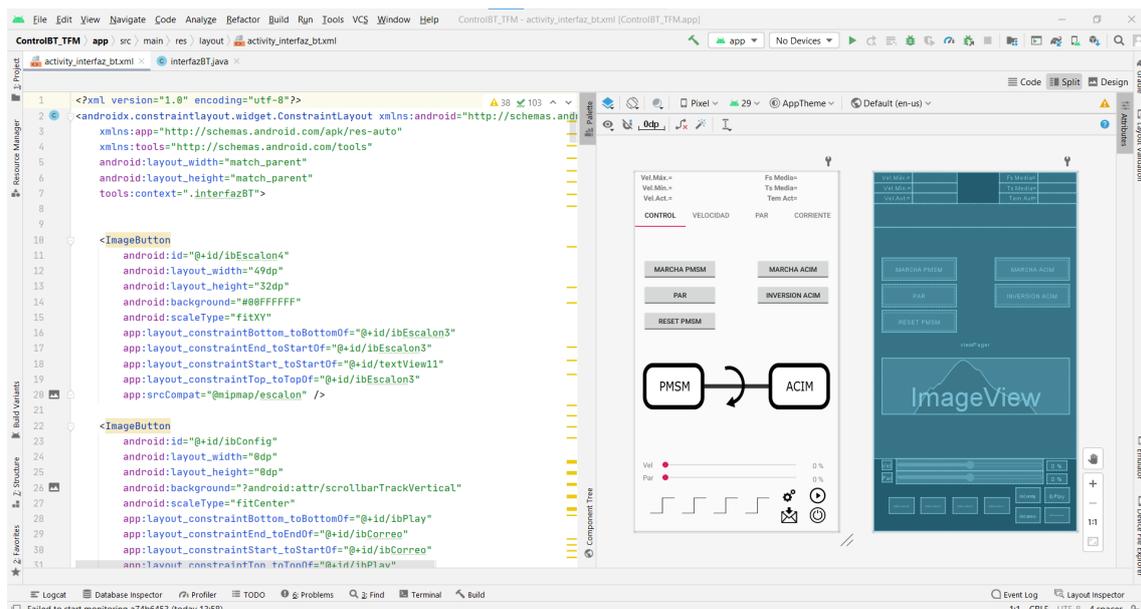


Figura 6.3. Comparativa programación gráfica y basada en texto.

En cualquier caso, independientemente de la forma de programar elegida, para diseñar una interfaz es necesario introducir los componentes que se deseen. Estos componentes pueden ser botones, barras deslizables, cuadros de verificación, etc. Seguidamente, en la tabla 6.1 se describen todos los componentes que se han empleado en el presente proyecto.

Tipo de elemento	Descripción
TextView	Se trata de un texto estático que no es editable desde la interfaz de usuario, pero sí que es modificable desde el código de la aplicación. Se ha empleado para indicar ciertas aclaraciones como qué magnitud o qué control se está visualizando en la interfaz.
ImageView	Se trata de una imagen estática no modificable desde la interfaz, pero que es posible modificarla desde el código.
EditText	Similar al TextView pero modificable desde la propia interfaz de usuario. Una característica importante de este elemento es la posibilidad de ejecutar alguna instrucción cada vez que se modifica el texto correspondiente. Se ha empleado en la configuración de los parámetros asociados a un banco de ensayos, es decir, en la modificación de la resolución del encoder, el número de par de polos de ambos motores,
ImageButton	Es un tipo de control empleado para ejecutar un conjunto de instrucciones cada vez que sea pulsado. Se ha empleado para iniciar un nuevo registro de datos, almacenarlos en un servicio en la nube o entrar en el modo de configuración de parámetros, entre otras funcionalidades.
ToggleButton	Similar al anterior, pero se trata de un botón de dos niveles, de tal modo que ejecuta un conjunto de instrucciones u otro en función de cuál sea el estado en el que se encuentre. Se ha empleado en ciertos controles de los variadores de frecuencia, tales como la marcha/paro de estos, la selección del tipo de control, el reset del variador del PMSM cuando se produzca un determinado fallo o la inversión de giro del ACIM.
CheckBox	Similar a un ToggleButton, cambiando únicamente la visualización del objeto.
RadioGroup	Es un conjunto de controles, de tipo RadioButton, que permiten ejecutar ciertas instrucciones cuando son pulsados.
SeekBar	Se trata de una barra deslizable que ejecuta ciertas instrucciones cada vez que se modifica su valor. Se ha empleado en las consignas de frecuencia y par de los variadores del ACIM y PMSM, respectivamente.
LineChart	Componente empleado, por ejemplo, en la representación gráfica de series temporales. Se ha utilizado en la visualización de las diferentes magnitudes del banco de ensayos monitorizadas.

Tabla 6.1. Elementos empleados en el proyecto de Android Studio.

6.1.2 GENERACIÓN DE CÓDIGO

Tal como se ha comentado anteriormente, la aplicación Android será programada mediante el lenguaje java. Cuando se desarrolla el código fuente de una aplicación hay que tener en cuenta que todo el código estará vinculado con la interfaz gráfica, es decir, se requiere que cada vez que el usuario interactúe con la interfaz, se ejecuten ciertas instrucciones.

Cuando se crea una nueva aplicación, en primer lugar es necesario definir todas las variables que vayan a interactuar con la interfaz gráfica para posteriormente asignar controles a través de estas variables.

6.2 INTERFAZ DE USUARIO

En el diseño de la interfaz gráfica, hay que tener en cuenta qué requerimientos debe cumplir la aplicación Android. Las funcionalidades que debe de implementar la aplicación son, principalmente, la transmisión de datos a través del protocolo MQTT, el envío de un registro de datos mediante *bluetooth* y la configuración de los parámetros característicos de un puesto de prácticas. En consecuencia, es posible agrupar las diferentes interfaces diseñadas en tres grupos, descritos en los epígrafes siguientes.

6.2.1 INTERFAZ REGISTRO DE DATOS MQTT

La aplicación Android debe de representar gráficamente la velocidad y la amplitud de la corriente consumida mediante la recepción de estas variables a través de MQTT. Se ha optado por definir como interfaz principal la correspondiente a la visualización de datos mediante este protocolo de comunicaciones. En consecuencia, desde esta interfaz debe ser posible acceder a la interfaz correspondiente a la transmisión de datos mediante *bluetooth*.

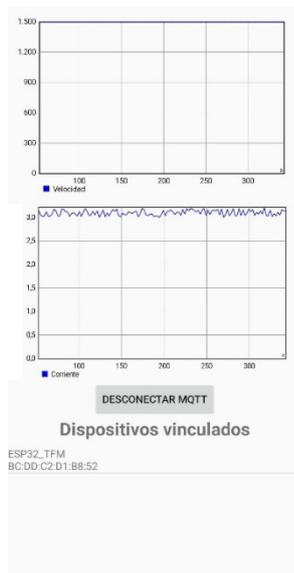


Figura 6.4. Interfaz de usuario - MQTT.

La interfaz diseñada se expone en la figura 6.4, donde es posible visualizar la representación de las variables citadas anteriormente, es decir, la velocidad mecánica del rotor y el módulo del fasor de corriente de estator.

Además, es posible desconectar el microcontrolador de la red Wifi y, finalmente, se permite iniciar la conexión a un dispositivo *bluetooth* previamente vinculado, dando paso a la interfaz que gestiona la transmisión de datos a través de *bluetooth*, cuyo diseño y explicación de las diferentes funcionalidades que ofrece esta interfaz se detalla en el epígrafe 6.2.2.

6.2.2 INTERFAZ REGISTRO DE DATOS BLUETOOTH

Seguidamente, se procede a describir las funcionalidades implementadas en la interfaz asociada a la representación de datos recibidos a través de *bluetooth*.

Esta interfaz debe de sustituir a las botoneras físicas que permiten regular los variadores de frecuencia del motor de inducción y de imanes permanentes. En particular, los controles que debe disponer la interfaz y que permiten establecer una configuración inicial de los variadores son los siguientes:

- Marcha/Paro ACIM.
- Inversión de giro ACIM.
- Marcha/Paro PMSM.
- Selección del tipo de control PMSM.
- Reset PMSM.

Tal como es posible evidenciar, los controles anteriores únicamente se emplean cuando se empieza a trabajar en el banco de ensayos. Adicionalmente, es necesario disponer de ciertos controles que permitan regular la velocidad y par durante la ejecución de los diferentes ensayos. Estos controles se detallan a continuación:

- Consignas de frecuencia/velocidad mediante cambios en escalón.
- Modificación de la frecuencia/velocidad de forma analógica.
- Modificación del par de forma analógica.

Otra de las principales características implementadas en esta interfaz consiste en la visualización de los registros históricos asociados a diferentes variables, permitiendo monitorizar el conjunto de motores acoplados. Cabe recordar que el paso de muestreo de las diferentes señales no es el mismo en todos los casos, por lo que el diseño de la aplicación Android debe ser lo suficientemente robusto como para poder gestionar la recepción de estos datos. Las variables monitorizadas son las siguientes:

- Posición angular. La obtención de estos datos permite analizar múltiples algoritmos que permitan determinar el valor de la velocidad angular.
- Velocidad angular.
- Corriente generada por una de las fases del PMSM. A partir de esta variable se calcula el par electromagnético. El hecho de disponer de un registro histórico de esta señal permite analizar diferentes alternativas en cuanto al cálculo del par, como por ejemplo la aplicación de un filtro digital de tipo Butterworth o bien la aplicación del filtro Hampel. Además, la representación de esta variable se ha realizado a modo de comprobación del correcto funcionamiento del producto diseñado.
- Par electromagnético.
- Módulo de la corriente consumida por el ACIM. Debido a que el módulo de la corriente se obtiene a partir del valor de la corriente asociada a cada una de las fases del motor, también se ha implementado la posibilidad de visualizar la corriente de cada una de las fases del ACIM. Esto se detalla en la interfaz de configuración de parámetros.
- Amplitud de la corriente consumida por el ACIM. De este modo se permite analizar qué alternativa es mejor para poder representar el nivel de corriente consumida por este motor.

Finalmente, esta interfaz debe permitir iniciar un nuevo registro de datos, almacenarlos mediante un servicio de almacenamiento en la nube y acceder a la interfaz de configuración de parámetros.

Debido a la gran cantidad de controles y funcionalidades relacionadas con la representación gráfica de datos, no es posible incluir todos los elementos en una única ventana. En consecuencia, se ha optado por agrupar todas las funcionalidades de esta interfaz en diferentes ventanas, de tal modo que la primera de estas estará orientada a la configuración inicial de los variadores, mientras que en el resto de ventanas se ha implementado la monitorización del conjunto de señales. El diseño propuesto se expone en la figura 6.5.

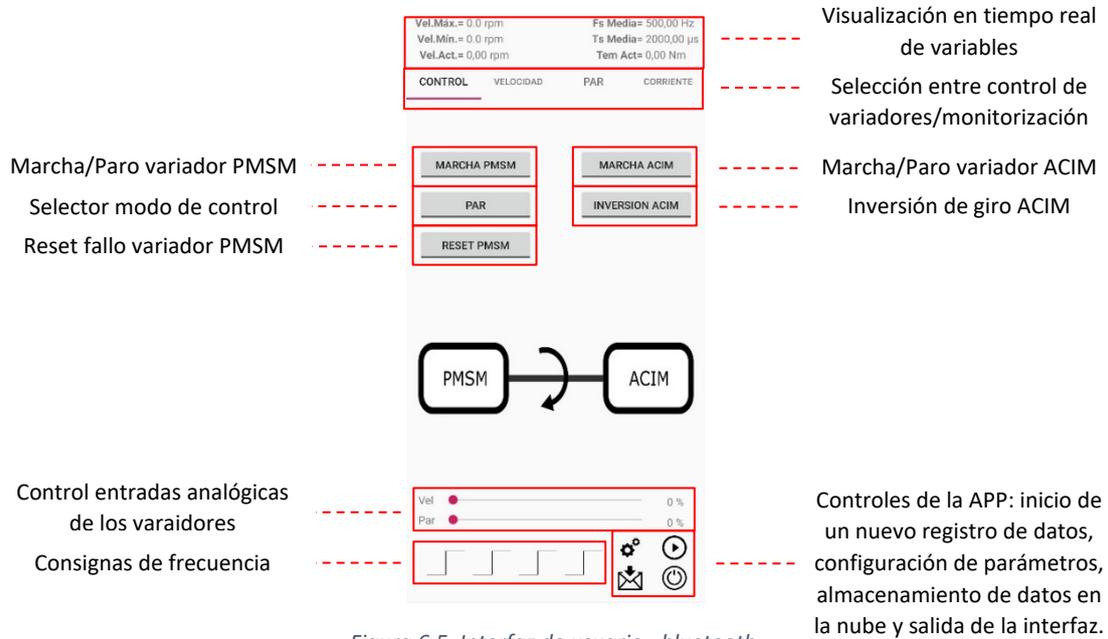


Figura 6.5. Interfaz de usuario - bluetooth.

En la figura anterior se han representado los controles principales de los variadores. En la figura 6.6 es posible visualizar el resto de pantallas, las cuales están orientadas a la representación de las diferentes señales.

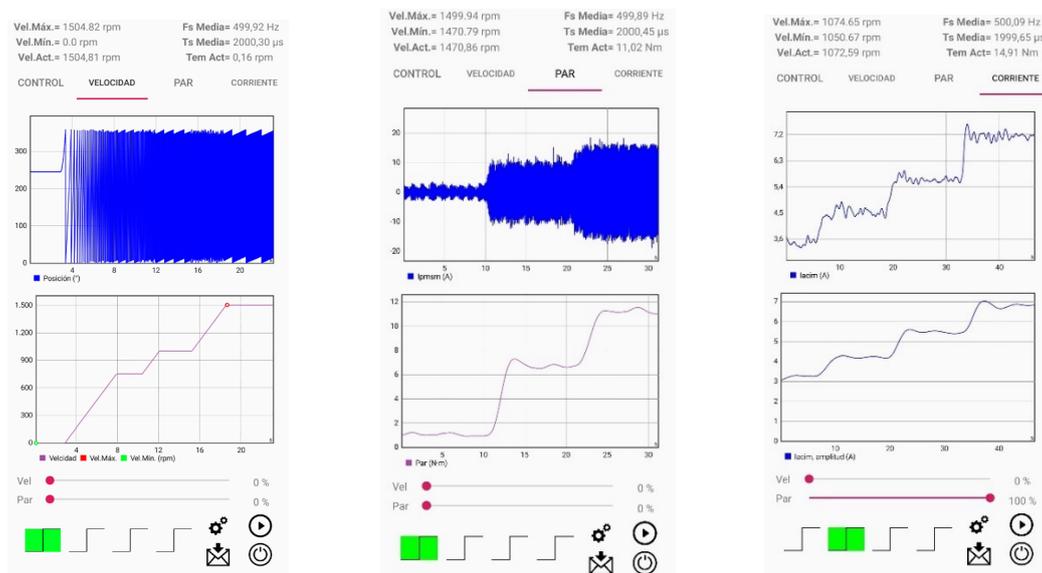


Figura 6.6. Monitorización del banco de ensayos.

6.2.3 INTERFAZ CONFIGURACIÓN DE PARÁMETROS

La última interfaz que se requiere diseñar es la asociada a la configuración de parámetros característicos del banco de ensayos, además de poder configurar otros aspectos como la red Wifi y el servidor MQTT. En particular, el conjunto de parámetros configurables se detalla a continuación:

- Resolución del encoder. Se posibilita la opción de modificar el número de bits del encoder, calculando seguidamente el número de pulsos por vuelta y la precisión en grados correspondiente.
- Activar/Desactivar la aplicación del filtro digital. Esto permite obtener un registro de datos no filtrados con el fin de realizar un posterior análisis.
- Selección entre el módulo de la corriente del ACIM y las fases asociadas. Permite alternar entre la visualización del módulo del fasor de corriente y la corriente que circula por cada una de las fases. Principalmente permite corroborar el correcto funcionamiento de los sensores de efecto Hall instalados. Por defecto se representa el módulo de la corriente.
- Factor de proporcionalidad entre el par electromagnético y el valor medio cuadrático de la corriente generada por una de las fases del PMSM.
- Número de par de polos del ACIM y PMSM. El número de par de polos del ACIM afecta a los valores del indicador de velocidad diseñado en la pantalla LCD del ESP32, mientras que el asociado al PMSM influye en el cálculo del periodo de la corriente asociada al PMSM, empleado en el cálculo del valor medio cuadrático correspondiente.
- Tiempo de registro. Se trata del intervalo de tiempo dentro del cual se monitorizan las diferentes señales.
- Configuración de la red Wifi y servidor MQTT. Permite seleccionar a qué red Wifi se desea conectar de entre las disponibles e introducir el servidor MQTT requerido, así como configurar las contraseñas correspondientes. Esta funcionalidad se ha implementado en una pantalla diferente, tal como se expone en la figura 6.8.

