

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

“Caracterización y control de sistema  
drive-by-wire en vehículo eléctrico”

Curso 2016/2017

**Alumno/a:**

Francisco José Mañas Álvarez

**Trabajo Monográfico**

**Director/es:**

José Luis Blanco Claraco  
José Luis Guzmán Sánchez





UNIVERSIDAD DE ALMERÍA  
ESCUELA SUPERIOR DE INGENIERÍA



GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL

---

TRABAJO FIN DE GRADO

---

*Caracterización y control de sistema drive-by-wire en vehículo eléctrico*

---

*Alumno:* Francisco José Mañas Álvarez  
*Director:* José Luis Blanco Claraco  
*Codirector:* José Luis Guzmán Sánchez  
*Fecha:* Septiembre de 2017

---

*Francisco José Mañas Álvarez*

---

*José Luis Blanco Claraco*

---

*José Luis Guzmán Sánchez*



*A mis padres, Francisco y Pilar,  
y mis hermanas, Laura y Cristina.*



# Índice general

	Página
Agradecimientos	VII
Resumen	IX
Abstract	XI
Siglas y acrónimos	XIII
Índice de figuras	XX
Índice de tablas	XXI
<b>1. Introducción</b>	<b>3</b>
1.1. Introducción . . . . .	3
1.2. Motivación del TFG . . . . .	4
1.3. Objetivos . . . . .	6
1.4. Contexto . . . . .	7
1.5. Resumen de resultados . . . . .	8
1.6. Fases de desarrollo y planificación . . . . .	12
1.7. Estructura del TFG . . . . .	13
<b>2. Revisión bibliográfica</b>	<b>15</b>
2.1. Introducción . . . . .	15
2.2. Estado actual de los sistemas de los vehículos eléctricos . . . . .	16
2.3. Sistemas By-Wire . . . . .	17
2.4. Control en vehículos eléctricos y autónomos . . . . .	19
<b>3. Vehículo eCARM</b>	<b>25</b>
3.1. Introducción . . . . .	25
3.2. Componentes hardware . . . . .	26
3.2.1. Controlador Curtis [1] . . . . .	26
3.2.2. PC Industrial . . . . .	27
3.2.3. Raspberry Pi . . . . .	27
3.2.4. Arduino Pro Mini . . . . .	28
3.2.5. Relé . . . . .	28
3.2.6. PowerBox . . . . .	28

3.2.7.	Fuentes de alimentación . . . . .	29
3.2.8.	Motor dirección . . . . .	30
3.2.9.	Encoder dirección . . . . .	30
3.3.	Arquitectura software . . . . .	31
3.3.1.	Robotics Operating System . . . . .	31
3.3.2.	Nodos de <i>ROS</i> en eCARM . . . . .	32
<b>4.</b>	<b>Throttle-by-wire</b>	<b>45</b>
4.1.	Introducción . . . . .	45
4.2.	Modelado del sistema . . . . .	46
4.2.1.	Ensayos de velocidad de avance . . . . .	47
4.2.2.	Ensayos de velocidad de marcha atrás . . . . .	49
4.2.3.	Ensayos con el vehículo en movimiento . . . . .	51
4.2.4.	Identificación de modelos. . . . .	53
<b>5.</b>	<b>Steer-by-wire</b>	<b>61</b>
5.1.	Introducción. . . . .	61
5.2.	Modelado. . . . .	62
5.3.	Diseño del sistema de control. . . . .	66
5.3.1.	Lazo interno. . . . .	66
5.3.2.	Lazo externo. . . . .	70
5.4.	Ensayos experimentales . . . . .	75
<b>6.</b>	<b>Conclusiones y trabajos futuros</b>	<b>79</b>
6.1.	Conclusiones . . . . .	79
6.2.	Trabajos futuros . . . . .	80



# Agradecimientos

En primer lugar, he de agradecer a mi familia todo el apoyo que me han brindado a lo largo de los cuatro años del grado. Sin ellos no hubiera sido posible llegar hasta este momento. Han sabido decirme las palabras necesarias en cada momento para impulsarme a seguir adelante, para afrontar cada problema y para dar mi mejor versión.

Muchos son los compañeros que he tenido a lo largo de estos cuatro años pero cinco de ellos son con los que más he compartido. La amistad con Álvaro, Andrés, Elena y Thomas me acompañará más allá del tiempo compartido en la universidad y les deseo lo mejor para cada uno de los retos que afronten. Por último, debo hacer una mención especial a mi compañera y amiga María Martell con la cual he compartido cuatro años de duro trabajo codo con codo, pero todo el esfuerzo ha tenido su recompensa.

Finalmente, me gustaría destacar y agradecer la labor y dedicación a tres de los docentes que he tenido durante mi formación. Manuel Berenguel, José Luis Guzmán y José Luis Blanco me han transmitido el cariño y la dedicación por una profesión y por la búsqueda de soluciones para los problemas que nos rodean. He aprendido mucho a través de sus clases, pero además, con su ejemplo y dedicación me han mostrado y enseñado el cariño hacia la profesión y el esfuerzo incansable hacia la mejora del mundo que nos rodea.

A todos ellos y a las personas que han influenciado de forma positiva en esta etapa de mi vida, gracias.



# Resumen

La elevada demanda de vehículos, el progreso tecnológico asociado a la automatización de nuestro entorno y la preocupación por la conservación del medio ambiente está guiando la investigación de vehículos hacia mejores resultados con vehículos eléctricos y autónomos. Pero el control a tan alto nivel no es posible si en los niveles inferiores, los sistemas de control no son lo suficientemente eficaces.

Este trabajo presenta como objetivo la realización de un control eficaz a bajo nivel sobre los sistemas *Steer-by-wire* y *Throttle-by-wire*. Para ello, en primer lugar se debe realizar el modelado de los sistemas actuales del vehículo. Una vez se comprueba la validez de los modelos obtenidos, se trabaja en el diseño de los sistemas de control mediante herramientas de simulación, en este caso se emplea principalmente *Simulink*. Los principales problemas tratados en el trabajo son la identificación de modelos de alto orden, en el caso del sistema *Throttle-by-wire*, y el control en cascada con un sistema de retardo dominante en el lazo interno para el caso del sistema *Steer-by-wire*. Posteriormente, con el sistema de control ya diseñado y comprobada su eficiencia en simulación, se implementa en el sistema real incorporándolo al sistema ya existente creado en *ROS*.

La principal aportación de este trabajo, además de la actualización de los modelos instalados en el vehículo *eCARM*, es la implementación del sistema de control completo en el nodo *steer\_controller* del sistema *ROS*.

**Palabras clave:** *Control en cascada, Predictor de Smith, ROS, Drive-by-wire.*



# Abstract

The high demand of vehicles, the technological improvement associated to the automation of our surrounding and the concern about the environment is guiding the vehicles' research to better results with electric and autonomous car. But the control at such a high level isn't possible if at the lower levels the control systems aren't efficient enough.

This work presents as goal to do a efficient control at low level over the *Steer-by-wire* and *Throttle-by-wire* systems. For that, in first place, the modeling of the current vehicle systems must be performed. Once the ratification of the obtained models is verified, we work on the design of control systems using simulation tools, in this case *Simulink* is mainly used. The main problems confronted in the work are the identification of high-order models in the case of the *Throttle-by-wire* system and the cascade control with a dominant delay system in the internal loop for the case of the *Steer-by-wire*. Subsequently, with the control system already designed and tested its efficiency in simulation, it's implemented in the real system incorporating it to the existing system compiled in *ROS*.

The main contribution of this work, in addition to updating the models installed in the *eCARM* vehicle, is the implementation of the complete control system in the *steer\_controller* node of the *ROS* system.

**Keywords:** *Cascade control, Smith Predictor, ROS, Drive-by-wire.*



# Siglas y acrónimos

<b>ARPANET</b>	Red de la Agencia de Proyectos de Investigación Avanzada (del inglés <i>Advanced Research Projects Agency NETwork</i> )
<b>ENIAC</b>	Computador e Integrador Numérico Electrónico (del inglés <i>Electronic Numerical Integrator And Computer</i> )
<b>EE.UU.</b>	Estados Unidos
<b>ROS</b>	Sistema Operativo Robótico (del inglés <i>Robotic Operating System</i> )
<b>ARM</b>	Grupo de Investigación TEP-197 <i>Automática, Robótica y Automática</i>
<b>eCARM</b>	Electric Car ARM
<b>ARX</b>	AutoRegresivo con entrada eXógena (del inglés <i>AutoRegressive eXogenous.</i> )
<b>PI</b>	Controlador de acción Proporcional e Integral
<b>SIMC</b>	Skogestad - Control de Modelo Interno (del inglés <i>Skogestad Internal Model Control</i> )
<b>PWM</b>	Modulación por ancho de pulsos (del inglés <i>Pulse Width Modulation</i> )
<b>PID</b>	Controlador de acción Proporcional, Integral e Derivativa
<b>RPM</b>	Revolución Por Minuto
<b>V</b>	Voltio
<b>s</b>	Segundo
<b>DBW</b>	Drive By Wire
<b>rad/s</b>	Radian/segundo
<b>UAL</b>	Universidad de Almería

<b>RRT</b>	Exploración de Árbol de forma Aleatoria (del inglés <i>Rapidly exploring Random Tree</i> )
<b>PRM</b>	Mapa de Ruta Probabilístico (del inglés <i>Probabilistic RoadMap</i> )
<b>PD</b>	Controlador de acción Proporcional y Derivativa
<b>mm</b>	Milímetro
<b>km</b>	Kilómetro
<b>h</b>	Hora
<b>kg</b>	Kilogramo
<b>kW</b>	Kilovatio
<b>m</b>	Metro
<b>Ah</b>	Amperio hora
<b>MOSFET</b>	Transistor de efecto de campo metal-óxido-semiconductor (del inglés <i>Metal-Oxide Semiconductor Field-Effect Transistor</i> )
<b>ISO</b>	Organización Internacional de Normalización (del inglés <i>International Organization for Standardization</i> )
<b>PC</b>	Computador Personal (del inglés <i>Personal Computer</i> )
<b>SSH</b>	Secure SHell
<b>CPU</b>	Unidad Central de Procesamiento (del inglés <i>Central Processing Unit</i> )
<b>RAM</b>	Memoria de Acceso Aleatorio (del inglés <i>Random Access Memory</i> )
<b>Gb</b>	GigaByte
<b>USB</b>	Bus Universal en Serie (del inglés <i>Universal Serial Bus</i> )
<b>HDMI</b>	Interfaz Multimedia de Alta Definición (del inglés <i>High – Definition Multimedia Interface</i> )
<b>mA</b>	Miliamperio
<b>STAIR</b>	STanford Aritificial Intelligence Robot
<b>MRPT</b>	Herramientas para Programar Robots Móviles (del inglés <i>Mobile Robot Programming Toolkit</i> )



<b>GPIO</b>	Entrada/Salida de Propósito General (del inglés <i>General Purpose Input/Output</i> )
<b>E/S</b>	Entrada/Salida
<b>ADC</b>	Convertor Analógico-Digital (del inglés <i>Analog to Digital Converter</i> )
<b>CC-CC</b>	Corriente Continua - Corriente Continua
<b>F.D.T.</b>	Función De Transferencia
<b>SISO</b>	Única Entrada - Única Salida (del inglés <i>Single Input Single Output</i> )
<b>CITE</b>	CIentífico-TÉcnico
<b>MOC</b>	Mantenedor de Orden Cero
$T_{rm}$	Tiempo de residencia medio
$Q_{rn}$	Índice de controlabilidad
<b>IMU</b>	Unidad de Medición Inercia (del inglés <i>Inertial Measurement Unit</i> )
<b>GPS</b>	Sistema de Posicionamiento Global (del inglés <i>Global Positioning System</i> )



# Índice de figuras

1.1. Ejemplo de optimización del producto. . . . .	5
1.2. Esquema de control planteado inicialmente. . . . .	6
1.3. Vehículo eCARM en la puerta del edificio Científico-Técnico IV (CITE IV). . . . .	7
1.4. Modelo construido en <i>Simulink</i> para la validación de datos. . . . .	9
1.5. Ensayo de validación de los modelos obtenidos. . . . .	9
1.6. Ensayo de validación para el control diseñado. . . . .	10
1.7. Modelo construido en <i>Simulink</i> para la validación de datos. . . . .	11
1.8. Modelo empleado en el toolbox <i>System Identification</i> . . . . .	11
1.9. Diagrama de Gantt. . . . .	13
2.1. Thomas A. Edison junto a las baterías desarrolladas por él mismo para los vehículos eléctricos de la época [2]. . . . .	16
2.2. Vehículo <i>La Jamais Contente</i> . Año 1899 [3]. . . . .	17
2.3. Comparación de un sistema de dirección tradicional y un sistema by-wire [4]. . . . .	18
2.4. Niveles de control de un robot móvil autónomo. . . . .	19
2.5. Resultado del control propuesto en [5]. . . . .	21
2.6. Planteamiento propuesto en [5]. . . . .	22
2.7. Esquema de control empleando estimador de parámetros en línea. [6]. . . . .	22
2.8. Resultados de emplear un control anticipativo frente a un controlador PD. [7]. . . . .	23
3.1. Imagen del Controlador Curtis [8]. . . . .	26

3.2. Imagen del PC industrial. . . . .	27
3.3. Componentes hardware encargados de soportar el sistema de control. . . . .	28
3.4. Componentes encargados de la administración de corriente. . . . .	29
3.5. Componentes software descritos en las secciones 3.2.8 y 3.2.9. . . . .	30
3.6. Captura del sistema implementado en el vehículo eCARM con el comando <code>rqt_graph</code> . . . . .	32
3.7. Captura del nodo <code>joystick_driving</code> con el comando <code>rqt_graph</code> . . . . .	34
3.8. Captura del nodo <code>arduino_daq</code> con el comando <code>rqt_graph</code> . . . . .	35
3.9. Captura del nodo <code>steer_controller</code> con el comando <code>rqt_graph</code> . . . . .	36
3.10. Declaración de variables. . . . .	37
3.11. Lectura encoder incremental. . . . .	37
3.12. Lectura rad/s y estimación de la velocidad. . . . .	38
3.13. Procesado señal PWM. . . . .	38
3.14. Procesado señal acelerador. . . . .	39
3.15. Procesado de la referencia. . . . .	39
3.16. Protección pendiente de subida. . . . .	40
3.17. Determinar error lazo externo y señal del controlador. . . . .	40
3.18. Error lazo interno y señal del controlador. . . . .	40
3.19. Mecanismo de protección en los extremos. . . . .	41
3.20. Saturación. . . . .	41
3.21. Mecanismo Antiwindup. . . . .	42
3.22. Envío de datos en ROS. . . . .	42
3.23. Actualización de valores. . . . .	43
4.1. Señal PWM para el rango de señales de entrada con distintos throttle map. . . . .	46
4.2. Resultados del ensayo con múltiples escalones de amplitud 0.95V. . . . .	48
4.3. Resultados del ensayo con múltiples escalones de amplitud 2.38V. . . . .	48
4.4. Resultados del ensayo con múltiples escalones de amplitud variable. . . . .	50

4.5. Resultados del ensayo con múltiples escalones de amplitud 2.38V marcha atrás. . . . .	50
4.6. Resultados del ensayo con entrada variable. . . . .	51
4.7. Ensayo realizado con el vehículo en movimiento (I). . . . .	52
4.8. Ensayo realizado con el vehículo en movimiento (II). . . . .	52
4.9. Ensayo realizado con el vehículo en movimiento (III). . . . .	53
4.10. Comparación de los modelos obtenidos para un ensayo en vacío. . . . .	55
4.11. Comparación de los modelos obtenidos para un ensayo en movimiento. . . . .	55
4.12. Esquema de <i>Simulink</i> para la comparación de resultados. . . . .	56
4.13. Validación del modelo obtenido en un ensayo en vacío (entradas en escalón). . . . .	57
4.14. Validación del modelo obtenido en un ensayo en vacío (entradas variables). . . . .	57
4.15. Validación del modelo obtenido en un ensayo en vacío con marcha inversa (entradas en escalón). . . . .	58
4.16. Validación del modelo obtenido en un ensayo en vacío con marcha inversa (entradas variables). . . . .	58
4.17. Validación del modelo obtenido en un ensayo en movimiento. . . . .	59
4.18. Error medido entre el sistema estimado y el real. . . . .	60
5.1. Ensayo realizado para la toma de datos. . . . .	62
5.2. Validación del modelo obtenido sobre uno de los ensayos realizados. . . . .	65
5.3. Esquema de implementación del Predictor de Smith. [9] . . . . .	67
5.4. Control implementado en simulación en el lazo interno. . . . .	68
5.5. Respuesta del lazo interno en bucle cerrado. . . . .	69
5.6. Respuesta del lazo interno en bucle cerrado frente a perturbaciones. . . . .	70
5.7. Ensayo simulado de los controladores diseñados. . . . .	71
5.8. Resultados en simulación frente a entradas en escalón. . . . .	72
5.9. Resultados en simulación frente a entradas en rampa. . . . .	72
5.10. Ensayo simulado de los controladores diseñados con filtro en la referencia. . . . .	73

5.11. Resultados en simulación frente a entradas en escalón con filtro en la referencia. . . . .	73
5.12. Resultados en simulación frente a entradas en rampa con filtro en la referencia. . . . .	74
5.13. Resultados en simulación con entradas en rampa frente a perturbaciones. . . . .	75
5.14. Esquema de <i>Simulink</i> para la validación de datos. . . . .	76
5.15. Resultados experimentales del primer ensayo. . . . .	77
5.16. Resultados experimentales del segundo ensayo. . . . .	77
5.17. Resultados experimentales del primer ensayo (Ampliado). . . . .	78
6.1. Resultados del modelado llevado a cabo en ensayos anteriores. . . . .	81

# Índice de tablas

1.1. Parámetros de los controladores implementados. . . . .	10
1.2. Distribución temporal. Total: 450h . . . . .	13
3.1. Características técnicas del vehículo <i>eCARM</i> . . . . .	25
3.2. Listado de nodos de ROS en el vehículo <i>eCARM</i> . . . . .	33





**Caracterización y control  
de sistema drive-by-wire  
en vehículo eléctrico**



# Capítulo 1

## Introducción

### 1.1. Introducción

La realización del trabajo fin de grado (TFG) supone la culminación de la formación académica que otorgan los estudios de grado. Mediante su realización se persigue que el autor refleje su capacitación para el abordaje de problemas y su solventación como muestra del nivel de conocimientos y competencias adquiridas durante el tiempo de estudio. Como valor añadido, la defensa del fruto resultante del esfuerzo realizado ante un tribunal, tal como la sociedad le exigirá durante el ejercicio de su profesión, es una experiencia que le otorgará mayor soltura el día de mañana. No obstante, reflejar en detalle la extensa formación recibida en cada uno de los campos que comprende la ingeniería industrial es, cuanto menos, una labor ardua que escapa de las posibilidades que brinda este trabajo a causa de su extensión y envergadura.

La primera dificultad del estudiante al afrontar este trabajo es la elección del modelo de trabajo y la temática del mismo para cumplir las competencias y objetivos estipulados. Mientras que durante el grado, los profesores y las mismas materias, guían al estudiante en la realización de trabajos con las temáticas, formatos, extensión, etc., la elección de este trabajo supone un punto de inflexión en la formación del estudiante. Se espera que tome una decisión propia que permita alcanzar con éxito la realización de un trabajo que presente relación con las materias estudiadas durante su formación.

Pese a tener un carácter de culminación de estudios, también supone un aprendizaje de gran valor, ya que al trabajar de forma más independiente, se consolidan valores y se aprecian mejor las indicaciones recibidas por el profesorado durante el periodo de formación. Indicaciones como que para la consecución de un objetivo con éxito no solo basta con una gran fuente de conocimientos; una gran planificación y orden en el proceso favorecen la obtención de resultados de forma más eficaz.

### 1.2. Motivación del TFG

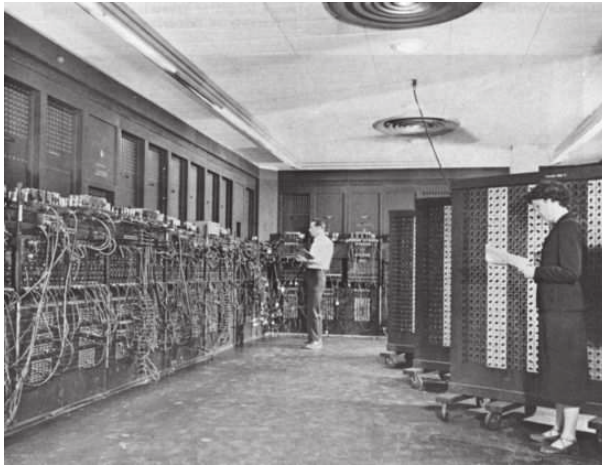
Es innegable la fuerte relación que une la tecnología con la sociedad. Tanto es así que la tecnología ha evolucionado de favorecer el desarrollo de la sociedad con sus avances a estar forzada a avanzar según las exigencias marcadas por la sociedad. Cada vez es mayor la celeridad que se exige en los avances tecnológicos puesto que la disconformidad constante con la situación actual provoca la aparición de nuevas necesidades que exigen una solución o la mejora de las soluciones ya existentes.

Desde la aparición de ARPANET a finales de los años 60 con una velocidad de comunicación de 50kb/s hasta la velocidad actual de 100Gb/s de ciertas redes inalámbricas de alta velocidad [11]. Mediante este ejemplo, queda evidenciado como en 50 años, una tecnología ha sufrido un desarrollo tan grande por la constante necesidad de la sociedad de mayores redes de comunicación y mayores velocidades en ellas.

Otra perspectiva muy visual que es reflejo de como se piensa que evolucionará la tecnología se encuentra en el mundo cinematográfico. En él, a modo de ciencia ficción, se especula sobre tecnologías revolucionarias con las que se convivirá en el día a día en el futuro. Hace 30-40 años se hacían películas como *Regreso al futuro* u *Odisea en el espacio* en las que se pensaba que en la actualidad la tecnología habría avanzado tanto que viajaríamos por el espacio, los coches volarían y otras tantas invenciones que aun hoy día siguen siendo ciencia ficción. Pero el hecho de que no se hayan materializado estas visiones futuristas no quiere decir que la tecnología no haya avanzado, sino más bien que la visión de avance tecnológico que se tenía no es el avance real que se ha producido.

Si se comparan los productos desarrollados originalmente con sus versiones actuales, se aprecia de forma notoria la evolución tecnológica. Puede parecer que el avance es pequeño puesto que siguen siendo los mismos productos los que suplen las mismas necesidades, pero nada más lejos de la realidad. Grandes avances tecnológicos que se han producido han surgido como mejoras o modificaciones de los productos que ya se emplean. Prueba de esto reside en las constantes mejoras a las que se someten, por ejemplo, los medios de transporte con motores de menor consumo y mayor potencia, mecanismos y diseños más eficientes, etc. Todas estas mejoras están incentivadas por una razón, la optimización. Una vez se encuentra la herramienta para suplir una necesidad no se debe considerar el problema resuelto completamente. Se debe revisar el proceso nuevamente en busca de conseguir los mismos resultados o mejores con menor esfuerzo y gasto. Una prueba muy visual de esto reside en la evolución de los computadores.

Desde el primer computador de propósitos generales (ENIAC) desarrollado durante la segunda guerra mundial en EE.UU. (Figura 1.1a) hasta el último Apple Watch (Figura 1.1b) el concepto de mejora y optimización queda patente al reducir un ordenador cuyas dimensiones eran de una habitación a un producto con mayor cantidad de aplicaciones en el tamaño de un reloj de pulsera.



(a) Computador ENIAC 1943.



(b) Apple Watch series 2.

Figura 1.1: Ejemplo de optimización del producto.

Volviendo la mirada al sector del transporte, no se puede limitar el ámbito de mejora al aspecto mecánico. Un importante campo de mejora reside en el uso y manejo que se realice de los distintos componentes. Así pues, si para realizar una tarea, como desplazarse de un lugar A a un lugar B, se toma la primera solución disponible sin considerar otras alternativas, puede que se adopte una solución que no garantiza la máxima eficiencia. En el caso del desplazamiento, las pérdidas de eficiencia pueden encontrarse en el tiempo empleado, distancia recorrida o incluso en las maniobras empleadas durante la conducción. La implementación de sistemas inteligentes y de control en los vehículos pueden facilitar la toma de decisiones y la mejora del rendimiento, pero no se debe limitar su aplicación a la optimización, también pueden tener un gran interés en la supresión de fallas producidas por descuidos o negligencias humanas que puedan desembocar en desperfectos materiales, y en el peor de los casos, en pérdidas de vidas humanas. Con esto, no se pretende eliminar al ser humano de la ecuación, sino más bien brindar un apoyo de confianza a los usuarios para mejorar la experiencia de la conducción.

Por las razones expuestas anteriormente, la principal motivación de este TFG reside en la mejora del sistema de control sobre la dirección y control de velocidad del vehículo eléctrico que posee la Universidad de Almería (UAL). Con estas mejoras se busca imponer un comportamiento suave de los distintos actuadores con el fin de evitar la fatiga y el sufrimiento mecánico en ellos y una mejor experiencia en la conducción de dicho vehículo. No obstante, como en todo sistema real, no se alcanzarán resultados perfectos como ocurre al trabajar con modelos teóricos en simulación. Tras la labor de diseño y simulaciones, quizás la labor más importante es ajustar el diseño en función de los resultados obtenidos con el fin de obtener mejores resultados. Pese a que el presente trabajo se realice sobre un vehículo concreto, los resultados obtenidos pueden ser extrapolados y adaptados a cualquier otro vehículo, autónomo o no, en sus correspondientes lazos de control.

### 1.3. Objetivos

Tras lo expuesto en las secciones anteriores, el objetivo marcado para este trabajo está en la consecución del establecimiento de una respuesta controlada, tanto para la velocidad como para la dirección, del vehículo con unas especificaciones establecidas en la etapa de diseño que alcance las referencias establecidas. Estas referencias pueden ser fijadas por el usuario mediante los mandos de control (volante, pedales, mandos, etc.) o en el caso de encontrarse el vehículo en modo autónomo, las consignas las establecerá el programa que determine las trayectorias vía software.

Para la consecución del objetivo descrito anteriormente se han de tratar otros objetivos adyacentes que garanticen la máxima eficiencia en el desarrollo. Dichos objetivos son:

- Modelado del sistema y diseño de posibles controladores en programas software, tales como Matlab o Simulink. En la figura 1.2 se muestra el esquema planteado inicialmente para el sistema *Steer-by-wire* para el control en tiempo continuo y discreto.
- Instalación y calibración de los sensores y actuadores, tales como motores y encoders, que permiten la clausura del lazo de control en el que se encuentren. Este subobjetivo requiere una gran atención en la etapa de diseño puesto que los componentes físicos no presentan comportamientos idílicos. Se debe realizar el diseño siendo conscientes de que, por ejemplo, la velocidad de transmisión de datos limita el tiempo de muestreo, parámetro crítico en el diseño de los controladores en tiempo discreto.
- Traducción de los controladores diseñados al lenguaje empleado a nivel de software. En este caso se trabajará con *Robotics Operating System* (ROS) el cual emplea como lenguaje *C++*.

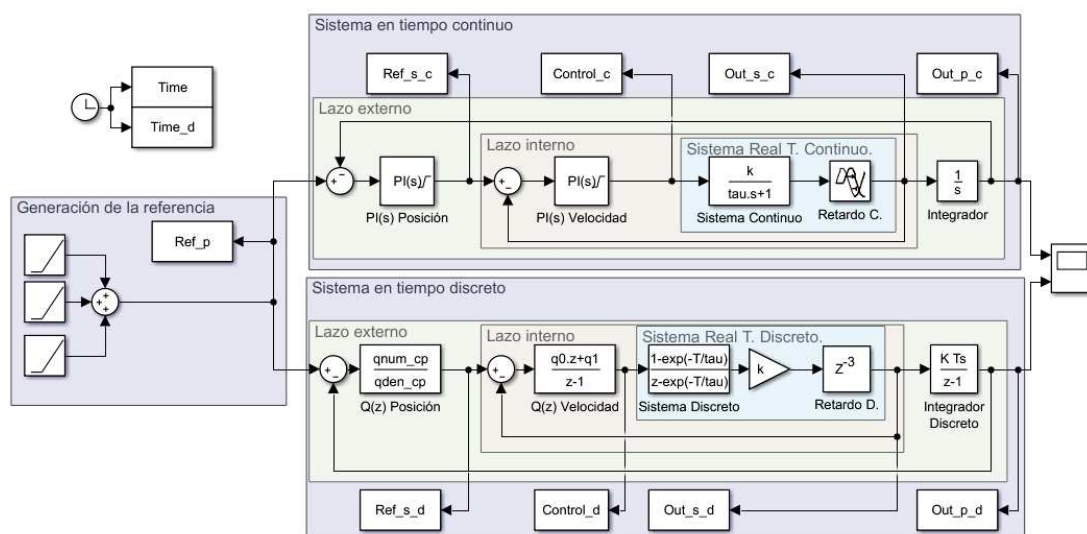


Figura 1.2: Esquema de control planteado inicialmente.

## 1.4. Contexto

Este trabajo surge como una propuesta por parte del profesorado que tuteló las prácticas curriculares del estudiante, también realizadas sobre el vehículo en el rediseño del layout y cableado de los elementos hardware de la arquitectura de control. La propuesta surge como búsqueda de un mayor aprovechamiento de los conocimientos y tiempo invertido en el en este tramo final de los estudios. La curva de aprendizaje que requiere este proyecto es lenta puesto que requiere comprender el proyecto en su estado inicial, las modificaciones a las que se ha sometido que lo han llevado hasta el estado actual y conocimientos sobre las modificaciones que se desean implantar y los mecanismos necesarios para llevarlas a cabo. El vehículo eléctrico que posee la UAL es del modelo *LITA CLe2-2S* (Changzhou Greenland Vehicle Co., s.f.) [12], denominándose también eCARM (Figura 1.3).



Figura 1.3: Vehículo eCARM en la puerta del edificio Científico-Técnico IV (CITE IV).

Este vehículo ha sido objeto de múltiples investigaciones desarrolladas por el grupo de investigación de la UAL *TEP-197 Automática, Robótica y Mecatrónica*(ARM). Las oportunidades que ofrece este vehículo son tan variadas que lo hacen un instrumento multidisciplinar de gran valor científico y formativo. A continuación se ejemplifica de forma breve las áreas involucradas en este trabajo.

- Área mecánica: Apreciable en todo lo relacionado con encoders, mecanismos de transmisión y estructura del vehículo, estudio y caracterización de motores eléctricos, etc.
- Área electrónica: Presente en la distribución del cableado, empleo de sensores, tarjetas de adquisición de datos y desarrolladas por el personal investigador, etc.
- Área automática y control: Visible en la obtención de modelos, desarrollo de los controladores, estudios de sostenibilidad y eficiencia energética, etc.

## 1.5. Resumen de resultados

Se ha diseñado e implementado con éxito un sistema de control que permite el alcance y seguimiento de consignas introducidas por el usuario en el vehículo *eCARM*. En el lazo de control se ven involucrados un joystick, el cual introduce las consignas a través de su correspondiente nodo de *ROS*, el controlador programado vía software en *ROS*, la placa arduino Pro mini, que actúa como tarjeta de entrada / salida, el motor maxon que actúa sobre el mecanismo de la dirección y los encoder incremental (HEDL5540) y absoluto (EMS22A) para realimentar el lazo de control.

