



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN INGENIERÍA ELECTRÓNICA, OPCIÓN
INSTRUMENTACIÓN ELECTRÓNICA**

**Tesis para obtener el grado de
MAESTRO EN INGENIERÍA ELECTRÓNICA**

**DESARROLLO DE UNA PLATAFORMA DE EMULACIÓN DE UN
VEHÍCULO ELÉCTRICO HÍBRIDO**

Presenta:

Manuel Aguilar Rodríguez*

Asesores:

Dra. Liliana Cortez

M.I. Ernest Cortez

Índice General

Lista de Figuras	III
Abreviaciones	VII
Agradecimientos	2
Resumen	3
1. Introducción	4
1.1. Vehículos eléctricos híbridos	4
1.2. Objetivo General	4
1.3. Objetivos Específicos	5
1.4. Justificación	5
1.5. Elementos de un vehículo eléctrico híbrido	7
1.5.1. Transmisión	7
1.5.2. Motor/Generador eléctrico	8
1.5.3. Motor de combustión interna	10
1.5.4. Sistema de almacenamiento de energía	12
1.6. Topología de los vehículos eléctricos híbridos	14
1.6.1. Topología serie	14
1.6.2. Topología paralela	15
1.6.3. Topología serie-paralela	16

2. Modelo de un vehículo eléctrico híbrido	18
2.1. Dinámica longitudinal	18
2.2. Modelo dinámico	20
2.3. Modelo cuasi-estático	24
3. Diseño de la plataforma de emulación	30
3.1. Desarrollo del sistema embebido	31
3.2. Diseño del protocolo de comunicación	38
3.3. Software para la plataforma de emulación	41
4. Implementación de la plataforma de emulación	45
4.1. Construcción y pruebas de funcionamiento de la plataforma de emulación . .	45
4.2. Simulación del vehículo híbrido en Simulink	50
4.3. Validación del funcionamiento de la plataforma de emulación	62
Conclusiones	65
.1. Firmware	67
.2. Vistas de software	73
Referencias	74

Lista de Figuras

1.1. Transmisión de un vehículo eléctrico híbrido.	7
1.2. Motor eléctrico.	8
1.3. Mapa de eficiencia de un motor [11].	9
1.4. Mapa de eficiencia de un motor/generador [14].	10
1.5. Motor de combustión interna [15].	11
1.6. Mapa bsfc [18].	11
1.7. Determinación del mapa bsfc [20].	12
1.8. Sistema de almacenamiento de energía [22].	12
1.9. Elementos de un sistema de almacenamiento de energía.	13
1.10. Estructura de una celda [26].	13
1.11. Modelo de un batería [27].	14
1.12. Tren de potencia de un VEH con topología serie.	15
1.13. Tren de potencia de un VEH con topología paralela.	16
1.14. Tren de potencia de un VEH con topología serie-paralela.	17
2.1. Fuerzas definidas en un vehículo eléctrico híbrido [38].	19
2.2. Controlador PID del modelo dinámico del VEH.	20
2.3. Modelo dinámico de la transmisión.	21
2.4. Modelo dinámico del motor/generador.	21
2.5. Modelo dinámico del convertidor de potencia.	22

2.6. Modelo dinámico del sistema de almacenamiento de energía.	22
2.7. Modelo dinámico del generador eléctrico.	23
2.8. Modelo dinámico del ICE.	23
2.9. Modelo dinámico del vehículo eléctrico híbrido.	24
2.10. Ciclo de conducción modal NEDC [42].	25
2.11. Ciclo de conducción transitorio UDDS [43].	25
2.12. Modelo cuasi-estático de un vehículo eléctrico híbrido.	26
2.13. Modelo cuasi-estático de la transmisión.	26
2.14. Modelo cuasi-estático de un motor/generador eléctrico.	27
2.15. Modelo cuasi-estático de un convertidor de potencia.	27
2.16. Modelo cuasi-estático de un sistema de almacenamiento de energía.	28
2.17. Modelo cuasi-estático de un sistema de un generador eléctrico.	28
2.18. Modelo cuasi-estático de un motor de combustión interna.	28
2.19. Modelo cuasi-estático de un VEH.	29
3.1. Plataforma de emulación.	30
3.2. Arquitectura Harvard en un microcontrolador.	32
3.3. Circuito mínimo para el microcontrolador PIC32MZ2048EFM144.	33
3.4. Diagrama UML del firmware.	34
3.5. Diagrama UML del sistema base.	35
3.6. Máquina de estados finitos del sistema base.	36
3.7. Máquina de estados finitos del modelo cuasi-estático.	37
3.8. Diagrama UML del protocolo de comunicación.	38
3.9. Algoritmo para la decodificación de la interfáz RS232.	39
3.10. Algoritmo para la transformación de datos.	40
3.11. Estructura de datos de configuración.	41
3.12. Diagrama UML del software.	41

3.13. Patrón MVC del software.	42
3.14. Modelo del patrón MVC.	43
3.15. Controlador del patrón MVC.	44
4.1. Placa de circuito impreso del sistema embebido.	46
4.2. Vista para la configuración de los parámetros del tren de potencia.	48
4.3. Resultado de la emulación de un vehículo eléctrico híbrido.	50
4.4. Modelo cuasi-estático en simulink.	51
4.5. Ciclo de conducción.	52
4.6. Fuerza de salida de la dinámica longitudinal del vehículo.	53
4.7. Velocidad angular de salida de la dinámica longitudinal del vehículo.	53
4.8. Simulación de la potencia de salida de la transmisión.	54
4.9. Simulación del convertidor de potencia hacia el generador.	55
4.10. SOC del sistema de almacenamiento de energía (Simulación).	56
4.11. Consumo de combustible (Simulación).	57
4.12. Emulación a través de la derivada del valor medio.	58
4.13. Emulación a través de la derivada del cociente de diferencia simétrica.	58
4.14. SOC del sistema de almacenamiento de energía (Emulación a través de la derivada del valor medio).	59
4.15. Consumo de combustible (Emulación a través de la derivada del valor medio).	59
4.16. SOC del sistema de almacenamiento de energía (Emulación a través de la derivada del cociente de diferencia simétrica).	60
4.17. Consumo de combustible (Emulación a través de la derivada del cociente de diferencia simétrica).	60
4.18. Análisis de tiempos del sistema embebido.	61
4.19. Comparación del SOC del sistema de almacenamiento de energía.	62
4.20. Comparación del consumo de combustible.	63

21. Vistas de la aplicación.	73
--------------------------------------	----

Abreviaciones

ADVISOR: Advanced Vehicle Simulator.

bsfc: Brake specific fuel consumption.

