

TRABAJO FIN DE MÁSTER

Máster en Ingeniería Industrial

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

“Modelado y control multivariable del
vehículo urbano eléctrico UAL-eCARM”



Curso 2018/2019

Alumno/a:

Francisco José Mañas Álvarez

Director/es:

José Luis Blanco Claraco
Francisco de Asís Rodríguez Díaz

UNIVERSIDAD DE ALMERÍA
ESCUELA SUPERIOR DE INGENIERÍA



MÁSTER EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

*Modelado y control multivariable del vehículo urbano eléctrico
UAL-eCARM*

Alumno: Francisco José Mañas Álvarez
Director: José Luis Blanco Claraco
Codirector: Francisco de Asís Rodríguez Díaz
Fecha: Septiembre de 2019

A ti.

Índice general

	Página
Agradecimientos	VII
Resumen	IX
Abstract	XI
Siglas y acrónimos	XIII
Índice de figuras	XX
Índice de tablas	XXII
1. Introducción	3
1.1. Introducción e interés	3
1.2. Objetivos	4
1.3. Contexto	5
1.4. Resumen de resultados	7
1.5. Fases de desarrollo y planificación	12
1.6. Competencias empleadas en este proyecto	13
1.7. Estructura del TFM	14
2. Revisión bibliográfica	15
2.1. Introducción	15
2.2. El vehículo eléctrico	16
2.3. Los sistemas X-by-Wire	17
2.4. Modelado en EVs	19
2.5. Control en vehículos eléctricos y autónomos	20
3. Vehículo eCARM	25
3.1. Introducción	25
3.2. Arquitectura hardware	26
3.2.1. Distribución de energía	26
3.2.2. Controlador Curtis	27
3.2.3. PC Industrial	27
3.2.4. Motor principal	29
3.2.5. Motor dirección y freno	30

3.2.6.	Codificador absoluto	30
3.2.7.	Placa microcontrolador Claraquino	31
3.2.8.	Monitor de baterías	31
3.2.9.	Relé.	31
3.2.10.	Codificadores	32
3.2.11.	Amperímetros	32
3.2.12.	Esquemas de conexiones	32
3.3.	Arquitectura software	33
3.3.1.	Robotics Operating System	33
3.3.2.	Nodos de <i>ROS</i> para el control de dirección y velocidad del vehículo . . .	34
3.4.	Arquitectura firmware	43
3.4.1.	Battery_charge	43
3.4.2.	Ual_ecar_vehicle_controller	44
4.	Modelado de la dinámica del vehículo	57
4.1.	Introducción	57
4.2.	Sistema Steer-by-wire	57
4.2.1.	Modelo Tensión-Giro	59
4.2.2.	Modelo Giro-Corriente	68
4.3.	Sistema Brake-by-wire	73
4.4.	Sistema Throttle-by-wire	77
4.4.1.	Modelo Acelerador-Velocidad	79
4.4.2.	Modelo Acoplamiento Velocidad-Dirección	85
4.5.	Vehículo UAL-eCARM	89
5.	Control del sistema <i>Drive-by-Wire</i>	93
5.1.	Introducción	93
5.2.	Control sistema <i>Steer-by-wire</i>	93
5.2.1.	Lazo interno. Control de velocidad de giro	94
5.2.2.	Lazo externo. Control del ángulo de Ackermann	97
5.3.	Control sistema <i>Brake-by-wire</i>	102
5.4.	Control sistema <i>Throttle-by-wire</i>	104
5.5.	Arquitectura de control completa	113
6.	Conclusiones y trabajos futuros	117
6.1.	Conclusiones	117
6.2.	Trabajos futuros	118
	Anexos	121
	A. Configuración actual Curtis	121
	B. Radio de Curvatura del vehículo.	123

Agradecimientos

Toda mi gratitud se la debo a aquellas personas que han estado a mi lado en esta etapa del viaje. Es para con los que han estado para celebrar los éxitos y no han dudado en apoyarme y alentarme en los momentos de difíciles. Cuando todo comenzó, este momento parecía algo muy lejano en el horizonte, pero ahora todo parece haber pasado en un abrir y cerrar de ojos.

También quiero plasmar mis mejores deseos y agradecimientos a mis compañeros de viaje. Todos somos diferentes y tenemos algo que nos hace únicos, y eso ha hecho que la maquinaria de la rutina en clase funcionase como el mejor de los relojes suizos. Han sido esos descansos entre clases y diapositivas los que harán de esta etapa algo duradero en la memoria.

Resumen

Los sistemas de gran complejidad, como los vehículos, necesitan que todos sus elementos funcionen bien tanto de forma individual como en conjunto. Si la sinergia entre los distintos subsistemas es elevada, mejor es el desempeño de los resultados obtenidos.

Este trabajo presenta como objetivo la mejora de la plataforma UAL-eCARM, un vehículo eléctrico urbano. El trabajo comprende el modelado y control de los sistemas de dirección y tracción del vehículo. Para ello se han mejorado los modelos y arquitecturas de control del mecanismo *Steer-by-Wire*. El sistema *Throttle-by-Wire*, un sistema no lineal, ha sido modelado como un sistema lineal entorno a cuatro puntos de operación. La estrategia de control implementada es un controlador PI adaptativo según el ajuste de Ziegler-Nichols. El sistema *Brake-by-Wire*, el más reciente instalado, se ha modelado y controlado mediante un controlador PI ajustado manualmente. La implementación está disponible en *Github*.

Las principales aportaciones de este trabajo son la ampliación y mejora de los modelos del vehículo UAL-eCARM y la migración de la arquitectura de control desde los nodos de ROS a microcontroladores.

Palabras clave: *Multiple-Input Multiple-Output (MIMO), Control PID, ROS*

Abstract

Highly complex systems, such as vehicles, need all their elements to work well both individually and together. If the synergy between the different subsystems is high, the performance of the results obtained is better.

This work presents as goal to improve the UAL-eCARM platform, an urban electric vehicle. The work includes the modeling and control of the steering and traction systems of the vehicle. To achieve it, the models and control architectures of the *Steer-by-Wire* mechanism have been improved. The *Throttle-by-Wire system*, a non-linear system, has been modeled as a linear system around four operating points. The control strategy implemented is an adaptive PI controller according to Ziegler-Nichols tuning. The *Brake-by-Wire* system, the most recently installed, has been modeled and controlled by a manually tuning PI controller. The implementation is available in *Github*.

The main contributions of this work are the expansion and improvement of the UAL-eCARM vehicle models and the migration of the control architecture from ROS nodes to microcontrollers.

Keywords: *Multiple-Input Multiple-Output (MIMO), PID Control, ROS*

Siglas y acrónimos

ROS	Sistema Operativo Robótico (del inglés <i>Robotic Operating System</i>)
NI	National Instrument
DbW	Drive By Wire
SbW	Steer By Wire
BbW	Brake By Wire
TbW	Throttle By Wire
TFG	Trabajo Fin de Grado
UAL	Universidad de Almería
ARM	Grupo de Investigación TEP-197 <i>Automática, Robótica y Automática</i>
UAL-eCARM	Universidad de Almería - Electric Car ARM
IMU	Unidad de Medición Inercia(del inglés <i>Inertial Measurement Unit</i>)
GPS	Sistema de Posicionamiento Global (del inglés <i>Global Positioning System</i>)
MRPT	Herramientas para Programar Robots Móviles (del inglés <i>Mobile Robot Programming Toolkit</i>)
PI	Controlador de acción Proporcional e Integral
PD	Controlador de acción Proporcional y Derivativa
PID	Controlador de acción Proporcional, Integral y Derivativa
CPU	Unidad Central de Procesamiento (del inglés <i>Central Processing Unit</i>)
IoT	Internet de las Cosas (del inglés <i>Internet of Things</i>)

IoV	Internet de los Vehículos (del inglés <i>Internet of Vehicles</i>)
IDAE	Instituto para la Diversificación y Ahorro de la Energía
EV	Vehículo Eléctrico (del inglés <i>Electric Vehicle</i>)
V2G	Del inglés <i>Vehicle to Grid</i>)
G2V	Del inglés <i>Grid to Vehicle</i>)
MGPC	Control Predictivo Generalizado Multivariable (del inglés <i>Multivariable Generalized Predictive Controller</i>)
GPC	Control Predictivo Generalizado (del inglés <i>Generalized Predictive Controller</i>)
MPC	Control Predictivo basado en Modelo (del inglés <i>Model Predictive Controller</i>)
RRT	Exploración de Árbol de forma Aleatoria (del inglés <i>Rapidly exploring Random Tree</i>)
PRM	Mapa de Ruta Probabilístico (del inglés <i>Probabilistic RoadMap</i>)
RRT*FN	Exploración de Árbol de forma Aleatoria con Nodos Fijos (del inglés <i>Rapidly exploring Random Tree Fixes Nodes</i>)
RRT*FND	Exploración de Árbol de forma Aleatoria con Nodos Fijos Dinámicos (del inglés <i>Rapidly exploring Random Tree Fixes Nodes Dynamics</i>)
SISO	Única Entrada - Única Salida (del inglés <i>Single Input Single Output</i>)
MISO	Múltiple Entrada - Única Salida (del inglés <i>Multiple Input Single Output</i>)
MIMO	Múltiple Entrada - Múltiple Salida (del inglés <i>Multiple Input Multiple Output</i>)
RGA	Matriz de Ganancias Relativas (del inglés <i>Relative Gain Array</i>)
rad/s	Radian/segundo
mm	Milímetro
km	Kilómetro
h	Hora
kg	Kilogramo

kW	Kilovatio
m	Metro
Ah	Amperio hora
PWM	Modulación por ancho de pulsos (del inglés <i>Pulse Width Modulation</i>)
RPM	Revolución Por Minuto
V	Voltio
s	Segundo
MOSFET	Transistor de efecto de campo metal-óxido-semiconductor (del inglés <i>Metal-Oxide Semiconductor Field-Effect Transistor</i>)
ISO	Organización Internacional de Normalización (del inglés <i>International Organization for Standardization</i>)
PC	Computador Personal (del inglés <i>Personal Computer</i>)
RAM	Memoria de Acceso Aleatorio (del inglés <i>Random Access Memory</i>)
GB	GigaByte
USB	Bus Universal en Serie (del inglés <i>Universal Serial Bus</i>)
HDMI	Interfaz Multimedia de Alta Definición (del inglés <i>High – Definition Multimedia Interface</i>)
mA	Miliamperio
ADC	Convertor Analógico-Digital (del inglés <i>Analog to Digital Converter</i>)
CC-CC	Corriente Continua - Corriente Continua
JTAG	Del inglés <i>Joint Test Action Group</i>)
STAIR	STanford Artificial Intelligence Robot
F.D.T.	Función De Transferencia
ARX	AutoRegresión con entrada eXógenas del inglés <i>AutoRegressive with eXogenous input</i>)
MOC	Mantenedor de Orden Cero
T_{rm}	Tiempo de residencia medio
Q_{rn}	Índice de controlabilidad

Índice de figuras

1.1. Modelo implementado en Simulink.	5
1.2. Flujo de recursos heterogéneos entre los diferentes elementos del distrito agroindustrial.	6
1.3. Vehículo UAL-eCARM.	7
1.4. Elementos involucrados en los sistemas estudiados.	8
1.5. Resultados del modelado de los distintos sistemas estudiados.	9
1.6. Validación de los controladores del sistema <i>Drive-by-Wire</i> en simulación.	11
3.1. Vehículo eCARM durante uno de los ensayos realizados.	26
3.2. Componentes encargados de la administración de corriente.	27
3.3. Elementos principales de control del vehículo.	28
3.4. Sensores instalados en el vehículo.	29
3.5. Motor XQ-4.3.	29
3.6. Sensores y actuadores acoplados a los motores Maxon.	30
3.7. Sensores instalados en el vehículo e-CARM.	32
3.8. Distribución de componentes y sistemas en el vehículo eCARM.	33
3.9. Captura del sistema implementado en el vehículo eCARM con el comando <code>rqt_graph</code>	35
3.10. Diagrama de funciones del joystick físico.	37
3.11. Captura del nodo <code>joystick_driving</code> con el comando <code>rqt_graph</code>	38
3.12. Captura del nodo <code>battery_charge</code> con el comando <code>rqt_graph</code>	39

3.13. Captura del nodo <code>ual_ecar_vehicle_controller</code> con el comando <code>rqt_graph</code>	40
3.14. Configuración del nodo para el sistema <i>Drive-by-Wire-by-Wire</i>	41
3.15. Captura del nodo <code>libredaq</code> y <code>ual_ecar_odometry</code> con el comando <code>rqt_graph</code>	42
3.16. Ejecución principal del proceso del sistema <i>Steer-by-Wire</i>	44
3.17. Lectura de parámetros del sistema <i>Steer-by-Wire</i>	45
3.18. Calibración de los codificadores acoplados al mecanismo de dirección.	46
3.19. Procesado de señal del sistema <i>Steer-by-Wire</i> en lazo abierto.	46
3.20. Protección en el extremo del mecanismo de dirección.	47
3.21. Lazo de control del sistema <i>Steer-by-Wire</i>	48
3.22. Procesado de la señal de control y envío de datos a <i>ROS</i>	49
3.23. Ejecución principal de los sistemas <i>Throttle-by-Wire</i> y <i>Brake-by-Wire</i>	50
3.24. Lectura de parámetros de los sistemas <i>Throttle-by-Wire</i> y <i>Brake-by-Wire</i>	51
3.25. Procesado de señal en lazo abierto de los sistemas <i>Throttle-by-Wire</i> y <i>Brake-by-Wire</i>	51
3.26. Controlador del sistema <i>Throttle-by-Wire</i>	52
3.27. Procesado de la señal de control del sistema <i>Throttle-by-Wire</i>	53
3.28. Controlador del sistema <i>Brake-by-Wire</i>	54
3.29. Procesado de la señal de control del sistema <i>Brake-by-Wire</i>	55
4.1. Mecanismo físico de la dirección.	58
4.2. Diagrama de bloques del sistema <i>Steer-by-Wire</i>	58
4.3. Determinación de la zona muerta del mecanismo.	60
4.4. Ensayos de giro con el mecanismo operando en vacío.	61
4.5. Error en función de los parámetros para un modelo de primer orden.	62
4.6. Error en función de los parámetros para un modelo de segundo orden.	62
4.7. Ensayos de giro con el mecanismo operando sobre el suelo.	63
4.8. Error en función de los parámetros para un modelo de primer orden.	64
4.9. Error en función de los parámetros para un modelo de segundo orden.	65

4.10. Evaluación de errores de los modelos obtenidos.	66
4.11. Validación de los modelos obtenidos.	67
4.12. Error en la alimentación del actuador del sistema.	68
4.13. Ensayo de corriente con el mecanismo operando en vacío.	69
4.14. Error en función de los parámetros para un modelo de primer orden para vacío.	71
4.15. Error en función de los parámetros para un modelo de primer orden para suelo.	71
4.16. Evaluación de errores de los modelos obtenidos.	72
4.17. Validación modelo de corriente en función de la velocidad de giro del mecanismo.	73
4.18. Actuador físico sistema Brake-by-wire.	74
4.19. Diagrama de bloques del sistema <i>Brake-by-Wire</i>	74
4.20. Ensayo realizado sobre el sistema <i>Brake-by-Wire</i>	75
4.21. Validación de los modelos propuestos para el sistema <i>Brake-by-Wire</i>	78
4.22. Diagrama de bloques del sistema <i>Throttle-by-Wire</i>	78
4.23. Señal PWM para el rango de señales de entrada con distintos <i>throttle map</i>	79
4.24. Ensayo realizado sobre el sistema <i>Throttle-by-Wire</i>	80
4.25. Determinación del retardo en la tracción del vehículo.	81
4.26. Error cuadrático medio para los parámetros estudiados.	82
4.27. Validación del modelado de la dinámica del sistema <i>Throttle-by-Wire</i> . MSE.	83
4.28. Evaluación de errores de los modelos obtenidos.	83
4.29. Validación del modelado de la dinámica del sistema <i>Throttle-by-Wire</i> . Manual	84
4.30. Ensayo realizado <i>Throttle-by-Wire</i> usando las dos marchas y el freno motor.	85
4.31. Datos del vehículo durante un ensayo realizado.	86
4.32. Error en función de los parámetros para un modelo de primer orden del acoplamiento.	87
4.33. Validación de la dinámica del acoplamiento estudiado.	88
4.34. Diagrama de bloques completo del del vehículo UAL-eCARM.	89
4.35. Diagrama resumen de la dinámica completa del vehículo.	90

4.36. Modelo del vehículo UAL-eCARM implementado en Simulink.	91
5.1. Configuración PID Ideal.	94
5.2. Esquema de control propuesto para el lazo interno del sistema <i>Steer-by-Wire</i> . . .	96
5.3. Validación para el control propuesto en el lazo interno del sistema <i>Steer-by-Wire</i> . . .	97
5.4. Configuración PID Interactivo.	98
5.5. Validación de los controladores para la posición del sistema <i>Steer-by-Wire</i>	100
5.6. Validación de los controladores para la posición del sistema <i>Steer-by-Wire</i>	101
5.7. Esquema de control propuesto para el sistema <i>Brake-by-Wire</i>	102
5.8. Validación del controlador para la posición del sistema <i>Brake-by-Wire</i>	103
5.9. Esquema de control propuesto para el sistema <i>Throttle-by-Wire</i>	109
5.10. Validación de los controladores del sistema <i>Throttle-by-Wire</i>	111
5.11. Validación de los controladores del sistema <i>Throttle-by-Wire</i> con filtro en la referencia.	112
5.12. Esquema completo de control propuesto para el vehículo UAL-eCARM.	114
5.13. Validación de los controladores del sistema <i>Drive-by-Wire</i>	115
B.1. Calibración del radio de curvatura con el mecanismo de dirección (1/3).	123
B.2. Calibración del radio de curvatura con el mecanismo de dirección (2/3).	124
B.3. Calibración del radio de curvatura con el mecanismo de dirección (3/3).	125
B.4. Trayectoria estudiada.	125
B.5. Puntos donde se sincroniza la señal GPS con el registro del sistema <i>Steer-by-Wire</i>	126
B.6. Puntos coincidentes válidos para el cálculo.	126
B.7. Correlación entre el radio de curvatura y la posición del mecanismo de dirección.	126
B.8. Correlación entre el ángulo de Ackermann y la posición del mecanismo de dirección.	127

Índice de tablas

1.1. Controladores diseñados para el sistema <i>Throttle-by-Wire</i>	11
3.1. Características técnicas del vehículo <i>eCARM</i>	25
3.2. Listado de nodos de ROS en el vehículo <i>eCARM</i>	36
4.1. Parámetros característicos del modelado del sistema de dirección del vehículo. . .	65
4.2. Estudio de errores en los modelos propuestos para la velocidad de giro.	66
4.3. Parámetros característicos del sistema de corriente para un modelo de primer orden.	72
4.4. Estudio de errores en los modelos del consumo de corriente.	72
4.5. Estudio de errores en los modelos del consumo de corriente.	77
4.6. Estudio de errores en los modelos propuestos para la tracción del vehículo.	83
4.7. Ecuaciones que rigen la dinámica del sistema <i>Throttle-by-Wire</i>	84
5.1. Estudio del error de los controladores propuestos para el sistema <i>Steer-by-Wire</i> . .	101
5.2. Estudio del error de los controladores propuestos para el sistema <i>Brake-by-Wire</i> .	103
5.3. Parámetros de controladores por el método λ	106
5.4. Parámetros de controladores por el método de Ziegler-Nichols.	106
5.5. Parámetros de controladores por el método de Cohen-Coon.	107
5.6. Parámetros de controladores por el método de Chien, Hrones y Reswick.	108
5.7. Parámetros de controladores por el método AMIGO.	108
5.8. Parámetros de controladores por el método SIMC.	109

5.9. Estudio del error de los controladores propuestos para el sistema <i>Throttle-by-Wire</i> .	110
5.10. Ecuaciones que rigen la dinámica del sistema <i>Throttle-by-Wire</i>	113
A.1. Parámetros del controlador Curtis durante el modelado. I.	121
A.2. Parámetros del controlador Curtis durante el modelado. II.	122

**MODELADO Y CONTROL MULTIVARIABLE
DEL VEHÍCULO URBANO ELÉCTRICO
UAL-ECARM**

Capítulo 1

Introducción

1.1. Introducción e interés

La sociedad y las ciudades crecen, y sus infraestructuras y servicios deben hacerlo a la par, como es el caso del sector del transporte. La visión que se tiene del transporte en un futuro es que los vehículos se desplacen de forma autónoma y puedan comunicarse entre sí para tener más información del entorno. En la actualidad hay competiciones como “Grand Cooperative Driving Challenge 2016”, donde se pueden poner a prueba nuevas arquitecturas de control en vehículos eléctricos autónomos que operen de forma cooperativa [1] [2]. Uno de los grandes retos que debe afrontar toda nueva tecnología o innovación es la seguridad [3]. Para abordar esta preocupación en el caso de los vehículos, los desarrolladores ya están implementando servicios como Toyota Safety Connect para aumentar la confianza de los usuarios. Autores, como Howser y McMillin [4], ya han estudiado estos sistemas, con especial énfasis en la capacidad de decisión del conductor sobre los sistemas automatizados del automóvil en situaciones críticas.

Otra de las vertientes de innovación con gran repercusión es el desarrollo de interfaces entre la máquina y el usuario. No solo se busca que el usuario esté seguro, sino que también se sienta seguro y cómodo con la tecnología. Es cada vez más frecuente hablar de interfaces hápticas que permiten al usuario tener una retroalimentación sensorial, sonora o de fuerza en el volante, de lo que está ocurriendo alrededor a pesar de que la conexión entre donde ocurre el fenómeno y el usuario sea electrónica [5] [6]. Una tecnología relativamente novedosa en el desarrollo de vehículos es la denominada X-by-wire. Esta tecnología está basada en la sustitución de los mecanismos tradicionales de transmisión del movimiento por una red cableada de dispositivos electrónicos, actuadores y sensores [7]. De este modo, en un vehículo, el mecanismo tradicional de la dirección se sustituiría por un sistema Steer-by-wire, y el de frenado por un sistema Brake-by-wire, entre otros [8]. Estos sistemas surgen a partir de la necesidad de imponer mayores restricciones en términos de emisiones, experiencia en la conducción, seguridad, etc,

No obstante, no se debe caer en el error de pensar que todo son ventajas en estos sistemas. Todas las exigencias que se le plantean a estos sistemas provienen del control a alto nivel, no obstante, los actuadores poseen dinámicas que no se pueden compensar por muy bueno que sea el control. Estas dinámicas pueden incluso llegar a la provocación de fallas críticas en un vehículo si no se tratan como es debido en la etapa de control [9]. Por esta razón, para alcanzar un correcto funcionamiento de un sistema Drive-by-wire se hace necesario de la implementación de una jerarquía de control multicapa que integre todos los dispositivos de un vehículo [10]. Múltiples autores, con sus trabajos, han demostrado que a mayor nivel de control que se implemente y mayores exigencias, la complejidad del sistema aumenta al requerir la correcta integración de todos los dispositivos con sus respectivas limitaciones y desafíos para el control que suponen [11] [12]. Otro detalle de la tecnología by-wire es que permite un nuevo tipo de interfaz humano máquina, como es el caso de la sustitución del volante y los pedales tradicionales por un joystick [13], lo que permite mayor modularidad y facilidad para el control por computador en la adquisición de datos del sistema.

1.2. Objetivos

El trabajo que se propone desarrollar es la continuación del trabajo fin de grado realizado en el curso 2016-2017 titulado “Caracterización y control de sistema drive-by-wire en vehículo eléctrico” [14]. En él, se obtuvo un modelo preciso del sistema de dirección e implementó un control efectivo a nivel software en ROS. Esta implementación sacó a la luz nuevos retos y mejoras necesarias. Como por ejemplo, relativa a la seguridad, la necesidad de implementar el sistema de control a bajo nivel. Esto se propone para evitar que frente a un fallo entre la comunicación software en ROS y los actuadores, estos últimos funcionen de forma descontrolada, pudiendo llegar a causar graves daños al vehículo o el entorno. Así mismo, tal como se comenta en las conclusiones del trabajo mencionado, el sistema de velocidad longitudinal no se pudo modelar ni controlar por un fallo en el sistema de adquisición de datos de los sensores. Esto implica la necesidad de mejora del sistema de adquisición de datos, permitiendo obtener la realimentación necesaria para la implementación de un sistema de control adecuado.

En vista a lo expuesto anteriormente, el trabajo propuesto persigue la mejora del sistema hardware, software y firmware implantado en el vehículo autónomo desarrollado en la universidad de Almería. Los objetivos propuestos para este trabajo son:

- Implementación de nuevas tarjetas de adquisición de datos. Se requiere la sustitución de la tarjeta Arduino Pro Mini por dos Claraquinos, sección 3.2.7, para la correcta gestión de las arquitecturas de control propuestas. Se sustituirá la tarjeta de adquisición NI USB-6211 por la placa LibreDAQ para la lectura de los codificadores de las ruedas traseras, sección 3.2.10 y otras dos tarjetas específicamente diseñadas, sección 3.2.8, para la lectura de las señales analógicas de los sensores.

- Modificar la programación del vehículo. Se debe reconfigurar la estructura de nodos ROS para integrar los nuevos dispositivos instalados y transferir la arquitectura del nivel software al nivel firmware.
- Modelado de los principales sistemas del vehículo hacia una arquitectura completa X-by-Wire. Completar la estructura Drive-by-Wire (DbW) con los modelos Brake-by-Wire (BbW), sistema de frenado, y el Throttle-by-Wire (TbW), sistema de tracción del vehículo. Se obtendrá nuevamente el modelo Steer-by-Wire, dirección del vehículo, para hallar nuevas relaciones de dependencia buscando la obtención de un modelo multivariable lo más exacto posible.
- Todos los modelos obtenidos se integrarán en un modelo en Simulink, figura 1.1. Con esto se pretende obtener una herramienta para su aprovechamiento en futuros trabajos en el estudio de diferentes estructuras de control.
- Diseño de arquitectura de control que satisfaga unas especificaciones establecidas que aseguren una experiencia en la conducción cómoda y segura. Se perseguirá una estructura de control multivariable con los correspondientes desacoplos en aquellos sistemas que se observen inferencias entre sí.

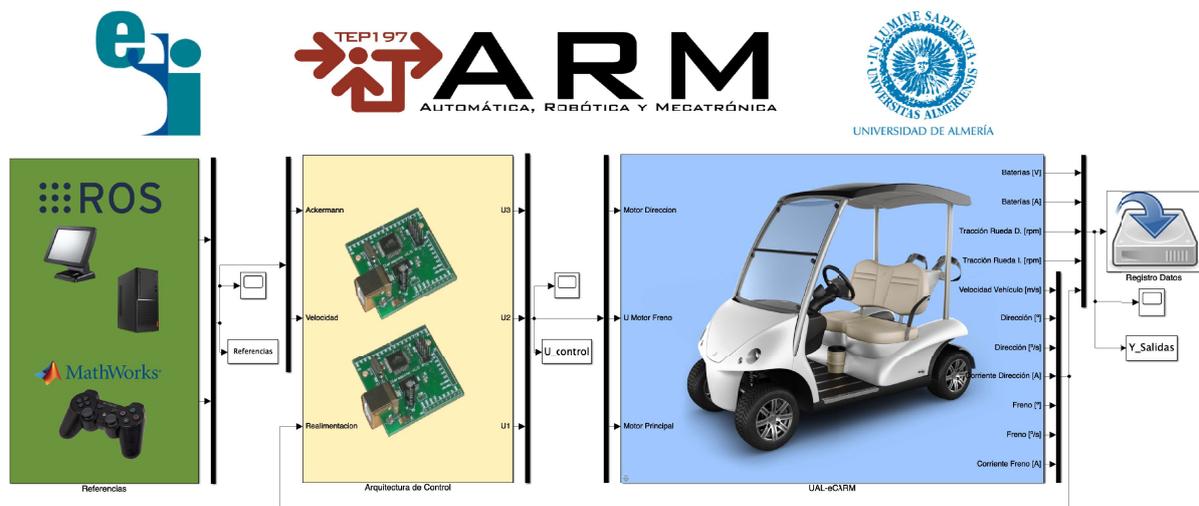


Figura 1.1. Modelo implementado en Simulink.

1.3. Contexto

Este trabajo es la continuación lógica del Trabajo Fin de Grado (TFG) en Ingeniería Electrónica Industrial "Caracterización y control de sistema drive-by-wire en vehículo eléctrico" [14] sobre el vehículo *UAL-eCARM*.

1.3. CONTEXTO

El vehículo se trata del modelo *LITA CLe2-2S* (Changzhou Greenland Vehicle Co. s.f.) [15], figura 1.3. Este vehículo ha sido objeto de múltiples investigaciones desarrolladas por el grupo de investigación de la Universidad de Almería (UAL) *TEP-197 Automática, Robótica y Mecatrónica* (ARM) en relación a la estructura mecánica del vehículo y desde la perspectiva energética [16] [17] [18]. Este trabajo se ha desarrollado bajo el marco del proyecto “Control y gestión óptima de recursos heterogéneos en distritos productivos agroindustriales integrando energías renovables” (CHROMAE, DPI2017-85007-R), coordinado por el grupo *ARM*. Este proyecto tiene como objetivo gestionar de manera eficiente el uso de diferentes recursos entre sistemas para minimizar el impacto medioambiental. En la figura 1.2 se muestra el flujo de recursos que considera CHROMAE. El vehículo se ubica dentro del proyecto bajo el paradigma *Vehicle-to-grid* (V2G) y *Grid-to-vehicle* (G2V) como consumidor y almacenamiento de energía eléctrica. Su función como agente del sistema consiste en el desempeño de tres funciones: consumir, almacenar y producir energía eléctrica.

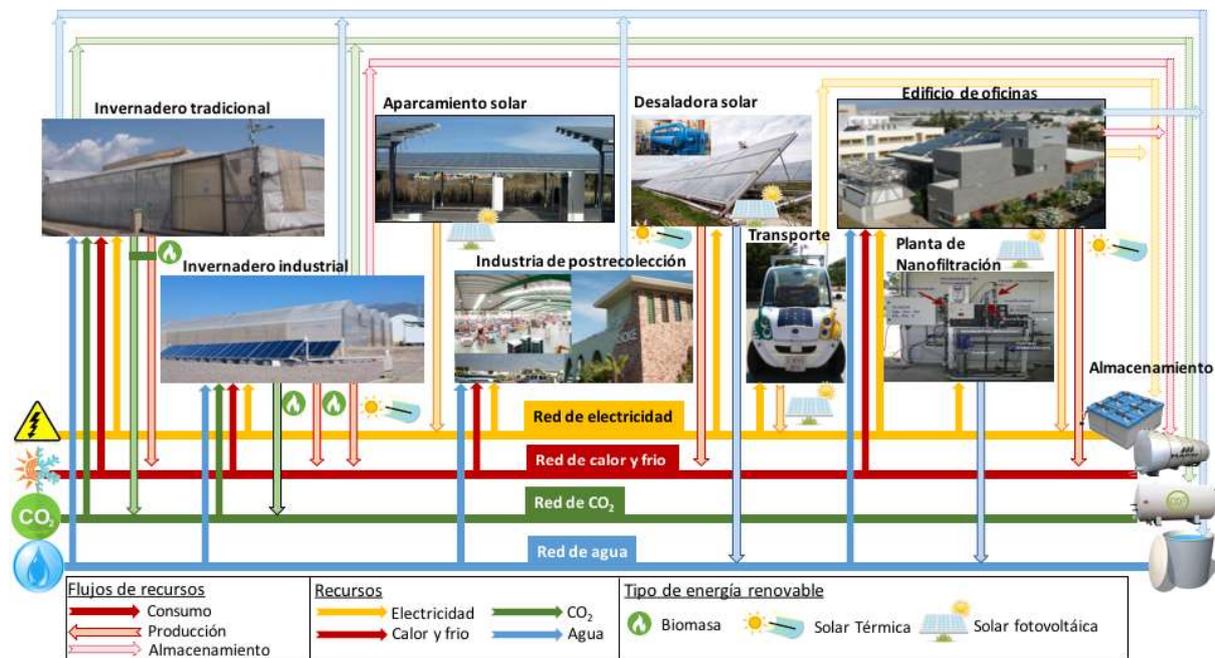


Figura 1.2. Flujo de recursos heterogéneos entre los diferentes elementos del distrito agroindustrial.

Con los avances alcanzados en [14], el vehículo pudo comenzar a moverse y utilizarse en trabajos de diversa índole, como trabajos con nubes de puntos, estudios energéticos, etc. No obstante, tal como se comenta en las conclusiones de dicho TFG, se requería la mejora del sistema de realimentación de los sensores que determinan la velocidad del vehículo, así como la mejora del sistema de adquisición de datos de los amperímetros y voltímetros para su integración en *ROS*.

Con la consecución de los objetivos planteados en este vehículo, se alcanzaría el estado propicio para comenzar a realizar trabajos de navegación y fusión con el resto de sensores como cámaras, láser, GPS o IMUs. Todo el equipo hardware instalado sobre el vehículo, así como la arquitectura software y firmware se amplían y detallan en el capítulo tres del presente trabajo.



Figura 1.3. Vehículo UAL-eCARM.

1.4. Resumen de resultados

Se ha realizado el modelado cinemático completo del vehículo durante su operación y diseñado un sistema de control sobre la posición del mecanismo de dirección y la velocidad del vehículo. Se ha planteado un sistema multivariable cuyas tres entradas principales se corresponde con la excitación del motor actuador del mecanismo de dirección, la excitación del motor actuador del mecanismo de freno y la señal analógica que simula un pedal de aceleración virtual. En caso de que el vehículo opere en modo manual, las señales de control se generan vía software desde un joystick o desde otro nodo, como puede ser uno de matlab o de la librería *Mobile Robot Programming Toolkit* (MRPT).

1.4. RESUMEN DE RESULTADOS

En el caso del funcionamiento automático, vía software se generan las referencias y es en la programación de los microcontroladores donde se aplica el algoritmo de control y se genera la señal de control pertinente. En la figura 1.4 se muestra de forma resumida el flujo de información y los componentes involucrados en los sistemas estudiados en el vehículo UAL-eCARM. Las señales de control U_1 , U_2 y U_3 se corresponden con la tensión que recibe el motor de la dirección, el freno y el controlador Curtis respectivamente. Las señales del sistema *Steer-by-Wire*, Y_{13} , Y_{12} y Y_{11} , se corresponden con el consumo de corriente, la velocidad de giro y la posición del motor acoplado al mecanismo respectivamente. En el sistema *Brake-by-Wire*, las señales Y_{23} , Y_{22} y Y_{21} se corresponden con el consumo de corriente, la velocidad de giro y la posición del motor que acciona el pedal de freno. Finalmente, en el caso del sistema *Throttle-by-Wire*, las señales Y_{31} , Y_{32} se corresponden con los codificadores acoplados a las ruedas traseras mientras que Y_{33} se corresponde con la velocidad del vehículo.

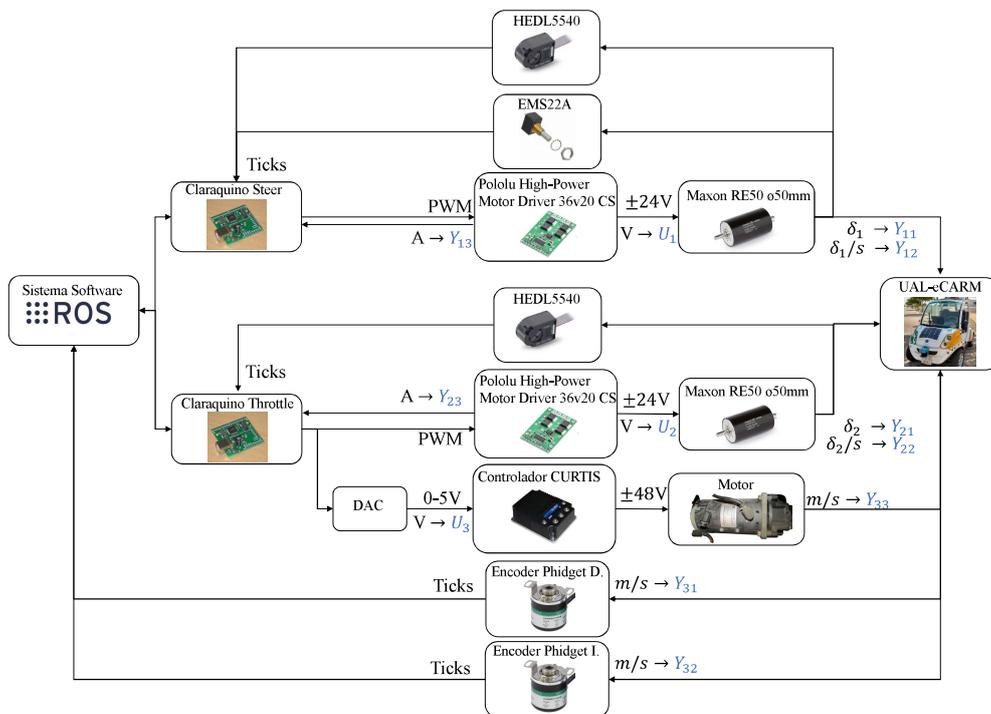


Figura 1.4. Elementos involucrados en los sistemas estudiados.

El modelado realizado, al igual que en el trabajo [14] se ha realizado en un análisis de “caja negra”, esto quiere decir que se ha modelado el sistema sin considerar los elementos internos, sino realizando un análisis de la correlación entre las entradas y salidas del sistema. Como técnica de identificación, se han evaluados distintos modelos posibles del sistema y se han determinados los parámetros de dichos modelos mediante la determinación del mínimo error cuadrático medio obtenido.

En la figura 1.5 se muestra de forma resumida todos los resultados obtenidos. Se han completado los objetivos planteados en cuanto al modelado. Se ha conseguido establecer el modelado completo para tener realmente modelado el sistema *Drive-by-Wire* del vehículo.

Respecto a la información obtenida en trabajos anteriores, se ha conseguido determinar la presencia de una zona muerta en el sistema de dirección, que anteriormente se había modelado como un retardo dominante. En el caso del sistema de tracción, se ha determinado que el modelo que mejor se ajusta a la dinámica del vehículo se trata de un sistema no lineal. Se ha podido modelar linealizando el sistema dividiéndolo en cuatro zonas de operación con dos regímenes distintos, aceleración y frenado. Se han determinado un total de 8 modelos con un retardo común de 0,91 [s]. Pese a que se han incluido en el sistema *Throttle-by-Wire* los modelos del régimen de frenado, la señal de control en estas zonas se corresponde con el efecto del freno sobre el sistema. Esto se debe a la dinámica de frenada libre propia del vehículo. En el caso del freno, se ha obtenido de su velocidad de giro que presenta una precisión aceptable en la mitad inferior del régimen de operación. Esto se debe a que es el sistema, constructivamente presenta una mayor no linealidad, por lo que para una mayor precisión se requiere un modelo no lineal.

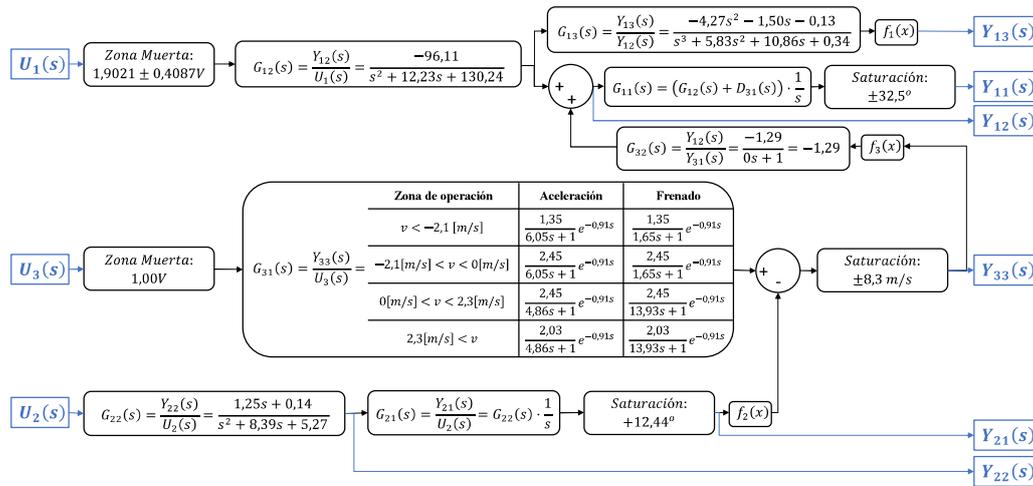


Figura 1.5. Resultados del modelado de los distintos sistemas estudiados.

A pesar de esto, se considera suficiente el modelo obtenido para establecer el sistema *Drive-by-Wire* del vehículo. Se ha podido determinar igualmente el acoplamiento entre el sistema de dirección y tracción del vehículo. Finalmente, se ha obtenido un modelo de consumo de corriente para el sistema de dirección para un mayor conocimiento del funcionamiento del vehículo y poder detectar problemas o límites del mecanismo por consumos anómalos excesivos. Todo el proceso y resultados detallados se encuentran recogidos en el capítulo 4.

En lo referente al control, se ha realizado un estudio detallado de diferentes controladores y métodos de sintonía. Todo el desarrollo de controladores y justificación de los controladores seleccionados se encuentra disponible en el capítulo 5.

Una estructura de control que se ha implementado en todos los casos es el mecanismo anti-windup para evitar la acumulación de errores en los términos integrales de los distintos controladores. En primer lugar, el sistema *Steer-by-Wire* dispone de dos lazos de control en cascada. El lazo interno se corresponde con el control de velocidad de giro del mecanismo. Este sistema ha requerido un control PID en su configuración ideal, ecuación 1.1, para simplificar la dinámica en bucle cerrado de un segundo orden a un primer orden. El controlador implementado cumple la especificación impuesta de una constante de tiempo de 0,2s. En el caso del control externo, se han evaluado un controlador PID interactivo con un filtro en la referencia, un controlador PD y un controlador PI, siendo el finalmente implementado el controlador PD, ecuación 1.2. La especificación establecida y alcanzada en este caso es un sistema sin sobreoscilaciones.

$$C_v(s) = K_{vp} \left(1 + \frac{1}{T_{vi}s} + T_{vd}s \right) = -0,6362 \left(1 + \frac{1}{0,0939s} + 0,0818s \right) [Vs\delta^{-1}] \quad (1.1)$$

$$C_p(s) = K_{pp} (1 + T_{pd}s) = 5 (1 + 0,2s) [Vs\delta^{-1}] \quad (1.2)$$

En el ámbito del control de velocidad del vehículo, el sistema *Throttle-by-Wire* será el encargado de controlar el régimen de aceleración y el sistema *Brake-by-Wire* el régimen de frenado. En el caso del sistema de frenado, se ha implementado un controlador del tipo PI para el control de posición del freno, ecuación 1.3. El ajuste se ha realizado de forma manual hasta alcanzar unas consignas temporales en bucle cerrado adecuadas, una constante de tiempo de 2,1s sin sobreoscilaciones. La referencia a este sistema se realiza cuando el error que recibe el sistema de tracción es negativo, es decir, cuando la velocidad del vehículo está por encima de la referencia y el “acelerador” es insuficiente para reducir la velocidad. En este caso solo se consigue reducir la constante de tiempo de respuesta del sistema durante la frenada.

En el control de tracción se han comparado los 7 métodos de sintonía más relevantes para sistemas de constante de tiempo dominante. Para aumentar el tiempo de control y evitar las zonas de frenado, se ha implementado un filtro en la referencia que elimina las sobreoscilaciones y por tanto mejora el desempeño y suavidad de la operación. La especificación temporal impuesta y satisfecha con el controlador seleccionado es una constante de tiempo en bucle cerrado de 3s. En la tabla 1.1 se muestran los controladores seleccionados finalmente mediante el método de sintonía de Ziegler-Nichols para cada zona de operación. En la figura 1.6 se muestran los resultados obtenidos para el sistema propuesto.

$$C_p(s) = K_{pp} \left(\frac{T_{pi}s + 1}{T_{pi}s} \right) = 3 \left(\frac{0,2s + 1}{0,2s} \right) [Vs\delta^{-1}] \quad (1.3)$$

Marcha	Zona Operación	Controlador	Unidades
Inversa	$v < -2,1 \text{ m/s}$	$\frac{4,432(3,03s + 1)}{3,03s}$	$[ms^{-1}V^{-1}]$
	$-2,1 < v < 0 \text{ m/s}$	$\frac{2,442(3,03s + 1)}{3,03s}$	$[ms^{-1}V^{-1}]$
Directa	$0 < v < 2,3 \text{ m/s}$	$\frac{1,962(3,03s + 1)}{3,03s}$	$[ms^{-1}V^{-1}]$
	$2,3 < v \text{ m/s}$	$\frac{2,367(3,03s + 1)}{3,03s}$	$[ms^{-1}V^{-1}]$

Tabla 1.1. Controladores diseñados para el sistema *Thorttle-by-Wire*.

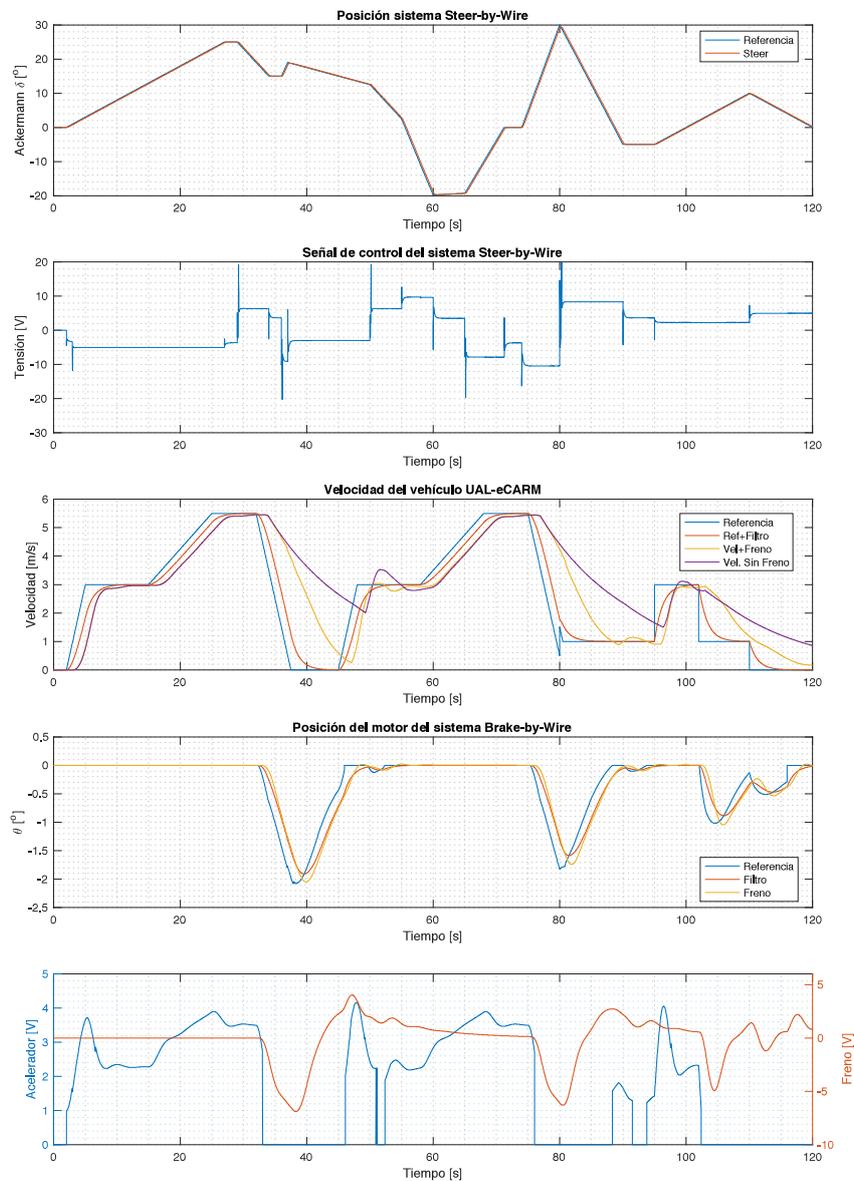


Figura 1.6. Validación de los controladores del sistema *Drive-by-Wire* en simulación.

1.5. Fases de desarrollo y planificación

En la elaboración de este trabajo se pueden distinguir las siguientes etapas de desarrollo, indicando también el número de horas empleadas en cada tarea sumando un total de 340 horas.

1. Estudio bibliográfico [50h].

En esta etapa se comprende la documentación en artículos científicos y publicaciones especializadas acerca del problema que se pretende abordar: Cómo se ha trabajado anteriormente, métodos actuales (sus ventajas e inconvenientes), diseño de distintas reglas de sintonía aplicables a la situación en cuestión, etc. Así mismo, se estudia la evolución en el desarrollo de la plataforma sobre la que se realiza el trabajo.

2. Puesta a punto y calibración de arquitectura hardware [90h].

Para la evolución de la plataforma de desarrollo, de forma previa al modelado de los sistemas abarcados, se debe realizar la instalación de nuevos sensores y desarrollo del nuevo sistema de adquisición de datos.

3. Caracterización, diseño y simulación [100h].

Se procede a determinar las características que rodean al vehículo en el que se va a implantar el sistema de control. Para esto se realizarán ensayos con el vehículo en distintas situaciones para obtener la información más completa posible, por ejemplo, ensayos en los que la dirección sigue un perfil determinado con las ruedas suspendidas y apoyadas. Algunas de las características analizadas son las constantes de tiempo de los distintos sistemas que se abordan. Con dichas características establecidas, se realizará un modelo de simulación completo del vehículo que permita determinar que método de sintonía ofrece un mejor comportamiento frente a distintos perfiles de operación.

4. Programación e implantación en el vehículo [60h].

Tras el diseño y elección del control óptimo, se implantará la parte correspondiente al software en el sistema operativo (ROS) que se ejecuta en la CPU del vehículo. La parte firmware se realizará en C++ en los microcontroladores Atmel que se encuentran instalados para asegurar su ejecución en tiempo real.

5. Redacción de la memoria final del trabajo fin de máster [40h].

En ella se ampliará la información de cada una de las fases citadas anteriormente. Se cerrará con las conclusiones sobre el estado en que queda el vehículo así como una serie de propuestas para futuros trabajos a partir de los resultados obtenidos en este.

1.6. Competencias empleadas en este proyecto

A continuación se listan las competencias que plantea la titulación y se han empleado a lo largo del desarrollo del TFM.

- CB6 - Poseer y comprender conocimientos.

Se han aumentado los conocimientos previos en el ámbito del control de procesos automatizados y la arquitectura software del sistema operativo ROS.

- CB7 - Aplicación de conocimientos.