Los primeros resultados obtenidos son producto del modelado de la dirección. Dicho modelado se ha realizado mediante el empleo de la técnica de las “cajas negras” y no basado en primeros principios como ya se ha realizado anteriormente [13]. A la finalización del proceso de modelado se determina con gran exactitud el comportamiento de la velocidad de la dirección instalada en el vehículo mediante las función de transferencia en tiempo continuo y discreto representadas en las ecuaciones 1.1 y 1.2. La ecuación que determina el comportamiento es resultado de añadir un término integrador a las funciones obtenidas, tal como se muestra en las ecuaciones 1.3 y 1.4.

$$G_v(s) = \frac{k_v}{\tau_v s + 1} e^{-tr_v s} = \frac{-0,0934}{0,0283s + 1} e^{-0,15s} \quad (1.1)$$

$$P_v(z) = \frac{bz^{-1}}{1 - az^{-1}} \cdot z^3 = \frac{-0,0782z^{-1}}{1 - 0,1713z^{-1}} \cdot z^3 \quad (1.2)$$

$$G_p(s) = \frac{k_v}{s \cdot (\tau_v s + 1)} e^{-tr_v s} = \frac{-0,0934}{s \cdot (0,0283s + 1)} e^{-0,15s} \quad (1.3)$$

$$P_p(z) = \frac{T \cdot (1 + z^{-1}) (bz^{-1})}{2 \cdot (1 - z^{-1}) (1 - az^{-1})} \cdot z^3 = \frac{0,05 \cdot (1 + z^{-1}) (-0,0782z^{-1})}{2 \cdot (1 - z^{-1}) (1 - 0,1713z^{-1})} \cdot z^3 \quad (1.4)$$

En función de los modelos obtenidos, su construcción en *Simulink* (figura 1.4) permite realizar la validación de los mismos como se muestra en la figura 1.5. Los resultados obtenidos en esta etapa del proyecto resultan de muy buena calidad para el desarrollo de las posteriores etapas de diseño.



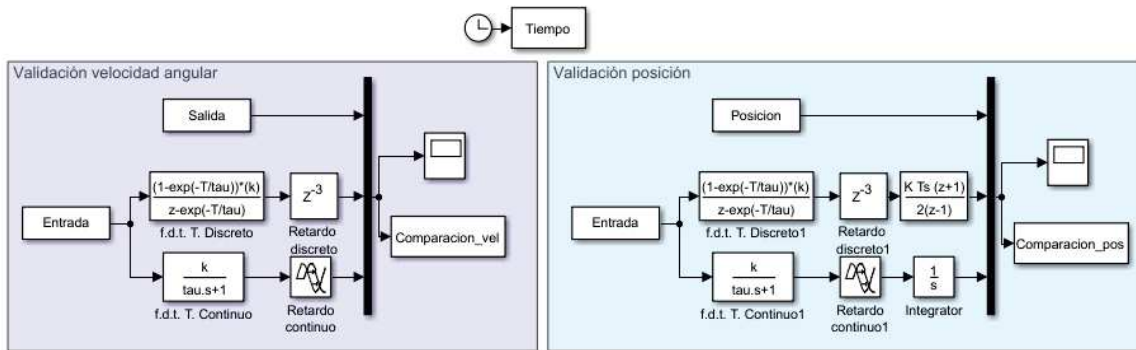


Figura 1.4: Modelo construido en *Simulink* para la validación de datos.

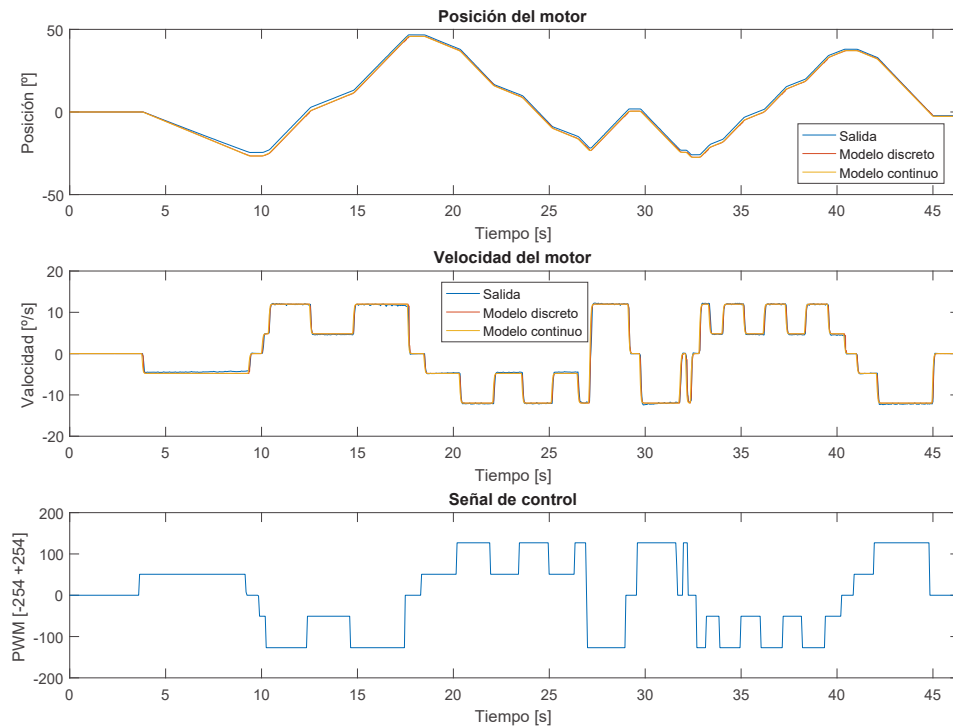


Figura 1.5: Ensayo de validación de los modelos obtenidos.

Una vez llevada a cabo la validación de los modelos y comprobado que el ajuste con la realidad reporta gran confianza para trabajar teóricamente se realiza el diseño del sistema de control. Este sistema presenta un tiempo de residencia medio ( $T_{rm}$ ) de valor 0.1783 y un tiempo de retardo normalizado ( $Q_{rn}$ ) de valor 0.8413. Este parámetro, conocido también como índice de controlabilidad, reporta la información inicial de que se trata de un sistema con retardo dominante y por tanto una mayor dificultad de controlar.

## 1.5. RESUMEN DE RESULTADOS

El sistema de control planteado es un control en cascada. En dicho control se emplea PI según el método de cancelación polo-cero haciendo uso del predictor de Smith para eliminar el efecto del retardo dominante en el lazo interno. Para el control del lazo externo, por la estructura del sistema y tras la prueba de diversos métodos, se emplea un PID según el método seleccionado es SIMC de *Skogestad* (tabla 1.1). También se hace uso de un filtro en la referencia para eliminar sobreoscilación provocada por el controlador PID empleado en el lazo externo. Finalmente, los parámetros obtenidos han sido objeto de un ligero ajuste manual tras observar su comportamiento en el sistema real. Su discretización se ha realizado empleando la aproximación trapezoidal de los parámetros obtenidos en tiempo continuo. Los resultados de la implantación en el sistema real y su comparación con el modelo contruido en *Simulink*, figura 1.7, se reflejan en la figura 1.6.

Parámetro T. Continuo	Valor	Parámetro T. discreto	Valor
$K_p$	2.2222	$q0_p$	11.1420
$T_{ip}$	1.8000	$q1_p$	-19.9691
$T_{dp}$	0.2000	$q2_p$	8.8888
$K_v$	-1.5005	$q0_v$	-2.8261
$T_{i_v}$	0.0283	$q1_v$	0.1750

Tabla 1.1: Parámetros de los controladores implementados.

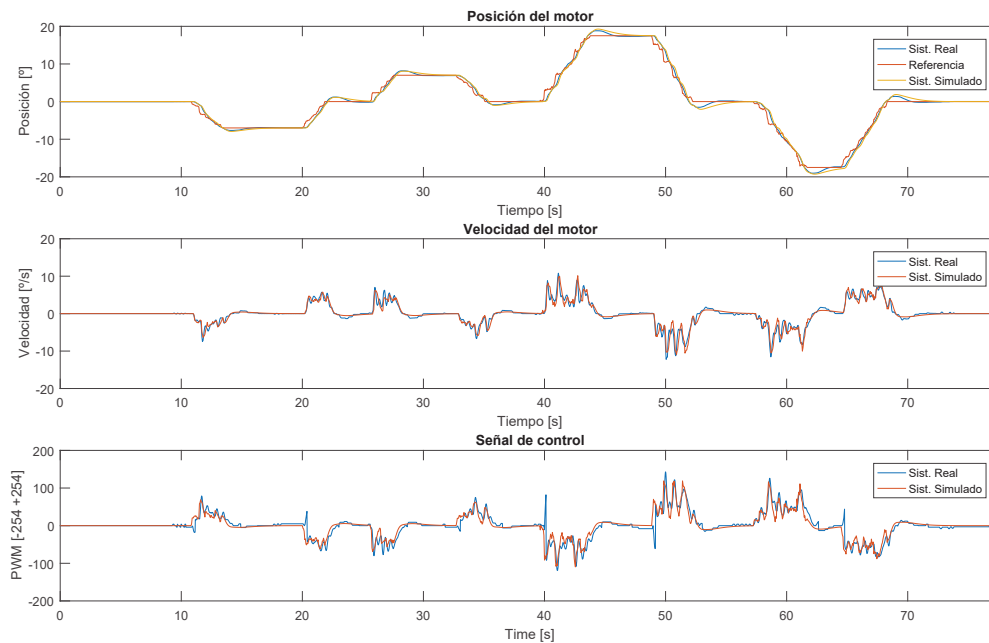


Figura 1.6: Ensayo de validación para el control diseñado.

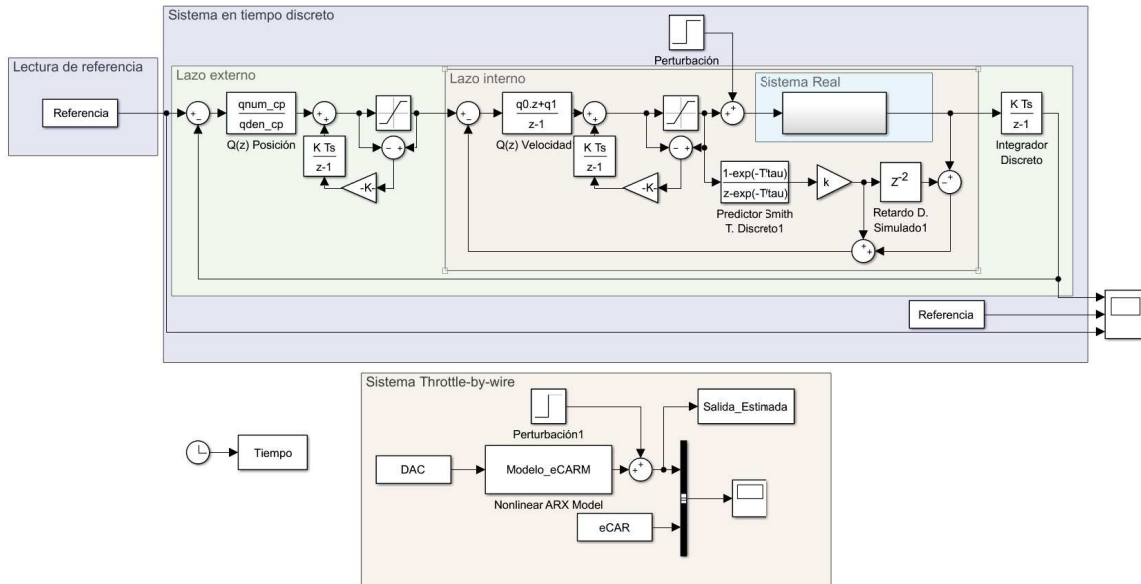


Figura 1.7: Modelo construido en *Simulink* para la validación de datos.

En lo referido al sistema *Throttle-by-wire*, pese a que el objetivo establecía un control sobre la velocidad de desplazamiento del vehículo, no se ha llegado a implementar un sistema de control sobre el mismo. El motivo de la decisión de no implementar este sistema es resultado de la observación de un modelo de alto orden en la etapa de modelado del sistema real. La realización de un control sobre dicho sistema requiere una mayor envergadura no contemplada en el presente trabajo. Por tanto, se ha considerado oportuno, por la carga de trabajo que supone la realización completa que implica el sistema *Steer-by-wire*, realizar hasta la etapa de modelado y el control plantearlo como un trabajo futuro, en el capítulo 6, previo a la conducción autónoma. El modelado del sistema en cuestión se realiza mediante el empleo de la herramienta *System Identification* haciendo uso del modelo no lineal ARX para la estimación de los parámetros del sistema. Este modelo empleado en el toolbox sigue el esquema mostrado en la figura 1.8 con el empleo de un bloque ARX lineal operando de forma paralela con un bloque que estima las no linealidades. Los resultados obtenidos de dicho proceso no se consideran concluyentes por ciertas inexactitudes observadas en los ensayos realizados para dicho proceso. Estas inexactitudes son producidas por limitaciones en la tarjeta de adquisición de datos empleada en el sistema, ya que la medición de la señal errónea medida durante el proceso con un osciloscopio, resultó ser correcta. Esto evidencia que para la obtención de un modelo más exacto y la implementación del sistema *Throttle-by-wire*, es necesario la sustitución del sistema de adquisición de datos.

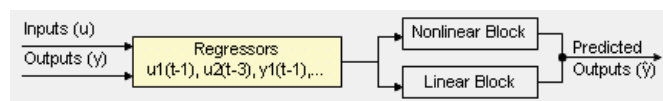


Figura 1.8: Modelo empleado en el toolbox *System Identification*.

### 1.6. Fases de desarrollo y planificación

La consecución de los objetivos alcanzados mostrados en la sección 1.5 son resultado de la realización de las tareas descritas a continuación.

- **Tarea 1: Investigación de antecedentes:** Se lleva a cabo un estudio acerca de los sistemas *by-wire*. El estado actual de los vehículos eléctricos y los controladores empleados en la literatura para abordar los sistemas *by-wire* abordados en el presente trabajo.
- **Tarea 2: Análisis del estado actual del vehículo:** Comprende el estudio y documentación de cada una de las partes que componen el vehículo. Dichas partes son los sensores instalados, arquitectura software, distribución de componentes hardware, etc.
- **Tarea 3: Modelado y diseño de los controladores Steer-by-wire y Throttle-by-wire:** Realización de ensayos para el modelado y validación de los sistemas reales del vehículo. Establecimiento de las especificaciones que se desean cumplir y posterior desarrollo del diseño de los controladores pertinentes.
- **Tarea 4: Instalación y puesta a punto de la arquitectura hardware y software:** Adecuación, previa a la implantación de los controladores diseñados, de todos los sensores implicados en los futuros pasos y su interconexión a nivel software hasta establecer un punto de seguridad al que poder volver si surgieren problemas en posteriores tareas.
- **Tarea 5: Implementación de los controladores diseñados:** Una vez en simulación se ha verificado el correcto funcionamiento del sistema de control diseñado, se procede con su adaptación al lenguaje empleado a nivel software.
- **Tarea 6: Realización de ensayos:** Toma de datos obtenidos de las pruebas llevadas a cabo para la verificación de los diseños establecidos en tareas anteriores. Estos datos son procesados en *Matlab* para posteriormente ser comparados con el sistema creado en la herramienta *Simulink*.
- **Tarea 7: Redacción de memoria:** Elaboración del presente documento que recopila la revisión bibliográfica, el material y la metodología aplicada, los resultados obtenidos y las conclusiones alcanzadas.

En la tabla 1.2 se muestra el tiempo requerido para la correcta finalización de cada una de las tareas descritas anteriormente. La ejecución de dichas tareas no se realiza de forma secuencial, por lo que algunas de las tareas son realizadas simultáneamente y el tiempo dedicado cada día no es homogéneo para todas las tareas. Esto implica que una tarea con menos duración puede extenderse más tiempo que otra con mayor labor. Como complemento se realiza un diagrama de Gantt con las diversas actividades, mostrado en la figura 1.9. El tiempo empleado registrado asciende a la cuantía de 450h, cantidad superior a la establecida en la guía docente.

Actividad	Comienzo	Fin	Duración	Tiempo
Tarea 1	30/3/2017	3/4/2017	4 días	20 horas
Tarea 2	4/4/2017	25/4/2017	12 días	60 horas
Tarea 3	26/4/2017	11/5/2017	16 días	60 horas
Tarea 4	3/5/2017	23/6/2017	32 días	160 horas
Tarea 5	20/6/2017	23/6/2017	4 días	10 horas
Tarea 6	26/6/2017	26/7/2017	30 días	80 horas
Tarea 7	1/6/2017	31/7/2017	60 días	60 horas

Tabla 1.2: Distribución temporal. Total: 450h

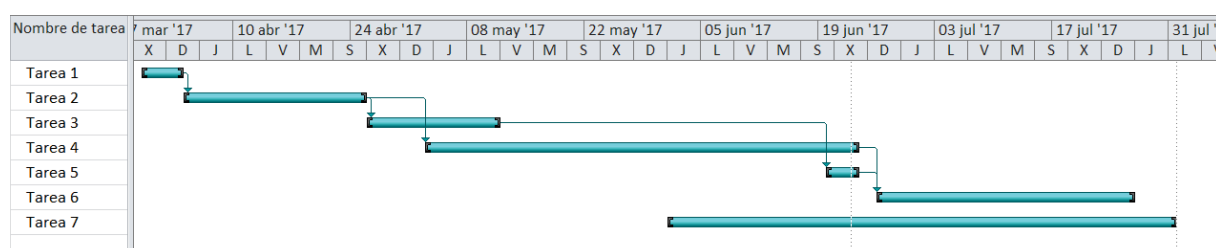


Figura 1.9: Diagrama de Gantt.

## 1.7. Estructura del TFG

En el presente trabajo, se ha considerado adecuado el establecimiento de una estructura de 6 capítulos, cuyo contenido se introduce en las siguientes líneas. El capítulo 1, que concluye con esta sección, consiste en una breve introducción al contenido que se desarrollará en capítulos posteriores. El capítulo 2 sirve como una contextualización del marco en el que se desarrolla el presente trabajo. En el capítulo 3, queda definido el estado actual del vehículo eléctrico eCARM, tanto la arquitectura hardware como la software. El capítulo 4 comprende el trabajo sobre el sistema *Throttle-by-wire* en el vehículo, su modelado y validación de los datos obtenidos. En el capítulo 5 se desarrolla el modelado, control y validación del sistema *Steer-by-wire*. Este trabajo concluye en el capítulo 6, el cuál está dedicado a unas conclusiones de los datos obtenidos y del trabajo realizado, junto con un apartado dedicado a la visión de futuro acerca de las posibilidades que ofrece el vehículo eCARM tomando de punto de partida el estado en el que se encuentra tras la finalización del trabajo desarrollado para la continuación de esta línea de investigación.



## Capítulo 2

# Revisión bibliográfica

### 2.1. Introducción

Con el aumento de la demanda de vehículos y sus prestaciones, tanto los modelos de vehículos como las infraestructuras de transporte actuales quedan, cada vez con mayor frecuencia, obsoletas. Mantener el sistema completo actualizado a dicha velocidad de cambio, supone una elevada inversión, por ello se debe alcanzar condiciones de compromiso que permitan un desarrollo sostenible. Por esta razón, los requerimientos en los referido a términos de control de emisiones, conducción y seguridad deben de establecerse más severos. No se puede contemplar que los nuevos avances tecnológicos en el transporte estén operativos en infraestructuras desarrolladas para sus versiones anteriores. Un ejemplo de la sinergia que debe existir entre vehículos y las infraestructuras se encuentra en la visión que se posee sobre el futuro de las carreteras con los vehículos autónomos. Conforme se extienda el uso de los vehículos autónomos, se requerirá que las infraestructuras estén equipadas con redes de comunicación más potentes y rápidas y con dispositivos que permitan adquirir a los vehículos información del entorno. Todas estas necesidades y nuevas normativas han acuñado el concepto de *Drive by Wire* (DBW) [14]. Puesto que la extensión del concepto DBW es muy amplio, en este trabajo se verá lo concerniente al control del acelerador, también conocido en la literatura como *Throttle by wire*, y a la dirección del volante, denominada *Steer by wire*.

Yanha Kim, Hiroshi Fujimoto y Yoichi Hori, en un su artículo *Realization of steer-by-wire system for electric vehicles using caster wheels and independent driving motors* [15], introducen la idea principal de los sistemas *by-wire*, la sustitución de las estructuras mecánicas convencionales en los vehículos son un punto a considerar en cuanto a la optimización del funcionamiento de dicho vehículo. Además, la supresión de los mecanismos tradicionales permite adoptar una mayor modularidad, dotando así al sistema de mayores grados de libertad para la distribución de componentes y para la elección de los mismos junto con de estrategias destinadas al sistema de control. No obstante, esta libertad debe ser tratada con cautela pues puede llegar a convertirse en un laberinto para los diseñadores y suponer una dificultad más que una ventaja.

### 2.2. Estado actual de los sistemas de los vehículos eléctricos

Se puede pensar, erróneamente, que los vehículos eléctricos son algo relativamente reciente y que comenzaron con la presentación del modelo *Impact* por parte de *General Motors* en el Salón de Los Ángeles en 1990. Si se profundiza en la historia de los vehículos eléctricos, sus orígenes se remontan al siglo XIX en pleno auge de las investigaciones sobre el electromagnetismo. Durante todo este siglo, las investigaciones llevadas a cabo por científicos como Jedlik, Thomas Davenport (Inventores del primer vehículo eléctrico), Robert Anderson (Primer carruaje de tracción eléctrica) o Thomas A. Edison (Figura 2.1) entre otros, llevaron al vehículo eléctrico a ser el más vendido en 1900. Un hito de esta época es el vehículo "*La Jamais Contente*" (Figura 2.2), que siendo completamente eléctrico batió el récord de velocidad superando por primera vez los 100km/h. Pero la hegemonía de estos vehículos se desvaneció cuando Henry Ford inventó la producción en cadena, los vehículos de gasolina incorporaron el motor de arranque y el precio de la gasolina descendió hasta niveles asequibles para la mayoría. En este punto, tanto el precio, como la autonomía de los vehículos, limitada en el caso de las baterías de los eléctricos, inclinaron la balanza a favor de los vehículos de gasolina, llevando casi a la extinción de los eléctricos. No fue hasta finales del siglo XX, cuando a causa de la crisis del petróleo y la preocupación ambiental, resurgieron los vehículos eléctricos y las grandes empresas de vehículos retomaron las investigaciones de los mismos.

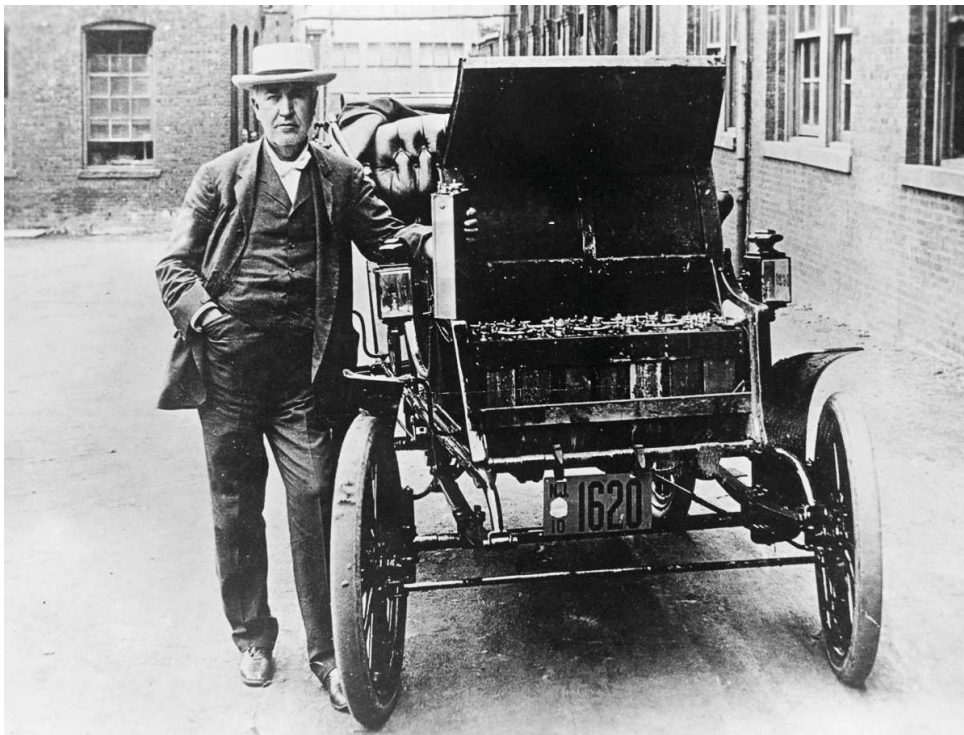


Figura 2.1: Thomas A. Edison junto a las baterías desarrolladas por él mismo para los vehículos eléctricos de la época [2].





Figura 2.2: Vehículo *La Jamais Contente*. Año 1899 [3] .

En la actualidad, tanto los vehículos eléctricos como los híbridos, se han convertido en importantes alternativas a tener en cuenta frente a los vehículos de gasolina o diesel. Esto conduce a que las empresas cada vez inviertan más para adaptarse al cambio de vehículos. Esto conlleva una vertiginosa mejora de las características técnicas (mayor autonomía y eficiencia, menor tiempo de recarga y consumo, etc.) provocada en parte por la competitividad empresarial existente. Este auge en los vehículos eléctricos sumado al constante interés en la automatización del mundo que nos rodea, lleva a un desarrollo simultáneo de dichos vehículos y la conducción autónoma.

### 2.3. Sistemas By-Wire

Pensar que lo único que evoluciona en el transporte son los vehículos es un error bastante importante y común. No se debe concebir que la conducción es similar en la actualidad a la que había en los años 60. Con el incremento de vehículos en circulación, y lo necesarios que se han vuelto para el desarrollo de nuestro día a día, es razonable que cada vez sean mayores las exigencias que se imponen a los nuevos vehículos. Dichas exigencias atienden a razones económicas y medioambientales, como se ha citado anteriormente, pero también a razones concernientes a la seguridad [16] y al confort en el trayecto [17]. Es en este punto donde entran en juego los sistemas *By-Wire*.

Estos sistemas surgen como un asistente al usuario en cada una de las tareas concernientes a la conducción. Su interpretación más coloquial consiste en sistemas electrónicos sustitutivos a los tradicionales controles manuales. Este concepto se ha extendido por el extenso campo de la ingeniería acuñando términos como *Fly-by-wire*, *Ride-by-wire*, *Drive-by-wire* y todas aquellas circunstancias susceptibles de ser controladas electrónicamente. Esta extensión en la literatura es señal de la polivalencia y vigencia de la idea que comprende. En la figura 2.3 se aprecia gráficamente el concepto by-wire.



Figura 2.3: Comparación de un sistema de dirección tradicional y un sistema by-wire [4].

Pero el concepto *by-wire* no solo abarca labores de optimización sino también la supresión de fallas producidas por descuidos o negligencias humanas que conlleven desperfectos materiales y en el peor de los casos, en pérdidas humanas. En torno al punto que concierne a la seguridad de estos tipos de vehículos se ha generado gran polémica en los últimos años con la aparición de los primeros prototipos de vehículos autónomos comerciales. La gran pregunta planteada es "¿Cómo determina el vehículo cual es el mal menor ante un accidente inevitable en el que en todas las opciones se pierde alguna vida?". Este y otros dilemas, como la reacción del coche frente a un accidente con un vehículo no autónomo, solo se pueden resolver si en las vías de circulación de vehículos estuviera todo perfectamente controlado para que no se presentasen perturbaciones imprevistas y todos los vehículos estuvieran automatizados y se pudieran comunicar entre sí en todo momento. En este planteamiento utópico no existiría la necesidad de un reglamento de circulación puesto que todos los vehículos se podrían comunicar entre sí evitando accidentes y llegando de la forma más óptima al destino.

No obstante frente a la imposibilidad actual de disponer de estos escenarios por limitaciones técnicas, como el hecho de que se precisaría una red de comunicación con capacidad de transmitir enormes cantidades de información a grandes velocidades, se debe considerar la irremediable sinergia que debe existir entre los sistemas autónomos y el ser humano, ya sea en forma de apoyo o supervisión. Una muestra de este trabajo en equipo hombre-máquina ya se puede ver en algunos coches autónomos de Tesla que para poder funcionar en modo autónomo se precisa que el conductor tenga ambas manos sobre el volante [18]. De esta forma, ante cualquier problema puede tomar el control, y de no tener las manos sobre el volante, el vehículo emite un sonido similar al producido cuando algún pasajero no tiene el cinturón de seguridad puesto.

En resumen, los sistemas *by-wire* surgen como un asistente o intermediario entre el usuario y el sistema hardware del vehículo. Desde el punto de vista ingenieril, el abanico de posibilidades que ofrecen es mucho más amplio. La modularidad de estos sistemas, si todos los fabricantes adoptan una misma norma en la fabricación de las distintas piezas, permite una mayor diversidad de posibilidades constructivas y una menor exclusividad en el aprovisionamiento de componentes. Pero más allá de la modularidad, el hecho de que la comunicación entre partes deje de ser mecánica y pase a ser electrónica permite transmitir perfiles de datos mucho más complejos vía software, lo cual es un gran avance para la implantación de sistemas de control de conducción en vehículos.

## 2.4. Control en vehículos eléctricos y autónomos

Pese a que se haga referencia al trabajo con un vehículo eléctrico, debe entenderse que lo referido a sistemas de control, no solo se ciñe a estos elementos. Todo lo referido al control en robótica móvil es extrapolable a los vehículos eléctricos y a los sistemas *by-wire*. El movimiento de un vehículo eléctrico conlleva tratar múltiples lazos de control para la consecución de un funcionamiento óptimo. La cantidad y complejidad de estos es directamente proporcional a lo complejo que se desee ascender en el nivel de control. En el artículo *Control Predictivo en Cascada de un Vehículo Autónomo* [19] se ejemplifica mediante la figura 2.4, de forma muy visual, los distintos niveles de control involucrados en el funcionamiento de un robot móvil. Cuanto más arriba se llegue en esa pirámide, mayor es la autonomía y complejidad del robot, pero cada nivel depende de los inmediatamente inferiores para su correcto funcionamiento. Por ejemplo, de nada sirve un control magnífico en la planificación dinámica de ruta si las especificaciones que establece no las pueden soportar cualquiera de sus niveles inferiores de control o trabajar para obtener el máximo rendimiento en el control de conducción del vehículo si el control de los subsistemas presenta errores que limiten su operatividad.



Figura 2.4: Niveles de control de un robot móvil autónomo.

Para abordar el problema completo del movimiento de un robot móvil se emplean gran cantidad de técnicas en la actualidad, como se refleja en *Sampling-Based Robot Motion Planning: A Review* [20]. A continuación se muestran cuatro de los planteamientos con mayor aceptación en la industria:

- Mediante la consideración del modelo cinemático [21] [22]. Este planteamiento no presenta fallas mientras se trabaje a bajas velocidades, bajas aceleraciones y las condiciones de operación se encuentren por debajo del umbral usual de operación.
- Mediante la consideración del modelo dinámico [23] [24]. Este planteamiento se focaliza en el análisis del movimiento mediante el estudio de las fuerzas involucradas y la posición como la integración de los estados del modelo dinámico.
- Mediante la consideración del modelo tanto cinemático como dinámico. Esta solución es la más completa pese a que en la literatura pocas son las veces que se emplea. En [25] se emplean ambos modelos, el cinemático para la descripción de la posición y orientación del robot y el modelo dinámico para el análisis del efecto de las fuerzas implicadas en el mismo. La estructura de control empleada para realizar el seguimiento de trayectorias es un control en cascada.
- Existen otros métodos basados en la exploración de árbol de forma aleatoria, *RRT* (Rapidly exploring Random Tree) [26]. Estos algoritmos se plantearon inicialmente como un apoyo a otras técnicas de planificación de trayectorias como PRM (Probabilistic RoadMap). En la actualidad, estos algoritmos son los más empleados en los últimos años en robótica y presentan tanto planteamientos cinemáticos como dinámicos.