CAD: Computer Aided Design.

DSP: Digital Signal Processor.

EG: Electric generator.

EM: Electric motor.

ESS: Energy storage sistem.

η : Eficiencia.

F_{ad} : Resistencia aerodinámica.

FPU: Floating Point Unit.

F_{rr} : Resistencia de rodadura.

g : Constante gravitacional.

γ : Número de marcha.

ICE: Internal Combustion Engine.

ICSP: In-Circuit Serial Programming.

LVH: Lower heating value.

$\dot{m}$: Flujo másico.

mefComunicacion: Maquina de estados finitos del protocolo de comunicación.

mefVEH: Maquina de estados finitos del vehículo eléctrico híbrido.

μ_{rr} : Coeficiente de fricción.

MVC: modelo-vista-controlador.

Nm: Newton metro

NREL: National Renewable Energy Laboratory.

ω : Velocidad angular.

PC: Power converter.

PCB :Print Circuit Board.

$P_{entrada}$: Potencia de entrada de la transmisión

PLL: Phase-Locked Loop, lazo de seguimiento de fase.

ρ : Densidad del aire.

RPM: Revoluciones por minuto.

SOC: State of charge.

SRAM: Static Random Access Memory.

τ : Torque.

T: Transmition.

TC: Torque converter.

UML: Unified Modeling Language.

USART: Universal Asynchronous Receiver-Transmitter.

VEH: Vehículo eléctrico híbrido.

W: Wheel.

Agradecimientos

Agradezco al Consejo de Ciencia y Tecnología (CONACyT) y a la Benemérita Universidad Autónoma de Puebla (BUAP) por u apoyo y patrocinio para la realización de esté proyecto de tesis titulado "Desarrollo de una plataforma de emulación de un vehículo eléctrico híbrido". También agradezco a la maestría en Ingeniería Electrónica, Opción Instrumentación Electrónica por la oportunidad de desarrollar habilidades y competencias

A mis asesores de tesis, Dra. Liliana Cortez y cDr. Ernest Cortez por guiarme y apoyarme en el desarrollo tanto del trabajo de tesis como en mis habilidades y aptitudes.

A Dr José Italo Cortez por sus consejos, sus frases motivadoras y todo su apoyo.

A mis papás, Manuel Aguilar Gutierrez y Yolanda Rodríguez López, por siempre estar a mi lado, en sus enseñansas, consejos y cariño. A mis hermanos pos ser parte importante de mi.

A LISDER por convertirse en parte de mi familia y ayudarme a crecer.

Resumen

El presente trabajo muestra el diseño y desarrollo de una plataforma para la emulación de vehículos eléctricos híbridos. El documento se encuentra dividido en cuatro capítulos donde el primer capítulo presenta la introducción y una revisión de los vehículos eléctricos híbridos, sus componentes, sus arquitecturas de diseño y su clasificación.

En el capítulo dos se presenta el modelo matemático de un vehículo eléctrico híbrido, presentado dos perspectivas de modelado, se presenta el modelo dinámico y el modelo cuasi-estático.

El capítulo tres se presenta el diseño de la plataforma, donde se revisan las diferentes partes que la conforman, el software, el protocolo de comunicación y el sistema embebido. En el capítulo cuatro se aborda la implementación de la plataforma en el sistema embebido.

Se realizó una comparación entre la simulación de un vehículo eléctrico híbrido implementado en Simulink y la emulación realizada con la plataforma desarrollada. Se utilizaron parámetros reales de un vehículo, para realizar las comparaciones.

Finalmente se presentan las conclusiones y los resultados obtenidos con la simulación y la plataforma de emulación.

Capítulo 1

Introducción

1.1. Vehículos eléctricos híbridos

Actualmente, los vehículos eléctricos híbridos (VEH) se han establecido como una alternativa eficiente para reducir el consumo de combustible fósil, con el fin de mitigar la generación de gases de efecto invernadero [1, 2].

Un VEH se caracteriza por un tren de potencia, que incorpora un motor de combustión interna y un sistema de almacenamiento de energía como fuente de energía. Lo que permite incrementar la autonomía del vehículo, evitando la necesidad de un consumo de combustible adicional [3].

Además, el tren de potencia de un VEH provee diversos niveles de rendimiento, que dependen directamente de la topología seleccionada y de una estrategia de control [4].

1.2. Objetivo General

Desarrollar una plataforma de emulación de un vehículo eléctrico híbrido en un sistema embebido.

1.3. Objetivos Específicos

- Realizar una revisión bibliográfica de los antecedentes de las herramientas de diseño de vehículos eléctricos híbridos.
- Determinar el modelo matemático de los componentes en una topología serie de un vehículo eléctrico híbrido.
- Desarrollar una topología serie de un vehículo eléctrico híbrido en un sistema embebido.
- Desarrollar un software que permita configurar los componentes de la topología serie en la plataforma de emulación.
- Implementar un protocolo de comunicación entre el software y el sistema embebido.
- Realizar la validación de la plataforma de emulación.

1.4. Justificación

Actualmente, se ha incrementado la preocupación por el medio ambiente, debido al cambio climático causado por la quema de combustible fósil. Además, con base en los graves problemas ambientales, se han realizado diversos estudios que pronostican una escasez de combustible fósil.