El diseño de la interfaz que permite configurar el conjunto de parámetros característicos de cada banco de ensayos se expone en la figura 6.7.

Cabe destacar que la posibilidad de modificar los parámetros que intervienen en el software incrementa notablemente la versatilidad del producto, permitiendo su aplicación a cualquier banco de ensayos sin la necesidad de modificar el software diseñado.

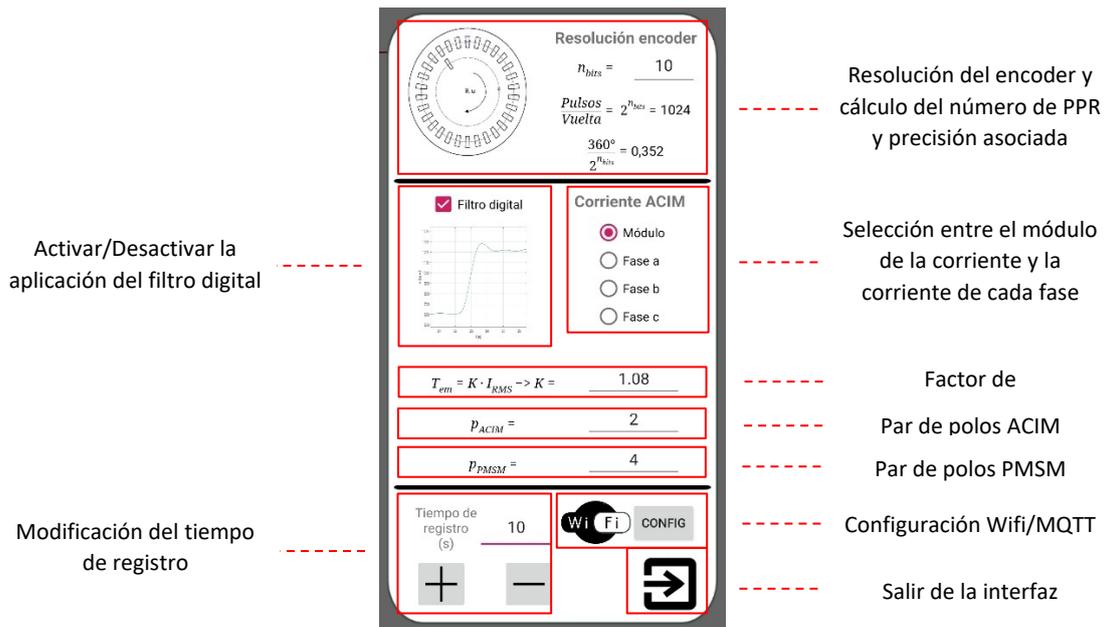


Figura 6.7. Configuración de parámetros del banco de ensayos.

Finalmente, en la figura 6.8 se expone la configuración de la red Wifi a la se que desea conectar y el servidor MQTT asociado. En la parte superior se indica, en el caso de que se haya realizado la conexión a la red Wifi de forma correcta, a qué rúter y servidor MQTT se ha conectado y la dirección IP local correspondiente. En la parte central se realiza un escaneo de qué redes Wifi se encuentran al alcance del microcontrolador y se posibilita la modificación del servidor MQTT y las contraseñas correspondientes. En la parte inferior se realiza la conexión o desconexión de la red Wifi, así como se efectúa la salida de la pantalla de configuración.

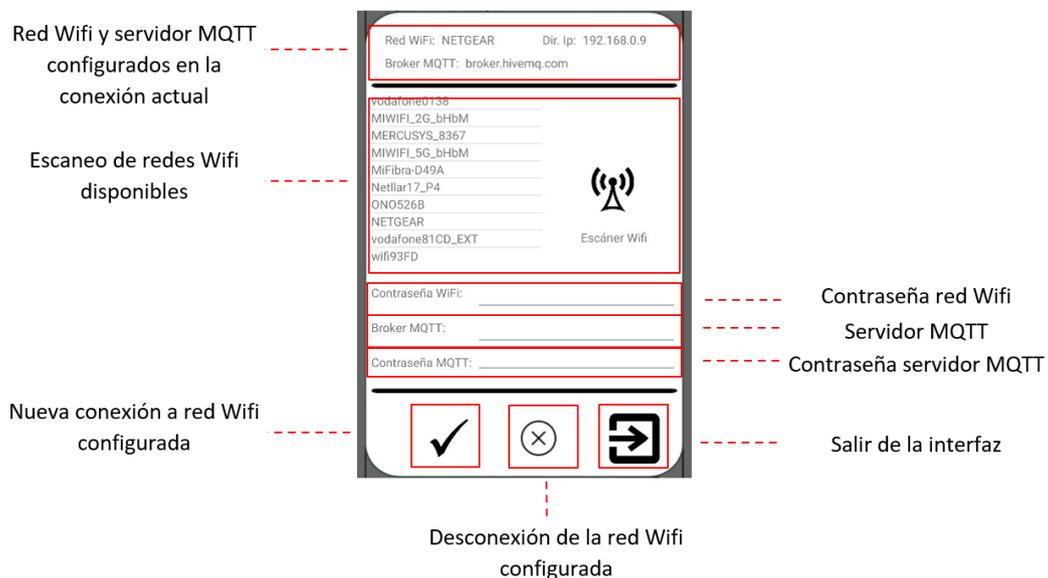


Figura 6.8. Configuración red Wifi y servidor MQTT.

CAPÍTULO 7: RESULTADOS

En el presente capítulo se presentarán los principales resultados obtenidos tras el desarrollo del proyecto. En primer lugar, en el apartado 7.1 se realizarán una serie de ensayos sobre dos bancos de ensayos compuestos por motores con diferentes eficiencias, permitiendo analizar el consumo de corriente y la evolución de magnitudes tales como velocidad o el par asociadas a cada uno de estos motores frente a las mismas condiciones de operación.

En el apartado 7.2 se efectúa un análisis de la eficiencia de dos motores de inducción diferentes empleando los resultados obtenidos en los ensayos experimentales. Los ensayos se han realizado mediante dos metodologías diferentes: conectando el motor de inducción al variador de frecuencia y mediante conexión directa a red. Esto permite analizar y comparar el ruido que el variador introduce en el sistema.

En el apartado 7.3 se analiza el grado de mejora en el acondicionamiento de las señales ofrecido por el proceso de filtrado digital. Para ello se obtiene diferentes registros de datos alternando entre la activación y desactivación del filtro digital a través de la aplicación Android.

En el epígrafe 7.4 se exponen los resultados que permite obtener la aplicación Android desarrollada, presentando todas las funcionalidades que ofrece, desde el control de los variadores, la visualización de los datos asociados a los diferentes ensayos, así como su almacenamiento mediante un servicio de almacenamiento en la nube y la configuración de los parámetros característicos de un banco de ensayos, permitiendo corroborar la modificación de las variables correspondientes en tiempo real a través de la pantalla LCD del microcontrolador.

Finalmente, en el apartado 7.5 se efectúa una comparativa entre los datos obtenidos mediante los ensayos experimentales y los correspondientes a las simulaciones. Esta es una etapa necesaria en la validación de un modelo digital cuya principal aplicación se encuentra esencialmente en el diseño de gemelos digitales. En definitiva, con el objetivo de validar la correcta respuesta del modelo digital es necesario comparar los datos ofrecidos por la simulación con los correspondientes al sistema real.

7.1 ENSAYOS EXPERIMENTALES

El objetivo de estos ensayos experimentales consiste en analizar la eficiencia de diferentes motores eléctricos a través de la corriente consumida por éstos. Además, se analizará la evolución de otras variables tales como la velocidad o el par electromagnético.

Los diferentes ensayos se han realizado sobre dos bancos de ensayos idénticos en todos los elementos que lo componen a excepción del motor de inducción. Un puesto de prácticas dispone de un motor cuya clase de eficiencia es IE2, presentando una eficiencia nominal del 84.3%, mientras que el siguiente puesto de prácticas ensayado presenta un motor con una clase de eficiencia IE1, ofreciendo una eficiencia nominal del 79.7%. En la tabla 7.1 es posible observar una comparativa de la placa de características de ambos motores, permitiendo analizar sus valores nominales.

Magnitud \ Clase de eficiencia	IE2	IE1
Tensión (V)	400 / 230 (Y / Δ)	400 / 230 (Y / Δ)
Corriente (A)	4.65 / 8.1 (Y / Δ)	4.85 / 8.43 (Y / Δ)
Potencia (kW)	2.2	2.2
Eficiencia (%)	84.3	79.7
cos φ	0.82	0.81

Tabla 7.1. Placa de características de los motores ensayados.

Los ensayos se realizarán en bucle abierto, sometiendo ambos bancos de ensayos a las mismas condiciones de operación, es decir, frente a las mismas consignas de frecuencia y par, posibilitando una correcta comparativa entre el nivel de corriente consumida por los motores de inducción.

En particular, los ensayos realizados mediante conexión a través del variador de frecuencia se han realizado a 33.33Hz y, en cualquier caso, se ha ido modificando paulatinamente el par resistente a intervalos de 15 segundos, aplicando de forma secuencial los valores de 0%, 40%, 70% y 100% de la carga máxima.

Tal como se ha expuesto anteriormente, los ensayos se realizarán empleando dos metodologías: alimentando el motor de inducción a través del variador de frecuencia y mediante conexión directa a red. El cambio de un ensayo a otro se realiza de una forma muy sencilla, simplemente conectando los devanados de entrada del producto diseñado al variador de frecuencia o directamente a red, tal como se expone en la figura 7.1. Cabe destacar que estos dos tipos de conexiones son los mismos que los que se pueden simular en el modelo digital del capítulo 3.



Figura 7.1a. Conexión mediante variador.

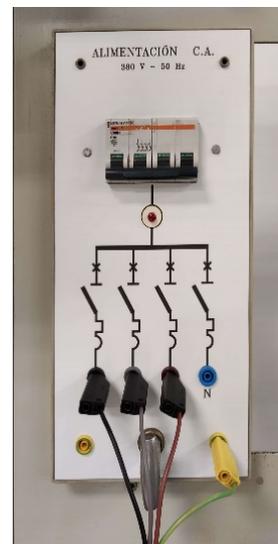


Figura 7.1b. Conexión directa a red.

Figura 7.1. Tipos de conexión del motor de inducción.

Previamente a la obtención de cualquier registro de datos, es necesario realizar todo el conexionado necesario, tanto de los variadores de ambos motores, los devanados del motor de inducción, los pines de conexión del sensor de efecto Hall del PMSM y el encoder óptico tal como se expone en la figura 7.2. Se observa que la conexión realizada en dicha figura es a través del variador de frecuencia.

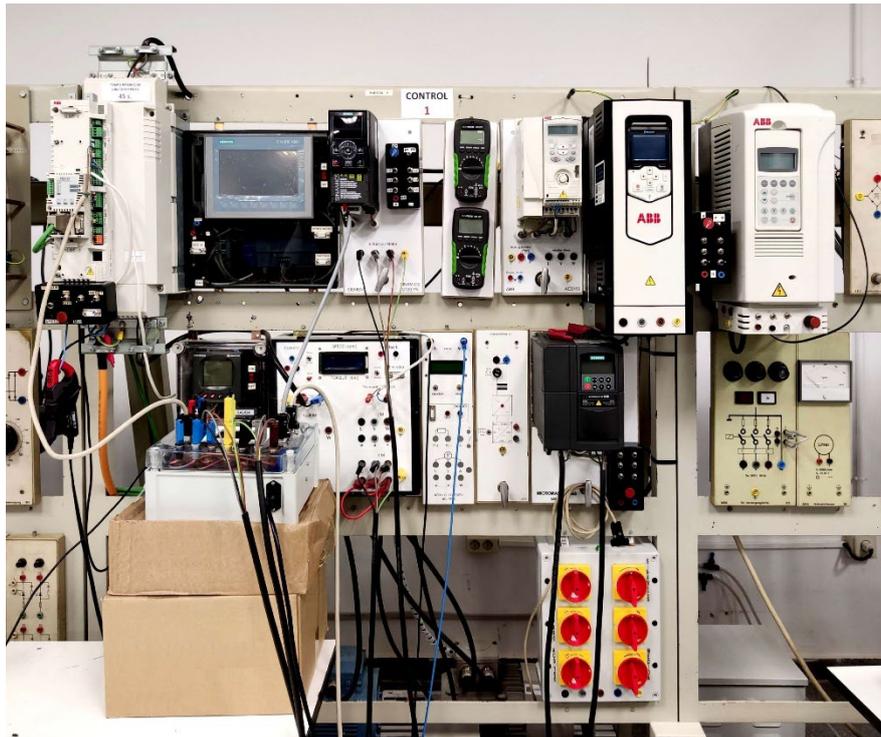


Figura 7.2. Conexión de la placa.

Seguidamente, se describen los resultados obtenidos en ambos ensayos, correspondiente al motor cuya clase de eficiencia es IE2. Posteriormente, en el epígrafe 7.3 se analizan los resultados obtenidos en ambos motores.

Tal como se ha expuesto anteriormente, los ensayos se realizan a una frecuencia de alimentación fija, en este caso de 33.33Hz, es decir, 1000 rpm trabajando en vacío. La carga aplicada en función del tiempo, en tanto por cien de su valor nominal, se expone en la tabla 7.2.

Intervalo de tiempo (s)	Valor de la carga (%)
[0, 15]	0
[15, 30]	40
[30, 45]	70
[45, 60]	100

Tabla 7.2. Valores de carga aplicada en función del tiempo.

Todas las señales serán registradas previamente en la aplicación Android para posteriormente enviarlas a través de un servicio de almacenamiento en la nube con el objetivo de realizar un posterior análisis mediante Matlab.

En los apartados 7.1.1 y 7.1.2 se detallan los resultados obtenidos tras la realización de los ensayos citados anteriormente.

7.1.1 CONEXIÓN DIRECTA A RED

En este caso se alimenta el motor mediante conexión directa a red realizando la conexión que se detalla en la figura 7.1b.

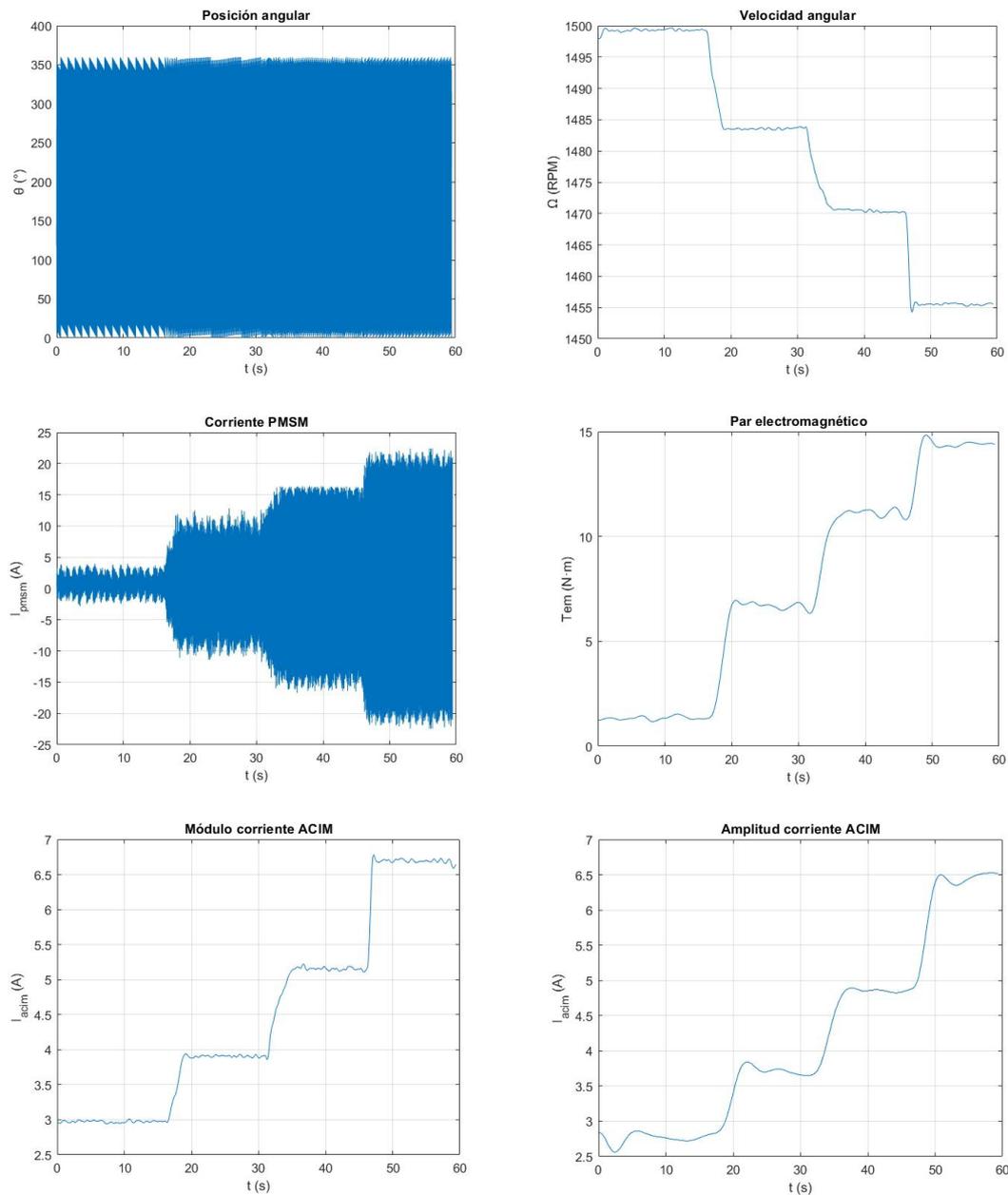


Figura 7.3. Ensayo 2 - Conexión directa a red.