Ha sido necesario la aplicación de los conocimientos adquiridos mencionados anteriormente para el correcto desarrollo del trabajo.

- CB8 - Capacidad de emitir juicios

Como se puede comprobar en el capítulo final de este trabajo, esta capacidad se ha adquirido al emitir un juicio sobre el estado final en el que queda la plataforma UAL-eCARM.

- CB9 - Capacidad de comunicar y aptitud social.

Esta capacidad se adquiere con la defensa del trabajo ante un tribunal, así como en las diversas visitas que han realizado grupos externos a la universidad para que se les enseñase el equipamiento del grupo ARM.

- CB10 - Habilidad para el aprendizaje.

Esta competencia queda patente en la adquisición de nuevos conocimientos además de los adquiridos en el máster, como el TI7, TI8 y G8.

- TI7 - Capacidad para diseñar sistemas electrónicos y de instrumentación industrial.

Esta competencia ha sido fundamental en el proceso de instalación de las nuevas tarjetas de adquisición de datos descritas en la sección 3.2.8.

- TI8 - Capacidad para diseñar y proyectar sistemas de producción automatizados y control avanzado de procesos.

La capacidad de diseño y proyección de sistemas de control avanzado es el eje fundamental del trabajo realizado, por lo que se podría considerar la competencia de mayor relevancia del proyecto.

- G8 - Capacidad para la gestión de la Investigación, Desarrollo e Innovación tecnológica.

Esta competencia queda patente en el capítulo 2, donde se realiza un estudio del contexto industrial donde se desarrolla el presente trabajo.

- TFM1 - Realización, presentación y defensa, una vez obtenidos todos los créditos del plan de estudios, de un ejercicio original realizado individualmente ante un tribunal universitario, consistente en un proyecto integral de Ingeniería Industrial de naturaleza profesional en el que se sinteticen las competencias adquiridas en las enseñanzas.

1.7. Estructura del TFM

El trabajo desarrollado se expone atendiendo a una estructura de seis capítulos. El capítulo uno, el cual concluye con esta sección, trata de posicionar el proyecto realizado e introducir brevemente el contenido ampliado en los posteriores capítulos. El capítulo dos, contextualiza el marco en el que se desarrolla el trabajo realizado en la literatura. En el capítulo tres, queda definido el estado actual del vehículo UAL-eCARM. El capítulo cuatro comprende el trabajo de modelado de los distintos sistemas que aborda el control que se pretende implementar. El capítulo cinco desarrolla el proceso de sintonía de los controladores para un esquema multivariable, así como la prueba de su desempeño en la plataforma real. Se cierra el proyecto con el capítulo seis donde se desarrollan las conclusiones a partir de los resultados finales obtenidos y se plantean posibles líneas para la continuación del trabajo desarrollado sobre la plataforma UAL-eCARM.

Capítulo 2

Revisión bibliográfica

2.1. Introducción

Actualmente, el Internet de las Cosas (IoT), está cambiando el paradigma de las comunicaciones en las ciudades. Es cada vez más frecuente encontrar la integración de los dispositivos tradicionales que nos rodean en redes que se encuentran conectados entre sí [19] y manden información a centros de procesamiento de datos que son capaces de extraer patrones de nuestros hábitos y ofrecer un servicio más acorde a las preferencias de los consumidores sin que estos se percaten. Aplicado a las ciudades, el concepto se desarrolla como *Urban IoT* y supone un reto por la complejidad de la conexión de la heterogeneidad de dispositivos, protocolos de comunicación, restricciones físicas, etc [20]. El ámbito de aplicación de estos sistemas abarca desde la automatización de los hogares e industrias, asistencia sanitaria, gestión inteligente de energía y tráfico entre otros [21]. La aplicación de este *Urban IoT* es la respuesta de las administraciones a la aplicación de las TIC a la gestión de los recursos públicos surgiendo así el concepto de *Smart Cities* [22], cuyo objetivo es aumentar la calidad de los servicios y la reducción del coste de los mismos. Como respuesta al incremento de volumen de información que deben manejar las Smart Cities, las estrategias de la generación distribuida y las micro-redes, se pueden considerar como una buena respuesta al dividir la red en volúmenes de información más manejables [23].

Dentro de las *Smart Cities*, a causa de su extensión y complejidad, existen gran cantidad de áreas de operación y se pueden clasificar según distintos criterios [24]. Considerando que según los datos del Instituto para la Diversificación y Ahorro de la Energía (IDAE), más del 40% del consumo energético, se produce en el sector del transporte, es lógico pensar que uno de las principales áreas a las que deben enfrentarse las *Smart Cities* es este sector. En este caso, se centrará la atención en los vehículos eléctricos por su semejanza a la plataforma sobre la que se desarrolla el presente trabajo. En [25] se realiza un estudio práctico centrado en la sinergia existente entre los recursos energéticos distribuidos (DER) de la ciudad de Madrid y los sistemas de transporte urbanos eléctricos. En él se observa el fuerte impacto de los vehículos eléctricos (EVs), el metro y los DER sobre el consumo y emisiones de una ciudad. El resultado obtenido es unos beneficios obtenidos con el modelo de control propuesto entre el 26.4% y el 34.2%.

Un área importante dentro de las micro-redes y de los recursos energéticos distribuidos es el almacenamiento energético [26]. Un planteamiento que entrelaza este área con los vehículos es la consideración de EVs e híbridos como plataformas de almacenamientos y transporte de energía además del enfoque tradicional como medios de transporte. Tal como se recoge en [27], el empleo de estos vehículos como propuesta para el almacenamiento de energía (ESSs) puede llegar a suponer una reducción del 70 % de la carga y el 15 % de los precios de la energía pico mediante el empleo de sistemas de carga inteligente. Esta integración de los EVs dentro de las redes energéticas de las ciudades han acuñado los términos V2G [28] y G2V para definir la interacción entre ambos sistemas. Tal como muestra [29], la relación entre los EVs y las redes energéticas supone grandes retos tecnológicos, como la actualización de la infraestructura actual, los protocolos de comunicación o el control en tiempo real de todos los puntos de recarga. No obstante, no es solo una cuestión técnica, sino también social, ya que para poder estudiar la viabilidad de estos sistemas se requiere una gran penetración de los EVs en el mercado.

2.2. El vehículo eléctrico

Como se ha comentado anteriormente, la adopción de la tecnología de los vehículos eléctricos es una de las principales vías de reducción de las emisiones de CO_2 [27] [30]. Así mismo, estudios como [31], van un paso más allá planteando los vehículos como un sensores dentro de las SmartGrids. Estos podrían recoger información del entorno y de los vehículos colindantes, mejorando así aspectos como la eficiencia en la conducción, la gestión del tráfico e incluso facilitando la integración de los vehículos autónomos a la circulación. Para esto, es requisito indispensable la mejora de las técnicas de control, sistemas embebidos en los vehículos y mejora de las infraestructuras de comunicación que intervengan en el proceso. Todo ello llevaría al vehículo eléctrico a integrarse aún mas en el IoT y alcanzar el siguiente nivel, denominado el Internet de los Vehículos (Internet of Vehicles, IoV). En este punto, los vehículos estarán conectados entre sí y con un servicio de almacenamiento de información denominado Vehicular Cloud.

Pero como se citaba en la sección anterior, para que los vehículos eléctricos tengan un impacto tangible en la sociedad, se debe producir una alta penetración en el mercado. Uno de los principales factores que limita este fenómeno es la reticencia del usuario respecto a la autonomía de las baterías. Los principales estudios sobre la autonomía buscan mejorar la eficiencia de las tecnologías empleadas actualmente. Para ello se buscan desarrollar modelos que permitan la estimación de consumos para optimizar el rendimiento de estas durante los trayectos. Chiara Fiori, Kyoungoo Ahn y Hesham A. Rakh desarrollan en [32] un modelo que comprende la aceleración, velocidad y frenada regenerativa del vehículo que permite estimar el consumo del vehículo de forma instantánea con un error medio del 5,9 %. Demuestran que la eficiencia es mayor en ciudad gracias al uso de la frenada regenerativa frente a trayectos en los que el vehículo se desplaza a gran velocidad.

No obstante, las baterías de los vehículos eléctricos no solo influyen en la autonomía, sino

en su interacción en las SmartGrids como unidades de almacenamiento de energía distribuida. En [33] se muestra un estudio detallado de la tecnología V2G, sus principales beneficios y sus mayores retos. Se trata de una tecnología relativamente joven y entre los principales retos que afronta es la degradación de las baterías y el alto coste de la inversión. Desde la perspectiva social existe la barrera de la desconfianza por parte de los usuarios de no disponer de energía cuando vayan a realizar sus desplazamientos al compartirla con la red. Y el problema no es solo el nivel de carga, sino el efecto del incremento de ciclos de carga y descarga que se produce. En [34] se observa un caso práctico en el que se estudia un perfil de consumo uniforme por parte de la red en E.E.U.U. y un consumo más agresivo en China para suplir picos de demandas. Se pudo concluir que la rentabilidad de las baterías depende mucho del perfil de consumo de la red donde se instalen, siendo más rentables aquellas que presentan un ciclo de vida más largo. Cuanto más uniforme sea la demanda de la V2G menos perjudicial será para las baterías. De todas las tecnologías que se emplean actualmente en las baterías, las de plomo-ácido son las más perjudicadas por el V2G y las más prometedoras las de ion-litio [35].

2.3. Los sistemas X-by-Wire

Los sistemas *X-by-Wire* están cada vez más extendidos en el mundo de la ingeniería [36]. Tradicionalmente, se ha asociado estos sistemas al reemplazo de un sistema mecánico tradicional, como puede ser el piñón-cremallera de la dirección de un vehículo, por un sistema electrónico. Este sistema capta a través de sensores o una interfaz con el usuario una referencia de movimiento y un ordenador central envía la correspondiente señal de control hasta un actuador final [7]. Cualquier sistema mecánico tradicional es susceptible de esta modificación tanto en coches, *Drive-by-Wire*, como en motocicletas, *Ride-by-Wire*, o en aviones, *Fly-by-Wire*, entre otros [37] [38]. Estos sistemas citados hacen referencia al sistema global del dispositivo y comprenden a su vez subsistemas que si están asociados directamente a los mecanismos físicos del sistema. En el caso particular de los vehículos, el sistema *Drive-by-Wire* se puede disgregar en el sistema de dirección, *Steer-by-Wire*, el sistema de frenado, *Brake-by-Wire* [39] y el sistema de aceleración, *Accelerate-by-Wire* o *Throttle-by-wire* [11].

Estos sistemas no se han instalado en la industria de forma caprichosa, sino como respuesta a las necesidades técnicas que se requieren cada vez más frecuentemente en los vehículos. Mediante la sustitución del mecanismo físico, la principal ventaja reside en la flexibilidad de diseño al eliminar la principal limitación de necesidad de espacio. Esto es una gran ventaja de diseño permitiendo tratar el sistema de forma modular, dotando al sistema de mayores grados de libertad para la distribución de componentes y elección de los mismos [40]. Esta libertad de diseño debe tratarse con cautela, puesto que como trata Y. Kim en [40], si se abusa de la modularidad puede convertirse el sistema en un laberinto para los diseñadores y suponer un mayor inconveniente que los sistemas tradicionales.

La tecnología *X-by-Wire* no se debe ser concebida solo como una herramienta para el diseño

mecánico [41]. Continuando con su aplicación en los vehículos eléctricos, estos sistemas se diseñan para optimizar el funcionamiento de cada componente individualmente, así como para la mejora de la seguridad durante su funcionamiento. Si se analiza el comportamiento de los conductores sobre un vehículo convencional, se puede apreciar que para un mismo recorrido, es muy probable encontrar que unos sean más eficientes que otros puesto que no todo el mundo tiene las mismas destrezas al volante. Mediante el empleo de esta tecnología, se puede mejorar el rendimiento del vehículo [42] [43]. Pero sería incorrecto pensar que estos sistemas están focalizados en la optimización. Intuitivamente, son especialmente útiles en el caso de los vehículos autónomos. Volviendo al caso de la dirección de un vehículo, para el sistema es más fácil actuar directamente con un motor sobre el mecanismo piñón-cremallera que tener que actuar sobre un extremo de la caña de la dirección.

Otro aspecto relevante de estos sistemas es su estrecha relación con el usuario que los maneja. Hasta que las infraestructuras permitan que todos los vehículos en circulación operen de forma autónoma, existirá una fuerte sinergia entre los humanos y la electrónica en el control de los vehículos. Las primeras interfaces que se desarrollaron para estos sistemas, y las más comunes de trabajar por su simplicidad, estaban basadas en joysticks y dispositivos móviles [44]. Una tecnología que recientemente se ha introducido son las denominadas interfaces hápticas [5]. Estas, a diferencia de sus predecesoras, ofrecen una realimentación sensorial a los usuarios que operan el dispositivo. A pesar de no estar conectados físicamente actuador y usuario, estas interfaces procuran una realimentación, por ejemplo de fuerza que se opone al movimiento, en el mando de control que emula el comportamiento de un mecanismo tradicional. Esto es importante para tener una mayor aceptación del sistema por parte de los usuarios. Esa realimentación provocan una sensación de mayor seguridad y control sobre el usuario que opera, no siendo solamente un extra añadido sin funcionalidad.

Finalmente, es importante destacar que con estos sistemas también se persigue incrementar la seguridad de los vehículos. Con la salida al mercado de los primeros vehículos autónomos, ya se plantea la cuestión “En un accidente inevitable ¿A quién mataría un coche autónomo?” [45]. La inteligencia del usuario es un factor crucial en la toma de decisiones en caso de accidente, aunque es importante tener presente que en ocasiones pueden producirse descuidos o negligencias. Cada vez, es más frecuente encontrar estos sistemas como asistentes del usuario en tareas como mantener el vehículo dentro del carril si se producen salidas indeseadas, avisos por si se produce un exceso de velocidad a partir de la lectura de señales viales o para aparcar el vehículo. Estas son tecnologías que ya se incorporan en gran cantidad de vehículos, al igual que asistentes para evitar colisiones frontales con obstáculos en la vía. En [46] se trata un caso menos tratado, pero igualmente importante, la colisión lateral con ciclistas. En este caso en particular, la modificación del sistema de control se realiza en el mecanismo *Steer-by-Wire*. Se trata de una prueba de que estos sistemas no solo buscan beneficiar a los usuarios que hacen uso de los vehículos, sino que también se diseñan para proteger a los que deben compartir entorno con ellos.

2.4. Modelado en EVs

Queda demostrada la versatilidad y potencia que ofrecen los sistemas *X-by-Wire* en el caso de los vehículos eléctricos. No obstante, hasta llegar a la operatividad adecuada de estos sistemas, se debe realizar un profundo trabajo en los distintos niveles de control que intervienen en el sistema. Todo proceso de diseño de una arquitectura de control debe tomar como base el sistema que se pretende controlar. Para ello lo primero es conocer la dinámica del sistema ya que cuanto mayor sea la exactitud del modelo, mejor será el desempeño de los controladores.

El primer aspecto que se debe tener en cuenta es el tipo sensores y actuadores de que dispone el sistema y su emplazamiento. Una de las definiciones de los sistemas *by-Wire* era que el sistema mecánico tradicional se reemplazaba por una conexión cableada, pero esto no siempre tiene porque ser así. En [47] se presenta un esquema mediante el cual la lectura de referencia y la señal de control se generan en dos microcontroladores diferentes, por lo que en este caso, no existiría ni siquiera esa conexión directa cableada. Por otro lado, los sensores no siempre están conectados o asociados directamente a un actuador o elemento terminal para realizar la realimentación [48]. En el caso de los motores que se emplean generalmente como actuadores en el sistema *Steer-by-Wire*, la realimentación se suele hacer mediante un codificador acoplado al mecanismo. No obstante, esta medida se puede realizar por odometría a partir de dos codificadores situados en las ruedas traseras. Otro sistema de realimentación que serviría tanto para la orientación como para la tracción del vehículo puede ser el empleo de GPS e IMU [11]. Esta posibilidad de realizar mediciones redundantes desde distintos sistemas del dispositivo presenta una ventaja más allá de la libertad en el diseño del sistema. Tal como se explica en [49], la posibilidad de sistemas de medidas independientes redundantes permiten estimar errores en los sensores. Esto es posible haciendo fusión de datos con técnicas como el filtro de Kalman y teniendo en cuenta la incertidumbre de las medidas. En [47] se realiza un estudio completo sobre un dispositivo real en el que se reemplaza el sistema de dirección de un mecanismo de cuatro barras por la actuación directa sobre las cuatro ruedas. Esto convierte el dispositivo en un vehículo omnidireccional, y demuestra que no existe una única solución para tratar estos sistemas.

El siguiente paso por dar en el modelado es la elección de un método de representación para la dinámica del sistema. Pese a todos representar la misma dinámica, cada uno permite obtener de una forma más directa distintas características del sistema. Las dos más extendidas es la representación en espacios de estados [50] [49] y mediante el uso de funciones de transferencia [51]. Mientras que la primera representación trabaja directamente con las ecuaciones diferenciales del sistema, las funciones de transferencia trabajan en el dominio de la frecuencia, por lo que o se obtienen directamente en este dominio, o se le debe aplicar la transformada de Laplace a las ecuaciones diferenciales del sistema. Otra decisión importante es el orden del sistema con el que se trabaja y si es lineal o no. En [51] se trabaja con un modelo lineal de función de transferencia de tercer orden con un cero porque es el modelo que mejor se ajusta.

No obstante, existen técnicas por las cuales mediante unas suposiciones concretas y bajo unas restricciones, permiten obtener un modelo de orden inferior sin perder demasiada exactitud en el proceso. En el caso de espacios de estado, las dimensiones de las matrices de trabajo están definidas por el número de variables y el orden de ecuaciones diferenciales de las que se disponga. Un ejemplo de sistema de dos dimensiones se encuentra en [49] mientras que en [50] se trabaja con tres dimensiones. Otro factor importante es la técnica empleada para la identificación de parámetros. Una de las técnicas más empleadas cuando no se conocen las ecuaciones que rigen el comportamiento del sistema es la denominada modelo de “caja negra”. Esta técnica se basa en dar señales de excitación diferentes a la entrada y ajustar a un modelo preestablecido la dinámica observada a la salida. Cuando se prevén dinámicas complejas, como el caso de la no linealidad de los sistemas *Brake-by-Wire*, se obtienen mejores resultados empleando técnicas más avanzadas como algoritmos genéticos o la denominada Group Method of Data Handling [52]. En el caso de que se conozcan las ecuaciones de la dinámica del sistema, la técnica más extendida es la del modelado basados en primeros principios [53] [51] [54] .

Finalmente, hay que mencionar que la obtención de modelos no sólo se realiza para la puesta en marcha de un sistema. Tal como se expone en [53], obtener un buen modelo puede servir como base para el estudio y comparativa de distintas arquitecturas de control, así como para labores de docencia.

2.5. Control en vehículos eléctricos y autónomos

Este trabajo se realiza sobre un vehículo eléctrico, no obstante todos los aspectos referidos al control en robótica móvil son extrapolables a estos. Al igual que ocurre con el modelado, al tratar los sistemas de control, no hay un único controlador válido. A continuación se expone múltiples clasificaciones de controladores que se pueden encontrar en la industria aplicados a vehículos eléctricos.

Una de las primeras clasificaciones es la referente a la ubicación del controlador en la pirámide de la automatización. En el mundo del *IoT*, uno de los principales ámbitos de control se realiza a alto nivel con la gestión almacenamiento y distribución de la energía entre los distintos elementos que conforman la red [55]. Esto podría situarse en el tercer nivel, el encargado de la supervisión y operación de un proceso en su conjunto. En [56] se muestra un caso en el que se aplica un control *Multivariable Generalized Predictive Controller* (MGPC) al almacenamiento de una microgrid. En los resultados mostrados, se compara el control de este sistema con un controlador PID, un controlador Borroso y un MGPC, siendo este último el que muestra un mejor desempeño y menor agresividad. Otro caso de control macroscópico ubicado en un nivel superior de control son los algoritmos para guiado de trenes de vehículos [57]. En [58] se muestra un caso práctico de un conjunto de vehículos equipados con sensores que proporcionan las distancias relativas entre ellos. Los controladores de bajo nivel propuestos garantizan una dinámica suave independientemente del comportamiento que pueda tener el vehículo líder.

Otro enfoque para la clasificación de controladores es atendiendo a la consideración que hacen del sistema estudiado en cuestión. En [59] se recogen algunas de las técnicas más empleadas en la literatura para abordar el problema completo de movimiento en robots móviles. Estos son los cuatro planteamientos con mayor aceptación según el modelo considerado:

- Modelo cinemático [60] [61] [62]. Estos modelos están enfocados en condiciones de operación de bajas velocidades y bajas aceleraciones, donde su desempeño es óptimo.
- Modelo dinámico [63]. Estos modelos se centran en el análisis del movimiento atendiendo a las fuerzas que intervienen en el sistema y la posición como integración del estado del modelo dinámico. Estos sistemas están muy extendidos en modelos altamente no lineales como el caso del sistema de freno *Brake-by-Wire* [7].
- Modelo cinemático y dinámico. Esta solución, aunque contempla el problema desde una visión más completa, no es la más empleada. En [64] se muestra un caso en el que para la determinación de posición y orientación del robot se emplea el modelo cinemático, empleando el modelo dinámico para el estudio de las fuerzas que intervienen en el proceso.
- Exploración de árbol de forma aleatoria, *Rapidly-exploring Random Tree* (RRT) [65]. Estos algoritmos son los más extendidos en la actualidad en aplicaciones de robótica. Inicialmente surgió como complemento a técnicas de planificación de trayectoria como *Probabilistic Road-Map* (PRM). Modificaciones de este algoritmo se encuentran en el caso del RRT* Fixes Nodes (RRT*FN) y RRT*FNDynamics (RRT*FND) [66].

Un método de clasificar los controladores es en función de su periodo de funcionamiento. Una forma de operar es de forma continuada en el tiempo, un enfoque en el que está basado el controlador PID convencional. En [53] se presenta un control de la velocidad de cruce de un vehículo mediante un controlador PI asíncrono, basado en eventos. No obstante, no todos los controladores de los que disponen los vehículos están relacionados con sistemas fundamentales para su operatividad. Como se ha comentado anteriormente, es cada vez más frecuente la implementación de interfaces hápticas para una realimentación perceptible por el usuario en los sistemas *by-Wire*. Los actuadores de estas realimentaciones, también requieren de la implementación de un sistema de control aunque su funcionamiento no sea tan crítico [6] [5].

Entrando en el control a bajo nivel de los vehículos, una de las estructuras más empleadas es el control en cascada [67] [60] [68]. Esto se debe a que los actuadores de los sistemas *by-Wire*, principalmente son motores eléctricos que requieren un control interno para la corriente que consume y un control externo para su velocidad de giro [69]. Pero dentro de esta arquitectura, no siempre se presenta una arquitectura con dos lazos de control. En [70] se presenta un control PI aplicado a un freno electromecánico que emplea tres niveles, el inferior para la corriente de consumo, el intermedio para la velocidad de giro del motor y el superior para la fuerza del actuador. Como es lógico, a mayor cantidad de niveles, mayor definición y control sobre el sistema, pero se debe tener presente el incremento de la complejidad de la estructura.

No siempre se puede conocer con exactitud el modelo que rige el sistema que se está estudiando. La estructura más extendida para compensar este fenómeno es el empleo de controladores autoajustables [71] [72]. Con esta implementación, los parámetros característicos de los controladores pasan a ser variables, adaptándose a las perturbaciones que reciben del entorno. Un ejemplo de aplicación de estos sistemas, aplicados a la circulación de vehículos, es la posibilidad de cambiar el controlador en función del tipo de superficie de rodadura, consiguiendo una dinámica en bucle cerrado homogénea con independencia de este agente externo. Una estructura muy relacionada con el control adaptativo, o autoajustable, es la identificación de parámetros [73] [74]. Esta técnica consiste en recalcular parámetros del modelo del sistema a partir de las mediciones en la salida del sistema. Esto ofrece un gran abanico de posibilidades a los diseñadores de lazos de control. De forma genérica en la industria, poder estimar los parámetros característicos de un sistema estudiando la señal de entrada y salida, es de gran utilidad en procesos que se encuentran funcionando en bucle cerrado y no pueden ser interrumpidos. Volviendo al caso de los robots móviles, estos sistemas están estrechamente relacionados con el control adaptativo ya que la modificación de los parámetros del modelo son los causantes directos de la modificación del valor de los parámetros del controlador. En el caso de que el modelo del sistema sea conocido con gran exactitud, la estimación de parámetros en línea ofrece la posibilidad de modelar el entorno del robot a partir de las perturbaciones que causa. De esta manera, se podría obtener el rozamiento entre el robot con las distintas superficies por las que se desplaza. En [74] se hace uso de un control autoajustable para un modelo no lineal y realiza la comparativa entre un controlador GPC y un PID. En los resultados expuestos, se comprueba que un controlador GPC autoajustable, mejora la estabilidad que se puede obtener con un controlador PID. Pavkovic et al [71], muestra un ejemplo de cómo combinar controladores PID con algoritmos de control adaptativo y estimación de parámetros en línea de forma experimental sobre un sistema Throttle-by-wire consiguiendo un control robusto frente a variaciones en los parámetros del sistema, ya sean externas producidas por el entorno, o internas a causa del desgaste del sistema.

Aunque en multitud de ocasiones se traten los distintos sistemas de un vehículo de forma independiente o monovariable (*Single Input Single Output*), SISO, esto no es un método del todo preciso. Gran parte de los sistemas están relacionados entre sí y dependen sus dinámicas del estado de los otros, recibiendo la denominación de sistemas MIMO (*Multiple Inputs Multiple Outputs*). En el caso de los vehículos, esto se observa en el sistema de dirección, el de frenado y el de tracción, pudiendo agruparse estos dos últimos en el sistema de velocidad longitudinal que se correspondería con un modelo MISO (*Multiple Inputs Single Output*). Por tanto, si se pretende realizar un desarrollo riguroso del funcionamiento de un sistema que comprende a su vez diferentes subsistemas interrelacionados, la estructura más adecuada a implementar consiste en un control multivariable [56] [75]. Este planteamiento aumenta la complejidad del modelo del sistema y su control, ya que se deben obtener los modelos del acoplo entre sistemas y diseñar los correspondientes desacoplos para anular dichas interconexiones. Para la determinación de los emparejamientos, la técnica más empleada es mediante el análisis de la Matriz de Ganancias Relativas (*RGA*).

El último aspecto a tratar del control en vehículos, tras recorrer la jerarquía de control desde los niveles superiores y exponer las principales estructuras de control que se usan, es el control a bajo nivel de los distintos subsistemas. La estrategia de control más extendida en la industria son los controladores PID (*Proporcional-Integrador-Derivativo*) gracias a sus prestaciones y gran potencia [53] [67] [70] [76]. No obstante, no son sistemas infalibles y perfectos. Considerando la estructura PID como base, otros autores como Villagrà en [77] proponen el desarrollo de PID inteligente, i-PID, a un sistema de velocidad de crucero en entornos urbanos para compensar las no linealidades a las que no puede hacerle frente un PID convencional. En [78], se puede observar el caso de un autobús que se desea convertir en autónomo e implementa un controlador del tipo PID. Fu, en [79], se expone un sistema completo que contiene gran cantidad de las arquitecturas expuestas en esta sección. Plantea un controlador validado en simulación que plantea un Control multivariable robusto y autoajutable mediante PI con desacoplos y feedforward. Los controladores, pese a ser los más empleados y válidos para casi todos los sistemas que requieren un controlador, es frecuente encontrar nuevos controladores basados en arquitecturas más recientes. El empleo de algoritmos como los genéticos o híbridos presentan mejores resultados que los métodos tradicionales en términos de error cuadrático medio para la identificación de sistemas y su posterior control [52]. Otra vertiente muy extendida es el empleo de controladores predictivos, tanto el *Generalized Predictive Controller* [80] [56], como el *Model Predictive Controller* [60]. En [81] se muestra un caso práctico de la implementación de un controlador MPC a una plataforma móvil similar al vehículo UAL-eCARM sobre el que se desarrolla el presente trabajo.

Capítulo 3

Vehículo eCARM

3.1. Introducción

Este proyecto se ha realizado como continuación de las investigaciones llevadas a cabo sobre el vehículo eléctrico que posee la Universidad de Almería (UAL), figura 3.1. Este vehículo fue adquirido en el año 2010 a la empresa Tesur, equipado inicialmente por dicha empresa con los componentes necesarios para el correcto funcionamiento de la propulsión y la dirección asistida del vehículo. Todo este equipamiento es denominado *Caja Tesur* y con el paso del tiempo y los distintos desarrollos llevados a cabo sobre el vehículo se ha sustituido por diferentes componentes. Las características técnicas del vehículo a nivel constructivo se pueden encontrar en la página web del fabricante [15] (*Changzhou Greenland Vehicle Co., Ltd.*) y se muestran en la tabla 3.1 para facilitar la labor.

Parámetro	Medida	Parámetro	Medida
Longitud	2680 mm	Velocidad máxima	45km/h
Anchura	1510 mm	Autonomía	90 km
Altura	1780 mm	Radio de giro mínimo	4.3 m
Distancia entre ejes	1830 mm	Peso	740 kg
Paso ruedas traseras	1285 mm	Peso sin baterías	460 kg
Paso ruedas delanteras	1260 mm	Peso máximo	950 kg
Pendiente máxima	20 %	Potencia máxima	4.3 kW

Tabla 3.1. Características técnicas del vehículo *eCARM*.

La fuente de energía que posee el vehículo son 8 baterías marca *Greensaver* modelo *SP210-6*. Estas baterías suministran una tensión de 6 Voltios (V) y poseen una capacidad de 210 Amperio-hora (Ah). Con estas características, su conexión se produce en serie obteniendo así una tensión total para distribuir en el vehículo de 48V. A pesar de que la tensión suministrada es de 48V, se considera necesario el empleo de convertidores DC-DC que suministren líneas de tensión a 5V, 12V y 24V, puesto que no todos los dispositivos electrónicos que incorpora el vehículo trabajan a igual tensión.



Figura 3.1. Vehículo eCARM durante uno de los ensayos realizados.

3.2. Arquitectura hardware

La estructura hardware del vehículo presenta un nivel de complejidad elevado pues debe de poseer la versatilidad para suplir la necesidad de información de las diversas investigaciones llevadas a cabo sobre él. A continuación se desarrollan los principales componentes involucrados en este proyecto. Cabe destacar que el vehículo posee más elementos de gran interés pero que se han visto excluidos por la carencia de relevancia en el desarrollo de este trabajo o no encontrarse instalados/operativos durante el desarrollo del mismo.

3.2.1. Distribución de energía

Las baterías, como ya se comentó en la introducción de este capítulo, suministran una tensión de 48V, pero no todos los dispositivos requieren dicha tensión. Para el establecimiento de las distintas tensiones requeridas se han empleado dos fuentes CC-CC las cuales convierten de 48V a 12V y 24V respectivamente y dos reguladores conmutados de 5V y 19V respectivamente. En primer lugar, se dispone de un componente denominado *PowerBox*, figura 3.2c, cuyas funciones principales son la de protección de la rama electrónica del vehículo mediante un fusible de 20A y la de poder encender los componentes de forma individual. Para la línea de 24V se han empleado dos convertidores CC-CC SD-500L-24 mean well, figura 3.2a. Una fuente se emplea para la alimentación del motor de la dirección y otra para el freno, los amperímetros de la marca LEM situados en el motor, el láser SICK y el regulador de 19V.

La fuente de 12V, reutilizada de otros proyectos realizados anteriormente, suministra corriente a los distintos sensores y dispositivos que emplean dicha tensión, como el GPS. El regulador de 5V, localizado en *PowerBox*, se emplea para la alimentación de elementos como el hub USB o el switch ethernet. El regulador de 19V se requiere para el funcionamiento del monitor situado en la cabina del vehículo. Para su correcto funcionamiento, puesto que el dispositivo integrado abarca un rango de tensiones de salida de 11.85V-22V, se construye un circuito físico para adecuar la tensión a 19V, figura 3.2b.

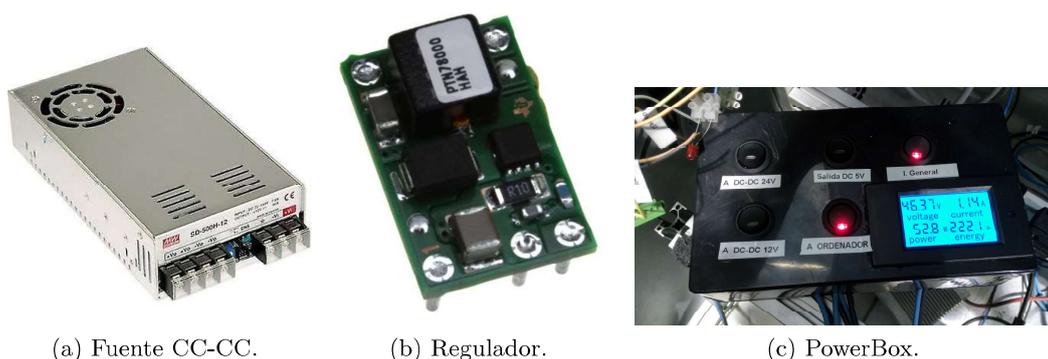


Figura 3.2. Componentes encargados de la administración de corriente.

3.2.2. Controlador Curtis

Este componente se trata de un controlador Curtis PMC [82] modelo 1268-5403, figura 3.3a con alimentación en el rango comprendido entre 36-48V. Su función principal consiste en realizar la etapa de potencia entre la parte electrónica de control y el motor de tracción del vehículo.

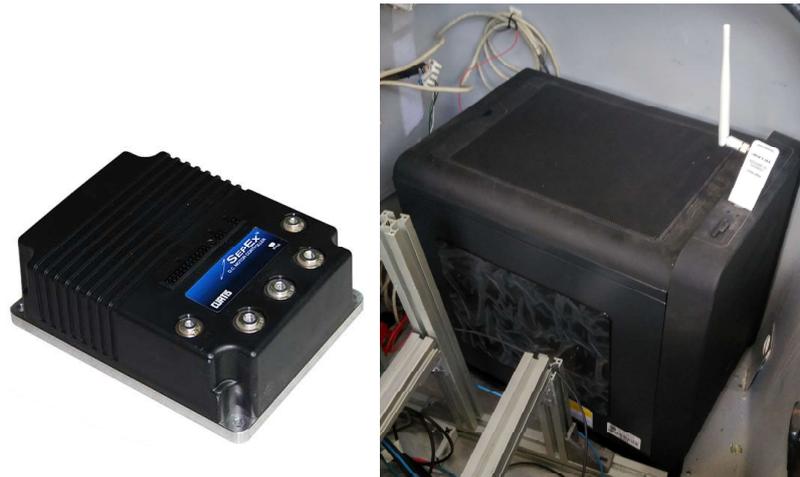
Se trata de un controlador programable, mediante software específico facilitado por el fabricante, basado en un microcontrolador que incluye una sección de potencia MOSFET para un mejor control sobre motores de excitación independiente. Abarca la lectura y actuación sobre gran cantidad de variables lo que permite una variada detección de fallos y un control de gran calidad sobre la velocidad de giro del motor. Su fabricación se lleva a cabo bajo un sistema de gestión de la calidad con la certificación correspondiente según las normas ISO 9001, además de un rango de protección ambiental IP64/IP67. Tiene una capacidad de hasta 400 amperios/2 minutos para la corriente nominal de la armadura y 50 amperios/2 minutos para la corriente nominal del campo.

3.2.3. PC Industrial

Se trata de un PC industrial diseñado especialmente para operar correctamente en ambientes hostiles, figura 3.3b. Esta condición debe estar presente pues su emplazamiento se encuentra en el maletero del vehículo y cuando se esté trabajando pueden alcanzarse elevadas temperaturas.

3.2. ARQUITECTURA HARDWARE

Al PC se encuentran conectados todas las placas de adquisición de datos y las claraquinos que soportan el esquema de control. Utiliza un sistema operativo Ubuntu Mate 16.04.2 sobre el cual se ha instalado el software *ROS*, el cual se describe en la sección 3.3. Dispone de una memoria instalada de 512Gb, una memoria RAM de 8Gb y un procesador i5 de 3.4GHz y 8 núcleos.



(a) Controlador Curtis 1268-5403 [83].

(b) PC Industrial.

Figura 3.3. Elementos principales de control del vehículo.

También se encuentran instalados otros dispositivos, que no están relacionados de forma directa con este trabajo. El sensor láser SICK Lidar LMS-200, figura 3.4a) situado en la parte frontal del vehículo permite la detección de obstáculos en un radio de alcance de 81m y 180° de visión. El sensor láser Velodyne Lidar Puck, figura 3.4b) posee un ángulo de visión de 360° y un alcance de 100m. Este dispositivo consta de 16 rayos y permite la construcción de entornos. También dispone de una cámara estereoscópica y en color del modelo Bumblebee2, figura 3.4c. Esta cámara se encuentra situada en la parte superior del vehículo y está conectada al PC Industrial mediante una conexión firewire.

Para estudios que requieren determinar la posición del vehículo se dispone de un sistema de posición global (GPS) y una Unidad de Medida Inercial (IMU). El GPS, figura 3.4e, se trata del modelo Hemisphere Crescent R100 Series Receiver. La antena receptora se sitúa en la parte superior del vehículo y el receptor en el maletero, conectado al PC a través de un puerto usb. No obstante, este sistema es insuficiente para incluirlo en los lazos de control propuestos en este trabajo por su baja tasa de muestreo, pero es un buen indicador de la posición del vehículo respecto al sistema de referencia geográfico. Finalmente, el IMU antes mencionado, se trata del modelo Xsens MTI 300, figura 3.4d. Este dispositivo posee una frecuencia de muestreo de 10kHz y se conecta al PC mediante USB.



Figura 3.4. Sensores instalados en el vehículo.

3.2.4. Motor principal

El actuador principal del sistema de tracción del vehículo es un motor de impulsión XQ-4.3, figura 3.5. Se trata de un motor de excitación en paralelo situado en la parte delantera del vehículo, con clase de aislamiento H y potencia 4.3kW. Se encuentra alimentado a los 48V de las baterías a través del controlador Curtis, el cual controla la excitación, el sentido de la marcha y la velocidad de respuesta del controlador.



Figura 3.5. Motor XQ-4.3.

3.2.5. Motor dirección y freno

Los motores instalados encargados del accionamiento de los mecanismos de la dirección y freno del vehículo, son dos motores de corriente continua, CC, Maxon RE 50 d50mm, escobillas de grafito, 200 Vatios. Este motor tiene una alimentación de 24V y su control físico está regido por la placa Pololu (*Pololu High-Power Motor Driver 36v20 CS*), figura 3.6c. Esta placa hace de etapa de potencia entre las señales generadas por la placa Claraquino encargada del sistema correspondiente, transformando la señal PWM y sentido de giro, y en una tensión proporcional a su alimentación. Esta placa incluye, entre otras posibilidades, la posibilidad de medir la corriente consumida por el motor. Además, ambos motores tienen instalados un reductor planetario GP 62 A d62mm, reductora 100:1 y un codificador en cuadratura incremental HEDL5540, 500ppv. 3 canales, con line driver RS 422, figura 3.6a.

3.2.6. Codificador absoluto

Pese a la buena calidad de señal que suministra el codificador incremental acoplado al motor del mecanismo de la dirección, la incertidumbre de la posición absoluta de la dirección supone un problema si el sistema no comienza en el cero del sistema. Como solución a este problema se lleva a cabo el acoplamiento de un codificador absoluto en el tramo en el cual se produce la unión del motor con el mecanismo de la dirección. El acoplamiento se lleva a cabo mediante un mecanismo de transmisión flexible de banda dentada atendiendo a la sección transversal de la banda. La relación entre el engranaje acoplado al codificador y el acoplado al mecanismo es de 1/3. El codificador empleado se trata del modelo EMS22A del fabricante Bourns (ver figura 3.6b). Dicho codificador posee una resolución de 10 bits, una tensión de alimentación de 5V o 3.3V $\pm 10\%$ y una corriente máxima de 20mA. En el caso del motor acoplado al mecanismo de frenada, por cuestión de espacio, esta opción para determinar la posición no es viable. Por lo tanto, considerando que el consumo de corriente aumenta conforme se acerca a una posición extrema del mecanismo, se hará que inicialmente el mecanismo se accione hasta detectar un límite de corriente marcado. Este punto se modelará y la posición será conocida, quedando así establecida la referencia del mecanismo para su respectivo control.



Figura 3.6. Sensores y actuadores acoplados a los motores Maxon.

3.2.7. Placa microcontrolador Claraquino

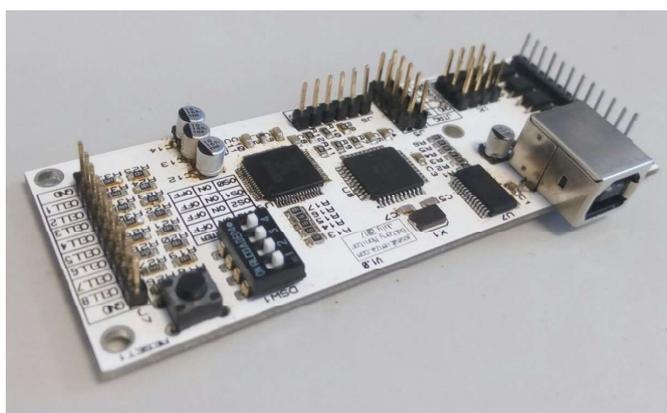
Se trata de una placa hardware de código abierto para prototipos con un microcontrolador basado en el ATmega328, depuración mediante JTAG y librería en C para conexión con periféricos comunes y sensores externos (Figura 3.7a). La programación y diseño de este dispositivo se encuentra disponible en un repositorio de Github¹.

3.2.8. Monitor de baterías

Se trata de una placa hardware de código abierto con un microcontrolador ATmega164, depuración JTAG y convertor analógico-digital AD7606 (Figura 3.7b). Su funcionalidad principal es la lectura de los valores de tensión que generan los amperímetros, así como la lectura de la tensión de las baterías. Su programación y el nodo de ROS encargado de su lectura se encuentran disponibles en Github, dentro del repositorio *ual-ecar-ros-pkg*².



(a) Claraquino v1.0.



(b) Monitor de baterías

3.2.9. Relé.

Se trata de un módulo ya hecho que incluye todos los componentes para que, frente a una señal digital, conmute la salida de un relé. En este trabajo se emplea un módulo que incluye dos relés con una alimentación de 5V. A la salida opera a 48V ya que su función principal es la de conmutar entre la marcha directa y reversa del vehículo. Dicha conmutación no es directa sobre el motor sino que se realiza a través de la unidad de control Curtis mediante las conexiones *J1-10* y *J1-11* del conector lógico *J1* [82].

¹Github: <https://github.com/jlblancoc/claraquino>

²Github: https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg/tree/master/battery_charge

3.2.10. Codificadores

Para la obtención de la odometría, el vehículo dispone de dos codificadores Phidget IHC3808, figura 3.7c, instalados en las ruedas traseras. Presentan el límite de escala en 4500 rpm, una resolución de 360 pulsos por revolución y deben ser alimentados en corriente continua a una tensión de 5V. Estos dispositivos se encuentran conectados a una tarjeta LibreDAQ, una plataforma modular de adquisición de datos desarrollada en la Universidad ³ [84]. Esta tarjeta es capaz de leer hasta un máximo de 10^6 pulso/s, por lo que es una plataforma de medida muy sobredimensionada en esta aplicación.

3.2.11. Amperímetros

El vehículo está dotado de tres amperímetros de la marca LEM, dos para la lectura de la corriente en el motor y un tercero para la lectura de corriente que suministran las baterías. Los del motor son del modelo LEM DHR 100, figura 3.7d, mientras que el de las baterías es del modelo LEM HO 240-P-0100, figura X2. El único inconveniente de este último dispositivo es que no detecta el sentido de la corriente. Actualmente si se desea determinar la corriente, su módulo si es medido directamente, mientras que el sentido se puede determinar a partir de variables indirectas como el estado de excitación del motor.

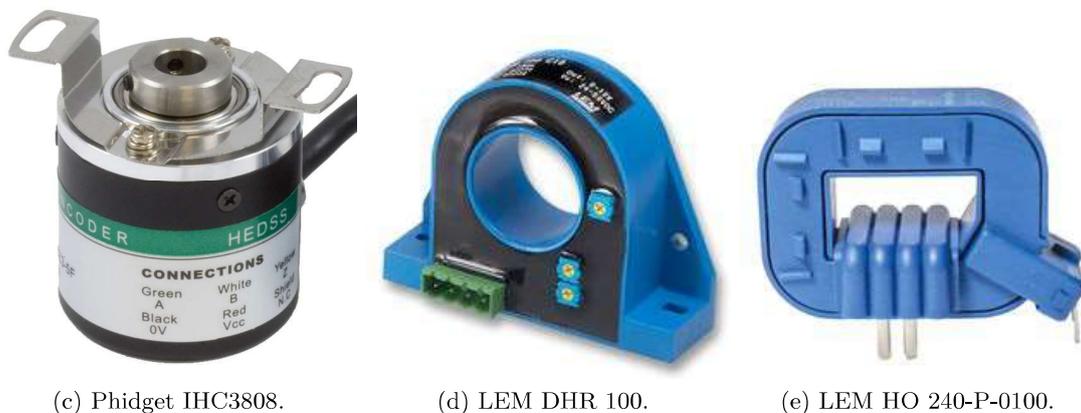


Figura 3.7. Sensores instalados en el vehículo e-CARM.

3.2.12. Esquemas de conexiones

Una vez descritos todos los componentes hardware que integran el vehículo, se muestra de forma gráfica el layout del vehículo en las figuras 3.8a y 3.8b. La figura 3.8a muestra la conexión y flujo de información entre los distintos componentes del vehículo. En la figura 3.8b se muestra la distribución de los principales dispositivos y sistemas involucrados en la navegación del vehículo.

³Github: <https://github.com/LibreDAQ/libredaq>

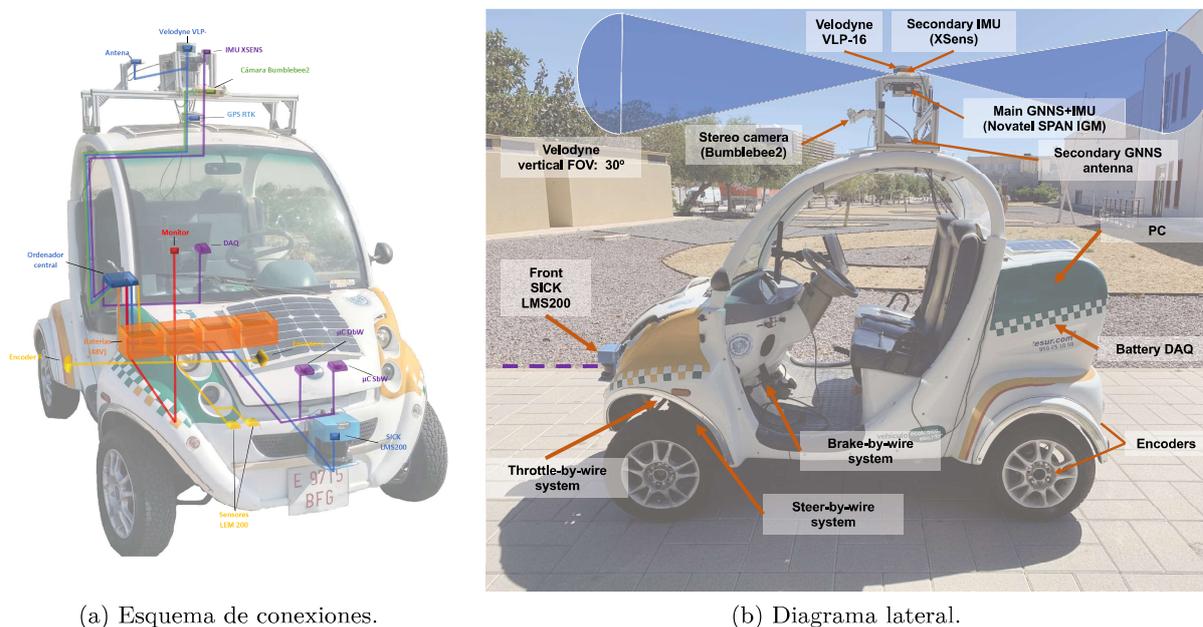


Figura 3.8. Distribución de componentes y sistemas en el vehículo eCARM.

3.3. Arquitectura software

La arquitectura software comprende la extensión de la estructura lógica que interconexiona los diferentes componentes hardware del vehículo para el correcto funcionamiento global del sistema. En el caso del vehículo eCARM, dicha arquitectura está basada en módulos autónomos realizados en *ROS*, los cuales emplean el lenguaje de programación *C++* como se ha indicado anteriormente. Los módulos reciben el nombre de “*Nodo*” en el argot de *ROS*. Estos módulos presentan una comunicación realizada a través de un modelo Publicación-Suscripción. Dicho modelo de comunicación presenta la ventaja de que cada nodo, para leer o transmitir información, no requiere establecer una conexión directa con los nodos de los que precisen o demanden la información. La comunicación se simplifica a que el nodo en lugar de mandar un dato, lo publica en un “lugar” software al que todos los nodos tienen acceso simultáneamente.

3.3.1. Robotics Operating System

ROS es un sistema operativo orientado al diseño de software para robots con *Inteligencia Artificial*, *AI*. Nació en el año 2007 en la Universidad de Stanford como una herramienta que permitiera la colaboración abierta de distintos grupos de investigación de robótica empleando licencia de código abierto. El proyecto para el cual se pensó este sistema fue el *STAIR*, *STanford Artificial Intelligence Robot*. Gracias a la posterior colaboración de *Willow Garage*, este software ha tenido una gran extensión de uso y colaboración a nivel internacional [85].

Desde sus comienzos, ROS está concebido con una estructura modular compuesta por unidades software denominadas “*packages*”. Al ser un proyecto desarrollado por múltiples instituciones e investigadores, el código fuente de los *packages* desarrollados por cada uno de ellos no se encuentran disponibles en un único servidor sino que por su estructura “federada”, cada desarrollador realiza su repositorio en servidores privados y con ello mantienen el control y propiedad completa sobre los mismos. Si autorizan para que estos sean públicos, el mayor beneficio que adquieren, además del reconocimiento y crédito pertinente, es la realimentación del resto de usuarios. Esta es la esencia entorno a la que gira el concepto de proyectos de software de código abierto. En la versión actual del vehículo eCARM, además de librerías y repositorios fundamentales como *MRPT*, se encuentra instalado el *package ual-ecar-ros-pkg*⁴, desarrollado por el equipo investigador involucrado en el vehículo, se encuentran disponibles en un repositorio de libre acceso de la plataforma *Github*, la cual promueve la divulgación de código abierto.