Hoy en día, la industria automotriz representa un 54 % en consumo de combustible fósil a nivel mundial. Por ello, se ha realizado una búsqueda de soluciones que permitan la reducción del consumo de combustible sin afectar el rendimiento.

Los vehículos eléctricos híbridos (Hybrid Electric Vehicle, HEV) se han establecido como una alternativa eficiente para reducir el consumo de combustible fósil, que permitirá lograr una reducción gradual en la producción de gases de efecto invernadero.

Debido a la viabilidad de los vehículos eléctricos híbridos, se ha desarrollado una metodología clásica que se fundamenta en la simulación del tren de potencia con el fin de lograr un diseño eficiente. Lo que ha permitido la realización de múltiples herramientas de software.

El Laboratorio Nacional de Energía Renovable (National Renewable Energy Laboratory, NREL) ha desarrollado una herramienta de software denominada *ADVISOR*[®] (Advanced Vehicle Simulator), que se ha destacado en la simulación de vehículos eléctricos híbridos. Además, *ADVISOR*[®] permite obtener una estimación del consumo de combustible y las emisiones producidas debido a un ciclo de conducción estándar.

Aunque la metodología clásica ha permitido establecer un diseño preliminar de un vehículo eléctrico híbrido, no obstante, presenta la desventaja de no considerar parámetros reales a partir de un vehículo específico. Además, requiere de diversos ajustes considerables durante la fase de diseño e implementación de un prototipo y la fase de pruebas.

La necesidad de realizar múltiples prototipos y pruebas durante el desarrollo de vehículos eléctricos híbridos, ha dado lugar a una nueva metodología que se fundamenta en la emulación en tiempo real del tren de potencia. La emulación consiste en igualar el comportamiento de cada componente del tren de potencia a partir de un modelo matemático.

Como consecuencia de la complejidad que involucra el modelo matemático de cada componente, se han desarrollado diversos trabajos que emulan sólo algunos componentes del tren de potencia. Además, las propuestas desarrolladas requieren de un equipo especializado, que requiere una importante inversión y capacitación.

Con el fin de contribuir en la búsqueda de soluciones que permitan la reducción del consumo de combustible, en el presente trabajo se realizará el desarrollo de una plataforma de emulación de un vehículo eléctrico híbrido en un sistema embebido.

Además, la plataforma permitirá fomentar el desarrollo de nuevas soluciones en el diseño de vehículos eléctricos híbridos, debido a que contempla la emulación de todos los componentes de una topología serie considerando una mínima inversión. Asimismo, a través de un software será posible realizar la configuración de parámetros del vehículo eléctrico híbrido.

1.5. Elementos de un vehículo eléctrico híbrido

1.5.1. Transmisión

La transmisión de un vehículo eléctrico híbrido es un elemento mecánico del tren de potencia, Fig. 21, que tiene como función realizar una transformación mecánica de la potencia de entrada [5].

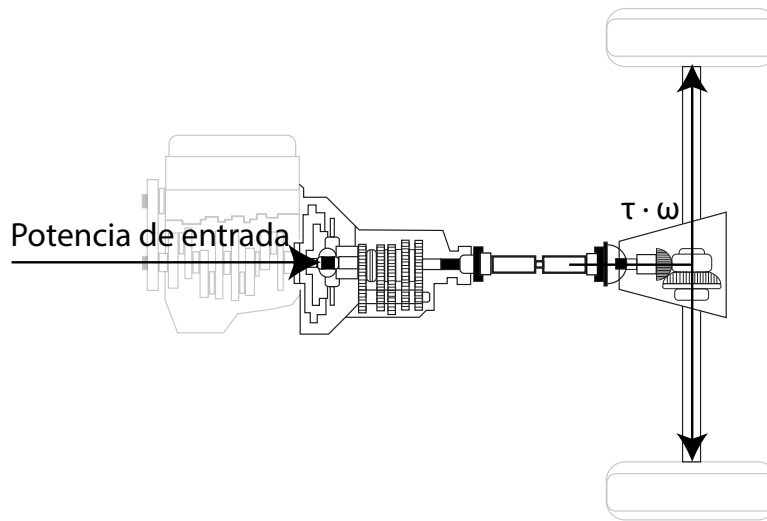


Fig. 1.1. Transmisión de un vehículo eléctrico híbrido.

La potencia de entrada (1.1) se define mediante una relación entre el torque y la velocidad angular.

$$P_{entrada} = \tau \omega. \quad (1.1)$$

La transformación que realiza la transmisión consiste en relacionar el número de marcha (γ) con un torque (1.2) y una velocidad angular (1.3) de salida.

$$\tau_{salida} = \frac{1}{\gamma} \tau_{entrada}, \quad (1.2)$$

$$\omega_{salida} = \gamma \omega_{entrada}. \quad (1.3)$$

Debido a las condiciones de funcionamiento de un VEH, es importante tomar en cuenta el flujo de potencia en la transmisión [6]. Se define un flujo positivo cuando el torque de entrada es mayor a 0 Nm (1.4), en caso contrario, se establece un flujo negativo (1.5).

$$\tau_{salida} = \frac{1}{\gamma \eta_t} \left(\tau_{entrada} + \frac{P_{ralenti}}{\omega_{entrada}} \right), \quad (1.4)$$

$$\tau_{salida} = \frac{1}{\gamma} \left(\eta_t \tau_{entrada} + \frac{P_{ralenti}}{\omega_{entrada}} \right). \quad (1.5)$$

donde, η_t es la eficiencia del motor generador y $P_{ralenti}$ es la potencia de operación mínima del motor de combustión interna en la que se encuentra en operación sin la necesidad de accionar el acelerador.

1.5.2. Motor/Generador eléctrico

El motor eléctrico, Fig. 1.2, es un elemento primordial, que tiene como función proveer una potencia mecánica al tren de potencia de un vehículo eléctrico híbrido.