En primer lugar, se observa que a medida que se incrementa la carga se produce una reducción de la velocidad del motor debido a que opera en bucle abierto.

Por otro lado, como era de esperar se produce un incremento en la amplitud de la corriente generada por el PMSM, derivando en un incremento en el par electromagnético. Cabe destacar que al 0% de la carga nominal no se obtiene un par nulo debido a que un 0 digital en el microcontrolador no se correspondía con un 0 analógico a la salida del ESP32, es decir, existía cierto *offset*.

Finalmente, el nivel de corriente aumenta de forma progresiva hasta que el par alcanza su valor nominal. A diferencia del ensayo alimentando el motor a través del variador, la lectura del nivel de corriente sería más adecuado realizarla a través del módulo del fasor de corriente.

7.1.2 CONEXIÓN MEDIANTE VARIADOR DE FRECUENCIA

En este caso se alimenta el motor a través del variador de frecuencia según la conexión detallada en la figura 7.1a.

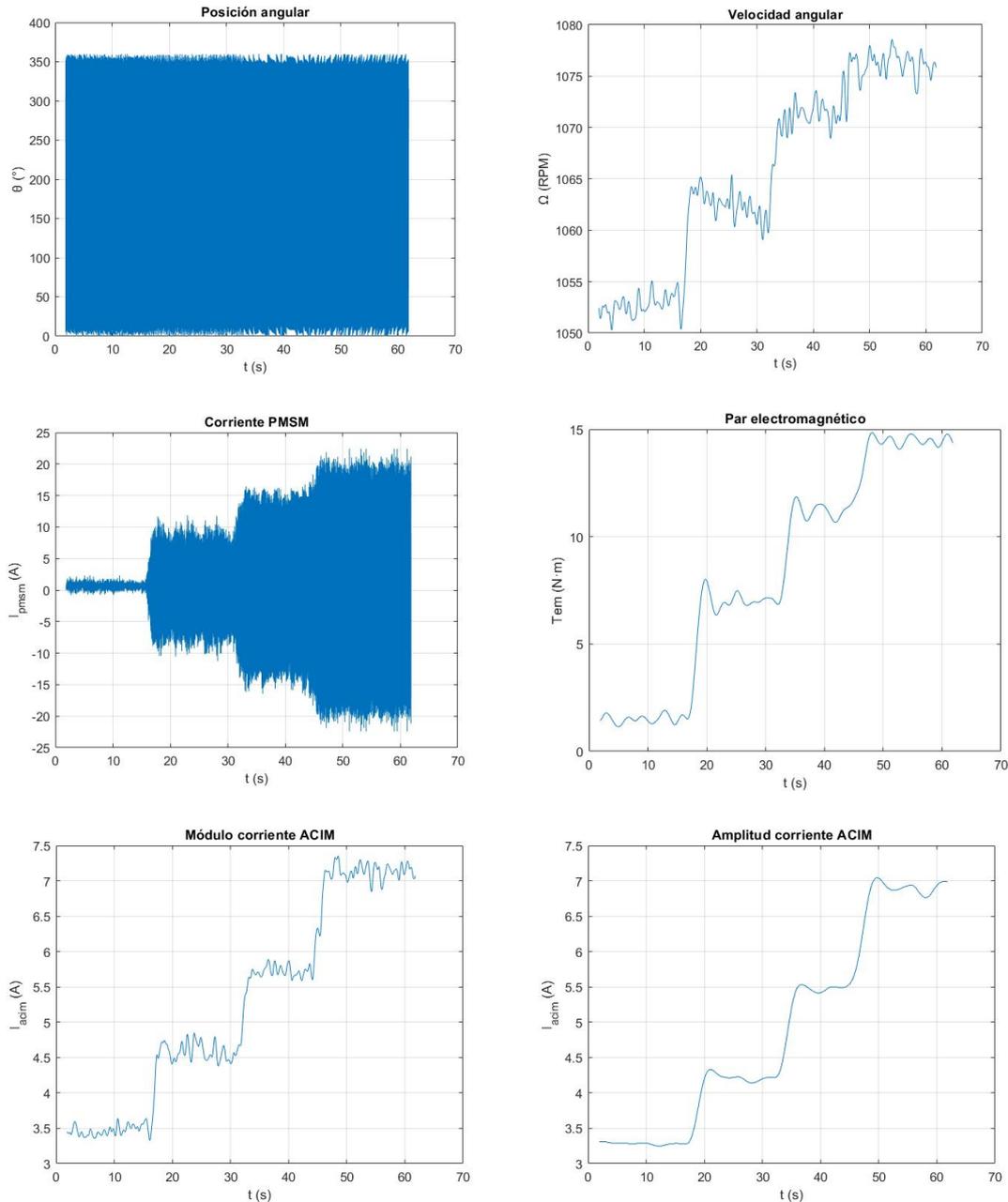


Figura 7.4. Conexión de la placa.

Cabe destacar, en primer lugar, un empeoramiento de la relación señal/ruido debido a la conexión del variador. Además, la velocidad se incrementa ya que, al aumentar la carga, el variador del ACIM introducía un incremento en el valor eficaz de la tensión de alimentación.

Por otra parte, los análisis asociados al par y al consumo de corriente son análogos al caso anterior. Cuando se produce un incremento en la carga resistente, aumenta el par transmitido al eje y el nivel de corriente consumida.

7.2 ANÁLISIS DE EFICIENCIA

Se han realizado los ensayos expuestos en el epígrafe 7.1 sobre los motores de inducción cuya clase de eficiencia es IE2 e IE1. Al someter ambos motores a las mismas condiciones de operación, es decir, mismos niveles de tensión y par, es posible realizar una comparativa del nivel de corriente consumida por ambos. Los resultados de los diferentes ensayos realizados se exponen en la figura 7.5.

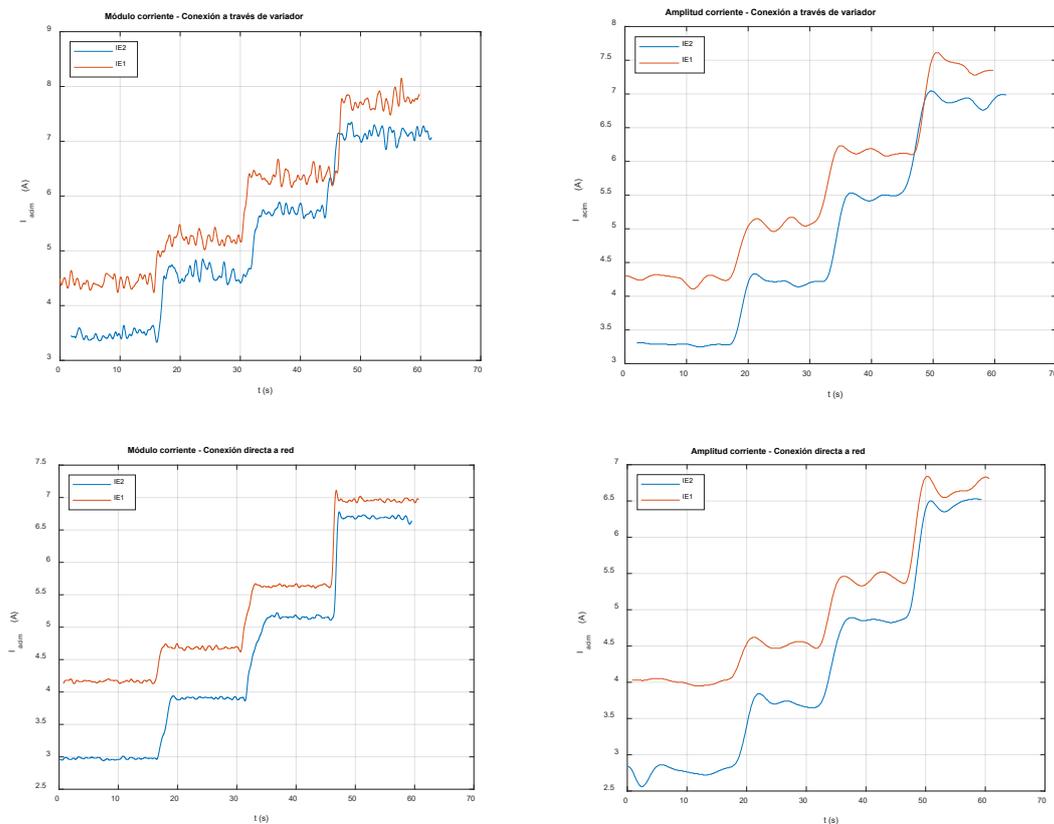


Figura 7.5. Comparativa del nivel de corriente.

Los resultados son los esperados, el motor con clase de eficiencia IE1 presenta un consumo de corriente mayor que el correspondiente al motor IE2 frente a las mismas condiciones de operación.

El hecho de disponer de una lectura del nivel de corriente consumida a través de dos variables, en particular el módulo y amplitud, permite analizar qué señal es más adecuada para poder medir dicho nivel de corriente. En el caso del ensayo mediante conexión a través del variador de frecuencia sería más conveniente emplear la amplitud como variable de medida, mientras que en el caso de conexión directa a red el módulo de la corriente presenta un valor más estable que la amplitud.

Adicionalmente, puede resultar de interés analizar y comparar otro tipo de señales tales como la velocidad o el par mecánico. En la figura 7.6 se realiza una comparativa de estas dos magnitudes para ambos motores.

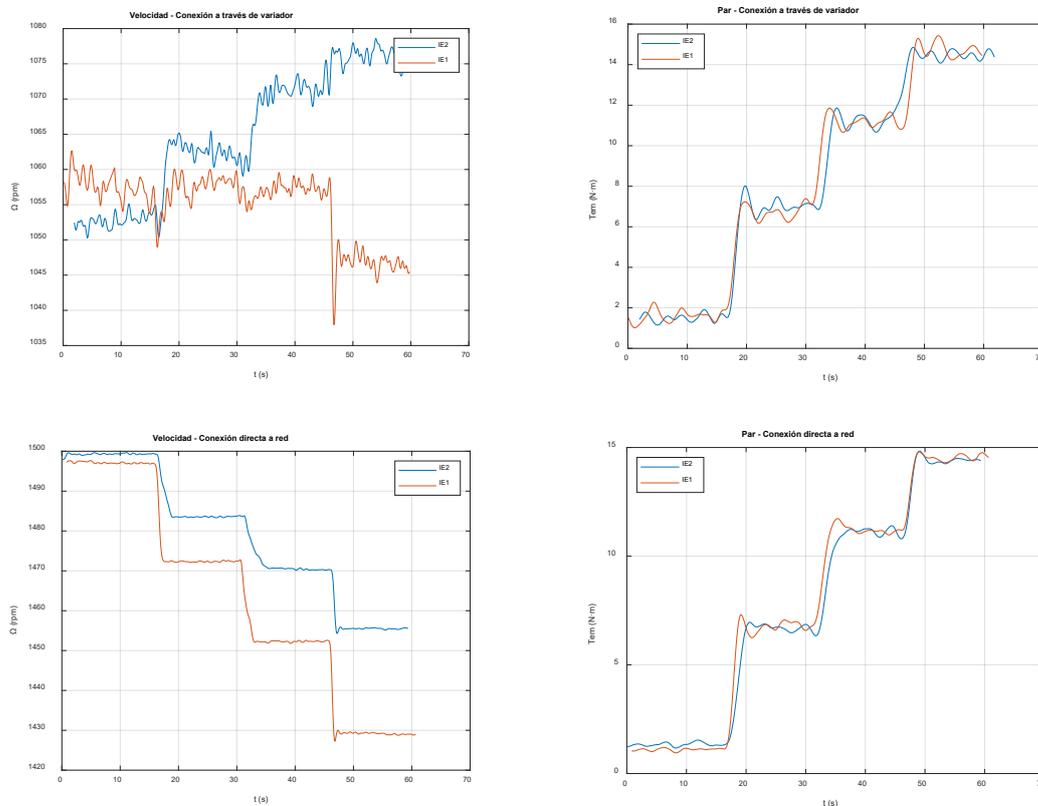


Figura 7.6. Comparativa velocidad y par.

Por una parte, se evidencia que, como era de esperar, el par transmitido al eje es el mismo en ambos casos, corroborando que se están ensayando los motores frente a las mismas condiciones de operación.

Cabe destacar que en las diferentes señales obtenidas se observa un menor nivel de ruido en los ensayos realizados mediante conexión directa a red, siendo posible realizar una lectura relativamente coherente de los datos incluso sin aplicar el proceso de filtrado digital. Un análisis más detallado de los beneficios derivados de la aplicación de un filtro digital se expone en el epígrafe 7.3.

7.3 ANÁLISIS FILTRADO DIGITAL

Seguidamente, se procede a realizar una comparativa de los datos obtenidos sin aplicar el filtro digital con las señales filtradas. Esto se realiza de una forma sencilla simplemente desactivando el proceso de filtrado desde la configuración de la aplicación Android.

Cabe recordar que se han diseñado funciones que permiten aplicar los filtros Butterworth y Hampel a cualquier señal que se desee, permitiendo su aplicación a otros proyectos. En particular, la velocidad, el par mecánico, las corrientes consumidas por el ACIM, y la amplitud y módulo de su corriente de estator se han filtrado mediante el filtro Butterworth, mientras que el filtro Hampel se ha aplicado a la corriente generada por una de las fases del PMSM.

En este apartado se expondrán los resultados obtenidos directamente desde la propia aplicación Android.

En cuanto a la velocidad, debido a que se ha empleado un paso de muestreo relativamente grande, de unos 100ms, las diferencias entre aplicar el filtro digital respecto a la señal original no son tan notorias.

Sin embargo, variables tales como el par mecánico o el nivel de corriente sí que requerían un filtrado previo a la visualización de los datos.

Finalmente, se analiza la aplicación del filtro Butterworth sobre el módulo y amplitud de la corriente consumida por el motor de inducción, tanto en los ensayos realizados mediante su conexión al variador de frecuencia como mediante conexión directa a red. En la figura 7.7 se analizan los resultados de estos ensayos obtenidos mediante la aplicación Android.

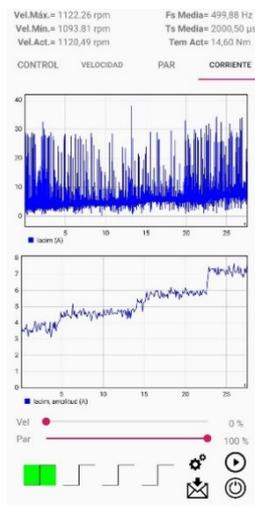


Figura 7.7a. Conexión a variador - Filtro desactivado

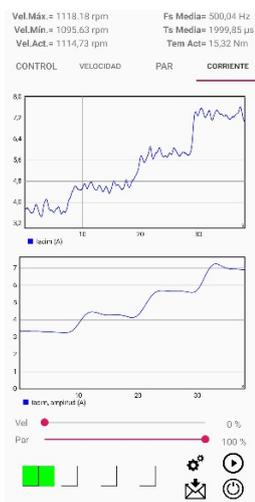


Figura 7.7b. Conexión a variador - Filtro activado

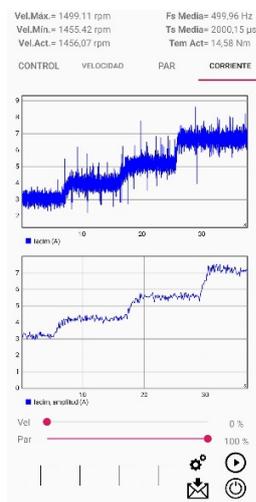


Figura 7.7c. Conexión a red - Filtro desactivado

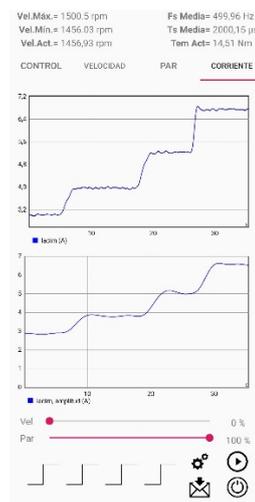


Figura 7.7d. Conexión a red - Filtro activado

Figura 7.7. Análisis filtrado digital - Nivel de corriente.