En el siguiente epígrafe, se detallan los nodos de ROS involucrados en el vehículo, su organización, funciones y objetivos, etc. Antes de comenzar la explicación hay que destacar el significado de ciertos términos concernientes a *ROS* que se emplearán de forma común a continuación.

En primer lugar, *topic* [86] es la nomenclatura empleada para hacer referencia a aquellas variables que forman parte de la comunicación entre nodos de *ROS*. El comando *rqt_graph* [87] ejecuta un nodo el cual incluye una herramienta que permite visualizar de forma gráfica la comunicación de *topics* entre los distintos nodos operativos. En las versiones más recientes de *ROS* permite además realizar análisis de la comunicación como el tiempo de actualización de cada dato o el tiempo de duración de la suscripción de un *topic* por parte de un nodo. Otro *package* fundamental para el registro de datos en los ensayos realizados es *rosbag* [88]. Este *package* incluye una serie de comandos que permite grabar, reproducir y registrar en distintos formatos los *topics* activos en el sistema. Mediante la ejecución del comando *rosbag record* se genera un fichero *.bag* que recoge aquellos *topics* que se incluyan como argumentos en la ejecución de dicho comando. Posteriormente, para traducir los archivos *.bag* a un lenguaje “amigable” para procesar los datos con Matlab, el comando *rosbag rosbag* permite reproducir el tiempo grabado y transferir los *topics* de uno en uno a un archivo *.txt* con cada reproducción.

3.3.2. Nodos de *ROS* para el control de dirección y velocidad del vehículo

Durante la operación normal del vehículo, el sistema precisa la ejecución de seis nodos. En la tabla 3.2 se indican estos nodos así como una descripción breve de su funcionalidad en el sistema. En la figura 3.9 se muestra un mapa global de las conexiones que se realizan a nivel software durante el correcto funcionamiento del vehículo obtenido con la herramienta *rqt_graph*.

⁴Github: <https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg>



Figura 3.9. Captura del sistema implementado en el vehículo eCARM con el comando `rqt_graph`.

roscore

La ejecución del nodo `roscore` lanza todos aquellos nodos y programas que requiere un sistema basado en *ROS* para funcionar [89]. Gracias a este comando, para el usuario es transparente la ejecución de aquellos comandos que se debe realizar al comienzo de cada sesión.

3.3. ARQUITECTURA SOFTWARE

Nodo	Descripción
/roscore	Nodo de ROS que se debe lanzar al iniciar el sistema.
/joystick_driving	Sirve de interfaz entre la estructura software del sistema y el joystick.
/amm/measure_current_sensors	Se emplea para la lectura de las distintas señales analógicas de sensores que se encuentran distribuidos por el vehículo.
/bat/measure_battery_cells	Realiza la lectura de la tensión en la conexión entre baterías. Realizando la resta entre las señales se puede medir la tensión de cada batería.
/ual_ecar_vehicle_controller	Establece la comunicación entre el nodo del joystick y los microcontroladores que rigen los sistemas <i>by-Wire</i> .
/libredaq	Es el nodo encargado de realizar la lectura de los codificadores instalados en las ruedas traseras.
/ual_ecar_odometry	El estado de los codificadores publicados por el nodo /libredaq permite determinar la odometría del vehículo respecto a su posición en el momento de ejecución del nodo.

Tabla 3.2. Listado de nodos de ROS en el vehículo *eCARM*.

joystick_driving

Este nodo se encuentra ubicado dentro del *package ual-ecar-ros-pkg* disponible en un repositorio en la plataforma *Github*⁵. Su función principal es ser la interfaz entre el usuario y el sistema ejecutado en ROS. En la figura 3.10 se muestran las funcionalidades de los distintos botones que puede accionar el usuario. El joystick izquierdo está asociado a la velocidad longitudinal del coche. Su desplazamiento vertical hacia la zona superior equivale a un incremento en la señal del acelerador virtual del sistema. Su desplazamiento vertical hacia la zona inferior equivale a un valor negativo del acelerador, lo que se interpreta a nivel software como la activación de la marcha reversa del vehículo. Estos valores del eje se transmiten al microcontrolador encargado del control del sistema *Throttle-by-Wire* a través del *topic /joystick_eje_y*. De forma complementaria, el botón “L2” está asociado directamente a la habilitación del control de este sistema y se corresponde con el *topic /vehicle_openloop_mode_throttle*. Por defecto, el sistema siempre arranca con el lazo de control abierto y solo se cierra mientras este botón esté accionado.

En el caso de que se use el eje vertical combinado con el botón con el cuadrado dibujado, *topic /vehicle_brake_enable*, se accede al control en lazo abierto del motor que acciona el pedal del freno. En el caso del freno, el desplazamiento vertical hacia la zona inferior equivale al accionamiento del pedal de freno. El movimiento opuesto se consigue con el desplazamiento vertical hacia la zona superior. Estos valores se transmiten al microcontrolador encargado del control del sistema *Brake-by-Wire* a través del *topic /joystick_eje_z*.

⁵Github: https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg/tree/master/joystick_driving

El joystick derecho está asociado al mecanismo de la dirección del vehículo, es decir, al sistema *Steer-by-Wire*. Su movimiento se realiza de forma horizontal. Su progreso hacia la derecha equivale al giro en sentido dextrógiro que se realizaría en un volante convencional, con su respectiva desviación del mecanismo en el mismo sentido. En el caso del desplazamiento del joystick hacia la izquierda, el movimiento equivalente en un volante convencional sería un giro levógiro, con su respectiva desviación del mecanismo en el mismo sentido. Estos valores del eje se transmiten al microcontrolador encargado del control del sistema *Steer-by-Wire* a través del *topic /joystick_eje_x*. Al igual que ocurría con el joystick izquierdo, el botón “R2” está asociado directamente a la habilitación del control de este sistema y se corresponde con el *topic /vehicle_openloop_mode_steering*.

Este nodo se ha preparado para el momento en el que el vehículo funcione de forma autónoma. Para ese momento, se ha considerado pertinente el empleo del botón “home” del dispositivo para deshabilitar la publicación de *topics* por su parte, dando paso a la generación de señales por parte del ordenador del vehículo. También se ha realizado como medida de seguridad, de forma que el control por defecto sea manual y en caso de pasar a modo autónomo, el usuario debe estar en el vehículo manteniendo el botón pulsado. Con esta medida, se garantiza que siempre haya un usuario dentro del vehículo que pueda desconectar la alimentación de los motores en caso de un comportamiento errático.



Figura 3.10. Diagrama de funciones del joystick físico.

De forma interna, las variables de los botones “L1” y “R1” se encuentran asociadas a los ejes “y/z” y “x” respectivamente. El valor de los ejes por defecto de los ejes “z” y “x” varía en el rango $[0,0'5]$ y el eje “y” entre los valores $[0,0'8]$. Con el accionamiento de estos botones, el rango se reduce a la mitad, duplicando por tanto la resolución de la señal generada. Estas implementaciones permiten un mayor manejo de cada variable por parte del usuario. A continuación, en la figura 3.11, se muestra una representación de este nodo donde se muestran los *topics* generados durante la ejecución del nodo *rqt_graph*.

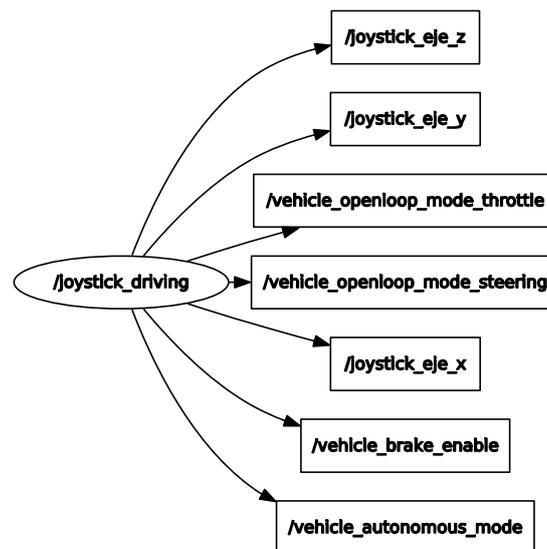


Figura 3.11. Captura del nodo `joystick_driving` con el comando `rqt_graph`.

battery_charge

Este nodo se encuentra disponible en el *package* `ual-ecar-ros-pkg`, disponible en un repositorio de *Github*⁶. Este nodo está reflejado en la tabla 3.2 como dos subnodos, `/amm/measure_current_sensors` y `/bat/measure_battery_cells`. Esto se debe a que se trata de un nodo diseñado para comunicar el monitor de baterías descrito en la sección 3.2.8 y el vehículo eCARM dispone de dos de estos dispositivos. Este nodo está configurado para que uno de los parámetros que debe de recibir al iniciarse sea el puerto con el que se debe comunicar. De esta forma se pueden instalar todas las tarjetas que se precisen sin tener que duplicar el código del nodo. Además de los archivos requeridos para el correcto funcionamiento de *ROS*, incluye la programación firmware de la tarjeta hardware. Otro parámetro importante que se debe introducir en su lanzamiento es el diezmado en la publicación de variables. Si este parámetro se establece con valor unitario, la frecuencia de publicación será la máxima. Esta implementación se ha considerado necesaria para evitar emplear tiempo innecesario en la lectura de variables en caso de estar estudiando un sistema independiente.

⁶Github: https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg/tree/master/battery_charge

Este nodo está configurado para la publicación de dos vectores de variables según el tipo de lectura que se haga de la variable. La tarjeta del monitor de baterías, está equipada con un microcontrolador *ATMEGA164PA* el cual dispone de la posibilidad de realizar tres mediciones analógicas, las cuales están vinculadas a los *topics* */ammeter_value*. Por otro lado, dispone de un conversor analógico digital *AD7606*, dispositivo que permite la lectura de 8 señales analógicas. Estas señales están vinculadas directamente a los *topics* */battery_voltaje*.

De forma complementaria, esta tarjeta está dotada con seis optoacopladores controlados a través del microcontrolador. Estos dispositivos permiten disponer de seis interruptores accionados de forma óptica que permite trabajar con una tensión en la parte del microcontrolador y en el otro lado una tensión más elevada el microcontrolador no puede soportar. El subnodo */amm/measure_current_sensors* se emplea para la lectura de las señales analógicas que generan los distintos sensores distribuidos por el vehículo, como el amperímetro situado en las baterías y los amperímetros y voltímetros situados en la parte delantera del vehículo para medir la corriente y tensión en el estator y rotor del motor principal del vehículo. El subnodo */bat/measure_battery_cells* está dedicado a la medición de la tensión en las uniones entre las 8 baterías de las que dispone el vehículo. Para obtener el valor de tensión que suministra cada batería, se debe realizar la diferencia entre mediciones consecutivas.

A continuación, en la figura 3.12, se muestra una representación de estos subnodos donde se muestran los *topics* generados durante la ejecución del nodo *rqt_graph*.

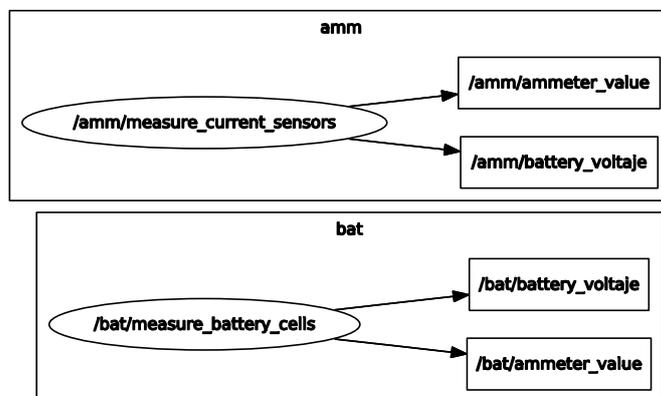


Figura 3.12. Captura del nodo *battery_charge* con el comando *rqt_graph*.

ual_ecar_vehicle_controller

Este nodo se encuentra disponible en el *package* *ual-ecar-ros-pkg*, disponible en un repositorio de *Github*⁷. Este nodo es el encargado de establecer la comunicación entre *ROS* y los dos microcontroladores *Claraquino v.1* encargados de controlar los sistemas *Steer-by-Wire* y *Throttle* y *Brake-by-Wire*.

⁷Github: https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg/tree/master/ual_ecar_vehicle_controller

Se encuentra suscrito a los *topics* que genera el nodo *joystick_driving* y traduce estos valores a las señales que deben recibir los lazos de control correspondientes. Como respuesta, publica las lecturas de todos los sensores involucrados y un vector de las variables del lazo de control capturadas en el mismo instante de tiempo. En la figura 3.13 se muestra el mapeo de variables que recibe y publica el nodo.

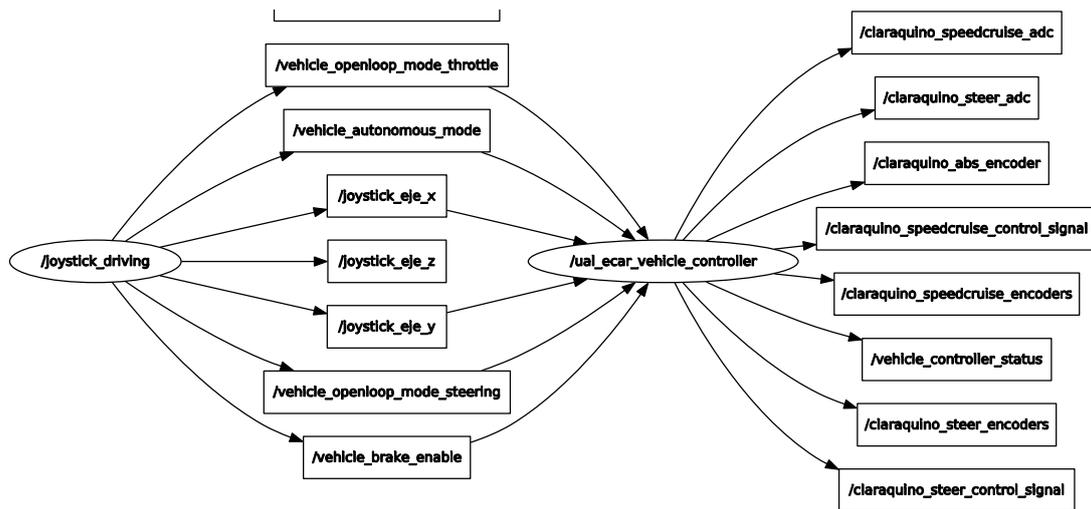


Figura 3.13. Captura del nodo *ual_ecar_vehicle_controller* con el comando *rqt_graph*.

A continuación, en la figura 3.14, se muestra el proceso principal que se desarrolla en el nodo. Como se ha comentado antes, el nodo lee las señales generadas por el joystick y en el código mostrado las adapta para enviarlas a los lazos de control correspondientes.

En primer lugar se realiza la comprobación de que el vehículo no esté en modo autónomo, */vehicle_autonomous_mode*, y además se haya enviado alguna señal nueva desde el joystick comprobando que se ha leído un valor de alguno de los ejes. Esto es una medida de seguridad para evitar que el vehículo se mueva sin haber enviado una consigna específicamente. En el siguiente paso, se comprueba si el control del sistema *Steer-by-Wire* está en lazo abierto o cerrado. Si el */vehicle_openloop_mode_steering* está a “1”, se considera que está en lazo abierto y por tanto */joystick_eje_x* se considera la señal de control PWM que se inyecta directamente al motor, por lo que se multiplica por 255. En caso contrario, la señal se considera que es la referencia de la posición que se desea para el mecanismo y se multiplica por 512. Este valor se debe a que se considera que la posición máxima del mecanismo oscila entre un valor mínimo de -512 y $+512$.

De igual forma, el procesamiento del sistema *Throttle-by-Wire* y *Brake-by-Wire* se realiza de forma simultánea y presenta una dinámica similar. Si el */vehicle_openloop_mode_throttle* está a “1”, se considera que el lazo de control está abierto y por tanto */joystick_eje_y* se corresponde con la señal del acelerador o con el freno si */vehicle_brake_eneable* está a “1”. La señal del acelerador oscila entre un valor máximo de 1 y un mínimo de -1 . Los valores negativos están asociados al valor del acelerador cuando se mueve con la marcha reversa activada.

```

if (!m_automonomous_driving_mode && m_joy_changed){
    // X: Steering
    if (m_mode_openloop_steer){
        TFrameCMD_OPENLOOP_STEERING_SETPOINT cmd;
        cmd.payload.SETPOINT_OPENLOOP_STEER_SPEED=m_joy_x*255.0;

        cmd.calc_and_update_checksum();
        WriteBinaryFrame(reinterpret_cast<uint8_t*>(&cmd),
            sizeof(cmd), m_serial_Steer);
        ROS_INFO_THROTTLE(1, "Sending openloop STEER: %.d",
            cmd.payload.SETPOINT_OPENLOOP_STEER_SPEED); }

    else{
        TFrameCMD_CONTROL_STEERING_SETPOINT cmd;
        cmd.payload.SETPOINT_STEER_POS = 512 * m_joy_x;
        cmd.calc_and_update_checksum();
        WriteBinaryFrame(reinterpret_cast<uint8_t*>(&cmd),
            sizeof(cmd), m_serial_Steer);
        ROS_INFO_THROTTLE( 1, "Sending closedloop STEER: %.d",
            cmd.payload.SETPOINT_STEER_POS); }

    // Y: Throttle+Brake
    if (m_mode_openloop_throttle){
        TFrameCMD_OPENLOOP_THROTTLE_SETPOINT cmd;
        TFrameCMD_OPENLOOP_BRAKE_SETPOINT cmd_brake;

        if (!m_mode_brake_enable){
            cmd.payload.SETPOINT_OPENLOOP_THROTTLE = m_joy_y;}
        else{
            cmd_brake.payload.SETPOINT_OPENLOOP_BRAKE=m_joy_y * 255.0
                * 0.5;}

        cmd.calc_and_update_checksum();
        WriteBinaryFrame(reinterpret_cast<uint8_t*>(&cmd), sizeof(cmd),
            m_serial_SpeedCruise);
        ROS_INFO_THROTTLE( 1, "Sending openloop THROTTLE: %.f", cmd.
            payload.SETPOINT_OPENLOOP_THROTTLE);

        cmd_brake.calc_and_update_checksum();
        WriteBinaryFrame(reinterpret_cast<uint8_t*>(&cmd_brake), sizeof(
            cmd_brake), m_serial_SpeedCruise);
        ROS_INFO_THROTTLE(1, "Sending openloop Brake: %.d", cmd_brake.
            payload.SETPOINT_OPENLOOP_BRAKE);}

    else{
        const float MAX_VEL_MPS = 2.0;
        float vel_mps = m_joy_y * MAX_VEL_MPS;
        TFrameCMD_CONTROL_THROTTLE_SETPOINT cmd;

        cmd.payload.SETPOINT_CONTROL_THROTTLE_SPEED = vel_mps;
        cmd.calc_and_update_checksum();
        WriteBinaryFrame(reinterpret_cast<uint8_t*>(&cmd),
            sizeof(cmd), m_serial_SpeedCruise);
        ROS_INFO_THROTTLE( 1, "Sending closedloop THROTTLE: %.03f m/s",
            vel_mps);
    }
}

```

Figura 3.14. Configuración del nodo para el sistema *Drive-by-Wire-by-Wire*.

En el caso del freno, al ser el actuador un motor que requiere una señal PWM, el valor de `/joystick_eje_y` debe multiplicarse por 255. En caso de que el lazo de control se encuentre cerrado, el valor de `/joystick_eje_y` se multiplica por la variable `MAX_VEL_MPS` que establece la velocidad límite que se le va a introducir como referencia al vehículo y el microcontrolador correspondiente se encarga de determinar las señales de control para el acelerador y para el freno.

libredaq

Este nodo se encuentra ubicado dentro del *package libredaq* disponible en un repositorio en la plataforma *Github*⁸. No es un nodo desarrollado para este vehículo, sino que forma parte de la plataforma modular de adquisición de datos desarrollada en la Universidad de Almería [84]. La función principal es la de hacer de enlace entre la tarjeta hardware y *ROS*. En esta configuración, el nodo no está suscrito a ningún *topic* y tiene la posibilidad de estar conectado hasta a seis codificadores de los cuales publica su estado en *topics* independientes con la nomenclatura `/joint_states_ch“X”_decim_speed` o de forma conjunta, pero con un muestreo diezmado, en el *topic* `/joint_states`. En el caso del vehículo UAL-eCARM, los *topics* `/joint_states_ch0_decim_speed` y `/joint_states_ch1_decim_speed` se corresponden con las señales de velocidad y posición de los codificadores acoplados a la rueda izquierda y derecha respectivamente.



Figura 3.15. Captura del nodo `libredaq` y `ual_ecar_odometry` con el comando `rqt_graph`.

⁸Github: <https://github.com/LibreDAQ/libredaq>

ual_ecar_odometry

Este nodo se encuentra ubicado dentro del *package ual-ecar-ros-pkg* disponible en un repositorio en la plataforma *Github*⁹. Este nodo depende directamente del nodo */libredaq* al necesitar la lectura de los codificadores situados en la parte posterior del vehículo que realiza dicho nodo. Como resultado de la ejecución, el nodo publica dos *topics*, la odometría del vehículo en */odom*, y la matriz de transformación de la posición actual relativa al punto de partida donde se inició la ejecución del nodo.

3.4. Arquitectura firmware

Una parte importante del grueso del trabajo realizado en este proyecto ha sido trasladar la arquitectura de control del nivel software a nivel firmware. Esta decisión ha sido respaldada por motivos de seguridad. Con la anterior arquitectura, si ocurría alguna interrupción en la comunicación entre la placa arduino, que enviaba la señal a los actuadores, y la raspberry pi, que hacía las veces de PC principal, podía darse el caso que las señales de control se quedasen activas y no respondiesen a las instrucciones del usuario. Esto podía desembocar en roturas de los mecanismos físicos si el usuario no intervenía inmediatamente cortando el suministro de corriente a los actuadores. Puesto que el código es muy extenso como para mostrarse completo y está disponible online¹⁰, se mostrarán las secciones más relevantes de la arquitectura de control, no siendo necesario su inclusión en un anexo al presente trabajo.

3.4.1. Battery_charge

También se ha realizado el proceso de construcción de las placas de adquisición de datos descritas en la sección 3.2.8, desde la soldadura de componentes hasta su instalación en el vehículo. Estas placas han sido necesarias para la sustitución del anterior sistema de adquisición de datos formado por una tarjeta DAQ de National Instrument NI UBS-6211. Dicha tarjeta presentaba el problema de no poder ser leída directamente en ROS, por lo que se optó por el desarrollo de unas nuevas tarjetas que permitieran centralizar todo el flujo de información del vehículo. Al tratarse de una tarjetas de lectura de datos, la envergadura de su programación es menor. Fundamentalmente, haciendo uso de la librería *libclaraquino/modules/adc_ad7606/ad7607.h*, se han configurado los registros del microcontrolador y los pines de conexión. Finalmente, el estado de esos registros se transforman en variables que se transfieren al nodo *battery_charge*, sección 3.3.2.

⁹Github: https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg/tree/master/ual_ecar_odometry

¹⁰Github: <https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg>

3.4.2. Ual_ecar_vehicle_controller

En el estado actual toda la arquitectura de control está implementada en el firmware de dos placas claraquino, sección 3.2.7, quedando el nivel software relegado a un puente en la comunicación con otros elementos como el joystick o los codificadores situados en las ruedas traseras. En este nivel también se han implementado mecanismos de seguridad como cambiar la consigna de control a un valor nulo en caso de corte de comunicación con el nivel software. Cada microcontrolador tiene su propio proyecto de firmware donde se han agrupado las funciones según el sistema al que pertenecen. De esta forma, todas aquellas funciones relacionadas con el sistema de dirección del vehículo se han concentrado en el proyecto “Steer controller firmware”. Por otra parte, las funciones de los sistemas para el control de la velocidad longitudinal del vehículo se han concentrado en el proyecto “Vehicle cruise control”.

Steer controller firmware

En este proyecto, se ha establecido todo lo concerniente al sistema *Steer-by-Wire*. En primer lugar, en la figura 3.16, se muestra la sección de la función principal del proyecto tras la inicialización de todos los dispositivos. Se debe destacar que antes de proceder a ejecutar la interacción del controlador, se leen los distintos sensores del sistema. El sensor de corriente es una medida que realiza la etapa de potencia y recoge el puerto “A” del microcontrolador con la función *processADCs()*. La función *processEncoders()* realiza la lectura del codificador incremental HEDL5540 acoplado al motor de la dirección. El codificador absoluto EMS22A, acoplado mecanismo de la dirección, es leído mediante la llamada a la función *processEMS22A()*. Finalmente, además de las estadísticas de tiempo de la CPU, se realiza la llamada a la función donde se encuentra implementada la arquitectura de control.

```
// ===== Infinite loop =====  
while(1){  
    // —— Run scheduled tasks ——  
    processIncommingPkts();  
    processADCs();  
    processEncoders();  
    processEMS22A();  
    processSteerController();  
    process_timeouts(); // Handle possible timeouts of previous commands:  
    const uint32_t t_end = millis();  
    const uint32_t dt = t_end-t_ini; // by computing it here, we also count  
        the cost of processCPUStats()  
    t_ini = millis();  
    // —— CPU busy time stats ——  
    processCPUStats(dt);  
}
```

Figura 3.16. Ejecución principal del proceso del sistema *Steer-by-Wire*.

En la figura 3.17, se realiza la lectura de parámetros principales del controlador, así como la constante del mecanismo antiwindup. Actualmente los parámetros se introducen como constantes, pero el sistema se ha diseñado con esta posibilidad de configuración para en un futuro, si se considera pertinente, implementar un control más avanzado, como uno autoajustable en línea. También se observa en dicha figura la lectura de la señal de control en caso de operar en lazo abierto y de la referencia de la dirección en caso de operar en lazo cerrado. Igualmente, se permite realizar la lectura del offset inicial del mecanismo de forma externa por si fuera necesario en un futuro frente a técnicas de fusión de sensores. Además de los parámetros del controlador, también se implementa la lectura de las señales de control del mecanismo de desacoplo respecto al sistema de tracción del vehículo.

```

void setSteer_ControllerParams(const
    TFrameCMD_CONTROL_STEERING_SET_PARAMS_payload_t &p){
    for (int i=0;i<3;i++){
        Q_STEER_INT[i] = p.Q_STEER_INT[i];
        Q_STEER_EXT[i] = p.Q_STEER_EXT[i];
    }

    Steer_offset = p.STEER_OFFSET;
    ANTIWINDUP_CTE = p.Antiwindup;
    for (int i=0;i<2;i++){
        U_STEER_DECOUPLING[i] = p.U_STEER_DECOUPLING[i];
        U_STEER_FEEDFORWARD[i] = p.U_STEER_FEEDFORWARD[i];
    }
}

void setSpeed_Vehicle(float speed_vehicle_ecarm){
    SPEED_ECARM = speed_vehicle_ecarm;
    SPEED_ECARM_TIMESTAMP = millis();
}

void setOpenLoopSetpoint_Steer(int16_t speed){
    SETPOINT_OPENLOOP_STEER_SPEED = speed;
    SETPOINT_OPENLOOP_STEER_TIMESTAMP = millis();
}

void setControllerSetpoint_Steer(int16_t pos){
    SETPOINT_STEER_POS=pos;
    SETPOINT_STEER_TIMESTAMP = millis();
}

```

Figura 3.17. Lectura de parámetros del sistema *Steer-by-Wire*.

En la figura 3.18 se muestra el fragmento de código encargado de la transformación de las señales de los codificadores. En primer lugar realiza la lectura del codificador absoluto y transforma su señal que originalmente oscila entre $[1 - 1365]$ pulsos a su equivalente en el ángulo de ackermann del vehículo entre $\pm 32,5$ [°]. Acto seguido se le aplica un filtro a la salida para evitar valores erróneos. El mismo proceso se realiza con el codificador incremental para poder trabajar con un sistema redundante que mantenga el nivel de seguridad en la realimentación del vehículo. Esta redundancia se observa en que si se aprecia una desviación superior a dos grados entre la señal de los codificadores, se recalcula el offset del sistema.

3.4. ARQUITECTURA FIRMWARE

```
// ===== Encoders calibration algorithm =====
// Incremental encoder reading
cli();
const int32_t enc_diff = enc_last_reading.encoders[0];
sei();

// Read abs encoder:
const int16_t abs_enc_pos_new=(enc_abs_last_reading.enc_pos-Steer_offset-512)
    *(32.5/(1138/2)); // Abs encoder (10 bit resolution)

// Filter out clearly erroneous readings from the abs encoder:
if (abs(abs_enc_pos_new - abs_enc_pos)<512)
    abs_enc_pos = abs_enc_pos_new;

// Calibration with absolute encoder:
const float K_enc_diff = 337.0f *(32.5/(1138/2))/(500.0f * 100.0f);

// Inc. encoder calibration if encoders differences are upper than 2
const float Adiff = enc_diff * K_enc_diff;
if (abs(abs_enc_pos - (enc_offset_correction+Adiff))>2){
    // Recalc offset:
    enc_offset_correction = abs_enc_pos - Adiff;
}
}
```

Figura 3.18. Calibración de los codificadores acoplados al mecanismo de dirección.

La realimentación del lazo externo se puede realizar directamente con la lectura del valor que devuelven los codificadores, no obstante, la del lazo interno debe realizarse como una medida indirecta. En la figura 3.19 se muestra el cálculo de la velocidad a partir del incremento en la medida del codificador y de el tiempo transcurrido entre iteraciones. En la misma figura, se muestra el procesado de la señal en control manual. Se observa la implementación de un mecanismo de seguridad para evitar que frente a una pérdida de comunicación, el sistema mande la señal de control anterior de forma indefinida. Por lo tanto, tras un tiempo preestablecido, se anula la señal de control.

```
// Control:
/*      Speed encoder reading*/
float rpm = (Encoder_dir[0] - Encoder_dir[1]) / T;

if (!STEERCONTROL_active){
    // open-loop mode:
    // Watchdog timer:
    if (tnow>(SEIPOINT_OPENLOOP_STEER_TIMESTAMP+ WATCHDOG_TIMEOUT_msth))
        SEIPOINT_OPENLOOP_STEER_SPEED = 0;

    U_steer_controller[0] = SEIPOINT_OPENLOOP_STEER_SPEED;
}
}
```

Figura 3.19. Procesado de señal del sistema *Steer-by-Wire* en lazo abierto.

En la figura 3.21 se muestra la ejecución principal del sistema de control en cascada del vehículo. Como se puede apreciar, todo está estructurado de forma genérica, por lo que la futura implementación de cualquier técnica de control es susceptible de ser introducida sin realizar grandes modificaciones del sistema ya existente. Además todas las acciones se encuentran comentadas debidamente a fin de que cualquiera que lea el código pueda saber interpretar que se realiza en cada una. Esto es de especial importancia, ya que el código se encuentra disponible online y cualquier usuario puede acceder a él. Toda la estructura de control desarrollada en la sección 5.2 se encuentra implementada en tiempo discreto en este fragmento de código. En primer lugar se realiza la lectura de la referencia de posición y se cambia el signo para corregir el sentido de movimiento del mecanismo. Se implementa una pequeña saturación en la pendiente de cambio de consigna. Esto se hace para evitar dinámicas agresivas sobre el mecanismo, aunque está diseñado para soportarlas. Acto seguido se calcula el error y el controlador externo genera la referencia para la velocidad del mecanismo. Tras calcular el error del lazo interno y su consecuente señal de control, se añade el efecto del desacoplo respecto del sistema de tracción. El desacoplo se calcula como un factor de 0,9515 si el mecanismo de la dirección y el vehículo se encuentran en movimiento.

En la figura 3.20 se muestra la implementación que asegura que si el sistema se encuentra próximo al límite, se corta la señal de control y no se deja que continúe avanzando para acercarse más al límite. Se estima oportuno que una distancia de 5 grados respecto a los extremos es una distancia de control prudencial. Con la instrucción *do_shift()* se actualizan los valores de los vectores del lazo de control para la próxima iteración.

```
// for both, open and closed loop: protection against steering mechanical limits:
steer_mech_limit_reached =
    steer_mech_limit_reached ?
    (abs(abs_enc_pos_new) > (steer_mech_limit_pos - 5))
    :
    (abs(abs_enc_pos_new) >= steer_mech_limit_pos);
// Disallow going further outwards:
if (steer_mech_limit_reached){
    if(abs_enc_pos_new > 0 && U_steer_controller[0] > 0)
        U_steer_controller[0] = 0;

    if(abs_enc_pos_new < 0 && U_steer_controller[0] < 0)
        U_steer_controller[0] = 0;
}
/* Values actualization*/
do_shift(Ref_pos);
do_shift(Antiwindup);
do_shift(Error_pos);
do_shift(Ref_speed);
do_shift(Error_speed);
do_shift(Encoder_dir);
do_shift(U_steer_controller);
```

Figura 3.20. Protección en el extremo del mecanismo de dirección.

```

else{
    // Automatic mode
    /*
    +-----+
    |  STEER-BY-WIRE  |
    +-----+ */
    if (tnow>(SETPOINT_STEER_TIMESTAMP + WATCHDOG_TIMEOUT_msth))
        SETPOINT_STEER_POS = 0;

    /*
    Position reference reading */
    Ref_pos[0] = SETPOINT_STEER_POS;
    /*
    Correction of the wheels' direction */
    Ref_pos[0] = - Ref_pos[0];
    /*
    Slope reference limit to over current protection*/
    float pendiente = (Ref_pos[0] - Ref_pos[1]) / T;
    if (pendiente >= sat_ref)
        Ref_pos[0] = (Ref_pos[1] + sat_ref);
    /*
    Position error. Extern loop*/
    Error_pos[0] = Ref_pos[0] - Encoder_dir[0];
    /*
    Position controller */
    Ref_speed[0] = Ref_speed[1] + Q_STEER_EXT[0] * Error_pos[0] + Q_STEER_EXT
        [1] * Error_pos[1] + Q_STEER_EXT[2] * Error_pos[2];
    /*
    Speed error. Intern loop*/
    Error_speed[0] = Ref_speed[0] - rpm;
    /*
    Speed controller */
    U_steer_controller[0] = U_steer_controller[1] + Q_STEER_INT[0] *
        Error_speed[0] + Q_STEER_INT[1] * Error_speed[1] + Q_STEER_INT[2] *
        Error_speed[2];
    /*
    Control Signal with feedforward & decoupling*/
    if(SPEED_eCARM !=0 && rpm !=0)
        U_steer_controller[0] = 0.9515;
    else
        U_steer_controller[0] = 0;
    U_steer_controller[0] = U_steer_controller[0] + U_STEER_DECOUPLING[0] +
        U_STEER_DECOUPLING[1] + U_STEER_FEEDFORWARD[0] + U_STEER_FEEDFORWARD
        [1];
    /*
    Variable to Anti-windup technique*/
    int m_v= round(U_steer_controller[0]);

    /*
    Saturation */
    bool has_sat = false;
    if (abs(U_steer_controller[0]) > 24){
        if (U_steer_controller[0] < -24)
            U_steer_controller[0] = -24;
        else
            U_steer_controller[0] = 24;
        has_sat = true;
    }
    /*
    Anti-windup technique*/
    if(has_sat)
        Antiwindup[0] = (U_steer_controller[0] - m_v) / ANTIWINDUP_CTE;
    else
        Antiwindup[0] = 0;
    U_steer_controller[0] = round(0.5 * (2 * U_steer_controller[0] + T * (
        Antiwindup[0] + Antiwindup[1])));
}

```

Figura 3.21. Lazo de control del sistema *Steer-by-Wire*.

En la figura 3.22 se muestra el procesado de la señal de control generada por la arquitectura de control propuesta. Esta señal debe convertirse de tensión en voltios a su equivalente en PWM. El signo de la señal se traduce en una señal lógica que se envía a parte a la etapa de potencia. Finalmente, el último proceso es registrar todos los valores relevantes del sistema de control en un único mensaje sincronizado. Este mensaje se envía a ROS donde se emplean los datos pertinentes en otros nodos o para ser grabados en datasets de ensayos.

```

/*      Direction*/
bool u_steer_is_positive = (U_steer_controller[0] >= 0);

uint8_t u_steer = abs(U_steer_controller[0]) *255/24;

// Output PWM:
pwm_set_duty_cycle(PWM_OUT_TIMER,PWM_OUT_PIN, u_steer);

// PWM direction:
gpio_pin_write(PWM_DIR, u_steer_is_positive);

TFrame_STEER_CONTROL_SIGNAL tx;

// Decimate the number of msgs sent to the PC:
if (++decim_sent_frame>global_decimate.decimate_CONTROLSIGNAL){
    decim_sent_frame=0;

    tx.payload.timestamp_ms_tenth = tnow;
    tx.payload.Steer_control_signal = U_steer_controller[0];
    tx.payload.Encoder_absoluto = abs_enc_pos;
    tx.payload.Encoder_incremental = enc_diff;
    tx.payload.Encoder_signal = Encoder_dir[0];
    tx.payload.Steer_ADC_current_sense = ADC_last_reading.adc_data[0];
    tx.payload.steer_mech_limit_reached = steer_mech_limit_reached;
    tx.payload.enc_offset_correction = enc_offset_correction;

    tx.calc_and_update_checksum();

    UART::Write((uint8_t*)&tx, sizeof(tx));
}

```

Figura 3.22. Procesado de la señal de control y envío de datos a ROS.

Vehicle cruise control

En este proyecto, queda comprendido todo lo concerniente a la operación, tanto manual como controlada de forma automática, del sistema de tracción del vehículo. Por lo tanto, todo el desarrollo de los sistemas *Throttle-by-Wire* y *Brake-by-Wire* debe implementarse en este proyecto. Todo el proceso desarrollado en las secciones 5.4 y 5.3 se muestra a continuación. Muchos de los procesos que se muestran a continuación presentarán una estructura similar a los mostrados para el sistema de dirección puesto que la filosofía seguida en la programación ha sido la misma.

En la figura 3.23, se muestra la sección de la función principal del proyecto tras la inicialización de todos los dispositivos. En primer lugar se realiza las lecturas de los pines analógicos del microcontrolador. Actualmente se encuentra conectado a este puerto el sensor de corriente de la etapa de potencia del motor del freno y la salida del conversor digital analógico que emula la señal del pedal de aceleración. Esta señal se ha considera oportuna a modo de detección de fallos. Si esta señal no coincide con la que supuestamente se genera en el microcontrolador, se corta la señal y notificaría un fallo. Por su parte, el sensor de corriente en el freno, puede ser utilizado para detectar el extremo del mecanismo cuando se detecten valores elevados de la señal. En segundo lugar se lee el codificador acoplado al motor de freno, que representa la realimentación del lazo de control del sistema *Brake-by-Wire*. En este caso, a diferencia de lo ocurrido con el mecanismo de dirección, la realimentación se realiza de la posición del motor, sin tener que calcular a posteriori su velocidad. El siguiente paso es la ejecución del control de freno y tracción del vehículo que se verá más adelante detalladamente. Finalmente, se termina el bucle principal del proyecto con las estadísticas temporales del microcontrolador.

```
// ===== Infinite loop =====  
while(1){  
    // —— Run scheduled tasks ——  
    processIncommingPkts();  
  
    processADCs();  
    processEncoders();  
    processBrakeController();  
    processThrottleController();  
    process_timeouts(); // Handle possible timeouts of previous commands:  
  
    const uint32_t t_end = millis();  
    const uint32_t dt = t_end-t_ini; // by computing it here, we also count  
        the cost of processCPUStats()  
    t_ini = millis();  
  
    // —— CPU busy time stats ——  
    processCPUStats(dt);  
}
```

Figura 3.23. Ejecución principal de los sistemas *Throttle-by-Wire* y *Brake-by-Wire*.

Pese a que el control del freno y del acelerador se realice en dos funciones distintas, estas se encuentran implementadas en el mismo fichero del proyecto, “*proces_controller.cpp*”. Por esta razón, en la figura 3.24 se muestra la lectura de los parámetros de control y referencias de los sistemas *Brake-by-Wire* y *Throttle-by-Wire*. Al igual que ocurría con el sistema de dirección, para estos dos sistemas se han implementado las estrategias de control de feedforward y desacoplos. Nuevamente la finalidad de implementar estrategias que no se emplean actualmente reside en dejar para futuros trabajos de control, el sistema lo más preparado posible.

```

void setThrottle_ControllerParams(const
    TFrameCMD_CONTROL_THROTTLE_SET_PARAMS_payload_t &p){
    for (int i=0;i<3;i++)
        Q_THROTTLE[i] = p.Q_THROTTLE_CONTROLLER[i];
    for (int i=0;i<2;i++)
        U_THROTTLE_FEEDFORWARD[i] = p.U_THROTTLE_FEEDFORWARD[i];
    U_THROTTLE_DECOUPLING = p.U_THROTTLE_DECOUPLING;
}
void setBrake_ControllerParams(const TFrameCMD_CONTROL_BRAKE_SET_PARAMS_payload_t
    &p){
    for (int i=0;i<3;i++)
        Q_BRAKE[i] = p.Q_BRAKE_CONTROLLER[i];
    for (int i=0;i<2;i++)
        U_BRAKE_FEEDFORWARD[i] = p.U_BRAKE_FEEDFORWARD[i];
    U_BRAKE_DECOUPLING = p.U_BRAKE_DECOUPLING;
}
void setSpeed_Vehicle(float speed_vehicle_ecarm){
    SPEED_eCARM = speed_vehicle_ecarm;
    SPEED_eCARM_TIMESTAMP = millis();
}
void setOpenLoopSetpoint_VehVel(float ol_vel_mps{
    SETPOINT_OPENLOOP_THROTTLE = ol_vel_mps;
    SETPOINT_OPENLOOP_THROTTLE_TIMESTAMP = millis();
}
void setOpenLoopSetpoint_Brake(int16_t ol_brakeforce){
    SETPOINT_OPENLOOP_BRAKE = ol_brakeforce;
    SETPOINT_OPENLOOP_BRAKE_TIMESTAMP = millis();
}
void setControllerSetpoint_VehVel(float vel_mps){
    SETPOINT_CONTROL_THROTTLE_SPEED = vel_mps;
    SETPOINT_CONTROL_THROTTLE_SPEED_TIMESTAMP = millis();
}

```

Figura 3.24. Lectura de parámetros de los sistemas *Throttle-by-Wire* y *Brake-by-Wire*.

En la figura 3.25 se muestra el fragmento de código que refleja el control en lazo abierto de los sistemas *Throttle-by-Wire* y *Brake-by-Wire*. La señal de la aceleración está comprendida entre 0 – 5 [V] y la del freno entre ± 24 [V].

```

if (!THROTTLECONTROL_active){
    if (tnow>(SETPOINT_OPENLOOP_THROTTLE_TIMESTAMP + WATCHDOG_TIMEOUT_msth))
        SETPOINT_OPENLOOP_THROTTLE = 0;
    U_throttle_controller[0] = SETPOINT_OPENLOOP_THROTTLE * 5;
}
...
if (!BRAKECONTROL_active){
    if (tnow>(SETPOINT_OPENLOOP_BRAKE_TIMESTAMP + WATCHDOG_TIMEOUT_msth))
        SETPOINT_OPENLOOP_BRAKE = 0;
    U_brake_controller[0] = SETPOINT_OPENLOOP_BRAKE*24;
}

```

Figura 3.25. Procesado de señal en lazo abierto de los sistemas *Throttle-by-Wire* y *Brake-by-Wire*.

En la figura 3.26 se refleja la ejecución del lazo de control del sistema *Throttle-by-Wire*. Se trata de una estructura simple de control donde el error se convierte en al referencia del sistema de frenado como se comenta más adelante. Los parámetros del controlador se implementan variables en función del punto de operación. Tras el cálculo de la señal de control pertinente, considerando los posibles desacoplos y feedforwards, se le suma el voltio correspondiente a la zona muerta del sistema.

```

else{ /*+-----+
/   THROTTLE-BY-WIRE   /
+-----+*/
if (tnow>(SETPOINT_CONTROL_THROTTLE_SPEED_TIMESTAMP +
WATCHDOG_TIMEOUT_msth))
SETPOINT_CONTROL_THROTTLE_SPEED = 0;

/*Speed reference reading*/
Ref_speed [0] = 0.9254*Ref_speed [1]+0.07459*SETPOINT_CONTROL_THROTTLE_SPEED
;
/*Speed error and brake control reference*/
Error_speed [0] = Ref_speed [0] - SPEED_eCARM;
if (Error_speed [0]<0){
Ref_brake [0] = Error_speed [0];
} else{
Ref_brake [0] = 0;
}
}
/*Speed controller*/
if (SPEED_eCARM>0){
if (SPEED_eCARM>2.3){ Q_THROTTLE[0] = 2.406;
Q_THROTTLE[1] = -2.328;
Q_THROTTLE[2] = 0.0;
} else{ Q_THROTTLE[0] = 1.994;
Q_THROTTLE[1] = -1.929;
Q_THROTTLE[2] = 0.0;
}
} else{
if (SPEED_eCARM>-2.1){ Q_THROTTLE[0] = 2.482;
Q_THROTTLE[1] = -2.402;
Q_THROTTLE[2] = 0.0;
} else{ Q_THROTTLE[0] = 4.505;
Q_THROTTLE[1] = -4.358;
Q_THROTTLE[2] = 0.0;
}
}
}
U_throttle_controller [0] = U_throttle_controller [1] + Q_THROTTLE[0] *
Error_speed [0] + Q_THROTTLE[1] * Error_speed [1] + Q_THROTTLE[2] *
Error_speed [2];
/*Control Signal with feedforward & decoupling*/
U_throttle_controller [0] = U_throttle_controller [0] +
U_THROTTLE_DECOUPLING[0] + U_THROTTLE_DECOUPLING[1] +
U_THROTTLE_FEEDFORWARD[0] + U_THROTTLE_FEEDFORWARD[1];
U_throttle_controller [0] = U_throttle_controller [0] + 1.0; // Dead Zone
}

```

Figura 3.26. Controlador del sistema *Throttle-by-Wire*.

En la figura 3.27 se muestra el control de la saturación y el mecanismo anti-windup para el sistema *Throttle-by-Wire*. Al igual que en el caso del sistema *Steer-by-Wire*, el proceso finaliza con la actualización de los vectores del controlador, envío de la señal correspondiente a los pines del microcontrolador y el registro de variables para el *ROS*.

```

/*      Variable to Anti-windup technique*/
int m_speed= round(U_throttle_controller[0]);
/*      Saturation */
bool has_sat = false;
if (abs(U_throttle_controller[0]) > 5){
    if (U_throttle_controller[0] < -5)
        U_throttle_controller[0] = -5;
    else
        U_throttle_controller[0] = 5;
    has_sat = true;
}
/*      Anti-windup technique*/
if(has_sat)
    Antiwindup[0] = (U_throttle_controller[0] - m_speed) / ANTIWINDUP_CTE;
else
    Antiwindup[0] = 0;
U_throttle_controller[0] = round(0.5 * (2 * U_throttle_controller[0] + T * (
    Antiwindup[0] + Antiwindup[1])));

/* Values actualization*/
do_shift(Ref_speed);
do_shift(Antiwindup);
do_shift(Error_speed);
do_shift(U_throttle_controller);

// Relay output = HIGH if going BACKWARDS.
gpio_pin_write(RELAY_FRWD_REV, U_throttle_controller[0]<0);
// Pedal enable relay (to Curtis controller J1-8 pin)
gpio_pin_write(RELAY_PEDAL_INTERLOCK,abs(U_throttle_controller[0])>0.1f);
// Output value:
uint16_t veh_speed_dac = abs(U_throttle_controller[0]* 4095/5.0); // 12bit DAC
    constant
mod_dac_max5500_update_single_DAC(0 /*DAC idx*/, veh_speed_dac);
TFrame_SPEEDCRUISE_CONTROL_SIGNAL tx;
// Decimate the number of msgs sent to the PC:
static uint8_t decim0 = 0;
if (++decim0>global_decimate.decimate_CONTROLSIGNAL){
    decim0=0;
    tx.payload.timestamp_ms_tenth = tnow;
    tx.payload.Throttle_control_signal = U_throttle_controller[0];
    tx.payload.Brake_control_signal = U_brake_controller[0];
    tx.payload.Throttle_analog_feedback = ADC_last_reading.adc_data[1];
    tx.payload.Brake_Encoder_incremental = enc_last_reading.encoders[0];
    tx.payload.Brake_ADC_current_sense = ADC_last_reading.adc_data[0];
    tx.calc_and_update_checksum();
    UART::Write((uint8_t*)&tx, sizeof(tx));
}

```

Figura 3.27. Procesado de la señal de control del sistema *Throttle-by-Wire*.

3.4. ARQUITECTURA FIRMWARE

En la figura 3.28 se muestra el lazo de control para el caso del sistema *Brake-by-Wire*. Es el sistema más simple de los tres implementados en el presente trabajo. Se trata de un controlador PI que recibe la señal de control del error del sistema *Throttle-by-Wire*. En este fragmento de código se incluye el control de saturación y el mecanismo anti-windup correspondiente.

```
else{
/*      +-----+
/   BRAKE-BY-WIRE   /
+-----+
*/
if (tnow>(SETPOINT_CONTROL_BRAKE_FORCE_TIMESTAMP + WATCHDOG_TIMEOUT_msth))
    SETPOINT_CONTROL_BRAKE_FORCE = 0;

/*Brake reference reading*/
Ref_brake[0]=0.9265*Ref_brake[1]+Ref_brake[0]*0.75; // Experimental factor
/*Position error*/
Error_brake[0] = Ref_brake[0] - enc_diff;
/*Position controller*/
U_brake_controller[0] = U_brake_controller[1]+Q_BRAKE[0]*Error_brake[0]+
    Q_BRAKE[1]*Error_brake[1]+Q_BRAKE[2]*Error_brake[2];
/*Control Signal with feedforward & decoupling*/
U_brake_controller[0] = U_brake_controller[0] + U_BRAKE_DECOUPLING[0] +
    U_BRAKE_DECOUPLING[1] + U_BRAKE_FEEDFORWARD[0] + U_BRAKE_FEEDFORWARD
    [1];
}

/*    Variable to Anti-windup technique*/
int m_brake= round(U_brake_controller[0]);
/*    Saturation */
bool has_sat = false;
if (abs(U_brake_controller[0]) > 24){
    if (U_brake_controller[0] < -24)
        U_brake_controller[0] = -24;
    else
        U_brake_controller[0] = 24;
    has_sat = true;
}
/*    Anti-windup technique*/
if(has_sat)
    Antiwindup_brake[0] = (U_brake_controller[0] - m_brake) /
        ANTIWINDUP_CTE_BRAKE;
else
    Antiwindup_brake[0] = 0;
U_brake_controller[0] = round(0.5 * (2 * U_brake_controller[0] + T * (
    Antiwindup_brake[0] + Antiwindup_brake[1])));
```

Figura 3.28. Controlador del sistema *Brake-by-Wire*.