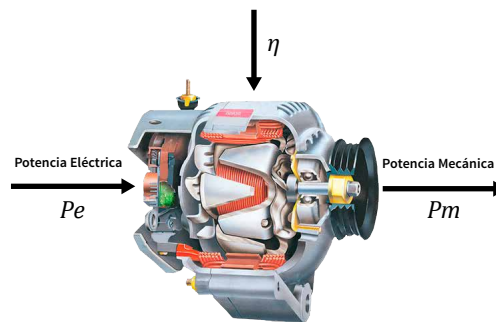


Fig. 1.2. Motor eléctrico.

Además, en determinadas condiciones un motor eléctrico puede ser utilizado como generador eléctrico, para proveer una energía eléctrica al sistema de almacenamiento de energía [7,8].

Actualmente, se emplean motores de corriente alterna en los VEH debido a la elevada eficiencia y densidad de potencia [9]. Aunque el costo de un motor de corriente alterna es reducido, requieren de sofisticados sistemas de control [10].

Para determinar el comportamiento de un motor/generador eléctrico sin la necesidad de un modelo matemático detallado, se requiere realizar un mapa de eficiencia denominado $\eta(\tau, \omega)$. Para ello, se procede a realizar sistemáticamente el cálculo de la eficiencia puntual, hasta cubrir el rango de operación del motor eléctrico [11,12].

La eficiencia (1.6) es un parámetro que relaciona la potencia mecánica (P_m) de salida y la potencia eléctrica (P_e) de entrada.

$$\eta(\tau, \omega) = \frac{P_m}{P_e} = \frac{\tau_m \omega_m}{P_e}. \quad (1.6)$$

Posteriormente, se determina el mapa de eficiencia del motor eléctrico, Fig. 1.3.

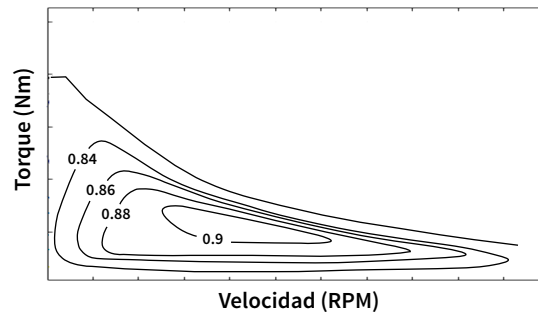


Fig. 1.3. Mapa de eficiencia de un motor [11].

Después de obtener el mapa de eficiencia de un motor eléctrico, se procede a determinar su comportamiento en modo generador. Para ello, se aplica el método denominado *mirroring*,

que consiste en reflejar la eficiencia sin considerar una pérdida (1.7) [13, 14].

$$\eta(-|\tau|, \omega) = \eta(|\tau|, \omega). \quad (1.7)$$

El método mirroring, permite determinar el mapa de eficiencia de un generador eléctrico a partir del mapa de eficiencia de un motor eléctrico, Fig. 1.4.

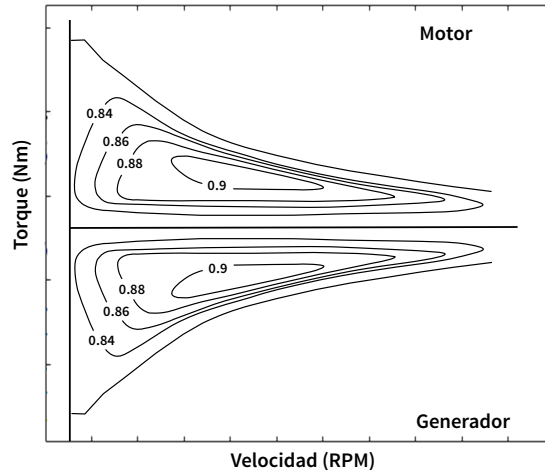


Fig. 1.4. Mapa de eficiencia de un motor/generador [14].

1.5.3. Motor de combustión interna

Un motor de combustión interna (Internal Combustion Engine, ICE), Fig. 1.5, consta de múltiples fenómenos que incrementan la complejidad del modelo matemático [15, 16].



Fig. 1.5. Motor de combustión interna [15].

Sin embargo, se ha logrado determinar el consumo de combustible a través de un mapa estático no lineal denominado *mapa bsfc* (Brake specific fuel consumption), Fig. 1.6 [17, 18].

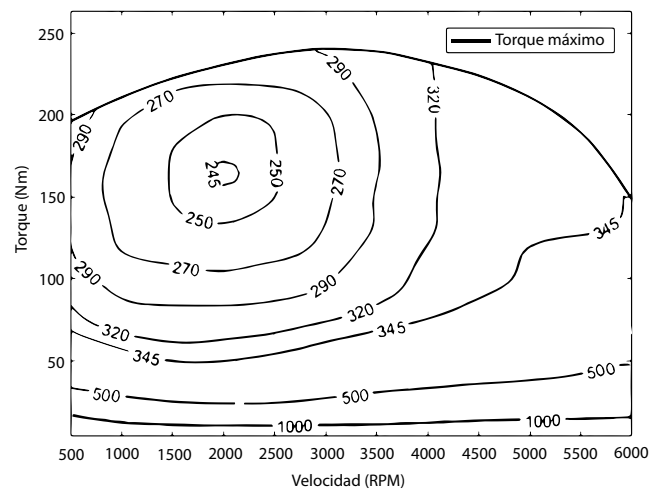


Fig. 1.6. Mapa bsfc [18].

El mapa bsfc se determina a partir de la medición del flujo de combustible que demanda un motor de combustión interna, cuando se le aplica un freno ante una determinada potencia mecánica, Fig. 1.7. El proceso se realiza sistemáticamente, con el fin de cubrir el rango de operación del motor de combustión interna [19, 20].