Retornando a los primeros niveles de control de un robot móvil, no siempre es posible el conocimiento completo del sistema o el modelado de todos los parámetros que intervienen en el mismo. Esto repercute en los niveles superiores, por ello, requiere sistemas de control que toleren pequeñas incertidumbres. En el artículo *Robust Control for Steer-by-Wire Systems With Partially Known Dynamics* [5] se aborda este escenario mediante el modelado basado en primeros principios y una estrategia de control por realimentación con el fin de estabilizar el modelo del sistema y el compensador de deslizamiento para eliminar el efecto de la incertidumbre existente en la dinámica del sistema y del efecto de la carretera en la actuación de la dirección. En la figuras 2.6a y 2.6b se muestra el esquema de un sistema *Steer-by-wire* clásico y el control robusto propuesto para solventar el control de nivel uno. El resultado de dicho control se muestra en la figura 2.5 con un seguimiento a la señal de referencia muy homogéneo para las variaciones del suelo del vehículo.

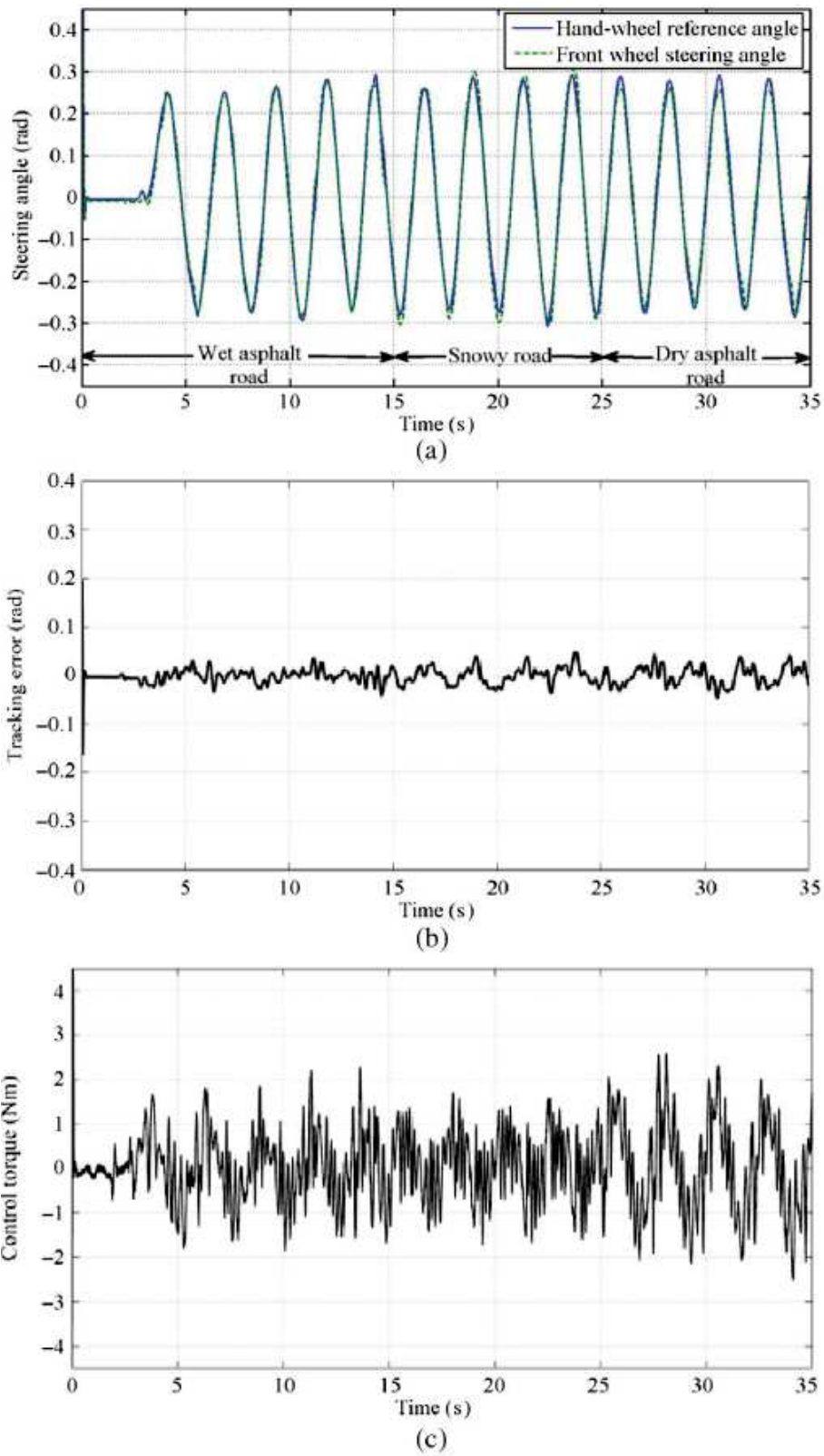


Figura 2.5: Resultado del control propuesto en [5].

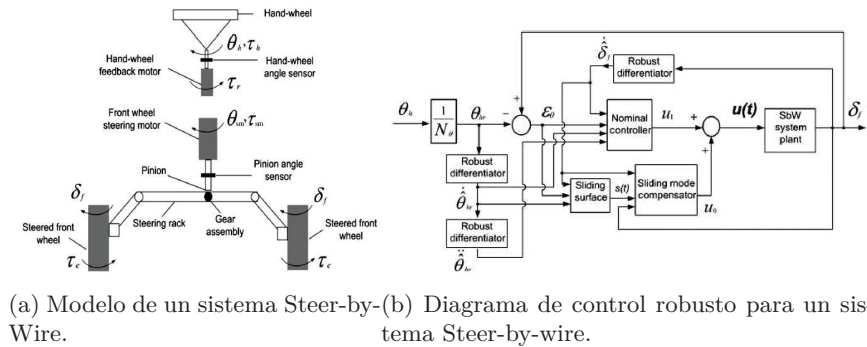


Figura 2.6: Planteamiento propuesto en [5].

Recientemente, en la literatura, otros autores han trabajado en estrategias de control avanzadas para obtener mejores características en las respuestas de estos sistemas. En [27] y [28], Y. Marumo implementa un control por realimentación implementando la técnica cuadrática lineal para el control *Steer-by-wire* en una motocicleta. Pese a la eficiencia de esta técnica, cuando la incertidumbre del sistema es elevada, pierde garantías de rendimiento en lazo cerrado. En [29], se hace uso del índice de giro de realimentación como un gradiente de subviraje para el control y considera el control *by-wire* fundamental para el control en altas velocidades. P. Setlur en [30] emplea como estrategia de control para estos sistemas un controlador de seguimiento no lineal. Con esta implementación se fuerza el error, de forma asintótica, a cero frente a incertidumbres en los parámetros del sistema. En el caso de los sistemas en los que el robot móvil interactúa con el ser humano, una de las técnicas aplicadas es el control de impedancia [31] [32] [33]. Esta técnica se basa en la elaboración de un modelo de referencia en el cual se incluyen los parámetros mecánicos involucrados como el par del volante o la fuerza realizada en el piñón-cremallera. Si además se emplea un control adaptativo para esta implementación, se puede tratar las no linealidades e incertidumbres presentes entre los ángulos de desplazamiento y las fuerzas laterales ejercidas sobre los vehículos [34]. Otra técnica muy extendida para sistemas de control que presenten cierta incertidumbre es la estimación de parámetros en línea, cuyo esquema se representa en la figura 2.7.

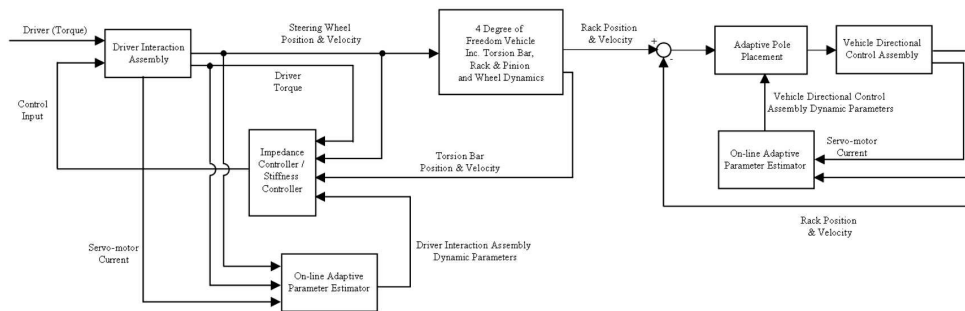


Figura 2.7: Esquema de control empleando estimador de parámetros en línea. [6]

En [6], la técnica empleada para el sistema de control es la asignación de polos haciendo uso de la estimación de parámetros. Esta implementación presenta buenos resultados para la reducción del error en problemas de seguimiento, no obstante, los efectos de la superficie de desplazamiento no se consideran en la estimación de parámetros. Esto conlleva la desventaja de no poder obtener los parámetros del modelo real del vehículo sin la perturbación introducida por el tipo de suelo si no se realiza un modelado basado en primeros principios. Esto puede ser considerado un inconveniente si el sistema presenta una determinada incertidumbre y el objetivo comprende el modelado preciso y control del sistema.

No obstante, si la dinámica del sistema se conociese de forma exacta, la estimación de parámetros en línea presenta la virtud de poder obtener los parámetros del sistema en lazo cerrado y, conocidos dichos parámetros de forma teórica, obtener la perturbación introducida por el tipo de suelo. Este tipo de modelado, al no ser basado en primeros principios, no supone la obtención de resultados extrapolables a cualquier robot móvil cualesquiera que sean sus propiedades constructivas para el desarrollo de sistemas de control anticipativo o feedforward perfectos [35] [36], pero si supone un punto de partida para el conocimiento y tratamiento del efecto de estos agentes como perturbaciones en los sistemas a controlar. Si por otra parte, el modelado del sistema se realiza basándose en primeros principios, una vez determinadas las ecuaciones para la estimación de parámetros del vehículo y se considera el efecto del suelo en ellas, es posible realizar una correcta estimación de parámetros en línea y por tanto un mejor sistema de control. Este es el caso desarrollado en el artículo "*Adaptive feedforward control of the steer-by-wire system based on the online parameter estimator*" [7]. En él las estimaciones se realizan mediante el estudio de la respuesta del sistema frente a señales de entrada sinusoidales en las cuales se varían la amplitudes y frecuencias de las señales de entrada y del valor de la ganancia del controlador. El resultado de la comparación de esta estrategia de control frente a un controlador PD se muestra en la figura 2.8, donde se aprecia un menor error para una señal de entrada de alta frecuencia.

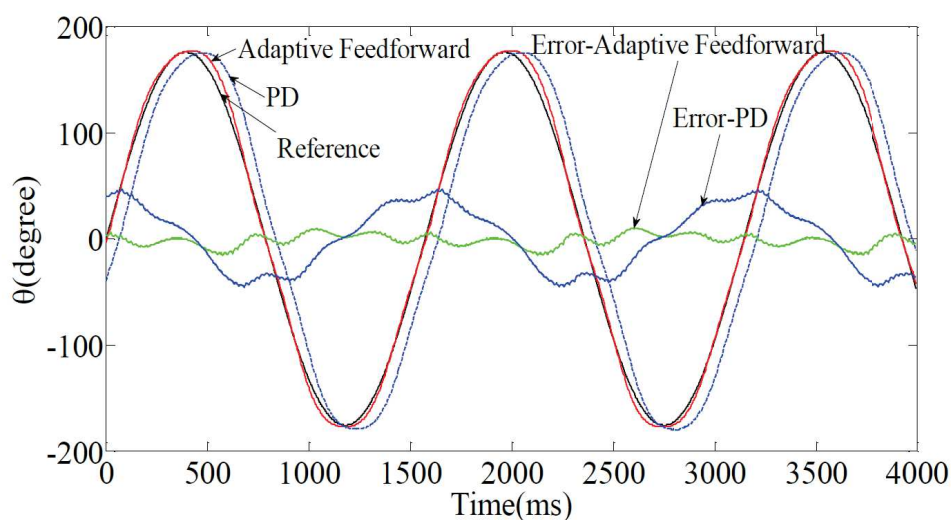


Figura 2.8: Resultados de emplear un control anticipativo frente a un controlador PD. [7]





## 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 (Figura 1.3). Este vehículo fue adquirido en el año 2010 a través de 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 las distintas investigaciones llevadas 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 [12] (*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. Como se describe más detalladamente en la siguiente sección, a pesar de que la tensión suministrada es de 48V, se considera necesario la instalación de tres fuentes de tensión que suministren 5V, 12V y 24V puesto que no todos los dispositivos electrónicos que incorpora el vehículo trabajan a igual tensión.

### 3.2. Componentes 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. Controlador Curtis [1]

Este componente se trata de un controlador Curtis PMC modelo 1268-5403 (Figura 3.1) 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 la parte del vehículo que requiere dicha potencia.

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 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.



Figura 3.1: Imagen del Controlador Curtis [8].

### 3.2.2. PC Industrial

Se trata de un PC diseñado especialmente para operar correctamente en ambientes hostiles (Figura 3.2). Esta condición debe estar presente pues su emplazamiento se encuentra en el maletero del vehículo y cuando se esté trabajando pueden alcanzarse elevadas temperaturas. Su función principal es la de servir de enlace entre el usuario y el sistema ROS encargado del control situado en la Raspberry Pi. Esta comunicación se lleva a cabo mediante una conexión *SSH*(Secure SHell) a la dirección 10.0.0.3 (Dirección fija configurada en la Raspberry Pi) vía Ethernet. Como tarea secundaria, o no crítica para la operatividad del sistema, sirve de *bridge* para la Raspberry Pi otorgándole de esta forma acceso a internet. Otras funciones que no están relacionadas de forma directa con este trabajo son la lectura de los encoders situados en las ruedas posteriores del vehículo (Odometría) y la operación con el sensor láser situado en la parte frontal del vehículo.



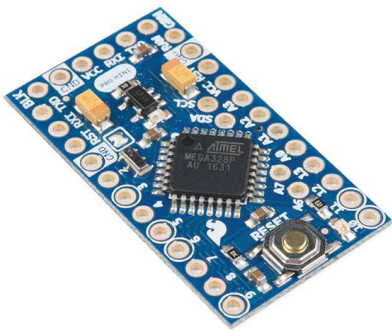
Figura 3.2: Imagen del PC industrial.

### 3.2.3. Raspberry Pi

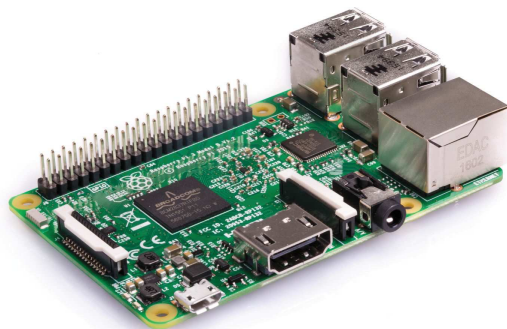
La Raspberry Pi empleada se trata del modelo 3 B (ver figura 3.3b), el cual forma parte de la tercera generación de estas placas. Las especificaciones técnicas de interés para este proyecto son que posee un procesador *Quad Core 1.2GHz Broadcom BCM2837 64bit CPU*, una memoria RAM (Random Access Memory) de 1GB, cuatro puertos USB (Universal Serial Bus), un puerto Micro USB para la alimentación a 5V hasta 2.5A y una conexión HDMI (High-Definition Multimedia Interface), fundamental para las configuraciones iniciales del sistema. Utiliza un sistema operativo Ubuntu Mate 16.04.2 sobre el cual se ha instalado el software *ROS*, el cual se describe en la siguiente sección. Su función principal es la de soportar el sistema de control y servir de conexión entre los distintos componentes que intervienen en dicho sistema.

### 3.2.4. Arduino Pro Mini

Se trata de una placa con un microcontrolador basado en el ATmega328 (ver figura 3.3a). Posee 14 pines de entrada/salida digital de los cuales 6 poseen una configuración especial que les permiten ser configurados para señales de salida PWM (Pulse Width Modulation) y 6 pines para entradas analógicas. Esta variedad de pines y su reducido tamaño hace de esta placa una opción con gran potencial frente a otras alternativas. En este proyecto se ha empleado como la tarjeta de E/S principal, actuando de intermediario directo entre las unidades software y los distintos componentes hardware a través del nodo *Arduino\_daq* que se explica en la siguiente sección.



(a) Arduino Pro Mini [37].



(b) Raspberry Pi 3 [38].

Figura 3.3: Componentes hardware encargados de soportar el sistema de control.

### 3.2.5. 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*. [1]

### 3.2.6. PowerBox

Este componente consiste en una caja diseñada por el estudiante en el periodo de prácticas curriculares (ver figura 3.4c). Sus dos funciones principales son la de protección de la rama electrónica del vehículo y la de poder encender los componentes de forma individual o en pequeñas agrupaciones.

A esta caja están conectadas las dos fuentes CC-CC, los dos reguladores y el PC industrial. Como protección principal se emplea un fusible de 20A a la entrada y un display que monitoriza la tensión, intensidad y consumo de todos los componentes conectados. Además se han implementado cinco interruptores que modularizan el encendido de cada componente y uno de ellos actúa como interruptor principal. Esto permite cierto nivel de ahorro energético que se produciría por el encendido de componentes que no se empleen en dicho momento y el encendido de componentes de forma individual para realizar ensayos sobre ellos.

### 3.2.7. Fuentes de alimentación

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.

La fuente CC-CC de 24V, SD-500L-24 mean well (ver figura 3.4a), se emplea para la alimentación del motor de la dirección, de los voltímetros situados en el maletero para la lectura de datos mediante la tarjeta NI-USB6211, de los distintos amperímetros de la marca LEM situados por el vehículo (Baterías y motor) y el sensor láser de la parte frontal. La fuente de 12V, reutilizada de otros proyectos realizados anteriormente, suministra corriente al amperímetro Honeywell que cuantifica la corriente suministrada por las baterías. El regulador de 5V localizado en la caja denominada *PowerBox* (sección 3.2.6) se emplea para la alimentación de la Raspberry Pi situada en la parte delantera del vehículo. 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 3.4b.

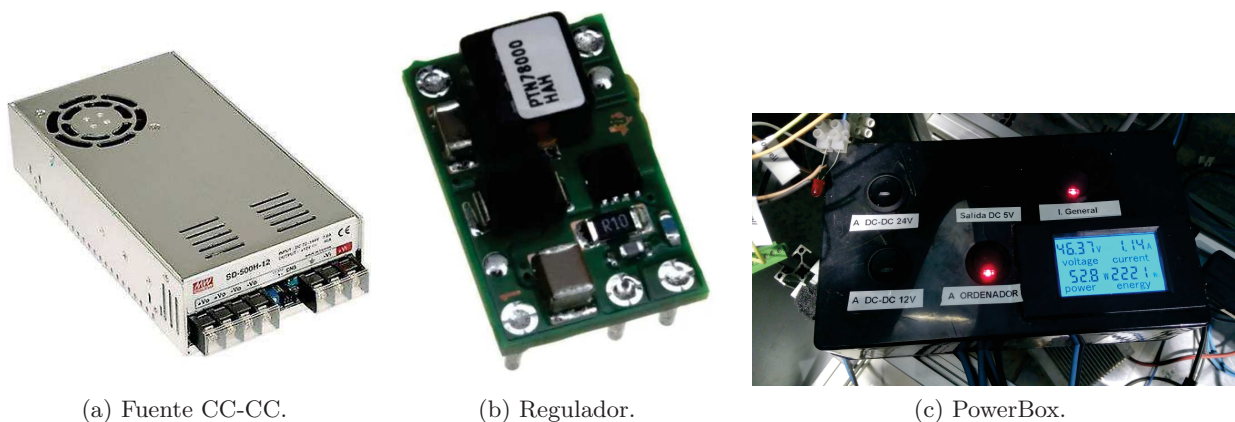


Figura 3.4: Componentes encargados de la administración de corriente.

### 3.2.8. Motor dirección

El motor instalado encargado del movimiento de la dirección del vehículo es un motor de corriente continua, CC, Maxon RE 50 d50mm (ver figura 3.5a), escobillas de grafito, 200 Vatios, figura 3.5a. Este motor tiene una alimentación de 24V y su control físico está regido por la placa Pololu (*Pololu High-Power Motor Driver 36v20 CS*). Esta placa hace de etapa de potencia entre las señales generadas por la placa Arduino Pro-mini, PWM y sentido de giro, y la actuación directa sobre el motor. Esta placa incluye, entre otras posibilidades, la posibilidad de medir la corriente consumida por el motor. Además, el motor tiene instalados un reductor planetario GP 62 A d62mm, reductora 100:1 y un encoder en cuadratura incremental HEDL5540,500ppv. 3 canales, con line driver RS 422. Con este último encoder se han realizado todos los ensayos de modelado de la dirección desarrollados en el capítulo 5.

### 3.2.9. Encoder dirección

Pese a la buena calidad de señal que suministra el encoder incremental acoplado al motor, la incertidumbre de la posición absoluta de la dirección supone un problema si el sistema no comienza en cero. Como solución a este problema se lleva a cabo el acoplamiento de un encoder absoluto en el tramo en el cual se produce la unión del motor con el mecanismo de la dirección. El encoder empleado se trata del modelo EMS22A del fabricante Bourns (ver figura 3.5b). Dicho encoder posee una resolución de 10 bits, una tensión de alimentación de 5V o 3.3V  $\pm 10\%$  y una corriente máxima de 20mA.



Figura 3.5: Componentes hardware descritos en las secciones 3.2.8 y 3.2.9.

### 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. En caso de demandar información, la encuentra disponible en ese mismo “lugar”.

#### 3.3.1. Robotics Operating System

*ROS* es un sistema operativo orientado al diseño de software para robots con *Inteligencia Artificial, AI*. Tiene sus orígenes en el año 2007 cuando la Universidad de Stanford comenzó el proyecto 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, para darle soporte, 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 [39].

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, 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 encuentran instalados dos *packages*: *ual-ecar-ros-pkg* y *arduino-daq-ros-pkg*. Ambos paquetes, desarrollados 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* [40] es la nomenclatura empleada para hacer referencia a aquellas variables que forman parte de la comunicación entre nodos de *ROS*. El comando *rqt\_graph* [41] 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 *rosvbag* [42]. 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 *rosvbag 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 *rosvbag 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* en eCARM

Para el correcto funcionamiento del conjunto software, el sistema requiere de la ejecución de cuatro nodos fundamentales. En la tabla 3.2 se introducen los distintos nodos desarrollados. En la figura 3.6 se muestra un mapa global de las conexiones que se realizan a nivel software durante el correcto funcionamiento del vehículo.

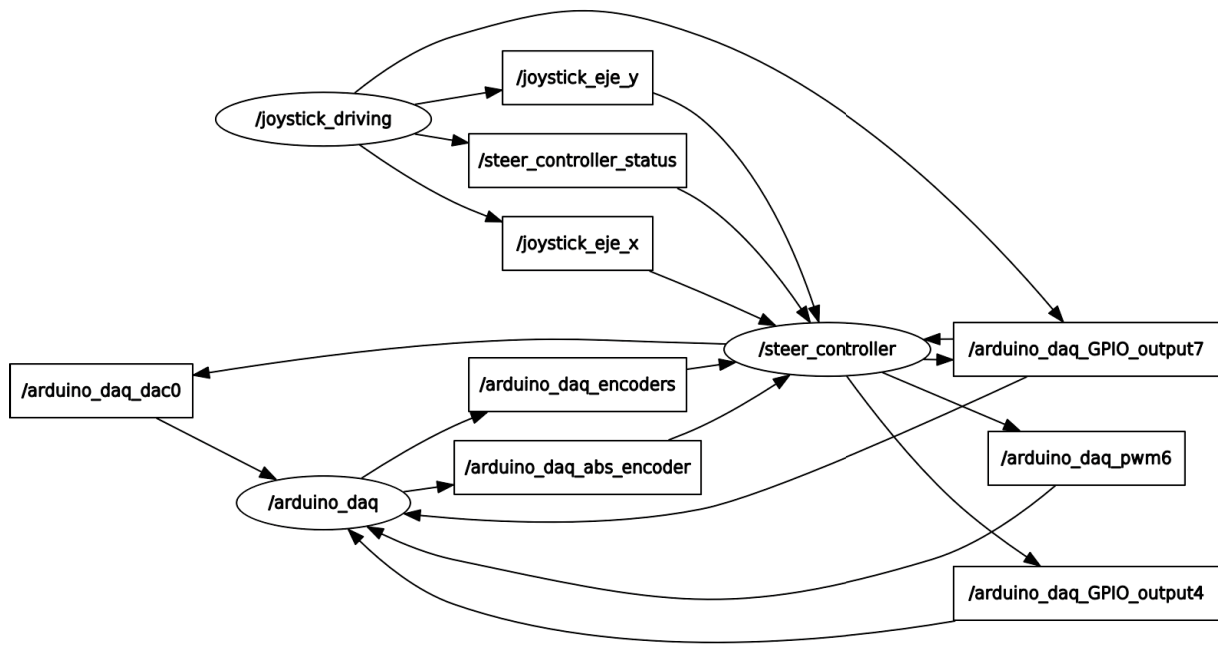


Figura 3.6: Captura del sistema implementado en el vehículo eCARM con el comando *rqt\_graph*.



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.
Arduino_daq	Tiene el control de la tarjeta Arduino mini-pro usada como tarjeta de E/S.
Steer_controller	Recoge la estructura principal de control. A diferencia de los nodos anteriores, es puramente software sin trabajar directamente con ningún componente hardware.

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

### Roscore

La ejecución del nodo *roscore* aúna la ejecución de todos aquellos nodos y programas que requiere un sistema basado en *ROS* para funcionar [43]. Gracias a este comando, para el usuario es transparente la ejecución de aquellos comandos que se debe realizar al comienzo de cada sesión, pudiendo así focalizar la atención del usuario en el desarrollo de su proyecto.

### Joystick\_driving

Este nodo se encuentra ubicado dentro del *package ual-ecar-ros-pkg* disponible en un repositorio en la plataforma *Github* [44]. Su función principal es la de comunicar los distintos botones y ejes que posee el joystick físico con el sistema software. Algunos de ellos se transmiten directamente como *topics* con funciones directas en el sistema como por ejemplo */arduino\_daq\_GPIO\_output7*, el cual es el encargado de controlar el relé conmutador entre la marcha directa y atrás. Otros *topics* tienen por función emplearse en otros nodos para realizar cálculos como es el caso de los ejes del joystick izquierdo o el *topics /steer\_controller\_status* el cual establece si el control sobre la posición de la dirección del vehículo es manual o automático, los cuales se emplean posteriormente en el nodo *steer\_controller*.

De forma interna, posee variables asignadas a cada botón que realizan tareas específicas sobre las variables que más tarde se asignarán a los *topics* que se publiquen por el nodo. El eje vertical del joystick izquierdo está asociado a la propulsión del vehículo mientras que el eje horizontal del joystick derecho se asocia a la dirección del mismo. Una de dichas tareas específicas está realizada por los botones “cuatro” y “cinco” del joystick los cuales aumentan en un 250 % la sensibilidad de los ejes vertical del joystick izquierdo y horizontal del joystick derecho. Como cada eje tiene un rango de movimiento limitado se ha optado por reducir un 50 % el valor emitido por cada uno optando por una ventana de movimiento de la mitad, duplicando así la resolución obtenida. Para poder desplazarse por el rango completo se han empleado los botones “cero” y “uno” para fijar el valor medio de operación al que se esté introduciendo de cada eje en el momento de su accionamiento.

Estas implementaciones permiten un mayor manejo de cada variable por parte del usuario. A continuación se muestra una captura donde se muestra los *topics* relacionados con este nodo mediante la ejecución del nodo *rqt\_graph* (Figura 3.7).

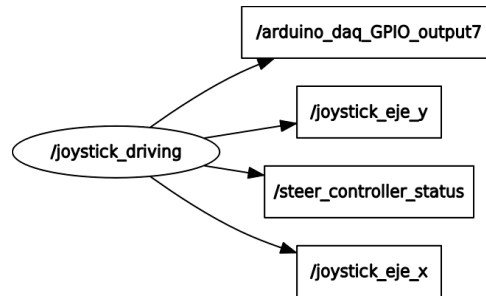


Figura 3.7: Captura del nodo *joystick\_driving* con el comando *rqt\_graph*.

#### Arduino\_daq

Este nodo está incluido en un *package arduino-daq-ros-pkg* disponible en un repositorio de *GitHub* [45]. Este *package*, además de los archivos requeridos para el correcto funcionamiento de *ROS*, incluye el firmware AVR8 que se encuentra precompilado para placas con microcontroladores atmega328p, como es el caso de la arduino Pro Mini empleada para este proyecto. No obstante existe la posibilidad de recompilar el proyecto en placas con mayor número de pines de entrada/salida. Este nodo se emplea como intermediario entre *ROS* y el hardware de la placa arduino. Esto permite configurar vía software los distintos pines del microcontrolador en función de la utilidad para la cual se vayan a emplear cada uno.

Para esta configuración del firmware concreta, se dan cuatro posibles tipos de *topics* para trabajar en el entorno de *ROS* concerniente a este nodo: GPIO inputs/outputs (E/S de propósito general), ADC inputs (Entradas analógicas), PWM outputs (Señales PWM de salida) y posibilidad de incorporar dos encoders en cuadratura, configurando cada pin concerniente al encoder de la forma propicia. En última instancia se ha incorporado un quinto *topic* encargado de reportar la lectura del encoder absoluto. A causa del alto nivel de ocupación de pines empleados, se estima oportuno el empleo de tres pines analógicos como GPIO para la lectura de este último encoder. Esto es posible a causa de que los pines 14-17 presentan las función digital/analógica multiplexada, quedando a disposición del usuario la elección de la misma.

Atendiendo a cada una de las funciones anteriores, los *topics* involucrados en este nodo con los que trabaja *ROS* en este proyecto poseen nombres que permiten identificar claramente su configuración, facilitando así su comprensión y empleo desde la línea de comandos a la que tiene acceso el usuario con el sistema en marcha. A continuación se muestra una captura donde se muestra los *topics* relacionados con este nodo mediante la ejecución del nodo *rqt\_graph* (Figura 3.8).

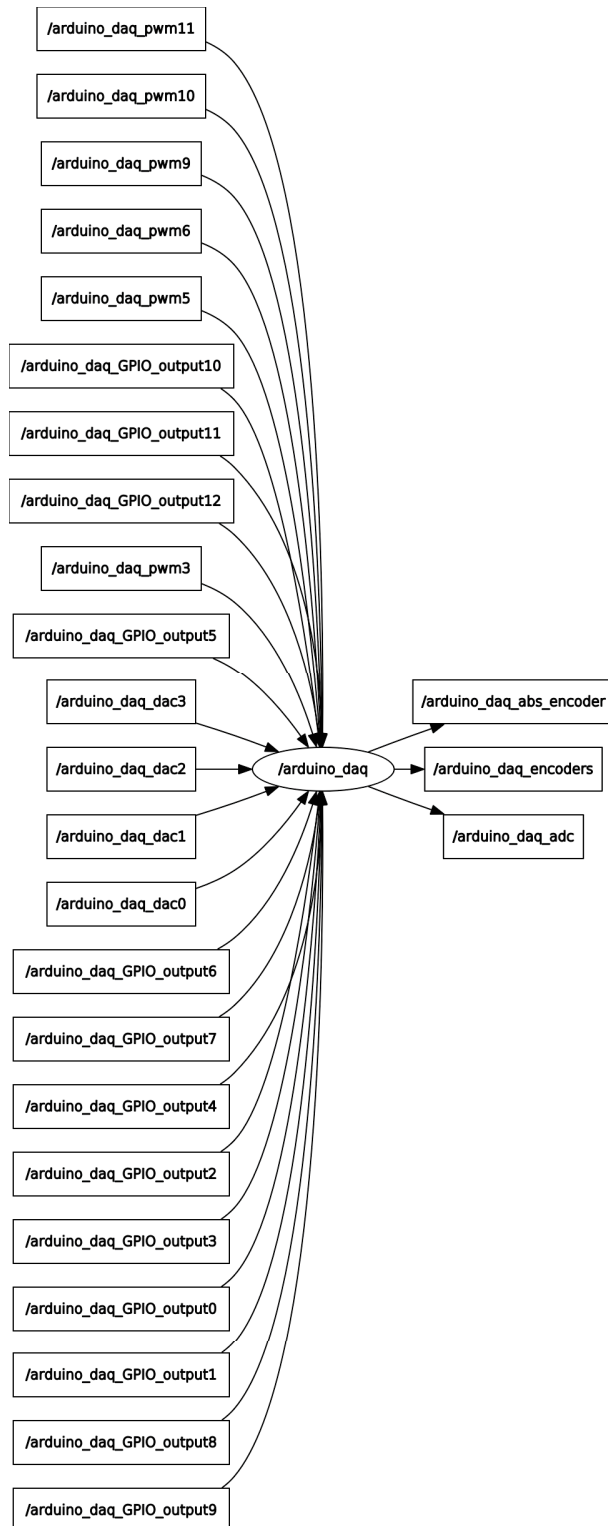


Figura 3.8: Captura del nodo `arduino_daq` con el comando `rqt_graph`.