Finalmente, en la figura 3.29 se muestra la actualización de los vectores involucrados en el lazo de control de este sistema y el envío de los valores PWM y el sentido de giro a los pines del microcontrolador donde se encuentra conectada la etapa de potencia del sistema.

```
/* Values actualization*/
do_shift(U_brake_controller);
do_shift(Ref_brake);
do_shift(Error_brake);
do_shift(Antiwindup_brake);

// Output PWM:
bool u_brake_is_positive = (U_brake_controller[0] >= 0);
uint8_t u_brake = abs(U_brake_controller[0]) * 255 / 24;
pwm_set_duty_cycle(PWM_OUT_TIMER, PWM_OUT_PIN, u_brake);

// PWM direction:
gpio_pin_write(PWM_DIR, u_brake_is_positive);
```

Figura 3.29. Procesado de la señal de control del sistema *Brake-by-Wire*.

Capítulo 4

Modelado de la dinámica del vehículo

4.1. Introducción

Una vez definido el estado del vehículo detalladamente en la sección anterior, se procede a modelar su comportamiento. Los tres principales sistemas que se van a tratar comprenden el mecanismo de dirección, *Steer-by-Wire*, el de freno, *Brake-by-Wire* y el de tracción del vehículo, *Throttle-by-Wire*. De todos los tipos de modelado posibles, se ha optado por realizar un modelado basado en “caja negra”. Este proceso, por tanto, se realizará estudiando las señales de control de las que dispone el coche y su efecto sobre las distintas variables de salida. En las siguientes secciones se detallará el proceso y resultados obtenidos para cada sistema.

4.2. Sistema Steer-by-wire

El sistema Steer-by-wire se relaciona con la posición del mecanismo piñón-cremallera del vehículo. El mecanismo del vehículo originalmente era un mecanismo de dirección convencional con un motor auxiliar, no obstante tras la rotura del mismo, se optó por la implementación de un sistema *by-Wire*. En la figura 4.1, se muestra el acoplamiento del motor con el mecanismo tradicional. El actuador del sistema es el motor maxon, sección 3.2.5, el cual recibe la tensión de control a través de la etapa de potencia. Esta etapa de potencia convierte la señal PWM en su correspondiente proporción de la tensión de alimentación, 24V en CC. La realimentación del lazo de control se realiza mediante un codificador incremental y uno absoluto, sección 3.2.6. El sensor principal es el codificador incremental por su alta resolución mientras que el codificador absoluto se emplea para referenciar inicialmente la posición del mecanismo. La señal de control del sistema se considerará la tensión de salida de la etapa de potencia, tensión en los bornes del motor. Como información añadida, la etapa de potencia del sistema dispone de un sensor de corriente que permite medir el consumo del motor.



Figura 4.1. Mecanismo físico de la dirección.

A continuación, en la figura 4.2, se muestra en forma de diagrama de bloques los elementos que intervienen en el sistema, así como el flujo de información del sistema.

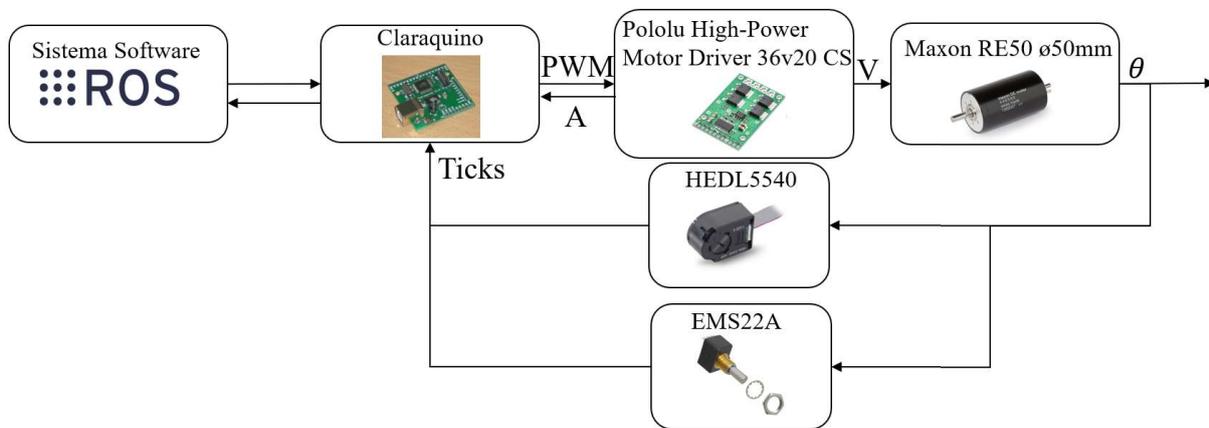


Figura 4.2. Diagrama de bloques del sistema *Steer-by-Wire*.

Para la identificación de parámetros del sistema, se supondrá un modelo inicial. Se hará un barrido en un margen amplio de los valores a estimar y se evaluará el error cuadrático medio, ecuación 4.1. Se determinarán como parámetros correctos aquellos que hacen mínimo ese error entre la señal real y la estimada.

$$E^2 = \frac{1}{n} \sum_{i=1}^n (Y_{real} - Y_{est})^2 \quad (4.1)$$

4.2.1. Modelo Tensión-Giro

La primera relación que se va a estudiar en este sistema es la que toma como señal de entrada la tensión de control de entrada al motor y como señal de salida la velocidad de giro de este. Esta se considera la relación principal del sistema puesto que otras entradas solo perturban a la dinámica del sistema, pero no pueden generar el movimiento de la posición del mecanismo. Referente a la lectura de señales, se realizará el modelado del motor incorporado al sistema, no obstante, mientras que la velocidad se puede medir a través del incremento de pulsos en los codificadores, la tensión con la que se alimenta el motor no es una medida directa. Se considera que la conversión PWM-V es lineal y por tanto, es una medida proporcional a la tensión de alimentación del motor, 24V. En lo referido a la señal de salida, en [16], se comprobó que el ángulo de Ackermann tiene una respuesta lineal en su rango de operación con un valor máximo $\delta = 32,5^\circ$. Con la configuración actual de sensores, el rango máximo de *ticks* que se produce en el codificador absoluto es de 1138, medido experimentalmente. Por lo tanto, la correlación entre la señal del codificador incremental y el ángulo de Ackermann del vehículo queda definida por la ecuación 4.2. Para el caso del codificador absoluto, su ecuación se corresponde con la expresión 4.3. Esta ecuación tiene una ordenada en el origen correspondiente a no colocar el codificador en su punto medio en el momento de fijar el acople en el mecanismo.

$$\delta = 0,000385 \cdot \text{pulsos} \quad (4.2)$$

$$\delta = 0,00571 \cdot \text{pulsos} + 355 \quad (4.3)$$

Se ha considerado oportuno, a fin de obtener un modelo más exacto y completo, realizar el proceso de modelado para situaciones en las que el vehículo se encuentra suspendido, el mecanismo en vacío, y para situaciones en las que el vehículo se encuentra apoyado en el suelo. En esta última situación se aprecia el efecto del peso sobre el rozamiento del mecanismo al moverse. Los perfiles de señales de entrada se han realizado mediante un script de Simulink con el toolbox de ROS, lo que permite replicar los mismos perfiles en distintas circunstancias. A partir de dichos ensayos se ha procedido a evaluar la dinámica del mecanismo.

El primer ensayo estudiado se ha realizado con las ruedas suspendidas y las señales de control a modo rampa, con una pendiente de $0,91 [Vs^{-1}]$. El primer resultado observable, que no se pudo obtener en anteriores trabajos, es la determinación de la zona muerta del sistema. En la figura 4.3 se muestran solapadas las señales de control y la salida, apreciándose claramente la zona muerta del sistema. Para comprobar que no era un fenómeno puntual o un retraso se realizaron múltiples ensayos con variaciones en la tipología y parámetros de las señales de control, confirmándose que el fenómeno se correspondía con la zona muerta del sistema. Este parámetro tiene un valor medio de $1,4723 \pm 0,2458 [V]$. En los mismos ensayos realizados con las ruedas apoyadas, el valor obtenido ligeramente superior como era de esperar por el efecto del rozamiento, $1,9021 \pm 0,4087 [V]$.

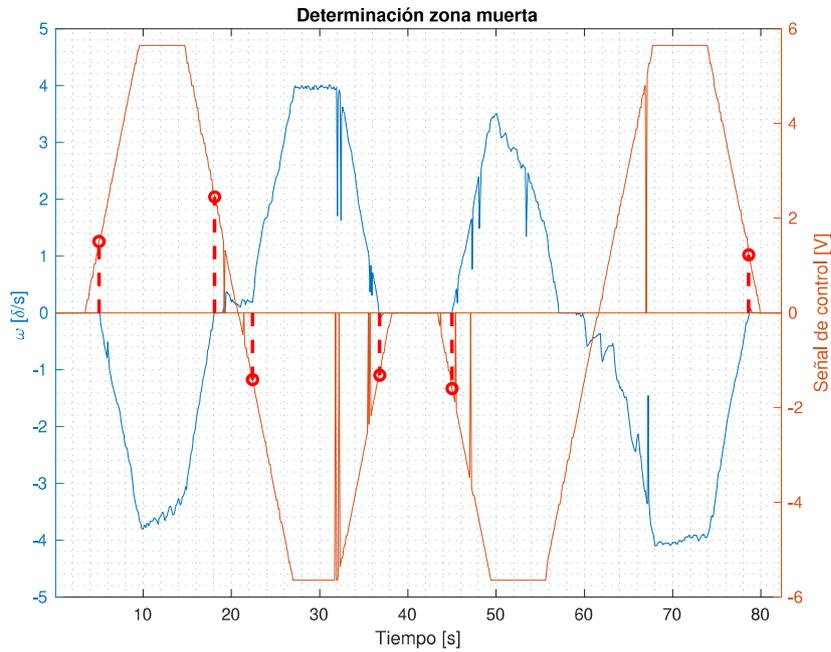


Figura 4.3. Determinación de la zona muerta del mecanismo.

Una vez determinada la principal no linealidad del sistema, se procede a determinar sus parámetros característicos. Se evaluarán dos posibles dinámicas, y se escogerá aquella cuyos parámetros representen mejor el comportamiento del sistema sin comprometer las siguientes etapas de diseño de controladores. Para ello se alcanzará una solución de compromiso entre la exactitud del sistema y su complejidad. Las dos dinámicas estudiadas se corresponden con un sistema de primer orden y un sistema de segundo orden. En el sistema de primer orden, al poseer solo un polo, se determinará la ganancia estática del sistema, k_v , y su constante de tiempo correspondiente, τ_v . En el caso del sistema de segundo orden, el cual poseerá dos polos y ningún cero, se determinará la frecuencia natural del sistema, ω_n , el coeficiente de amortiguamiento, ζ , y la ganancia estática del sistema, k_v . Como es lógico, la ganancia estática que se obtiene es la misma para ambos sistemas pues ante una señal en escalón, ambos modelos deben alcanzar la misma consigna en estacionario.

$$G_v(s) = \frac{k_v}{\tau_v s + 1} \quad (4.4)$$

$$G_v(s) = \frac{k_v \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \quad (4.5)$$

En primer lugar se evaluará el sistema para una dinámica de primer orden cuando el vehículo se encuentra con el mecanismo de dirección suspendido en el aire. Para ello se concatenarán los 14 ensayos realizados con el mecanismo suspendido. Esto es posible ya que la señal de salida, la velocidad de giro, siempre es nula durante un periodo de tiempo al inicio y final de cada ensayo.

En la figura 4.4 se muestran los datos por analizar tras la eliminación de la zona muerta en la señal de control. El barrido de parámetros para la ganancia estática es desde $0 [\delta s^{-1}V^{-1}]$ hasta $-1,5 [\delta s^{-1}V^{-1}]$, con un paso $-0,001 [\delta s^{-1}V^{-1}]$. En el caso de la constante de tiempo, el barrido de parámetros se realiza desde $0 [s]$ hasta $0,3 [s]$, con un paso $0,001 [s]$.

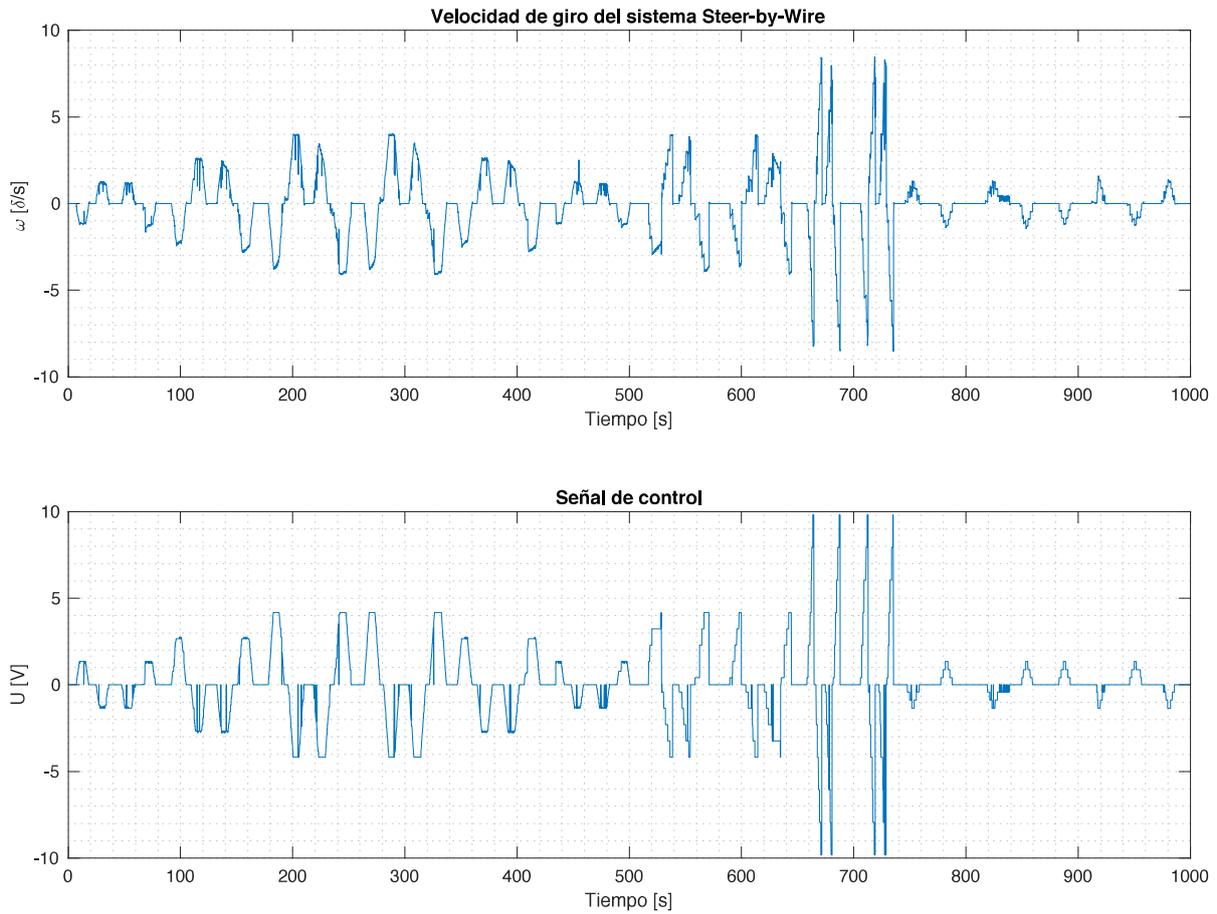


Figura 4.4. Ensayos de giro con el mecanismo operando en vacío.

Como resultado del análisis, se ha obtenido que los parámetros óptimos para un modelo de primer orden se corresponden con una ganancia estática de $-0,864 [\delta s^{-1}V^{-1}]$ y una constante de tiempo de valor $0,127 [s]$, ecuación 4.6. Para estos parámetros, el error cuadrático medio obtenido es de $0,1199 [\delta^2 s^{-2}]$, lo que se corresponde con una exactitud del 80,65% respecto de los datos experimentales. En la figura 4.5a se muestra gráficamente los errores para cada configuración de parámetros planteados. En la figura 4.5b se muestra la vista en planta de la figura anterior.

$$G_v(s) = \frac{-0,864}{0,127s + 1} [\delta s^{-1}V^{-1}] \quad (4.6)$$

Una vez definido el modelo de primer orden, se procede a realizar el mismo análisis para un modelo de segundo orden. Como ya se ha comentado anteriormente, en este caso se fijará la ganancia estática, k_v , al valor obtenido en el caso del modelo de primer orden, $-0,864 [\delta s^{-1} V^{-1}]$. El barrido de parámetros para la frecuencia natural del sistema es desde $5,00 [rad s^{-1}]$ hasta $-20,00 [rad s^{-1}]$, con un paso $0,001 [rad s^{-1}]$. En el caso del factor de amortiguamiento, el barrido se realiza desde $0,1 [s]$ hasta $1,5 [s]$, con un paso $0,001 [s]$. Los parámetros óptimos obtenidos se corresponden con un factor de amortiguamiento de $0,592$ y una frecuencia natural de $11,489 [rad s^{-1}]$, ecuación 4.7. Para estos parámetros, el error cuadrático medio obtenido es de $0,1044 [\delta^2 s^{-2}]$, lo que equivale a una exactitud del $81,95\%$. En la figura 4.6a se muestra gráficamente los errores para cada configuración de parámetros planteados. En la figura 4.6b se muestra la vista en planta de la figura anterior.

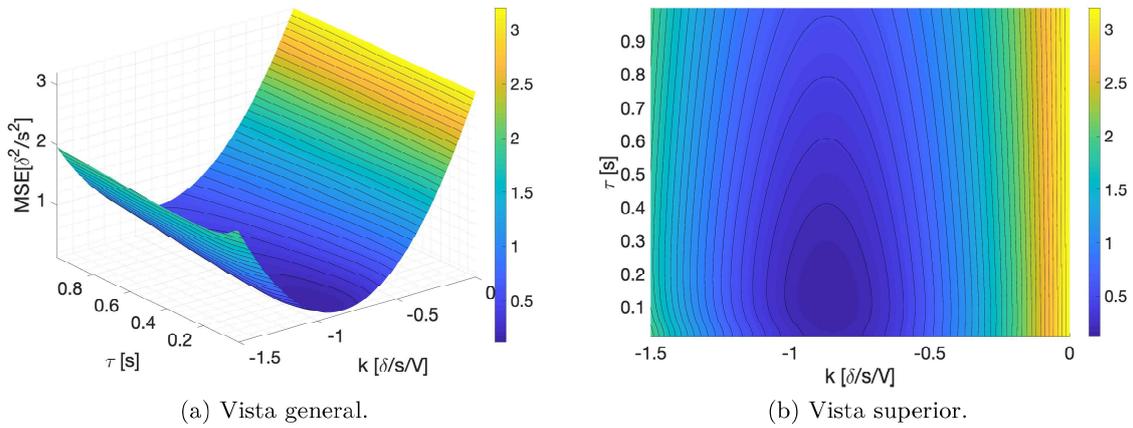


Figura 4.5. Error en función de los parámetros para un modelo de primer orden.

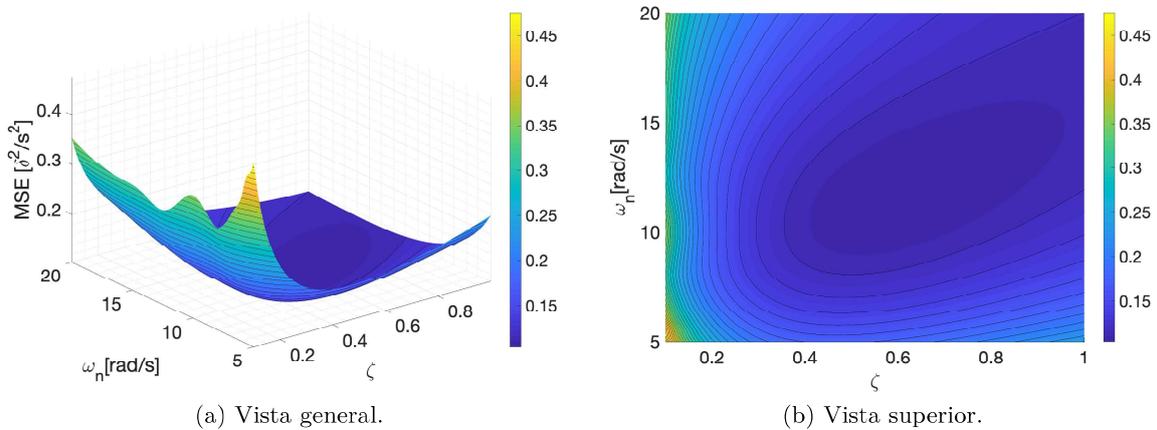


Figura 4.6. Error en función de los parámetros para un modelo de segundo orden.

$$G_v(s) = \frac{-114,0455}{s^2 + 13,6030s + 131,9971} [\delta s^{-1} V^{-1}] \quad (4.7)$$

Una vez definido el sistema para la situación de vacío, se procede a realizar el mismo proceso para el caso en el que el vehículo se encuentra en contacto con el suelo. Se han realizado nuevamente 14 ensayos con los mismos perfiles que los empleados en el caso del vehículo suspendido. En la figura 4.7 se muestran los datos por analizar tras la eliminación de la zona muerta de $1,9021 \pm 0,4087 [V]$.

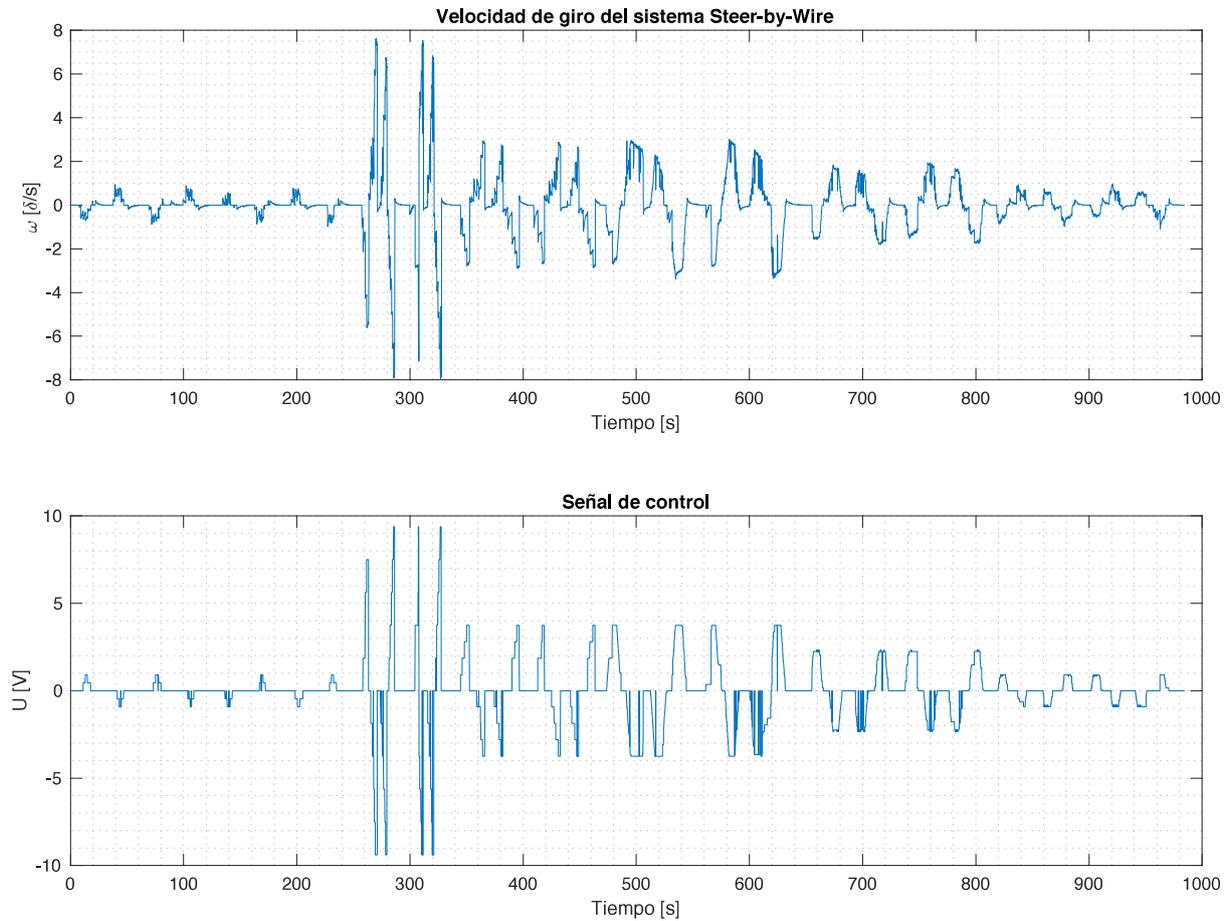


Figura 4.7. Ensayos de giro con el mecanismo operando sobre el suelo.

En primer lugar se determinará el modelo de primer orden que mejor se ajusta al sistema. El barrido de parámetros para la ganancia estática es desde $0 [\delta s^{-1} V^{-1}]$ hasta $-1,0 [\delta s^{-1} V^{-1}]$, con un paso $-0,001 [\delta s^{-1} V^{-1}]$. En el caso de la constante de tiempo, el barrido de parámetros se realiza desde $0 [s]$ hasta $,5 [s]$, con un paso $0,001 [s]$. Los parámetros óptimos obtenidos se corresponden con una ganancia estática de $-0,738 [\delta s^{-1} V^{-1}]$ y una constante de tiempo de $0,115 [s]$, ecuación 4.8.

Para estos parámetros, el error cuadrático medio obtenido es de $0,1252 [\delta^2 s^{-2}]$, lo que se corresponde con una exactitud del $70,65\%$. En la figuras 4.8a y 4.8b se muestran gráficamente los errores para cada configuración de parámetros planteados.

$$G_v(s) = \frac{-0,738}{0,115s + 1} [\delta/s \cdot V] \quad (4.8)$$

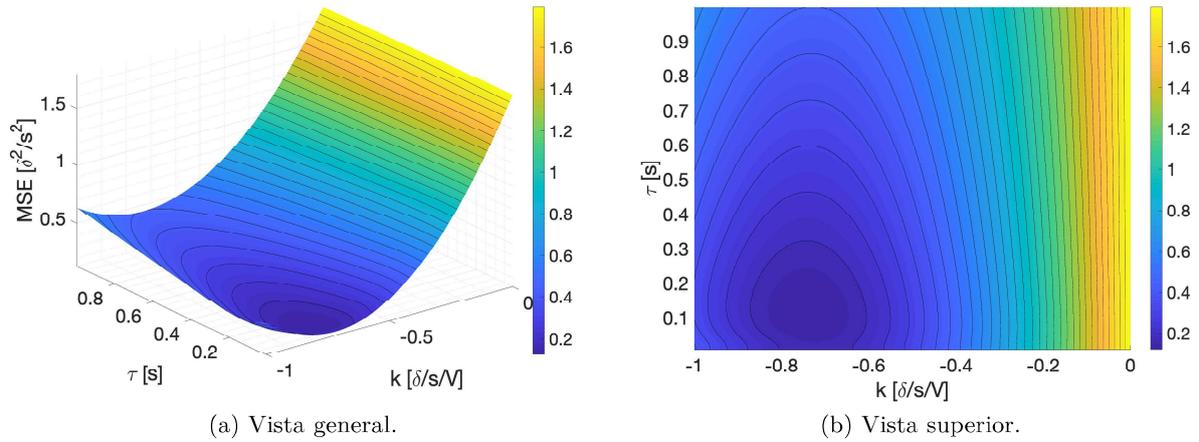


Figura 4.8. Error en función de los parámetros para un modelo de primer orden.

En el caso de la estimación del modelo de segundo orden, el barrido de la frecuencia natural del sistema se realiza entre $5,00 [rad s^{-1}]$ hasta $-20,00 [rad s^{-1}]$, con un paso $0,001 [rad s^{-1}]$. En el caso del factor de amortiguamiento, el barrido abarca desde $0,1$ hasta $1,5$, con un paso $0,001$. Los resultados obtenidos muestran que el valor óptimo para la frecuencia natural del sistema es $11,412 [rad s^{-1}]$ mientras que el factor de amortiguamiento se corresponde con el valor $0,536$, ecuación 4.9. Para estos valores, el error obtenido es de $0,1125 [\delta^2 s^{-2}]$, lo que se corresponde con una exactitud del $74,97\%$.

$$G_v(s) = \frac{-96,1125}{s^2 + 12,2337s + 130,2337} [\delta s^{-1} V^{-1}] \quad (4.9)$$

En la tabla 4.1 se reúnen a modo de resumen todos los datos obtenidos en el modelado de este sistema. De los resultados obtenidos, además del modelo del sistema, se aprecian otros fenómenos al comparar el sistema en vacío y en suelo. Al igual que ocurría con el valor de la zona muerta, el rozamiento al operar sobre el suelo provoca que se reduzca ligeramente el valor de la ganancia estática y de la frecuencia natural del sistema. Al considerarse que el vehículo siempre se va a operar sobre una superficie donde el rozamiento influirá en la dinámica, se estudiarán como posibles modelos aquellos que se han obtenido de los ensayos realizados sobre el suelo.

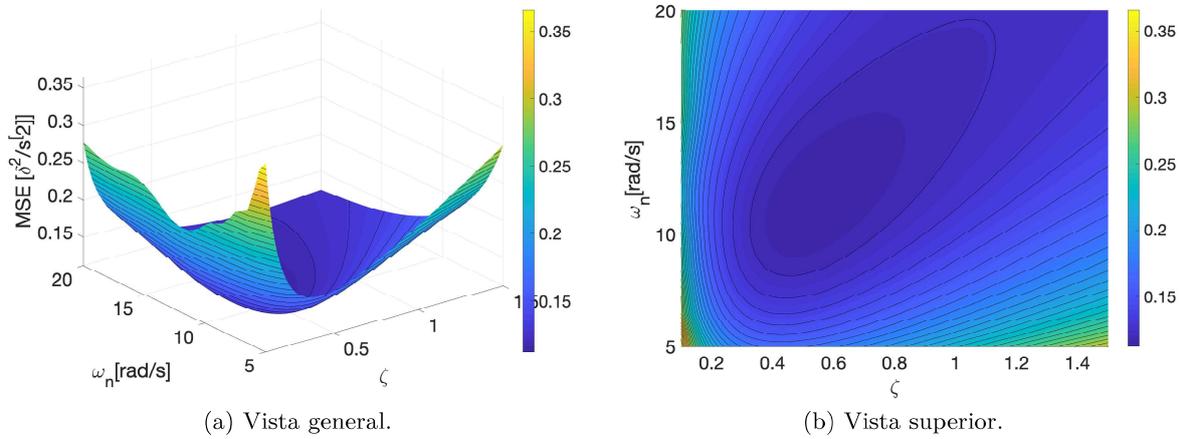


Figura 4.9. Error en función de los parámetros para un modelo de segundo orden.

Ensayo	Parámetro	Valor	Unidades
Vacío. Primer Orden	Ganancia estática	-0,8640	$\delta s^{-1} V^{-1}$
	Constante de tiempo	0,1270	s
	Exactitud	80.65	%
Vacío. Segundo Orden	Ganancia estática	-0,8640	$\delta s^{-1} V^{-1}$
	Coefficiente de amortiguamiento	0,592	-
	Frecuencia natural	11,4890	$rad s^{-1}$
	Exactitud	81.95	%
Suelo. Primer Orden	Ganancia estática	-0,7380	$\delta s^{-1} V^{-1}$
	Constante de tiempo	0,1150	s
	Exactitud	70,45	%
Suelo. Segundo Orden	Ganancia estática	-0,7380	$\delta s^{-1} V^{-1}$
	Coefficiente de amortiguamiento	0,536	-
	Frecuencia natural	11,4120	$rad s^{-1}$
	Exactitud	74,97	%

Tabla 4.1. Parámetros característicos del modelado del sistema de dirección del vehículo.

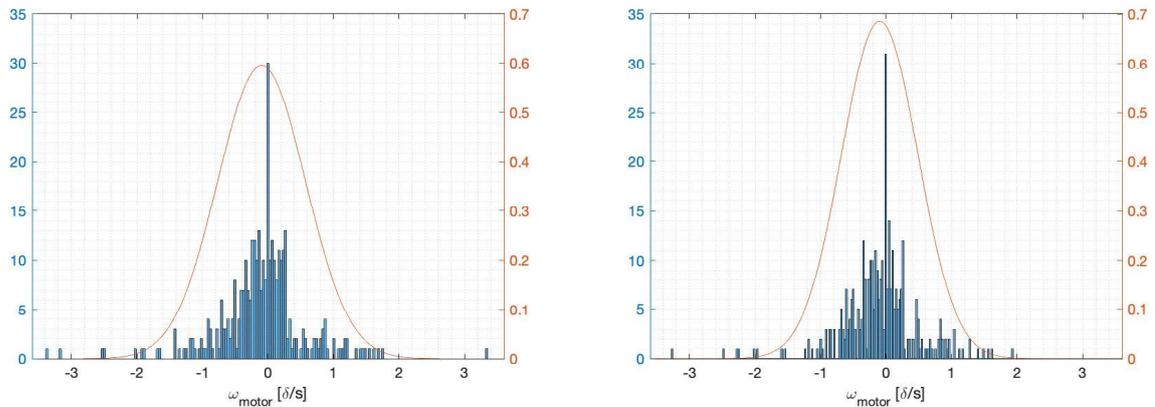
Comparando los errores obtenidos entre los modelos obtenidos de primer y segundo orden, se observa que la diferencia es mínima y casi despreciable, un 4,52%. Con el fin de evaluar mejor el modelo más adecuado para el diseño del controlador, se realiza un nuevo ensayo de validación. Los resultados de esta validación se muestran en la figura 4.11. En este caso, la dinámica que se observa no es tan homogénea como en los ensayos empleados para el modelado. Esto evidencia la presencia de otras no linealidades en el mecanismo que no se han considerado en el modelado. Evaluando objetivamente los datos obtenidos en la validación, el error cuadrático medio para el sistema de primer orden alcanza un valor de 0,5603 $[\delta^2 s^{-2}]$. En el caso del modelo de segundo orden, el valor del error cuadrático medio es de 0,3923 $[\delta^2 s^{-2}]$.

Tras un ajuste manual en el valor de la zona muerta, al observar un retraso en las señales estimadas, se ha conseguido una mejora en la exactitud de los resultados. El modelo de primer orden reduce su error a $0,4569 [\delta^2 s^{-2}]$ mientras que el de segundo orden a $0,3472 [\delta^2 s^{-2}]$. Referido al porcentaje de precisión de ambos modelos, el de primer orden alcanza una exactitud del 79,92% y el de segundo orden una exactitud del 82,44%.

En consecuencia, se determina que el mejor modelo es el de segundo orden, ecuación 4.11, considerando una zona muerta de $1,4723 [V]$, presentando un error medio de $-0,0979\delta/s$ y una desviación estándar de $-0,5819\delta/s$. En las figuras 4.10a y 4.10b se muestran los histogramas y distribución de errores para los dos modelos evaluados. La evaluación se ha desarrollado en el ensayo mostrado en la figura 4.11. En la tabla 4.2 se muestran los resultados estadísticos de los errores de cada modelo, donde se aprecia nuevamente la mayor precisión del modelo de segundo orden.

Modelo	Error medio	Desviación estándar
Primer Orden	$-0,0980 [\delta s^{-1}]$	$0,6697 [\delta s^{-1}]$
Segundo Orden	$-0,0979 [\delta s^{-1}]$	$0,5819 [\delta s^{-1}]$

Tabla 4.2. Estudio de errores en los modelos propuestos para la velocidad de giro.



(a) Histograma del error del modelo de primer orden. (b) Histograma del error del modelo de segundo orden.

Figura 4.10. Evaluación de errores de los modelos obtenidos.

Finalmente, se procede a discretizar el modelo que representa a este sistema. Para la realización de este proceso, se aplica la transformada Z a la ecuación 4.12 junto con la adición de un Mantenedor de Orden Cero, M.O.C. La transformación se realizará para un muestreo de $0,1s$. Este tiempo de muestreo se toma cumpliendo el teorema del muestreo (Nysquist-Shannon), el cual establece que la frecuencia de muestreo debe ser, como mínimo, el doble de la frecuencia más alta a la que la señal posee energía, ecuación 4.10.

$$\omega_n = \frac{2\pi}{T} > 2\omega_0 \rightarrow T < \frac{\pi}{\omega_0} \rightarrow T < 0,275[s] \quad (4.10)$$

$$G_{12}(s) = \frac{Y_{12}(s)}{U_1(s)} = \frac{-96,1125}{s^2 + 12,2337s + 130,2337} [\delta s^{-1}V^{-1}] \quad (4.11)$$

$$G_{12}(Z) = \frac{Y_{12}(Z)}{U_1(Z)} = \frac{-0,3008z^{-1} - 0,1974z^{-2}}{1 - 0,6192z^{-1} + 0,2942z^{-2}} [\delta s^{-1}V^{-1}] \quad (4.12)$$

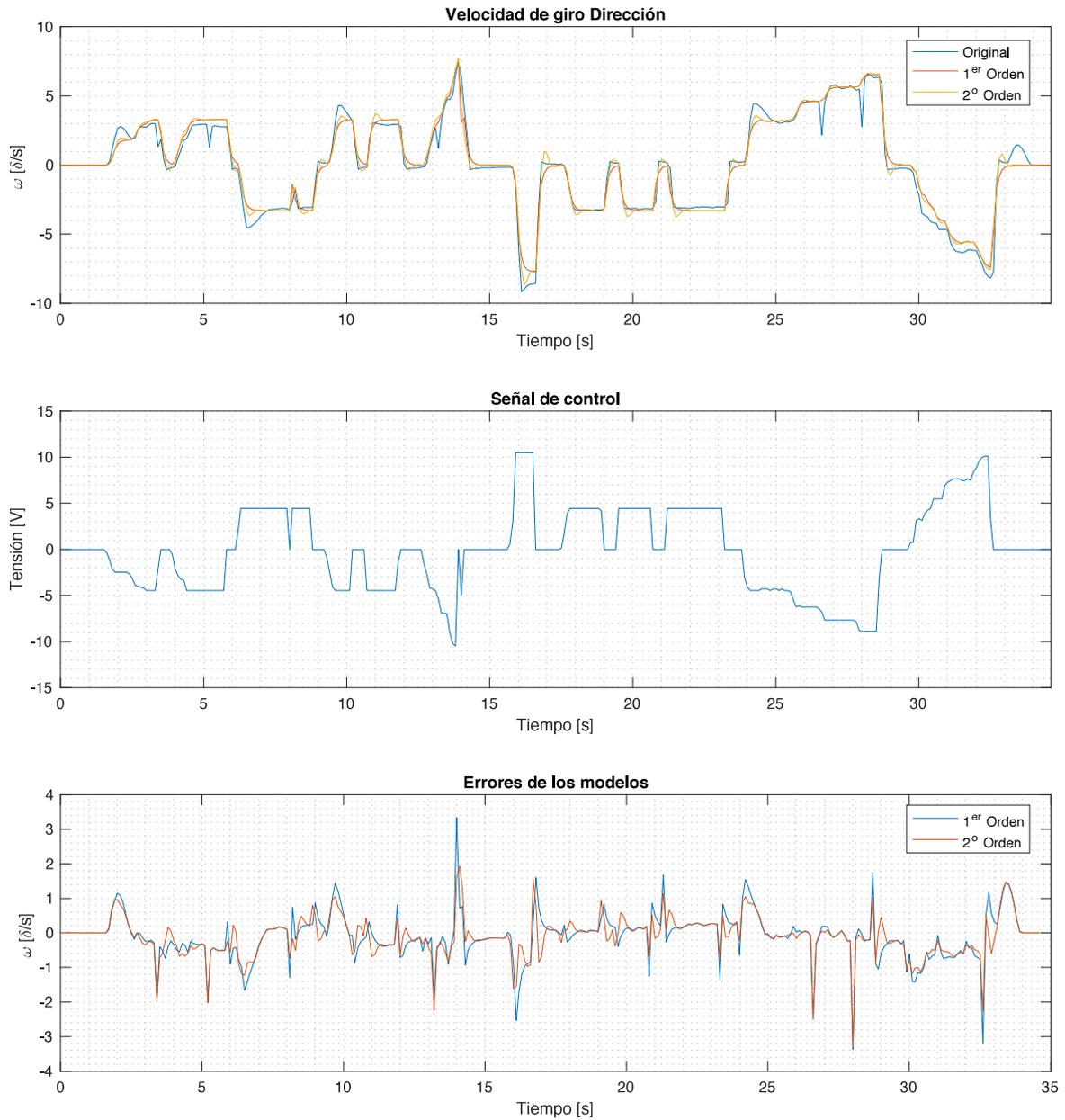


Figura 4.11. Validación de los modelos obtenidos.

4.2. SISTEMA STEER-BY-WIRE

La última observación de la que se deja constancia para un mayor conocimiento del sistema, se observa en la figura 4.12. En ella se aprecia que el valor de la corriente satura en su nivel inferior y la salida del sistema deja de responder a las señales de control. Se ha comprobado experimentalmente que en esas circunstancias la tensión de alimentación de la etapa de potencia se reduce a un valor despreciable. Se aprecia que la segunda y tercera vez que ocurre tienen la misma duración, 7, 70 [s]. Este fenómeno deberá de tenerse en cuenta cuando se realice el diseño del controlador para evitar incidencias mayores en su implementación en el vehículo real.

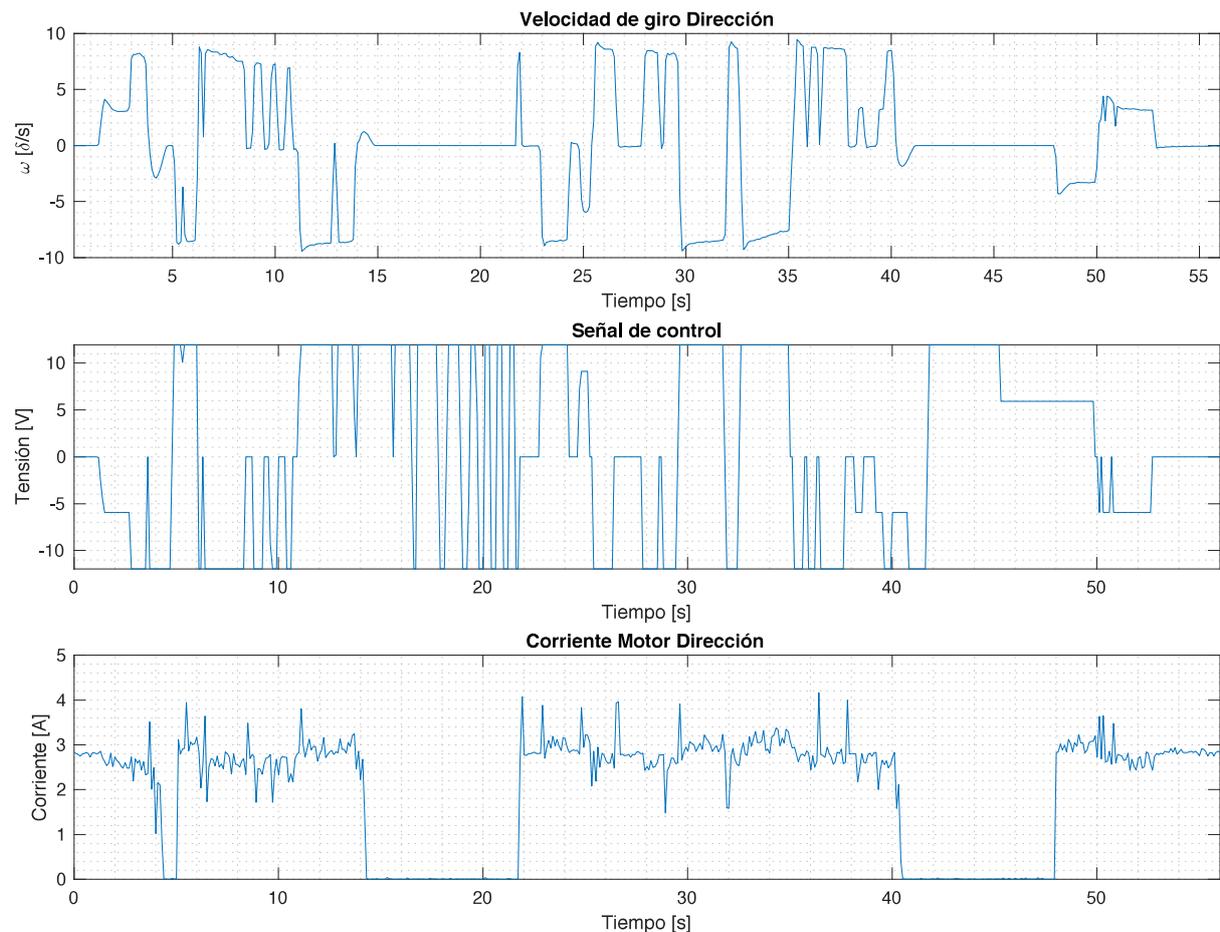


Figura 4.12. Error en la alimentación del actuador del sistema.

4.2.2. Modelo Giro-Corriente

En este caso, se modela el consumo de corriente por parte del motor del sistema. Se considerará como señal de salida la señal generada por el sensor de corriente del que dispone la etapa de potencia, y como señal de control, la velocidad de giro del motor. Sobre este sistema no se pretende establecer un sistema de control, sino obtener una primera aproximación al consumo que se produce en este dispositivo y evitar que se alcancen niveles peligrosos de corriente.

Tras una primera observación de los datos obtenidos en los ensayos realizados, se observan dos factores importantes a considerar antes de realizar el modelado. Por un lado, no se observa zona muerta en los datos analizados a diferencia del modelo anterior. Por otra parte, la dinámica observada se corresponde con un sistema de primer orden, o un segundo orden sobreamortiguado, lo que permitiría la aproximación a un modelo de primer orden. Siguiendo el mismo procedimiento que para el caso de la relación tensión-giro, haciendo uso del error cuadrático medio, se estimarán los parámetros de un sistema de primer orden que mejor se ajusten a los resultados experimentales. En la figura 4.13 se muestran los ensayos analizados para el caso del mecanismo suspendido. Se debe tener en cuenta que el signo negativo de la salida no indica que devuelva energía, sino el cambio de sentido en el giro del motor.

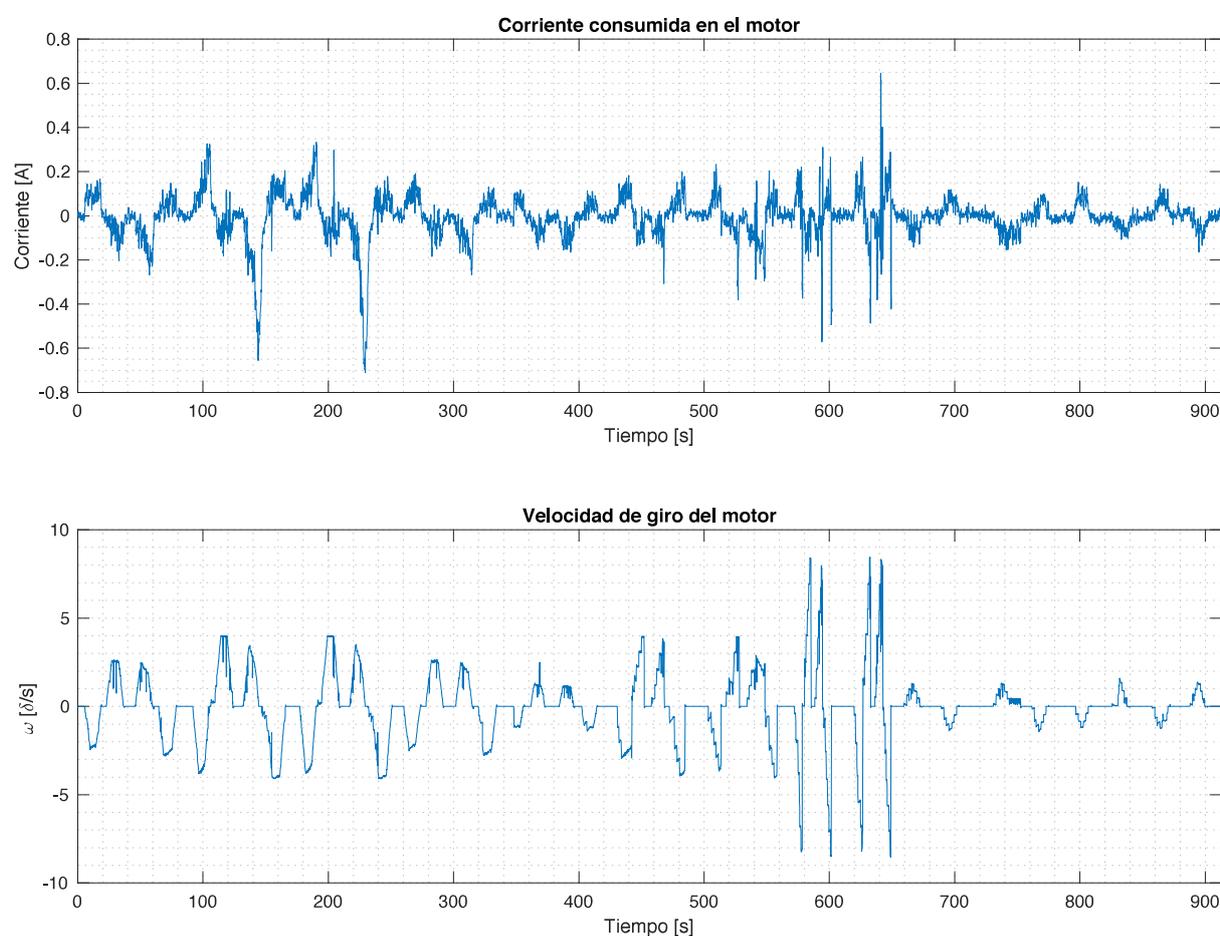


Figura 4.13. Ensayo de corriente con el mecanismo operando en vacío.

El sistema de primer orden que se propone será de un solo polo y ningún cero. Se determinará la ganancia estática del sistema, k_c , y su constante de tiempo correspondiente, τ_c . El barrido de parámetros para la ganancia estática comienza en $0 [As\delta^{-1}]$ hasta $-0,2 [As\delta^{-1}]$ con un paso de $-0,0001 [As\delta^{-1}]$.