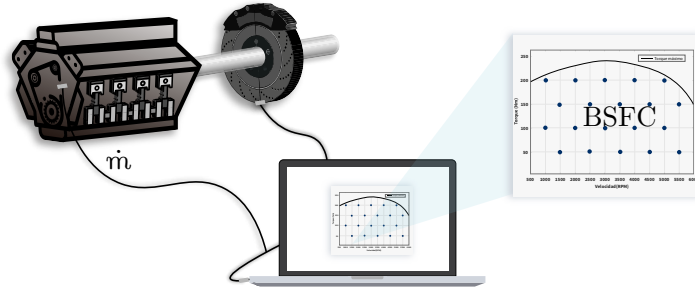


Fig. 1.7. Determinación del mapa bsfc [20].

Posteriormente, se emplea el mapa bsfc para determinar el flujo másico (1.8), con el fin de calcular el consumo de combustible que se requiere para producir un determinado torque y una velocidad angular [21].

$$\dot{m} = BSFC(\tau_{t-ICE}, \omega_{t-ICE}) \tau_{t-ICE} \omega_{t-ICE}. \quad (1.8)$$

1.5.4. Sistema de almacenamiento de energía

El sistema de almacenamiento de energía, Fig. 1.8, es un elemento primordial de un vehículo eléctrico híbrido, tiene como función almacenar y proveer energía eléctrica al tren de potencia.

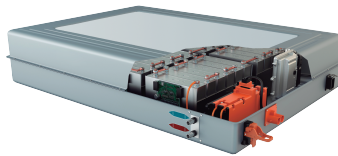


Fig. 1.8. Sistema de almacenamiento de energía [22].

Aunque un sistema de almacenamiento de energía puede incluir diversos elementos como baterías, ultracapacitores y/o celdas de combustible, Fig. 1.9, actualmente las baterías actúan como elemento primario [22, 23].

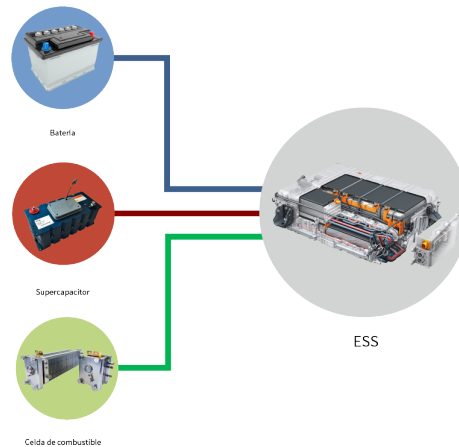


Fig. 1.9. Elementos de un sistema de almacenamiento de energía.

Actualmente, los vehículos eléctricos híbridos requieren de baterías que provean una alta energía específica y potencia específica. Además, debido al desgaste causado por la carga y descarga, ha incrementado el interés por un ciclo de vida elevado [24, 25].

Una batería se compone de diversas celdas individuales, Fig. 1.10, que poseen dos electrodos y un medio para proporcionar un mecanismo de transporte de iones entre los electrodos [26].

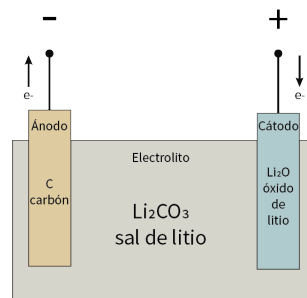


Fig. 1.10. Estructura de una celda [26].

Es posible modelar matemáticamente el comportamiento de una batería a través del modelo de Thevenin, Fig. 1.11 [27].

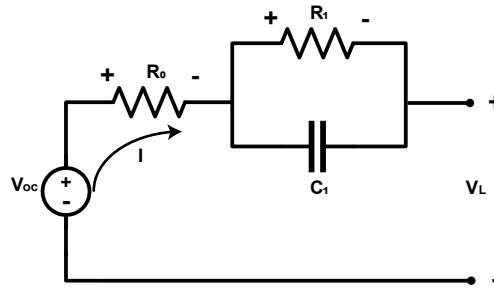


Fig. 1.11. Modelo de un batería [27].

A través de un análisis de circuitos se determina el voltaje V_L (1.9).

$$V_L = V_{OC} - R_0 I_L - \int \left(\frac{I_L}{C_1} - \frac{V_{R_1 C_1}}{R_1 C_1} \right) dt. \quad (1.9)$$

1.6. Topología de los vehículos eléctricos híbridos

La interconexión de los elementos en un tren de potencia, tiene un impacto directo en la eficiencia del vehículo eléctrico híbrido [28]. Actualmente, un tren de potencia de un VEH es posible clasificarlo en tres principales topologías:

- Topología serie
- Topología paralela
- Topología serie-paralela

1.6.1. Topología serie

La topología serie, Fig. 1.12, emplea el motor de combustión interna como unidad de potencia auxiliar, con el fin de extender el rango de conducción mediante un modo 100 % eléctrico [29].

Además, es posible emplear el motor eléctrico como generador, con el fin de recuperar la energía durante un proceso de frenado y enviarla al sistema de almacenamiento de energía [30].

Debido a que el ICE es considerado como un elemento auxiliar, se elimina el consumo causado por el motor en estado de espera. Por lo tanto, el motor puede funcionar en un punto de eficiencia y emisiones óptimas [31].

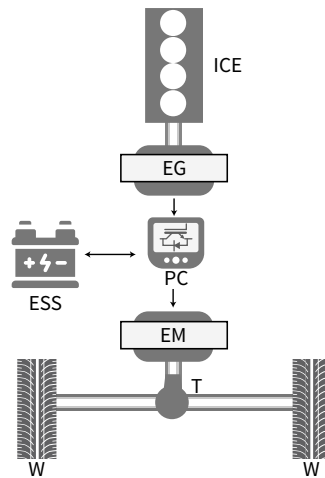


Fig. 1.12. Tren de potencia de un VEH con topología serie.