### Steer\_controller

Este nodo concentra el trabajo realizado en los capítulos 4 y 5. Al igual que el nodo *joystick\_driving*, se encuentra comprendido dentro del *package ual-ecar-ros-pkg* localizado en el repositorio de *Github* [44]. Este nodo opera de forma software exclusivamente, es decir, no interactúa de forma directa con ningún elemento hardware como es el caso de los dos descritos en los epígrafes anteriores. Su función principal es la de procesar los *topics* generados por el nodo *joystick\_driving*, recabar la información del sistema reportada por el nodo *arduino\_daq* y generar los necesarios para ser leídos nuevamente por el nodo *arduino\_dac* con el fin de un correcto funcionamiento de los actuadores. En la figura 3.9 se muestra la captura de la ejecución del nodo *rqt\_graph* donde se muestran los *topics* involucrados en este nodo.

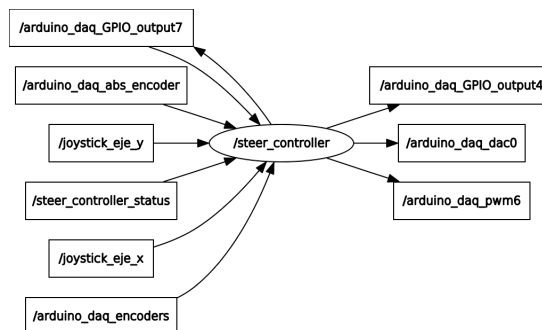


Figura 3.9: Captura del nodo *steer\_controller* con el comando *rqt\_graph*.

En este nodo se realiza la implementación del sistema de control en tiempo discreto para el sistema *Steer-by-wire*, la adecuación de la señal de control para el sistema *Throttle-by-wire* y las implementaciones de mecanismos de control como es la estrategia de transferencia sin salto. Otra implementación de gran valor es la redundancia en la medida de los dos encoders concernientes a la dirección del vehículo. El fragmento de código referido a esta implementación solo tiene como finalidad la advertencia al usuario de la discrepancia de datos. El valor reportado por el encoder absoluto será predominante frente al encoder relativo a no ser que con el paso del tiempo y transcurso de los ensayos se aprecie que el fallo proviene de él.

Puesto que el código es muy extenso y su comprensión se vuelve dificultosa al visualizarlo en distintas páginas, a continuación se desglosará y explicará por fragmentos según su finalidad. En primer lugar es fundamental la declaración correcta de variables para operar en cualquier programa correctamente. En la figura 3.10 se muestran los tres tipos de declaraciones empleadas en este caso. En las tres primeras líneas se muestra una declaración en la cual se especifica el tipo y nombre de la variable, más adelante en el código se operará con ellas. El segundo tipo de declaración es aquella en la cual la declaración se realiza la primera vez que se opera con dicha variable y comprende su tipo, nombre y valor. El tercer tipo de declaración es la empleada para las variables empleadas como *topics* y presentan una tipología especial definida en librerías del proyecto. No obstante presentan una estructura similar a las definidas en primera instancia.

```

double voltaje_pedal ,rpm;
bool b2 ,b1 ,b3;
int pwm_steering ,max_p = 40;
std_msgs::UInt8 msg_ui;
std_msgs::Float64 msg_f;
std_msgs::Bool msg_b;
// Lectura del modo de control
bool ok = Status_mode;

```

Figura 3.10: Declaración de variables.

Atendiendo a la figura 3.10, se aprecia que se operará en las sucesivas secciones de código con tres tipos de variables principalmente. El tipo más empleado es el “double” o “Float64” que son variables de punto flotante de doble precisión de 64bits para perder la menor información en el proceso matemático. El siguiente tipo es el “bool” el cual se emplea para variables lógicas tales como el sentido de giro de los motores o el modo de control del sistema. Finalmente la tipología “int” y “UInt8” se emplea para la señal PWM que toma valores enteros comprendidos entre 0 y 254.

En la figura 3.11 se muestra la lectura del *topic* generado por el nodo *arduino\_daq* referido a la lectura del encoder incremental. El nodo publica el *topic* con el valor leído directamente de la tarjeta de E/S y es en este fragmento de código donde se convierte, haciendo uso de los factores de conversión facilitados en la hoja de datos del fabricante del encoder, de pulsos a grados con los que operar en el lazo de control. También se implementa aquí la redundancia de valor de los encoders para comprobar que todo funcione correctamente.

```

m_Encoder[0] = - (msg->encoder_values[0]) / 1824.9;
.
.
if (red){
    double ang_inicial = m_Encoder_Abs[0];
    double aux = m_Encoder[0];
    red = false;
}
m_Encoder[0] = m_Encoder[0] - aux + ang_inicial;
if (abs(m_Encoder[0]-m_Encoder_Abs[0])>2)
{
    red = true;
}

```

Figura 3.11: Lectura encoder incremental.

En la figura 3.12 se muestra la obtención del valor de la velocidad de giro de la dirección, denominado rpm. Este valor se obtiene como la diferencia de la lectura actual del encoder con la lectura de la iteración anterior dividida por el tiempo de muestreo del sistema, 0.05s en este sistema. Además de la determinación de este valor, también se obtiene la estimación de la señal de velocidad empleada en el predictor de Smith como se desarrolla en el capítulo 5.

```
rpm = (m_Encoder[0] - m_Encoder[1]) / 0.05;
m_ys[0] = m_ys[1] * 0.1709 - 0.0775 * m_us[1+3];
```

Figura 3.12: Lectura rad/s y estimación de la velocidad.

A continuación, en la figura 3.13, se muestra la generación de la señal PWM, publicada en el *topic /arduino\_daq\_pwm6*, en el control manual de la dirección. En este caso, se emplea el *topic* publicado por el nodo *joystick\_driving* “Eje\_x” y puesto que su valor comprende el rango  $[-1, 1]$  y el valor máximo del PWM puede ser 254, se considera la señal como el producto mostrado en la primera línea forzado a adoptar una tipología de número entero. Posteriormente, en función del signo resultante se establece el valor booleano de la variable “msg\_b” que controla el sentido de giro del motor maxon mediante el *topic /arduino\_daq\_GPIO\_output4* y se asigna a la señal *pwm* su valor absoluto y este al vector de valores de la señal de control.

```
pwm_steering = (int)(Eje_x * 254);
m_us[0] = pwm_steering;

if (pwm_steering < 0){
    pwm_steering = - pwm_steering;
    msg_b.data = false;
}
else{
    msg_b.data = true;
}
msg_ui.data = pwm_steering;
m_pub_rev_steering.publish(msg_b);
m_pub_pwm_steering.publish(msg_ui);
```

Figura 3.13: Procesado señal PWM.

El procesado de la señal *Throttle-by-wire* se lleva a cabo de forma manual, figura 3.14, puesto que, como se verá en el capítulo correspondiente a este sistema, no se ha podido implementar una estrategia de control sobre el mismo. En la primera línea se observa la generación de la señal que se introduce en el controlador *Curtis*. Tras realizar ensayos sobre el motor se observó que posee una zona muerta de 1V, por lo que se ha optado por implementar que la señal mínima en todo momento sea de dicho valor. Con esto se consigue eliminar la zona muerta del rango de operación del sistema de mandos físico. Acto seguido, a ese valor de inicio se le suma la señal generada en el nodo *joystick\_driving* para la propulsión del vehículo. Conocido que el valor máximo de la señal son 5,76V y el *topic* abarca el rango  $[0, 1]$ , se opta por multiplicar dicho valor por  $V_{max} - V_{zonamuerta}$  obteniendo un valor de 4,76V. En este caso, la variable se publica en el *topic /arduino\_daq\_dac0* y por el rango de valores que adopta,  $[0,00V, 5,76V]$ , requiere un tipo de dato “Float64” para evitar en la medida de lo posible la pérdida de información en las operaciones matemáticas realizadas.

```

voltage_pedal = 1.0 + Eje_y * 4.76;

msg_f.data = voltage_pedal;

m_pub_voltage_pedal.publish(msg_f);

```

Figura 3.14: Procesado señal acelerador.

En el código mostrado a continuación en la figura 3.15 se observa el procesado de la señal de referencia del sistema *Steer-by-wire* cuando se encuentra en control automático. Puesto que, como se ha dicho con anterioridad, “Eje\_x” se trata de una señal comprendida entre  $[-1, 1]$  para generar la referencia se multiplica por el valor máximo que alcanza el mecanismo en sus extremos. Como opción para la eliminación de la sobreoscilación que presenta el sistema se implementa el filtro, no obstante no se activará su función por las razones comentadas en la sección 5.3.2. La discretización de dicho filtro se muestra en la última línea de este fragmento de código con el cálculo de la variable “*m\_R\_steer[0]*”.

```

/*      Lectura de la referencia de posicion */
m_R_steer[0] = (double)(Eje_x * 40);
//      m_R_steer_f[0] = (double)(Eje_x * 40);

/*      Filtro en la referencia */
//      m_R_steer[0] = 0.9649 * m_R_steer[1] + 0.0351 * m_R_steer_f[0];

```

Figura 3.15: Procesado de la referencia.

En la figura 3.16 se muestra un código necesario tras la realización de diversos ensayos sobre el sistema real. El fenómeno causante de dicha necesidad es la limitación en el suministro de corriente del convertidor CC-CC que alimenta el motor de la dirección y el monitor que sirve de interfaz para el usuario. Se ha observado que cuando el motor recibe elevadas señales de control, en el inicio del movimiento, como ocurre en todos los motores eléctricos, la corriente que debe suministrar la fuente es insuficiente para todos los dispositivos conectados. Esto provoca que el monitor sufra una desconexión momentánea y probablemente, aunque no se han llegado a monitorizar durante un pico de corriente, los sensores instalados suministrados por la misma fuente reporten valores erróneos durante ese instante. Por el riesgo para la seguridad del usuario que puede suponer este incidente se decide limitar la pendiente de subida de la referencia (o señal de control) a seguir. En la variable “*sat\_ref*” se define la máxima pendiente de subida y de registrarse un valor superior en la variable “*pendiente*”, calculada de forma similar a la velocidad de giro de la dirección descrita anteriormente, se bloquea la señal de referencia al valor que tenía en la iteración anterior más el valor fijado de límite.

```
/*      Limitacion en la pendiente de subida de la referencia para
      proteccion contra sobrecorrientes en el actuador*/
      double sat_ref = 20;

      double pendiente = (m_R_steer[0] - m_R_steer[1]) / 0.05;

      if (pendiente >= sat_ref)
      {
          m_R_steer[0] = (m_R_steer[1] + sat_ref);
      }
```

Figura 3.16: Protección pendiente de subida.

El siguiente fragmento de código, mostrado en la figura 3.17, comprende la clausura del lazo de control externo y la generación de su señal de control. En primer lugar se determina el error, el cual se obtiene de la diferencia entre la referencia obtenida de los códigos anteriores y se registra como un vector con la nomenclatura “*m\_ep[0]*”. Tras la determinación de este parámetro, con el cual se cierra la realimentación del lazo de control, se introduce la ecuación del controlador en tiempo discreto diseñado en la sección 5.3.2.

```
/*      Lazo de control externo. Posicion de la direccion*/

/*      Calculo del error */
m_ep[0] = m_R_steer[0] - m_Encoder[0];

/*      Controlador lazo externo */
m_up[0] = m_up[1] + 2.9082 * m_ep[0] - 2.8061 * m_ep[1];
```

Figura 3.17: Determinar error lazo externo y señal del controlador.

Una vez determinada la señal de control del lazo externo, se procede a determinar el control del lazo interno, cuya representación en *C++* se muestra en la figura 3.18. El error de este lazo se determina como la diferencia entre la señal de control del lazo externo y la estimación de la señal de velocidad de la dirección (figura 3.12). Si se estima oportuno, se puede añadir realimentación real a este lazo añadiendo a la fórmula anterior la resta del término resultante de la diferencia entre el valor de la velocidad de giro de la presente iteración con el valor obtenido en el *Predictor de Smith* tres iteraciones atrás.

```
/*      Lazo de control interno. Velocidad de giro de la direccion*/

/*      Calculo el error */
m_es[0] = m_up[0] - m_ys[0]; /* - (rpm - m_ys[3]); */

/*      Introduccion de la ecuacion del controlador */
m_us[0] = (int)(m_us[1] - 2.8261 * m_es[0] - 0.1750 * m_es[1]);
```

Figura 3.18: Error lazo interno y señal del controlador.



Las próximas tres figuras representan tres mecanismos de protección necesarios para preservar el sistema real de daños causados por señales de control inadecuadas. La primera medida de protección, figura 3.19, consiste en una limitación software para evitar que se intente sobrepasar los límites mecánicos de la estructura del sistema real. Si este mecanismo detecta que se ha alcanzado un extremo, solo permite que se introduzcan señales de control que muevan el mecanismo hacia su zona central. Esta implementación se encuentra acompañada por un mensaje para el usuario que le notifica este límite para que modifique la referencia. Dicha notificación desaparecerá cuando el sistema descienda el valor de su salida cinco grados hacia el centro del mecanismo.

```

int m_v= m_us[0] + m_u[1]; // Variable para el mecanismo antiwindup

/*      Proteccion en el extremo del mecanismo*/

if (abs(m_Encoder[0]) >= max_p)
lim = 1;
if (abs(m_Encoder[0]) <= (max_p - 5) && lim == 1)
lim = 0;
if (lim ==1)
{
    ROS_INFO("Direccion en el extremo, reduzca referencia");
    if(m_Encoder[0] > 0 && m_us[0] > 0)
        m_us[0] = 0;
    if(m_Encoder[0] < 0 && m_us[0] < 0)
        m_us[0] = 0;
}

```

Figura 3.19: Mecanismo de protección en los extremos.

El siguiente mecanismo de protección consiste en la implementación de la saturación de la señal de control del lazo interno (figura 3.20). Esto evita comportamientos anómalos a nivel hardware en la interpretación de una señal que escapa de su ventana de operación. Este mecanismo de protección está además respaldado por la etapa de diseño de los controladores, en la cual un factor concluyente para la determinación del controlador definitivo ha sido el empleo de señales de control suaves y valores lejanos de los límites.

```

/*      Saturacion */

if (m_us[0] > 254)
{
    m_us[0] = 254;
}
if (m_us[0] < -254)
{
    m_us[0] = -254;
}

```

Figura 3.20: Saturación.

El último mecanismo de protección, figura 3.21, es el denominado “*Anti-windup*”. Pese a que los cálculos de este mecanismo se determinan en este fragmento de código, su auténtico efecto se localiza en la declaración de la variable “*m\_v*” en el código anterior. Esta estructura del mecanismo no es la tradicional, sino más bien una adaptación de la misma a las condiciones del sistema sobre el que se implanta sin tener que recalcular parámetros del controlador principal en cada iteración. Además incorpora un mensaje por pantalla que reporta una advertencia al usuario a través de la interfaz gráfica de que este mecanismo se encuentra operativo, ya que en condiciones normales de operación no debería ser así.

```
/*      Mecanismo Anti-windup*/

    if(m_us[0] - m_v != 0){
        ROS_INFO(" Activacion del mecanismo anti-windup");
        m_antiwindup[0] = (m_us[0] - m_v) / sqrt(0.0283);
    }
    else{
        m_antiwindup[0] = 0;
    }
    m_u[0] = int(0.5 * (2 * m_u[1] + 0.05 * (m_antiwindup[0] +
        m_antiwindup[1])));
```

Figura 3.21: Mecanismo Antiwindup.

En la figura 3.22 se encuentra el fragmento de código encargado de publicar el valor de los *topics* encargados de controlar la señal de control en la tarjeta de E/S. Puesto que la placa pololu, epígrafe 3.2.8, requiere que la señal PWM esté comprendida entre  $[0, 254]$  y el sentido determinado por un pin lógico, al estar las señales de control comprendidas entre  $[-254, 254]$ , se requiere que al ser publicadas sea su valor absoluto y el sentido quede registrado en un *topic* de tipo booleano.

```
/*      Envio de datos a los parametros correspondientes de ROS*/

    if (m_us[0] < 0){
        msg_b.data = false;
    }
    else{
        msg_b.data = true;
    }

    msg_ui.data = abs(m_us[0]);
    m_pub_rev_steering.publish(msg_b);
    m_pub_pwm_steering.publish(msg_ui);

    ROS_INFO("PWM: %a", msg_ui.data);
```

Figura 3.22: Envío de datos en ROS.

Finalmente, el último fragmento de código que se considera relevante en el funcionamiento de todo el código descrito anteriormente queda reflejado en la figura 3.23. En las líneas de código mostradas se realiza el desplazamiento del registro de cada variable un elemento atrás en su vector de datos correspondiente. Para los vectores de dos elementos exclusivamente es suficiente con una operación de asignación mientras que para los más extensos se requiere del empleo de un bucle for() o similar. El correcto funcionamiento de esta porción de código es fundamental para el correcto funcionamiento del sistema en tiempo discreto.

```
/* Actualizacion de valores*/

m_R_steer[1] = m_R_steer[0];
m_R_steer_f[1] = m_R_steer_f[0];
m_u[1] = m_u[0];
m_antiwindup[1] = m_antiwindup[0];
for (int i=2;i>=1;i--){
    m_yp[i] = m_yp[i-1];
    m_ep[i] = m_ep[i-1];
}
for (int i=5;i>=1;i--){
    m_up[i] = m_up[i-1];
}
m_es[1] = m_es[0];
m_Encoder[1] = m_Encoder[0];
for (int i=3;i>=1;i--){
    m_ys[i] = m_ys[i-1];
}

for (int i=4;i>=1;i--){
    m_us[i] = m_us[i-1];
}
```

Figura 3.23: Actualización de valores.



## Capítulo 4

# Throttle-by-wire

### 4.1. Introducción

El primer aspecto a tratar en este trabajo es, de los dos lazos de control a realizar, el control sobre la velocidad del vehículo. Ello presenta gran similitud con los sistemas de control de velocidad de crucero. El usuario, al trabajar en modo de control automático inserta una referencia de velocidad y el vehículo la mantiene hasta que se introduce otra instrucción que modifique las condiciones del sistema.

Antes de comenzar, se debe tener presente que el sistema real requiere operar a unos parámetros de tensión y corriente que no puede suministrar el sistema de control instalado. No obstante, este hecho no constituye un problema puesto que el controlador Curtis, descrito en la sección 3.2.1. Este controlador incluye un sistema de control de potencia que adecua la señal analógica producida en el acelerador del vehículo, ya sea físico o vía software, a una señal PWM equivalente encargada de la excitación el motor incorporado en el vehículo a las condiciones de tensión y corriente adecuadas. Esto además de evitar un conflicto en la etapa de diseño, añade un parámetro extra en el diseño no citado hasta ahora. Tal como se muestra en la figura 4.1, el controlador Curtis instalado en el vehículo permite la configuración, a través del software específico del dispositivo (*Curtis Controller 1413*), configurar el parámetro *Throttle Map* que determina la relación entre los parámetros citados anteriormente. Para este caso, puesto que se pretende realizar un control software a través de *ROS*, este parámetro se fijará en el 50 % para que la relación sea lineal y ambos parámetros sean equivalentes en todo momento. Otros parámetros que se permite configurar vía software son las tensiones mínimas y máximas del acelerador (Throttle 0%-100 %) o indicadores que en el momento que la señal los sobrepase o decaiga por debajo de ellos indiquen al usuario un error (Throt Fault Lo-Hi). Toda esta información está disponible en el manual facilitado por el fabricante [1].

El enfoque inicial de trabajo planteado para el tratamiento de este sistema comprende tres fases citadas a continuación. No obstante, a causa de ciertos imprevistos que se comentan en la sección 1, el diseño e implementación del sistema de control no se ha podido llevar a cabo.

- Modelado y caracterización del sistema de propulsión del vehículo como un sistema de “*caja negra*”.
- Diseño e implementación del controlador en el nodo *steer\_controller*.
- Realización de ensayos reales y simulados para la verificación del proceso realizado.

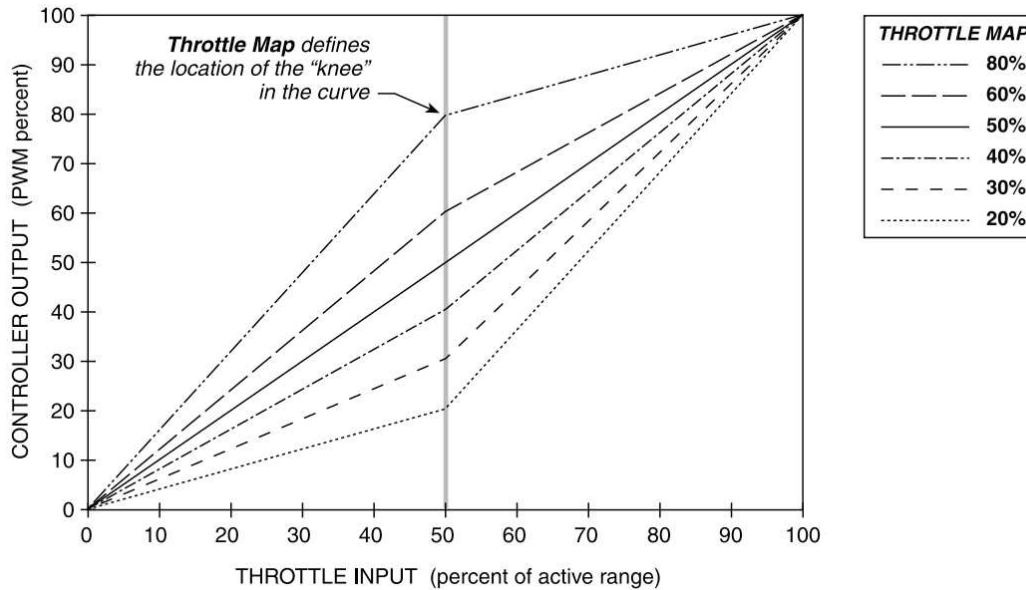


Figura 4.1: Señal PWM para el rango de señales de entrada con distintos throttle map.

## 4.2. Modelado del sistema

Para el modelado de este sistema, se ha planteado su estudio como un sistema de “caja negra”. Por esta razón, los principales datos de interés que se considerarán para la obtención del modelo que lo rige serán: como entrada la señal analógica generada por software que emula el pedal del acelerador y como salida la lectura del sensor de efecto Hall acoplado en la zona lateral del motor. Ciertamente, que un enfoque mecánico con un estudio de todos los componentes involucrados en el sistema, denominado modelado basado en primeros principios, aportaría ecuaciones que describen con mayor exactitud el comportamiento obtenido y aportan mayor riqueza de información de cara al diseño de controladores y técnicas a implementar. A pesar de ello, se considera que entrar en dicho enfoque implica una carga de trabajo propia de un TFG centrado en ella, por lo tanto, se considerarán exclusivamente para justificar los comportamientos obtenidos. Una vez determinado el enfoque al abordar el modelado es necesario definir algunas consideraciones para la comprensión de los siguientes apartados.

Para definir un sistema existen múltiples modelos distintos que ofrecen información, que otros modelos no recogen, acerca de dicho sistema. En este caso, puesto que se busca el análisis de la respuesta transitoria de un sistema con SISO (Single Input Single Output), se empleará la representación mediante *función de transferencia (f.d.t.)*. Para trabajar con esta representación hay que tener presente dos consideraciones a nivel matemático para poder aplicar la transformada de Laplace: el sistema debe ser lineal y las condiciones iniciales nulas. Una versión reducida de la definición de un sistema lineal es que la respuesta frente a una señal de entrada es equivalente en todo el rango de operación del sistema. Para un sistema mecánico como el tratado en este caso, se conoce que no se trata de un sistema lineal, pues de partida cuenta con una zona muerta de un voltio. No obstante, se considera que la simplificación a un sistema lineal entorno a un punto de operación no perjudicará de forma notoria al sistema cuando se opere en lazo cerrado siempre que se tomen medidas para compensar las no linealidades en la mayor medida posible.

El tema de las condiciones iniciales nulas se abordará con el diseño del sistema en torno a un punto de operación el cual se restará al resto de valores de respuesta para emular la nulidad de dichas condiciones. Además del fenómeno de la zona muerta descrita anteriormente, se hacen patentes otros fenómenos no lineales en el funcionamiento del sistema como es el efecto del margen dinámico. Dicho efecto se percibe en que el motor no alcanza la máxima velocidad establecida por el fabricante. Causantes de este fenómeno residen en las características constructivas del vehículo como el peso del mismo, estado de carga de las baterías o pérdidas en los mecanismos de transmisión del movimiento. Por otro lado, fenómenos como el rozamiento entre las ruedas del vehículo y el suelo, dentro del cual se engloban factores complejos de modelar como el peso del vehículo o tipo de superficie del suelo, se consideran como una perturbación del sistema que deberá corregirse con el sistema de control.

#### 4.2.1. Ensayos de velocidad de avance

Al considerar el rozamiento como una perturbación del sistema, el planteamiento inicial comprende el modelado con la realización de ensayos con el vehículo suspendido y sin rozamiento en el sistema más allá del propio del mecanismo, el cual se englobará en el modelo obtenido. Con el vehículo suspendido se han realizado ensayos en ambos sentidos de giro del motor para un modelado más preciso en posteriores etapas de trabajo.

A continuación, en las figuras 4.2 y 4.3, se muestran los resultados obtenidos de la realización de dos ensayos sobre la velocidad de giro de avance del motor. Las señales de entrada se han procurado en forma de escalón con un ventana temporal entre las mismas suficiente para que el sistema alcance para cada señal su valor en estacionario. En cada ensayo se ha procurado llevar el sistema desde su estado de reposo hasta su límite superior de funcionamiento para posteriormente devolverlo a su estado inicial. De todos los ensayos realizados para la captura de esta dinámica se considera oportuno mostrar dos de ellos en los cuales se aprecian las principales cualidades del sistema.

## 4.2. MODELADO DEL SISTEMA

---

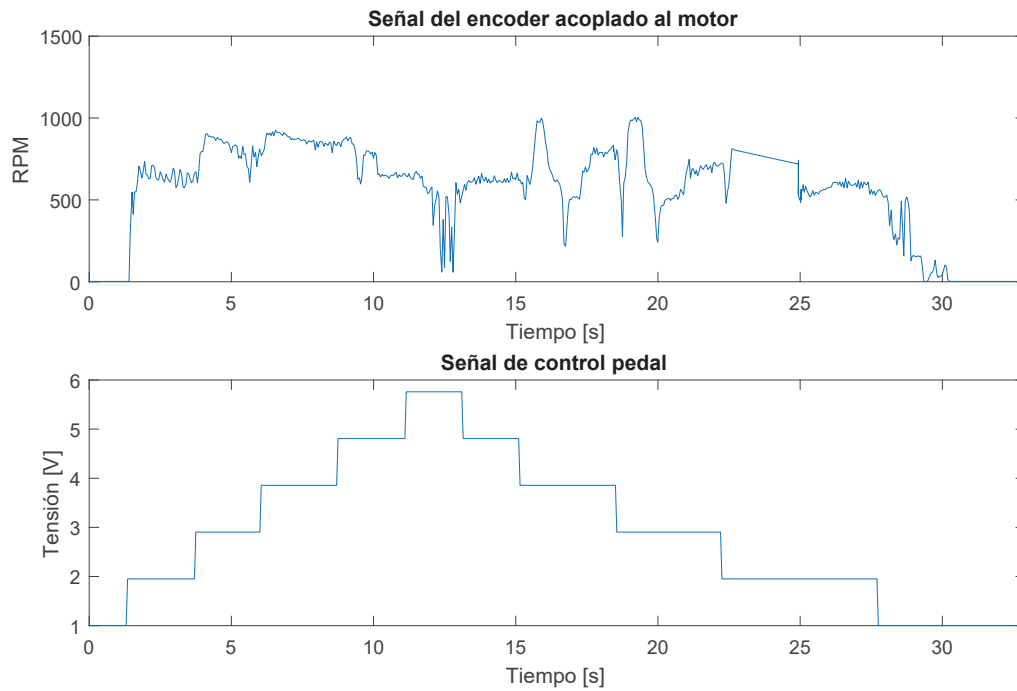


Figura 4.2: Resultados del ensayo con múltiples escalones de amplitud 0.95V.

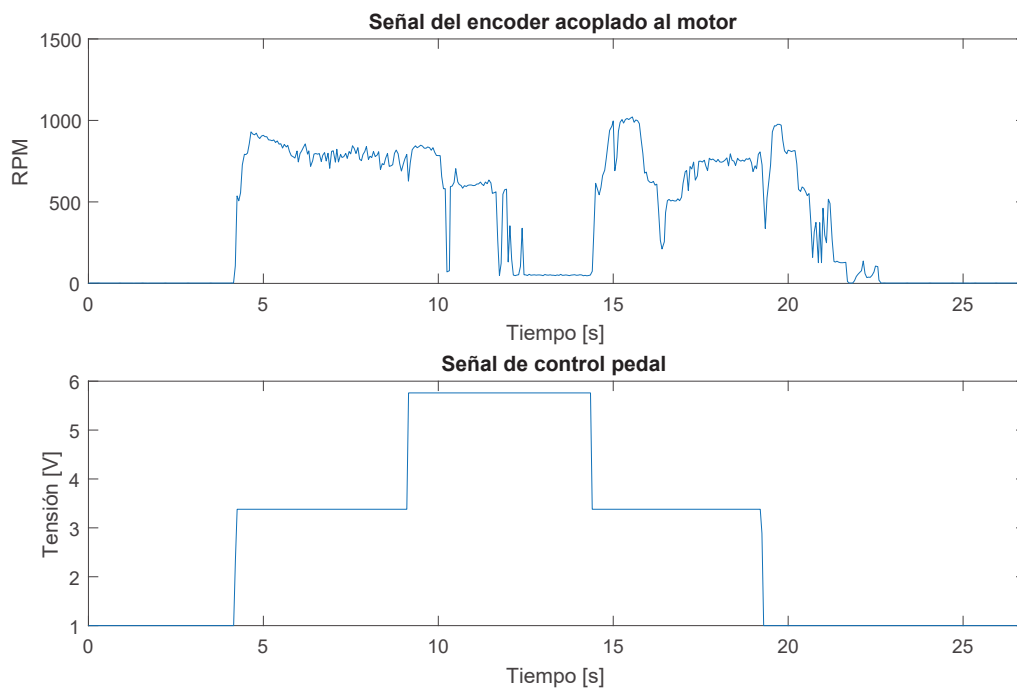


Figura 4.3: Resultados del ensayo con múltiples escalones de amplitud 2.38V.