El barrido para la constante de tiempo se realiza desde 0 [s] hasta 10 [s] con un paso de 0,001 [s]. Los errores cuadráticos medios obtenidos para cada parámetro se observan en las figuras 4.14a y 4.14b. Los parámetros que consiguen minimizar el error hasta un valor de 0,086 [A] son una constante de tiempo de 0,521 [s] y una ganancia estática de $-0,0384 [As\delta^{-1}]$, ecuación 4.13. Este error se corresponda con una exactitud del 68,58 %, inferior al modelo del sistema de velocidad de giro, pero suficiente para la finalidad de este sistema.

$$G_c(s) = \frac{-0,0384}{0,512s + 1} [As/\delta] \quad (4.13)$$

Se repite el mismo procedimiento para el caso en el que el vehículo se encuentra apoyado sobre el suelo. En este caso, se realiza el barrido de parámetros para la ganancia estática comienza en 0 [$As\delta^{-1}$] hasta $-0,5 [As\delta^{-1}]$ con un paso de $-0,0001 [As\delta^{-1}]$. El barrido para la constante de tiempo se realiza desde 0 [s] hasta 10 [s] con un paso de 0,001 [s]. Los errores cuadráticos medios obtenidos para cada parámetro se observan en las figuras 4.15a y 4.15b. Los parámetros que alcanzan un mínimo en el error cuadrático medio son una ganancia estática $-0,1177 [As\delta^{-1}]$ y una constante de tiempo de 0,678 [s], ecuación 4.14. Estos parámetros alcanzan un error mínimo de valor 0,083[A]. En este caso, aunque el error es superior ligeramente superior, la exactitud del modelo a alcanzado un valor del 86,70 %. En la tabla 4.3 se reúnen los valores de los parámetros obtenidos para modelos de primer orden. En ella se observa un fenómeno interesante que era de esperar, que al apoyar el vehículo, con la presencia de rozamiento externo, la ganancia del modelo aumenta, es decir, la corriente que consume le motor para realizar el mismo movimiento aumenta y presenta una dinámica más lenta.

$$G_c(s) = \frac{-0,1177}{0,678s + 1} [As/\delta] \quad (4.14)$$

Como es un modelo con el que se busca obtener mayor información del sistema, se busca un modelo de mayor orden para mejorar el índice de exactitud. Finalmente, el modelo más exacto se alcanza con un tercer orden con tres polos y dos ceros, ecuación 4.16. En la figura 4.17 se puede observar el ensayo en el cual se valida el modelo establecido. Al igual que se realizó para el estudio de la dinámica de giro del sistema, se realizará un breve estudio estadístico sobre los errores obtenidos en validación para los modelos propuestos. Los resultados obtenidos se muestran en la tabla 4.4. En las figuras 4.16a y 4.16b se muestran los histogramas de errores para cada modelo. Para el modelo de primer orden el error medio obtenido en este caso es de $-0,0445 \pm 0,0071 [As\delta^{-1}]$. Este modelo se implementará en las estimaciones que se hagan con el vehículo en funcionamiento por su simpleza. No obstante, para implementar una estimación del nivel de corriente que se demanda en función de la velocidad de giro, se trabajará con el modelo obtenido según la ecuación 4.15, el cual presenta la mitad de error pero con una ligera desviación estándar, $-0,0220 \pm 0,0077 [As\delta^{-1}]$.

$$G_{13}(s) = \frac{Y_{13}(s)}{Y_{12}(s)} = \frac{k_c}{\tau_c s + 1} = \frac{-0,1177}{0,678s + 1} [As\delta^{-1}] \quad (4.15)$$

$$G_{13,sim}(s) = \frac{-4,2705s^2 - 1,5041s - 0,1279}{s^3 + 5,828s^2 + 10,86s + 0,3386} [As\delta^{-1}] \quad (4.16)$$

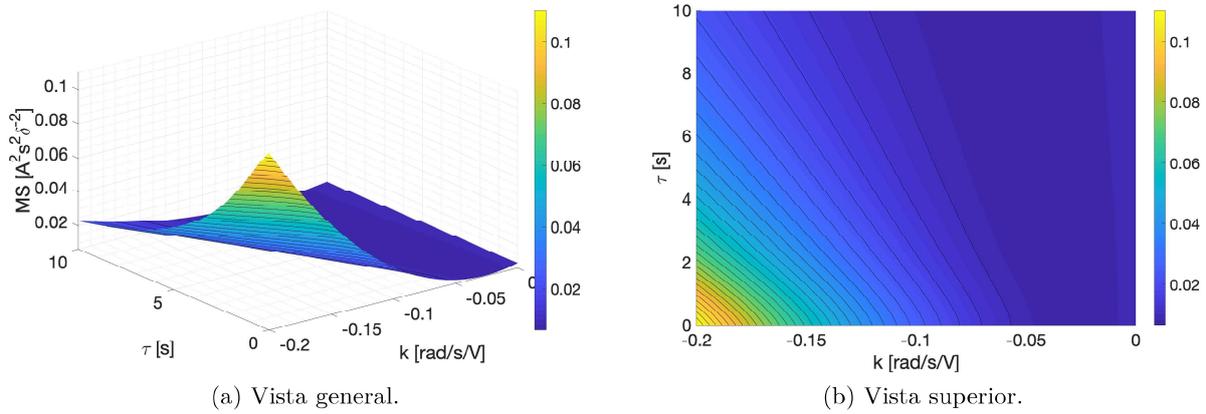


Figura 4.14. Error en función de los parámetros para un modelo de primer orden para vacío.

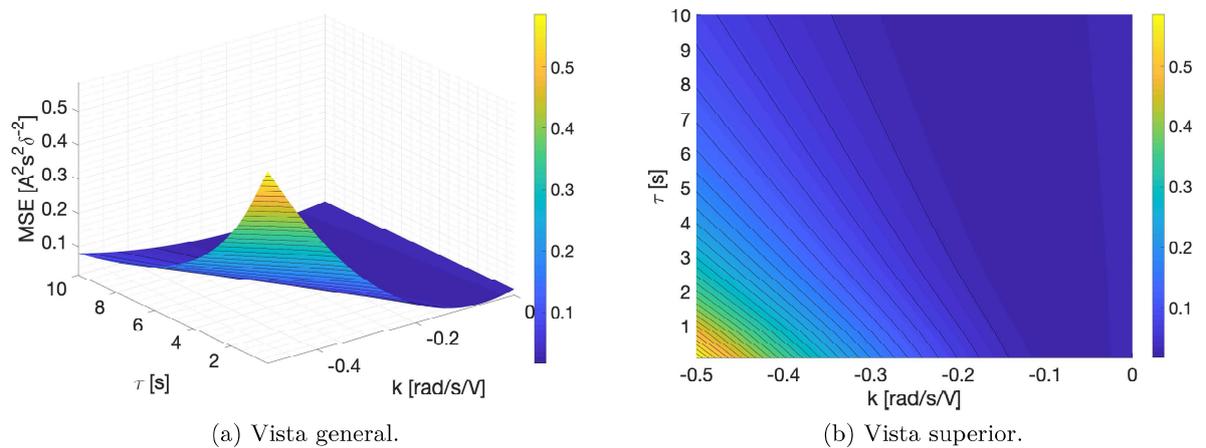


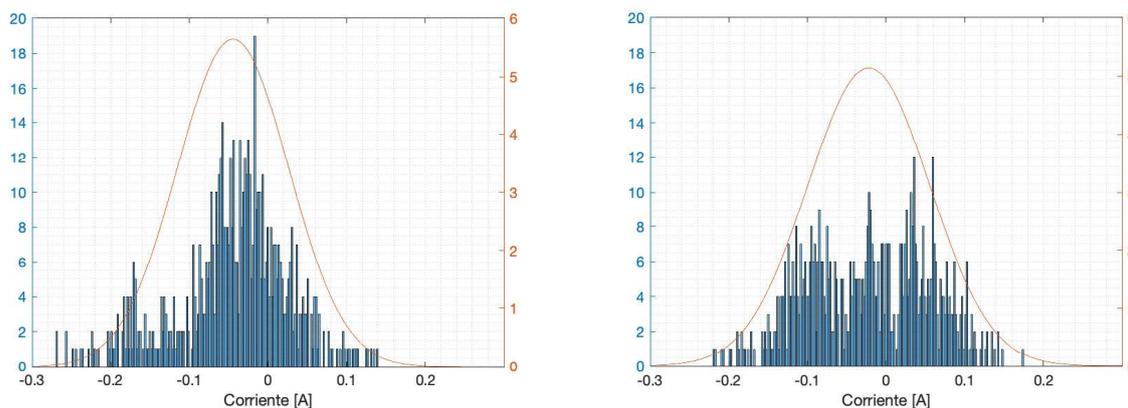
Figura 4.15. Error en función de los parámetros para un modelo de primer orden para suelo.

Ensayo	Parámetro	Valor	Unidades
Vacío. Primer Orden	Ganancia estática	-0,0384	As/δ
	Constante de tiempo	0,512	s
	Exactitud	68,58	%
Suelo. Primer Orden	Ganancia	-0,1177	As/δ
	Constante de tiempo	0,678	s
	Exactitud	86,70	%

Tabla 4.3. Parámetros característicos del sistema de corriente para un modelo de primer orden.

Modelo	Error medio	Desviación estándar
Primer Orden	-0,0445 [$As\delta^{-1}$]	0,0707 [$As\delta^{-1}$]
Tercer Orden	-0,0220 [$As\delta^{-1}$]	0,0767 [$As\delta^{-1}$]

Tabla 4.4. Estudio de errores en los modelos del consumo de corriente.



(a) Histograma del error del modelo de primer orden. (b) Histograma del error del modelo de tercer orden.

Figura 4.16. Evaluación de errores de los modelos obtenidos.

Finalmente, se procede a discretizar la f.d.t. seleccionada para poder integrar un estimador del consumo del motor en el microcontrolador del sistema. Para la realización de este proceso se aplica la transformada Z a la ecuación 4.15, a la cual se añade un M.O.C. El tiempo de muestreo empleado para esta transformación es de 0,1s al igual que el resto de transformaciones realizadas, ya que será el periodo de operación de los microcontroladores empleados.

$$G_{13}(Z) = \frac{Y_2(Z)}{Y_1(Z)} = \frac{-0,01614z^{-1}}{1 - 0,8629z^{-1}} [As/\delta] \quad (4.17)$$

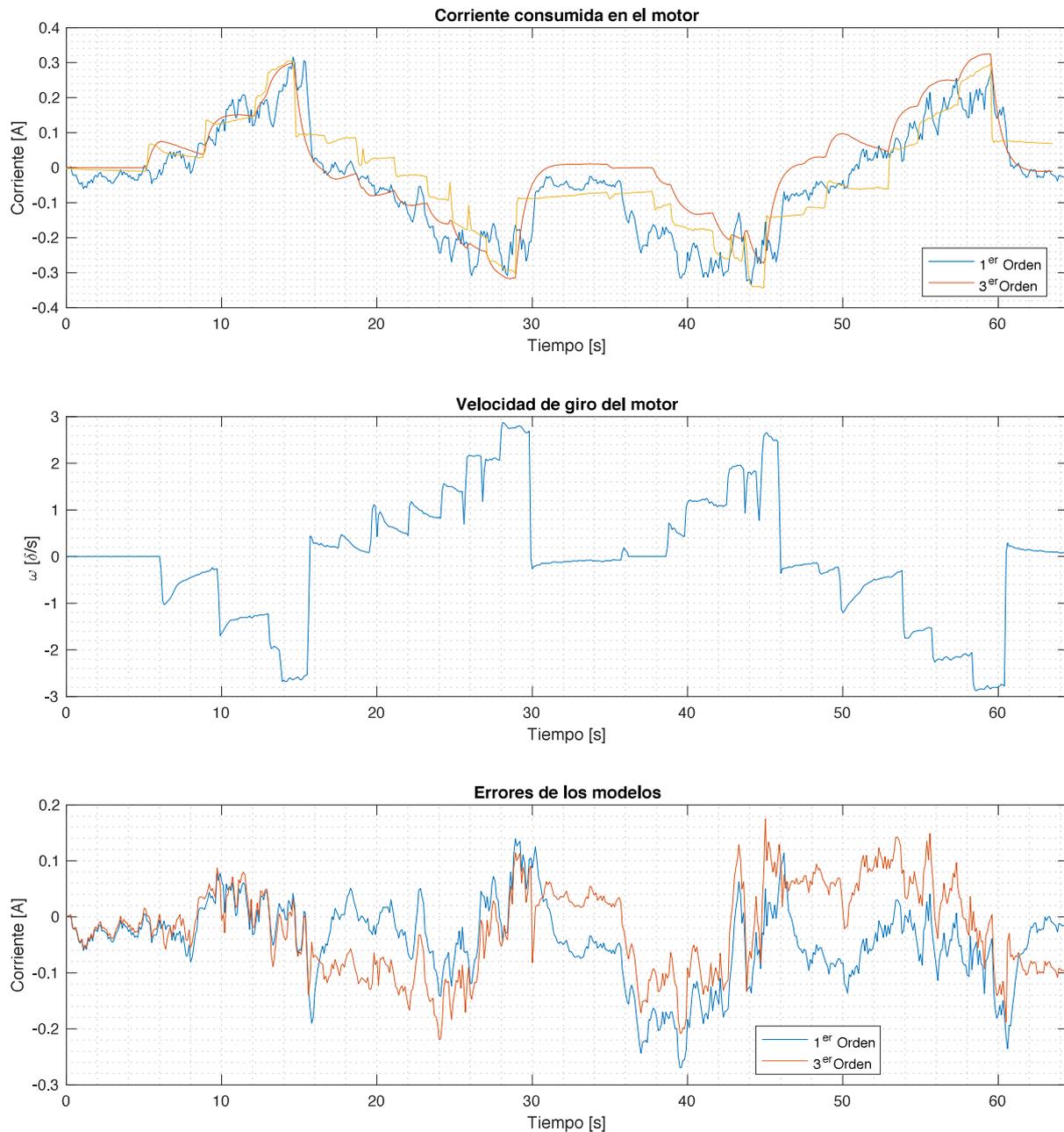


Figura 4.17. Validación modelo de corriente en función de la velocidad de giro del mecanismo.

4.3. Sistema Brake-by-wire

El sistema *Brake-by-Wire* se trata de la instalación más reciente del vehículo y es el encargado de actuar sobre el sistema de frenado del vehículo. Consiste en un motor similar al empleado en la dirección del vehículo que al girar un disco arrastra una banda trapezoidal.

4.3. SISTEMA BRAKE-BY-WIRE

Esta banda se encuentra anclada a dicho disco en un extremo y al pedal del freno en el otro. Por lo tanto, este sistema no actúa directamente sobre el sistema de frenada del vehículo, sino más bien sobre el actuador convencional del sistema. En la figura 4.18 se observa la instalación física del sistema, en la parte central el disco acoplado al motor y tras él el motor del sistema de dirección.



Figura 4.18. Actuador físico sistema Brake-by-wire.

El modelado se realizará nuevamente como un modelo de “caja negra”. En la figura 4.19 se muestra tanto el diagrama de bloques del sistema como el flujo de datos que se produce en él. El actuador del sistema se corresponde con un motor eléctrico maxon alimentado con una fuente CC de 24 [V]. Este motor recibe la tensión de control a través de una etapa de potencia *Pololu High-Power Motor Driver 36v20 CS* que transforma una señal de control PWM en su proporcional a la tensión de alimentación. Esta señal de control es generada por un microcontrolador integrado en la tarjeta claraquino. La realimentación se realiza mediante la lectura de pulsos de un codificador incremental acoplado al eje del motor. Finalmente, la referencia de entrada al sistema, tanto en lazo abierto como en lazo cerrado, es generada a nivel software en ROS.

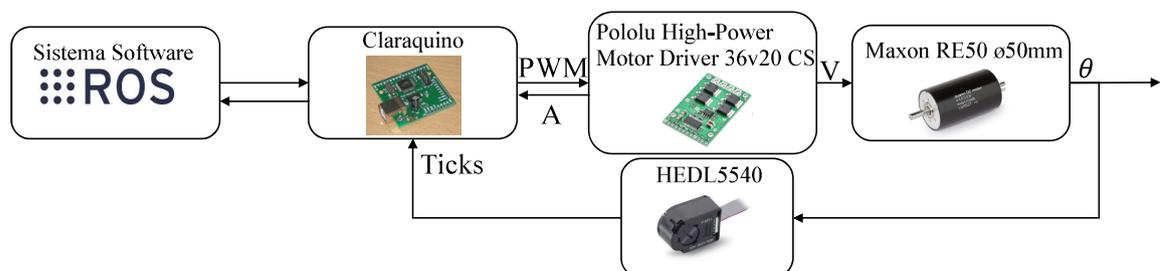


Figura 4.19. Diagrama de bloques del sistema *Brake-by-Wire*.

El objetivo de este sistema es suplir un apoyo al control de velocidad longitudinal del vehículo en los cambios descendentes en las consignas de velocidad. En esos momentos, la inercia del vehículo muestra una fuerte presencia en la dinámica del vehículo, relentizando el cambio de consigna.

El modelado se realizará tomando como señal de control del sistema la tensión que recibe el motor y como salida la velocidad de giro del motor. En la figura 4.20 se muestra un ensayo realizado sobre el sistema de frenado. Durante el ensayo realizado, además de capturar la dinámica característica del sistema, se ha medido experimentalmente el máximo ángulo posible del mecanismo, $12,44^\circ$. Además se observa que la respuesta no lineal del sistema y los picos de corriente que se producen cuando el sistema alcanza su extremo de actuación. Esto permite la implementación de mecanismos de seguridad e incluso de calibración.

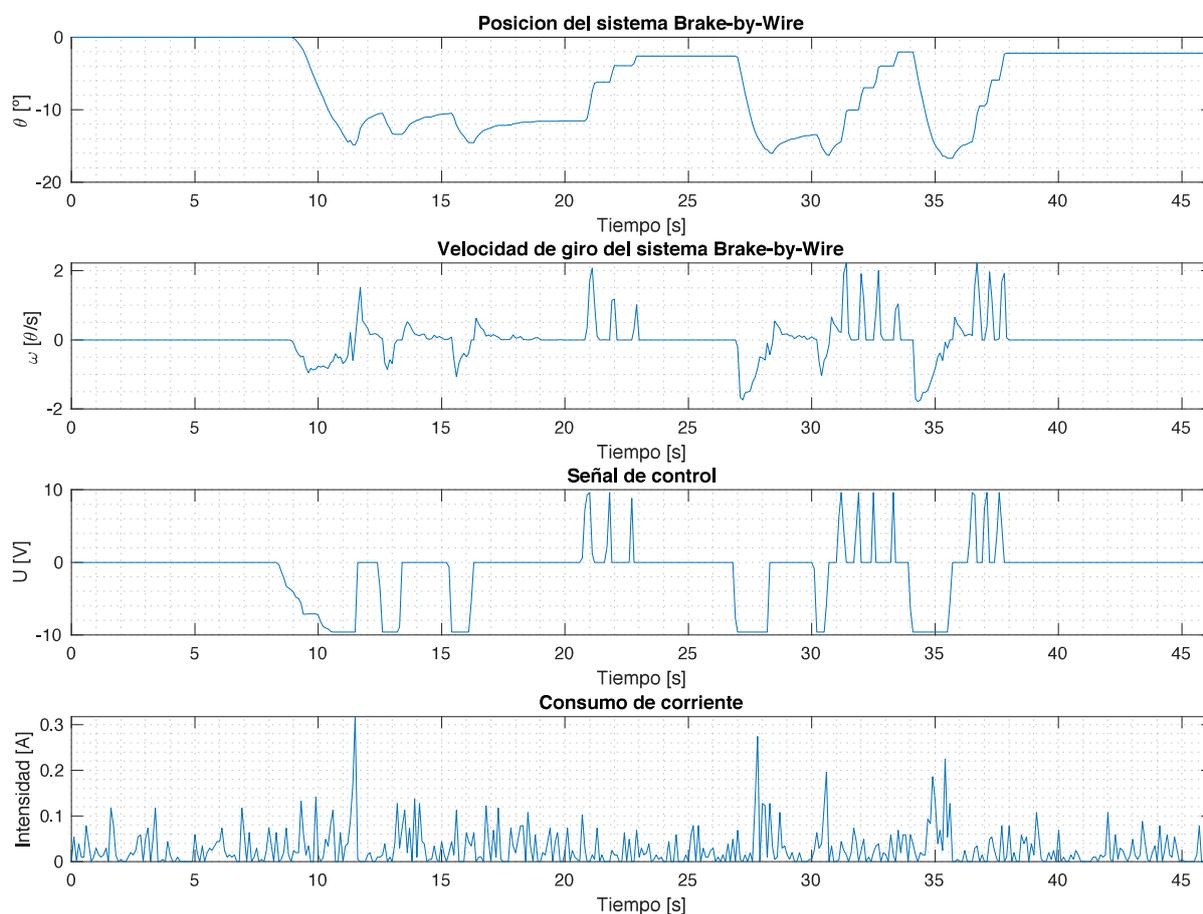


Figura 4.20. Ensayo realizado sobre el sistema *Brake-by-Wire*.

En este sistema, a diferencia del caso del motor acoplado a la dirección, no se dispone de un codificador absoluto que detecte la posición del mecanismo al inicio del funcionamiento. A falta de dicho codificador absoluto, una alternativa es la detección del pico de corriente.

En el momento que se produce dicho evento, conocida la amplitud del recorrido, se desplazaría en sentido contrario hasta el inicio del sistema, 0° . Esto hace que el sistema se coloque en el origen del sistema al arrancar el vehículo. Como ya se ha visto en el estudio bibliográfico, el comportamiento de un sistema de frenado presenta una dinámica no lineal. Esto impide que se pueda realizar la misma técnica de modelado que en el caso de la dirección. Haciendo uso del toolbox de Matlab, *systemIdentification*, se estimarán modelos de mayor complejidad para obtener la mayor exactitud posible sin comprometer la complejidad del sistema. Se han examinado distintos modelos para buscar el mejor ajuste del sistema y los resultados analíticos obtenidos se muestran en la tabla 4.5, siendo los parámetros estudiados el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE) y el error de predicción final (FPE). De todos estos, se han seleccionado los cuatro que mejores resultados han dado.

El mejor resultado se ha obtenido para un modelo dinámico no lineal de *Autoregresión con Variables eXógenas* (ARX) con 10 términos de entrada y salida para la regresión. Este modelo ha alcanzado una exactitud del 63,87% en la validación. Los siguientes modelos son funciones de transferencia lineales, pero a causa del poco rango de operación del sistema, y de la homogeneidad de la dinámica hasta pocos grados antes del límite del mecanismo, se consiguen aproximaciones con gran exactitud. Una f.d.t. con tres polos y dos ceros, ecuación 4.18, se posiciona en segundo lugar por exactitud con un valor de 63,16%. La diferencia de exactitud en la validación es muy poca, lo que permitiría considerar este modelo como aceptable no olvidando las restricciones que permiten la aproximación lineal de un modelo no lineal. En este caso el poco rango de operación y no superar nunca el límite del mecanismo. El siguiente modelo, más distanciado en el índice de exactitud, se corresponde con una f.d.t. de segundo orden con un cero, ecuación 4.19. Este modelo alcanza un valor de exactitud del 59,84%. De la ecuación característica de este modelo se puede determinar que el factor de amortiguamiento del sistema, aproximadamente, es superior a la unidad. Esto hace que el último modelo propuesto se corresponde con la f.d.t. más simple, orden uno con un polo, ecuación 4.20. Este último modelo presenta la exactitud más baja de los tipos estudiados, 46,9%.

$$G(s) = \frac{-2,071^2 + 40,39s + 2,498}{s^3 + 18,84s^2 + 293,9s + 141,4} [\theta/s \cdot V] \quad (4.18)$$

$$G(s) = \frac{1,246s + 0,1426}{s^2 + 8,392s + 5,266} [\theta s^{-1} V^{-1}] \quad (4.19)$$

$$G(s) = \frac{0,1179}{0,0853s + 1} [\theta s^{-1} V^{-1}] \quad (4.20)$$

En la figura 4.21 se muestran los resultados de los cuatro modelos propuestos. En esta, se pueden distinguir las dinámicas características de cada sistema, como la ligera sobreoscilación del modelo de segundo orden y como el modelo de primer orden no presenta dicha sobreoscilación.

Modelo	Exactitud	MSE	RMSE	FPE
ARX No Lineal	63,87 [%]	0,0646 [$\theta^2 s^{-2}$]	0,2542 [θs^{-1}]	0,0665 [θs^{-1}]
Tercer Orden	63,16 [%]	0,0982 [$\theta^2 s^{-2}$]	0,3134 [θs^{-1}]	0,0990 [θs^{-1}]
Segundo Orden	59,84 [%]	0,1244 [$\theta^2 s^{-2}$]	0,3527 [θs^{-1}]	0,1252 [θs^{-1}]
Primer Orden	46,90 [%]	0,1773 [$\theta^2 s^{-2}$]	0,4211 [θs^{-1}]	0,1778 [θs^{-1}]

Tabla 4.5. Estudio de errores en los modelos del consumo de corriente.

Como era de esperar por la cercanía entre los índices de exactitud obtenidos, no se puede apreciar de forma evidente cual se la traza del modelo con mejor desempeño, por lo que objetivamente, el mejor es el ARX No Lineal. Este modelo se implementará en la herramienta de simulación del vehículo completo. En esa herramienta, prima la exactitud a la complejidad, por pequeña que sea la diferencia, pues se busca que permita evaluar de la forma más realista posibles futuros controladores a implementar en el vehículo. No obstante, como modelo para trabajar en futuras etapas de diseño de controladores, se considerará como adecuado la f.d.t. de segundo orden con dos polos y un cero, ecuación 4.21. En la ecuación 4.22 se muestra la f.d.t. discretizada para un tiempo de muestreo de 0,1 [s].

$$G_{22}(s) = \frac{Y_{22}(s)}{U_2(s)} = \frac{1,246s + 0,1426}{s^2 + 8,392s + 5,266} [\theta s^{-1} V^{-1}] \quad (4.21)$$

$$G_{22}(Z) = \frac{Y_{22}(Z)}{U_2(Z)} = \frac{0,08414z^{-1} - 0,08318z^{-2}}{1 - 1,397z^{-1} - 0,4321z^{-2}} [\theta s^{-1} V^{-1}] \quad (4.22)$$

4.4. Sistema Throttle-by-wire

El sistema *Throttle-by-Wire* se relaciona con velocidad longitudinal del vehículo, junto con el sistema *Brake-by-Wire*. En la figura 4.22, se muestra el diagrama de bloques de este sistema. El actuador principal del sistema es el motor principal del vehículo, sección 3.2.4, principal responsable del desplazamiento longitudinal del mismo. Como se puede apreciar en el diagrama, a diferencia de los sistemas anteriores, la realimentación no se realiza por una conexión directa con el resto del sistema. Mientras que el actuador genera el movimiento en las ruedas delanteras, los codificadores se sitúan en las ruedas posteriores, figura 4.24b. La información de estos codificadores se envía directamente a ROS y este se la transmite al microcontrolador. La señal de control en este sistema es generada a nivel software en ROS y se transmite al microcontrolador, el cual la envía a un conversor analógico digital que la transforma en una señal analógica entre 0 [V] y 5 [V]. No obstante, el motor no se puede controlar con dicha tensión, por lo que se requiere la etapa de potencia, sección 3.2.2. Esta etapa incluye un control de potencia que adecua la señal analógica a una señal PWM con niveles de tensión y corriente adecuados.

4.4. SISTEMA THROTTLE-BY-WIRE

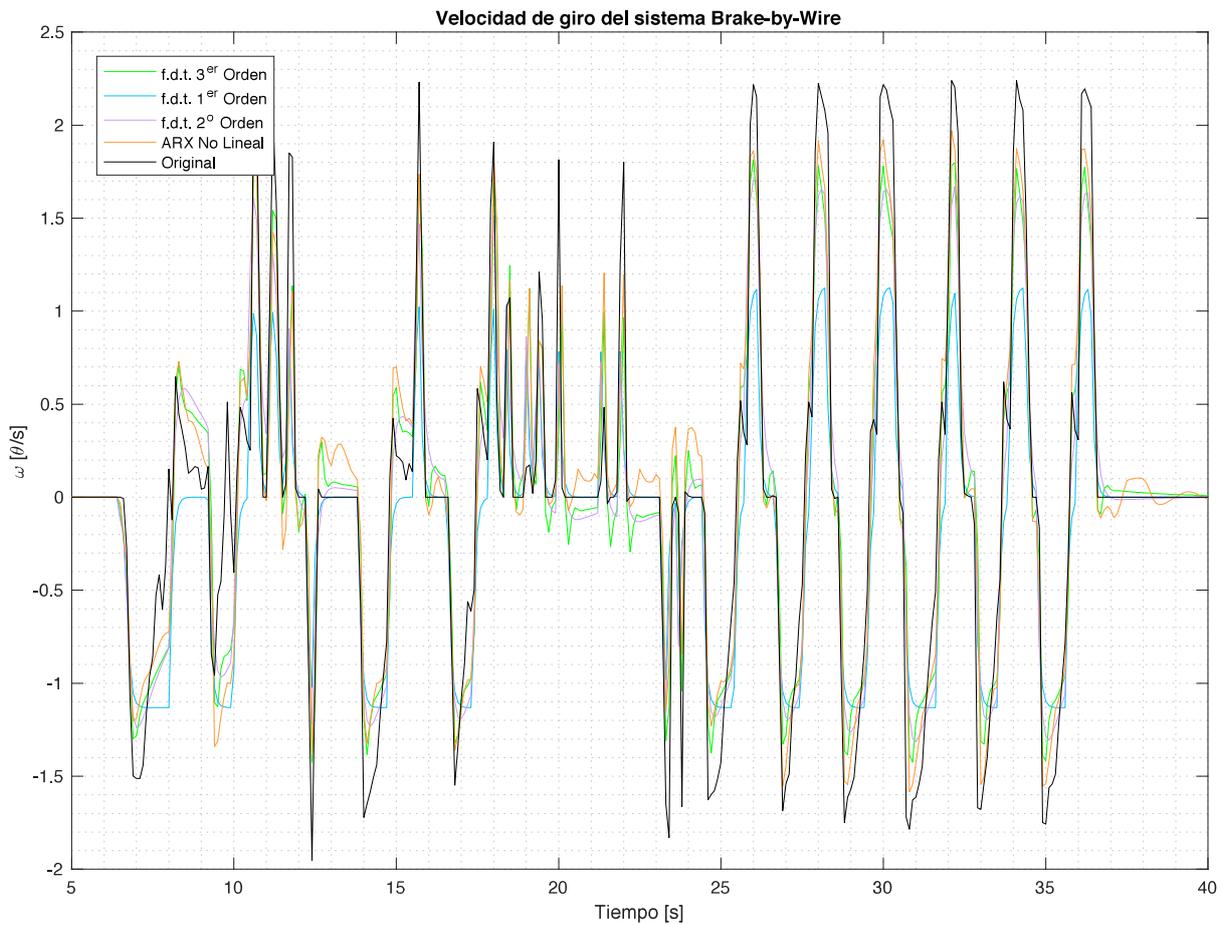


Figura 4.21. Validación de los modelos propuestos para el sistema *Brake-by-Wire*.

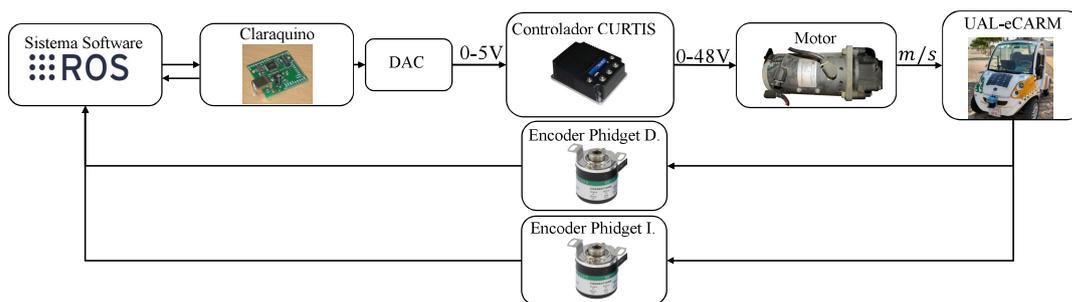


Figura 4.22. Diagrama de bloques del sistema *Throttle-by-Wire*.

Tal como se puede ver en [82], este controlador incluye gran cantidad de parámetros configurables, lo que dan al sistema mayor libertad y versatilidad en el control de la velocidad del vehículo.

La mayor parte de estos parámetros están enfocados a la seguridad y se usan para establecer unos valores máximos y mínimos que pueden adquirir ciertas variables, como la corriente del motor o la velocidad del vehículo. El más relevante desde la perspectiva del modelado del sistema es el que establece la relación entrada-salida del controlador y se denomina *Throttle map*. Este parámetro establece la correlación entre el porcentaje de la señal de control analógica que recibe el controlador, y la señal PWM que recibe el motor, figura 4.23. Esta variable permite la modificación de la linealidad del sistema modificando la linealidad de la respuesta en la etapa de potencia. Según la configuración actual, este parámetro se encuentra en su valor medio para asegurar que la respuesta se corresponda con el modelo real del motor en la mayor medida posible. En el apéndice A se muestra la configuración actual del vehículo.

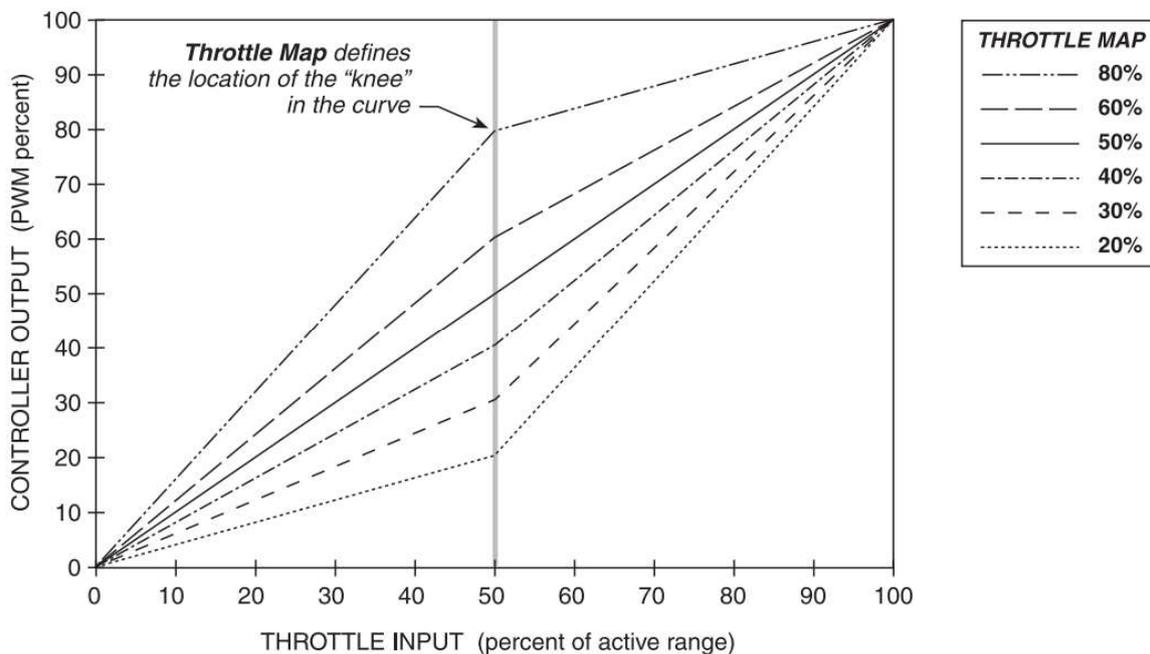
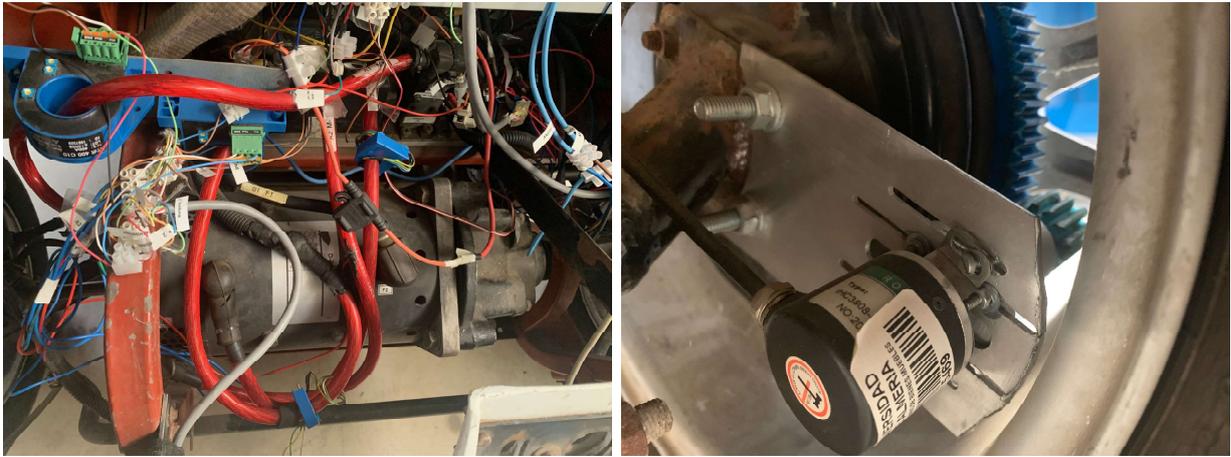


Figura 4.23. Señal PWM para el rango de señales de entrada con distintos *throttle map*.

4.4.1. Modelo Acelerador-Velocidad

Al igual que se ha realizado con modelos anteriores, se procederá a evaluar este sistema para distintos modelos mediante el error cuadrático medio obtenido en la variación de sus parámetros. En la figura 4.24 se muestran los datos recogidos durante unos de los ensayos realizados. Se considerará dos modelos distintos, uno para la velocidad ascendente y otro para la decreciente. De esta manera, uno de los modelos permitirá determinar el efecto del acelerador y el otro será el modelo que se deba controlar con el uso del sistema *Brake-by-Wire*.

4.4. SISTEMA THROTTLE-BY-WIRE



(a) Motor y electrónica en la parte frontal del vehículo. (b) codificador instalado en la rueda trasera derecha.

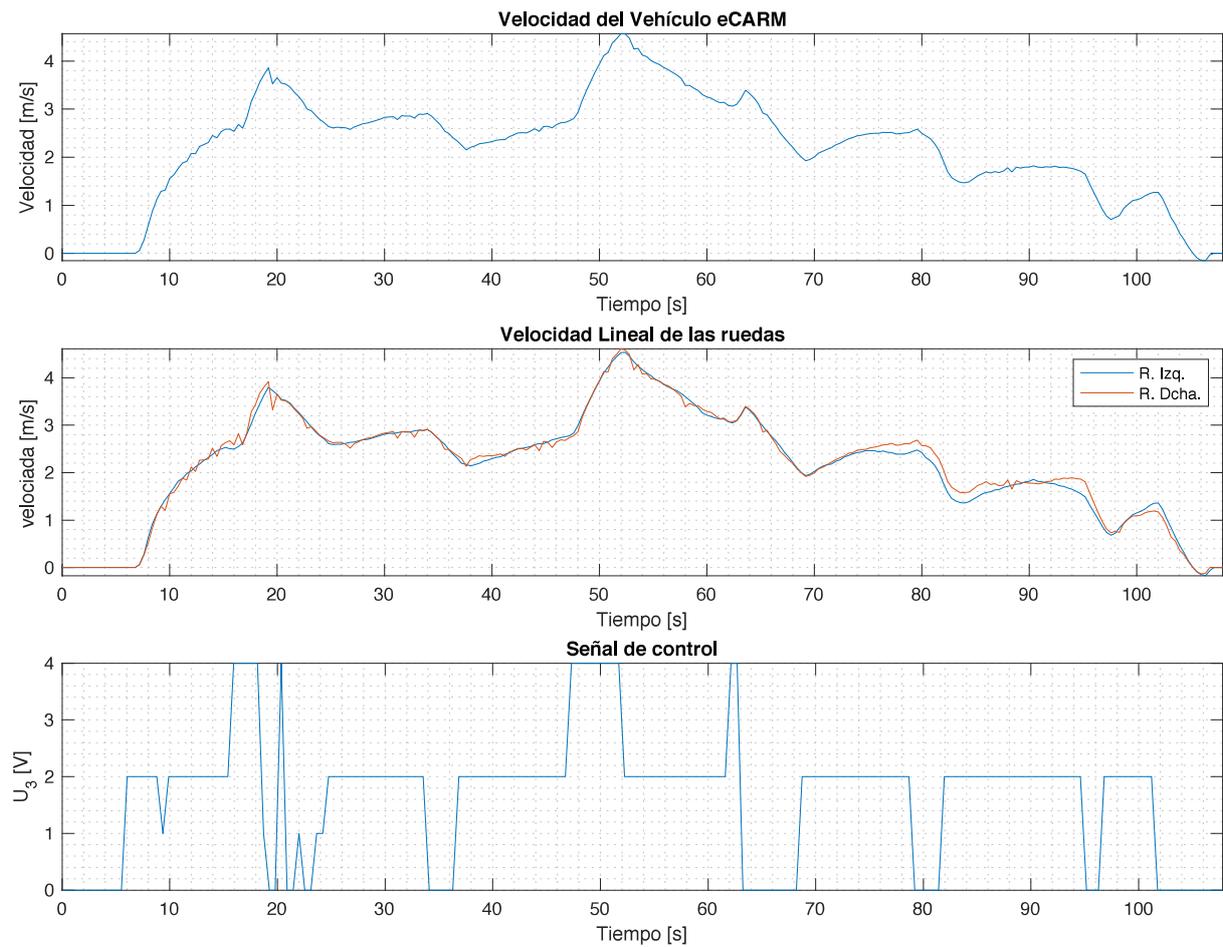


Figura 4.24. Ensayo realizado sobre el sistema *Throttle-by-Wire*.

Este sistema, al presentar la particularidad de poseer dos dinámicas distintas, una de aceleración y otra de frenada, el proceso de obtención de parámetros por el valor del error cuadrático medio requiere un sistema que distinga la zona en la que opera el sistema. De igual forma, tras observar la dinámica del sistema en todos los ensayos realizados, se ha considerado oportuno emplear dos ganancias estáticas según si se opera en la zona inferior, hasta $2,3 [ms^{-1}]$, o por encima de esta velocidad. Puesto que el vehículo se diseña para operar dentro del recinto universitario, se limitará su velocidad a la máxima permitida, $8,3 [ms^{-1}]$. También se estudiará la dinámica de frenada del vehículo haciendo uso del freno motor y con la marcha reversa.

Como información adicional que se debe tener presente en el proceso de modelado, de trabajos anteriores se conoce que el sistema posee una zona muerta de $1 [V]$. Por esta razón, a las señales de control que se empleen en el proceso de modelado se eliminará dicha zona muerta. Esta compensación ya se realiza a nivel software haciendo que la señal mínima que se introduce en el sistema sea de $1 [V]$. El siguiente fenómeno que se puede apreciar visualmente en las gráficas de los ensayos es el posible retardo del sistema. En la gráfica 4.25 se ha representado en distintos ejes de ordenadas señal de salida, velocidad del vehículo, y la señal de control, tensión que emula el pedal de aceleración.

En ella se aprecia claramente la presencia de un tiempo de retardo en el sistema. Analizando en detalle todos los tiempos de respuesta obtenidos, se puede concluir que el tiempo de retardo que se debe considerar en el modelado de este sistema es $0,91 \pm 0,46 [s]$.

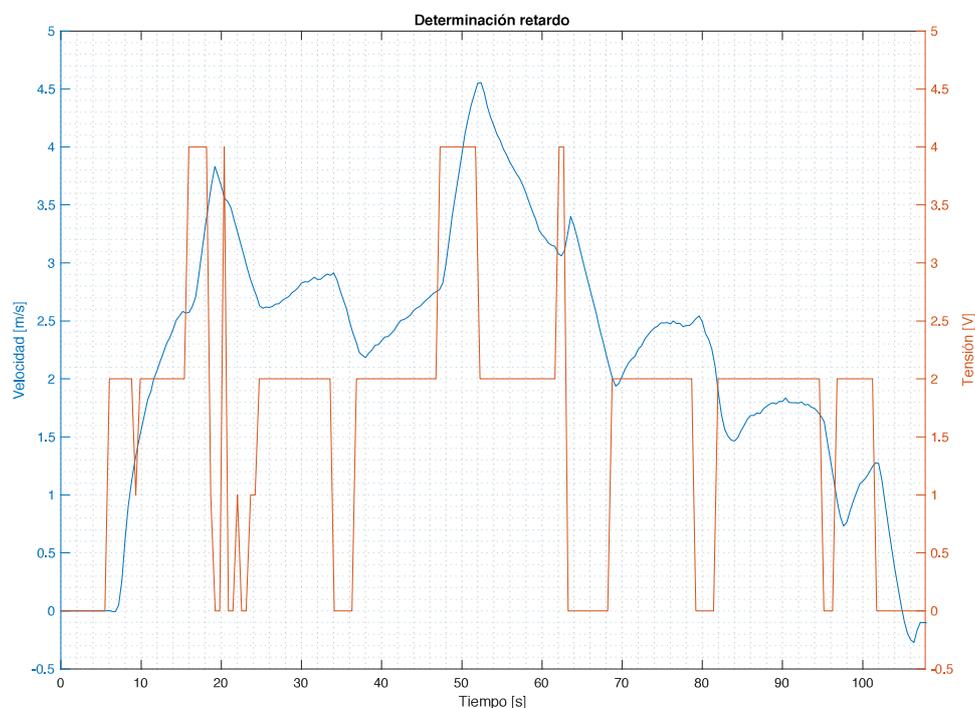


Figura 4.25. Determinación del retardo en la tracción del vehículo.

Haciendo un análisis de los ensayos realizados se han obtenido valores para las ganancias estáticas y constantes de tiempo distintos para los diferentes ensayos. Tras esta observación, se ha decidido obtener los valores de cada parámetro para cada ensayo distinto y proceder a la elección de un valor medio. La ganancia estática obtenida para la zona inferior del rango de operación es $2,45 \pm 0,38 [ms^{-1}V^{-1}]$ mientras que para el tramo superior su valor es $2,03 \pm 0,33 [ms^{-1}V^{-1}]$. En el caso de las constantes de tiempo, para los tramos de subida su valor es $4,86 \pm 1,09s$. La constante de tiempo para tramos descendentes de la señal de control toma el valor $13,93 \pm 3,51 [s]$, una dinámica más lenta tal como se apreciaba visualmente en los datos de los ensayos realizados. Estos parámetros se han obtenido tras realizar un ajuste manual a los resultados obtenidos por el método del error cuadrático medio. Con los parámetros originales, la ganancia del tramo inferior presentaba un valor de $5,5 [ms^{-1}V^{-1}]$ mientras que la del tramo superior era $1,1 [ms^{-1}V^{-1}]$. Según el procedimiento, con estos parámetros y para las constantes de tiempo establecidas, el error obtenido en la validación es de $1,62 [ms^{-1}]$.

No obstante, la respuesta visual del sistema dista en gran manera de la obtenida experimentalmente, figura 4.27. Tras el proceso de ajuste manual, las ganancias se establecieron en los valores citados primeramente, obteniendo un error de $1,84 [ms^{-1}]$. Esta diferencia en el error se considera asumible al apreciarse que la respuesta se corresponde en mayor medida con los resultados experimentales, figura 4.29. Las *f.d.t.* que modelan la dinámica del sistema se reflejan en la tabla 4.7. Se realiza el mismo análisis que en los casos anteriores sobre los errores obtenidos en el vehículo para las dos estimaciones realizadas: el ajuste manual y el óptimo según el error cuadrático medio. En las figuras 4.28a y 4.28b se muestran los resultados del estudio del error para ambos modelos mientras que los datos estadísticos quedan recogidos en la tabla 4.6. Se aprecia claramente que pese a tener una desviación estándar similar, el ajuste manual presenta un error mejor, $-0,2392 \pm 1,8335 [ms^{-1}]$.

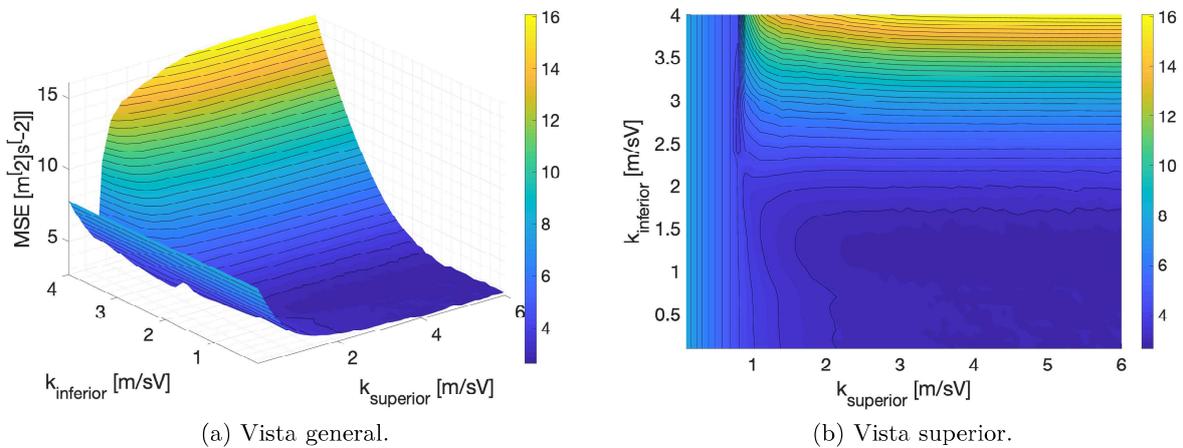


Figura 4.26. Error cuadrático medio para los parámetros estudiados.