1.6.2. Topología paralela

Un tren de potencia en topología paralela, Fig. 1.13, provee un grado de libertad adicional, debido a que permite suministrar la potencia a la tracción mediante el motor eléctrico o el motor de combustión interna [32].

La topología permite optimizar la distribución de potencia, debido a la existencia de una conexión directa entre la transmisión y dos fuentes de energía independientes [33].

A diferencia de una topología serie, no se requiere un generador independiente, sin embargo, es necesario incluir un acoplador de torque.

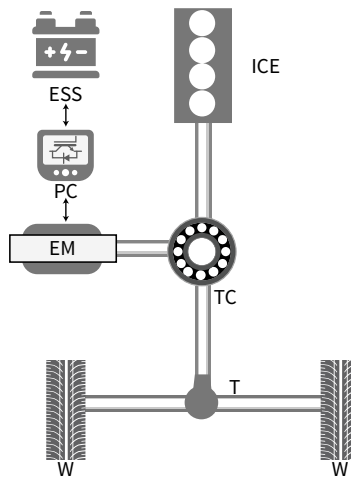


Fig. 1.13. Tren de potencia de un VEH con topología paralela.

1.6.3. Topología serie-paralela

Una topología serie-paralela, Fig. 1.14, provee un tren de potencia que incluye la capacidad de proveer una potencia a la tracción mediante el motor eléctrico o el motor de combustión interna. Además, se considera una retroalimentación a través de un generador, con el objetivo de aprovechar la energía que provee el motor de combustión interna para cargar el sistema de almacenamiento de energía [34].

A través de los elementos que conforman un vehículo eléctrico híbrido, se ha logrado definir su comportamiento. Además, la interconexión de los elementos del tren de potencia de un vehículo eléctrico híbrido, ha sido clasificada en diversas topologías.

En el presente capítulo se realizó una revisión bibliográfica de los elementos que conforman un vehículo eléctrico híbrido y la interconexión de los elementos en un tren de potencia.

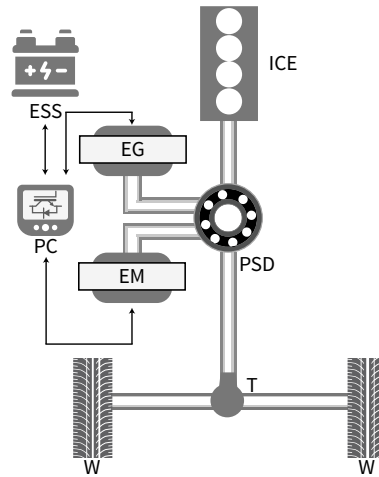


Fig. 1.14. Tren de potencia de un VEH con topología serie-paralela.

Capítulo 2

Modelo de un vehículo eléctrico híbrido

Actualmente, el desarrollo de vehículos eléctricos híbridos requiere de modelos matemáticos, que permitan describir el comportamiento de los elementos del tren de potencia [35], [36]. Es posible definir dos tipos de modelo matemático, el modelo dinámico y el modelo cuasi-estático [37]. Sin embargo, ambos modelos requieren conocer el comportamiento del vehículo eléctrico híbrido, que se describe mediante su dinámica longitudinal.

2.1. Dinámica longitudinal

La dinámica longitudinal de un vehículo eléctrico híbrido representa matemáticamente el comportamiento físico, que se obtiene a partir de un análisis de fuerzas [38]. Para ello, se definen aquellas fuerzas que actúan longitudinalmente sobre el vehículo eléctrico híbrido, Fig.

2.1.

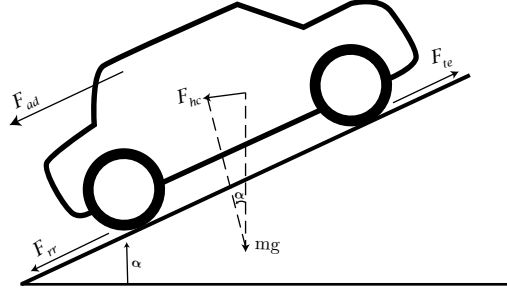


Fig. 2.1. Fuerzas definidas en un vehículo eléctrico híbrido [38].

Se define la fuerza F_{rr} (2.1) como la resistencia a la rodadura, que representa una variable relacionada con la pendiente de la superficie.

$$F_{rr} = mg\mu_{rr} \cos(\alpha). \quad (2.1)$$

Donde m es la masa total del vehículo eléctrico híbrido en kg, g la constante gravitacional ($g = 9.81 \frac{m}{s^2}$) y μ_{rr} es el coeficiente de fricción.

La fuerza F_{ad} (2.2), se define como la resistencia aerodinámica, que presenta el vehículo eléctrico híbrido para desplazarse en dirección frontal.

$$F_{ad} = \frac{1}{2} \rho A \mu_d v^2. \quad (2.2)$$

Donde ρ representa la densidad del aire, A el área frotal del VEH en m^2 y μ_d el coeficiente aerodinámico.

La fuerza F_{hc} (2.3), se define como la fuerza requerida para desplazar el vehículo eléctrico híbrido por una pendiente.

$$F_{hc} = mg \sin(\alpha). \quad (2.3)$$

Aplicando la segunda Ley de Newton a un vehículo eléctrico híbrido, se obtiene la dinámica

longitudinal (2.4).

$$F_{te} = ma + mg\mu_{rr} \cos(\alpha) + \frac{1}{2}\rho A\mu_d v^2 + mg \sin(\alpha). \quad (2.4)$$

2.2. Modelo dinámico

En el enfoque dinámico cada subcomponente tiene variables de interconexión con el subsistema anterior y los siguientes bloques. De esta manera, cada submodelo está entrelazado con los demás y su comportamiento tiene influencia en el sistema total. Este enfoque requiere de un gran tiempo y carga computacional, debido a ello, su aplicación es a menudo limitada a las áreas de investigación que se ocupan del desarrollo del motor de combustión interna.