En los ensayos mostrados en las figuras anteriores se aprecia de forma clara la complejidad que presenta el sistema para su modelado y control. En primer lugar se analiza el ensayo mostrado en la figura 4.2. Las señales de control se han insertado en forma de escalón amplitud homogénea,  $0,952V$ . Los escalones descendientes se han extendido más en el dominio temporal puesto que visualmente el sistema tardaba más en alcanzar el equilibrio. Esto se debe a que el sistema cuando deja de percibir la señal de excitación posee un nivel de inercia el cual, sin ningún accionamiento de freno, lleva al sistema a relentizar el alcance del nuevo estado estacionario. Esto crea una mayor dependencia del modelo de las características mecánicas del sistema. Analizando visualmente la respuesta transitoria de los escalones ascendentes, se aprecia una sobreoscilación lo cual es un indicador de el orden superior a uno o la presencia de ceros en la *f.d.t.*

Otra característica no lineal importante del sistema que se aprecia en todos los ensayos realizados es el salto producido al inicio del movimiento. Además se aprecia en los datos, y también se apreció de forma visual durante la realización de los mismos, que al sobrepasar el umbral de tensión en la señal de entrada de  $3,8V$  la velocidad el motor no varía frente a señales de entrada mayores y la lectura del encoder acoplado presenta un comportamiento errático. Dicho comportamiento queda patente en un descenso de la señal medida pese a que debería de aumentar entre el umbral [ $3,85V - 4,8V$ ] y un descenso hasta niveles ínfimos en el rango [ $4,8V - 5,75V$ ]. Puesto que se apreció insitu que el motor giraba correctamente a su máxima velocidad, esta anomalía en los datos se asocia a un error en la lectura del encoder en la zona de operación límite. Esta consideración es de gran importancia para la comprensión del sistema y el correcto planteamiento de un sistema de control viable.

#### 4.2.2. Ensayos de velocidad de marcha atrás

Para el modelado del sistema cuando la velocidad del vehículo es marcha atrás se sigue la dinámica de ensayo realizada en el epígrafe anterior. El vehículo con las ruedas motrices suspendidas en el aire se somete a una serie de señales de entrada en forma de escalón de distintas amplitudes a fin de capturar su dinámica con la mayor fiabilidad posible. En las figuras 4.4, 4.5 y 4.6 se muestran los resultados de dos de los ensayos realizados.

A la luz de los resultados obtenidos en los diversos ensayos se aprecia una dinámica más pobre que en el caso de velocidad de avance. En comparación a la respuesta obtenida en las figuras 4.2 y 4.3, se aprecian dos diferencias fundamentalmente. La primera de ellas es una lectura “fantasma” de forma previa al inicio del ensayo con la primera señal de control. Para descubrir la procedencia de dicho error, se debe medir con otros dispositivos simultáneamente para registrar la máxima información posible al respecto. La otra diferencia se encuentra en el valor del margen dinámico. En este caso cuando la tensión del pedal supera los  $3,38V$  se satura la señal de salida sin percibir cambios en la entrada. No obstante, a diferencia del sistema con la marcha directa, cuando se llega a la tensión límite, no se producen errores en la lectura del encoder.

## 4.2. MODELADO DEL SISTEMA

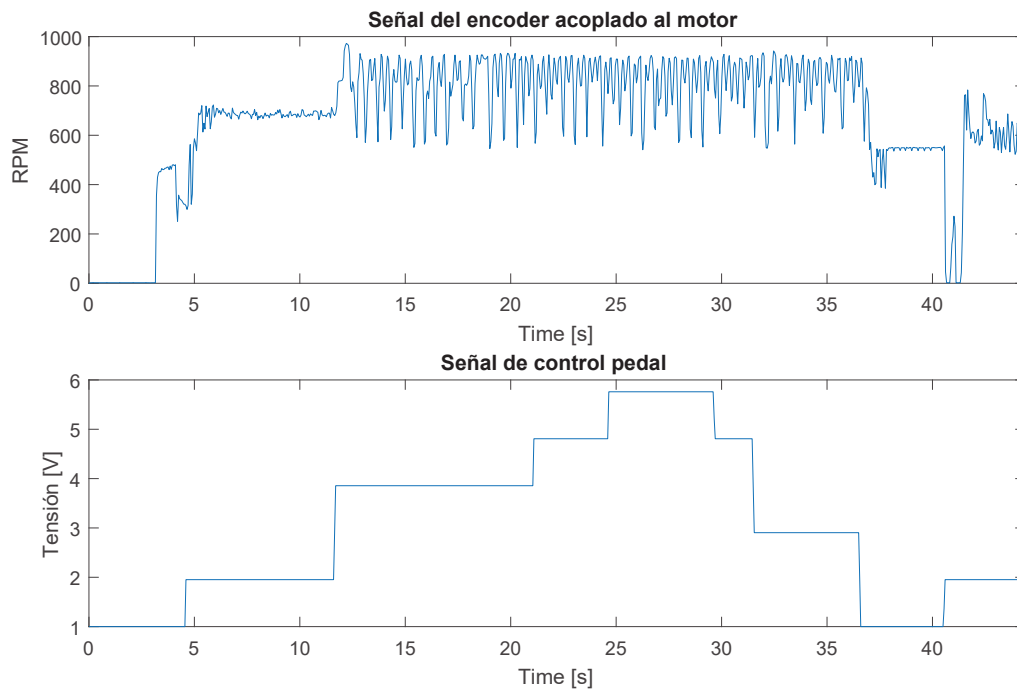


Figura 4.4: Resultados del ensayo con múltiples escalones de amplitud variable.

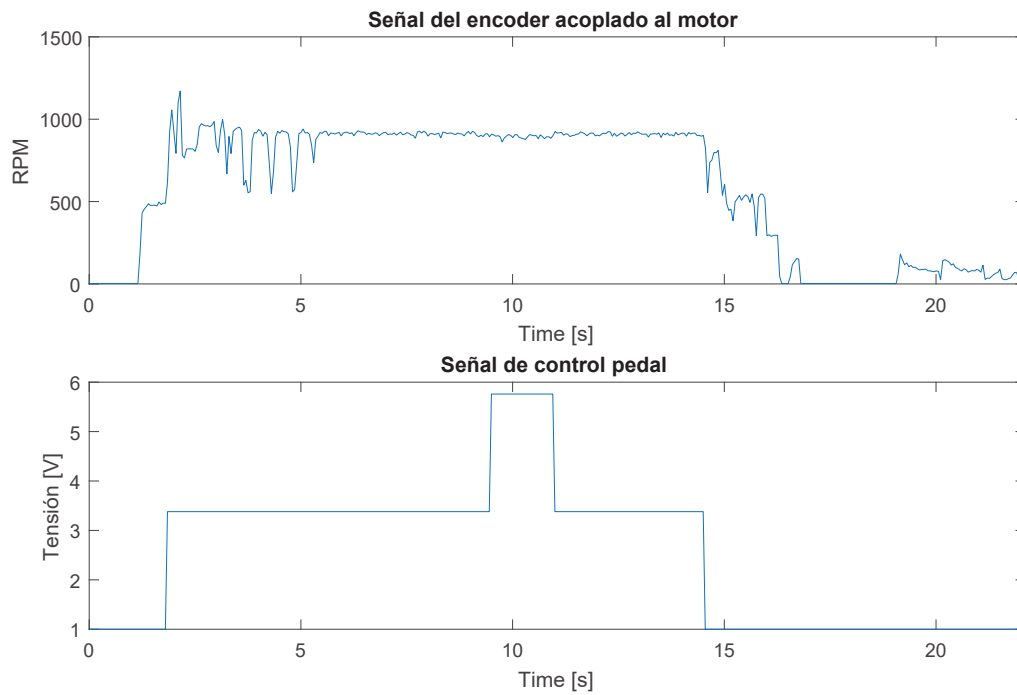


Figura 4.5: Resultados del ensayo con múltiples escalones de amplitud 2.38V marcha atrás.

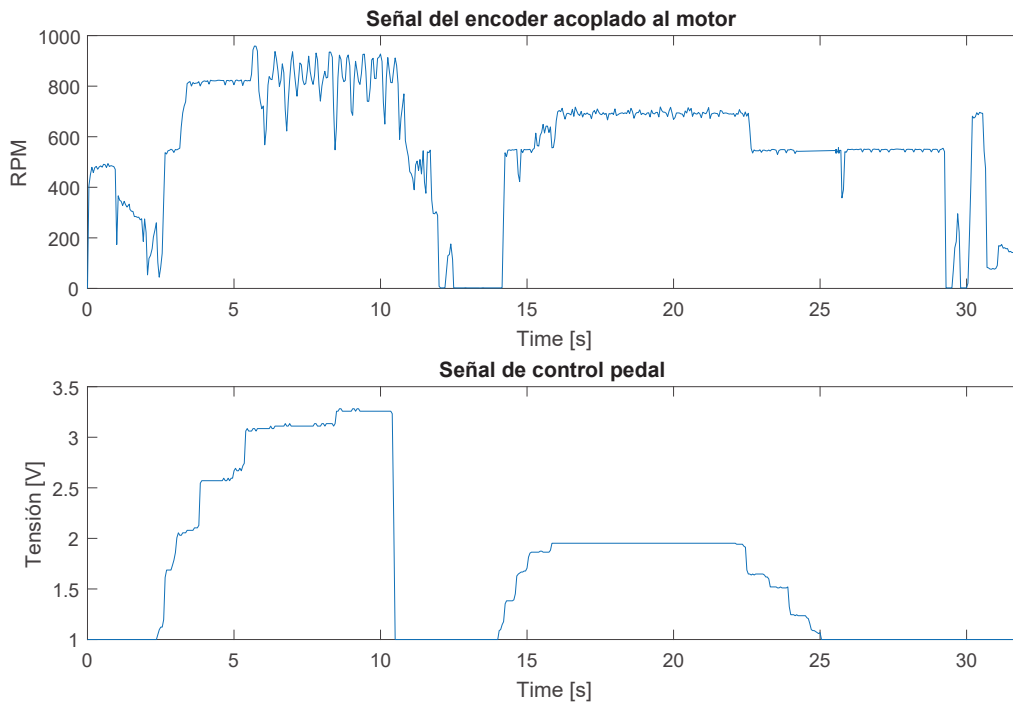


Figura 4.6: Resultados del ensayo con entrada variable.

En el ensayo representado en el figura 4.6 se ha introducido una serie de señales de entrada distintas a los ensayos anteriores con el fin de obtener más información del sistema. Tras analizar los resultados obtenidos se estima oportuno la realización de nuevos ensayos en el entorno de operación del vehículo con la esperanza de obtener una respuesta del sistema que presente un modelado más simple con el empleo de *f.d.t.* Dicho entorno es considera el movimiento del vehículo por los alrededores del edificio *CITE IV* de la Universidad de Almería.

### 4.2.3. Ensayos con el vehículo en movimiento

Los ensayos se han realizado desplazando el vehículo en tramos rectos variando la señal de entrada entre el 0 – 50%. Esta consideración se ha estimado oportuna pues la velocidad alcanzada por el vehículo en ensayos anteriores de prueba era lo suficientemente elevada como para realizar un trayecto seguro para el vehículo y el entorno en el que se desarrollaba. Otra consideración mecánica a tener presente es que los ensayos se realizaron con las baterías en pleno estado de carga y una persona abordo en todo el trayecto.

En las figuras 4.7,4.8 y 4.9 se muestran los resultados de tres del total de ensayos realizados. En ellos se aprecian nuevamente no linealidades, al igual que ocurría en los ensayos realizados con las ruedas motrices suspendidas, que repercuten en el modelado mediante *f.d.t.*

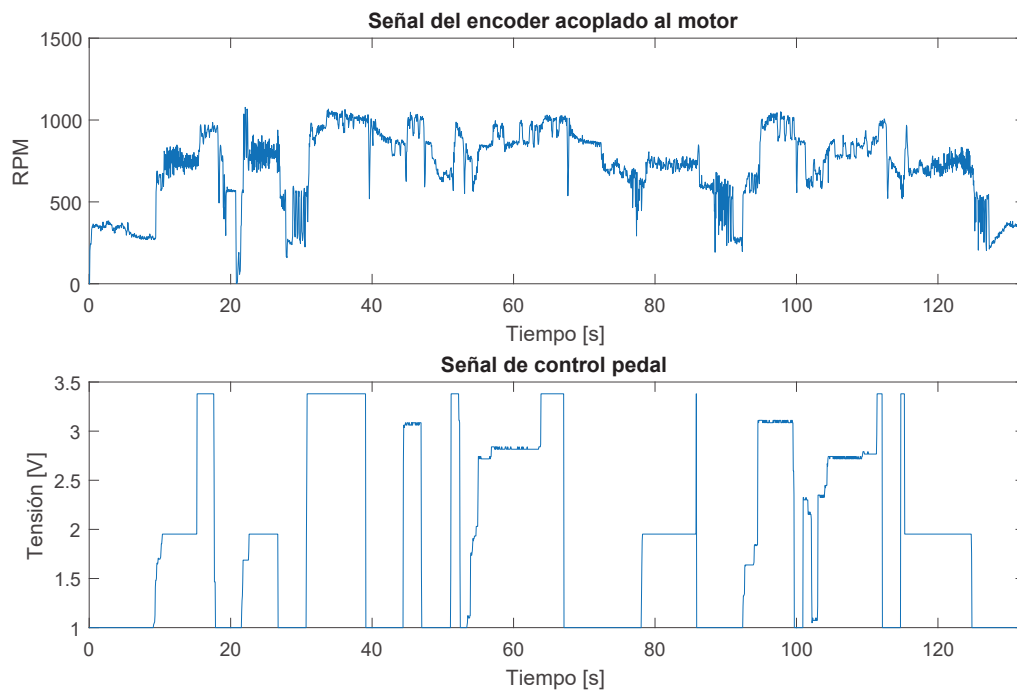


Figura 4.7: Ensayo realizado con el vehículo en movimiento (I).

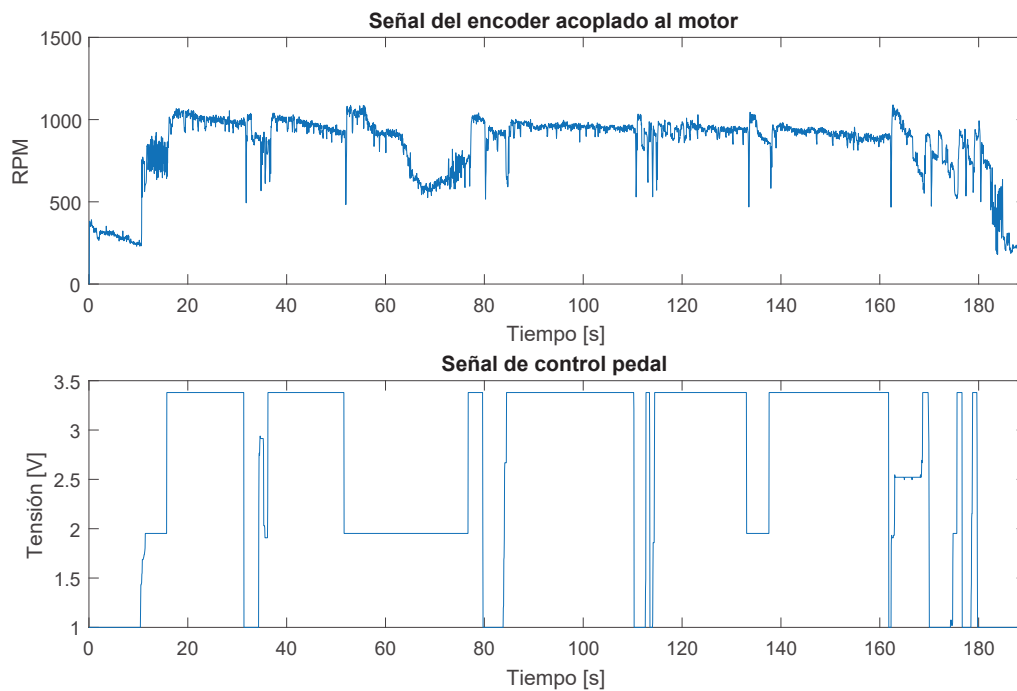


Figura 4.8: Ensayo realizado con el vehículo en movimiento (II).

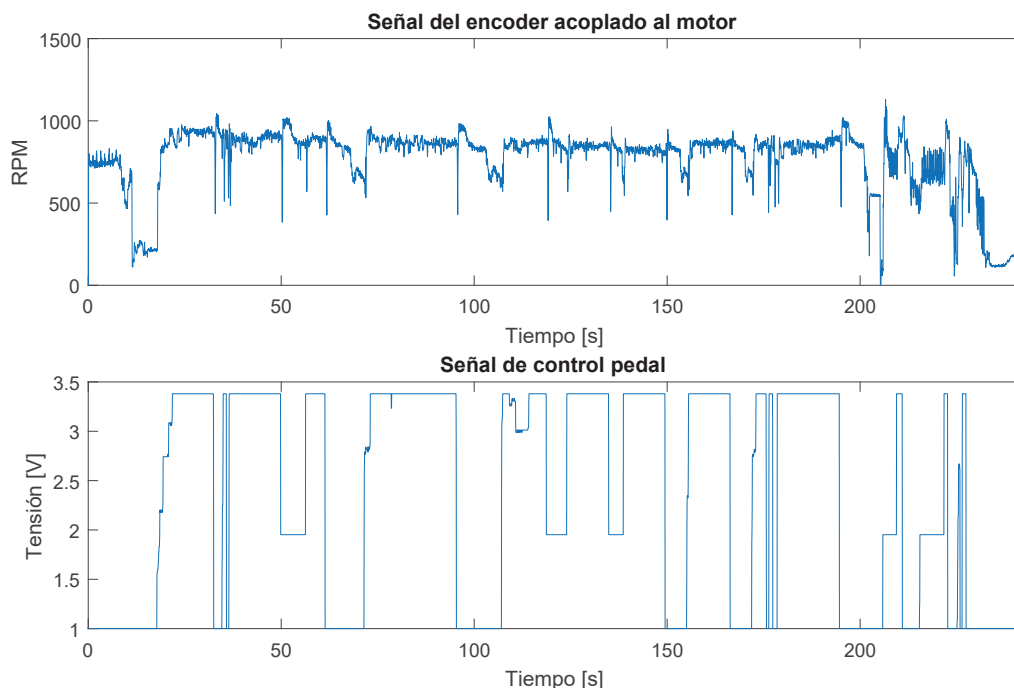


Figura 4.9: Ensayo realizado con el vehículo en movimiento (III).

Tras los resultados obtenidos en cada uno de los ensayos se considera que no se puede modelar el sistema mediante *f.d.t* de bajo orden. Por lo que en el siguiente epígrafe se detallará el proceso para realizar un modelado de mayor orden.

#### 4.2.4. Identificación de modelos.

Atendiendo a los resultados obtenidos en los distintos ensayos realizados en el epígrafe anterior, se llega a tres conclusiones principales: El sistema presenta un comportamiento no lineal tanto en vacío como en movimiento, no se puede modelar correctamente con los sensores instalados la frenada del vehículo y se debe realizar el modelado como un sistema de alto orden y emplear herramientas de identificación de sistemas para el proceso.

Para afrontar la dificultad presentada en la frenada del vehículo la solución pasa por la implementación de un actuador mecánico sobre el sistema de frenado que pueda ser controlado vía electrónica. Esto requeriría un modelado más exigente sobre el nuevo funcionamiento del sistema para el establecimiento de un correcto sistema de control. Para poder llevar a cabo de forma óptima dicho modelado también sería oportuno la instalación de nuevos sistemas de medición de la velocidad del vehículo como IMUs o GPS. Dicha redundancia, además de la comprobación de la veracidad de la medida, permitirían detectar el origen de anomalías como las lecturas “fantasma” detectadas en los ensayos marcha atrás.

Con los planteamientos establecidos anteriormente, se limitarán los ensayos para la obtención de modelos a aquellos donde no se aprecian errores en los sensores y con escalones ascendentes. De igual forma, el carácter no lineal determina que los modelos obtenidos frente a escalones más pequeños, como en el ensayo mostrado en la figura 4.2, posean más relevancia que el resto. Esta consideración se adopta porque en los niveles superiores de control del vehículo no se va a buscar grandes cambios en la señal de control, sino más bien dinámicas suaves que hagan la conducción segura y confortable. En sistemas, como el frenado electrónico, esta condición dinámica no es posible puesto que se puede necesitar grandes variaciones para evitar accidentes.

Para la identificación del sistema se emplea un modelo no lineal compuesto por un modelo lineal “Autoregresivo con entrada exógena”,  $ARX$  lineal, en paralelo con un bloque encargado de estimar las no linealidades. Este tipo de modelos,  $ARX$  lineal, operan con ecuaciones en diferencias y presentan la ecuación característica mostrada en la ecuación 4.1, donde  $\hat{y}_k$  representa la estimación de la señal de salida,  $y_{k-i}$  la señal real medida,  $u_{k-j}$  la señal de control del sistema,  $e_k$  ruido blanco de media nula y los términos  $a_i, b_j$  los parámetros estimados.

$$\hat{y}(k) = \sum a_i y(k-i) + \sum b_j u(k-j) + e(k) \quad (4.1)$$

El proceso de cálculo se realiza con el toolbox “*System Identification*”. Los modelos obtenidos para los ensayos con el coche suspendido en el aire y con el coche desplazándose por el suelo reportan grandes diferencias. En la figura 4.10, se muestra la comparación entre los distintos modelos obtenidos mediante el toolbox con un ensayo realizado en vacío mientras que en la figura 4.11 se realiza la comparación con un ensayo realizado en movimiento. Los modelos 6 y 7 representados en ambas figuras son los obtenidos del modelado de los ensayos con el vehículo en movimiento. Los modelos 1, 2 y 5 son los datos obtenidos para los ensayos con el vehículo suspendido en el aire con la marcha directa mientras que los modelos 3 y 4 son los obtenidos con el vehículo suspendido con la marcha inversa. Los resultados obtenidos reportan un efecto esperado cuando se diseñaron los distintos ensayos. La realización de ensayos en vacío obvia el efecto del rozamiento con el suelo y las implicaciones dinámicas asociadas a la estructura del vehículo, como el caso del peso propio, la distribución de cargas, etc. Estas implicaciones se reflejan en los ensayos realizados en movimiento y en los modelos obtenidos a partir de ellos.

De los modelos obtenidos, se aprecia este efecto en que los modelos 6 y 7 son los que presentan un peor ajuste a un ensayo en vacío mientras que el resto de modelos, obtenidos de este tipo de ensayos, presentan un peor ajuste en comparación con los ensayos realizados en movimiento. Esto conlleva a la adopción de una solución de compromiso. Se debe elegir entre la elección de un modelo basado en la dinámica del sistema en vacío o basado en la dinámica que presenta el vehículo en funcionamiento. En este trabajo se estima oportuno que se adopte la dinámica observada en vacío para luego considerar la diferencia medida entre el modelo y el ensayo real como una perturbación introducida por la interacción entre el vehículo y la superficie donde se produce el desplazamiento. Para el caso de los ensayos en vacío esta perturbación será nula.

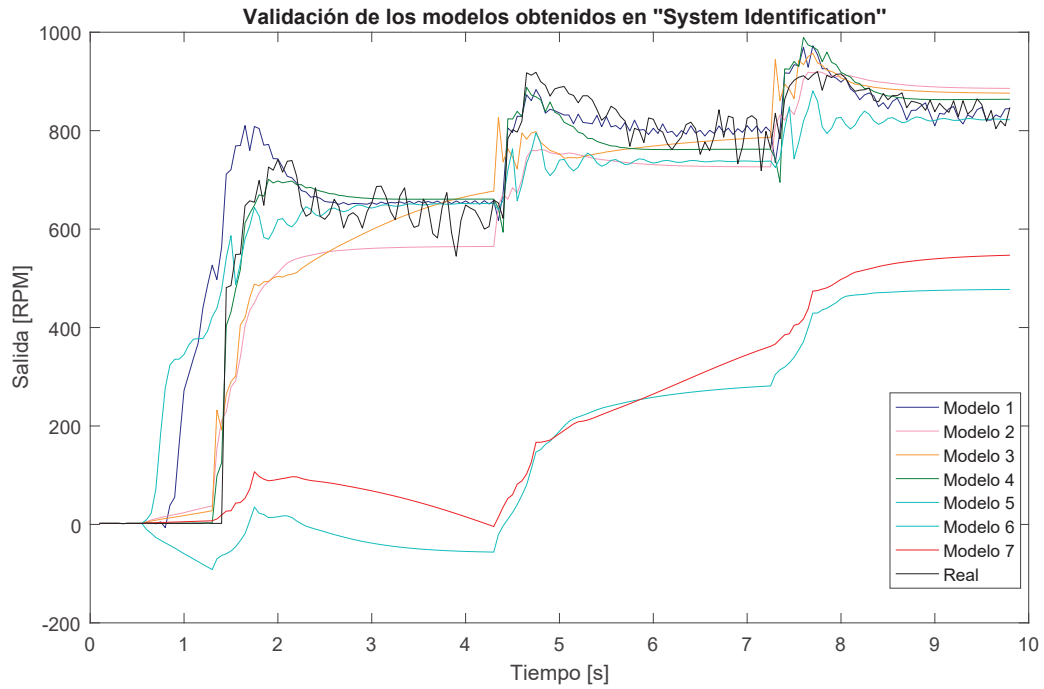


Figura 4.10: Comparación de los modelos obtenidos para un ensayo en vacío.

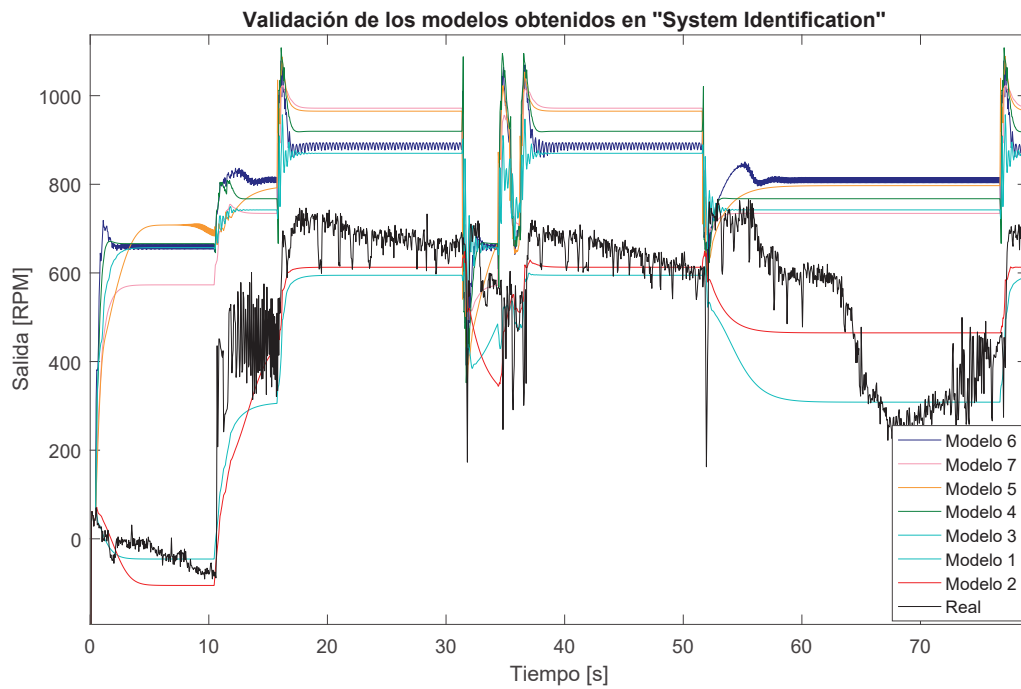


Figura 4.11: Comparación de los modelos obtenidos para un ensayo en movimiento.

## 4.2. MODELADO DEL SISTEMA

Con la elección adoptada, se tomará el modelo “uno”, de las figuras anteriores, como la representación más fidedigna de sistema real puesto que tras compararlo con todos los ensayos realizados, siempre se obtiene una similitud entre el 65-88 % en el toolbox. El modelo correspondiente a la componente lineal de la estimación queda recogido en la ecuación 4.2. Pese a la cuantía de términos empleados en la estimación, no todos son equivalentes en la estimación del sistema. Se aprecia de forma clara la dominancia de los seis términos iniciales.

$$\hat{y}_k = 204,1y_{k-1} - 59,9y_{k-2} - 24,6y_{k-3} + 6,2y_{k-4} - 4,0y_{k-5} + 12,8y_{k-6} + 5,2y_{k-7} - 0,3y_{k-8} + 6,2y_{k-9} - 5,6y_{k-10} - 12,1u_{k-1} - 16,1u_{k-2} + 3,1u_{k-3} + 8,2u_{k-4} + 11,2u_{k-5} + 4,8u_{k-6} + 14,4u_{k-7} + 4,1u_{k-8} - 0,2u_{k-9} - 8,3u_{k-10} + e_k \quad (4.2)$$

A continuación se muestra la comparativa resultante en *Simulink* de la adopción del modelo seleccionado con el sistema real. En la figura 4.12 se muestra el esquema montado para la comparación de datos. En las figuras 4.13, 4.14, 4.15 y 4.16 se muestran los resultados de simular el modelo seleccionado con los datos obtenidos en dos ensayos reales con el vehículo en vacío, tanto para la marcha directa como inversa. En dichas figuras se aprecian tres características importantes que conducen a la conclusión que el modelo estimado anteriormente es el adecuado para el estado actual del vehículo.

En primer lugar, cuando la señal de control supera un umbral, el modelo falla, estimando la salida del sistema en función de la dinámica obtenida para señales inferiores. Esto no puede considerarse un factor determinante en la validación del modelo puesto que, como ya se comentó anteriormente, se trata de un error en la lectura del sensor. La forma adecuada de comprobar la validez del sistema en ese rango, y el error de lectura del sensor, queda para un trabajo futuro en el que se implemente una lectura redundante con otro sensor de mayor precisión y con un comportamiento correcto en todo el rango de operación comprobado. En segundo lugar, se aprecia que el sistema denota poca correspondencia en la frenada libre del mecanismo. Esto como ya se estableció en la elección del modelo y rango de datos para su obtención, solo se puede mejorar mediante la implantación de un sistema que permita la actuación sobre dicho mecanismo o un modelado basado en primeros principios con un conocimiento detallado de las características del vehículo. Finalmente, lo que hace este modelo válido es un seguimiento con relativa exactitud del sistema para ambas marchas en las señales de entrada ascendentes.

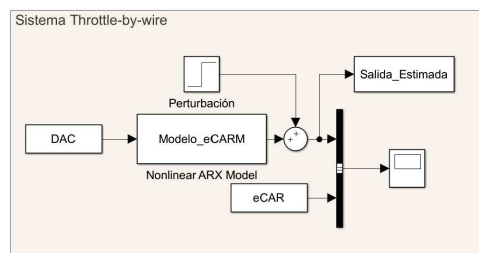


Figura 4.12: Esquema de *Simulink* para la comparación de resultados.



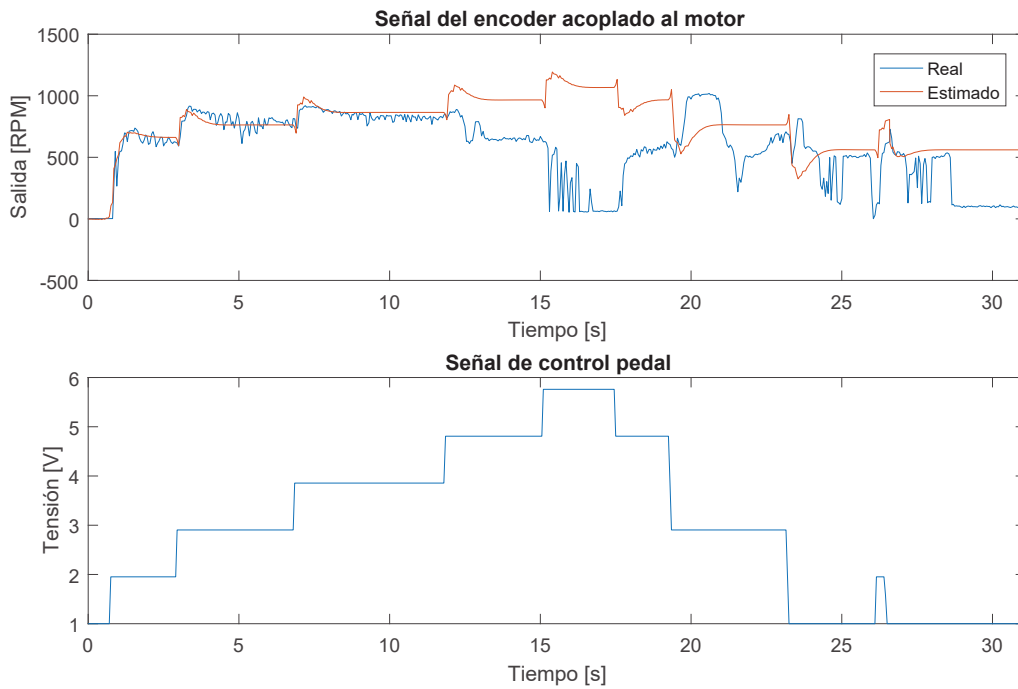


Figura 4.13: Validación del modelo obtenido en un ensayo en vacío (entradas en escalón).