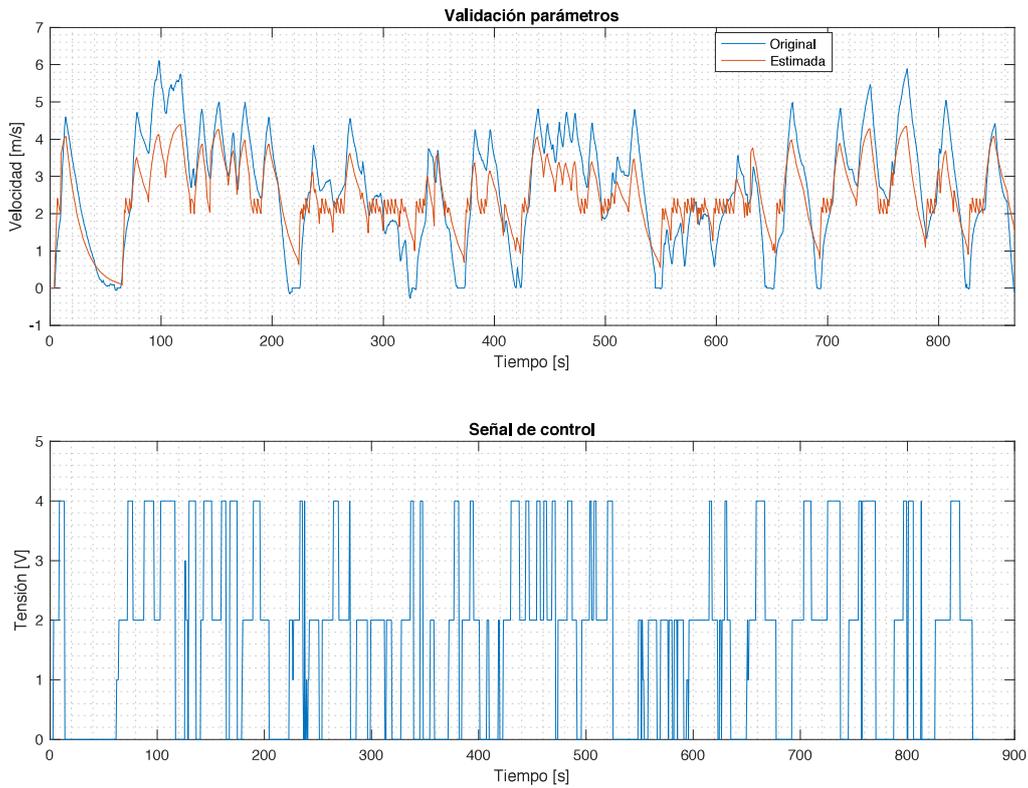
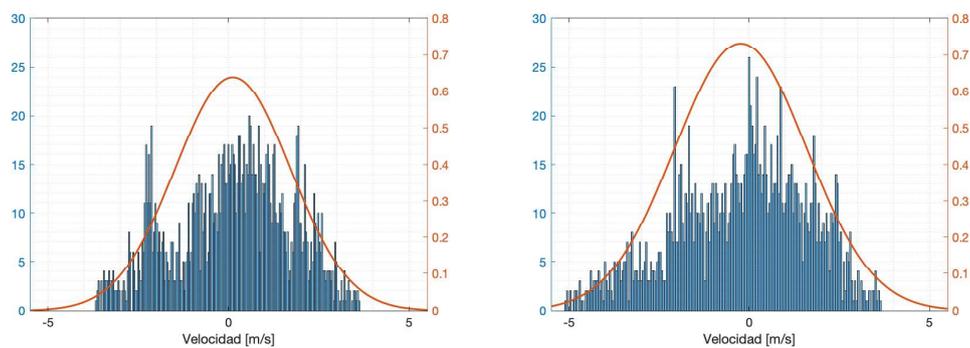


Figura 4.27. Validación del modelado de la dinámica del sistema *Throttle-by-Wire*. MSE.

Modelo	Error medio	Desviación estándar
MSE	0,1149 [$m s^{-1}$]	1,6012 [$m s^{-1}$]
Manual	-0,2392 [$m s^{-1}$]	1,8335 [$m s^{-1}$]

Tabla 4.6. Estudio de errores en los modelos propuestos para la tracción del vehículo.



(a) Histograma del error del modelo según MSE. (b) Histograma del error del modelo manual.

Figura 4.28. Evaluación de errores de los modelos obtenidos.

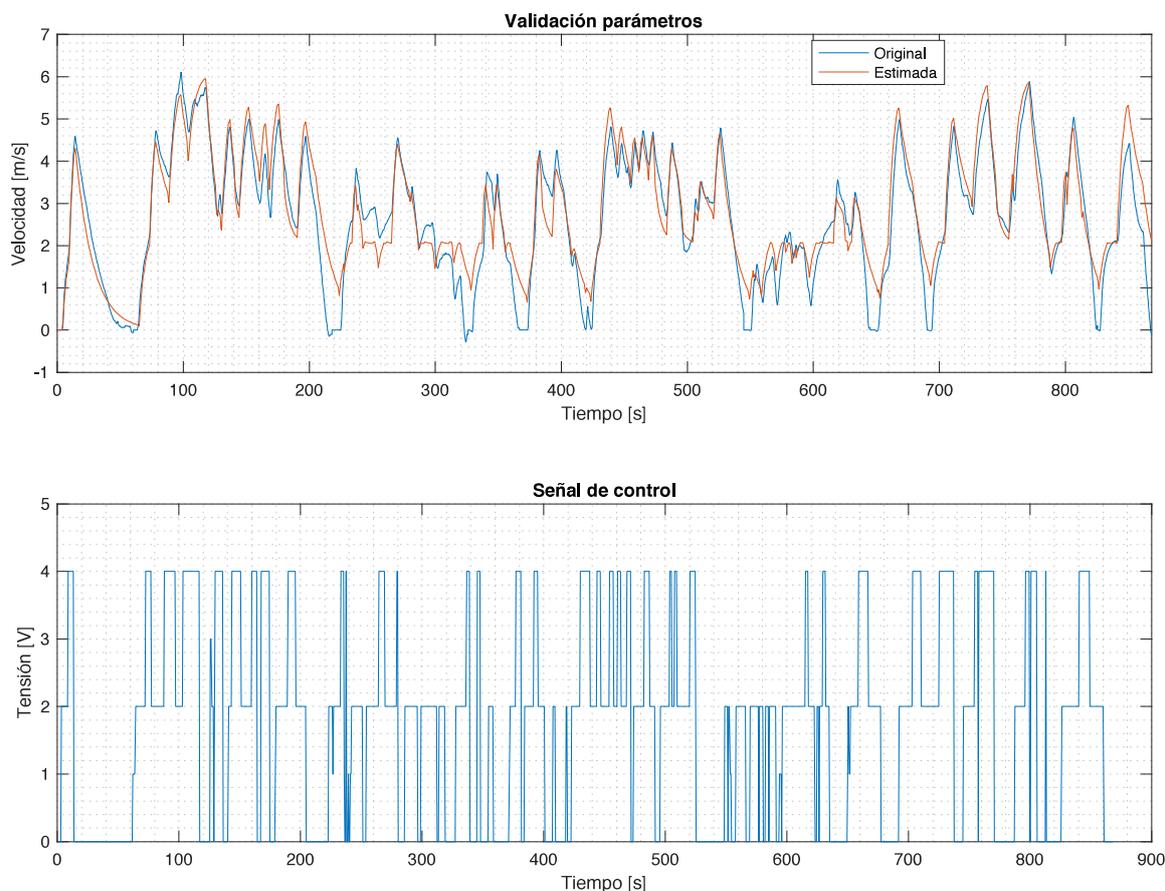


Figura 4.29. Validación del modelado de la dinámica del sistema *Throttle-by-Wire*. Manual

En el caso del estudio de la dinámica de funcionamiento con el sentido de marcha inversa, el estudio se ha realizado de forma similar al empleado en la marcha directa. En la gráfica 4.30 se muestran los resultados de un ensayo en el que se combina las dos marchas del vehículo y el freno motor en los tramos de frenada durante la marcha inversa. En el estudio de la marcha inversa del vehículo, se vuelve a apreciar la misma no linealidad en su operación que en el caso de la marcha directa. Esta no linealidad está asociada a distintas ganancias estáticas en su rango de operación. Estos nuevos valores se muestran en la tabla 4.7 junto con los determinados para la marcha directa.

Marcha	Zona Operación	Aceleración	Frenado
Directa	$0 < v < 2,3 \text{ m s}^{-1}$	$\frac{2,45}{4,86s+1} [\text{ms}^{-1}\text{V}^{-1}]$	$\frac{2,45}{13,93s+1} [\text{ms}^{-1}\text{V}^{-1}]$
	$2,3 < v \text{ m s}^{-1}$	$\frac{2,03}{4,86s+1} [\text{ms}^{-1}\text{V}^{-1}]$	$\frac{2,03}{13,93s+1} [\text{ms}^{-1}\text{V}^{-1}]$
Inversa	$-2,05 < v < 0 \text{ m s}^{-1}$	$\frac{2,45}{6,05s+1} [\text{ms}^{-1}\text{V}^{-1}]$	$\frac{2,45}{1,65s+1} [\text{ms}^{-1}\text{V}^{-1}]$
	$v < -2,05 \text{ m s}^{-1}$	$\frac{1,35}{6,05s+1} [\text{ms}^{-1}\text{V}^{-1}]$	$\frac{1,35}{1,65s+1} [\text{ms}^{-1}\text{V}^{-1}]$

Tabla 4.7. Ecuaciones que rigen la dinámica del sistema *Throttle-by-Wire*.

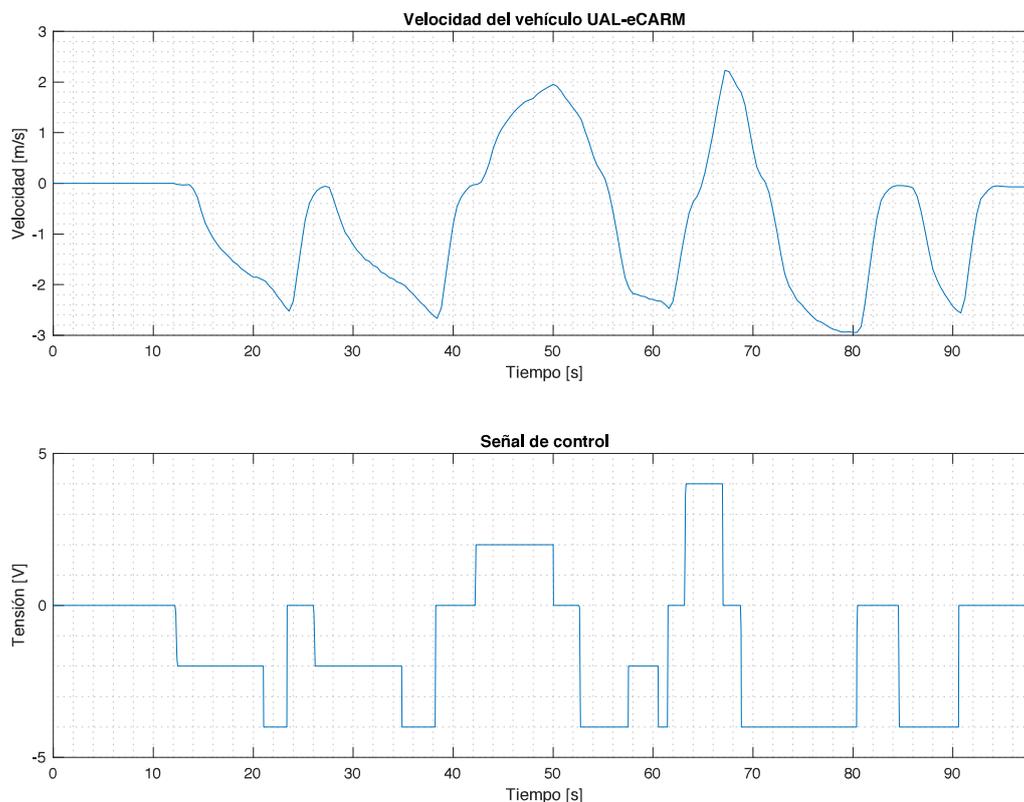


Figura 4.30. Ensayo realizado *Throttle-by-Wire* usando las dos marchas y el freno motor.

El empleo del freno motor solo afecta a la constante de tiempo del sistema en su tramo descendente, disminuyendo su valor. En este caso el valor obtenido para la constante de tiempo es de 1,65 [s] independientemente de la zona o en la marcha en la que opere. Con este valor, en el caso de un cambio de referencia hacia la velocidad nula, este mecanismo implica unos mejores resultados de desempeño en cuanto al tiempo de respuesta frente a dejar la dinámica propia del vehículo o empleo del mecanismo *Brake-by-Wire*.

4.4.2. Modelo Acoplamiento Velocidad-Dirección

Para complementar la información del modelado del vehículo, se procede a determinar el efecto de la velocidad del sistema sobre la dinámica del mecanismo de la dirección. Este acoplamiento entre sistemas no se ha planteado en trabajos anteriores. Para su determinación, se comparará la dinámica del mecanismo de la dirección con el vehículo en movimiento con el modelo de segundo orden obtenido en la sección 4.2, ecuación 4.11. Se toma como señal de entrada la velocidad del vehículo y como señales de salida se estudiarán la velocidad de giro y el consumo de corriente del mecanismo de dirección. En las figura 4.31 se muestra un ensayo realizado con los datos que se emplearán para la determinación del modelo de interferencia.

4.4. SISTEMA THROTTLE-BY-WIRE

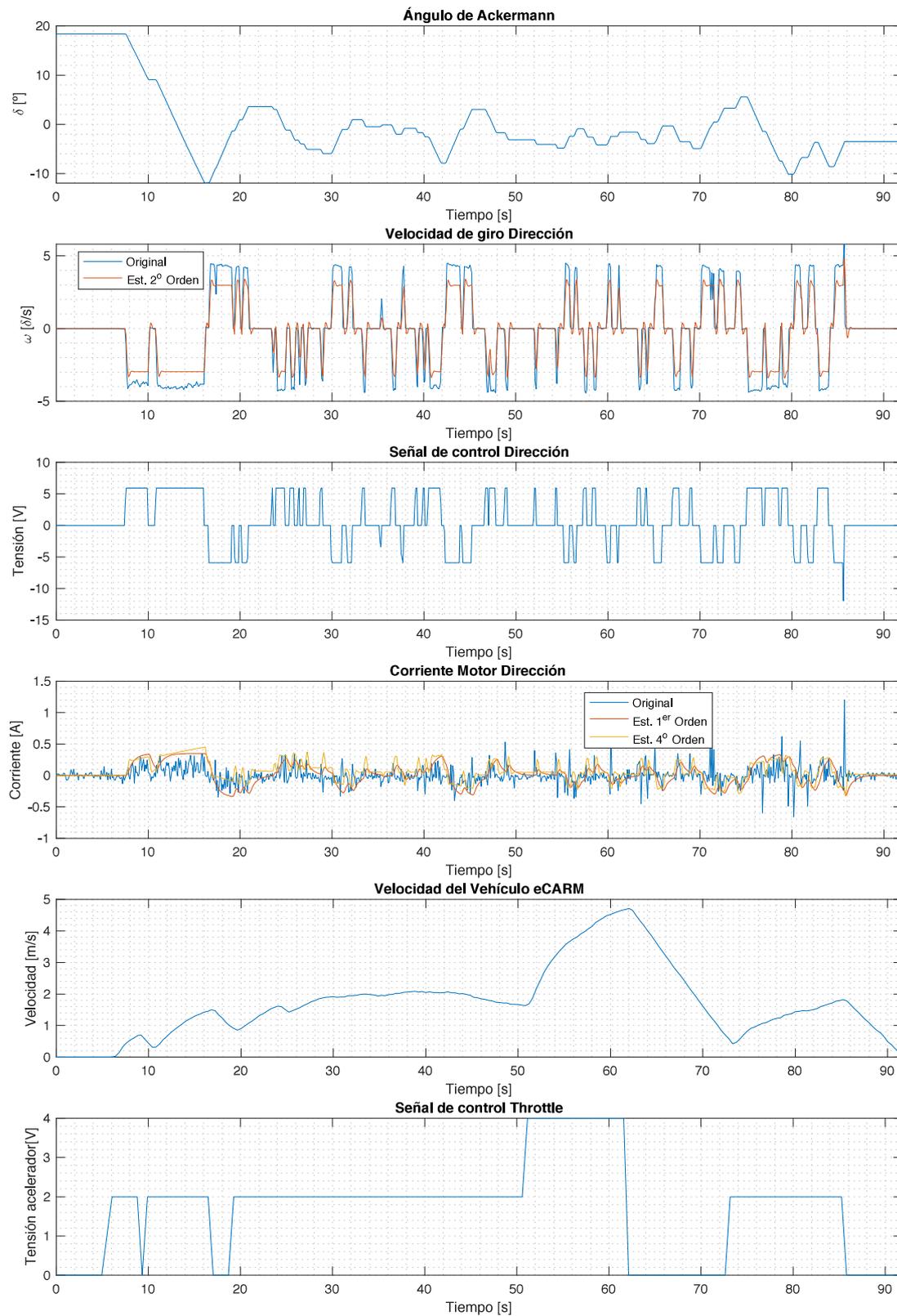


Figura 4.31. Datos del vehículo durante un ensayo realizado.

Como se puede apreciar, la dinámica de la dirección con el vehículo en movimiento difiere de la obtenida con el vehículo estacionado. En el caso de los modelos de la velocidad de giro del motor de la dirección respecto de la señal PWM de control, se aprecia claramente que el hecho de que el vehículo se encuentre en movimiento influye en la dinámica calculada en la sección 4.2. Tal como se esperaba, la dinámica del sistema *Steer-by-Wire* muestra una menor resistencia del mecanismo. Esto se traduce en una ganancia estática mayor en la ecuación 4.11 y un menor consumo de corriente. Esta última diferencia es ligeramente menor puesto que al aumentar la velocidad de giro del motor, en consecuencia también se incrementa el consumo del mismo. En el caso de la ganancia estática, se aprecia que la diferencia entre los datos experimentales y el modelo, es prácticamente la misma a pesar de las diferencias entre los datos de velocidad del vehículo. Esto implica que el acoplamiento no depende de la velocidad del vehículo, sino del hecho de que el vehículo se encuentre en movimiento o estático.

Para capturar la dinámica más precisa, que sumada al modelo previo más se aproxima a los datos experimentales, se hace un barrido similar al llevado a cabo en todos los procesos de modelado realizados este capítulo considerando la señal de control como una señal unitaria cuando el vehículo está en movimiento. Se estudiará esta dinámica como un sistema de primer orden. El barrido de la ganancia estática se realiza desde $0 [\delta s^{-1}]$ hasta $-3 [\delta s^{-1}]$ con un paso de $-0,001 [\delta s^{-1}]$. En el caso de la constante de tiempo, el barrido comienza en $0 [s]$ y finaliza en $1 [s]$ con un paso de $0,01 [s]$. En las figuras 4.32a y 4.32b se muestran los errores cuadráticos medios obtenidos para el modelo propuesto. Como resultado, los parámetros que presentan mejores resultados se corresponden con una constante de tiempo nula y una ganancia estática de valor $-1,29 [\delta/s]$. Para el caso de la corriente, el ajuste se ha realizado modificando la ganancia estática de los modelos obtenidos en la sección 4.2. Ajustando el nuevo valor de la ganancia del modelo, se ha obtenido que el nuevo ajuste óptimo se corresponde con una reducción del 65%.

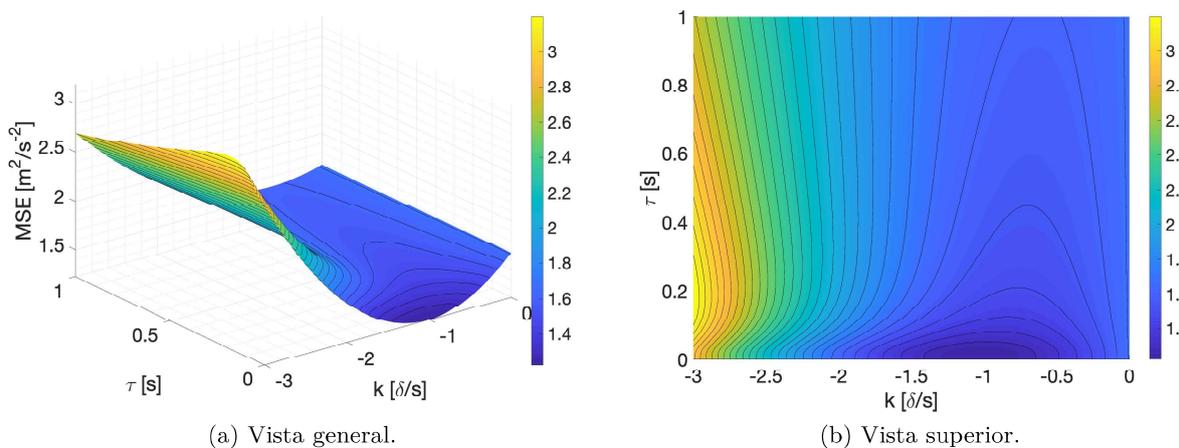


Figura 4.32. Error en función de los parámetros para un modelo de primer orden del acoplamiento.

Por lo tanto, el acoplamiento de la velocidad del vehículo con el modelo de consumo de corriente de la dirección se corresponde con un factor de corrección a la salida del modelo. Este factor adoptará el valor unitario cuando el vehículo se encuentra estático y el valor 0,35 cuando el vehículo se encuentre en movimiento. En la figura 4.33 se muestra los resultados de la validación de los parámetros escogidos para el acoplamiento, donde se puede apreciar la mejora en la correlación de la señal estimada, considerando el acoplamiento, con la señal experimental, respecto de la señal sin dicha corrección.

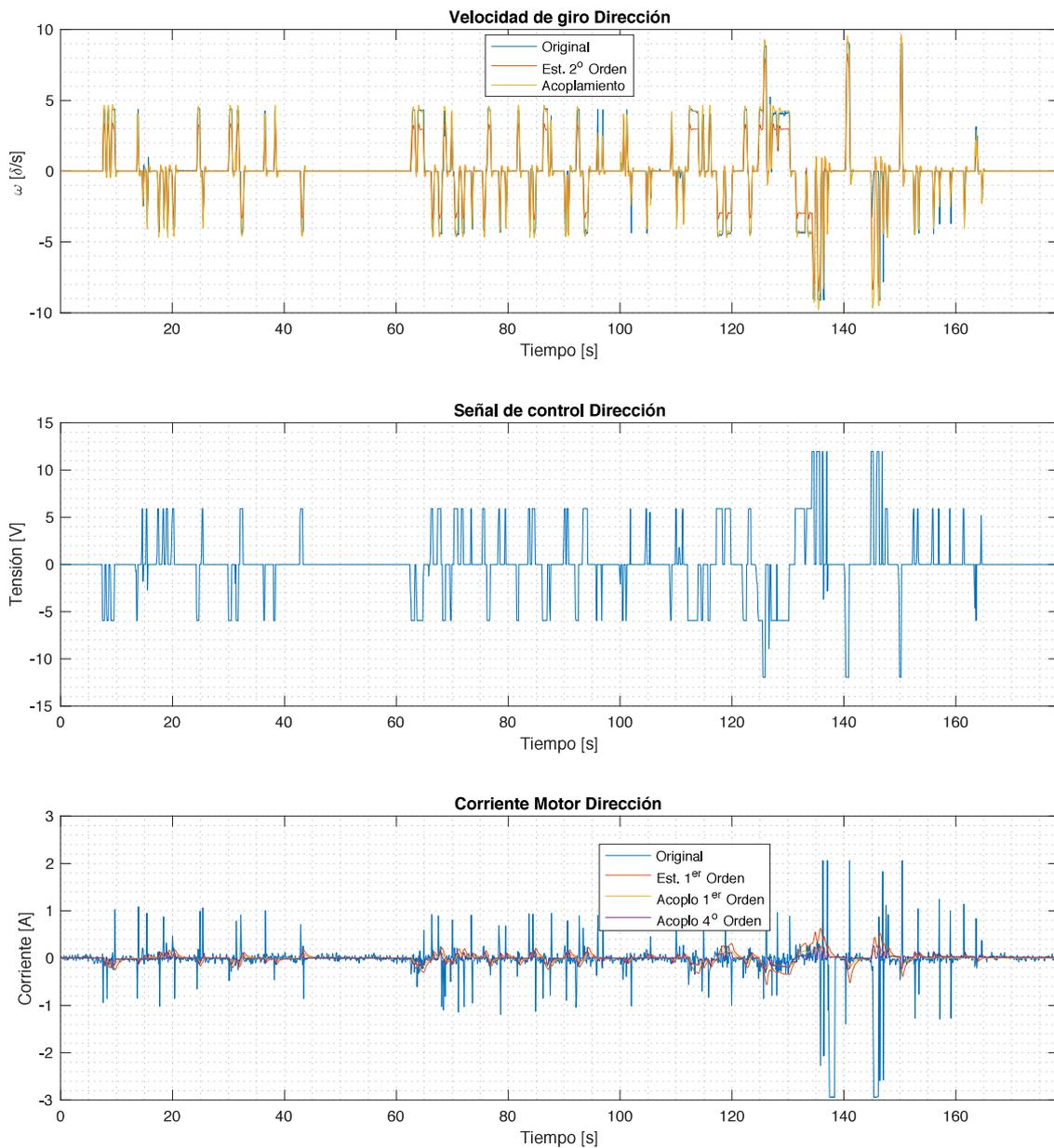


Figura 4.33. Validación de la dinámica del acoplamiento estudiado.

4.5. Vehículo UAL-eCARM

A partir de toda la información obtenida en el presente capítulo, se procede a unificarla en un mismo modelo en la herramienta *Simulink*. Este modelo se tomará como base para posteriores trabajos a realizar sobre el vehículo relativos a su dinámica. En este caso, el modelo servirá como fundamento para el diseño de la arquitectura de control a bajo nivel propuesta para el vehículo en el próximo capítulo.

En primer lugar, la figura 4.34, muestran los diagramas de bloques de cada sistema, estudiados anteriormente, unificados en un diagrama de bloques completo del vehículo indicando los flujos de información que se producen. Con la finalidad de poder ser usado como mapa de cara a próximos desarrollos numéricos, en azul se han indicado la ubicación y significado de cada señal involucrada en el proceso. Las señales de control y entradas al sistemas reciben la nomenclatura U_x , siendo la señales de tensión que reciben los motores de dirección y freno, y la etapa de potencia *Curtis*. En el caso de las señales de salida, reciben la nomenclatura Y_x . Estas señales son las posiciones, velocidades de giro y corriente demandada en los motores de dirección y freno, y las velocidades de avance de cada rueda, así como la del vehículo.

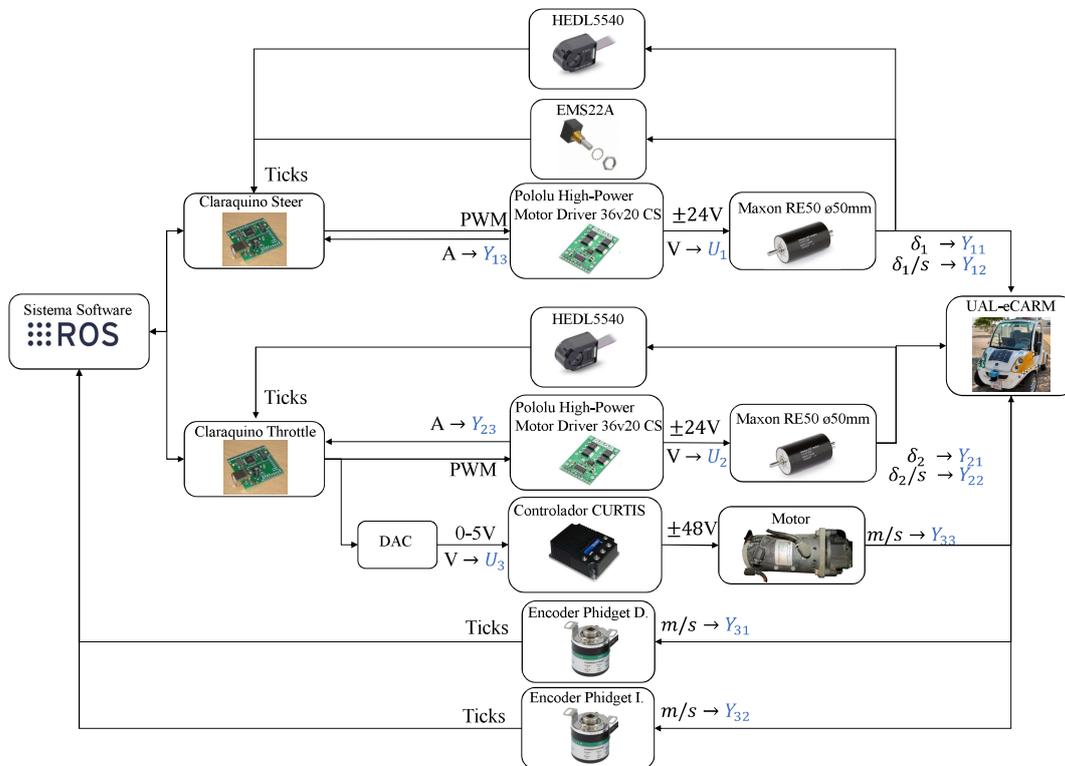


Figura 4.34. Diagrama de bloques completo del del vehículo UAL-eCARM.

A continuación, en la figura 4.35, se muestra, en forma de diagrama de bloques, las *f.d.t.* determinadas para el comportamiento del sistema. Con esto queda demostrado que se trata de un sistema multivariable del tipo *M.I.M.O.*, cuyas entradas y salidas están relacionadas entre sí. No obstante, no será preciso el análisis de la matriz *RGA* para determinar el emparejamiento de las entradas y salidas del sistema. Esto se debe a que cada entrada controla el actuador de cada sistema y son las salidas de estos, las que modifican las dinámicas.

En la representación de la figura 4.35, las interacciones entre sistemas pasan a través de las funciones $f_{1,2,3}(x)$. La función $f_1(x)$ es la encargada de aplicar un factor de corrección a la ganancia de la *f.d.t.* que estima el consumo de corriente en la dirección. Cuando el vehículo se encuentra detenido, $f_c = 1,00$, mientras que cuando se encuentra en movimiento, $f_c = 0,35$. La función $f_2(x)$ situada a la salida de la posición del motor del freno, representa la dinámica que de frenado que implica el empleo del pedal de freno en el sistema. En este estudio no se ha modelado, no obstante, se debe tener en cuenta que esa interacción existe y su dinámica es desconocida si se pretende desarrollar un controlador para el sistema *Brake-by-Wire*. Finalmente, la función $f_3(x)$ se trata de la adecuación de la señal de control de la función de acoplo $D_{31}(s)$. Esta señal de control se trata de una señal en escalón de valor unitario que se encuentra activa siempre que se de la casuística de la señal de control $U_1 \neq 0$ y $Y_{33} \neq 0$. Las señales de control U_1 , U_2 y U_3 se corresponden con la tensión que recibe el motor de la dirección, el freno y el controlador Curtis respectivamente. La señales del sistema *Steer-by-Wire*, Y_{13} , Y_{12} y Y_{11} , se corresponden con el consumo de corriente, la velocidad de giro y la posición del motor acoplado al mecanismo respectivamente. En el sistema *Brake-by-Wire*, las señales Y_{23} , Y_{22} y Y_{21} se corresponden con el consumo de corriente, la velocidad de giro y la posición del motor que acciona el pedal de freno.

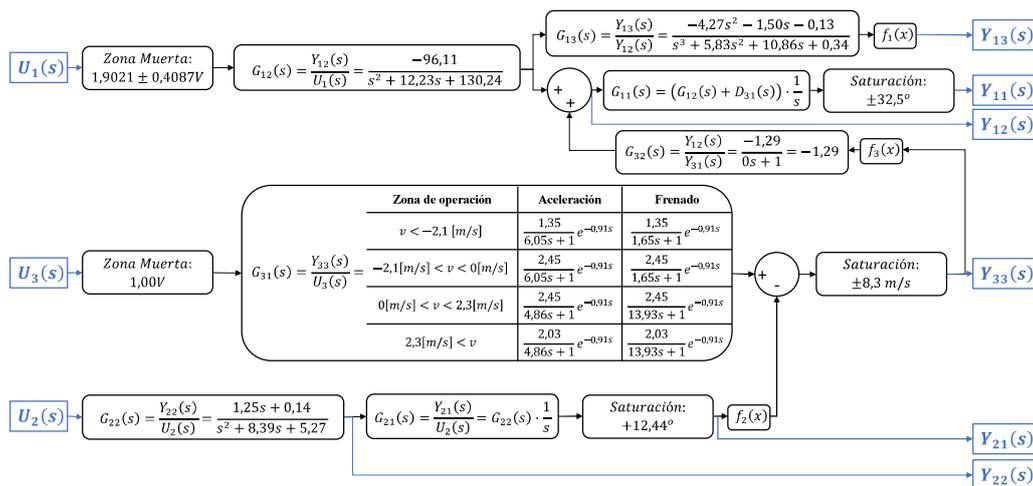


Figura 4.35. Diagrama resumen de la dinámica completa del vehículo.

Este proyecto se ha realizado con la finalidad modelar y controlar la dinámica del vehículo durante su operación normal. No obstante, gracias a los dispositivos instalados, como las tarjetas descritas en la sección 3.2.8, es posible realizar el modelado del vehículo desde la perspectiva energética. El modelo implementado en simulación, mostrado en la figura 4.36, es el resultado de la implementación práctica del modelo mostrado en la figura 4.35 en la herramienta de simulación *Simulink* como se ha dicho anteriormente.

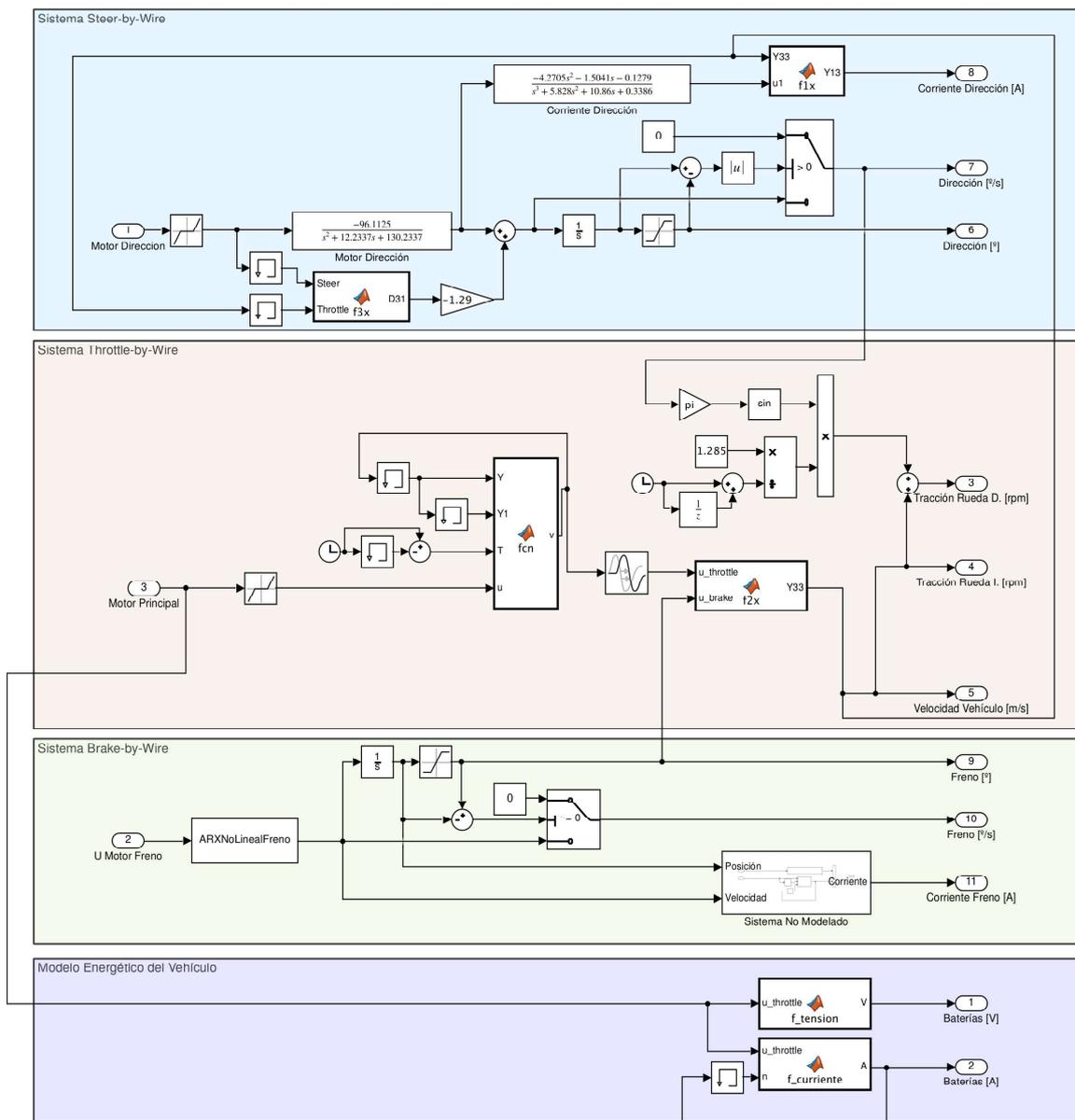


Figura 4.36. Modelo del vehículo UAL-eCARM implementado en Simulink.

Capítulo 5

Control del sistema *Drive-by-Wire*

5.1. Introducción

Una vez determinada la respuesta de cada uno de los sistemas principales del vehículo UAL-eCARM, el siguiente paso en su estudio es el desarrollo de una arquitectura de control que actúe adecuadamente sobre el sistema. A la luz de lo que se ha observado en la sección 4.5, la arquitectura de control que se debe implementar se puede desarrollar de forma independiente para cada sistema. No obstante, para cada sistema se debe tener en cuenta las relaciones que guarda con el resto de sistemas.

Como se observa tras el estudio realizado en la sección 2.5 del estudio bibliográfico, la naturaleza de los sistemas que componen un vehículo implica que no existe una solución única para el desarrollo de controladores y que funcione para todo el vehículo. De esta forma, es evidente que los sistemas de la dirección y el freno implican un control en cascada para establecer un control de velocidad y posición para los distintos motores, mientras que el motor principal del vehículo presenta retardo y una alta no linealidad lo que requiere de arquitecturas más complejas. Por esta razón es de gran interés el desarrollo de un simulador del vehículo que permita estudiar y evaluar distintas estrategias de control entre sí.

5.2. Control sistema *Steer-by-wire*

El primer sistema estudiado es el correspondiente al mecanismo de la dirección del vehículo. En este caso, el sistema de control requiere una arquitectura del tipo cascada. Con esta implementación se pretende controlar la posición del mecanismo de dirección, mientras que el modelo obtenido para la dinámica velocidad de giro - corriente, ecuación 4.15, se empleará como mecanismo de seguridad como ya se ha comentado anteriormente. Esta implementación consistirá en limitar la referencia de velocidad si se detecta que la velocidad que se exige al motor implica un nivel inadecuado de corriente.

5.2.1. Lazo interno. Control de velocidad de giro

El lazo interno del sistema se corresponde con el control de la velocidad de giro del mecanismo de dirección. En este caso, atendiendo al diagrama mostrado en la figura 4.35, se observa que además de la función de transferencia principal, ecuación 4.11, se debe considerar el acoplamiento que se produce cuando el vehículo se encuentra en movimiento. Antes de comenzar con el desarrollo del controlador, hay que destacar que se trata de un sistema de constante de tiempo dominante. Tanto el acoplamiento como la ausencia de retardo son las principales diferencias y mejoras con respecto al anterior trabajo de modelado realizado sobre el vehículo en [14].

En primer lugar se determina el controlador, $C_{vi}(s)$, para la función principal en relación a la velocidad de giro del mecanismo, G_{12} . Se trata de un sistema de segundo orden con dos polos y ningún cero. El procedimiento realizado para el diseño del controlador consiste en el empleo de un controlador PID con una estructura ideal, figura 5.1, con el cual se cancelará la ecuación característica del sistema y se podrá reducir a un primer orden en bucle cerrado. En la ecuación 5.1, se muestra la expresión que determina la dinámica en bucle cerrado del sistema.

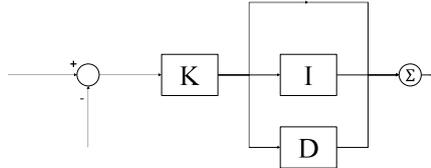


Figura 5.1. Configuración PID Ideal.

$$G_{bc}(s) = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)} = \frac{L(s)}{1 + L(s)} \quad (5.1)$$

$$G_v(s) = \frac{k_v \cdot \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$C_v(s) = K_{vp} \left(1 + \frac{1}{T_{vis}} + T_{vds} \right) = K_{vp} \frac{T_{vis} + 1 + T_{vi}T_{vds}^2}{T_{vis}} = \frac{K_{vp}T_{vd}}{s} \left(s^2 + \frac{1}{T_{vd}}s + \frac{1}{T_{vi}T_{vd}} \right)$$

$$L_{vi}(s) = C_v(s) \cdot G_{12}(s) = \frac{K_{vp}T_{vd}}{s} \left(s^2 + \frac{1}{T_{vd}}s + \frac{1}{T_{vi}T_{vd}} \right) \cdot \frac{k_v \cdot \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\left. \begin{array}{l} \frac{1}{T_{vd}} = 2\zeta\omega_n \\ \frac{1}{T_{vi}T_{vd}} = \omega_n^2 \end{array} \right\} L_{vi}(s) = \frac{k_v \cdot \omega_n^2 \cdot K_{vp} \cdot T_{vd}}{s} = \frac{k_v \cdot K_{vp}}{T_{vis}} \quad (5.2)$$

$$G_{v,bc}(s) = \frac{\frac{k_v K_{vp}}{T_{vis}}}{1 + \frac{k_v K_{vp}}{T_{vis}}} = \frac{k_v K_{vp}}{T_{vis}(1 + \frac{k_v K_{vp}}{T_{vis}})} = \frac{k_v K_{vp}}{T_{vis} + k_v K_{vp}} = \frac{1}{\frac{T_{vi}}{k_v K_{vp}}s + 1} \rightarrow \tau_{bc,v} = \frac{T_{vi}}{k_v K_{vp}} \quad (5.3)$$

A partir de las ecuaciones 5.2 y 5.3 se pueden determinar los términos del controlador PID a partir de los parámetros del modelo y de la constante de tiempo que se persigue en bucle cerrado. Para este caso, el término derivativo, T_{vd} , adopta el valor 0,0818 [s], mientras el término integral, T_{vi} , adopta el valor 0,0939 [s]. Estos parámetros, como se aprecia en la ecuación 5.2, son los que dependen exclusivamente de la dinámica del proceso a controlar. El término proporcional del controlador, K_{vp} , depende de la constante de tiempo que se desee en bucle cerrado, $\tau_{bc,v}$ y del término integral ya diseñado. En este caso, se fija este parámetro en 0,2 [s], un valor conservador que no compromete las prestaciones del sistema. Para este parámetro, el valor obtenido para el término K_{vp} adquiere el valor $-0,6362 [Vs\delta^{-1}]$. El controlador resultante se muestra en la ecuación 5.4.

$$C_v(s) = K_{vp} \left(1 + \frac{1}{T_{vi}s} + T_{vd}s \right) = -0,6362 \left(1 + \frac{1}{0,0939s} + 0,0818s \right) [Vs\delta^{-1}] \quad (5.4)$$

Como implementación complementaria para mejorar el desempeño del controlador, se implementará el mecanismo anti-windup. Este mecanismo persigue evitar la acumulación de error por parte del término integral del controlador frente a la saturación de la señal de control. En el vehículo, esta saturación se produce a $\pm 24V$, la tensión de alimentación de la etapa de potencia. Esta implementación mide la diferencia de la señal que introduce en el sistema y la salida del controlador, e inyecta nuevamente este término a la salida del controlador tras pasarlo por una ganancia de valor $\frac{1}{\sqrt{T_{vi}}}$ y un integrador.

Puesto que el controlador debe operar implementado en un microcontrolador, se debe trabajar con su equivalente en tiempo discreto. Para llevar a cabo dicha transformación al plano Z, se empleará la aproximación trapezoidal (Tustin), ecuación 5.5, para la discretización de parámetros del controlador. En la ecuación 5.6 se muestra la expresión genérica de un controlador PID en tiempo discreto mientras que en la ecuación 5.7 se muestra el cálculo de parámetros para el caso de un tiempo de muestreo de 0,1 [s].

$$s \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (5.5)$$

$$C_v(z) = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \quad (5.6)$$

$$\left. \begin{aligned} q_0 &= K_p \left(1 + \frac{T}{2T_i} + \frac{T_d}{T} \right) \\ q_1 &= K_p \left(-1 + \frac{T}{2T_i} + \frac{T_d}{T} \right) \\ q_2 &= K_p \left(\frac{T_d}{T} \right) \end{aligned} \right\} C_p(z) = \frac{-1,4954 - 0,2229z^{-1} - 0,5204z^{-2}}{1 - z^{-1}} \quad (5.7)$$

El siguiente paso del diseño del lazo interno del sistema *Steer-by-Wire* es el desarrollo del desacoplo con el sistema de tracción del vehículo. Tal como se puede observar en la figura 4.35, el acoplamiento consiste en un factor que solo depende en un factor constante de $-1,29 [\delta s^{-1}]$ cuando el mecanismo y el vehículo se encuentran en movimiento. Para contrarrestar este efecto se debe inyectar a la salida del controlador la señal opuesta dividida la f.d.t. del proceso, ecuación 5.8. Como se puede apreciar en dicha ecuación, se obtiene un sistema no causal. Esta implementación no es factible por lo que la solución que se debe adoptar una solución de compromiso. En este caso se opta por emplear una ganancia estática que asegure la causalidad del sistema y permita el desacoplo parcial de la dirección y tracción del vehículo. Se ha partido del cociente entre ganancias estáticas para posteriormente realizar un ajuste manual que ha permitido mejorar sensiblemente la respuesta del desacoplo. El ajuste ha finalizado con una ganancia estática de valor $0,9519 [Vs\delta^{-1}]$.

$$D_{31} = -\frac{G_{32}}{G_{12}} = -\frac{-1,29}{\frac{-96,1125}{s^2+12,2337s+130,2337}} = -\frac{1,29 \cdot (s^2 + 12,2337s + 130,2337)}{96,1125} \quad (5.8)$$

En la figura 5.2 se muestra de forma esquemática la implementación del lazo de control interno. La referencia se ha realizado mediante dos escalones de igual amplitud en distintos instantes de tiempo de forma que se puede evaluar tanto la dinámica de ascenso y descenso. Se observa la implementación anti-windup para tratar la posible saturación de la señal de control. Antes de la saturación de la señal de control, se encuentra acoplado un sumador. A este elemento se encuentra acoplado la señal estimada para el desacoplo de la incidencia de la velocidad del vehículo sobre la dinámica del sistema. Tras la función de saturación se ha considerado oportuno la inclusión de una señal en escalón a modo de perturbación del sistema. Esta señal se puede equivaler a factores externos como el cambio de tipo de superficie donde opere el vehículo. De esta forma, se puede evaluar el desempeño del controlador frente al rechazo de perturbaciones. Finalmente, tras la f.d.t. que simula el proceso se añade la señal de acoplamiento con la tracción del vehículo.

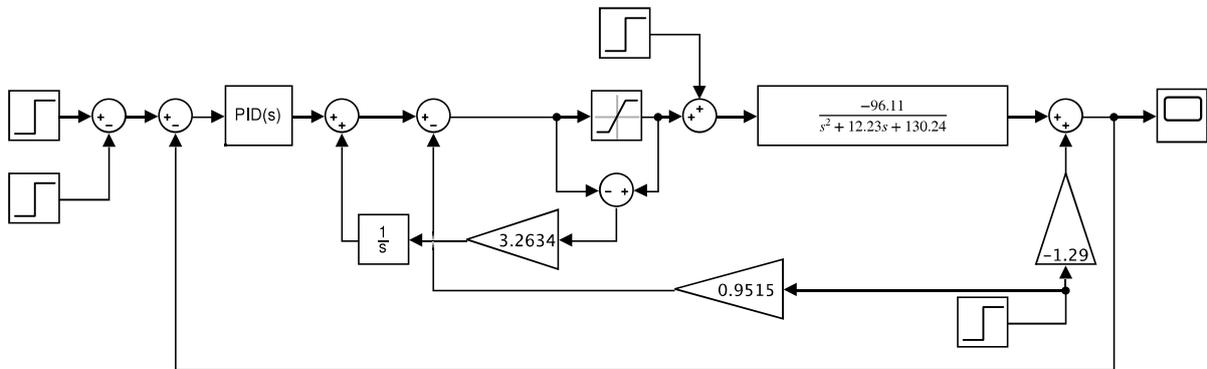


Figura 5.2. Esquema de control propuesto para el lazo interno del sistema *Steer-by-Wire*.

En la figura 5.3 se muestran los resultados de la validación del control implementado en el lazo interno del mecanismo. Se cumple la especificación impuesta de 0,2 [s] como constante de tiempo frente al seguimiento de la consigna de referencia. En el instante 4 s se introduce una perturbación de amplitud unitaria que se rechaza en 0,65 [s]. Este tiempo se puede considerar suficiente para valorar el desempeño del controlador como adecuado. En el instante 6 s se introduce la señal de acoplamiento la cual es rechazada en un tiempo de 0,5 [s]. Se puede apreciar que se produce una ligera sobreoscilación en la salida, originada por el hecho del ajuste manual que se ha realizado sobre la ganancia de desacoplo. Se ha considerado como una solución de compromiso aceptable para acelerar el rechazo a la señal del acoplamiento.