Por lo que respecta a la conexión del ACIM al variador, en la figura 7.7a se observa que la señal asociada al módulo de la corriente no es posible visualizarla de forma aceptable debido a que la relación señal/ruido es muy pobre, aunque dicha señal evidencia cierta tendencia. Sin embargo, la amplitud sí que se puede representar de forma satisfactoria pese a que el filtrado digital está desactivado. La aplicación del filtro Butterworth proporciona una notable mejora especialmente en la lectura del módulo de la corriente, incrementando en gran medida la relación señal/ruido tal como se observa en la figura 7.7b.

Por otro lado, al realizar los ensayos conectando el ACIM directamente a la red y desactivando el filtro digital, en la figura 7.7c se observa que el módulo de la corriente presenta una relación señal/ruido bastante mayor en comparación al ensayo mediante conexión a través del variador, aunque dicha relación es bastante mejorable. Además, la amplitud presenta una variación bastante menor que en la figura 7.7a. Al activar el filtro, tal como se expone en la figura 7.7d, se obtienen buenos resultados en la lectura de ambas señales.

7.5 COMPARATIVA MODELO DIGITAL Y SISTEMA REAL

Es posible obtener de forma muy sencilla datos provenientes de las simulaciones como del sistema real. En el caso del modelo digital, cuando finaliza una simulación se ha programado que de forma automática se almacenen las variables de interés en un archivo de datos de Matlab. Por otra parte, la obtención de datos reales del laboratorio se efectúa simplemente almacenando en ficheros de texto los registros obtenidos del ensayo experimental y enviándolos a través de un servicio externo, ya sea correo electrónico o un sistema de almacenamiento de datos en la nube.

Los resultados se exponen en la figura 7.10. Es posible observar discrepancias de unas 10rpm en la velocidad mecánica del motor y de unos 0.5A en el módulo de la corriente del estator. Se evidencia, además, que la diferencia entre la corriente real y la simulada se reduce a medida que se incrementa la carga. Cabe destacar que no se han tenido en cuenta efectos de saturación magnética, siendo este uno de los posibles motivos por los cuales existen discrepancias entre el sistema real y el modelo digital.

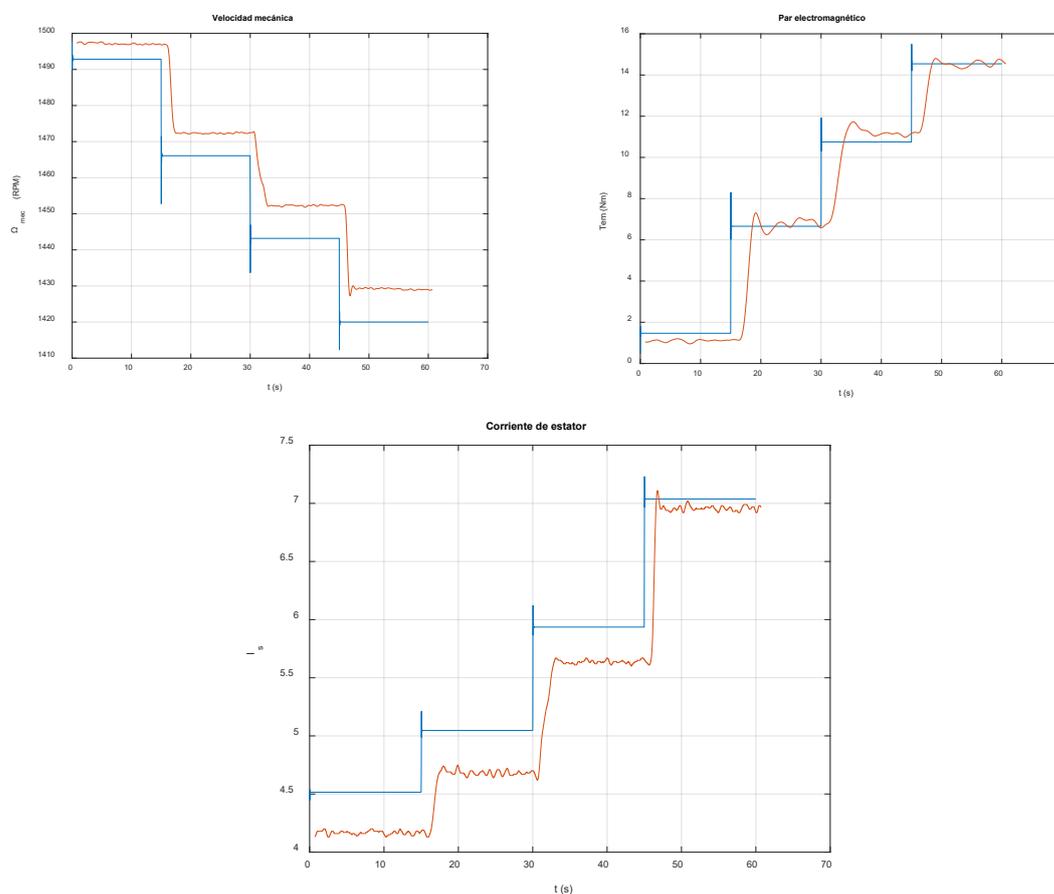


Figura 7.10. Comparación de resultados de simulaciones y banco de ensayos real.

CAPÍTULO 8: CONCLUSIONES Y LÍNEAS FUTURAS

El desarrollo del presente proyecto ha permitido diseñar un modelo digital del banco de ensayos, integrando tanto el motor de inducción como el variador de frecuencia, logrando implementar las estrategias de control escalar, vectorial y directo de par. En definitiva, los modelos diseñados permiten realizar los mismos ensayos que el puesto de prácticas de laboratorio, incluyendo las estrategias de control anteriores y ofreciendo la posibilidad de alimentar el motor mediante conexión directa a red o bien a través del variador de frecuencia. Además, las diferentes simulaciones han sido integradas mediante una interfaz elaborada en App Designer que permite configurar los parámetros característicos de los modelos.

Todos los modelos digitales se han diseñado a partir del sistema de ecuaciones que los definen, sin emplear ningún bloque de la librería Simscape en los que ya vienen integrados, por ejemplo, el modelo del motor de inducción o la técnica SVPWM. Esto permite adaptar las simulaciones a softwares de código abierto tales como Python.

En cuanto a la parte experimental del proyecto, se ha diseñado y desarrollado un nuevo producto basado en dispositivos de bajo coste que permite regular y monitorizar de forma remota el banco de ensayos. Este producto es lo suficientemente versátil como para sustituir a los clásicos PLCs industriales y analizadores de energía, suponiendo una alternativa económica a estos.

Además, todo el código implementado en el microcontrolador y en el dispositivo móvil ha sido diseñado mediante software de código abierto, facilitando su adaptación a cualquier proyecto de este ámbito.

En el ámbito de los accionamientos eléctricos y la industria 4.0, el desarrollo del proyecto tiene multitud de aplicaciones. Los modelos digitales elaborados se podrían integrar en un gemelo digital, comparando la respuesta simulada y la del puesto de prácticas en tiempo real permitiendo, por ejemplo, anticiparse al diagnóstico de averías de los motores.

Por otra parte, el producto diseñado permite comandar cualquier variador de frecuencia y monitorizar el motor controlado, por lo que se puede emplear en cualquier ámbito industrial donde se utilicen variadores de frecuencia y/o motores de inducción. Por ejemplo, se permite controlar y monitorizar el estado de estaciones de bombeo que utilicen bombas de velocidad variable. E incluso, en el caso de bombas de velocidad fija, el producto permite monitorizar variables de interés tales como la velocidad o la corriente consumida por el motor.

Cabe destacar que el producto diseñado no limita sus aplicaciones únicamente al ámbito de los accionamientos eléctricos. Una parte importante del proyecto consiste en el análisis de la eficiencia a través de la corriente consumida. En consecuencia, es posible emplear el producto desarrollado sobre cualquier sistema trifásico, tales como las redes eléctricas inteligentes. Conectando los devanados trifásicos a los terminales de entrada y salida de corriente del producto es posible realizar la lectura del nivel de corriente consumida de una forma muy sencilla.

En lo que respecta a las posibles líneas futuras que se pueden derivar de la elaboración del presente proyecto, es posible destacar:

- Adaptación de las simulaciones de Matlab/Simulink a software de código abierto. Uno de los lenguajes de código abierto ampliamente conocidos y empleados es Python, el cual incluye multitud de librerías cuya aplicación abarca desde el análisis avanzado de datos, técnicas de aprendizaje automático e incluso el control de procesos. Es en este último en el que existen librerías que permiten implementar simulaciones dinámicas tanto de sistemas lineales, tales como reguladores PID, o bien no lineales, como podría ser el motor de inducción. En el capítulo 2 se han planteado las bases para integrar las diferentes simulaciones dentro de este lenguaje de programación, adaptando en el presente proyecto únicamente el motor de inducción mediante conexión directa a red, quedando como líneas futuras la integración del variador de frecuencia con sus diferentes sistemas de control.
- Implementación de un gemelo digital. En los capítulos 2 y 3 se ha elaborado un modelo digital del puesto de prácticas, siendo una de sus principales aplicaciones en el ámbito de los gemelos digitales. Para ello sería necesario simular en tiempo real las diferentes simulaciones y comparar cada cierto intervalo de tiempo la respuesta ofrecida por la simulación con la correspondiente al sistema real.
- Implementación de filtros analógicos de señales. Se han realizado grandes esfuerzos en el diseño e implementación de filtros digitales de tipo Butterworth y Hampel, ofreciendo un ahorro de costes ya que no es necesaria la integración de instrumentación electrónica adicional tal como cables apantallados o condensadores. Como línea futura, se podría plantear filtrar las diferentes señales físicas de forma analógica mediante el empleo de condensadores.
- Ampliación de la arquitectura MQTT. No se han explotado en profundidad todas las posibilidades que ofrece el protocolo MQTT. Se podría instalar un servidor MQTT en plataformas tales como Raspberry Pi e integrar un sistema de análisis de datos en tiempo real, permitiendo, por ejemplo, comparar la respuesta ofrecida por el sistema real con los resultados ofrecidos por la simulación en tiempo real del modelo digital diseñado.

En definitiva, se ha desarrollado con éxito un proyecto multidisciplinar cuyo eje central es la ingeniería eléctrica, pero en el que se han aplicado conceptos de múltiples especialidades de la ingeniería industrial, tales como la ingeniería automática, electrónica e informática. La sinergia de todos estos ámbitos de la ingeniería ha supuesto el desarrollo de un proyecto innovador con amplias aplicaciones a nivel industrial.

BIBLIOGRAFÍA

- [1] Bocker, J. & Mathapati S. (2007). State of the Art of Induction Motor Control. *IEEE Xplore*, 7, 1459-1464. doi: [10.1109/IEMDC.2007.383643](https://doi.org/10.1109/IEMDC.2007.383643).
- [2] Arnanz, R., García, F., & Miguel, L. (2016). Métodos de control de motores de inducción: síntesis de la situación actual. *Revista Iberoamericana de Automática e Informática industrial*, 13(4), 381-392. doi: [10.1016/j.riai.2015.10.001](https://doi.org/10.1016/j.riai.2015.10.001).
- [3] Ioannides, M. & Bejan, C. (2014). Monitoring and Control of a Variable Frequency Drive Using PLC and SCADA. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(10), 3092-3098.
- [4] Pulver, T. (2019). *Hands-On Internet of Things with MQTT*. Birmingham, Reino Unido: Packt.
- [5] Fuller, A., Fan, Z., Day, C. & Barlow, C. (2020). Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Xplore*, 8, 108952-108971. doi: [10.1109/ACCESS.2020.2998358](https://doi.org/10.1109/ACCESS.2020.2998358).
- [6] Martínez, J. & Serrano, L. (2017). *Máquinas Eléctricas*. Valencia, España: Editorial UPV.
- [7] Abu-Rub, H., Iqbal, A. & Guzinski, J. (2012). *High Performance Control of AC Drivers with Matlab/Simulink*. Chichester, United Kingdom: Wiley.
- [8] Kim, S. (2017). *Electric Motor Control*. Elsevier.
- [9] Renukadevi, G. & Rajambal, K. (2011). Generalized model of multi-phase induction motor drive using matlab/Simulink. 114-119. doi: [10.1109/ISET-India.2011.6145366](https://doi.org/10.1109/ISET-India.2011.6145366).
- [10] Zhen, G. & Zhang, Q. (2018). The Study on Mathematical Model and Simulation of Asynchronous Motor Considering Iron Loss. 446-449. doi: [1060. 012085. 10.1088/1742-6596/1060/1/012085](https://doi.org/10.1088/1742-6596/1060/1/012085).
- [11] Yu, Z. (1999). *Space-Vector PWM With TMS320C24x/F24x Using Hardware and Software Determined Switching Patterns (Application Report No. SPRA524)*. Texas Instruments.
- [12] Kameswara, R., Srinivas, P. & Suresh, M. (2014). Design and Analysis of Various Inverters Using Different PWM Techniques. *The International Journal Of Engineering And Science*, 41-51.
- [13] Akin, B. & Garg, N. (2013). *Scalar (V/f) Control of 3-Phase Induction Motors (Application Report No. SPRABP8)*. Texas Instruments.
- [14] Bilal, A. & Bhardwaj, M. (2013). *Sensored Field Oriented Control of 3-Phase Induction Motors (Application Report No. SPRABP8)*. Texas Instruments.
- [15] Bhardwaj, M. (2013). *Sensorless Field Oriented Control of 3-Phase Induction Motors Using F2833x (Application Report No. SPRABQ0)*. Texas Instruments.
- [16] Bhardwaj, M. (2013). *Sensorless Field Oriented Control of 3-Phase Induction Motors Using Control Law Accelerator (Application Report No. SPRABQ1)*. Texas Instruments.
- [17] Seguí, S., Gimeno, F., Masot, R. & Orts, G. (2003). *Control Vectorial de Inversores Trifásicos*. Universitat Politècnica de València.

[18] Franco, A (Consulta: 06 de julio de 2021). Efecto Hall. UPV/EHU.

<http://www.sc.ehu.es/sbweb/fisica3/magnetico/hall/hall.html>

[19] Monari, G. (Consulta: 06 de julio de 2021). Understanding Resolution In Optical And Magnetic Encoders. Electronic Design.

<https://www.electronicdesign.com/technologies/components/article/21798142/understanding-resolution-in-optical-and-magnetic-encoders>

[20] Collins, D. (Consulta: 06 de julio de 2021). How do magnetic encoders work?. Motion Control Tips.

<https://www.motioncontroltips.com/faq-how-do-magnetic-encoders-work/>

[21] Craig, K. (Consulta: 06 de julio de 2021). Actuators & Sensors in Mechatronics - Optical Encoders.

http://engineering.nyu.edu/mechatronics/Control_Lab/Craig/Craig_RPI/SenActinMecha/S&A_Optical_Encoders.pdf

[22] Castillo, A. (2019). Desarrollo de un sistema de realimentación de posición y velocidad para banco de ensayos de accionamientos eléctricos (Trabajo de Fin de Grado). Universitat Politècnica de València.

[23] NEX Instrument (Consulta: 06 de julio de 2021). Siemens – Rotary Encoder.

<https://www.nexinstrument.com/1XP8001-1-1024>

[24] Utmel Elctronic (Consulta: 10 de julio de 2021). Classification of Current Sensors.

<https://www.utmel.com/blog/categories/sensors/classification-of-current-sensors>

[25] Gos, R., Talpone, J. & Raiti, S. (Consulta: 10 de julio de 2021). Transformadores de Instrumentos. Transformación y Distribución de la Energía Eléctrica. Universidad Tecnológica Nacional.

<http://www.frlp.utn.edu.ar/materias/tydee/moduloii.pdf>

[26] Electronics Tutorials (Consulta: 10 de julio de 2021). The Current Transformer.

<https://www.electronics-tutorials.ws/transformer/current-transformer.html>

[27] Liu, C. & Liu, J. (2014). The 5th IEEE International Workshop on Applied Measurements for Offset Error reduction in Open Loop Hall Effect Current Sensors Powered with Single Voltage Source, 1-6. doi: 10.13140/2.1.4077.4401.