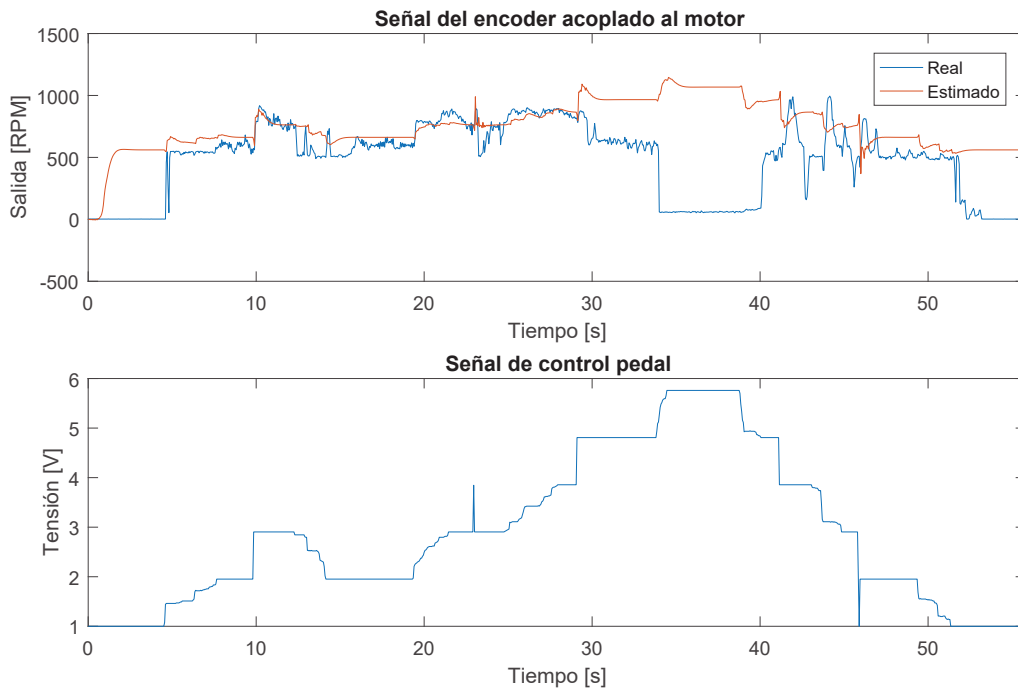


Figura 4.14: Validación del modelo obtenido en un ensayo en vacío (entradas variables).

## 4.2. MODELADO DEL SISTEMA

---

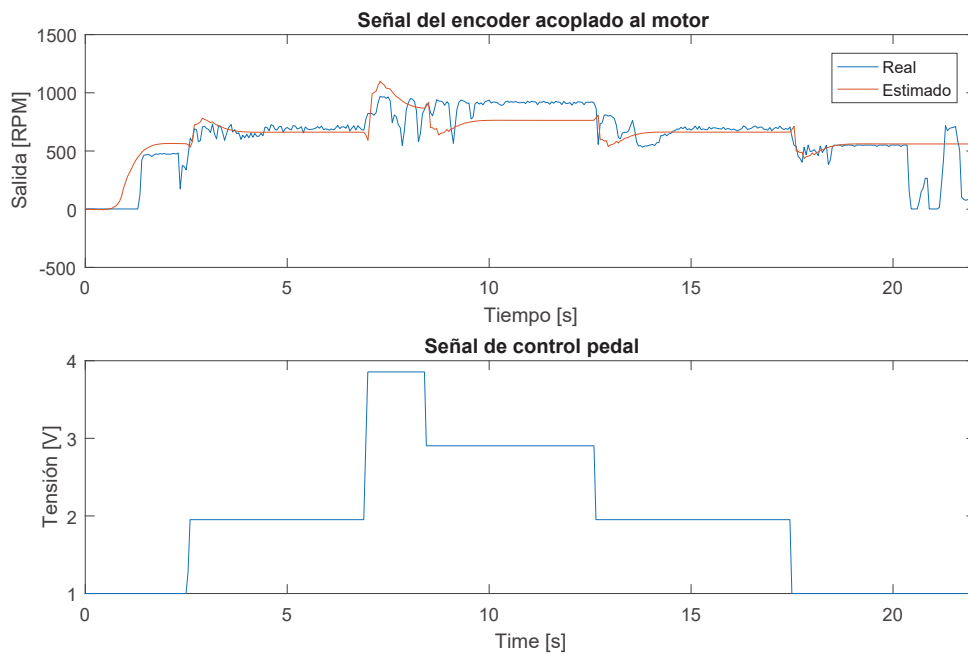


Figura 4.15: Validación del modelo obtenido en un ensayo en vacío con marcha inversa (entradas en escalón).

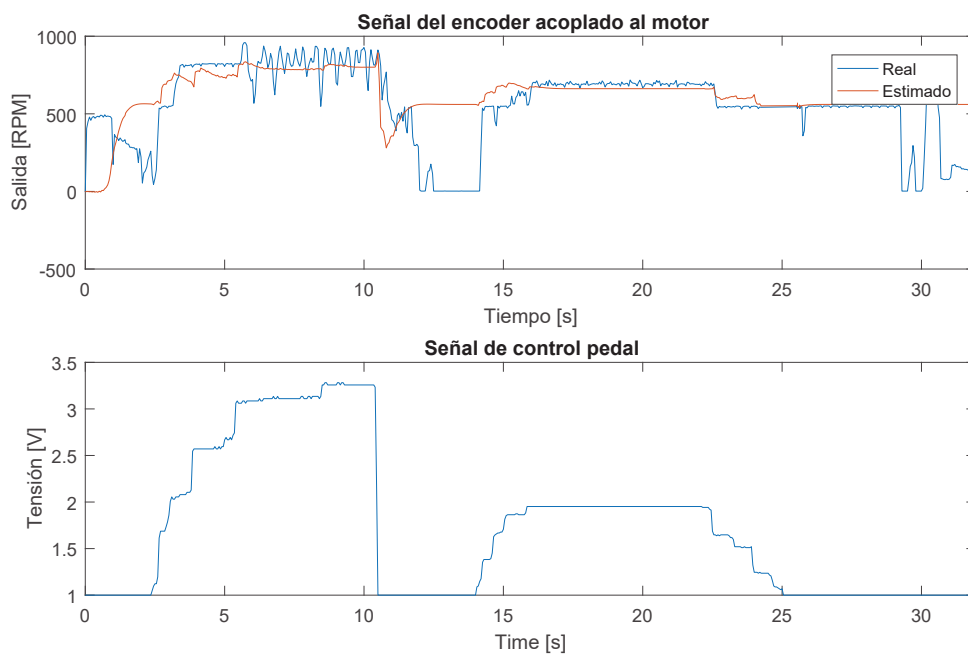


Figura 4.16: Validación del modelo obtenido en un ensayo en vacío con marcha inversa (entradas variables).

Una vez realizada la validación en los ensayos en vacío se procede a realizar el mismo proceso en los ensayos realizados en movimiento. En la figura 4.17 se muestra el resultado de la comparación del modelo con los resultados reales mientras que en la figura 4.18 se muestra el error cometido entre ambas gráficas. A la luz de los resultados obtenidos, no es tanto el error obtenido como se esperaba según la figura 4.11. Se aprecia que el sistema estimado sigue la dinámica del sistema en los escalones ascendentes, presentando las zonas de mayor disparidad cuando la señal de control es decreciente. El fenómeno de este causante es el mismo que el que producía el mismo efecto para los ensayos en vacío. Por otro lado, analizando en detalle la figura 4.18, se aprecia que las mayores discrepancias mientras el vehículo se encuentra en funcionamiento se producen al inicio y final del ensayo y puntualmente en los cambios bruscos en la señal de control. Los errores en al comienzo y finalización de los ensayos se producen, nuevamente, por errores en el sistema de lectura del encoder, puesto que el sistema en todos los ensayos, se inicia y finaliza en estado de reposo. Por otro lado, los errores producidos por cambios en la señal de control, tienen su origen lógico en la sobrecorriente que requieren los motores frente a cambios en escalón en sus señales de entrada para el alcance de su estado estacionario. Dichas sobrecorrientes repercuten momentáneamente en el campo magnético medido por el encoder de efecto Hall. Estos fenómenos no quedan completamente recogidos en el modelado del sistema, por tanto, se reflejan como errores en la validación de los modelos obtenidos.

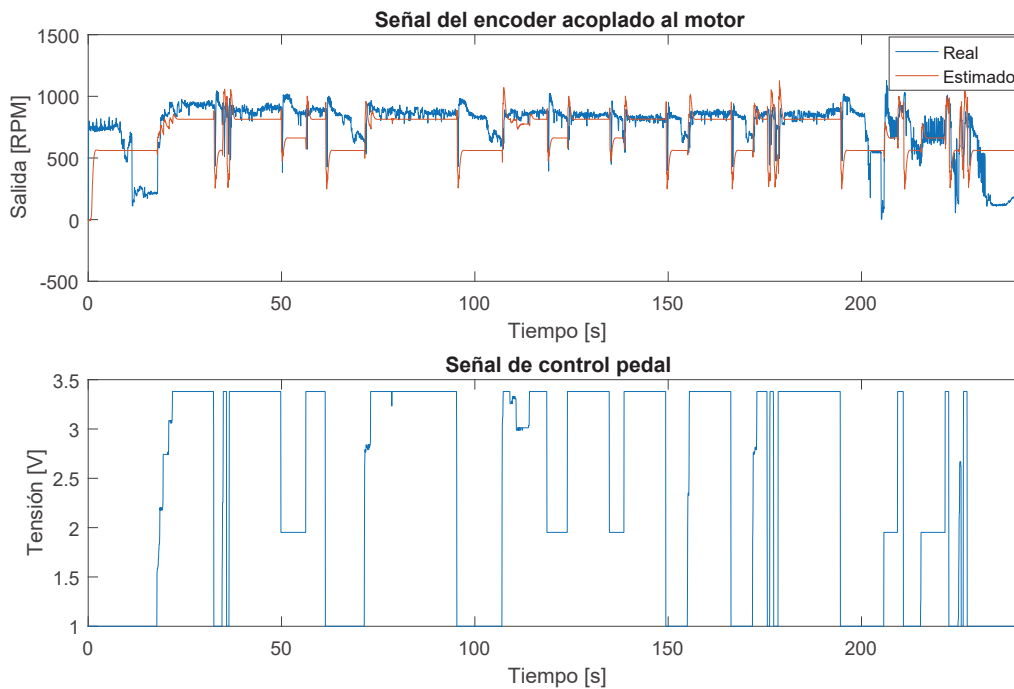


Figura 4.17: Validación del modelo obtenido en un ensayo en movimiento.

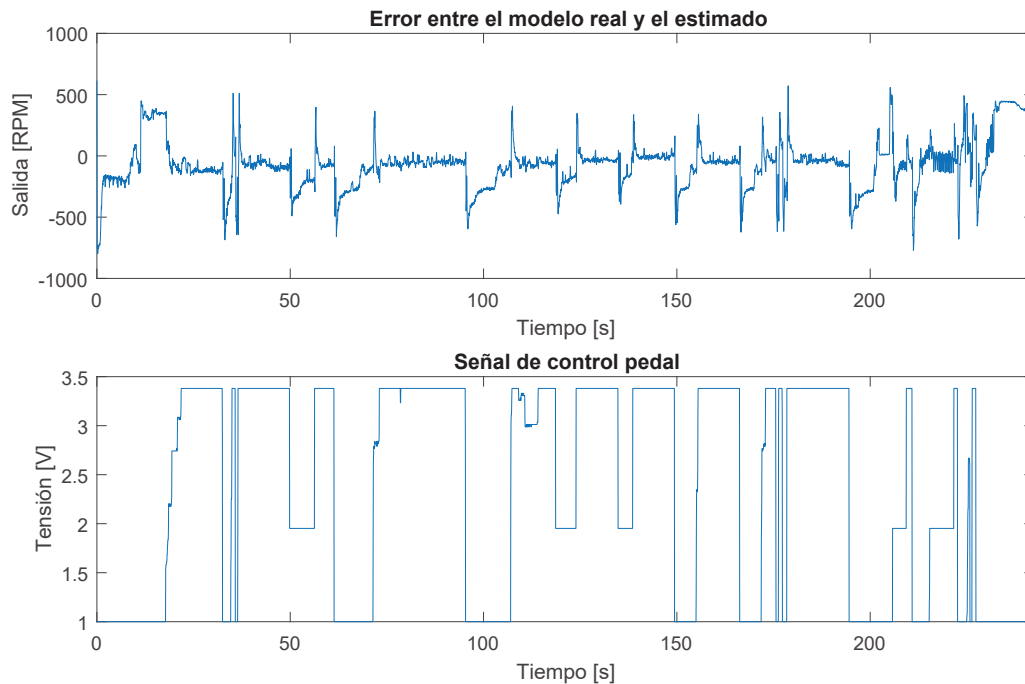


Figura 4.18: Error medido entre el sistema estimado y el real.

Con estos últimos resultados se concluye el modelado del actual sistema *Throttle-by-wire* implementado en el vehículo *eCARM*. Según los resultados obtenidos en los ensayos de validación, el modelado llevado a cabo empleando el modelo *ARX* no lineal ha determinado un sistema cuyos parámetros presentan gran similitud con el sistema real. Esto supone una base para el empleo y operación de este sistema en futuros trabajos. Analizando los términos lineales estimados, se observa que en posteriores trabajos se puede reducir la cuantía de términos pues los dominantes son los seis iniciales tanto para la entrada como para la salida. Esto favorece para la realización de controladores con menor carga computacional en el sistema de control que se implemente. Sin embargo, por los errores observados en los datos, se recomienda una mejora en el sistema de adquisición de datos del vehículo para la obtención de un modelo del sistema más preciso. Puesto que el sistema obtenido presenta un elevado grado de complejidad, se estima oportuno no continuar con el sistema de control de este sistema por la elevada carga de trabajo que supondría habiéndose realizado en el presente trabajo, de forma simultánea a este sistema, el modelado y control completo del sistema *Steer-by-wire* presentado en el próximo capítulo.

# Capítulo 5

## Steer-by-wire

### 5.1. Introducción.

Una vez tratado el Throttle-by-wire en el capítulo anterior, en este se abordará el modelado y control correspondiente a la dirección del vehículo. El correcto funcionamiento de ambos sistemas de forma individual es condición indispensable para el desarrollo de investigaciones de mayor complejidad en el vehículo, como puede ser la conducción autónoma. Como bien es sabido, en el proceso de conducción, la capacidad de giro está estrechamente supeditada a la velocidad del vehículo. Por tanto, la correcta realización de este trabajo supone una base para futuros trabajos desarrollados sobre el mismo vehículo.

No obstante, el problema del control de la dirección de un vehículo no posee exclusivamente un enfoque y punto de vista para ser abordado. En este caso se abordará empleando el método de la “Caja negra”, es decir, solo se considerarán las señales de entrada y la respuesta global del sistema, lo que ocurre en medio se aproximará a una función de transferencia para la cual se adoptarán diversas consideraciones matemáticas. Este enfoque se puede considerar la simplificación del tratamiento mecánico del problema. En él, se aborda la obtención de dicha función de transferencia atendiendo a las relaciones entre los distintos componentes que intervienen en el proceso, así como sus propiedades mecánicas y relación con la evolución dinámica y temporal del sistema. En un trabajo fin de grado realizado en el año 2014 por Jerónimo Ramos Teodoro, titulado *Modelado y control de la dirección de un vehículo eléctrico* [18], se aborda el problema de la dirección con este enfoque sobre la versión anterior del vehículo *eCARM*.

Como ocurría en el sistema Throttle-by-wire, el modelado mediante una función de transferencia requiere la adopción de las consideraciones matemáticas comentadas en la sección 4.2. Se ha realizado el modelado entorno a un punto de operación y se ha aproximado su comportamiento como lineal entorno a dicho punto. Los ensayos se han realizado con las ruedas directrices suspendidas en el aire por lo que las ecuaciones obtenidas no consideran el efecto del peso del vehículo ni su interacción con el suelo. Estos parámetros se considerarán como perturbaciones del sistema pues su valor depende de agentes externos al propio vehículo.

## 5.2. Modelado.

Para el modelado de este sistema, se han introducido una serie de escalones en la señal PWM destinada al motor *Maxon RE50* que rige la dirección y se ha medido el encoder *HEDL 5540* que lleva acoplado para medir la respuesta del mismo. Con el fin de obtener el máximo de información posible para el modelado del sistema, se somete a seis ensayos en los cuales se introducen entradas en escalón entre el 0% y el 50% y señales generadas sin valor ni estructura fija para realizar la validación de los modelos obtenidos. En la figura 5.1 se muestra el resultado de uno de los ensayos realizados.

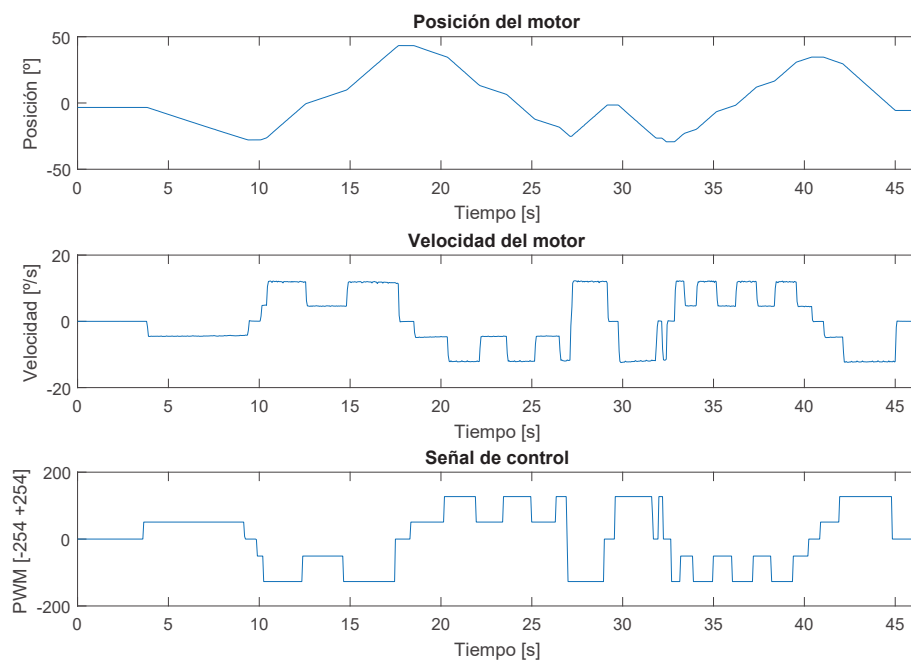


Figura 5.1: Ensayo realizado para la toma de datos.

El encoder acoplado al motor se trata de un encoder incremental, por lo que aunque su respuesta sea de posición, al no conocerse con certeza su valor absoluto, se estima oportuno obtener la velocidad del mismo derivando la señal obtenida. Puesto que la velocidad si es un factor conocido con precisión, se considera el sistema como un sistema en cascada cuyo lazo interno comprende la velocidad del motor y el externo la posición. El hecho que de la respuesta en posición no se conozca el valor absoluto no implica que sea una señal inútil. La señal de posición es bastante precisa, pero presenta la desventaja que considera como cero la posición en la que se inicia el sistema. Como solución a este problema, se instalará a posteriori un encoder absoluto que permita referenciar la dirección al iniciarse y corregir pequeñas desviaciones durante el funcionamiento.

Al ser la velocidad la respuesta del lazo interno del sistema, se considera oportuno obtener la función de transferencia que la relaciona con la señal de control en primer lugar, y considerar la posición como la adición de un integrador a dicha función. Tras la realización de los diversos ensayos establecidos, los datos se procesan en *Matlab*, más concretamente, mediante el toolbox *System Identification*, el cual permite obtener modelos a partir de señales de entrada y salida correctamente adaptadas para la herramienta empleada. Tras este proceso, se obtiene como resultado múltiples funciones de transferencia de primer y segundo orden con cierto valor de retardo. Tras un estudio detallado de cada una de las funciones obtenidas, se obtiene que las dos que reportan mayor semejanza con el sistema real son las ecuaciones 5.1 y 5.2.

$$G_v(s) = \frac{k_1}{\tau_1 s + 1} e^{-tr_1 s} = \frac{-0,0903}{0,0283s + 1} e^{-0,1497s} \quad (5.1)$$

$$G_v(s) = \frac{k_2(\beta_2 s + 1)}{s^2 + 2\xi\omega_n s + \omega_n^2} e^{-tr_2 s} = \frac{-0,557(1,5415s + 1)}{s^2 + 29,91s + 20,22} e^{-0,15s} \quad (5.2)$$

No obstante, pese a que según el toolbox empleado, la *f.d.t.* de segundo orden presenta una 96.6% de semejanza frente al 96.2% de la considerada de primer orden, se estima oportuno el uso de la segunda. Esto se debe a que el empleo en este caso de una función de mayor orden no repercute en una mejora significativa del modelado y el empleo de una aproximación correcta de primer orden, facilita el desarrollo de posteriores etapas de trabajo. Tras la determinación de la *f.d.t.* teórica a emplear, la ecuación 5.1 se ha ajustado manualmente para la obtención de una respuesta que satisfaga el mayor abanico de ensayos realizados con mayor precisión, quedando como ecuación resultante para las siguientes etapas de diseño la ecuación 5.3, donde  $k_v$  y  $\tau_v$  representan la ganancia estática y la constante de tiempo del sistema en lazo abierto respectivamente.

$$G_v(s) = \frac{k_v}{\tau_v s + 1} e^{-tr_v s} = \frac{-0,0934}{0,0283s + 1} e^{-0,15s} \quad (5.3)$$

Una vez determinada la ecuación en tiempo continuo que modela el sistema, el siguiente paso es obtener su equivalente en tiempo discreto. Esto es de vital importancia puesto que el sistema en el que se operará con este modelo es un computador con su correspondiente tiempo de muestreo. Para evitar carga computacional, los parámetros de cada elemento de control se determinarán con la herramienta *Matlab* y se introducirán de forma fija en el código de *ROS*.

En este caso, para realizar la discretización, se prescinde de los desarrollos matemáticos complejos y se opta por un desarrollo más pragmático que obtiene resultados equivalentes a los que se pueden encontrar en la literatura del ámbito de los sistemas de control en tiempo discreto [46]. Este proceso más ingenieril consiste en aplicar la transformada Z a la *f.d.t.* que modela el sistema a la cual se le añade un Mantenedor de Orden Cero (MOC), mostrado en la ecuación 5.4. A continuación se muestra dicho desarrollo.

$$MOC(s) = \frac{1 - e^{-sT}}{s} \quad (5.4)$$

$$P_v(s) = MOC(s) \cdot G_v(s) = \frac{1 - e^{-sT}}{s} \cdot \frac{-0,0934}{0,0283s + 1} e^{-0,15s} \quad (5.5)$$

$$P_v(z) = Z \cdot (MOC(s) \cdot G_v(s)) = Z \left( \frac{1 - e^{-sT}}{s} \cdot \frac{-0,0934}{0,0283s + 1} e^{-0,15s} \right) \quad (5.6)$$

La transformada  $Z$  posee ciertas propiedades que permiten la simplificación del proceso de discretización a partir de la descomposición en funciones más simples de las cuales se conoce previamente su transformada. La primera simplificación consiste en la separación de la función completa en tres miembros cuyas transformadas son conocidas o se encuentran en tablas ya realizadas.

$$P_v(z) = Z \left( 1 - e^{-sT} \right) \cdot Z \left( \frac{-0,0934}{s \cdot (0,0283s + 1)} \right) \cdot Z \left( e^{-0,15s} \right) \quad (5.7)$$

Tras esta separación de términos y conociendo que  $z = e^{st}$  y haciendo uso de la tabla 2-1 que se encuentra en el libro *Sistemas de Control en Tiempo Discreto* de Katsuhiko Ogata [47], se obtiene la ecuación 5.12.

$$Z \left( 1 - e^{-sT} \right) = 1 - z^{-1} \quad (5.8)$$

$$Z \left( \frac{a}{s \cdot (s + a)} \right) = \frac{(1 - e^{-aT}) \cdot z^{-1}}{(1 - z^{-1}) \cdot (1 - e^{-aT} z^{-1})} \quad (5.9)$$

$$P_v(z) = Z \cdot (1 - e^{sT}) \cdot k_v \left( \frac{\frac{1}{\tau_v}}{s \cdot (s + \frac{1}{\tau_v})} \right) \cdot Z \left( e^{-tr_v s} \right) \quad (5.10)$$

$$P_v(z) = (1 - z^{-1}) \cdot k_v \frac{(1 - e^{-\frac{T}{\tau_v}}) \cdot z^{-1}}{(1 - z^{-1}) \cdot (1 - e^{-\frac{T}{\tau_v} z^{-1}})} \cdot \left( z^{\frac{-tr_v}{T}} \right) \quad (5.11)$$

$$P_v(z) = k_v \frac{(1 - e^{-\frac{T}{\tau_v}}) \cdot z^{-1}}{1 - e^{-\frac{T}{\tau_v} z^{-1}}} \cdot z^{\frac{-tr_v}{T}} \quad (5.12)$$

La ecuación obtenida no solo presenta dependencia respecto del sistema modelado en cuestión. El parámetro “ $T$ ” representa el tiempo de muestreo que poseerá el controlador. En este caso adopta el valor de 50ms a causa de las limitaciones impuestas por el sistema físico sobre el que se implanta el sistema de control. La función de transferida en tiempo discreto resultante de la sustitución de los distintos parámetros por sus valores reales queda representada en la ecuación 5.13.



$$P_v(z) = \frac{-0,0782z^{-1}}{1 - 0,1713z^{-1}} \cdot z^{-3} \quad (5.13)$$

A fin de obtener un mayor grado de certeza sobre el modelado realizado, se ha realizado un modelo en Simulink en el cual una señal, grabada previamente en el Workspace de Matlab, se introduce como entrada a dos bloques que equivalen a las funciones de transferencia obtenidas tanto en tiempo discreto como continuo. Las salidas son registradas para ser comparadas posteriormente en un script de Matlab. En la figura 5.2 se muestra el resultado de establecer como señal de entrada la empleada en uno de los múltiples ensayos realizados para el modelado a modo de validación.

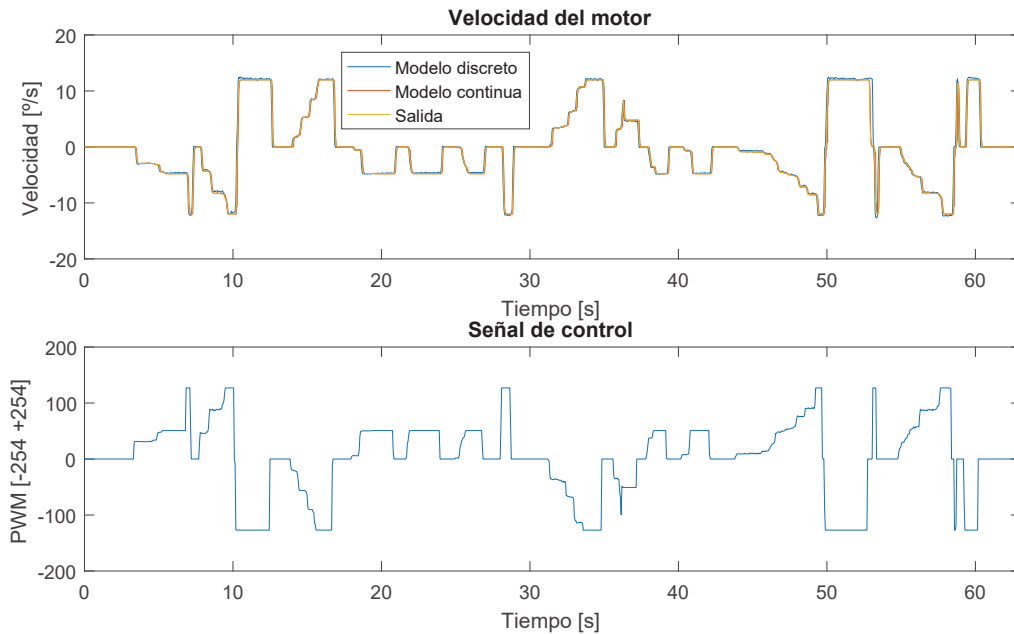


Figura 5.2: Validación del modelo obtenido sobre uno de los ensayos realizados.

En la figura anterior se aprecia la gran fiabilidad que ofrece el modelo obtenido, estableciéndose de esta forma una base sólida para el trabajo a desempeñar en la posterior etapa de diseño. Cabe resaltar que la elección de un modelo de primer orden frente a un modelo de segundo orden que ofrecía un 0.4% más de fiabilidad a resultado adecuada y el error cometido se puede considerar despreciable. Como se verá en la próxima sección, para los pequeños errores de modelado que se hayan podido cometer, se buscará compensarlos mediante el diseño de controladores robustos que permitan errores nulos en estado estacionario.

### 5.3. Diseño del sistema de control.

Como ya se ha citado anteriormente, el sistema se compone de dos lazos a controlar, el interno referido a la velocidad de giro del motor y el lazo externo referido a la posición del mecanismo de la dirección. Este tipo de sistemas requieren de una estructura de control denominada en cascada, cuya condición indispensable es la constante de tiempo del lazo interno sea como mínimo cinco veces menor que la del lazo externo.

#### 5.3.1. Lazo interno.

La ecuación 5.3 muestra la dinámica de este sistema. A continuación se analizarán ciertos parámetros de interés que ayudarán a determinar el esquema de control apropiado para este lazo de control.

En primer parámetro a analizar del sistema es el “*Tiempo de residencia medio*” ( $T_{rm}$ ). Este parámetro está asociado a la cuantificación de la velocidad de respuesta de un sistema como el tiempo que tarda la salida del sistema en reflejar un cambio en la entrada. En la siguiente ecuación se muestra el fórmula para determinar dicho parámetro.

$$T_{rm} = \frac{\int_0^{\infty} t \cdot g(x) \cdot dt}{\int_0^{\infty} g(x) \cdot dt} \quad (5.14)$$

donde  $g(x)$  es la respuesta impulsional del sistema.

A nivel práctico, operando en el dominio de la frecuencia mediante la transformada de Laplace, el cálculo del tiempo de residencia medio se obtiene como se muestra en la ecuación 5.15.

$$\left. \begin{aligned} G(s) &= \int_0^{\infty} e^{-st} \cdot g(x) \cdot dt \\ G'(s) &= \int_0^{\infty} e^{-st} \cdot t \cdot g(x) \cdot dt \end{aligned} \right\} T_{rm} = \frac{\int_0^{\infty} t \cdot g(x) \cdot dt}{\int_0^{\infty} g(x) \cdot dt} = -\frac{G'(0)}{G(0)} \quad (5.15)$$

donde  $G(s)$  equivale a la f.d.t. del sistema real y  $G'(s)$  a la derivada de dicha f.d.t.