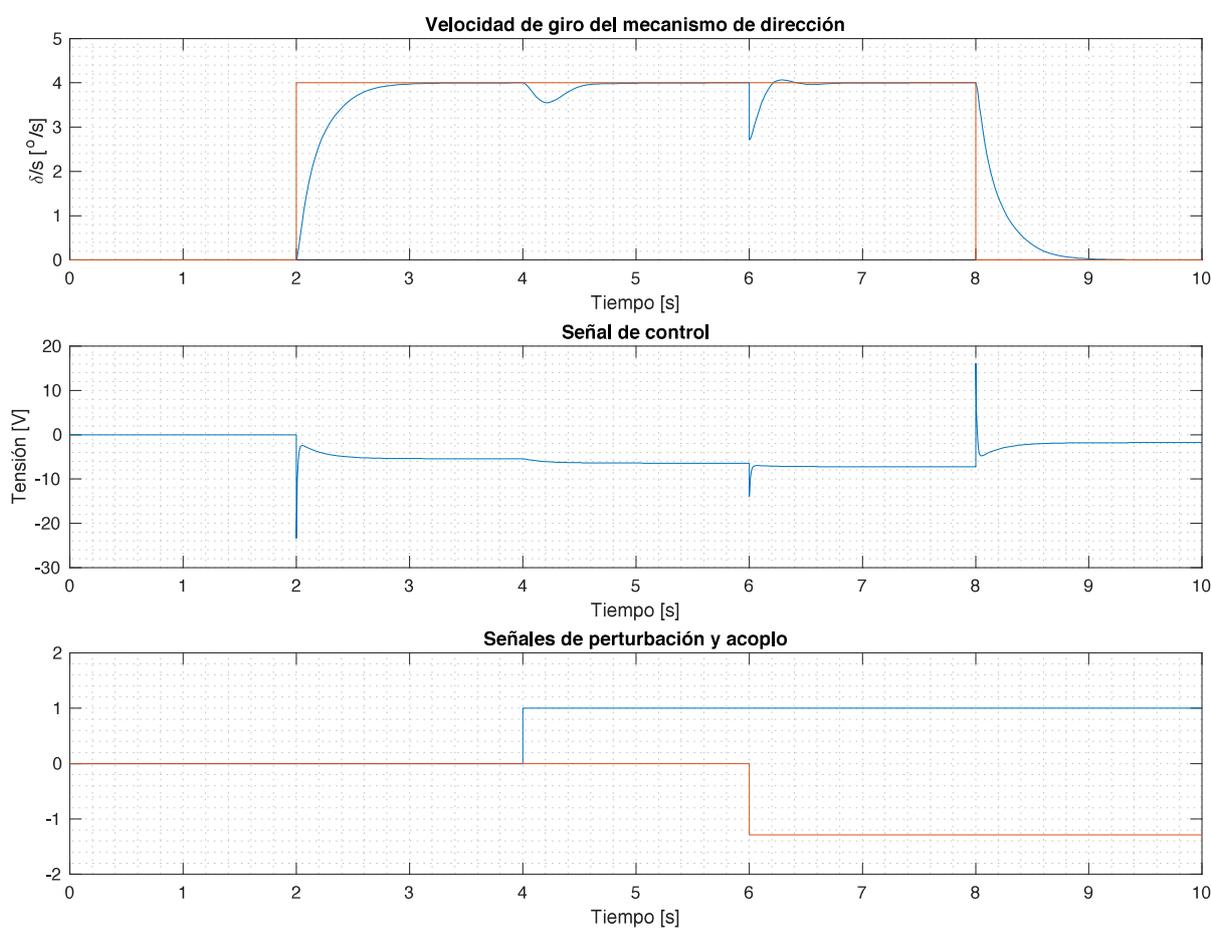


Figura 5.3. Validación para el control propuesto en el lazo interno del sistema *Steer-by-Wire*.

5.2.2. Lazo externo. Control del ángulo de Ackermann

Una vez determinado el control de velocidad del mecanismo, se procede a desarrollar el sistema de control de posición. Este sistema recibirá la referencia de posición deseada para el mecanismo de dirección en forma de ángulo de Ackermann.

La señal de control generada se corresponde con la referencia de velocidad de giro que se introduce en el lazo interno de control desarrollado anteriormente. En la ecuación 5.9 se muestra la dinámica del proceso que se pretende controlar como resultado de añadir un integrador al bucle cerrado del lazo interno.

$$G_{11} = \frac{1}{s} \cdot \frac{1}{0,2s + 1} \quad (5.9)$$

Según la dinámica del proceso a controlar, se propone tres diversos controladores para evaluar las ventajas e inconvenientes de cada uno a fin de implementar el de mejor desempeño. Los tres controladores se diseñarán haciendo uso del método de cancelación polo-cero y asignación de polos. Se implementará la saturación de la señal de control para limitar la velocidad de giro del mecanismo y no comprometer el funcionamiento del sistema. Todos aquellos controladores que poseen término integral se le añadirá el mecanismo anti-windup. El primer controlador que se diseñará se trata de un controlador del tipo PD, ecuación 5.10, para obtener una dinámica de lazo cerrado equivalente a un modelo de primer orden con un polo. A continuación se expone el desarrollo realizado para determinar el valor de los términos proporcional y derivativo del controlador propuesto. Se establecerá como constante de tiempo en bucle cerrado un valor de 0,2 [s].

$$C_p(s) = K_{pp}(1 + T_{pd}s) \quad (5.10)$$

$$L_p(s) = C_p(s) \cdot G_{11}(s) = K_{pp}(1 + T_{pd}s) \cdot \frac{1}{s} \cdot \frac{1}{s(0,2s + 1)} \rightarrow T_{pd} = 0,2 \rightarrow L_p(s) = \frac{K_{pp}}{s} \quad (5.11)$$

$$G_{p,bc}(s) = \frac{\frac{K_{pp}}{s}}{1 + \frac{K_{pp}}{s}} = \frac{1}{\frac{1}{K_{pp}}s + 1} \rightarrow \tau_{bc} = \frac{1}{K_{pp}} \rightarrow K_{pp} = 5 [V\delta^{-1}] \quad (5.12)$$

El siguiente controlador valorado consiste en un PID del tipo Interactivo. Este tipo se caracteriza porque cualquier modificación de las constantes temporales afecta a las tres acciones, figura 5.4. En este caso, el término derivativo se empleará para cancelar el polo del proceso. La dinámica en bucle cerrado del proceso controlado será una dinámica de segundo orden que se establecerá por el método de asignación de polos.

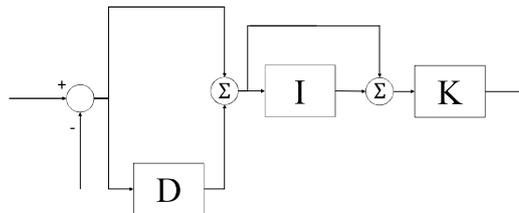


Figura 5.4. Configuración PID Interactivo.

$$C_p(s) = K_{pp} \left(1 + \frac{1}{T_{pi}s} \right) (1 + T_{pd}s) \quad (5.13)$$

$$L_{pi}(s) = C_p(s) \cdot G_{11}(s) = K_{pp} \left(1 + \frac{1}{T_{pi}s} \right) (1 + T_{pd}s) \cdot \frac{1}{s(0,2s+1)} \rightarrow T_{pd} = 0,2$$

$$L_{pi}(s) = K_{pp} \left(1 + \frac{1}{T_{pi}s} \right) \cdot \frac{1}{s} = K_{pp} \frac{T_{pi}s + 1}{T_{pi}s^2} \quad (5.14)$$

$$G_{p,bc}(s) = \frac{K_{pp}(T_{pi}s + 1)}{T_{pi}s^2 + T_{pi}K_{pp}s + K_{pp}} = \frac{\frac{K_{pp}}{T_{pi}}(T_{pi}s + 1)}{s^2 + K_{pp}s + \frac{K_{pp}}{T_{pi}}} \quad (5.15)$$

De la expresión del sistema en lazo cerrado, ecuación 5.15, se observa que este controlador introduce un cero en la respuesta del sistema. Este fenómeno puede ser contrareestado, si perjudica a la dinámica del sistema, con un filtro de constante de tiempo próxima al término integral del controlador, ecuación 5.16. Dicho filtro se ha ajustado manualmente para obtener la mejor respuesta sin producir sobreoscilaciones indeseadas. Por otro lado, como se busca que la dinámica de segundo orden sea críticamente amortiguado, $\zeta \geq 1$ para evitar sobreoscilaciones, los valores de los términos integral y proporcional del controlador dependen exclusivamente del valor que se establezca para la frecuencia natural del sistema en lazo cerrado, ecuación 5.17. En este caso, se ha estimado oportuno que el valor de la frecuencia natural del sistema en lazo cerrado sea $\omega_n = 7$ [rad/s].

$$F_p(s) = \frac{1}{0,38s + 1} \quad (5.16)$$

$$\left. \begin{array}{l} K_{pp} = 2\zeta\omega_n \\ T_{pi} = \frac{K_{pp}}{\omega_n^2} \end{array} \right\} C_p(s) = K_{pp} \left(1 + \frac{1}{T_{pi}s} \right) (1 + T_{pd}s) = 14 \left(1 + \frac{1}{0,2857s} \right) (1 + 0,2s) \quad (5.17)$$

Finalmente, se propone estudiar un tercer tipo de controlador. En este caso se trata de un PI ideal cuyo término integral se empleará para cancelar el polo del lazo interno en bucle cerrado. A continuación se expone el desarrollo teórico para determinar los valores de los parámetros del controlador. Como se puede apreciar en la ecuación 5.20, el resultado de este controlador en bucle cerrado se trata de un sistema de segundo orden críticamente estable, $\zeta = 0$. Esto implica que la respuesta del sistema frente al primer cambio en la referencia lleva al sistema a convertirse en un oscilador cuyo valor medio es la referencia a seguir. Esto hace que este controlador quede descartado directamente para el sistema estudiado.

$$C_p(s) = K_{pp} \left(1 + \frac{1}{T_{pi}s} \right) = K_{pp} \left(\frac{T_{pi}s + 1}{T_{pi}s} \right) \quad (5.18)$$

$$L_{pi}(s) = C_p(s) \cdot G_{11}(s) = K_{pp} \left(\frac{T_{pi}s + 1}{T_{pi}s} \right) \cdot \frac{1}{s(0,2s + 1)} \rightarrow T_{pi} = 0, 2$$

$$L_{pi}(s) = \frac{K_{pp}}{T_{pi}s} \cdot \frac{1}{s} = \frac{K_{pp}}{T_{pi}s^2} \quad (5.19)$$

$$G_{p,bc}(s) = \frac{\frac{K_{pp}}{T_{pi}s^2}}{1 + \frac{K_{pp}}{T_{pi}s^2}} = \frac{\frac{T_{pi}}{K_{pp}}}{s^2 + \frac{T_{pi}}{K_{pp}}} \quad (5.20)$$

En la figuras 5.5 y 5.6, se muestran el resultados de la validación de los controladores propuestos para el lazo externo del sistema *Steer-by-Wire*. Como se puede observar, los resultados obtenidos en ambas gráficas son distintos pese a corresponderse con los mismos controladores. Esto se debe a que en la figura 5.5 no se han implementado saturaciones a las señales de control de cada controlador mientras que en la figura 5.6 sí, siendo esta última una representación más fidedigna del comportamiento que se esperado en el vehículo. Se ha considerado de gran valor realizar esta comparativa para apreciar que lo calculado teóricamente, se corresponde con la evolución del sistema simulado. No obstante, queda patente la necesidad de implementación del mecanismo de saturación puesto que los controladores propuestos excenden las señales de control posibles del sistema, alcanzando el controlador PD la señal de control de 2000 [V].

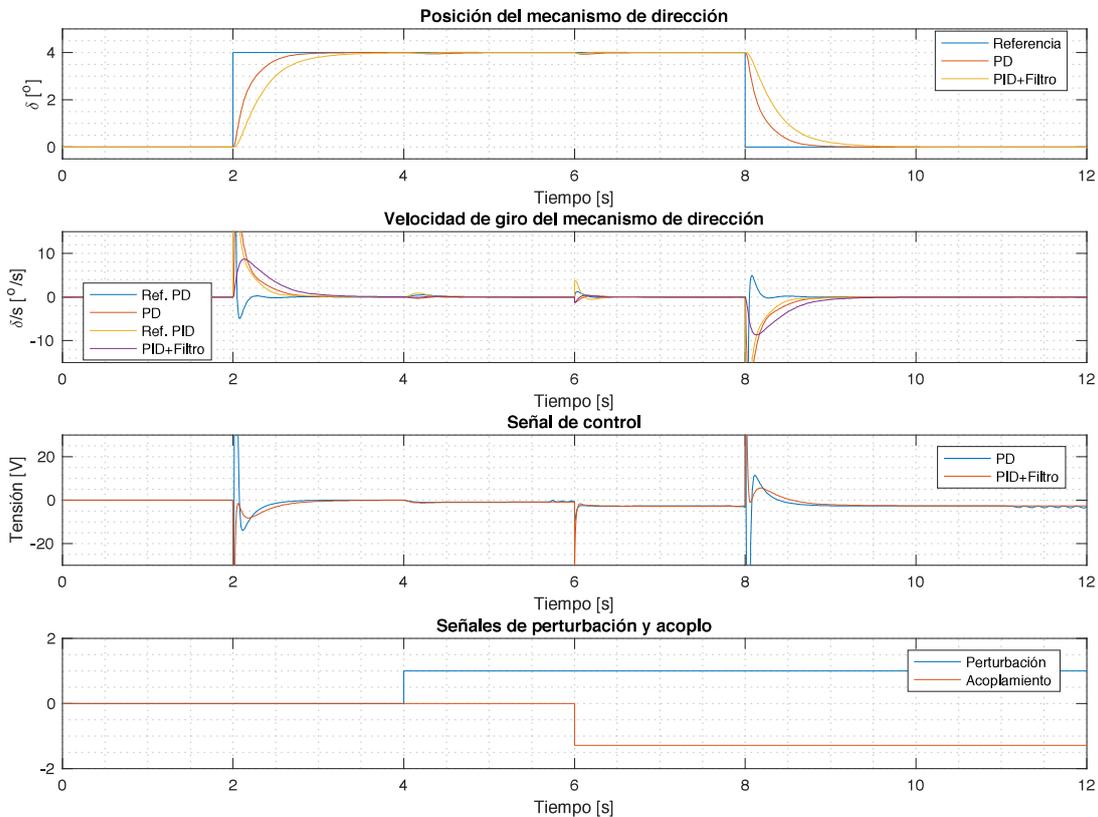


Figura 5.5. Validación de los controladores para la posición del sistema *Steer-by-Wire*.

Los resultados del sistema con las saturaciones implementadas, no cumplen las especificaciones temporales impuestas, siendo la constante de tiempo del sistema resultante del doble de lo diseñado. No obstante, sabido que el origen del incumplimiento reside en la presencia de las saturaciones del sistema, no linealidades físicas y por seguridad, se considera asumible la desviación de 0,2 [s]. Comparando los resultados de los dos controladores, en la tabla 5.1 se muestra el parámetro analítico que determinará la elección del controlador, la exactitud a partir de integral absoluta de error, ecuación 5.21. Ambos presentan un desempeño similar, por lo que la elección de controladores se realiza en función de su implementación en los microcontroladores. Finalmente, se opta por la elección del controlador PD por requerir una implementación más sencilla.

$$Error = 1 - \frac{IAE = \int_0^{\infty} |ref - y| \cdot dt}{\int_0^{\infty} |ref| \cdot dx} \quad (5.21)$$

Controlador	Sin saturación	Con saturación
PD	6,8901 [%]	15,2513 [%]
PID+Filtro	12,7123 [%]	13,6779 [%]

Tabla 5.1. Estudio del error de los controladores propuestos para el sistema *Steer-by-Wire*.

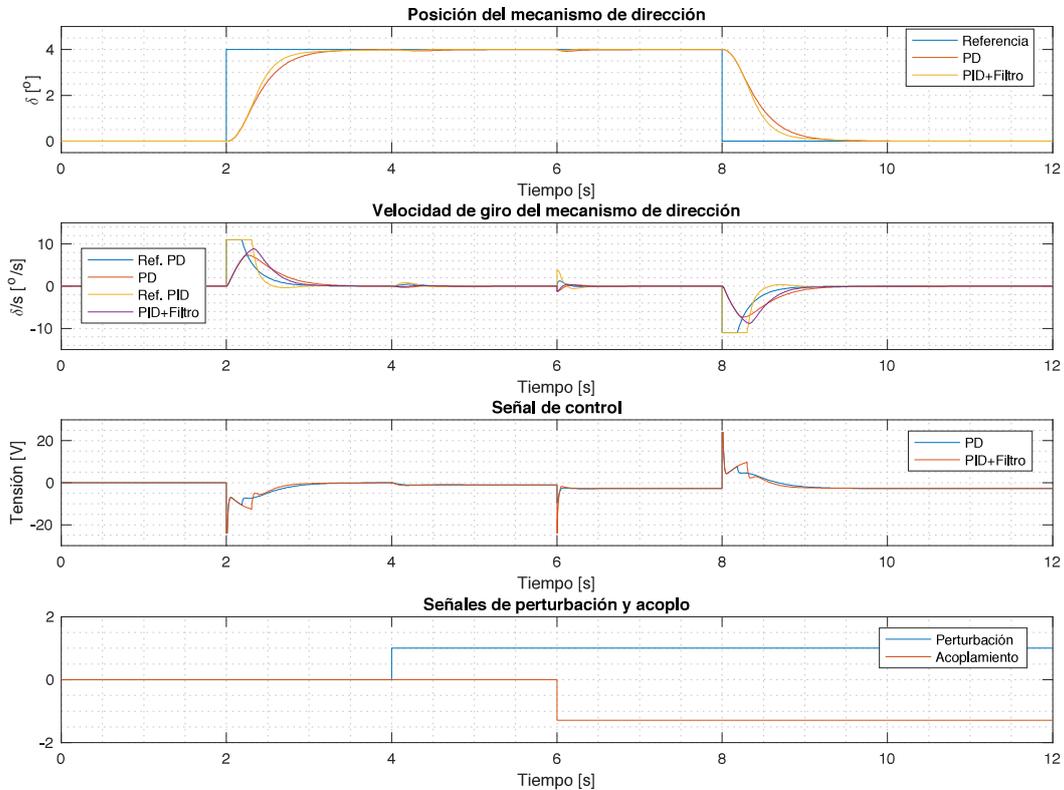


Figura 5.6. Validación de los controladores para la posición del sistema *Steer-by-Wire*.

Al igual que se realizaba para el controlador del lazo interno, se empleará la aproximación trapezoidal (Tustin) para la discretización del controlador. En este caso, en lugar de aplicar las reglas expuestas en la ecuación 5.7, se aplicará directamente la equivalencia entre s y z tal como se expone en la ecuación 5.22.

$$C_p(s) = K_{pp}(1 + T_{pd}s) \rightarrow s \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \rightarrow C_p(z) = K_{pp} \left(1 + T_{pd} \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

$$C_p(z) = \frac{K_{pp}T(1 + z^{-1}) + 2T_{pd}(1 - z^{-1})}{T(1 + z^{-1})} = \frac{9 + z^{-1}}{1 + z^{-1}} \quad (5.22)$$

5.3. Control sistema *Brake-by-wire*

Este sistema presenta el actuador principal del sistema de frenado del vehículo. Su función en la arquitectura global de control del vehículo será servir de apoyo al sistema *Throttle-by-Wire*, desarrollado en la siguiente sección. Esto implica que se tratará de un control basado en eventos, es decir, cuando el vehículo por sí solo no sea capaz de alcanzar la referencia de velocidad en un tramo de frenada, este sistema actuará brevemente para alcanzar el error nulo en el sistema. Puesto que este sistema no se trata de un objetivo principal del vehículo, el diseño del controlador pertinente no se ha realizado con la rigurosidad del resto de sistemas. Como se ha visto en la revisión bibliográfica, los motores de corriente continua se deben controlar como un sistema en cascada, al igual que se ha hecho en el caso del mecanismo de la dirección. No obstante, para el motor que acciona el pedal de freno se propone establecer un solo lazo de control para su posición.

En este caso, para un sistema de tercer orden con un cero, se propone un controlador del tipo PI. El ajuste de parámetros del controlador propuesto se ha llevado a cabo de forma manual hasta que se han alcanzado unas consignas temporales y de sobreoscilación aceptables. Los parámetros de control seleccionados se corresponden con un valor de 0,2 [s] para el término integral y 3 [$V\theta^{-1}$]. Al incluir término integral el controlador, y poseer una saturación a $\pm 24V$ la señal de control, se implementa nuevamente el mecanismo anti-windup. En la figura 5.7 se muestra el esquema del control propuesto.

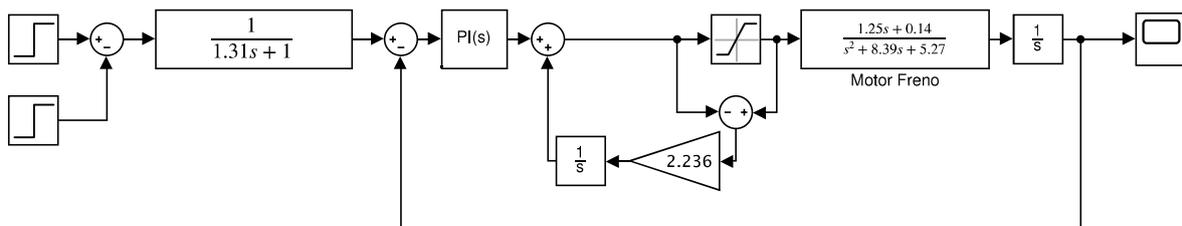


Figura 5.7. Esquema de control propuesto para el sistema *Brake-by-Wire*.

Finalmente, como se puede observar en la figura 5.8, frente a la respuesta original, se produce una respuesta en bucle cerrado de segundo orden con una sobreoscilación del 50% y un tiempo de pico de 2 [s]. Como propuesta para evitar la sobreoscilación, se implementa un filtro en la referencia, como se puede observar en la figura 5.7. La constante de tiempo de dicho filtro se ha ajustado para reducir la sobreoscilación a un valor inferior al 10%. El valor que consigue este efecto es 1,31 [s]. Siguiendo el mismo procedimiento que en el caso del control del sistema *Steer-by-Wire*, se procede a determinar el porcentaje de error que conlleva la elección de cada controlador, tabla 5.10. Pese a que el uso del filtro retrasa el tiempo de pico hasta el valor de 3,1 [s], se determina que es un valor asumible por la mejora que ofrece en exactitud y por tanto se implementa este controlador.

Controlador	Error
PI	38,2506 [%]
PI+Filtro	38,0901 [%]

Tabla 5.2. Estudio del error de los controladores propuestos para el sistema *Brake-by-Wire*.

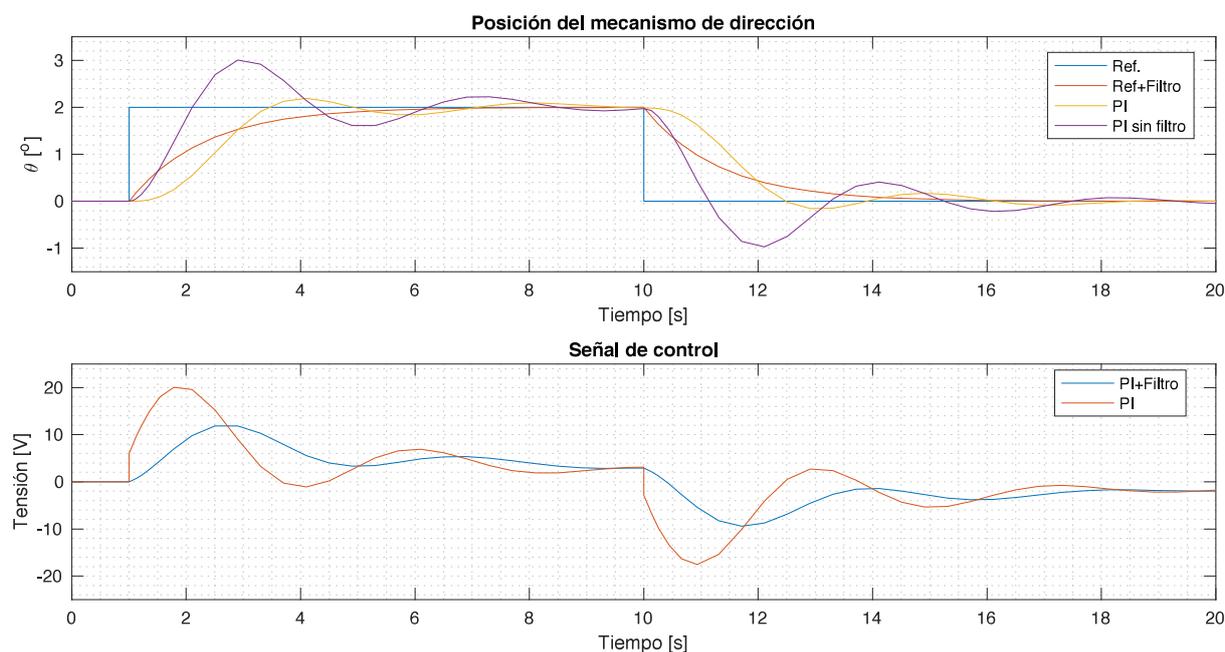


Figura 5.8. Validación del controlador para la posición del sistema *Brake-by-Wire*.

Una vez determinada la validez del controlador propuesto, se procede a discretizar mediante la aproximación trapezoidal (Tustin) el controlador PI, ecuación 5.23 con un tiempo de muestreo de 0,1 [s]. Con dicho tiempo de muestreo y un *M.O.C.* se discretiza de igual manera el filtro de la referencia, ecuación 5.24.

$$\left. \begin{aligned} q_0 &= K_p \left(1 + \frac{T}{2T_i} \right) \\ q_1 &= K_p \left(-1 + \frac{T}{2T_i} \right) \end{aligned} \right\} C_{brake}(z) = \frac{3,75 - 2,25z^{-1}}{1 - z^{-1}} \quad (5.23)$$

$$F_{brake}(s) = \frac{1}{1,31s + 1} \rightarrow F_{brake}(z) = \frac{0,0735}{1 + -0,9265z^{-1}} \quad (5.24)$$

5.4. Control sistema *Throttle-by-wire*

Este sistema, como ya se ha comentado anteriormente, es el encargado de la tracción del vehículo y se ha modelado entorno a 8 puntos de operación. Estos puntos de operación comprenden cuatro rangos de velocidad, dos de marcha directa y dos para la marcha inversa. Dentro de estos rangos de operación, se distinguen entre zonas en las que el vehículo acelera y desacelera, obteniendo en total las 8 dinámicas comentadas inicialmente. Como es de esperar, por las características constitutivas del sistema implementado, solo es posible establecer un control de este sistema dentro del rango de operación de aceleración. En la sección 5.5 se implementará la arquitectura de control completa donde se combina el control para el rango de aceleración, con el control en la zona de frenado realizado por el sistema *Brake-by-Wire*.

Para el desarrollo del sistema de control para las cuatro dinámicas del sistema de aceleración se estudiará la eficacia de un controlador adaptativo cuyos parámetros se ajusten al rango de operación del sistema. Además, también se valorará el desempeño de distintos métodos de sintonía, como Ziegler-Nichols, AMIGO, SIMC o Improved-SIMC.

Al ser un sistema con retardo, el primer parámetro que se debe analizar para comenzar el desarrollo del sistema de control es “*Tiempo de residencia medio*” (T_{rm}). Este parámetro se asocia a la cuantificación de la velocidad de respuesta de un sistema. En la ecuación 5.25 se muestra la expresión general del cálculo de dicho parámetro, siendo $g(x)$ la respuesta impulsional del sistema. En la práctica, cuando se trabaja en el dominio de la frecuencia haciendo uso de la transformada de Laplace, este parámetro se puede obtener más rápidamente como se muestra en la ecuación 5.26, siendo $G'(s)$ la derivada de $G(s)$, f.d.t. del sistema real.

$$T_{rm} = \frac{\int_0^{\infty} t \cdot g(x) \cdot dt}{\int_0^{\infty} g(x) \cdot dt} \quad (5.25)$$

$$\left. \begin{aligned} G(s) &= \int_0^{\infty} e^{-st} \cdot g(x) \cdot dt \\ G'(s) &= \int_0^{\infty} e^{-st} \cdot t \cdot g(x) \cdot dt \end{aligned} \right\} T_{rm} = \frac{\int_0^{\infty} t \cdot g(x) \cdot dt}{\int_0^{\infty} g(x) \cdot dt} = -\frac{G'(0)}{G(0)} \quad (5.26)$$

Este parámetro se evalúa en estado estacionario, $s = 0$, por lo que a efectos prácticos, la ecuación 5.26 en un sistema que presente “x” ceros, “y” polos y tiempo de retardo, el tiempo de residencia medio equivale a la ecuación 5.27. En este caso, este parámetro adopta distintos valores en función del rango de operación, se evaluará para el caso más crítico, es decir, el que presenta menor constante de tiempo, $\tau = 4,86[s]$. El valor resultante obtenido es de $5,77 [s]$ tal como se muestra en la ecuación 5.28.

$$T_{rm} = \sum_{i=1}^x \tau_i - \sum_{i=1}^y \beta_i + t_r \quad (5.27)$$

$$T_{rm} = \tau + t_r = 4,86 [s] + 0,91 [s] = 5,77 [s] \quad (5.28)$$

Este parámetro sirve para el posterior cálculo del “índice de controlabilidad” o “Tiempo de retardo normalizado”, (Q_{rn}). Este parámetro es un indicador de la complejidad de un sistema en su control. Su cálculo se realiza mediante el cociente entre el tiempo de retardo y el tiempo de residencia medio del sistema. El valor de este parámetro debe estar comprendido entre $[0 - 1]$. En la ecuación 5.29 se muestra el valor que le corresponde a este sistema. En el caso de el sistema estudiado, se trata de un sistema con constante de tiempo dominante al ser un valor inferior a 0,5, caso en el que recibe el nombre de sistema balanceado.

$$Q_{rn} = \frac{t_r}{T_{rm}} = \frac{0,91}{5,77} = 0,157 \quad (5.29)$$

Una vez caracterizado el sistema, se procede al desarrollo de los controladores evaluados. Para todos los casos planteados, se buscará un constante de tiempo en bucle cerrado de $3 [s]$. El primer método de sintonía que se plantea es el **método** λ . Este método se trata de una variante del método de cancelación de polos para sistemas con retardo. La principal modificación es realizar la aproximación del retardo a un cero aplicando el desarrollo de Taylor en serie a la función de retardo, ecuación 5.30.

$$G(s) = \frac{k_p}{\tau s + 1} e^{-t_r s} \rightarrow e^{-t_r s} \approx (1 - t_r s) \rightarrow G(s) \approx \frac{k_p (1 - t_r s)}{\tau s + 1} \quad (5.30)$$

A partir de esta aproximación, el proceso de cancelación de polos es equivalente a los ya realizados anteriormente. En la tabla 5.3 se muestran los valores resultantes para un controlador para cada zona de operación.

$$L(s) = K_p \frac{(T_i s + 1)}{T_i s} \cdot \frac{k(1 - t_r s)}{\tau s + 1} \rightarrow T_i = \tau \rightarrow L_p(s) = \frac{K_p k (1 - t_r s)}{\tau s} \quad (5.31)$$

$$G_{bc}(s) = \frac{K_p k (1 - t_r s)}{\tau s + K_p k (1 - t_r s)} = \frac{(1 - t_r s)}{\left(\frac{\tau}{K_p k} - t_r\right) s + 1} \rightarrow \lambda = \frac{\tau - K_p k t_r}{K_p k} \rightarrow K_p = \frac{\tau}{k(t_r + \lambda)} \quad (5.32)$$

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	0,949	Vsm^{-1}
	T_i	6,050	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	0,632	Vsm^{-1}
	T_i	6,050	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	0,507	Vsm^{-1}
	T_i	4,860	s
$2,3 [m/s] < v$	K_p	0,612	Vsm^{-1}
	T_i	4,860	s

Tabla 5.3. Parámetros de controladores por el método λ .

El siguiente método de sintonía es el **Ziegler-Nichols** basado en respuesta a escalón, formulado en 1942 muy presente en la industria actualmente. Esta es una regla de sintonía directa para aquellos sistemas en los que se cumple que el cociente entre el tiempo de retardo y la constante de tiempo debe estar comprendido entre 0,1 – 1,0. En este caso, para las dos constantes de tiempo con las que se trabaja se cumple la condición, ecuación 5.33. En la ecuaciones 5.34 y 5.35 se reflejan la reglas de sintonía de este método para un controlador del tipo PI.

$$0,1 < \frac{t_r}{\tau} < 1,0 \begin{cases} v < 0 [m/s] \rightarrow \frac{0,91}{6,05} = 0,150 \\ v > 0 [m/s] \rightarrow \frac{0,91}{4,86} = 0,187 \end{cases} \quad (5.33)$$

$$K_p = \frac{0,9}{k} \left(\frac{\tau}{t_r} \right) \quad (5.34)$$

$$T_i = 3,33t_r \quad (5.35)$$

En la tabla 5.4 se muestran los parámetros para un controlador para cada zona de operación.

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	4,432	Vsm^{-1}
	T_i	3,030	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	2,442	Vsm^{-1}
	T_i	3,030	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	1,962	Vsm^{-1}
	T_i	3,030	s
$2,3 [m/s] < v$	K_p	2,367	Vsm^{-1}
	T_i	3,030	s

Tabla 5.4. Parámetros de controladores por el método de Ziegler-Nichols.

El siguiente método que se analiza es una modificación que mejora las reglas de sintonía de Ziegler-Nichols, el método de **Cohen-Coon**. Este método está orientado al rechazo de perturbaciones y permite conseguir de forma más eficiente la especificación de que la respuesta decaiga con un factor de un cuarto en amplitud. Sus reglas de sintonía se extienden a controladores del tipo PD. En las ecuaciones 5.37 y 5.38 se muestran las reglas de sintonía para el controlador PI, propuesto en este caso. En la tabla 5.5 se muestran los valores resultantes para cada controlador.

$$T = \frac{t_r}{t_r + \tau} \quad (5.36)$$

$$K_p = 0,9 \left(1 + \frac{0,092T}{1 - T} \right) \cdot \frac{\tau}{t_r k} \quad (5.37)$$

$$T_i = \frac{3,3 - 3,0T}{1,0 + 1,2T} \quad (5.38)$$

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	4,493	Vsm^{-1}
	T_i	3,448	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	2,476	Vsm^{-1}
	T_i	3,448	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	1,995	Vsm^{-1}
	T_i	3,486	s
$2,3 [m/s] < v$	K_p	2,437	Vsm^{-1}
	T_i	3,486	s

Tabla 5.5. Parámetros de controladores por el método de Cohen-Coon.

Otra modificación que tiene su origen en Ziegler-Nichols es el método de **Chien, Hrones y Reswick**. Este método aporta reglas de sintonía con mayor robustez que Ziegler-Nichols y plantea reglas tanto para regulación como para seguimiento, así como para respuestas con sobreoscilación del 20 % como sin ella. En este caso se estudia el problema de seguimiento sin sobreoscilaciones, por lo que las reglas de sintonía que se deben emplear quedan recogidas en las ecuaciones 5.39 y 5.40. En la tabla 5.6 se muestran los valores resultantes para cada controlador.

$$K_p = \frac{0,35}{k} \left(\frac{\tau}{t_r} \right) \quad (5.39)$$

$$T_i = 1,2t_r \quad (5.40)$$

Otro método estudiado en este caso es la sintonía mediante el método **AMIGO**. Este método desarrollado por Karl Aström y Tore Hägglund busca reglas de sintonía que presenten un mayor compromiso entre robustez y rechazo a perturbaciones con la sencillez de Ziegler-Nichols. Este método permite obtener respuestas suaves tanto para seguimiento como en regulación.

Además tiene validez para procesos integradores. En las ecuaciones 5.41 y 5.42 se muestran las reglas de sintonía para controladores tipo PI frente a procesos de primer orden con retardo. En la tabla 5.7 se muestran los valores resultantes para un controlador para cada zona de operación.

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	1,723	Vsm^{-1}
	T_i	1,092	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	0,949	Vsm^{-1}
	T_i	1,092	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	0,763	Vsm^{-1}
	T_i	1,092	s
$2,3 [m/s] < v$	K_p	0.921	Vsm^{-1}
	T_i	1,092	s

Tabla 5.6. Parámetros de controladores por el método de Chien, Hrones y Reswick.

$$K_p = \frac{0,15}{k} + \frac{\tau}{t_r k} \left(0,35 - \frac{\tau t_r}{(t_r + \tau)^2} \right) \quad (5.41)$$

$$T_i = 0,35t_r + \frac{13t_r\tau^2}{\tau^2 + 12\tau t_r + 7t_r^2} \quad (5.42)$$

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	1,275	Vsm^{-1}
	T_i	4,311	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	0,703	Vsm^{-1}
	T_i	4,311	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	0,535	Vsm^{-1}
	T_i	3,706	s
$2,3 [m/s] < v$	K_p	0,645	Vsm^{-1}
	T_i	3,706	s

Tabla 5.7. Parámetros de controladores por el método AMIGO.

El último método de sintonía estudiado es una modificación del método de sintonía IMC (Internal Model Control) de Rivera, Morari y Skogestad [90], realizada por en 2004, conocida como **SIMC** [91]. Esta propuesta se trata de una mejora al método λ , el cual genera una respuesta muy conservadora para procesos de retardo dominante. A continuación, en las ecuaciones 5.43 y 5.44 se muestran las reglas de sintonía para un controlador PI frente a un proceso de primer orden con retardo. En la tabla 5.8 se muestran los valores resultantes para un controlador para cada zona de operación.

$$K_p = \frac{1}{k} \frac{\tau}{\tau_{bc} + t_r} \quad (5.43)$$

$$T_i = \min \{ \tau, 4(\tau_{bc} + t_r) \} \quad (5.44)$$

Zona de operación	Parámetro	Valor	Unidades
$v < -2,1 [m/s]$	K_p	1,146	Vsm^{-1}
	T_i	6,050	s
$-2,1 [m/s] < v < 0 [m/s]$	K_p	0,632	Vsm^{-1}
	T_i	6,050	s
$0 [m/s] < v < 2,3 [m/s]$	K_p	0,507	Vsm^{-1}
	T_i	4,860	s
$2,3 [m/s] < v$	K_p	0,612	Vsm^{-1}
	T_i	4,860	s

Tabla 5.8. Parámetros de controladores por el método SIMC.

Una vez determinados los distintos controladores a evaluar, se procede a implementar el sistema en simulación. Para ello, se procede a implementar los controladores con parámetros variables y la implementación del mecanismo de transferencia sin salto. Este mecanismo se realiza para las secciones en las que el controlador propuesto no tiene acción, es decir los tramos de frenado. En dichos tramos se empleará un control manual cuya señal de control será nula hasta alcanzar la referencia deseada, donde se vuelve a conectar nuevamente el sistema de control.

Como se verá en la próxima sección, este tramo se controlará e intentará reducirse al máximo haciendo uso del control del sistema *Brake-by-Wire*. En la figura 5.9 se muestra la implementación realizada en Simulink para evaluar los distintos controladores. En la figura 5.10, se muestra la comparación entre todos los controladores diseñados para este sistema. Pese a la gran cantidad de controladores probados, se pueden observar claramente las diferencias entre ellos.

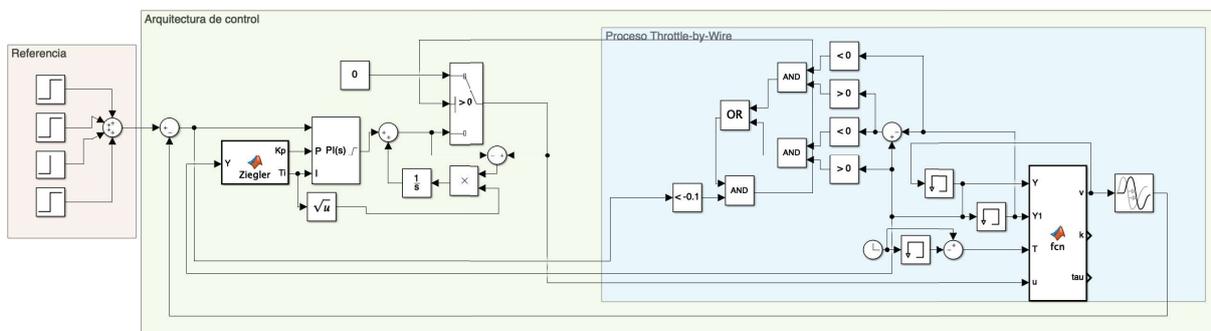


Figura 5.9. Esquema de control propuesto para el sistema *Throttle-by-Wire*.

A priori, el mejor desempeño lo poseen aquellos que poseen señales más conservadoras, como es el caso de Ziegler-Nichols o AMIGO. Eso se debe a que no producen sobreoscilaciones excesivas que provoquen que el control pase a manual. Como contramedida a esto y para reducir el tiempo que el sistema está en manual, se introduce un filtro en la referencia, ecuación 5.45. Este filtro disminuirá la agresividad del controlador reduciendo la sobreoscilación. Su diseño depende del valor de la sobreoscilación sin filtro y el tiempo de pico del sistema. En el cálculo de la constante de tiempo del filtro, M representa el valor de la sobreoscilación en tanto por ciento y t_p es el tiempo de pico del sistema expresado en segundos. En la tabla 5.9 se muestra, como en los sistemas anteriores el error porcentual que implica el uso de cada controlador para los casos en que funcionan solos y con un filtro en la referencia. En la figura 5.11 se muestra el resultado de seleccionar los cuatro mejores controladores y aplicarles su correspondiente filtro en la referencia. Se aprecia claramente la mejora respecto al sistema sin filtros. Se observa que la respuesta del método AMIGO y λ son respuestas más conservadoras y que obtienen una constante de tiempo de 5,2 [s] mientras que por el método de Ziegler-Nichols y Cohen-Coon, se obtiene una respuesta mucho más estable y rápida, $\tau_{bc} = 1,6$ [s], a costa de una señal de control más agresiva que no alcanza la saturación. Además, como se buscaba, no se interrumpe nunca el control automático al no entrar en zona de frenada en ningún momento.

$$F_{ref}(s) = \frac{1}{T_f s + 1} \rightarrow T_f = t_p \sqrt{M^2 - 1} \quad (5.45)$$

Método	Sin filtro	Con filtro
Lambda	8,8539 [%]	11,9894 [%]
Ziegler-Nichols	6,6414 [%]	5,2788 [%]
Cohen-Coon	6,7261 [%]	5,4579 [%]
Chien	14,4856 [%]	26,9356 [%]
AMIGO	10,5077 [%]	11,5321 [%]
SIMC	14,9447 [%]	16,9761 [%]

Tabla 5.9. Estudio del error de los controladores propuestos para el sistema *Throttle-by-Wire*.

A la luz de los resultados obtenidos, se decide que el controlador a implementar un controlador de tipo PI con configuración idealsiguiendo el método de sintonía de Ziegler-Nichols, mecanismo anti-windup, de transferencia sin salto y un filtro en la referencia de constante de tiempo 1,29 [s]. En la ecuación 5.46 se muestra la discretización del controlador PI elegido empleando la aproximación trapezoidal y con un muestreo de 0,1 [s]. Los valores de los parámetros de dicho controlador para cada zona de operación se muestran en la tabla 5.10.

$$\left. \begin{aligned} q_0 &= K_p \left(1 + \frac{T}{2T_i} \right) \\ q_1 &= K_p \left(-1 + \frac{T}{2T_i} \right) \end{aligned} \right\} C_{brake}(z) = \frac{q_0 + q_1 z^{-1}}{1 - z^{-1}} \quad (5.46)$$

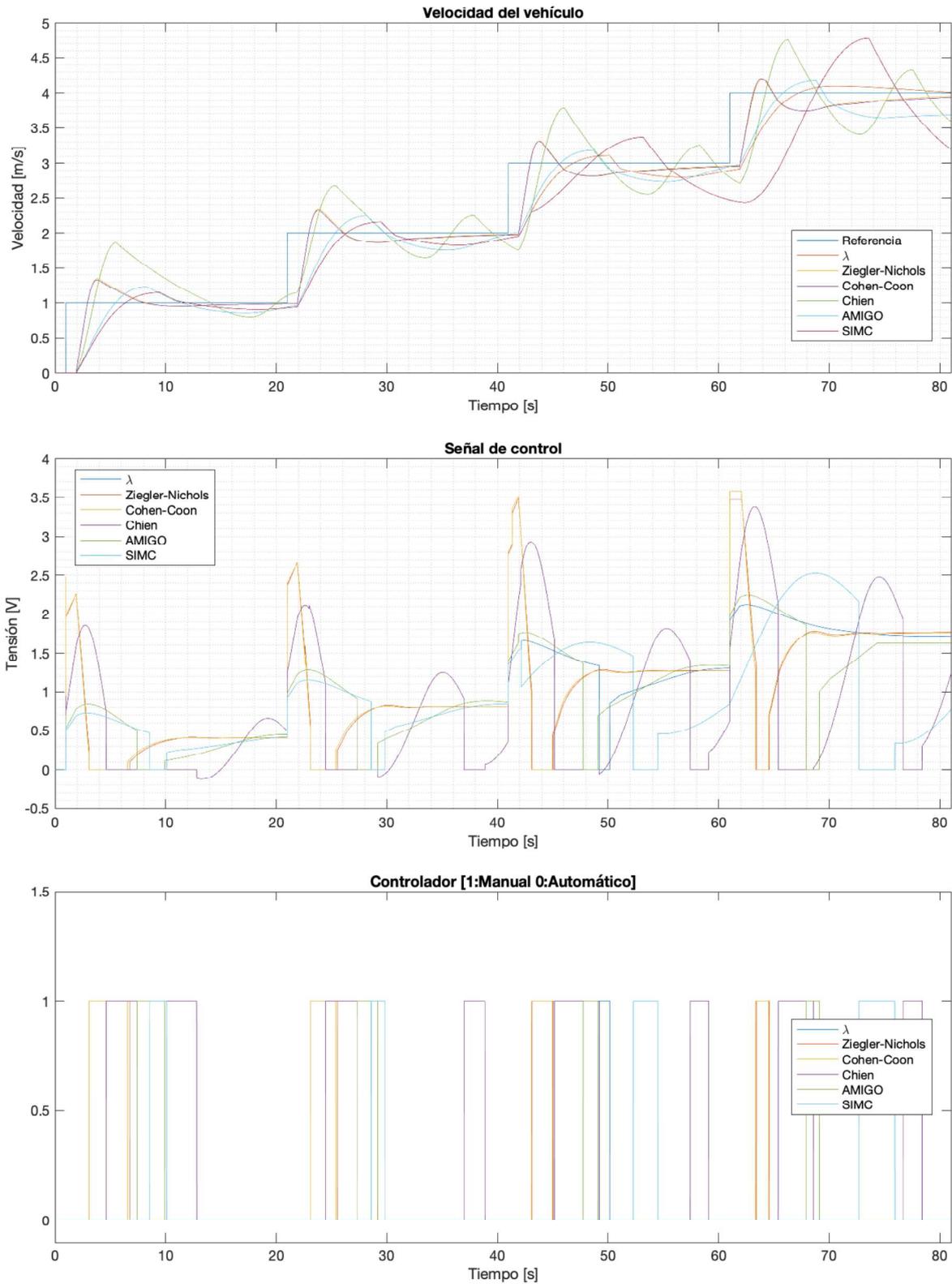


Figura 5.10. Validación de los controladores del sistema *Throttle-by-Wire*.

5.4. CONTROL SISTEMA *THROTTLE-BY-WIRE*

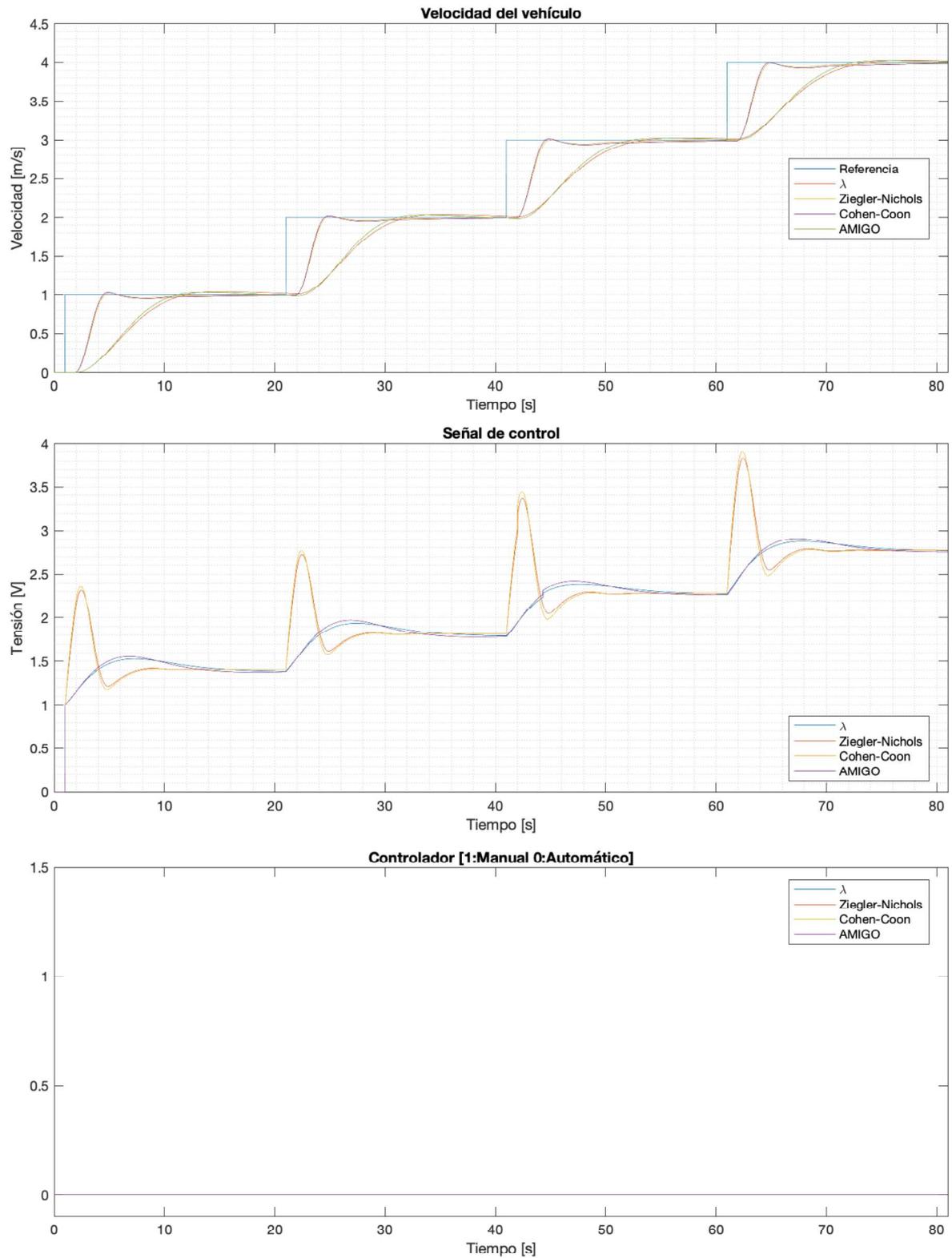


Figura 5.11. Validación de los controladores del sistema *Throttle-by-Wire* con filtro en la referencia.

Marcha	Zona Operación	Controlador	Unidades
Inversa	$v < -2,1 \text{ m/s}$	$\frac{4,505 - 4,358z^{-1}}{1 - z^{-1}}$	$[ms^{-1}V^{-1}]$
	$-2,1 < v < 0 \text{ m/s}$	$\frac{2,482 - 2,402z^{-1}}{1 - z^{-1}}$	$[ms^{-1}V^{-1}]$
Directa	$0 < v < 2,3 \text{ m/s}$	$\frac{1,994 - 1,929z^{-1}}{1 - z^{-1}}$	$[ms^{-1}V^{-1}]$
	$2,3 < v \text{ m/s}$	$\frac{2,406 - 2,328z^{-1}}{1 - z^{-1}}$	$[ms^{-1}V^{-1}]$

Tabla 5.10. Ecuaciones que rigen la dinámica del sistema *Thorttle-by-Wire*.