El enfoque dinámico se basa en una descripción matemática del sistema. Por lo general, el modelo del tren de potencia se formula utilizando conjuntos de ecuaciones diferenciales ordinarias. Otras descripciones son posibles, por ejemplo, usando ecuaciones derivadas parciales.

Para el modelo dinámico, en el control del ciclo de manejo se emplea un controlador PID, donde es usado principalmente para modular el torque hacia el ICE, Fig. 2.2.

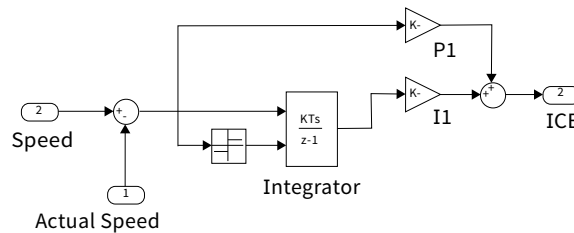


Fig. 2.2. Controlador PID del modelo dinámico del VEH.

Se define el modelo dinámico de la transmisión, es un elemento que transforma la potencia mecánica proporcionada por una fuente de potencia a cierta velocidad ω_1 y un torque τ_1 a

una velocidad diferente ω_2 y torque τ_2 , (Fig. 2.3). Despreciando las pérdidas causadas por dicho dispositivo, las siguientes relaciones son (2.5, 2.6):

$$\omega_1 = \gamma\omega_2, \quad (2.5)$$

$$\tau_2 = \gamma\tau_1. \quad (2.6)$$

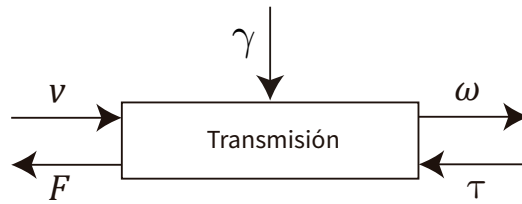


Fig. 2.3. Modelo dinámico de la transmisión.

La representación del modelo dinámico de la causalidad de un motor/generador es representada en la Fig. 2.4. Las variables de entrada son el voltaje DC $U_m(t)$ y la velocidad angular del eje del motor ω_m . Las variables de salida son el torque τ_m y la corriente intercambiada $I_m(t)$. Un valor positivo de $I_m(t)$ es absorbido por la máquina eléctrica cuando opera como motor, mientras que un valor negativo de $I_m(t)$ es entregado por la máquina cuando opera como generador.

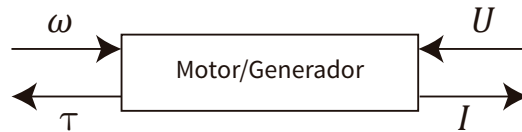


Fig. 2.4. Modelo dinámico del motor/generador.

En una arquitectura de vehículo convencional, todas las cargas eléctricas son suministradas por un enlace DC-DC conectado a una batería y a un alternador accionado por un motor.

Las cargas eléctricas más comunes son: sistema de encendido, sistema de iluminación y motor eléctrico de arranque. Los accesorios restantes generalmente son accionados por correa por el motor.

La representación de un convertidor de potencia se representa por la Fig. 2.5. Las variables de entrada son el voltaje en el primer puerto (generalmente del sistema de almacenamiento de energía), $U_1(t)$, y la corriente en los otros puertos, $I_j(t), j = 2, \dots, m + 1$. Las variables de salida son las corrientes $I_1(t)$ y el voltaje en los puertos, $U_j(t), j = 2, \dots, m + 1$.

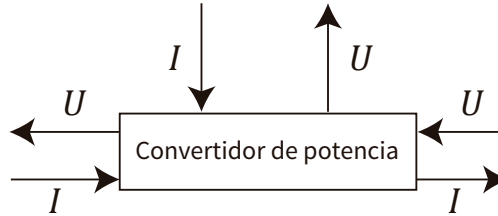


Fig. 2.5. Modelo dinámico del convertidor de potencia.

Los modelos dinámicos describen el comportamiento transitorio de una batería, incluidos la tasa de cambio de la tensión del terminal de la batería. En el circuito equivalente dinámico los modelos tienen en cuenta implícitamente los efectos capacitivos e inductivos que se desprecian en los circuitos de estado-estacionario. La representación del modelo dinámico de una batería es bosquejado en la Fig. 2.6. La entrada variable es la corriente de la terminal $I_b(t)$. Un valor positivo de $I_b(t)$ descarga la batería, mientras que un valor negativo de $I_b(t)$ lo carga. Las variables de salida son el voltaje $U_b(t)$.

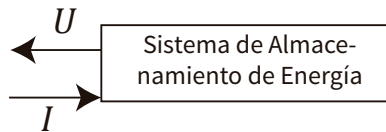


Fig. 2.6. Modelo dinámico del sistema de almacenamiento de energía.

La máquina eléctrica funciona como un generador para velocidades altas ($T_m < 0$). El modelo dinámico un generador se representa en la Fig. 2.7. Las variables de entrada son el voltaje DC $U_m(t)$ y la velocidad angular del eje del motor ω_m . Las variables de salida son el torque τ_m y la corriente $I_m(t)$.

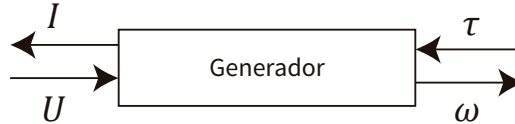


Fig. 2.7. Modelo dinámico del generador eléctrico.

El modelo dinámico del motor de combustión interna está representado por la Fig. 2.8. Donde la eficiencia se encuentra definida por 2.7:

$$\eta = \frac{\omega \tau}{P}. \quad (2.7)$$

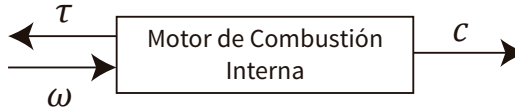


Fig. 2.8. Modelo dinámico del ICE.