La evaluación de este parámetro se hace en  $s = 0$  puesto que equivale a un tiempo infinito, representando así el estado estacionario del sistema. A efectos prácticos, la ecuación anterior equivale a que en un sistema que presente “x” ceros, “y” polos y tiempo de retardo, el tiempo de residencia medio equivale a la ecuación 5.16. En el caso del sistema concerniente al lazo interno, su valor se calcula en la ecuación 5.17.

$$T_{rm} = \sum_{i=1}^x \tau_i - \sum_{i=1}^y \beta_i + t_r \quad (5.16)$$

$$T_{rmv} = \tau_v + t_r = 0,0283 [s] + 0,15 [s] = 0,1783 [s] \quad (5.17)$$

La siguiente dato empleado para la caracterización de un sistema se denomina “*tiempo de retardo normalizado*” o “*índice de controlabilidad*” ( $Q_{rn}$ ). Este parámetro indica la dificultad de un sistema para ser controlado. Se obtiene como el cociente entre el tiempo de retardo del sistema y su tiempo de residencia medio, por lo que su valor está comprendido en el rango  $[0 - 1]$  y es adimensional (ecuación 5.18).

$$Q_{rnv} = \frac{t_r}{T_{rm}} = \frac{0,15}{0,1783} = 0,8413 \quad (5.18)$$

Obtener un valor de 0,8413 en este parámetro es un indicador que el sistema se trata de un proceso con **tiempo de retardo dominante**. Este hecho provoca que el sistema presente una elevada dificultad para su control por lo que se considera que la única estrategia viable para afrontar el sistema de control es el empleo del *Predictor de Smith*. Esta estrategia consiste en realimentar el sistema con la señal resultante de someter el modelo obtenido a la señal generada por el controlador e introducida al sistema real. Si el modelo reproduce con fidelidad el sistema real, la realimentación simulada introduce la señal equivalente que daría el sistema real si no tuviera retardo. Para tratar el caso de que existieran pequeñas incertidumbres en el modelado, se pueden corregir con el empleo de filtros cuya salida se restaría en la realimentación y cuya entrada se compondría por la resta de la señal del sistema real a la señal simulada tras pasarla por el tiempo de retardo modelado [9]. El esquema de dicha implementación se muestra en la figura 5.3.

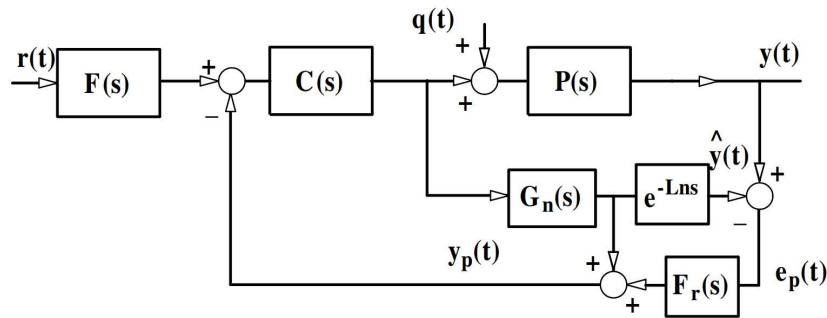


Figura 5.3: Esquema de implementación del Predictor de Smith. [9]

En la figura anterior  $P(s)$  representa el sistema,  $G_n(s)$  el modelo obtenido del mismo,  $Ln$  el tiempo de retardo obtenido durante el modelado,  $C(s)$  el controlador implantado,  $F_r(s)$  el filtro mencionado anteriormente para corregir las incertidumbres del modelado y  $F(s)$  un filtro para la señal de referencia. En el caso del vehículo eCARM, puesto que el modelado se considera muy preciso, no se requerirá del filtro en la realimentación. En la figura 5.4 se muestra el esquema de control propuesto en *Simulink* en tiempo discreto para el lazo interno. Dicho esquema incluye el mecanismo anti-windup para evitar la acumulación de error si se produce saturación en la señal de control.

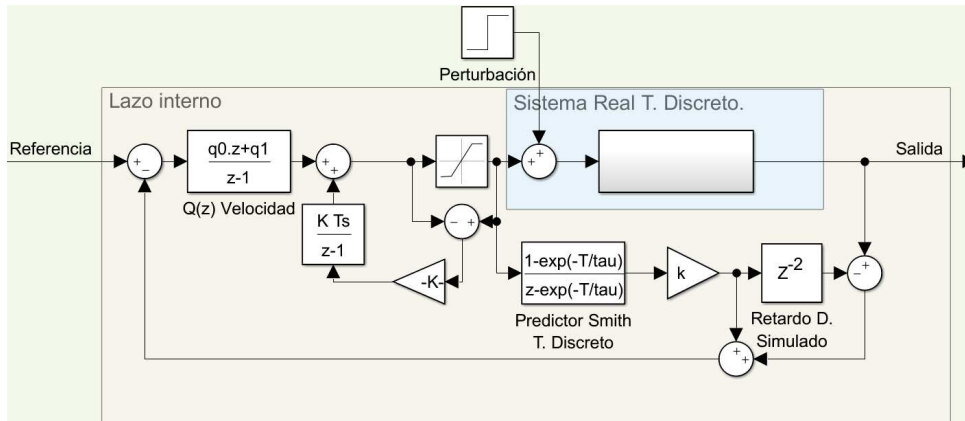


Figura 5.4: Control implementado en simulación en el lazo interno.

En el mecanismo anti-windup, se persigue evitar la acumulación de error en el controlador frente a saturaciones en la señal de control. Por ello se mide la señal de control calculada y la que se introduce en el sistema, si existe diferencia, se pasa por un término  $\frac{1}{\sqrt{T_i}}$  y un integrador. De esta manera queda compensado el efecto de la saturación en el controlador. La respuesta de dicha y al hacerse pasar por un integrador Una vez el problema del retardo ha sido solventado, la dificultad del sistema a controlar queda reducida a un sistema de primer orden. El método de sintonía para el controlador PID que se ha escogido es el método analítico de cancelación polo cero para un controlador PI. A continuación, en las ecuaciones 5.19 y 5.20, se recoge el proceso seguido para la determinación de los valores del controlador.

$$L_v(s) = \frac{K_{pv}(T_{iv}s + 1)}{T_{iv}s} \frac{k_v}{\tau_v s + 1} \rightarrow \boxed{T_{iv} = \tau_v} \rightarrow L_v(s) = \frac{K_{pv}k_v}{sT_{iv}} \quad (5.19)$$

$$G_{bcv}(s) = \frac{L_v(s)}{1 + L_v(s)} = \frac{1}{1 + \frac{K_{pv}k_v}{sT_{iv}}} \rightarrow \tau_{bcv} = \frac{T_{iv}}{K_{pv}k_v} \rightarrow \boxed{K_{pv} = \frac{T_{iv}}{k_v\tau_{bcv}}} \quad (5.20)$$

donde  $K_{pv}$  y  $T_{iv}$  representan los términos proporcional e integral del controlador respectivamente.

Una vez determinada la dependencia de los parámetros del controlador de los valores propios del sistema en lazo abierto, se fijan las especificaciones que se imponen en lazo cerrado. La única especificación que se debe imponer en este caso es la constante de tiempo en lazo cerrado,  $\tau_{bcv}$ . Esto se debe a que el controlador es tipo PI, el cual introduce dos grados de libertad y uno de ellos se elimina al imponer la cancelación polo-cero. Con el fin de evitar la dominancia del retardo en el sistema en bucle cerrado, se fija la constante de tiempo de dicho sistema como 0.2s. Con este último dato, los parámetros del controlador en tiempo continuo quedan como  $T_{iv} = 0,0283s$  y  $K_{pv} = -1,5005$ . Para operar en tiempo discreto se emplea la aproximación trapezoidal (ecuación 5.21) en la discretización de los parámetros del controlador. La estructura del controlador discreto equivalente a un PI en continuo se refleja en la ecuación 5.22 y el cálculo del controlador definitivo en la ecuación 5.23.

$$s \approx \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (5.21)$$

$$C_v(z) = \frac{q_0 + q_1 z^{-1}}{1 - z^{-1}} \quad (5.22)$$

$$\left. \begin{aligned} q_0 &= K_{pv} \left( 1 + \frac{T}{2T_{iv}} \right) \\ q_1 &= K_{pv} \left( -1 + \frac{T}{2T_{iv}} \right) \end{aligned} \right\} C_v(z) = \frac{q_0 + q_1 z^{-1}}{1 - z^{-1}} = \frac{-2,8261 + 0,175 z^{-1}}{1 - z^{-1}} \quad (5.23)$$

El resultado de dicho controlador se puede apreciar en la figura 5.5. En ella se muestra la simulación del esquema implementado en *Simulink*, figura 5.4, frente a una referencia en escalón de valor unitario.

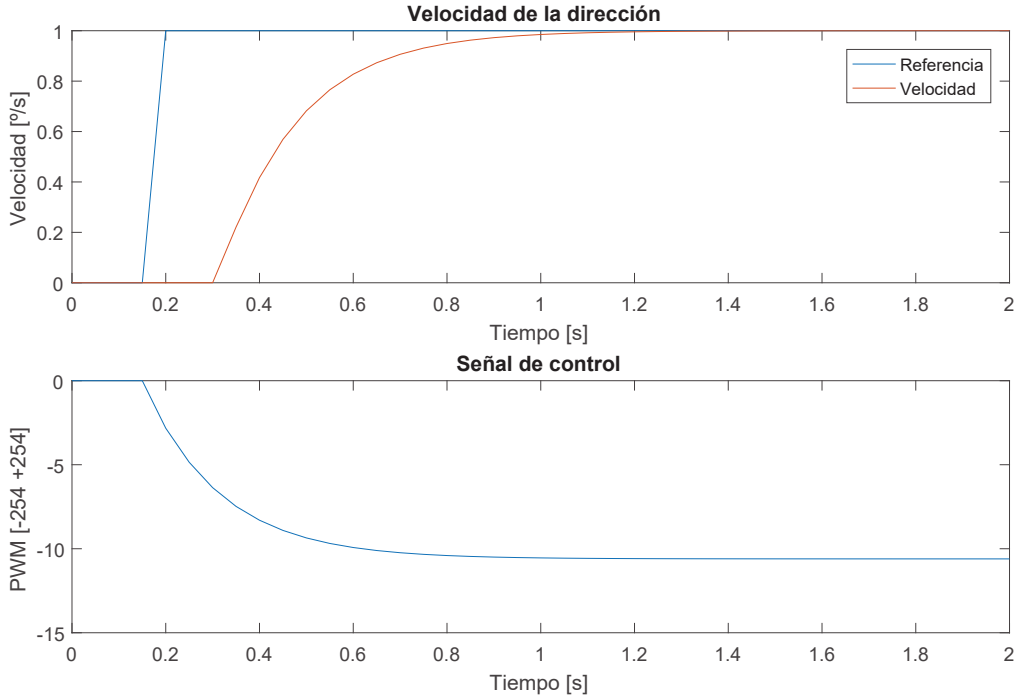


Figura 5.5: Respuesta del lazo interno en bucle cerrado.

Con los resultados obtenidos se puede concluir que el lazo interno del sistema en cascada se ha controlado de forma exitosa. No obstante, con el fin de obtener más información acerca del comportamiento del sistema, se realizará un nuevo ensayo introduciendo una perturbación de amplitud unitaria para visualizar como rechazaría el controlador implementado semejante situación. Los resultados reportados por dicha simulación se muestran en la figura 5.6. Dichos resultados son buenos ya que el sistema rechaza de forma eficaz y en un tiempo aceptable las perturbaciones introducidas.

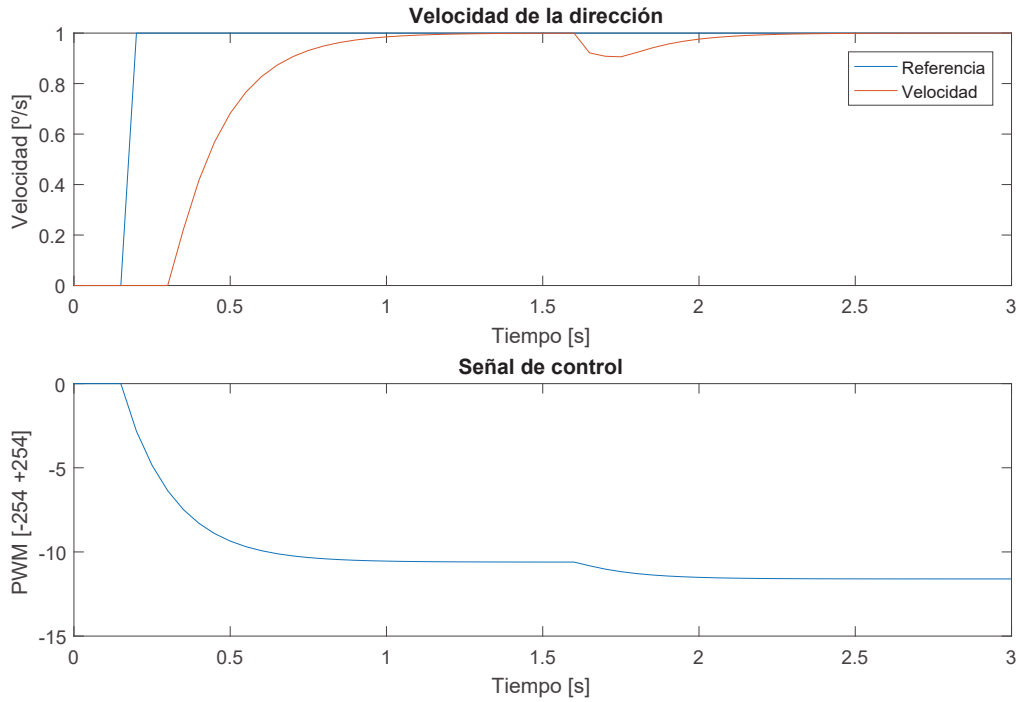


Figura 5.6: Respuesta del lazo interno en bucle cerrado frente a perturbaciones.

Con estos últimos datos, se concluye el control del lazo interno del sistema obteniendo una f.d.t. en lazo cerrado de primer orden con retardo. Su representación en tiempo continuo,  $G_{bcv}(s)$ , se muestra en la ecuación 5.24 mientras que en tiempo discreto,  $P_{bcv}(z)$ , se muestra en la ecuación 5.25.

$$G_{bcv}(s) = \frac{1}{\tau_{bcv}s + 1} e^{-tr_v s} = \frac{1}{0,2s + 1} e^{-0,15s} \quad (5.24)$$

$$P_{bcv}(z) = \frac{0,221z^{-1}}{1 - 0,779z^{-1}} \cdot z^{-3} \quad (5.25)$$

### 5.3.2. Lazo externo.

Una vez determinado el lazo de control interno, el sistema a controlar por parte del lazo externo se comprende como la adición de un integrador a la ecuación 5.24, obteniendo de resultado la ecuación 5.26. En este caso el índice de controlabilidad es 0 al poseer un integrador ya que este introduce una constante de tiempo infinita, como demuestra la ecuación 5.27, por tanto no encontramos frente a un **sistema con constante de tiempo dominante**.

$$G_{bap}(s) = \frac{1}{s(\tau_{bcv}s + 1)} e^{-tr_v s} = \frac{1}{s(0,2s + 1)} e^{-0,15s} \quad (5.26)$$

$$Q_{rn} = \frac{t_r}{T_{rm}} = \frac{0,15}{0,15 + 0,2 + \infty} = 0 \quad (5.27)$$

Para el diseño del sistema de control de este nuevo lazo se acude a la literatura. El método empleado en este caso es una modificación del método de sintonía IMC de *Rivera, Morari y Skogestad* [48], llevada a cabo por el propio *Skogestad* en 2004. A esta modificación la denominó SIMC. A continuación, en la ecuación 5.28, se muestra el valor de cada parámetro del controlador según este método de sintonía en tiempo continuo, y en la ecuación 5.29 su equivalente en tiempo discreto aplicando las mismas aproximaciones en la discretización que para el lazo interno.

$$\left. \begin{array}{l} K_p = \frac{1}{k} \cdot \frac{1}{\tau_c + \theta} = 2,222 \\ T_i = 4(\tau_c + \theta) = 1,8 \\ T_d = \tau_{bcv} = 0,2 \end{array} \right\} C_p(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) = 2,222 \left( 1 + \frac{1}{1,8s} + 0,2s \right) \quad (5.28)$$

$$\left. \begin{array}{l} q_0 = K_p \left( 1 + \frac{T}{2T_i} + \frac{T_d}{T} \right) \\ q_1 = K_p \left( -1 + \frac{T}{2T_i} + \frac{T_d}{T} \right) \\ q_2 = K_p \left( \frac{T_d}{T} \right) \end{array} \right\} C_p(z) = \frac{q_0 + q_1 z^{-1}}{1 - z^{-1}} = \frac{11,142 - 19,969z^{-1} + 8,888z^{-2}}{1 - z^{-1}} \quad (5.29)$$

El esquema montado en *Simulink* para la valoración del controlador de forma previa a su implementación en el sistema real se muestra en la figura 5.7. Al igual que se realizaba en el lazo interno, se considera oportuno la implementación del mecanismo anti-windup. Esto es así porque aunque se trata de una señal software sin efecto directo sobre un actuador, es la referencia de velocidad del motor del sistema. Esto quiere decir que tiene un nivel de saturación como se detalló en la sección 3.3.2. En las figuras 5.8 y 5.9 se ve el resultado del control propuesto en las ecuaciones anteriores.

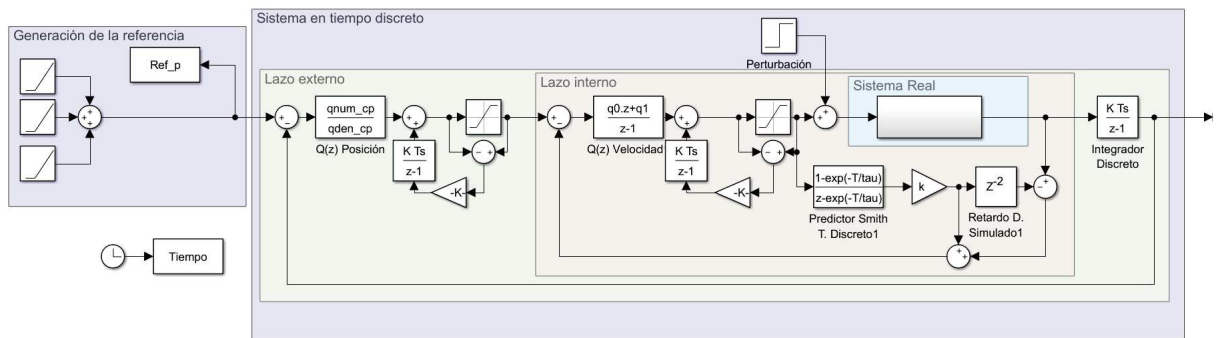


Figura 5.7: Ensayo simulado de los controladores diseñados.

### 5.3. DISEÑO DEL SISTEMA DE CONTROL.

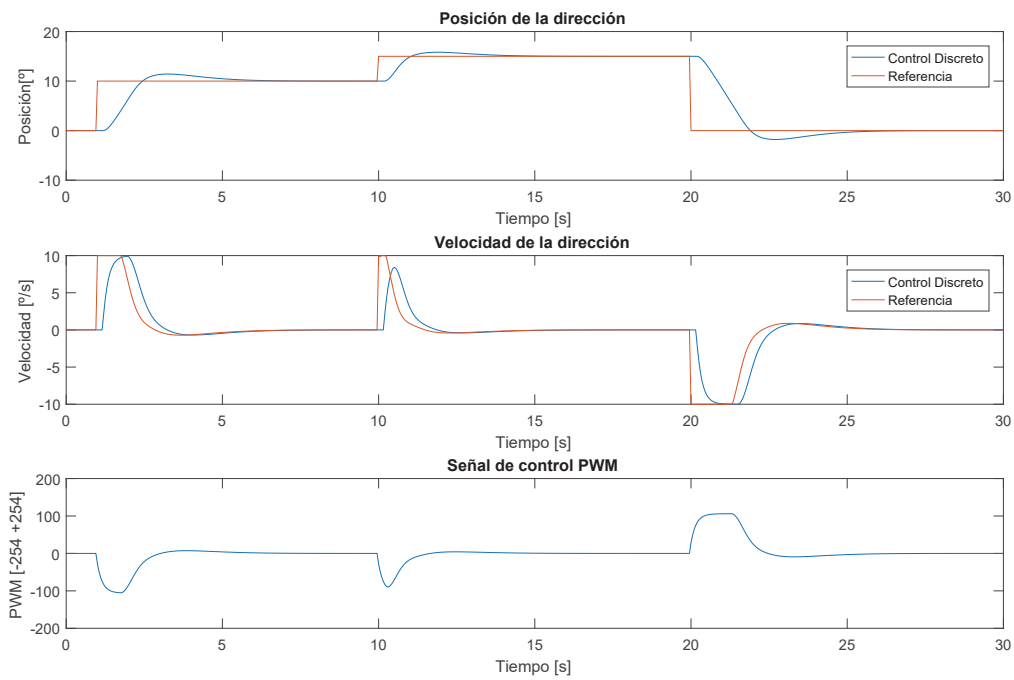


Figura 5.8: Resultados en simulación frente a entradas en escalón.

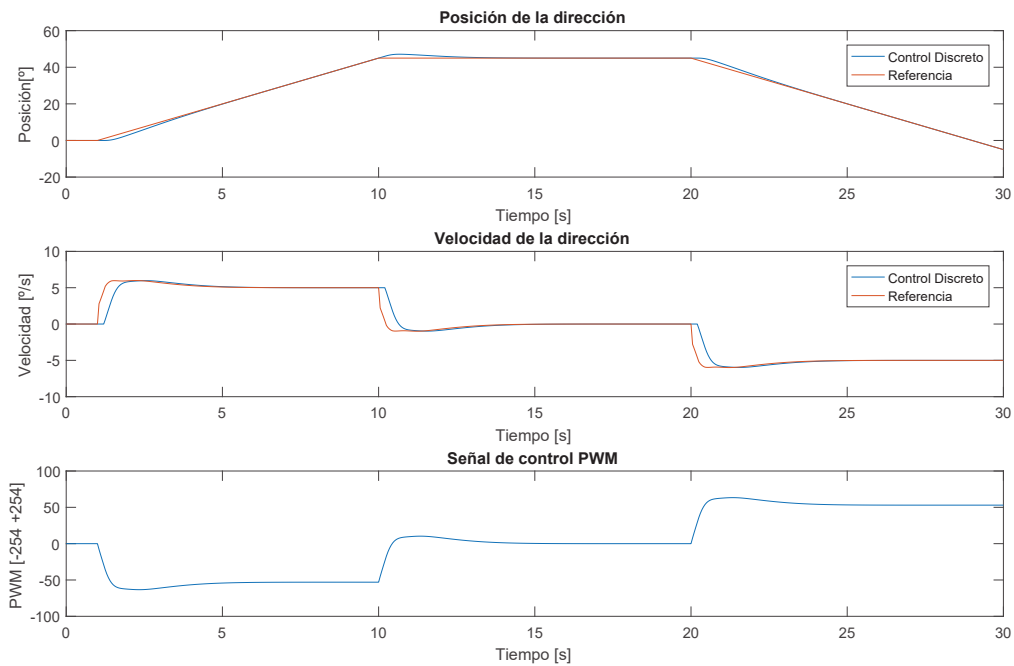


Figura 5.9: Resultados en simulación frente a entradas en rampa.



Con los resultados obtenidos, se puede considerar que el sistema de control funciona correctamente y que el método de sintonía planteado por *Skogestad*. No obstante, a fin de perfeccionar lo máximo posible la respuesta del sistema, a continuación se implementará un filtro en la referencia para eliminar la sobreoscilación observada (figura 5.10). Aunque cualitativamente no se aprecia que la sobreoscilación sea muy relevante, no se debe olvidar que el sistema se implementará en un vehículo real, por tanto, una sobreoscilación despreciable en el diseño puede desembocar en subir el vehículo en una acera o golpear algún objeto. Los resultados de la implementación de este filtro se muestran en las figuras 5.11 y 5.12.

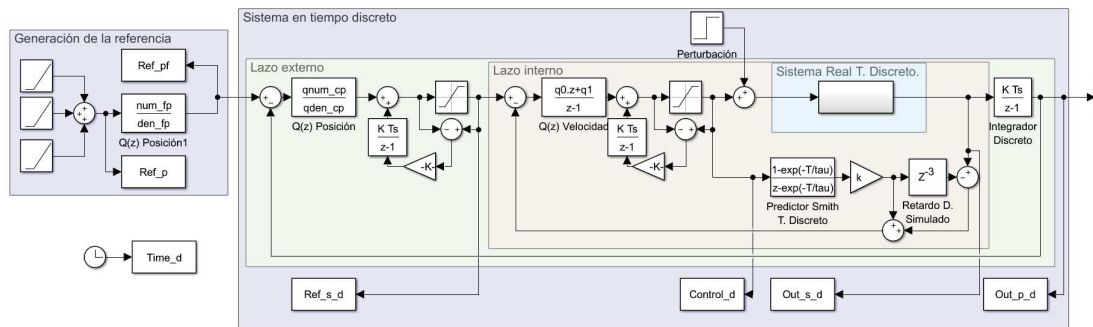


Figura 5.10: Ensayo simulado de los controladores diseñados con filtro en la referencia.

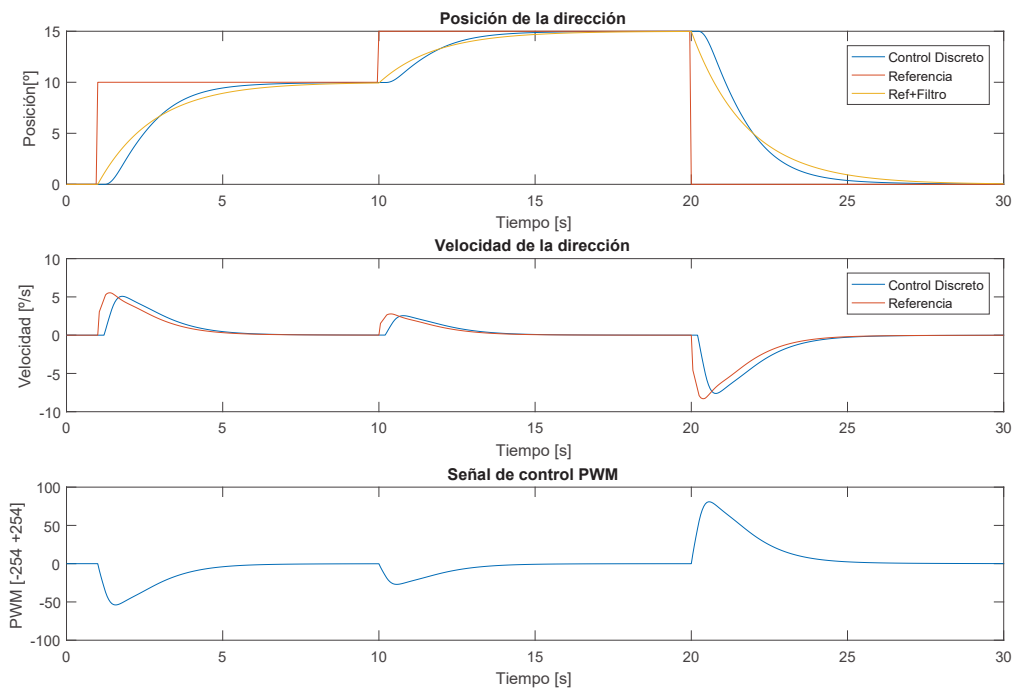


Figura 5.11: Resultados en simulación frente a entradas en escalón con filtro en la referencia.

### 5.3. DISEÑO DEL SISTEMA DE CONTROL.

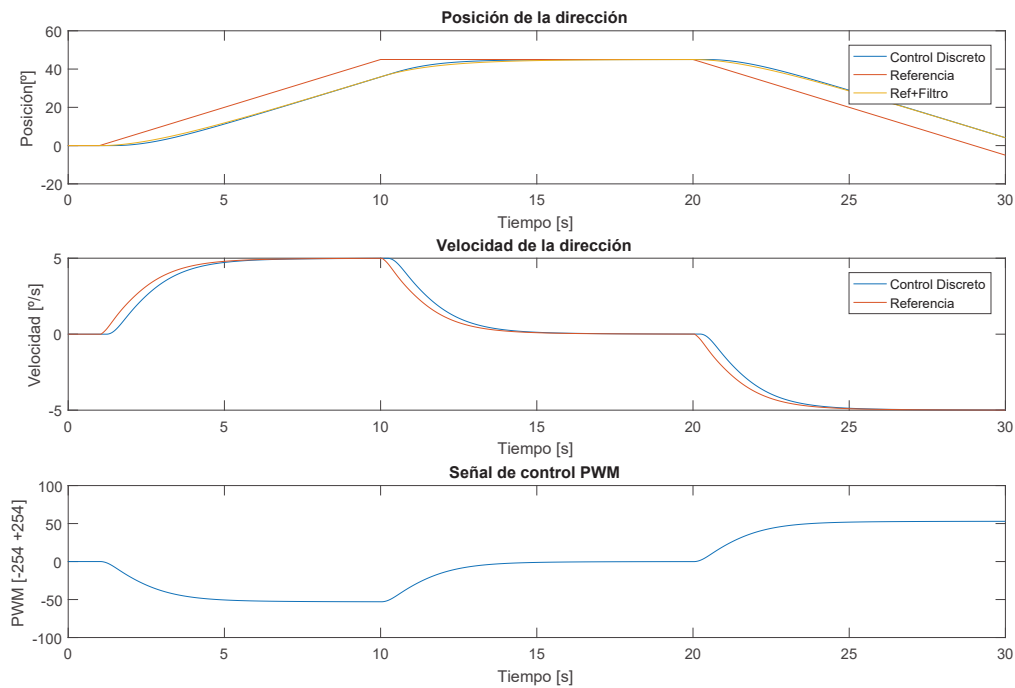


Figura 5.12: Resultados en simulación frente a entradas en rampa con filtro en la referencia.

El filtro impuesto en la referencia está calculado para eliminar el efecto del término integrador, ya que es el cero dominante frente al que introduce el término derivativo. No obstante, el resultado obtenido no es del todo satisfactorio en el seguimiento a rampas, pues presenta un error medido de 0.45, lo que en algunos casos puede ser intolerable. Esta situación plantea el deber adoptar una solución de compromiso. Se debe elegir entre un control con una ligera sobreoscilación y otro con un error notorio frente al seguimiento de referencias en forma de rampa.

La solución adoptada en este caso será la de considerar la sobreoscilación del sistema sin filtro como asumible frente al error de seguimiento a rampa. Esto es así porque se considera más importante el seguimiento a referencias para posteriores trabajos que una señal más conservadora. No obstante, puesto que la implementación del código relativo a un filtro en la referencia supone dos líneas de código extra, se dejará comentado en el código base por si se estima oportuno, en el futuro, su empleo en el sistema real.

Con el fin de obtener mayor información sobre la respuesta del sistema, se introducirá una perturbación en forma de escalón de amplitud 50. Se observarán los resultados de la respuesta frente a referencias en forma de rampa (figura 5.13). Los resultados obtenidos son muy positivos pues aunque en velocidad se percibe la perturbación, en posición apenas llega a notarse, siendo esto indicador del buen comportamiento que presenta el sistema en lazo cerrado.