[28] Liu, J., Sanli, A. & Liu, C. (2012). Error Compensation of Closed Loop Hall Effect Current Sensors.

[29] LEM (Consulta: 10 de julio de 2021). Current Transducer LTSR 6-NP (Datasheet).

https://www.lem.com/sites/default/files/products_datasheets/ltsr_6-np.pdf

[30] FUTEK – Advanced Sensor Technologies. (Consulta: 14 de julio de 2021). Torque Measurement.

<https://willrich.com/direct-indirect-torque-measurement-techniques-drive-train-technology/>

[31] Texas Instruments. (Consulta: 14 de julio de 2021). Low-Cost, CMOS, Rail-to-Rail, I/O Operational Amplifiers.

https://www.ti.com/lit/ds/symlink/opa705.pdf?ts=1629218180176&ref_url=https%253A%252F%252Fwww.google.com%252F

[32] Espressif Systems. (Consulta: 16 de julio de 2021). ESP-WROVER-KIT V4.1 Getting Started Guide.

https://www.mouser.com/datasheet/2/891/Espressif_Systems_01162019_ESP_WROVER_KIT_V4-1522995.pdf

[33] Kester, W. (2003). Mixed-signal and DSP Design Techniques.

[34] Smith, S. (2002). Digital Signal Processing: A Practical Guide for Engineers and Scientists. Newnes.

[35] Yao, Z., Wang, R., Zhi & J. Shi, Q. (2020). Robust Locally Weighted Regression for Profile Measurement of Magnesium Alloy Tube in Hot Bending Process. Mathematical Problems in Engineering. 2020. 1-10. 10.1155/2020/7952649.

[36] Wicklin, R. (Consulta: 02 de agosto de 2021). The Hampel Identifier: Robust outlier detection in a time series. SAS News & Announcements.

<https://blogs.sas.com/content/iml/2021/06/01/hampel-filter-robust-outliers.html>

[37] Burgess, P. (Consulta: 03 de agosto de 2021). Adafruit GFX Graphics Library. Adafruit Learning System.

<https://learn.adafruit.com/adafruit-gfx-graphics-library>

[38] DIY Projects. (Consulta: 03 de agosto de 2021). ESP32 - Get started with the SPIFFS.h library to read, write, modify files.

<https://diyprojects.io/esp32-get-started-spiff-library-read-write-modify-files/>

[39] Random Nerd Tutorials. (Consulta: 03 de agosto de 2021). ESP32 Useful Wi-Fi Library Functions (Arduino IDE).

<https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/>

[40] Last Minute Engineers. (Consulta: 03 de agosto de 2021). Configuring & Handling ESP32 GPIO Interrupts In Arduino IDE.

<https://lastminuteengineers.com/handling-esp32-gpio-interrupts-tutorial/>

[41] Luis Llamas. (Consulta: 03 de agosto de 2021). Ordenación rápida con Arduino mediante algoritmo QuickSort.

<https://www.luisllamas.es/arduino-quicksort/>

[42] Castillo, A. (2021). TFM [Código Fuente].

<https://github.com/castillo97/TFM>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

**PROYECTO DIGITALIZACIÓN DE UN PUESTO DE
PRÁCTICAS DE LABORATORIO PARA ANÁLISIS DE LA
EFICIENCIA DE MOTORES Y SU INTEGRACIÓN EN LA
INDUSTRIA 4.0**

PRESUPUESTO

AUTOR: ANDREU CASTILLO PENADÉS

TUTOR: ÁNGEL SAPENA BAÑÓ

COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21

La elaboración del presupuesto pretende cuantificar el coste económico derivado de la elaboración del proyecto. El presupuesto ha sido dividido en un total de 4 capítulos que incluyen la elaboración del modelo digital, el diseño y montaje del prototipo, el diseño del software integrando la programación del microcontrolador y de la aplicación Android y, finalmente, la redacción de la memoria.

En cada uno de estos capítulos se detallarán los costes tanto materiales como humanos. En estos costes se incluirán todos los elementos que han intervenido en el desarrollo del modelo digital y del producto diseñado, de modo que quedan fuera del ámbito del presupuesto elementos tales como los motores de inducción y de imanes permanentes, los variadores de frecuencia, el encoder y el sensor de efecto Hall acoplado a una de las fases del motor de imanes permanentes, es decir, no se incluyen aquellos elementos sobre los que actúa el producto diseñado.

En cuanto al uso del ordenador personal, se ha considerado un coste de 1000€ amortizado durante 36 meses, por lo que el precio unitario derivado de su uso es de 27.78€/mes.

CAPÍTULO 01 DESARROLLO DEL MODELO DIGITAL				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
MATERIALES	Computador personal	3 meses	27.78 €/mes	83 €
	Software Matlab (Licencia corporativa)	1 ud.	2000 €/ud.	2000 €
HUMANOS	Estudios previos	30 h	17 €/h	510 €
	Diseño e implementación	60 h	17 €/h	1020 €
	Pruebas	10 h	17 €/h	170 €
COSTE CAPÍTULO 01:				3783 €

Por lo que respecta al diseño del del modelo digital, los estudios previos se corresponden con el análisis del estado del arte de la modelización de motores asíncronos y al estudio de las diferentes estrategias de control implementadas.

CAPÍTULO 02 DISEÑO Y MONTAJE DEL PROTOTIPO				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
MATERIALES	ESP32 WROVER KIT	1 ud.	34.20 €/ud.	34.20 €
	Módulo de 8 relés	1 ud.	6.95 €/ud.	6.95 €
	Amplificador operacional OPA705	2 ud.	0.96 €/ud.	1.92 €
	Convertidor DC/DC B2412S -1W	1 ud.	1.00 €/ud.	1.00 €
	Convertidor DC/DC B2405S -1W	1 ud.	0.56 €/ud.	0.56 €
	Sensor de efecto Hall ltr 6np	2 ud.	15.68 €/ud.	31.36 €
	Perfboard	1 ud.	4.24 €/ud.	4.24 €
	Caja de conexiones	1 ud.	3.00 €/ud.	3.00 €
	Tornillería	6 ud.	0.39 €/ud.	2.34 €
	Cable de cobre	3 ud.	0.51 €/ud.	1.53 €
	Soldador de estaño tipo F	1 ud.	22.34 €/ud.	22.34 €
	Estaño	200 m	0.05 €/m	9.72 €
	Conector DB9	3 ud.	2.57 €/ud.	7.71 €
HUMANOS	Estudios previos	10 h	17 €/h	170 €
	Diseño e implementación	100 h	17 €/h	1700 €
	Pruebas	20 h	17 €/h	340 €
COSTE CAPÍTULO 02:				2336.87 €

Los estudios previos del capítulo 2 han sido invertidos en la valoración de diferentes tecnologías asociadas al conjunto de sensores implementados en el proyecto. Los precios de los diferentes componentes electrónicos del capítulo tienen como fuente los fabricantes Mouser Electronics y RS Components.

CAPÍTULO 03 DISEÑO DEL SOFTWARE				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
MATERIALES	Analog discovery 2	1 ud.	420.77 €/ud.	420.77 €
	Computador personal	3 ud.	27.78 €/mes	83.34 €
	Sonda osciloscópica	2 ud.	30.66 €/ud.	61.32 €
	Software Arduino IDE	1 ud.	0 €/ud.	0 €
	Software Matlab (Licencia corporativa)	1 ud.	0 €/ud.	0 €
	Software Android Studio	1 ud.	0 €/ud.	0 €
HUMANOS	Estudios previos	30 h	17 €/h	510 €
	Diseño e implementación	150 h	17 €/h	2550 €
	Pruebas	20 h	17 €/h	340 €
COSTE CAPÍTULO 03:				3965.43 €

En el capítulo 3, los estudios previos incluyen el estudio de los diferentes procesos de filtrado digital que se han analizado en la memoria.

CAPÍTULO 04 MEMORIA				
RECURSOS	CONCEPTO	CANTIDAD	PRECIO UNITARIO	COSTE
MATERIALES	Computador personal	2 meses	27.78 €/mes	56 €
HUMANOS	Redacción de la memoria	100 h	17 €/h	1700 €
COSTE CAPÍTULO 04:				1756 €

Finalmente, se resume el coste asociado a cada uno de los capítulos anteriores y se expone el coste total al que asciende el presupuesto.

RESUMEN PRESUPUESTO	
CONCEPTO	COSTE
Capítulo 01. Desarrollo del modelo digital	3783.34 €
Capítulo 02. Diseño y montaje del prototipo	2336.87 €
Capítulo 03. Diseño del software	3965.43 €
Capítulo 04. Memoria	1755.56 €
Presupuesto de ejecución material	11841.20 €
Gastos generales (12%)	1420.94 €
Beneficio industrial (6%)	710.47 €
Presupuesto de ejecución contrata	13972.62 €
IVA (21%)	2934.25 €
Presupuesto base de licitación	16906.87 €

El presupuesto total asciende a la cantidad de “DIECISÉIS MIL NOVECIENTOS SEIS CON OCHENTA Y SIETE EUROS”



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



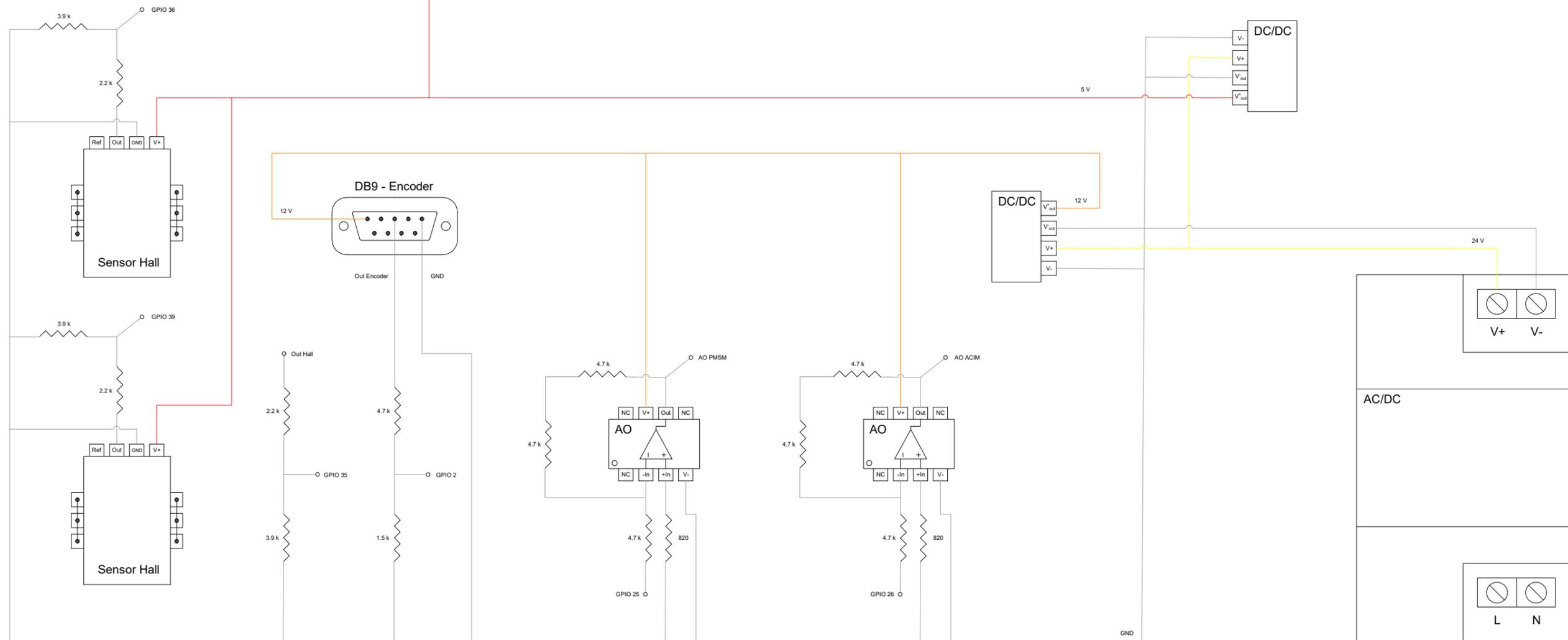
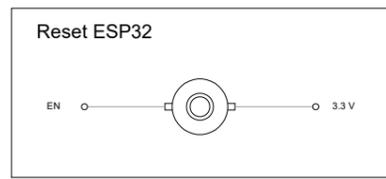
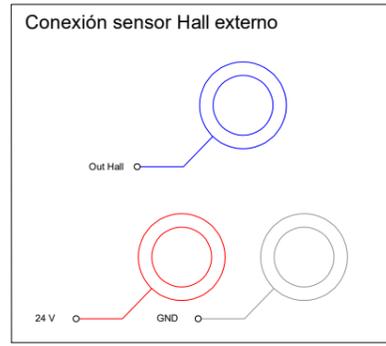
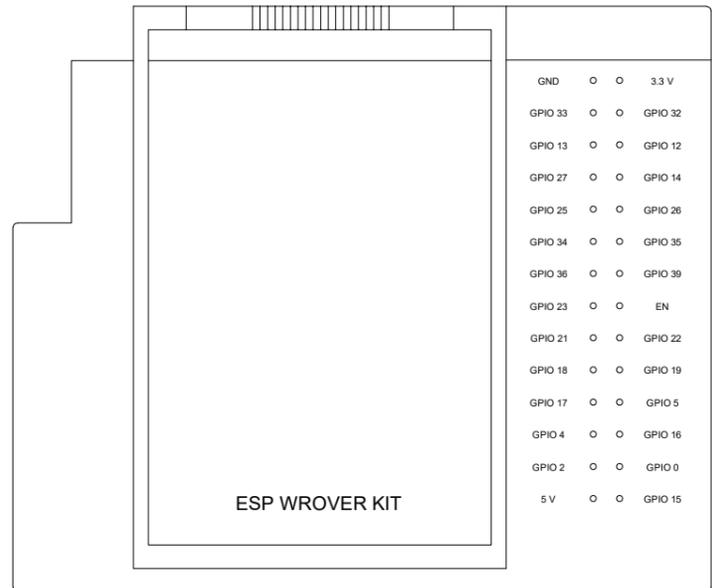
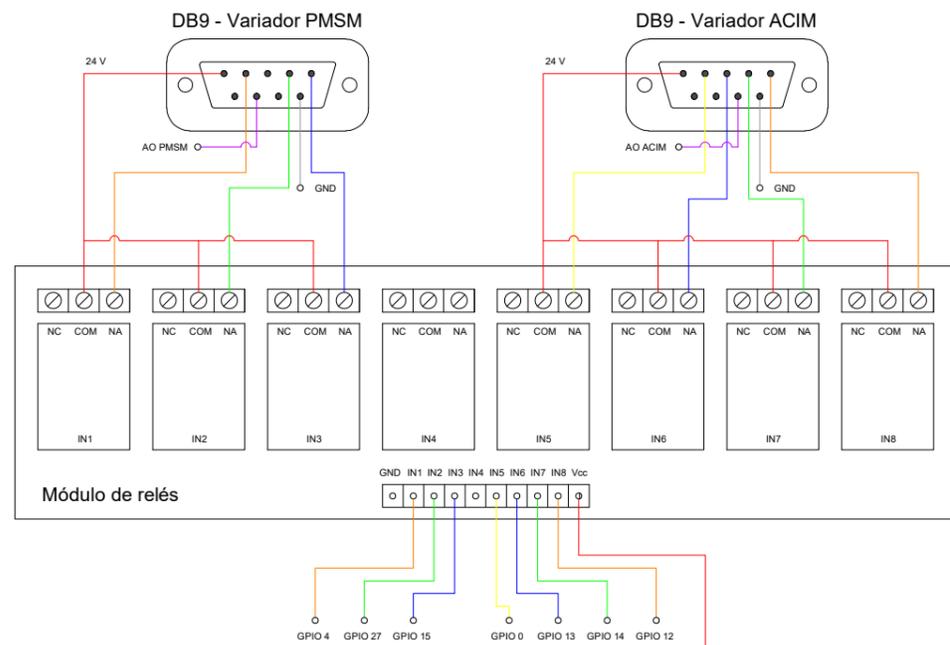
ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