5.5. Arquitectura de control completa

Una vez determinados los controladores a implementar en el vehículo, se procede a implementarlos previamente en la herramienta de simulación ya presentada en capítulos anteriores. La ventaja de realizar este tipo de implementaciones reside en la versatilidad que ofrecen para evaluar distintas estrategias de control entre sí. En la figura 5.12 se muestra el esquema final del módulo de controladores implementado en simulación.

Se concluye el estudio del sistema de control en el vehículo con una simulación completa de todos los sistemas operando, figura 5.13. El funcionamiento completo de todos los sistemas de control de forma conjunta se engloba bajo el término *Drive-by-Wire*. Las señales de referencia se han introducido en formas de rampas de distintas pendientes puesto que es lo más frecuente en casos de operación normal, siendo las señales abruptas, como los cambios de referencia en escalón más propios de circunstancias excepcionales como volantazos o frenazos para evitar colisiones. Para el sistema de tracción no se han evitado las zonas de frenado para poder observar el apoyo que supone el freno al seguimiento de referencias de velocidad reduciendo la constante de tiempo de los regímenes de frenado. Se aprecia claramente la efectividad del uso del control del freno frente a la respuesta del sistema sin él. No sólo con la reducción de las constantes de tiempo en bucle cerrado, sino suavizando la respuesta en los cambios de consignas.

Como se puede apreciar en los resultados obtenidos, se produce una buena combinación de todos los sistemas en el seguimiento de referencias. A fin de obtener unos resultados cuantitativos, se aplica el mismo análisis que para cada sistema por individual. Se ha obtenido que el error en el sistema de dirección es de un 2,38%. En el caso de la velocidad del vehículo, se ha estudiado el comportamiento del sistema con el freno y sin él. En el primer supuesto se alcanza un error del 43,25% mientras que gracias al uso del freno, el error se reduce un 12,12% hasta un valor de 31,13%. Por tanto, se puede concluir que se ha alcanzado una arquitectura de control para el sistema *Drive-by-Wire* completa y adecuada para el estado actual del vehículo.

5.5. ARQUITECTURA DE CONTROL COMPLETA

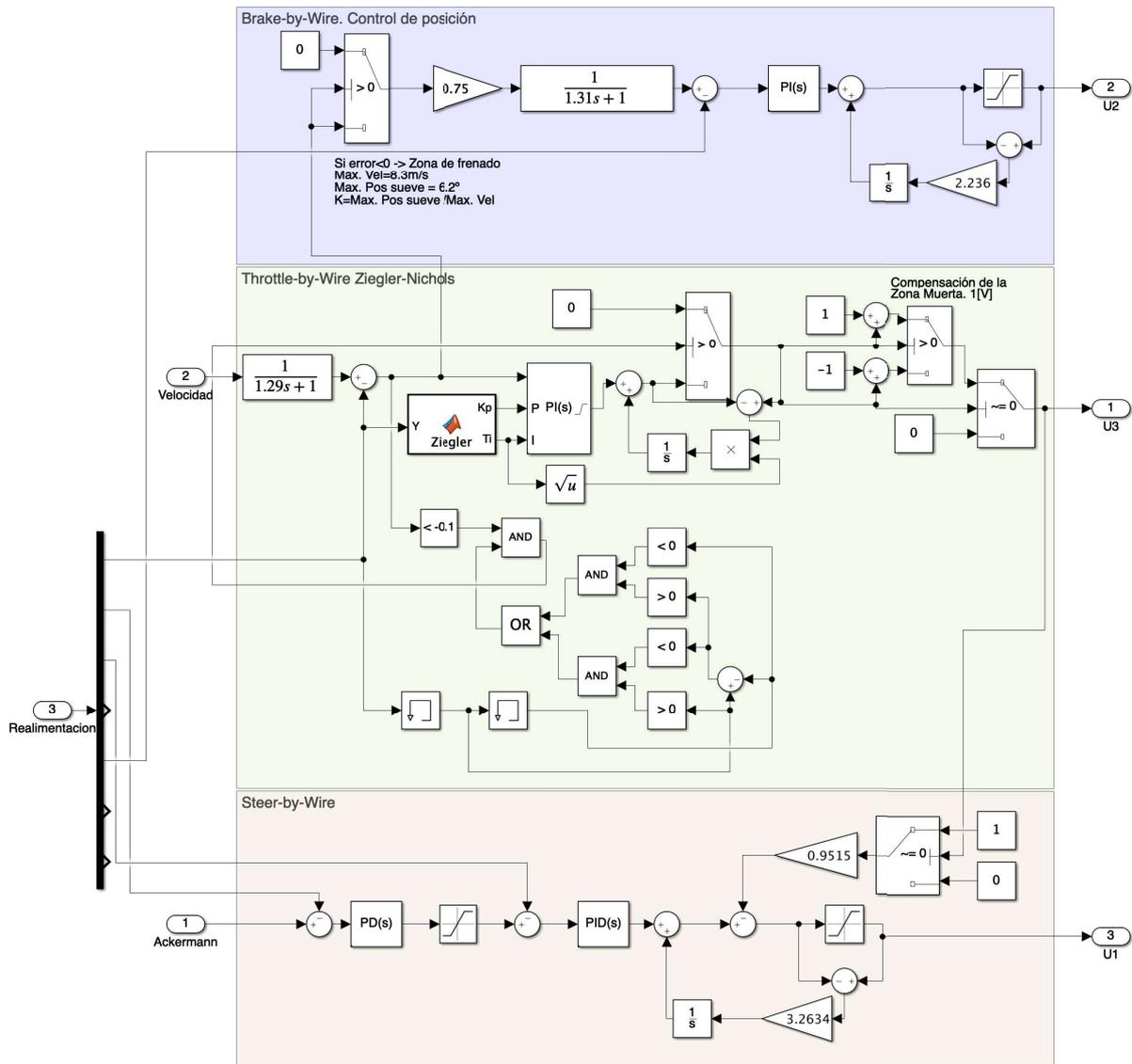


Figura 5.12. Esquema completo de control propuesto para el vehículo UAL-eCARM.

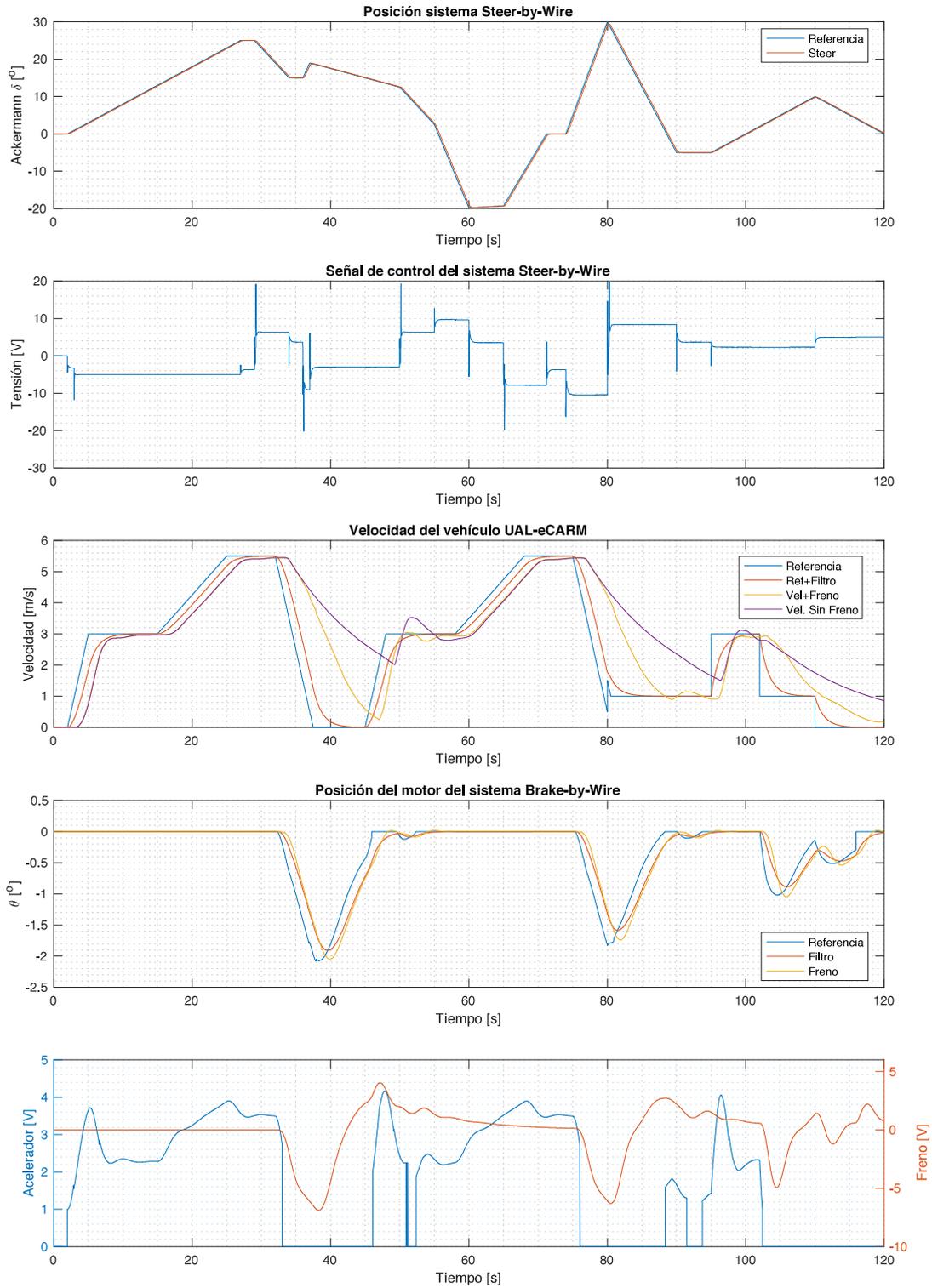


Figura 5.13. Validación de los controladores del sistema *Drive-by-Wire*.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

Tras la conclusión del trabajo realizado en los capítulos anteriores, se puede concluir que los objetivos establecidos inicialmente en la sección 1.2 se han cubierto de forma satisfactoria. Tal como se explica en el capítulo 3, se ha llevado a cabo la mejora de componentes del vehículo necesaria para el incremento del sistema de adquisición de datos. Se ha realizado la instalación de componentes nuevos y la ramificación y reestructuración de la arquitectura firmware y software ya existente para mejorar su comprensión y desempeño.

Respecto al modelado del sistema *Drive-by-Wire* del vehículo, se ha mejorado sustancialmente el conocimiento del mismo. El sistema de la dirección era el más elaborado hasta el momento. No obstante, al realizarse un análisis más exhaustivo se ha determinado, que bien por el la implementación software que tenía anteriormente o por el método empleado en la identificación del sistema, el retardo que hacía al sistema necesitar un Predictor de Smith para la realimentación del lazo interno, se ha identificado ahora como una zona muerta. De igual forma, la nueva identificación realizada ha mostrado un comportamiento más propio de un segundo orden en el lazo interno al producirse sobreoscilaciones y el acoplamiento existente con el sistema de tracción del vehículo. En este mismo sistema se ha podido modelar el consumo de corriente, por lo que permite estimar su consumo frente a un estudio energético del vehículo.

En el caso del sistema *Throttle-by-Wire*, el principal objetivo era obtener un modelo fiable del sistema, ya que no se pudo realizar en [14] por limitaciones técnicas, aunque ya se apreciaba la no linealidad modelada en esta ocasión. Se ha podido modelar con éxito mediante 8 f.d.t. el comportamiento del vehículo entorno a 4 puntos de operación y dos regímenes de movimiento. Los modelos obtenidos han sido de primer orden con un retardo no dominante. En el caso del sistema *Brake-by-Wire*, al tratarse de un sistema completamente nuevo en el vehículo y cuyo funcionamiento va a ser reducido, se puede considerar que se ha alcanzado con éxito un primer modelo sobre el que trabajar.

Todos esta información se ha podido integrar en un modelo de Simulink que permite trabajar con él y probar de forma más simple distintas estrategias y comparar resultados. A partir de los modelos obtenidos, se procedido a diseñar una arquitectura de control completa sobre el sistema *Drive-by-Wire* del vehículo. A causa del cambio significativo del retardo en el caso de la dirección, se han remplazado los controladores PI y PID con Predictor de Smith por controladores PID y PD respectivamente diseñados por los métodos de asignación de polos y cancelación polo-cero. Se han estudiado diferentes configuraciones de controladores hasta alcanzar una que ofrezca resultados adecuados. En el caso del sistema de freno, al tratarse de un sistema de alto orden se ha optado por un ajuste manual hasta alcanzar referencias temporales de seguimiento adecuadas. El estudio de arquitecturas de control complejas, al tratarse de un sistema altamente no lineal, se han considerado fuera del alcance del presente trabajo por suponer una carga excesiva más propia de otro trabajo exclusivo sobre el sistema en cuestión.

En el caso del sistema *Throttle-by-Wire*, el estudio realizado sobre su arquitectura de control ha sido el más exhaustivo. Se ha implementado un control adaptativo a los puntos donde se había linealizado. Se han comparado hasta seis métodos de sintonía para sistemas de primer orden con retardo no dominante. Finalmente, el método implementado ha sido el de Ziegler-Nichols pese a ser el más conservador por ofrecer una mejor respuesta y mayor rapidez en la respuesta. No obstante, para reducir en la medida de lo posible las zonas sin control por estar en desaceleración, se ha considerado imprescindible la implementación de un filtro en la referencia que garantice una señal siempre ascendente sin comprometer las especificaciones temporales.

6.2. Trabajos futuros

A pesar de que los objetivos planteados se han alcanzado, aún hay muchas posibilidades de trabajos futuros relacionados con el trabajo desarrollado. Aplicable a los tres sistemas estudiados, sería de gran interés realizar un modelado basado en primeros principios y compararlo con el obtenido hasta ahora mediante “caja negra”. Otro aspecto susceptible de ser estudiado y poco tratado hasta ahora sería realizar un estudio energético. No solo estudiando las placas solares y las baterías, sino los consumos de los distintos componentes instalados en el vehículo y el efecto que tiene el consumo global sobre las dinámicas de los distintos actuadores. Este efecto se conoce que ocurre debido a que se es claramente apreciable que con el descenso de carga en las baterías, la velocidad del vehículo se reduce considerablemente.

Otro trabajo de gran relevancia sobre el vehículo es el desarrollo de una interfaz gráfica en ROS que permita la visualización de información del vehículo de una forma más intuitiva. Esta aplicación se hace necesaria para que mayor número de investigadores puedan hacer uso del vehículo sin depender exclusivamente de los principales desarrolladores del mismo. En la misma línea, poder realizar una aplicación en herramientas como Matlab que permita conectarse al vehículo de forma y remota o hacer ensayos de mayor complejidad, sería de gran utilidad.

Volviendo al modelado y control de los sistemas trabajados, como ya se ha comentado anteriormente, para un mejor aprovechamiento del sistema de freno, se debe realizar un modelado más exhaustivo y complejo, con su correspondiente estructura de control que permita alcanzar resultados de forma más eficiente. Un trabajo de gran interés que aún no se ha podido realizar es la creación de un nodo en ROS para la identificación de parámetros en línea, con un enfoque en el desarrollo de controladores autoajustables más avanzados para los distintos sistemas del vehículo. De esta forma se podría contrarrestar más eficazmente, y modelar incluso, el efecto de agentes externos incontrolables a priori sobre el vehículo. Un ejemplo puede ser la identificación del tipo de superficie de operación a partir de los parámetros del sistema de dirección, sobre el cual se ha demostrado en este trabajo la fuerte dependencia del rozamiento.

Finalmente, el trabajo que se persigue realizar como fin último es la navegación autónoma del vehículo. Ya se han asentado las bases de control completo a bajo nivel de la pirámide de la automatización. No obstante, hasta alcanzar la completa autonomía del vehículo es necesario la integración de sensores de mayor complejidad y precisión, como GPS o láseres 2D y 3D, en el sistema que le permitan reaccionar frente a modificaciones temporales del entorno, como cruces con personas u otros vehículos. De igual forma, también es necesario el estudio y realización de una arquitectura de control capaz de generar y seguir trayectorias.

Anexo A

Configuración actual Curtis

Parámetro	Valor	Mín.	Máx.	Parámetro	Valor	Mín.	Máx.
M1 MAIN C/L	400	100	400	REV SPEED	8.0	4.0	16.0
M2 MAIN C/L	400	100	400	REV ACCEL	4.0	0.6	5.0
REGEN C/L	350	50	400	REV DECEL	2.0	0.2	10.0
PLUG C/L	250	50	400	REV BRAKE MIN	20	10	100
AUX DRVR MODE	1	0	2	REV BRAKE MAX	30	10	100
AUX PULL IN	1	1	3	REV BRAKE MAP	22.0	20.0	80.0
AUX HOLDING	1	1	3	REV BRK START	4.0	0.0	40.0
EMB DELAY	2.5	0.0	5.0	REV BRAKE END	8.0	0.0	40.0
FLD BRAKE	1	0	1	M2 FWD SPEED	20.0	5.6	40.0
FLD BRAKE SPD	0.0	0.0	6.0	M2 ACCEL	0.6	0.6	5.0
FLD BRAKE MAX	0.0	0.0	50.0	M2 DECEL	2.0	0.6	10.0
FLD BRAKE RATE	0	0	6	M2 BRAKE MIN	10	10	100
EMB SPD CHECK	1	0	1	M2 BRAKE MAX	100	10	100
EMB SPD VALUE	0.4	0.0	40.0	M2 BRAKE MAP	22.0	20.0	80.0
RPM TO SPEED	165	20	255	M2 BRAKE START	5.0	0.0	40.0
TACHO POLES	8	4	20	M2 BRAKE END	14.0	0.0	40.0
MODE AFTER KEY	1	0	1	OVERVOLTAGE	62.0	40.0	65.0
WARNING OPTION	3	0	7	LOW VOLTAGE	37.5	27.0	42.0
M1 FWD SPEED	20.0	4.0	40.0	THROTTLE MAP	50.0	20.0	80.0
M1 ACCEL	4.0	0.6	5.0	THROTTLE 0%	0.5	0.0	2.0
M1 DECEL	2.0	0.2	10.0	THROTTLE 100%	4.5	2.1	5.0
M1 BRAKE MIN	10	10	100	THROT FAULT LO	0.2	0.0	2.0
M1 BRAKE MAX	30	10	100	THROT FAULT HI	5.0	3.0	5.0
M1 BRAKE MAP	22.0	20.0	80.0	M1 FWD FLD MIN	3.5	2.0	15.0
M1 BRAKE START	14.0	0.0	40.0	REV FIELD MIN	3.5	2.0	15.0
M1 BRAKE END	18.0	0.0	40.0	M2 FWD FLD MIN	5.0	2.0	15.0

Tabla A.1. Parámetros del controlador Curtis durante el modelado. I.

Parámetro	Valor	Mín.	Máx.	Parámetro	Valor	Mín.	Máx.
FLD MAP START	100	0	200	NEG FLD RAMP	35.0	20.0	80.0
FIELD RAMP	40.0	20.0	80.0	NEG FIELD MAX	40.0	10.0	50
FIELD MAX	40.0	20.0	50.0	NEG FLD MAP END	300	200	400
FIELD MAP END	350	200	400	WALKAWAY C/L	21.5	6.0	25.0
NEG FLD MAP START	30	0	300	KSI SRO ENABLE	0	0	1

Tabla A.2. Parámetros del controlador Curtis durante el modelado. II.

Anexo B

Radio de Curvatura del vehículo.

En este apéndice se muestra el código empleado para la calibración del radio de curvatura del sistema así como las gráficas resultantes obtenidas donde se aprecian los resultados gráficamente.

```
% Determinacion del radio de curvatura
% Lectura de datos
load("Data.mat");
Encoder_Abs(:,1) = Encoder_Abs1(:,1)/10^9;
Encoder_Abs(:,2) = Encoder_Abs1(:,3);
% Representacion de la trayectoria del ensayo
figure('Name','Trayectoria')
plot3(vehiclepathENU(:,2),vehiclepathENU(:,3),vehiclepathENU(:,4))
title('Trayectoria del ensayo')
xlabel(' [m] ')
ylabel(' [m] ')
zlabel(' [m] ')
zlim([-10 10])
axis equal
% Correccion continuidad encoder absoluto
for a=2:length(Encoder_Abs(:,2))
    if Encoder_Abs(a,2) - Encoder_Abs(a-1,2) > 1000
        Encoder_Abs(a,2) = Encoder_Abs(a,2) - 1024;
    else
        if Encoder_Abs(a,2) - Encoder_Abs(a-1,2) < -1000
            Encoder_Abs(a,2) = Encoder_Abs(a,2) + 1024;
        else
            Encoder_Abs(a,2) = Encoder_Abs(a,2);
        end
    end
end
end
% Ajuste de la escala de tiempo
Encoder_Abs(:,1) = Encoder_Abs(:,1) - min(Encoder_Abs(:,1));
vehiclepathENU(:,1) = vehiclepathENU(:,1) - min(Encoder_Abs(:,1));
Ackermann(:,1) = Encoder_Abs(:,1);
Ackermann(:,2) = (Encoder_Abs(:,2)-Encoder_Abs(1,2));
```

Figura B.1. Calibración del radio de curvatura con el mecanismo de dirección (1/3).

```

% Búsqueda de datos y emparejamiento
j=2;
puntos(1,1)=0;
for i=1:length(vehiclepathENU)
    for k=1:length(Ackermann)
        if (abs(vehiclepathENU(i,1)-Ackermann(k,1)) < 0.01) && (vehiclepathENU(i,1) ~=
            puntos(j-1,1))
            puntos(j,1:7)=vehiclepathENU(i,1:7);
            puntos(j,8)=Ackermann(k,2);
            puntos(j-1,9)=puntos(j,1)-puntos(j-1,1);
            j=j+1;
        end
    end
end
figure('Name','Datos_iniciales')
plot(vehiclepathENU(:,1),vehiclepathENU(:,6),'bo'); hold on
plot(Ackermann(:,1),Ackermann(:,2),'r'); hold on
plot(puntos(:,1),puntos(:,8),'kx')
% Definición de variables
N = length(puntos);
theta = zeros(N,1);
T = zeros(4,4,N);
p1Tp2 = zeros(4,4,N-1);
Rc = zeros(N-1,1);
delta_theta = zeros(N-1,1);
% Determinación de las matrices de transformación
for i=1:N
    theta(i) = puntos(i,5);
    T(:, :, i) = [cos(theta(i)) -sin(theta(i)) 0 puntos(i,2);
                 sin(theta(i))  cos(theta(i)) 0 puntos(i,3);
                 0              0             1 puntos(i,4);
                 0              0             0 1];
end
last_i=1;
xx=[];
yy=[];
giro = [];
for j=1:N
    p1Tp2(:, :, j)=T(:, :, last_i)\T(:, :, j);
    delta_theta(j,1) = theta(j)-theta(last_i);
    Ax=p1Tp2(1,4,j);
    Rc(j) = Ax/sin(delta_theta(j,1));
    if (abs(Ax)<1.0) % keep accum.
    else
        if abs(delta_theta(j,1)*(180/pi)) < 21.5
            last_i = j;
            enc= puntos(j,8);
            yy=[yy Rc(j)];
            % Encoder abs:
            xx=[xx enc];
            giro = [giro delta_theta(j,1)];
        end
    end
end
end
end

```

Figura B.2. Calibración del radio de curvatura con el mecanismo de dirección (2/3).

```

%% Representacion
figure('Name','Radio_de_Curvatura','NumberTitle','off','Units','centimeters','
    Position',[4 4 29.7 14])
plot(xx,giro*(180/pi),'.')
xlabel('Pulsos_encoder')
ylabel('Ackermann_[]')
grid minor

figure('Name','Radio_de_Curvatura','NumberTitle','off','Units','centimeters','
    Position',[4 4 29.7 14])
plot(xx,1./yy,'bx');
xlabel('\Delta_theta_[]');
ylabel('Rc[m]');
title('Correlacion_curvatura_/delta')
grid minor

```

Figura B.3. Calibración del radio de curvatura con el mecanismo de dirección (3/3).

En la figura B.4 se muestra la trayectoria del ensayo empleada para el cálculo del radio de curvatura. En la figura B.5 se muestran en negro los puntos en los que se registra simultáneamente, según el reloj del ordenador, los datos GPS (azul) y la posición del mecanismo de dirección (rojo). De todos los datos sincronizados, sólo son realmente útiles aquellos en los que se registren una serie de posiciones del mecanismo con valor constante, y del GPS variables, puntos rojos. Finalmente, con los puntos válidos obtenidos, se puede determinar la curvatura frente a los pulsos del encoder, figura B.7, determinar parámetros como el offset del mecanismo y determinar la correlación entre el ángulo de Ackermann del vehículo y la posición del encoder absoluto que indica la posición del mecanismo.

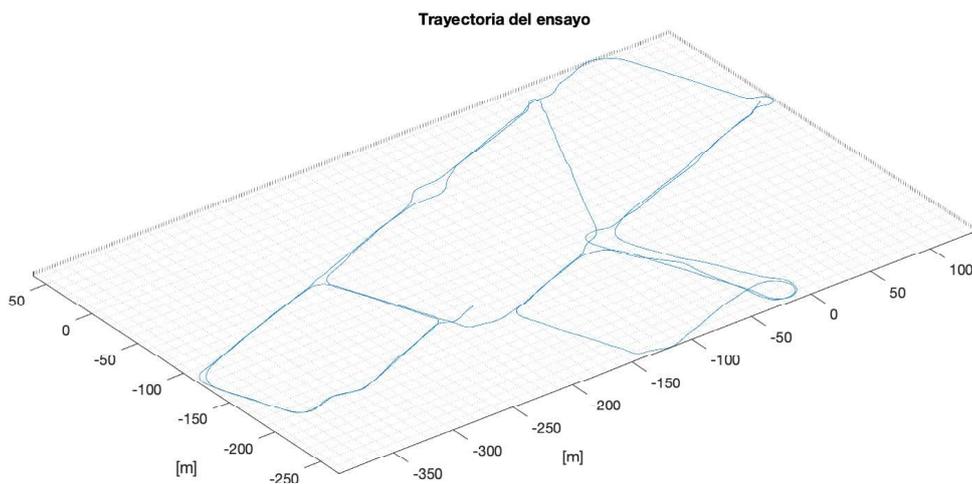


Figura B.4. Trayectoria estudiada.

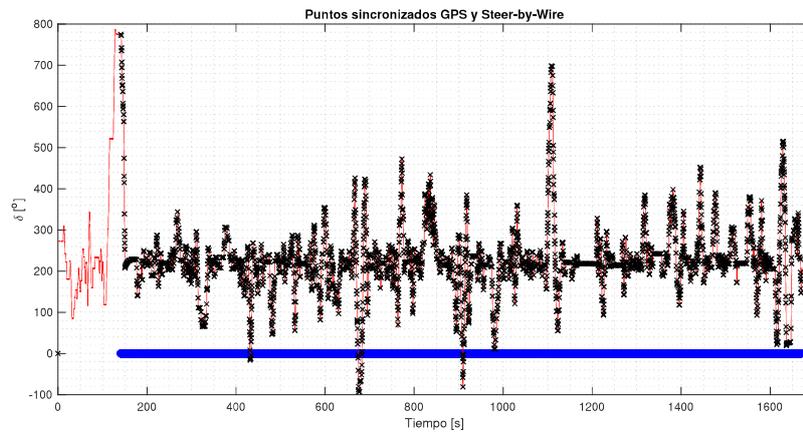


Figura B.5. Puntos donde se sincroniza la señal GPS con el registro del sistema Steer-by-Wire.

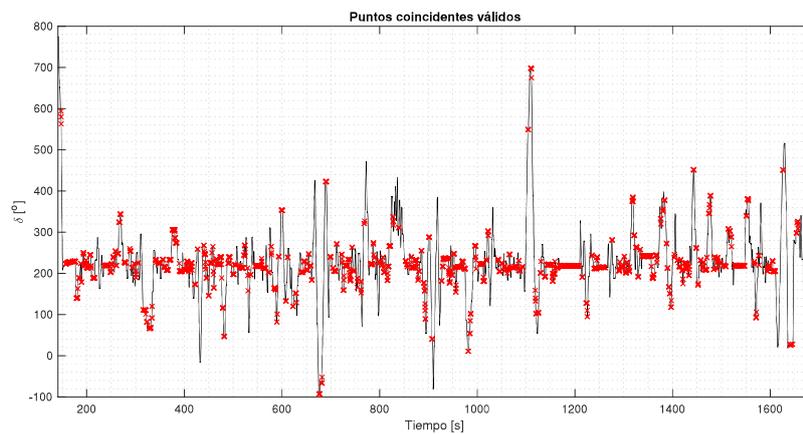


Figura B.6. Puntos coincidentes válidos para el cálculo.

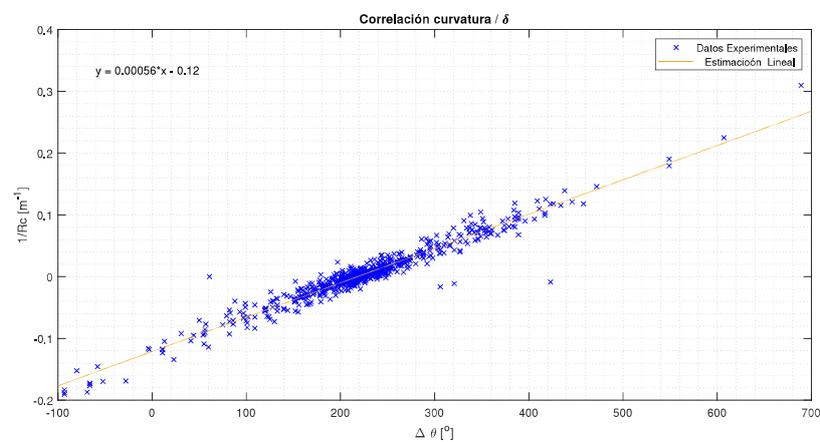


Figura B.7. Correlación entre el radio de curvatura y la posición del mecanismo de dirección.

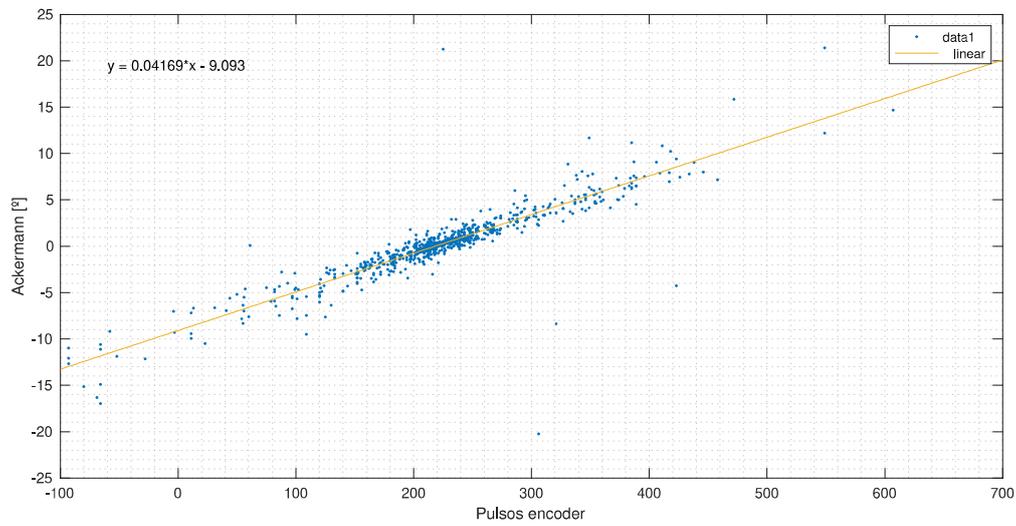


Figura B.8. Correlación entre el ángulo de Ackermann y la posición del mecanismo de dirección.

Bibliografía

- [1] P. Xu, A. Abidli, P. Bonnifait, E. Hery, and G. Dherbomez, “System Architecture of a Driverless Electric Car in the Grand Cooperative Driving Challenge,” *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 1, pp. 47–59, 2018.
- [2] T. Kessler, J. Bernhard, M. Buechel, K. Esterle, P. Hart, D. Malovetz, M. T. Le, F. Diehl, T. Brunner, and A. Knoll, “Bridging the gap between open source software and vehicle hardware for autonomous driving,” *arXiv preprint arXiv:1905.02980*, 2019.
- [3] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving,” *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, 2015.
- [4] G. Howser and B. McMillin, “Modeling and reasoning about the security of drive-by-wire automobile systems,” *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 127–134, 2012.
- [5] J. J. Gil, I. Díaz, P. Ciáurriz, and M. Echeverría, “New driving control system with haptic feedback Design and preliminary validation tests,” *Transportation Research Part C: Emerging Technologies*, vol. 33, pp. 22–36, 2013.
- [6] A. Baviskar, J. R. Wagner, D. M. Dawson, D. Braganza, and P. Setlur, “An adjustable steer-by-wire haptic-interface tracking controller for ground vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 546–554, 2009.
- [7] S. Haggag, S. Cetinkunt, D. Alstrom, and A. Egelja, “Modeling, control, and validation of an electro-hydraulic steer-by-wire system for articulated vehicle applications,” *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 6, pp. 688–692, 2006.
- [8] P. Sinha, “Architectural design and reliability analysis of a fail-operational brake-by-wire system from iso 26262 perspectives,” *Reliability Engineering & System Safety*, vol. 96, no. 10, pp. 1349–1359, 2011.
- [9] S. Yim, “Fault-tolerant yaw moment control with steer - and brake-by-wire devices,” *International Journal of Automotive Technology*, vol. 15, no. 3, pp. 463–468, 2014.

- [10] P. Song, M. Tomizuka, and C. Zong, “A novel integrated chassis controller for full drive-by-wire vehicles,” *International Journal of Vehicle Mechanics and Mobility*, vol. 53, no. 2, pp. 215–236, 2015.
- [11] T. Drage, J. Kalinowski, and T. Braunl, “Integration of Drive-by-Wire with Navigation Control for a Driverless Electric Race Car,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 23–33, 2014.
- [12] F. Todeschini, S. Formentin, G. Panzani, M. Corno, S. M. Savaresi, and L. Zaccarian, “Non-linear Pressure Control for BBW Systems via Dead-Zone and Antiwindup Compensation,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1419–1431, 2016.
- [13] H. Zheng, S. Ma, and X. Na, “Design of a variable steering ratio for steer-by-wire vehicle with a joystick,” *Advances in Mechanical Engineering*, vol. 9, no. 11, pp. 1–14, 2017.
- [14] F. J. Mañas-Álvarez, *Caracterización y control de sistema drive-by-wire en vehículo eléctrico*. PhD thesis, Universidad de Almería, 2017.
- [15] “Changzhou Greenland Vehicle Co.” [Consulta el 2-05-2017]. Disponible en: http://www.gl-ev.com/en/pro_show.php?Id=155.
- [16] J. Ramos Teodoro, *Modelado y control de la dirección de un vehículo eléctrico*. PhD thesis, Universidad de Almería, 2014.
- [17] J. L. Torres-Moreno, *Estudio del comportamiento dinámico de un vehículo eléctrico mediante simmechanics*. PhD thesis, Universidad de Almería, 2011.
- [18] A. P. Kurucz, “Modelado del sistema de almacenamiento de energía y del controlador del motor para un vehículo eléctrico tesur,” 2015.
- [19] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [20] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [21] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, “Convergence of MANET and WSN in IoT urban scenarios,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558–3567, 2013.
- [22] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, “Smart cities and the future internet: Towards cooperation frameworks for open innovation,” *Lecture Notes in Computer Science*, vol. 6656, pp. 431–446, 2011.
- [23] M. L. Tuballa and M. L. Abundo, “A review of the development of Smart Grid technologies,” *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 710–725, 2016.

-
- [24] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. A. Pardo, and H. J. Scholl, "Understanding smart cities: An integrative framework," *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 2289–2297, 2012.
- [25] C. F. Calvillo, A. Sanchez-Miralles, and J. Villar, "Synergies of electric urban transport systems and distributed energy resources in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2445–2453, 2018.
- [26] X. G. Tan, W. Hui, and Q. M. Li, "Multi-port topology for composite energy storage and its control strategy in micro-grid," *7th International Power Electronics and Motion Control Conference - ECCE Asia*, vol. 1, pp. 351–355, 2012.
- [27] C. F. Calvillo, A. Sanchez-Miralles, and J. Villar, "Energy Management and Planning in Smart Cities," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 273–287, 2016.
- [28] J. L. Torres-Moreno, A. Gimenez-Fernandez, M. Perez-Garcia, and F. Rodriguez, "Energy management strategy for micro-grids with PV-battery systems and electric vehicles," *Energies*, vol. 11, no. 3, 2018.
- [29] F. Mwasilu, J. J. Justo, E. K. Kim, T. D. Do, and J. W. Jung, "Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration," *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 501–516, 2014.
- [30] S. F. Tie and C. W. Tan, "A review of energy sources and energy management system in electric vehicles," *Renewable and Sustainable Energy Reviews*, vol. 20, pp. 82–102, 2013.
- [31] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE world forum on internet of things (WF-IoT)*, pp. 241–246, IEEE, 2014.
- [32] C. Fiori, K. Ahn, and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Applied Energy*, vol. 168, pp. 257–268, 2016.
- [33] K. M. Tan, V. K. Ramachandramurthy, and J. Y. Yong, "Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 720–732, 2016.
- [34] C. Zhou, K. Qian, M. Allan, and W. Zhou, "Modeling of the cost of EV battery wear due to V2G application in power systems," *IEEE Transactions on Energy Conversion*, vol. 26, no. 4, pp. 1041–1050, 2011.
- [35] A. Ahmadian, M. Sedghi, A. Elkamel, M. Fowler, and M. Aliakbar Golkar, "Plug-in electric vehicle batteries degradation modeling for smart grid studies: Review, assessment and conceptual framework," *Renewable and Sustainable Energy Reviews*, vol. 81, no. June 2017, pp. 2609–2624, 2018.

- [36] M. V. Noroña and M. F. Gómez, “Desarrollo e innovación de los sistemas mecatrónicos en un automóvil: una revisión,” *Enfoque UTE*, vol. 10, no. 1, pp. 117–127, 2019.
- [37] F. Seidel, “X-by-wire,” in *Operation Systems, Chemnitz University of Technology, In seminar Transportation Systems*, Citeseer, 2009.
- [38] J. Ni, J. Hu, and C. Xiang, “Robust control in diagonal move steer mode and experiment on an x-by-wire ugv,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 572–584, 2019.
- [39] D. Zhao, L. Chu, N. Xu, C. Sun, and Y. Xu, “Development of a cooperative braking system for front-wheel drive electric vehicles,” *Energies*, vol. 11, no. 2, 2018.
- [40] Y. Kim, H. Fujimoto, and Y. Hori, “Realization of steer-by-wire system for electric vehicles using caster wheels and independent driving motors,” in *8th International Conference on Power Electronics-ECCE Asia*, pp. 1688–1695, IEEE, 2011.
- [41] A. Á. Lara Rivero, “De sistema mecánico a sistema tecnológico complejo. El caso de los automóviles,” *Contaduría y Administración*, vol. 59, no. 2, pp. 11–39, 2014.
- [42] J. Ni, J. Hu, and C. Xiang, “Control-configured-vehicle design and implementation on an x-by-wire electric vehicle,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3755–3766, 2018.
- [43] J. S. Cheon, “Brake By Wire System Configuration and Functions using Front EWB (Electric Wedge Brake) and Rear EMB (Electro-Mechanical Brake) Actuators,” *SAE Technical Paper Series*, vol. 1, 2010.
- [44] C. R. Kelber, R. S. Dreger, G. K. Comes, D. Webber, J. Schirmbeck, R. H. Netto, and D. A. Borges, “Cell-phone guided vehicle, an application based on a drive-by-wire automated system,” *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 645–649, 2003.
- [45] D. A. Ciencia, “¿A quién mataría un coche autónomo en un accidente inevitable?” [Consulta el 30-06-2019]. Disponible en: https://www.abc.es/ciencia/abci-coche-autonomo-quien-debe-salvar-accidente-mortal-inevitable-201810241922_noticia.html.
- [46] S. Maejima and T. Murakami, “Driving assist control of electric vehicle by steer-by-wire system considering collision avoidance of traveling bicycle,” in *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 001657–001661, IEEE, 2015.
- [47] Z. Zhang, X. Zhang, H. Pan, W. Salman, Y. Rasim, X. Liu, C. Wang, Y. Yang, and X. Li, “A novel steering system for a space-saving 4ws4wd electric vehicle: design, modeling, and road tests,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 114–127, 2016.

-
- [48] K. Scicluna, C. S. Staines, and R. Raute, “Sensorless position tracking for steer-by-wire applications,” in *2017 19th International Conference on Electrical Drives and Power Electronics (EDPE)*, pp. 341–346, IEEE, 2017.
- [49] J. M. M. Tur, J. L. Gordillo, and C. A. Borja, “A closed-form expression for the uncertainty in odometry position estimate of an autonomous vehicle,” *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 1017–1022, 2005.
- [50] M. Halton, M. J. Hayes, and P. Jordanov, “State-space μ analysis for an experimental drive-by-wire vehicle,” *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 18, no. 9, pp. 975–992, 2008.
- [51] A. Dardanelli, G. Alli, and S. M. Savaresi, “Modeling and control of an electro-*** mechanical brake-by-wire actuator for a sport motorbike,” *IFAC Proceedings Volumes*, vol. 43, no. 18, pp. 524–531, 2010.
- [52] J. Bae, S. Lee, D.-H. Shin, J. Hong, J. Lee, J.-H. Kim, *et al.*, “Nonlinear identification of electronic brake pedal behavior using hybrid gmdh and genetic algorithm in brake-by-wire system,” 2017.
- [53] J. S.-T. P. Amaya-Trinidad, J.F. Guerrero-Castellanos, G.A. Muñoz-Hernández, G. Mino-Aguilar, “Model, simulation and control of an electric vehicle.,” *Memorias del Congreso Nacional de Control Automático*, no. October 2012, 2012.
- [54] C. L. J. Line, *Modelling and control of an automotive electromechanical brake*. PhD thesis, University of Melbourne, 2007.
- [55] C. C. Lin, H. Peng, and J. W. Grizzle, “A stochastic control strategy for hybrid electric vehicles,” *Proceedings of the American Control Conference*, vol. 5, pp. 4710–4715, 2004.
- [56] J. Yang, Z. Zeng, Y. Tang, J. Yan, H. He, and Y. Wu, “Load frequency control in isolated micro-grids with electrical vehicles based on multivariable generalized predictive theory,” *Energies*, vol. 8, no. 3, pp. 2145–2164, 2015.
- [57] G. Lee and S. Kim, “A longitudinal control system for a platoon of vehicles using a fuzzy-sliding mode algorithm,” *Mechatronics*, vol. 12, no. 1, pp. 97–118, 2002.
- [58] Z. Ioannou, P.; Xu, “Throttle and Brake Control Systems for Automatic Vehicle Following,” *UC Berkeley Research Reports*, 1994.
- [59] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *Ieee access*, vol. 2, pp. 56–77, 2014.
- [60] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, “A predictive controller for autonomous vehicle path tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92–102, 2009.

- [61] F. G. Rossomando, C. Soria, and R. Carelli, “Control de robots móviles con incertidumbres dinámicas usando redes de base radial,” *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 7, no. 4, pp. 28–35, 2010.
- [62] X. Yang, K. He, M. Guo, and B. Zhang, “An intelligent predictive control approach to path tracking problem of autonomous mobile robot,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, vol. 4, pp. 3301–3306, IEEE, 1998.
- [63] F. D. Boyden and S. A. Velinsky, “Dynamic modeling of wheeled mobile robots for high load applications,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 3071–3078, IEEE, 1994.
- [64] B. Kim, D. Neculescu, and J. Sasiadek, “Model predictive control of an autonomous vehicle,” in *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No. 01TH8556)*, vol. 2, pp. 1279–1284, IEEE, 2001.
- [65] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous uav guidance,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.
- [66] O. Adiyatov and H. A. Varol, “A novel rrt*-based algorithm for motion planning in dynamic environments,” in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1416–1421, IEEE, 2017.
- [67] C. Line, C. Manzie, and M. Good, “Control of an Electromechanical Brake for Automotive Brake-By-Wire Systems with an Adapted Motion Control Architecture,” *SAE Technical Paper Series*, vol. 1, no. 724, 2010.
- [68] F. Todeschini, M. Corno, G. Panzani, S. Fiorenti, and S. M. Savaresi, “Adaptive cascade control of a brake-by-wire actuator for sport motorcycles,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1310–1319, 2015.
- [69] C. Rossi, A. Tilli, and A. Tonielli, “Robust control of a throttle body for drive by wire operation of automotive engines,” *IEEE Transactions on control systems technology*, vol. 8, no. 6, pp. 993–1002, 2000.
- [70] C. Line, M. C. Manzie, and M. C. Good, “Electromechanical brake modeling and control: From PI to MPC,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 446–457, 2008.
- [71] D. Pavković, J. Deur, M. Jansz, and N. Perić, “Adaptive control of automotive electronic throttle,” *Control Engineering Practice*, vol. 14, no. 2, pp. 121–136, 2006.

-
- [72] H. Wang, Z. Man, W. Shen, Z. Cao, J. Zheng, J. Jin, and D. M. Tuan, “Robust control for steer-by-wire systems with partially known dynamics,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2003–2015, 2014.
- [73] X. Wu, M. Zhang, M. Xu, and Y. Kakogawa, “Adaptive feedforward control of a steer-by-wire system by online parameter estimator,” *International journal of automotive technology*, vol. 19, no. 1, pp. 159–166, 2018.
- [74] Y. Yamaguchi and T. Murakami, “Adaptive control for virtual steering characteristics on electric vehicle using steer-by-wire system,” *IEEE transactions on industrial electronics*, vol. 56, no. 5, pp. 1585–1594, 2008.
- [75] M. Doumiati, A. Victorino, and A. Charara, “Dynamics control of an In-Wheel Electric Vehicle with Steer-by-Wire,” *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, pp. 348–353, 2014.
- [76] L. De Novellis, A. Sorniotti, P. Gruber, and A. Pennycott, “Comparison of feedback control techniques for torque-vectoring control of fully electric vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3612–3623, 2014.
- [77] J. Villagra, V. Milanes, J. Perez, and T. de Pedro, “Control basado en PID inteligentes: aplicacion al control de cruceo de un vehiculo a bajas velocidades,” *Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 7, no. 4, pp. 44–52, 2012.
- [78] H. Montes, C. Salinas, R. Fernandez, and M. Armada, “An experimental platform for autonomous bus development,” *Applied Sciences*, vol. 7, no. 11, p. 1131, 2017.
- [79] Y. Fu and T. Chai, “Robust self-tuning PI decoupling control of uncertain multivariable systems,” *International Journal of Adaptive Control and Signal Processing*, vol. 2522619, no. 26, pp. 316–332, 2012.
- [80] Z. Xiong, X. Guo, B. Yang, X. Pei, and J. Zhang, “Modeling and pressure tracking control of a novel electro-hydraulic braking system,” *Advances in Mechanical Engineering*, vol. 10, no. 3, pp. 1–16, 2018.
- [81] G. Hernandez, C. Bordons, D. Marcos, and C. Montero, “Control De Estabilidad Basado En Mpc Para Un Vehiculo Electrico Con Motores En Rueda,” *XXXVI Jornadas de Automatica*, 2015.
- [82] E. M. Controller, “Manual Model 1268,” 2012.
- [83] “evolutionelectricvehicles.com.” [Consulta el 31-07-2017]. Disponible en: <http://evolutionelectricvehicles.com/product-408.html>.
- [84] J. L. Blanco-Claraco, J. Lopez-Martinez, J. L. Torres-Moreno, and A. Gimenez-Fernandez, “A low-cost modular platform for heterogeneous data acquisition with accurate interchannel synchronization,” *Sensors*, vol. 15, no. 10, pp. 27374–27392, 2015.

- [85] “Powering the world’s robots.” [Consulta el 31-07-2017]. Disponible en: <http://www.ros.org/>.
- [86] “Topics - ROC Wiki.” [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/Topics>.
- [87] “RQT Graph - ROS Wiki.” [Consulta el 31-07-2017]. Disponible en: http://wiki.ros.org/rqt_graph.
- [88] “ROSBAG - ROS Wiki.” [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/rosbag>.
- [89] “ROSCORE - ROS Wiki.” [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/roscore>.
- [90] D. E. Rivera, M. Morari, and S. Skogestad, “Internal model control: Pid controller design,” *Industrial & engineering chemistry process design and development*, vol. 25, no. 1, pp. 252–265, 1986.
- [91] S. Skogestad, “Simple analytic rules for model reduction and pid controller tuning,” *Journal of process control*, vol. 13, no. 4, pp. 291–309, 2003.

Los sistemas de gran complejidad, como los vehículos, necesitan que todos sus elementos funcionen bien tanto de forma individual como en conjunto. Si la sinergia entre los distintos subsistemas es elevada, mejor es el desempeño de los resultados obtenidos.

Este trabajo presenta como objetivo la mejora de la plataforma UAL-eCARM, un vehículo eléctrico urbano. El trabajo comprende el modelado y control de los sistemas de dirección y tracción del vehículo. Para ello se han mejorado los modelos y arquitecturas de control del mecanismo Steer-by-Wire. El sistema Throttle-by-Wire, un sistema no lineal, ha sido modelado como un sistema lineal entorno a cuatro puntos de operación. La estrategia de control implementada es un controlador PI adaptativo según el ajuste de Ziegler-Nichols. El sistema Brake-by-Wire, el más reciente instalado, se ha modelado y controlado mediante un controlador PI ajustado manualmente. La implementación está disponible en Github.

Las principales aportaciones de este trabajo son la ampliación y mejora de los modelos del vehículo UAL-eCARM y la migración de la arquitectura de control desde los nodos de ROS a microcontroladores.

Highly complex systems, such as vehicles, need all their elements to work well both individually and together. If the synergy between the different subsystems is high, the performance of the results obtained is better.

This work presents as goal to improve the UAL-eCARM platform, an urban electric vehicle. The work includes the modeling and control of the steering and traction systems of the vehicle. To achieve it, the models and control architectures of the Steer-by-Wire mechanism have been improved. The Throttle-by-Wire system, a non-linear system, has been modeled as a linear system around four operating points. The control strategy implemented is an adaptive PI controller according to Ziegler-Nichols tuning. The Brake-by-Wire system, the most recently installed, has been modeled and controlled by a manually tuning PI controller. The implementation is available in Github.

The main contributions of this work are the expansion and improvement of the UAL-eCARM vehicle models and the migration of the control architecture from ROS nodes to microcontrollers.