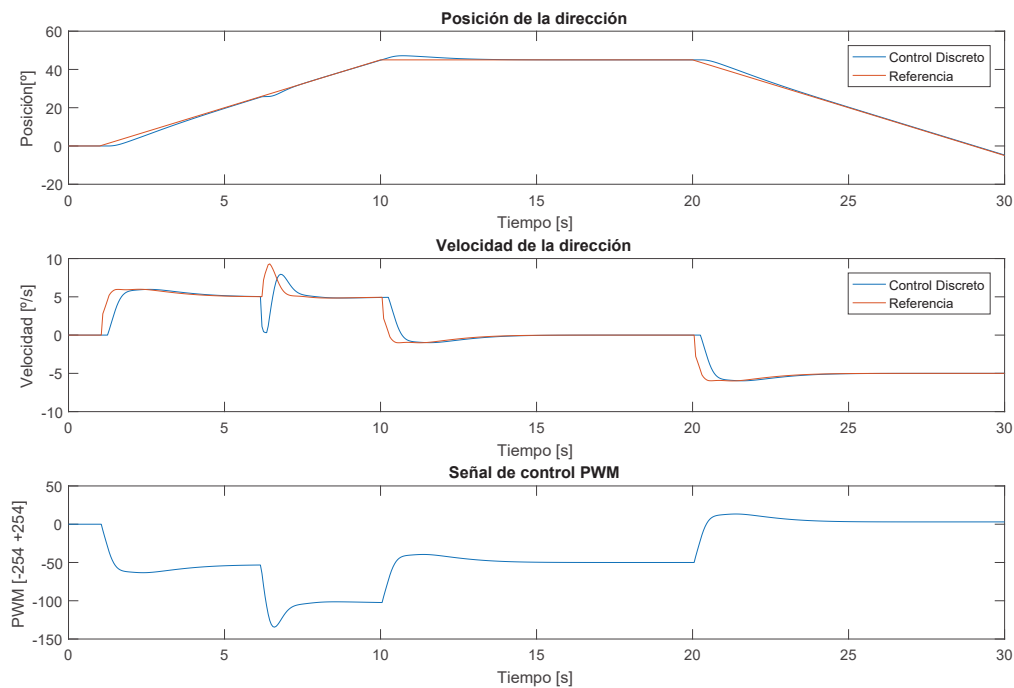


Figura 5.13: Resultados en simulación con entradas en rampa frente a perturbaciones.

## 5.4. Ensayos experimentales

Una vez determinada la viabilidad del sistema de control desarrollado en la sección anterior, se implementa en el sistema real como se puede ver en el código descrito en la sección 3.3.2. Tras la correcta implementación del sistema discreto se compila en el proyecto de *ROS* operativo en el vehículo *eCARM*. Para la prueba del sistema completo se han realizado diversos ensayos en dos tipos de suelo distinto: El suelo de la nave uno del CITE IV y el suelo de la zona comprendida entre el CITE IV y los aparcamientos de la universidad. Un ensayo de gran interés consiste en una vez comprobado el correcto funcionamiento del sistema en bucle cerrado, introducir alguna perturbación para evaluar la respuesta del sistema. No obstante no se ha realizado para evitar daños en el sistema del vehículo al forzar el mecanismo a conciencia, aunque si el sistema responde igual que en simulación es de esperar que rechace con la misma eficacia las perturbaciones reales que en la figura 5.13.

El ensayo realizado en el suelo de la nave uno se muestra en la figura 5.15 y además se realiza en él la comparación con el modelo construido en *Simulink* (figura 5.14) sometido a la misma señal de referencia. En la figura 5.16 se encuentra el ensayo realizado en el otro suelo mencionado.

## 5.4. ENSAYOS EXPERIMENTALES

Hay que señalar en el segundo ensayo, hasta el segundo 30 el control se realiza de forma manual, por tanto este primer periodo de tiempo carece de relevancia para evaluar el control. Según los resultados obtenidos, se puede considerar que el trabajo desarrollado a lo largo del presente capítulo se ha realizado de forma satisfactoria cumplimentando los objetivos establecidos para el mismo. La solución de compromiso adoptada en la elección del rechazo del filtro en la referencia ha resultado ser apropiada y no repercute negativamente sobre la señal de respuesta. La señal referida a la posición del motor presenta un error nulo, o cercano a él, en el seguimiento a referencias en forma de rampa con una ligera sobreoscilación.

Para concluir se realiza una ampliación de los datos obtenidos en el ensayo mostrado en la figura 5.16 con el fin de apreciar mejor las características del sistema. Dicha ampliación se muestra en la figura 5.17. En ella se aprecia con mayor claridad la fidedigna reproducción obtenida en el modelo realizado en *Simulink*. Esta exactitud no se obtiene en la posición del mecanismo, sino en la velocidad y en la señal de control generada. Esto indica dos consideraciones importantes. La primera es que los mecanismos empleados para el modelado y técnicas de control como el *Predictor de Smith*, el mecanismo anti-windup y control en cascada se han desarrollado correctamente y son viables para la implementación en un sistema real como el vehículo *eCARM*. La segunda consideración es que el modelo obtenido y sistema implantado es lo suficientemente bueno como para poder desarrollar técnicas más avanzadas de control y poder experimentar con el en simulación sin comprometer el vehículo real.

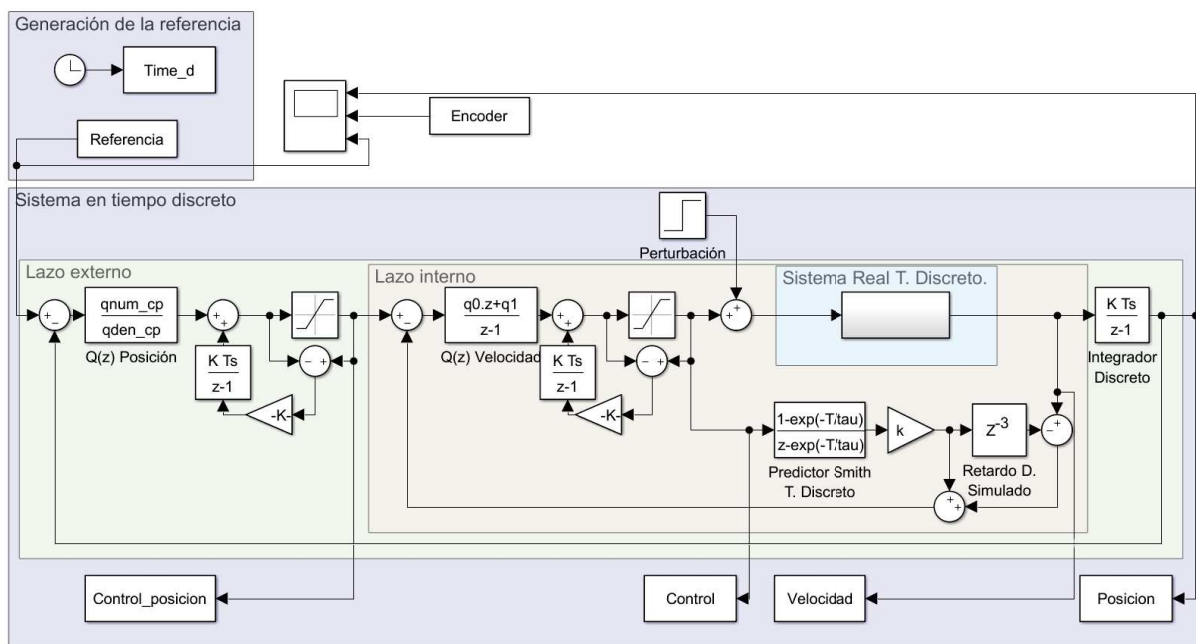


Figura 5.14: Esquema de *Simulink* para la validación de datos.

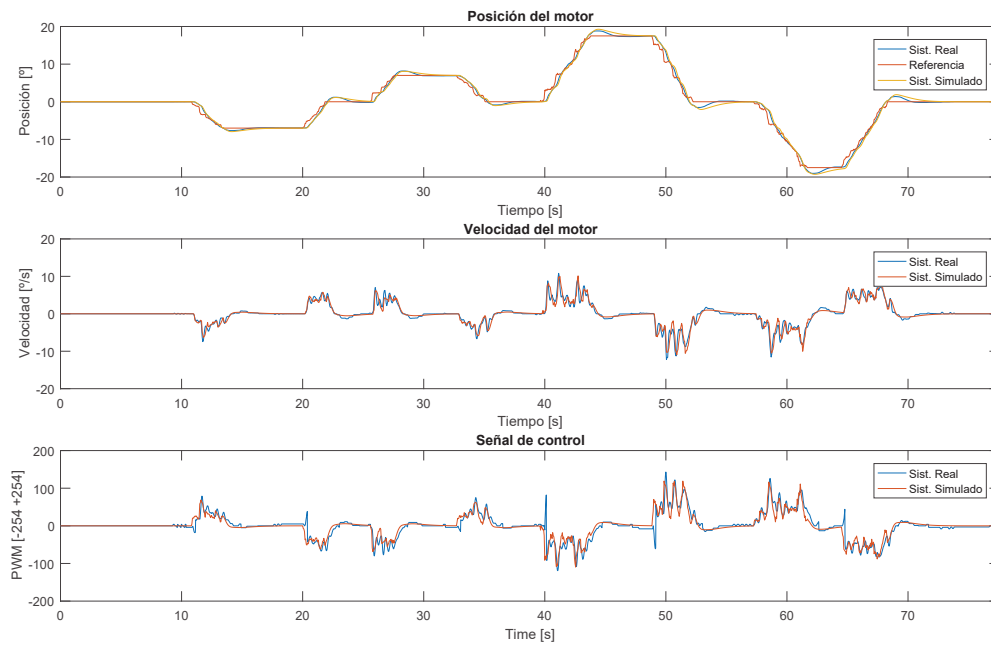


Figura 5.15: Resultados experimentales del primer ensayo.

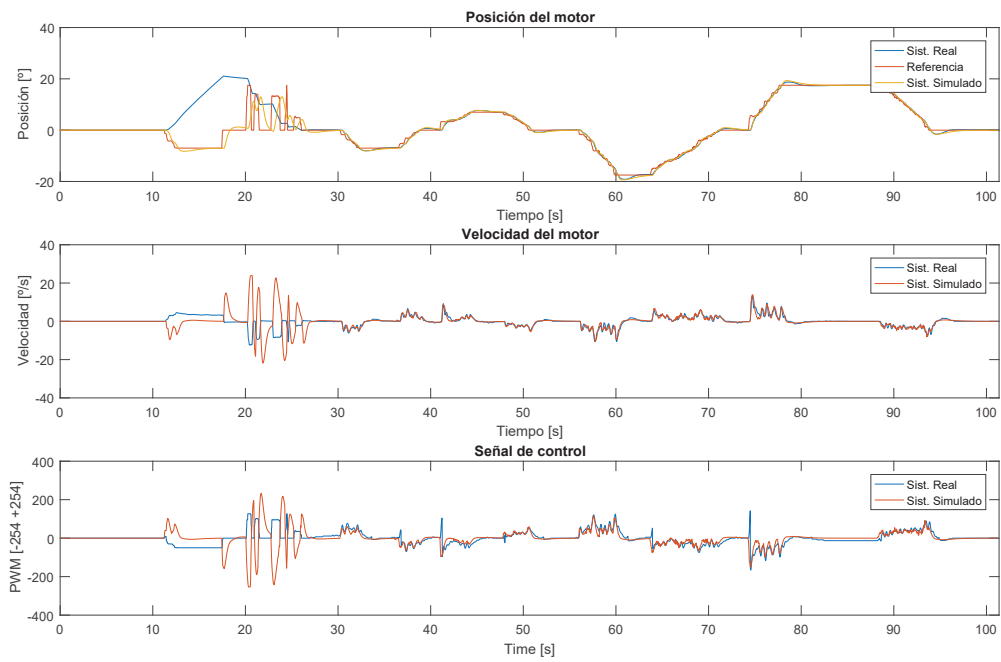


Figura 5.16: Resultados experimentales del segundo ensayo.

## 5.4. ENSAYOS EXPERIMENTALES

---

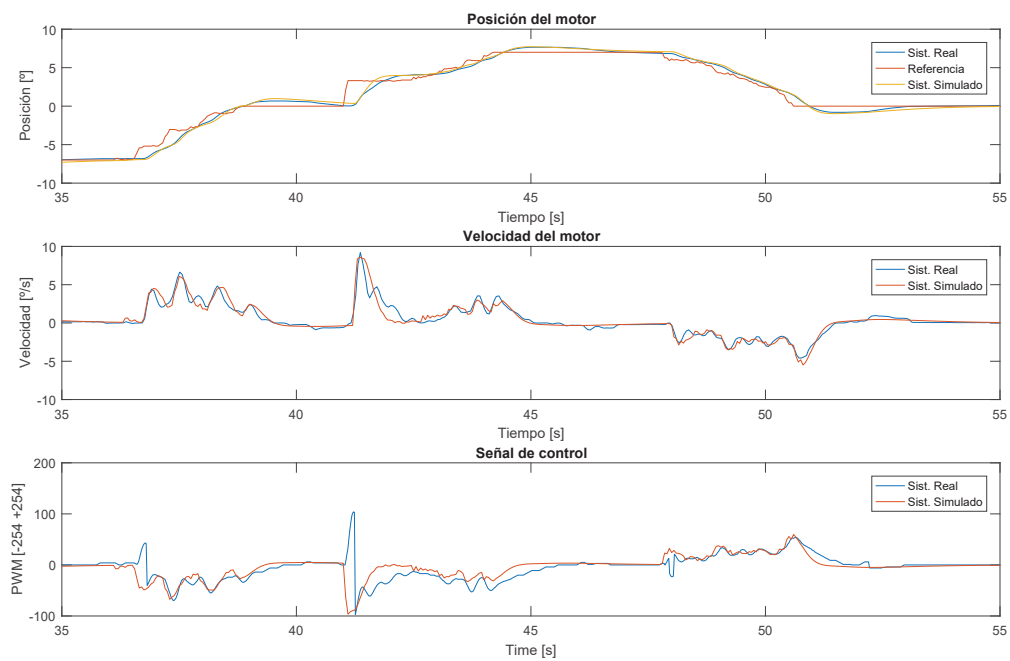


Figura 5.17: Resultados experimentales del primer ensayo (Ampliado).

## Capítulo 6

# Conclusiones y trabajos futuros

### 6.1. Conclusiones

Del trabajo desarrollado en los capítulos del actual trabajo ha reportado información de gran valor para el conocimiento del estado actual del vehículo *eCARM*. En esta sección se comentarán los resultados obtenidos para los dos sistemas *by-wire* tratados englobados bajo el concepto *Drive-by-wire*.

En primer lugar, el sistema *Throttle-by-wire*, abordado en el capítulo 4, ha podido ser modelizado mediante el empleo de herramientas de indentificación de sistemas con el modelo no lineal ARX. Su modelado se ha llevado a cabo considerando la velocidad angular del motor como salida y el voltaje del pedal como entrada. Esto se ha considerado así y no tomando como salida la velocidad lineal del vehículo puesto que gran parte de los ensayos se han realizado con el vehículo inmóvil y las ruedas motrices suspendidas en el aire. No obstante, si se desea operar con la velocidad lineal, basta con dividir la velocidad angular del motor por la relación de transmisión entre el motor y las ruedas, 9,75, y el diámetro de las mismas, 0,2m. Esta transformación será de utilidad en niveles superiores de control donde las referencias introducidas sean de velocidad lineal.

El trabajo en este sistema se ha limitado a su modelado puesto que tras la complejidad del sistema obtenido, se ha considerado que obtener un control de calidad implica una carga de trabajo superior, más propia de un trabajo dedicado en exclusiva a este sistema.

El sistema *Steer-by-wire* se abordó en el capítulo 5. Como resultado del trabajo realizado en dicho capítulo se consigue un modelado mediante una función de transferencia de primer orden para la velocidad de giro y un primer orden con un integrador en la posición. La validación de estos modelos muestra que los resultados obtenidos son una fiel reproducción del sistema real tanto para ensayos con el vehículo suspendido como en el vehículo sobre el cualquier superficie. Una vez determinadas las propiedades del sistema, se aborda como un control en cascada haciendo uso de dos controladores tipo PI y PID para los lazos internos y externos respectivamente.

El sistema de control implementado consigue cumplir con los objetivos establecidos para este sistema al inicio del trabajo realizado. Para ello hace uso de estrategias de control y protección empleadas frecuentemente en la industria como es el uso del modelo PID para los controladores, el *Predictor de Smith* para controlar sistemas de retardo dominante o mecanismo de saturación y antiwindup para la protección de los actuadores del sistema. Con esto se consigue finalizar el control de este sistema para su empleo en niveles de control superiores del vehículo *eCARM*.

### 6.2. Trabajos futuros

Una de las características que hace del vehículo *eCARM* un sistema de gran valor es amplio abanico de posibilidades temáticas de trabajo que ofrece. En este trabajo se ha buscado caracterizar dos de los sistemas de control principales requeridos para la conducción del vehículo. No obstante el trabajo no está finalizado con los resultados obtenidos. Se ha llevado a cabo el modelado desde el enfoque de ambos sistemas como una “caja negra” de una sola entrada y una única salida, pero el sistema acepta otros enfoques. Un trabajo de gran interés reside en repetir el trabajo desarrollado en estos capítulos con un enfoque basado en primeros principios y comparar los resultados obtenidos. Otra posibilidad del enfoque es no considerar el sistema *Throttle-by-wire* como un sistema SISO, sino como un sistema MISO, cuyas entradas, además del voltaje del pedal fueran el peso del vehículo y el estado de carga de las baterías. Estos parámetros de entrada se consideran relevantes puesto que modifican la dinámica del sistema con el transcurso del funcionamiento. Conocer dichas dependencias externas al motor del vehículo son cruciales para la implementación de un sistema de control correcto.

Pero el vehículo no ofrece más trabajo solo con el cambio de enfoque del modelado. En el caso del sistema *Throttle-by-wire* se han observado errores que han perjudicado directamente en el trabajo sobre él. Esto implica la necesidad del sistema de instalación de nuevos sensores de mayor calidad que faciliten un mejor conocimiento de la dinámica del sistema. Con estos sensores se podría obtener datos más precisos para un mejor modelado y comprobar que los errores asociados a la lectura de los sensores actuales pueden ser solventados con su sustitución y no requieren de cambios de otros componentes como la tarjeta de adquisición de datos.

Otra vertiente de trabajo con gran interés por las características del vehículo sería la adecuación de los sensores, ya instalados pero en desuso en este trabajo, para su lectura por parte del sistema *ROS*. Esto permitiría al usuario del vehículo tener un mayor conocimiento del sistema en tiempo real. Algunos de estos parámetros de interés son la tensión que suministran las baterías, la corriente circulante por el motor o las que suministran las baterías. Estos datos, al igual que otros recogidos de diversos sensores instalados están cableados hacia una tarjeta de adquisición de datos de *National Instrument*, pero no se encuentran asociados al sistema *ROS*. Además de presentar interés para la lectura de datos a tiempo real, otro interés que presentan reside en la posibilidad de implementar estrategias de control multiobjetivo.



Un trabajo susceptible a ser realizado, como complemento a la puesta en marcha de nuevos sensores o a la implementación de distintas estrategias de control, se encuentra en la comparación de los resultados obtenidos en el vehículo en los diferentes trabajos realizados en él. Un ejemplo sería comparar el resultado del actual trabajo con el obtenido en ensayos previos con distintos sensores como el mostrado en la figura 6.1. En dicho ensayo se pudo modelar la velocidad del coche en función de la tensión de entrada como una *f.d.t.* de primer orden, no obstante en este trabajo resulta imposible llegar a dicho modelado.

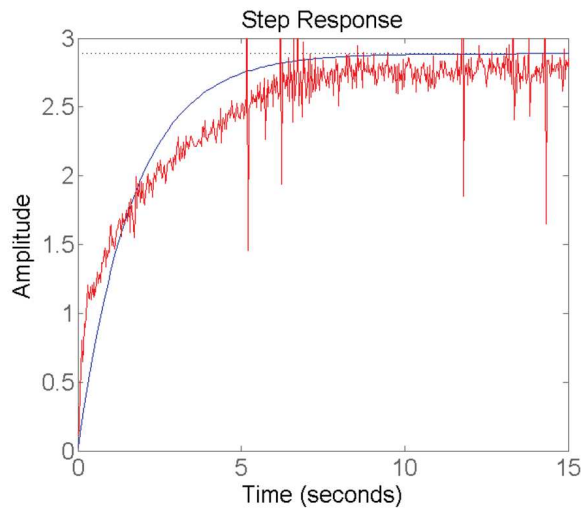


Figura 6.1: Resultados del modelado llevado a cabo en ensayos anteriores.

Finalmente, el trabajo que se busca realizar a través del desarrollo de sistemas de control en los niveles inferiores de la pirámide del control de un robot móvil autónomo (figura 2.4), es el desarrollo de la conducción autónoma del vehículo. Para poder llevar a cabo dicho nivel de control, antes es necesario que los niveles inferiores posean sistemas de control robustos que respondan de forma adecuada a los niveles superiores como ya se ha comentado anteriormente. Para la operación en estos niveles, el vehículo está equipado, aunque sin hacerse uso de él, de un escáner láser instalado en el vehículo y en el sistema operativo *ROS* se puede encontrar un nodo en *mrpt* para navegación autónoma del coche de alto nivel, *mrpt\_reactivenav2d* [49].



# Bibliografía

- [1] Electronic Motor Controller. Manual Model 1268. 2012.
- [2] Historia del coche eléctrico. [Consulta el 31-07-2017]. Disponible en: <https://hipertextual.com/2016/01/historia-del-coches-electrico>.
- [3] Jamais contente parade - La Jamais contente — Wikipédia. [Consulta el 31-07-2017]. Disponible en: [https://fr.wikipedia.org/wiki/La\\_Jamais\\_contente](https://fr.wikipedia.org/wiki/La_Jamais_contente).
- [4] Infiniti utilizará la tecnología Fly-by-Wire en las direcciones de sus modelos. [Consulta el 31-07-2017]. Disponible en: <http://www.highmotor.com/infiniti-utilizara-tecnologia-fly-by-wire-direcciones-modelos.html>.
- [5] 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*, 10(4):2003–2015, 2014.
- [6] A. E. Cetin, M. A. Adli, D. Erol Barkana, and H Kucuk. Implementation and development of an adaptive steering-control system. *IEEE Transactions on Vehicular Technology*, 59(1):75–83, 2010.
- [7] M. Zhang, X. Wu, and M. Xu. Adaptive feedforward control of the steer-by-wire system based on the online parameter estimator. *2016 IEEE International Conference on Robotics and Biomimetics, ROBIO 2016*, pages 892–897, 2017.
- [8] evolutionelectricvehicles.com. [Consulta el 31-07-2017]. Disponible en: <http://evolutionelectricvehicles.com/product-408.html>.
- [9] J. E. Normey-Rico and Eduardo F. Camacho. Unified approach for robust dead-time compensator design. *Journal of Process Control*, 19(1):38–47, 2009.
- [10] L. Kleinrock. An early history of the internet [History of Communications]. *IEEE Communications Magazine*, 48(8):26–36, 2010.
- [11] S. Gong and M. Karlsson. Pushing the Wireless Data Rate to the Internet Speed. *IEEE Access*, 4:8787–8792, 2016.
- [12] Changzhou Greenland Vehicle Co. [Consulta el 2-05-2017]. Disponible en: [http://www.gl-ev.com/en/pro\\_show.php?Id=155](http://www.gl-ev.com/en/pro_show.php?Id=155).

- [13] J. Ramos Teodoro. *Modelado y control de la dirección de un vehículo eléctrico*. PhD thesis, Universidad de Almería, 2014.
- [14] 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*, 8(6):993–1002, 2000.
- [15] Y. Kim, H. Fujimoto, and Y. Hori. Realization of steer-by-wire system for electric vehicles using caster wheels and independent driving motors. *8th International Conference on Power Electronics - ECCE Asia: "Green World with Power Electronics", ICPE 2011-ECCE Asia*, pages 1688–1695, 2011.
- [16] S. Maejima and T. Murakami. Driving assist control of electric vehicle by steer-by-wire system considering collision avoidance of traveling bicycle. *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pages 1657–1661, 2015.
- [17] G. Howser and B. McMillin. Modeling and reasoning about the security of drive-by-wire automobile systems. *International Journal of Critical Infrastructure Protection*, 5(3-4):127–134, 2012.
- [18] R. Jiménez Cano. El dueño de un Tesla, primer muerto en un coche con piloto automático. [Consulta el 1-08-2017]. Disponible en: [https://elpais.com/tecnologia/2016/07/01/actualidad/1467337732\\_779288.html](https://elpais.com/tecnologia/2016/07/01/actualidad/1467337732_779288.html), jul 2016.
- [19] G. V. Raffo, J. E. Normey-Rico, F. R. Rubio, and C. R. Kelber. Control Predictivo en Cascada de un Vehículo Autónomo. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 6(May):287–293, 2008.
- [20] M. Elbanhaw and M. Simic. Sampling-Based Robot Motion Planning. *IEEE Access*, 2:56–77, 2014.
- [21] X. Yang, K. He, M. Guo, and B. Zhang. An Intelligent Predictive Control Approach to Path Tracking Problem. *IEEE International Conference on Systems, Man, and Cybernetics*, 4:3301–3306, 1998.
- [22] J. E. Normey-Rico, J. Gómez-Ortega, and E. F. Camacho. A Smith-predictor-based generalised predictive controller for mobile robot path-tracking. *Control Engineering Practice*, 7:729–740, 1999.
- [23] F.D. Boyden and S.a. Velinsky. Dynamic modeling of wheeled mobile robots for high load applications. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994.
- [24] J-M. Yang and J-H. Kim. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 15(3):578–587, 1999.
- [25] B. Kim, D. Neculescu, and J. Sasiadek. Model predictive control of an autonomous vehicle. *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)*, 2(July):1279–1284, 2001.

- 
- [26] 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*, 2009.
- [27] Y. Marumo and M. Nagai. Steering control of motorcycles using steer-by-wire system. *Vehicle System Dynamics*, 45(5):445–458, 2007.
- [28] Y. Marumo and N. Katagiri. Control effects of steer-by-wire system for motorcycles on lane-keeping performance. *Vehicle System Dynamics*, 49(8):1283–1298, 2011.
- [29] S. H. Jang, T. J. Park, and C. S. Han. A control of vehicle using steer-by-wire system with hardware-in-the-loop-simulation system. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 1:389–394, 2003.
- [30] P. Setlur, D. Dawson, J. Chen, and J. Wagner. A nonlinear tracking controller for a haptic interface steer-by-wire systems. *Proceedings of the IEEE Conference on Decision and Control*, 3(December):3112–3117, 2002.
- [31] R. Ikeura, T. Moriguchi, and K. Mizutani. Optimal variable impedance control for a robot and its application to lifting an object with a human. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pages 500–505, 2002.
- [32] K. Iqbal and Y.F. Zheng. Arm-manipulator coordination for load sharing using predictive control. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 4(May):0–5, 1999.
- [33] A. E. Cetin and M. A. Adli. Cooperative control of a human and a robot manipulator for positioning a cart on a frictionless plane. *Mechatronics*, 16(8):461–469, 2006.
- [34] T. Fukao, S. Miyasaka, K. Mori, N. Adachi, and K. Osuka. Active steering systems based on model reference adaptive nonlinear control. *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, (May 2001):37–41, 2001.
- [35] T. Hägglund. A unified discussion on signal filtering in PID control. *Control Engineering Practice*, 21(8):994–1006, 2013.
- [36] J. L. Guzmán and T. Hägglund. Simple tuning rules for feedforward compensators. *Journal of Process Control*, 21(1):92–102, 2011.
- [37] Arduino Pro Mini 328 - 3.3V/8MHz - DEV-11114 - SparkFun Electronics. [Consulta el 31-07-2017]. Disponible en: <https://www.sparkfun.com/products/11114>.
- [38] Raspberry Pi 3 Model B - Raspberry Pi. [Consulta el 31-07-2017]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [39] Powering the world’s robots. [Consulta el 31-07-2017]. Disponible en: <http://www.ros.org/>.
- [40] Topics - ROS Wiki. [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/Topics>.
- [41] RQT Graph - ROS Wiki. [Consulta el 31-07-2017]. Disponible en: [http://wiki.ros.org/rqt\\_graph](http://wiki.ros.org/rqt_graph).
- [42] ROSBAG - ROS Wiki. [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/rosbag>.

- [43] ROSCORE - ROS Wiki. [Consulta el 31-07-2017]. Disponible en: <http://wiki.ros.org/roscore>.
- [44] J. L. Blanco. ROS packages and config files for University of Almeria autonomous eCAR. [Consulta el 31-07-2017]. Disponible en: <https://github.com/ual-arm-ros-pkg/ual-ecar-ros-pkg>.
- [45] J. L. Blanco. ROS node for UAL eCAR's Arduino-based DAQ system: 4 analog outputs (MAX5500), GPIO, built-in ADCs, PWM, quadrature encoders. [Consulta el 31-07-2017]. Disponible en: <https://github.com/ual-arm-ros-pkg/arduino-daq-ros-pkg>.
- [46] K. Ogata. *Ingeniería de Control Moderna*. Prentice Hall, Madrid, 5ª edición edition, 2010.
- [47] K. Ogata, J. Aranda Pérez, and F. Rodriguez Ramírez. *Sistemas de control en tiempo discreto*. Prentice Hall Hispanoamericana, México, 2ª edición edition, 2003.
- [48] D. E. Rivera, M. Morari, and S. Skogestad. Internal model control: PID controller design. *Industrial Engineering Chemistry Process Design and Development*, pages 252–265, 1986.
- [49] Mrpt\_reactivenav2d - ROS Wiki. [Consulta el 31-07-2017]. Disponible en: [http://wiki.ros.org/mrpt\\_reactivenav2d](http://wiki.ros.org/mrpt_reactivenav2d).



La elevada demanda de vehículos, el progreso tecnológico asociado a la automatización de nuestro entorno y la preocupación por la conservación del medio ambiente está guiando la investigación de vehículos hacia mejores resultados con vehículos eléctricos y autónomos. Pero el control a tan alto nivel no es posible si en los niveles inferiores, los sistemas de control no son lo suficientemente eficaces.

Este trabajo presenta como objetivo la realización de un control eficaz a bajo nivel sobre los sistemas *Steer-by-wire* y *Throttle-by-wire*. Para ello, en primer lugar se debe realizar el modelado de los sistemas actuales del vehículo. Una vez se comprueba la validez de los modelos obtenidos, se trabaja en el diseño de los sistemas de control mediante herramientas de simulación, en este caso se emplea principalmente *Simulink*. Los principales problemas tratados en el trabajo son la identificación de modelos de alto orden, en el caso del sistema *Throttle-by-wire*, y el control en cascada con un sistema de retardo dominante en el lazo interno para el caso del sistema *Steer-by-wire*. Posteriormente, con el sistema de control ya diseñado y comprobada su eficiencia en simulación, se implementa en el sistema real incorporándolo al sistema ya existente creado en ROS.

La principal aportación de este trabajo, además de la actualización de los modelos instalados en el vehículo *eCARM*, es la implementación del sistema de control completo en el nodo *steer\_controller* del sistema ROS.

The high demand of vehicles, the technological improvement associated to the automation of our surrounding and the concern about the environment is guiding the vehicles' research to better results with electric and autonomous car. But the control at such a high level isn't possible if at the lower levels the control systems aren't efficient enough.

This work presents as goal to do a efficient control at low level over the *Steer-by-wire* and *Throttle-by-wire* systems. For that, in first place, the modeling of the current vehicle systems must be performed. Once the ratification of the obtained models is verified, we work on the design of control systems using simulation tools, in this case *Simulink* is mainly used. The main problems confronted in the work are the identification of high-order models in the case of the *Throttle-by-wire* system and the cascade control with a dominant delay system in the internal loop for the case of the *Steer-by-wire*. Subsequently, with the control system already designed and tested its efficiency in simulation, it's implemented in the real system incorporating it to the existing system compiled in ROS.

The main contribution of this work, in addition to updating the models installed in the *eCARM* vehicle, is the implementation of the complete control system in the *steer\_controller* node of the ROS system.