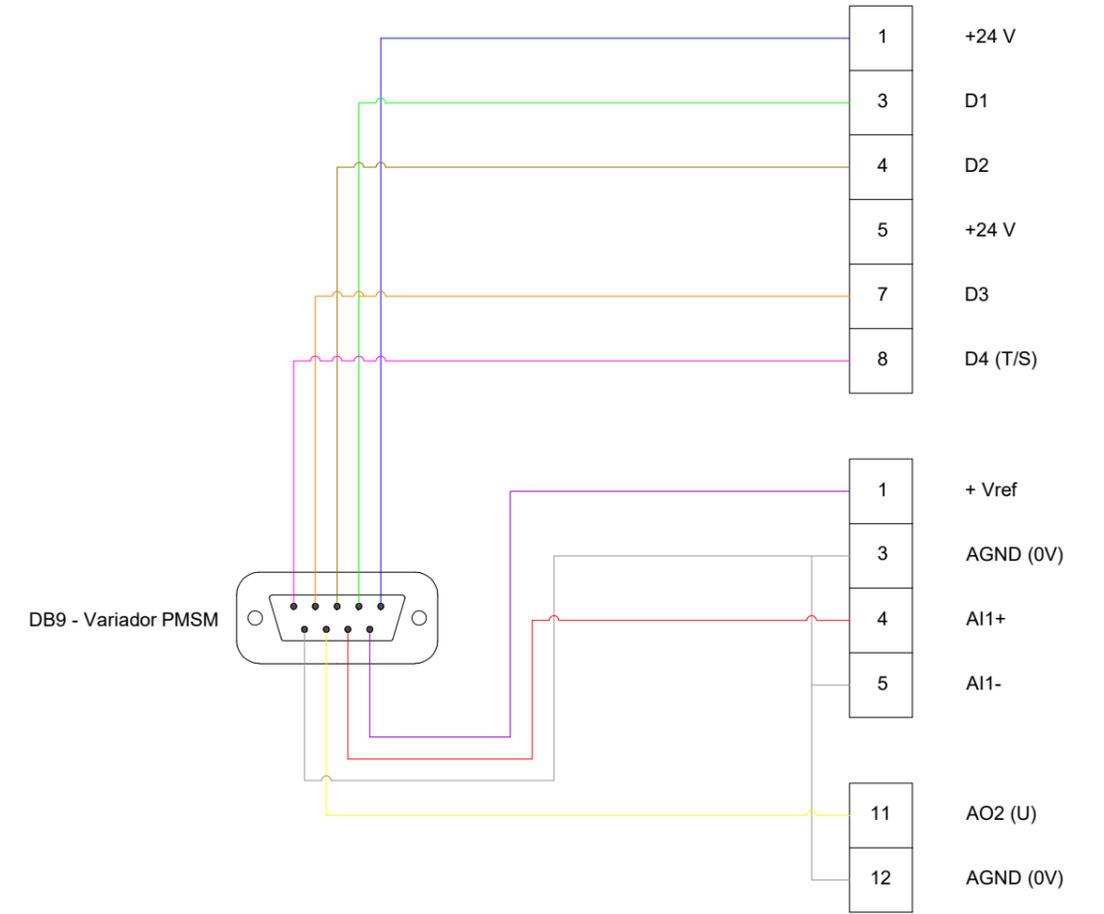
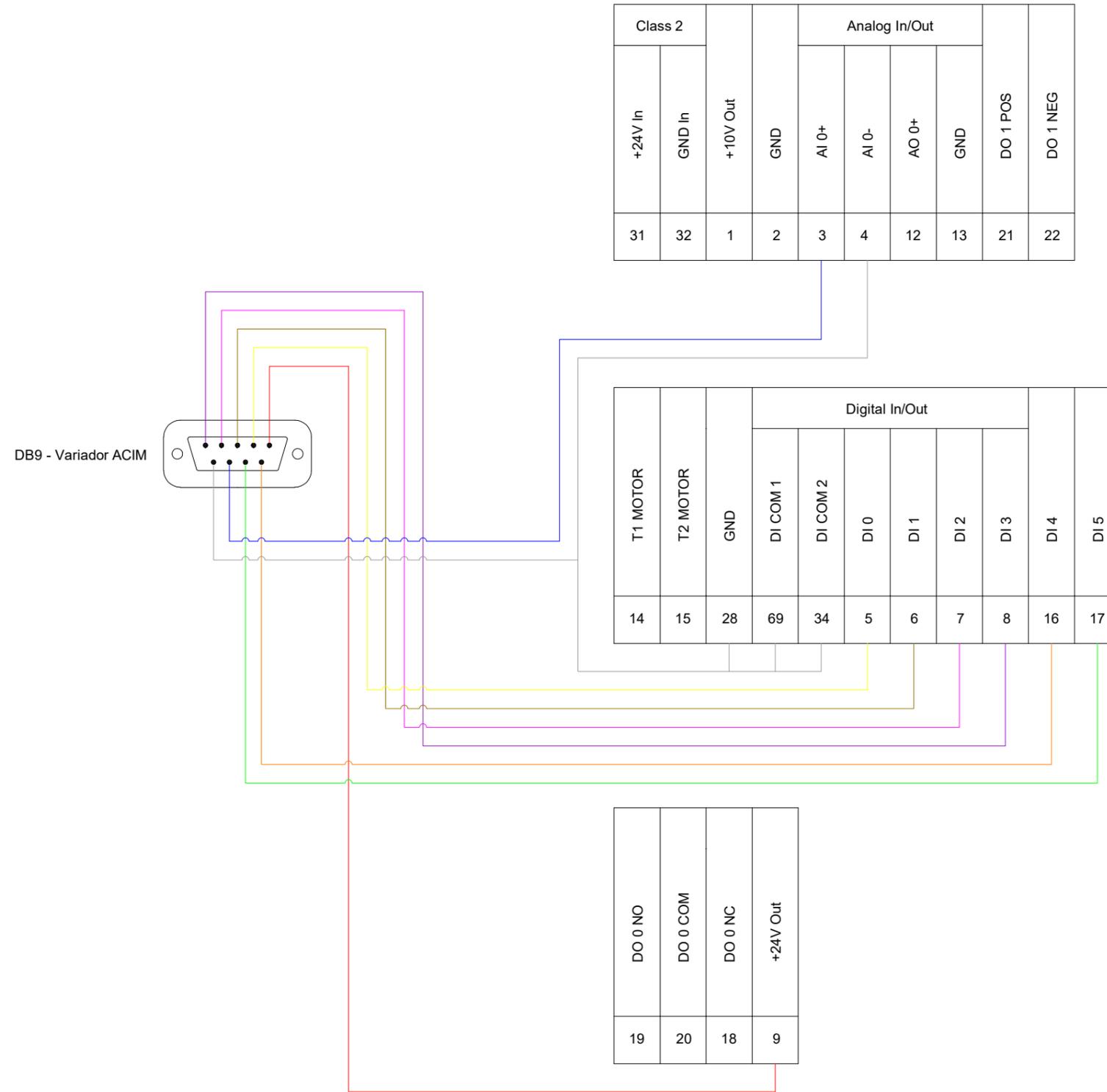
**PROYECTO DIGITALIZACIÓN DE UN PUESTO DE
PRÁCTICAS DE LABORATORIO PARA ANÁLISIS DE LA
EFICIENCIA DE MOTORES Y SU INTEGRACIÓN EN LA
INDUSTRIA 4.0**

PLANOS

AUTOR: ANDREU CASTILLO PENADÉS
TUTOR: ÁNGEL SAPENA BAÑÓ
COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21







UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

**PROYECTO DIGITALIZACIÓN DE UN PUESTO DE
PRÁCTICAS DE LABORATORIO PARA ANÁLISIS DE LA
EFICIENCIA DE MOTORES Y SU INTEGRACIÓN EN LA
INDUSTRIA 4.0**

ANEXOS

AUTOR: ANDREU CASTILLO PENADÉS
TUTOR: ÁNGEL SAPENA BAÑÓ
COTUTOR: RUBÉN PUCHE PANADERO

Curso Académico: 2020-21

ÍNDICE DE CONTENIDOS

1	ANEXO 1: MODELADO Y SIMULACIÓN EN MATLAB/SIMULINK	1
1.1	MODELADO DEL MOTOR DE INDUCCIÓN EN EJES D-Q.....	1
1.2	TÉCNICA SVPWM BASADA EN COMPONENTES α, β	4
2	ANEXO 2: RESOLUCIÓN NUMÉRICA Y VALIDACIÓN DEL MODELO DEL ACIM.....	5
2.1	RESOLUCIÓN NUMÉRICA.....	5
2.2	VALIDACIÓN E IMPLEMENTACIÓN EN EL MICROCONTROLADOR.....	6
3	ANEXO 3: DISEÑO DE FILTROS DIGITALES MEDIANTE MATLAB.....	9
3.1	FILTRO BUTTERWORTH	9
3.2	FILTRO HAMPEL.....	11
4	ANEXO 4: CÓDIGO DISEÑADO	12

ÍNDICE DE FIGURAS

Figura 1.1.	Ecuaciones de flujo de estator y rotor.	1
Figura 1.2.	Ecuaciones de corriente de estator y rotor – Forma implícita.....	2
Figura 1.3.	Ecuaciones de corriente de estator y rotor – Forma explícita.	3
Figura 1.4.	Ecuaciones mecánicas.	3
Figura 1.5.	Diagrama general del modelo del ACIM	3
Figura 1.6.	Validación técnica SVPWM basada en componentes α, β	4
Figura 4.1.	Estructura del código diseñado.....	12
Figura 4.2.	Código de la aplicación Android.....	12
Figura 4.3.	Código desarrollado en Matlab/Simulink.....	13

ÍNDICE DE CÓDIGOS

Código 2.1.	Definición de variables.....	7
Código 2.2.	Resolución mediante ode45.	7
Código 2.3.	Resolución numérica.....	8
Código 3.1.	Carga de datos.....	10
Código 3.2.	Análisis filtro Butterworth.....	10
Código 3.3.	Validación del filtro Butterworth.	10
Código 3.4.	Validación del filtro Hampel.....	11

ANEXO 1: MODELADO Y SIMULACIÓN EN MATLAB/SIMULINK

En el presente anexo se detallará, en primer lugar, el modelado del motor de inducción en ejes móviles d, q . Tal como se ha comentado en la memoria, al inicio del proyecto se había desarrollado un modelo del ACIM en ejes móviles cuya velocidad es la de sincronismo. Debido a que este modelo requiere el conocimiento de la velocidad de sincronismo, la cual es difícilmente accesible en los controles vectorial y directo de par, finalmente se ha descartado este modelo y se ha optado por el modelo en ejes estacionarios para el desarrollo del proyecto.

Por otra parte, este primer anexo expone las ecuaciones necesarias para implementar la técnica SVPWM que presenta como entradas las componentes α, β .

1.1 MODELADO DEL MOTOR DE INDUCCIÓN EN EJES D-Q

Se procede a presentar el sistema de ecuaciones diferenciales que modelan el comportamiento dinámico del ACIM en ejes síncronos. Se describirán los diferentes conjuntos de ecuaciones y su implementación en Simulink.

Las ecuaciones dinámicas de flujo de estator y rotor se detallan en 1.1, mientras que su implementación en Simulink se expone en la figura 1.1

$$(1.1) \quad \begin{aligned} \frac{d\psi_{sd}}{dt} &= v_{sd} - i_{sd} \cdot R_s - \omega_s \cdot \psi_{sq}, & \frac{d\psi_{sq}}{dt} &= v_{sq} - i_{sq} \cdot R_s + \omega_s \cdot \psi_{sd} \\ \frac{d\psi_{rd}}{dt} &= -i_{rd} \cdot R_r + (\omega_s - \omega_r) \cdot \psi_{rq}, & \frac{d\psi_{rq}}{dt} &= -i_{rq} \cdot R_r - (\omega_s - \omega_r) \cdot \psi_{rd} \end{aligned}$$

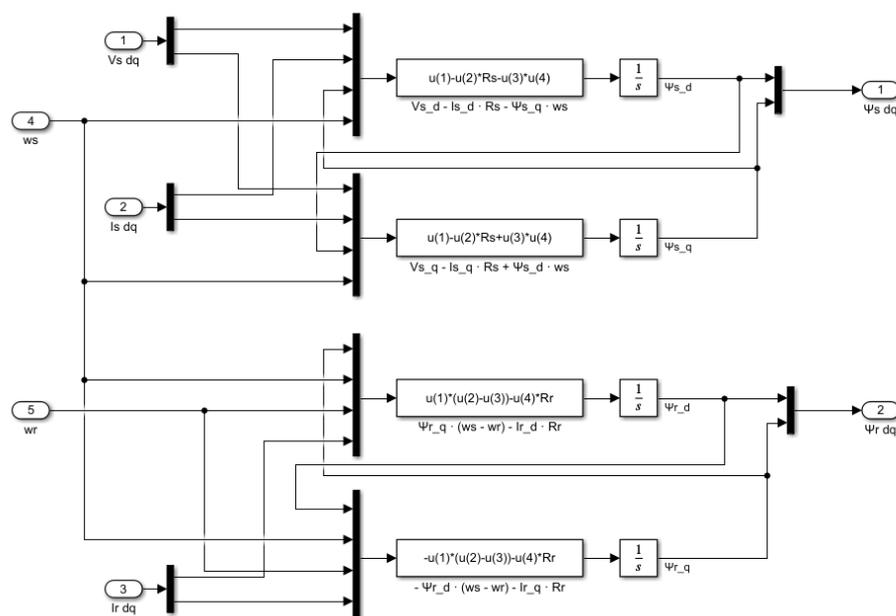


Figura 1.1. Ecuaciones de flujo de estator y rotor.

Es posible expresar los flujos de estator y rotor en función de las respectivas corrientes tal como sigue:

$$(1.2) \quad \begin{aligned} \psi_{sd} &= (i_{sd} + i_{rd}) \cdot L_m + i_{sd} \cdot L_{ls}, & \psi_{sq} &= (i_{sq} + i_{rq}) \cdot L_m + i_{sq} \cdot L_{ls} \\ \psi_{rd} &= (i_{sd} + i_{rd}) \cdot L_m + i_{rd} \cdot L_{lr}, & \psi_{rq} &= (i_{sq} + i_{rq}) \cdot L_m + i_{rq} \cdot L_{lr} \end{aligned}$$

Despejando las corrientes de las ecuaciones anteriores se obtiene el siguiente sistema de ecuaciones, cuya resolución permite obtener las corrientes de estator y rotor en ejes d, q

$$(1.3) \quad \begin{aligned} i_{sd} &= \frac{\psi_{sd} - i_{rd} \cdot L_m}{L_{ls} + L_m}, & i_{sq} &= \frac{\psi_{sq} - i_{rq} \cdot L_m}{L_{ls} + L_m} \\ i_{rd} &= \frac{\psi_{rd} - i_{sd} \cdot L_m}{L_{lr} + L_m}, & i_{rq} &= \frac{\psi_{rq} - i_{sq} \cdot L_m}{L_{lr} + L_m} \end{aligned}$$

Es posible implementar el sistema de ecuaciones anterior directamente en Simulink tal como se expone en la figura 1.2.

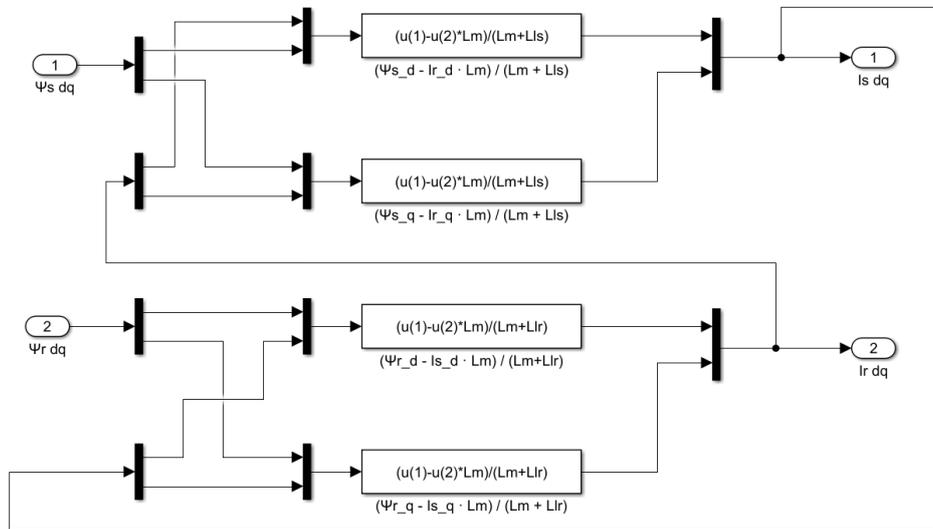


Figura 1.2. Ecuaciones de corriente de estator y rotor – Forma implícita.

Tal como se puede observar en las ecuaciones anteriores, no existe acoplamiento entre el eje d y el eje q , de tal modo que es posible resolver el sistema anterior de forma sencilla, obteniendo las corrientes en función únicamente de los flujos de rotor y estator.

$$(1.4) \quad \begin{aligned} i_{sd} &= \frac{L_{lr} \cdot \psi_{sd} + L_m \cdot \psi_{sd} - L_m \cdot \psi_{rd}}{L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m} = \frac{L_r \cdot \psi_{sd} - L_m \cdot \psi_{rd}}{L_x} \\ i_{sq} &= \frac{L_{lr} \cdot \psi_{sq} + L_m \cdot \psi_{sq} - L_m \cdot \psi_{rq}}{L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m} = \frac{L_r \cdot \psi_{sq} - L_m \cdot \psi_{rq}}{L_x} \\ i_{rd} &= \frac{L_{ls} \cdot \psi_{rd} + L_m \cdot \psi_{rd} - L_m \cdot \psi_{sd}}{L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m} = \frac{L_s \cdot \psi_{rd} - L_m \cdot \psi_{sd}}{L_x} \\ i_{rq} &= \frac{L_{ls} \cdot \psi_{rq} + L_m \cdot \psi_{rq} - L_m \cdot \psi_{sq}}{L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m} = \frac{L_s \cdot \psi_{rq} - L_m \cdot \psi_{sq}}{L_x} \end{aligned}$$

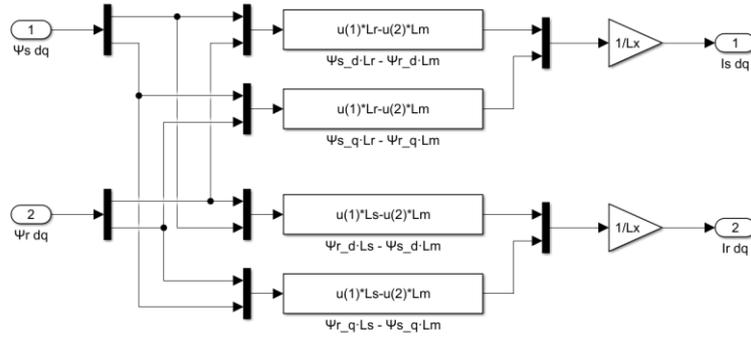


Figura 1.3. Ecuaciones de corriente de estator y rotor – Forma explícita.

Finalmente, se detalla el conjunto de ecuaciones dinámicas y su implementación en Simulink.

$$T_{em} = \frac{3}{2} \cdot p \cdot \frac{Lm}{Ll_r + Lm} \cdot (i_{sq} \cdot \psi_{rd} - i_{sd} \cdot \psi_{rq})$$

$$(1.5) \quad \frac{d\Omega_r}{dt} = \frac{T_{em} - F \cdot \Omega_r - T_{carga}}{J}$$

$$\theta = \int \Omega_r \cdot dt$$

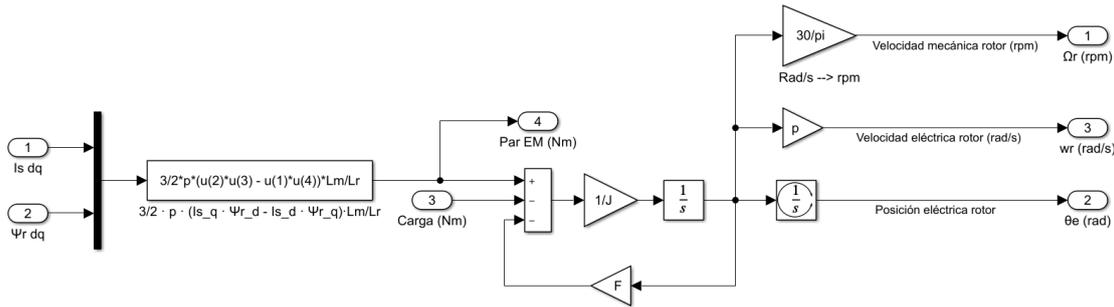


Figura 1.4. Ecuaciones mecánicas.

El esquema general, en el que se observan los diferentes bloques interconectados, se expone en la figura 1.5.

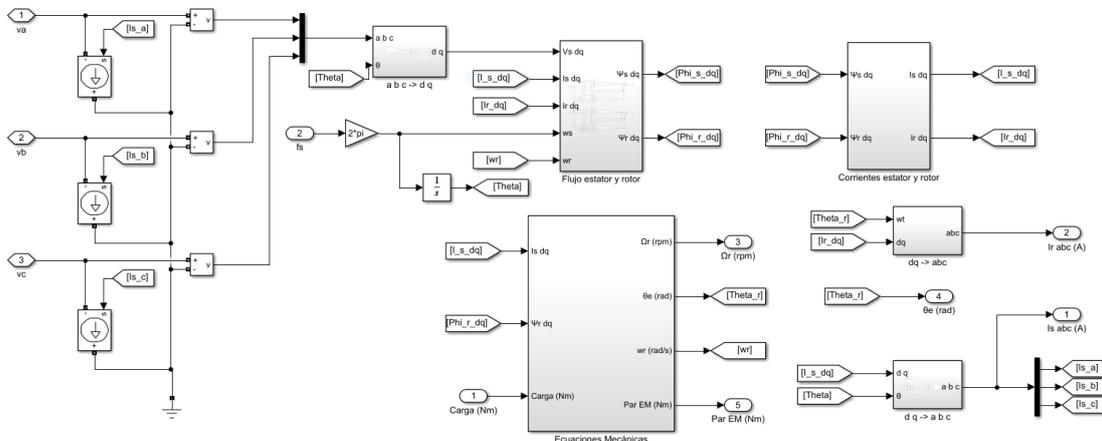


Figura 1.5. Diagrama general del modelo del ACIM

1.2 TÉCNICA SVPWM BASADA EN COMPONENTES α, β

Se procede a demostrar las ecuaciones que permiten obtener los tiempos de aplicación para el primer sector. Para ello es necesario sustituir en la ecuación 1.6 los vectores adyacentes al sector 1 y descomponerlos en sus componentes real e imaginaria.

$$(1.6) \quad \vec{V}_{ref} \cdot t_z = \vec{V}_1 \cdot t_a + \vec{V}_2 \cdot t_b + \vec{V}_0 \cdot t_0$$

$$(1.7) \quad \begin{aligned} V_{ref} e^{j\theta} \cdot t_z &= 2/3 V_{dc} e^{j0} \cdot t_a + 2/3 V_{dc} e^{j\frac{\pi}{3}} \cdot t_b \\ V_{\beta} \cdot t_z &= 2/3 V_{dc} \cdot t_b \cdot \sin\left(\frac{\pi}{3}\right) \rightarrow t_b = \sqrt{3} \cdot \frac{V_{\beta}}{V_{dc}} \cdot t_z \\ V_{\alpha} \cdot t_z &= 2/3 V_{dc} \cdot \left(t_a + t_b \cdot \frac{1}{2}\right) \rightarrow t_a = \frac{t_z}{2 \cdot V_{dc}} \cdot (3 \cdot V_{\alpha} - \sqrt{3} \cdot V_{\beta}) \end{aligned}$$

Procediendo de forma análoga se obtienen las ecuaciones generales aplicables a cualquier sector. Cabe destacar que en los sectores pares es necesario alternar las ecuaciones de t_a y t_b .

$$(1.8) \quad \begin{aligned} t_a &= \frac{t_z \cdot \sqrt{3}}{V_{dc}} \cdot \left(V_{\alpha} \cdot \text{sen}\left(k \cdot \frac{\pi}{3}\right) - V_{\beta} \cdot \text{cos}\left(k \cdot \frac{\pi}{3}\right) \right) \\ t_b &= \frac{t_z \cdot \sqrt{3}}{V_{dc}} \cdot \left(V_{\beta} \cdot \text{cos}\left((k-1) \cdot \frac{\pi}{3}\right) - V_{\alpha} \cdot \text{sen}\left((k-1) \cdot \frac{\pi}{3}\right) \right) \\ t_z &= t_a + t_b + t_0 \rightarrow t_0 = t_z - t_a - t_b \end{aligned}$$

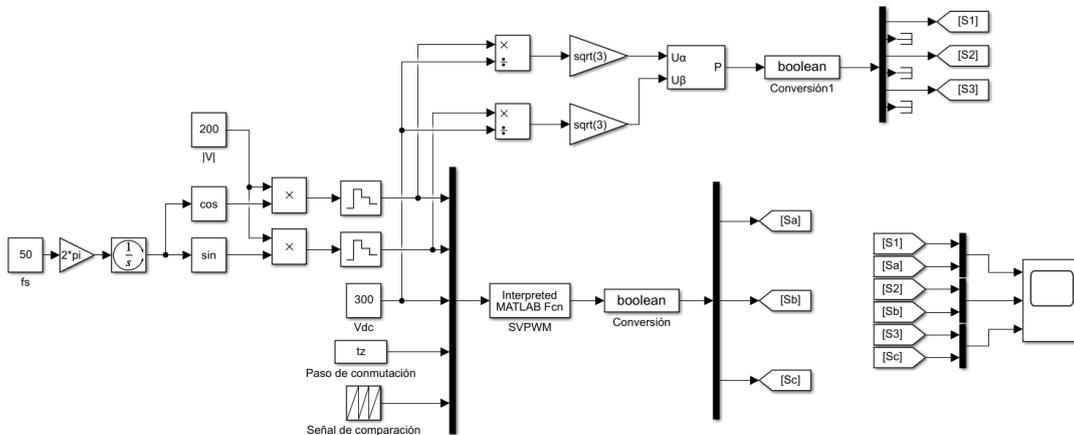


Figura 1.6. Validación técnica SVPWM basada en componentes α, β .

ANEXO 2: RESOLUCIÓN NUMÉRICA Y VALIDACIÓN DEL MODELO DEL ACIM

2.1 RESOLUCIÓN NUMÉRICA

Con el objetivo de poder implementar el modelo del ACIM en el microcontrolador, es necesario resolver de forma numérica el sistema no lineal de ecuaciones diferenciales de primer orden que componen el modelo del motor ya que no existe una solución explícita del mismo.

Para la resolución del modelo del motor, se plantea en primer lugar la resolución del problema de valores iniciales de primer orden, definido según la expresión 2.1.

$$(2.1) \quad y'(t) = f(t, y), \quad y(t_0) = y_0$$

La resolución numérica de dicho problema consiste en obtener unas aproximaciones $\{y_1, y_2, \dots\}$ de los valores exactos $\{y(t_1), y(t_2), \dots\}$, donde cada instante de tiempo t_i se asocia a un valor discreto de tiempo definido según la ecuación 2.2, donde h representa el tamaño de paso, el cual se ha definido como constante.

$$(2.2) \quad t_i = t_0 + h \cdot i$$

El problema de valores iniciales se ha resuelto mediante el método más sencillo disponible en la literatura, es decir, el método de Euler. Para ello se aplica un desarrollo de Taylor truncando en el término de primer orden, por lo que se obtiene:

$$(2.3) \quad y(t_i + h) \approx y(t_i) + y'(t_i) \cdot h \rightarrow y'(t_i) \approx \frac{y(t_i + h) - y(t_i)}{h}$$

Denotando por y_i una aproximación de la función $y(t_i)$ en el instante t_i se obtiene la ecuación que permite determinar las aproximaciones y_{n+1} de $y(t_i + h)$ de forma recursiva.

$$(2.4) \quad y'(t_i) \approx y'_n = f(t_n, y_n) = \frac{y_{n+1} - y_n}{h} \rightarrow y_{n+1} = h \cdot f(t_n, y_n) + y_n$$

En consecuencia, aproximando las derivadas presentes en las diferentes ecuaciones de estado del modelo del ACIM es posible obtener las aproximaciones de las variables de estado. De este modo, se procede a resolver de forma numérica el sistema de ecuaciones diferenciales.

Ecuaciones de flujo de estator y rotor:

$$(2.5) \quad \begin{aligned} \left(\frac{d\psi_{s\alpha}}{dt}\right)^n &= v_{s\alpha}^n - i_{s\alpha}^n \cdot R_s = \frac{\psi_{s\alpha}^{n+1} - \psi_{s\alpha}^n}{h} \rightarrow \psi_{s\alpha}^{n+1} = (v_{s\alpha}^n - i_{s\alpha}^n \cdot R_s) \cdot h + \psi_{s\alpha}^n \\ \left(\frac{d\psi_{s\beta}}{dt}\right)^n &= v_{s\beta}^n - i_{s\beta}^n \cdot R_s = \frac{\psi_{s\beta}^{n+1} - \psi_{s\beta}^n}{h} \rightarrow \psi_{s\beta}^{n+1} = (v_{s\beta}^n - i_{s\beta}^n \cdot R_s) \cdot h + \psi_{s\beta}^n \end{aligned}$$

$$\begin{aligned}
(2.6) \quad \left(\frac{d\psi_{r\alpha}}{dt}\right)^n &= -i_{r\alpha}^n \cdot R_r - w_r^n \cdot \psi_{r\beta}^n = \frac{\psi_{r\alpha}^{n+1} - \psi_{r\alpha}^n}{h} \rightarrow \\
&\rightarrow \psi_{r\alpha}^{n+1} = (-i_{r\alpha}^n \cdot R_r - w_r^n \cdot \psi_{r\beta}^n) \cdot h + \psi_{r\alpha}^n \\
\left(\frac{d\psi_{r\beta}}{dt}\right)^n &= -i_{r\beta}^n \cdot R_r + w_r^n \cdot \psi_{r\alpha}^n = \frac{\psi_{r\beta}^{n+1} - \psi_{r\beta}^n}{h} \rightarrow \\
&\rightarrow \psi_{r\beta}^{n+1} = (-i_{r\beta}^n \cdot R_r + w_r^n \cdot \psi_{r\alpha}^n) \cdot h + \psi_{r\beta}^n
\end{aligned}$$

Ecuaciones de corrientes asociadas a estator y rotor:

$$(2.7) \quad i_{s\alpha}^n = \frac{L_r \cdot \psi_{s\alpha}^n - L_m \cdot \psi_{r\alpha}^n}{L_x}, \quad i_{s\beta}^n = \frac{L_r \cdot \psi_{s\beta}^n - L_m \cdot \psi_{r\beta}^n}{L_x}$$

$$(2.8) \quad i_{r\alpha}^n = \frac{L_s \cdot \psi_{r\alpha}^n - L_m \cdot \psi_{s\alpha}^n}{L_x}, \quad i_{r\beta}^n = \frac{L_s \cdot \psi_{r\beta}^n - L_m \cdot \psi_{s\beta}^n}{L_x}$$

Donde la inductancia L_x viene definida por la ecuación 2.9.

$$(2.9) \quad L_x = L_{lr} \cdot L_{ls} + L_{lr} \cdot L_m + L_{ls} \cdot L_m$$

Finalmente, se realiza el cálculo del par electromagnético y, a partir de un balance de momentos aplicado sobre el eje del motor eléctrico, se determina la velocidad de posición angular del mismo.

$$(2.10) \quad T_{em} = \frac{3}{2} \cdot p_{ACIM} \cdot \frac{L_m}{L_r} \cdot (i_{s\beta}^n \cdot \psi_{r\alpha}^n - i_{s\alpha}^n \cdot \psi_{r\beta}^n)$$

Obtención de la velocidad angular:

$$\begin{aligned}
(A2.11) \quad \left(\frac{d\Omega_r}{dt}\right)^n &= \frac{T_{em}^n - F \cdot \Omega_r^n - T_{carga}^n}{J} = \frac{\Omega_r^{n+1} - \Omega_r^n}{h} \rightarrow \\
&\rightarrow \Omega_r^{n+1} = \frac{T_{em}^n - F \cdot \Omega_r^n - T_{carga}^n}{J} \cdot h + \Omega_r^n
\end{aligned}$$

Cálculo de la posición angular:

$$(2.12) \quad \Omega_r^n = \left(\frac{d\theta}{dt}\right)^n = \frac{\theta^{n+1} - \theta^n}{h} \rightarrow \theta^{n+1} = \Omega_r^n \cdot h + \theta^n$$

La resolución numérica del modelo ha sido implementada y validada mediante Matlab

2.2 VALIDACIÓN E IMPLEMENTACIÓN EN EL MICROCONTROLADOR

Previamente a la implementación de la resolución numérica propuesta en el ESP32, es necesario validar la simulación. Para ello, se ha elaborado un script de Matlab que permite comparar la respuesta ofrecida por el método de resolución de ecuaciones diferenciales ode45 y la respuesta obtenida mediante el procedimiento descrito anteriormente para diferentes tamaños de paso.

Por lo descrito anteriormente, el código de Matlab desarrollado presenta dos partes bien diferenciadas: por una parte, se debe de resolver el sistema de ecuaciones mediante el método ode45 para posteriormente comparar sus respuesta con la ofrecida por el método propuesto en el epígrafe anterior.

Inicialmente, se definen una serie de variables que deben ser accesibles desde las diferentes funciones desarrolladas y que se asocian con los parámetros del motor de inducción.

```
global Rs Rr Ls Lr Lm Lx J F p

% Parámetros del modelo:
Rs = 3.3; Rr = 2.9050;
Lls = 0.0138; Llr = 0.0138;
Lm = 0.2167;
Ls = Lm + Lls; Lr = Lm + Llr;
Lx = Llr*Lls + Llr*Lm + Lls*Lm;

J = 0.01; p = 2; F = 0.007;

tmax = 1.5; Tc = 5; f = 50;
```

Código 2.1. Definición de variables.

La resolución mediante ode45 se efectúa de forma similar a cualquier lenguaje de programación que permita la resolución de sistemas de ecuaciones diferenciales.

```
function dydt = odefcn(t,y,f,Tc)
global Rs Rr Ls Lr Lm Lx J F p
Av = 230/50*f;

% Tensión trifásica:
Va = Av*cos(2*pi*f*t);
Vb = Av*cos(2*pi*f*t-2*pi/3);
Vc = Av*cos(2*pi*f*t-4*pi/3);

% Transformada directa de Clarke
vsa = (2*Va - Vb - Vc)/3;
vsb = sqrt(3)/3*(Vb-Vc);

Phi_sa = y(1); Phi_sb = y(2);
Phi_ra = y(3); Phi_rb = y(4);

wm = y(5); wr = wm*p;

isa = (Lr*Phi_sa - Lm*Phi_ra)/Lx; isb = (Lr*Phi_sb - Lm*Phi_rb)/Lx;
ira = (Ls*Phi_ra - Lm*Phi_sa)/Lx; irb = (Ls*Phi_rb - Lm*Phi_sb)/Lx;
Tem = 3/2*p*Lm/Lr*(isb*Phi_ra - isa*Phi_rb);

dydt = zeros(5,1);
dydt(1) = vsa - isa*Rs;
dydt(2) = vsb - isb*Rs;
dydt(3) = -ira*Rr - wr*Phi_rb;
dydt(4) = -irb*Rr + wr*Phi_ra;
dydt(5) = (Tem - F*wm - Tc)/J;
end
```

Código 2.2. Resolución mediante ode45.

La función que permite resolver de forma numérica el modelo del ACIM, y la que posteriormente se debe de adaptar al ESP32, es la siguiente:

```
function [Phi_sa2, Phi_sb2, Phi_ra2, Phi_rb2, wm2, Tem] = modeloACIM(Phi_sa, Phi_sb, Phi_ra,
Phi_rb, wm, Tc, t, f, h)
global Rs Rr Ls Lr Lm Lx J F p
Av = 230/50*f;

% Tensión trifásica:
Va = Av*cos(2*pi*f*t);
Vb = Av*cos(2*pi*f*t-2*pi/3);
Vc = Av*cos(2*pi*f*t-4*pi/3);

% Transformada directa de Clarke
vsa = (2*Va - Vb - Vc)/3;
vsb = sqrt(3)/3*(Vb-Vc);

% Velocidad eléctrica del rotor:
wr = wm*p;

isa = (Lr*Phi_sa - Lm*Phi_ra)/Lx; isb = (Lr*Phi_sb - Lm*Phi_rb)/Lx;
ira = (Ls*Phi_ra - Lm*Phi_sa)/Lx; irb = (Ls*Phi_rb - Lm*Phi_sb)/Lx;

Phi_sa2 = vsa*h - isa*Rs*h + Phi_sa; Phi_sb2 = vsb*h - isb*Rs*h + Phi_sb;
Phi_ra2 = -ira*Rr*h - wr*Phi_rb*h + Phi_ra; Phi_rb2 = -irb*Rr*h + wr*Phi_ra*h + Phi_rb;

Tem = 3/2*p*Lm/Lr*(isb*Phi_ra - isa*Phi_rb);
wm2 = (Tem - F*wm - Tc)*h/J + wm;
end
```

Código 2.3. Resolución numérica.

ANEXO 3: DISEÑO DE FILTROS DIGITALES MEDIANTE MATLAB

Previamente a la implementación de cualquier filtro discreto en un microcontrolador es necesario diseñarlo y validarlo mediante Matlab. El presente anexo tiene como objetivo presentar los pasos a seguir para desarrollar y validar los algoritmos asociados a los filtros Butterworth y Hampel.

3.1 FILTRO BUTTERWORTH

Seguidamente, se expone el código de Matlab que permite obtener los coeficientes del filtro digital, analizar los resultados de aplicar el proceso de filtrado y la respuesta en frecuencia de la función de transferencia asociada frente a diferentes frecuencias de corte y orden del filtro digital.

Durante el desarrollo del código se han empleado un conjunto de funciones, cuyo detalle se expone a continuación:

- $[b, a] = butter(n, f_c/(f_s/2))$. Permite obtener los coeficientes del filtro digital de tipo Butterworth de orden n que se emplean en la implementación de la ecuación en diferencias asociada.
- $y = filter(b, a, x)$. Aplica la ecuación en diferencias de un filtro digital, no necesariamente de tipo Butterworth, sobre una señal x a partir de los coeficientes a, b obtenidos del diseño del filtro.
- $FdT = filt(b, a, T_s)$. Calcula la función de transferencia discreta dados los coeficientes del numerador y denominador de la misma (parámetros a y b). El paso de muestreo en base al cual se ha diseñado la función de transferencia se define en la variable T_s .
- $bodeplot(FdT)$. Representa gráficamente el diagrama de bode de una determinada función de transferencia, es decir, representa la amplitud y desfase de dicha función de transferencia. Mediante el parámetro `setoptions` es posible configurar las unidades deseadas o graficar únicamente la amplitud o el desfase.
- $N = numel(x)$. Devuelve como valor el número total de elementos que componen el vector x .
- $flip(x)$. Invierte el orden de los elementos del vector x .

Inicialmente, es necesario cargar los datos experimentales correspondientes a la señal muestreada que se desea filtrar. Esto se realiza en el código 3.1, donde además se representa gráficamente la señal original, permitiendo evidenciar la necesidad de aplicar un proceso de filtrado digital.

```

datos=load('datos2.txt');
t = datos(:,1); var = datos(:,3);

plot(t,var), grid on
eje_y = "Tem (Nm)";
ylabel(eje_y), xlabel("t (s)")

```

Código 3.1. Carga de datos.

Seguidamente, se realizan tres análisis asociado al filtro Butterworth, aplicando el proceso de filtrado sobre la señal original y comparando los resultados obtenidos frente a diferentes frecuencias de corte definidas por el usuario.

```

Ts = 0.002; fs = 1/Ts; fc = [1 5]; n = 4;

% Análisis y comparativa de la aplicación del proceso de filtrado digital
% Butterworth de 4o orden sobre una variable muestreada dado un conjunto
% de frecuencias de corte definidas por el usuario:
 analisisFrecc(fs,fc,var,n,t)
 xlabel("t (s)", ylabel(eje_y)

function analisisFrecc(fs,fc,x,n,t)
plot(t,x), hold on
for i=1: numel(fc)
    [b,a] = butter(n,fc(i)/(fs/2));
    % Aplicamos el filtro digital:
    y = filter(b,a,x);
    % Comparación señal original y filtrada:
    plot(t,y), grid on
end
legend(['x' (string(fc)+' Hz')]), hold off
end

```

Código 3.2. Análisis filtro Butterworth.

Si la señal filtrada presenta las características deseadas, el siguiente paso consiste en generar el algoritmo que se encuentra implementado internamente en la función *filter* y que consiste en aplicar la ecuación en diferencias del filtro *Butterworth*.

```

x = zeros(1,n+1); % Vector de muestras anteriores - Señal original
y = zeros(1,n+1); % Vector de muestras anteriores - Señal filtrada
y_v = zeros(1,numel(var));
for i=1: numel(var)
    for j=1:n % Se desplazan los vectores de muestras
        x(j) = x(j+1);
        y(j) = y(j+1);
    end
    x(end) = var(i);
    y(end) = b*flip(x') - a(2:end)*flip(y(1:end-1)');
    y_v(i) = y(end);
end

```

Código 3.3. Validación del filtro Butterworth.

3.2 FILTRO HAMPEL

El filtro Hampel permite la identificación y eliminación de *outliers* de forma online. Mediante Matlab es posible aplicar este filtro digital de forma sencilla mediante la siguiente expresión:

- $y = \text{hampel}(x, k, N_{\sigma})$. El parámetro x se corresponde con el vector de datos no filtrados, k es el tamaño de media ventana y N_{σ} es el umbral (número de veces la desviación típica) a partir del cual un dato es identificado como un *outlier*.

En cada iteración, es necesario calcular la mediana de la ventana de datos, calcular la desviación mediana absoluta y finalmente aplicar el proceso de identificación y eliminación de un *outlier*. El código de Matlab que permite validar el algoritmo Hampel previamente a su implementación se detalla a continuación.

```
function [y,j,xmedian,xsigma] = hampelFilter(x,k,Nsigma)
n = numel(1,x); y = zeros(n,1); j = zeros(n,1);
xmedian = zeros(n,1); xsigma = zeros(n,1);

for i=1:n
    if i<=k
        w = x(1:i+k);
    elseif i<=n-k
        w = x(i-k:i+k);
    else
        w = x(i-k:end);
    end
    xmedian(i) = median(w); xsigma(i) = 1.4826*mad(w,1);

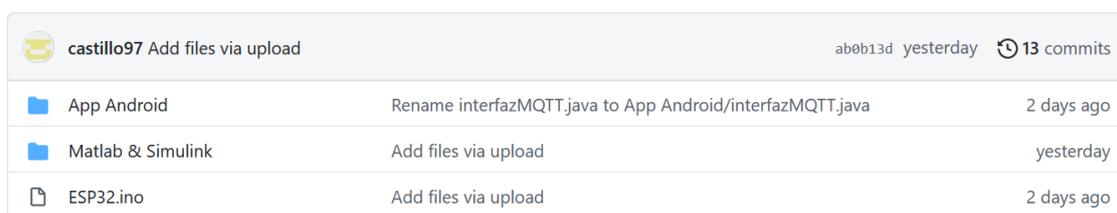
    if abs(x(i) - xmedian(i)) > Nsigma*xsigma(i)
        y(i) = xmedian(i);
        j(i) = 1;
    else
        y(i) = x(i);
    end
end
end
```

Código 3.4. Validación del filtro Hampel.

ANEXO 4: CÓDIGO DISEÑADO

Tal como se ha comentado en la memoria del presente documento, todo el software diseñado, tanto en Matlab/Simulink, el código implementado en el microcontrolador mediante Arduino IDE como el código java asociado a la aplicación Android ha sido publicado en un repositorio online en GitHub. Este anexo está dedicado a presentar la estructura del código diseñado durante la elaboración del proyecto.

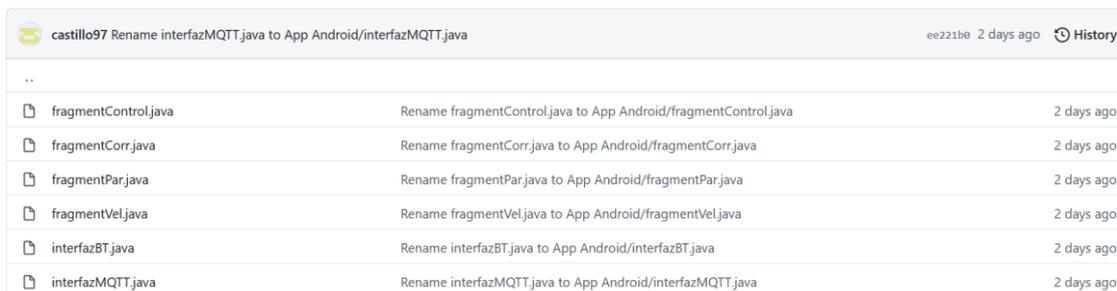
Tal como se detalla en la figura 4.1, el software desarrollado se divide en tres grupos: la aplicación Android, las simulaciones de Matlab/Simulink y el código implementado en el microcontrolador ESP32.



Repository	Commit Message	Timestamp
castillo97	Add files via upload	ab0b13d yesterday 13 commits
App Android	Rename interfazMQTT.java to App Android/interfazMQTT.java	2 days ago
Matlab & Simulink	Add files via upload	yesterday
ESP32.ino	Add files via upload	2 days ago

Figura 4.1. Estructura del código diseñado.

En cuanto a la aplicación Android, tal como se observa en la figura 4.2, se detallan todos los archivos que se encargan de gestionar todas las funcionalidades que ofrece la aplicación desarrollada.



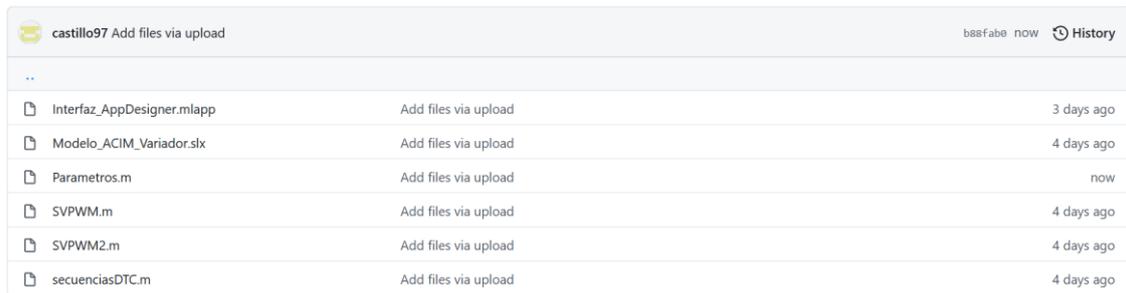
Repository	Commit Message	Timestamp
castillo97	Rename interfazMQTT.java to App Android/interfazMQTT.java	ee221b0 2 days ago History
..		
fragmentControl.java	Rename fragmentControl.java to App Android/fragmentControl.java	2 days ago
fragmentCorr.java	Rename fragmentCorr.java to App Android/fragmentCorr.java	2 days ago
fragmentPar.java	Rename fragmentPar.java to App Android/fragmentPar.java	2 days ago
fragmentVel.java	Rename fragmentVel.java to App Android/fragmentVel.java	2 days ago
interfazBT.java	Rename interfazBT.java to App Android/interfazBT.java	2 days ago
interfazMQTT.java	Rename interfazMQTT.java to App Android/interfazMQTT.java	2 days ago

Figura 4.2. Código de la aplicación Android.

En particular, las funciones que integran cada uno de los ficheros anteriores son las siguientes:

- fragmentControl. Gestiona el envío de las consignas de velocidad/frecuencia y de carga resistente.
- fragmentCorr. Se encarga de la visualización del nivel de corriente consumida por el ACIM a través de la amplitud y módulo de esta.
- fragmentPar. Representa gráficamente la corriente generada por el PMSM y el par transmitido al eje.
- fragmentVel. Visualiza la posición y velocidad angular.
- interfazBT. Gestiona el funcionamiento interno asociado a la transmisión de datos mediante *bluetooth*.
- interfazMQTT. Gestiona la transmisión de datos mediante el protocolo MQTT y la visualización de la velocidad y nivel de corriente consumida a través de internet.

Seguidamente, en cuanto a las simulaciones realizadas en Matlab/Simulink, en la figura 4.3 se exponen todos los ficheros generados.



The screenshot shows a file upload interface for a user named 'castillo97'. The interface includes a header with the user's name and a 'History' button. Below the header is a table listing files that have been uploaded. Each row contains a file icon, the filename, the upload method ('Add files via upload'), and the time since upload.

File Name	Upload Method	Time
Interfaz_AppDesigner.mlapp	Add files via upload	3 days ago
Modelo_ACIM_Variador.slx	Add files via upload	4 days ago
Parametros.m	Add files via upload	now
SVPWM.m	Add files via upload	4 days ago
SVPWM2.m	Add files via upload	4 days ago
secuenciasDTC.m	Add files via upload	4 days ago

Figura 4.3. Código desarrollado en Matlab/Simulink.

Concretamente, las funcionalidades que implementa cada fichero son las siguientes:

- Interfaz_AppDesigner. Integra la configuración de las simulaciones y también la visualización de un conjunto de magnitudes.
- Modelo_ACIM_Variador. Implementa el modelo digital del motor de inducción y el variador de frecuencia con las correspondientes estrategias de control.
- Parámetros. Carga al *workspace* de Matlab los parámetros del circuito equivalente del motor de inducción calculados a partir de los ensayos de este.
- SVPWM. Implementación de la técnica SVPWM basada en las componentes α, β de la tensión de referencia.
- SVPWM2. Implementación de la técnica SVPWM basada en el ángulo y módulo de la tensión de referencia.
- secuenciasDTC. Calcula las secuencias de conmutación en el control directo de par.

Finalmente, todo el código implementado en el microcontrolador se encuentra disponible en el fichero ESP32.ino.