Finalmente, se define el modelo dinámico de un vehículo eléctrico híbrido de topología serie, Fig. 2.9.

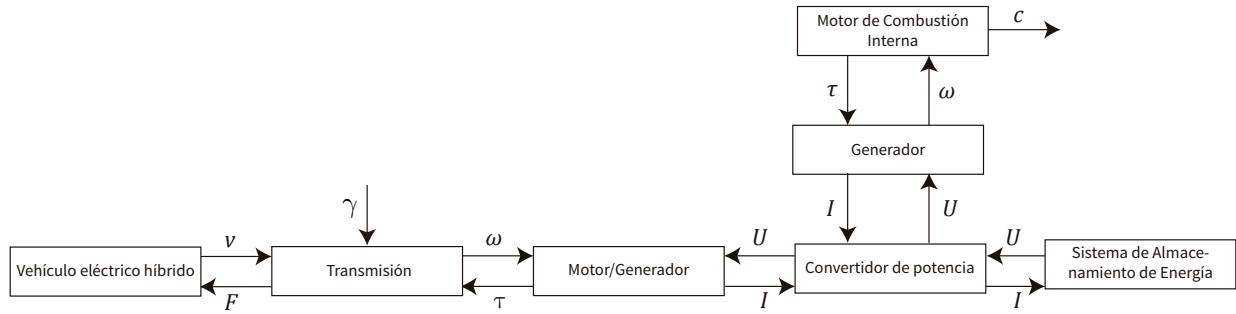


Fig. 2.9. Modelo dinámico del vehículo eléctrico híbrido.

2.3. Modelo cuasi-estático

El modelo cuasi-estático permite definir el comportamiento del tren de potencia de un vehículo eléctrico híbrido [39]. Además, el modelo proporciona un método que permite determinar el consumo de combustible a través de un ciclo de conducción [40].

Un ciclo de conducción consiste en una serie de datos, que representan la velocidad de un vehículo. Los ciclos de conducción se clasifican en ciclos modales (Fig. 2.10) y ciclos transitorios (Fig. 2.11) [41]. Sin embargo, los ciclos de conducción transitorios proveen información de las condiciones de un trayecto acotado [42, 43].

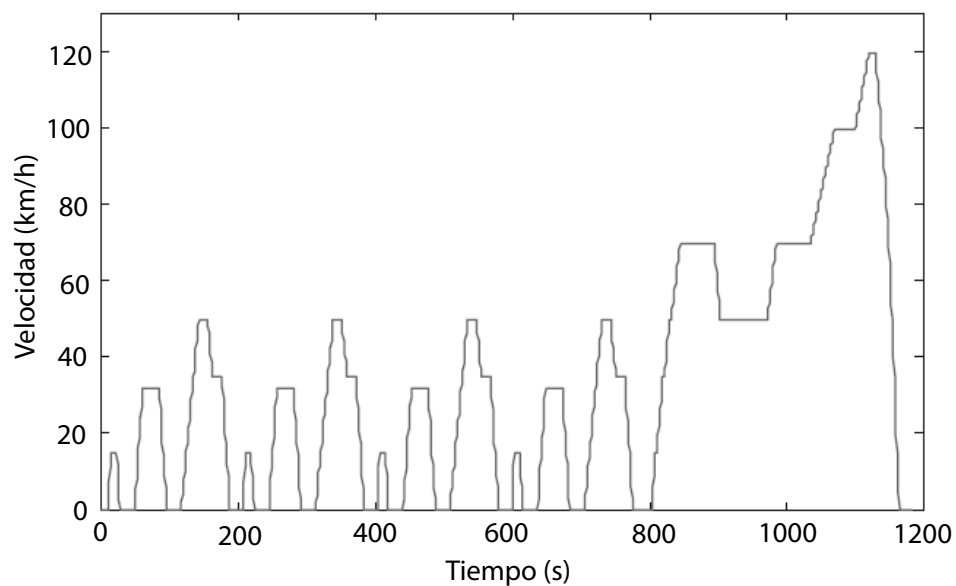


Fig. 2.10. Ciclo de conducción modal NEDC [42].

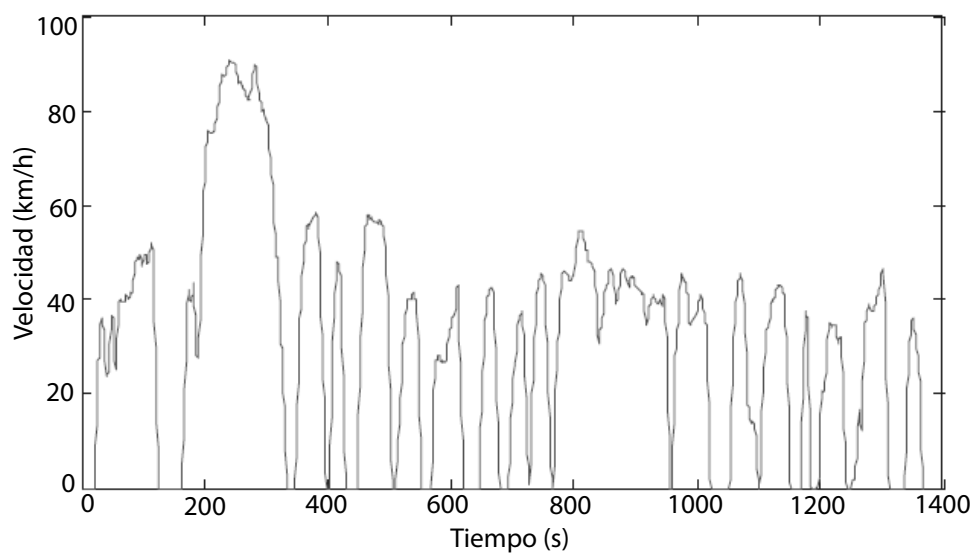


Fig. 2.11. Ciclo de conducción transitorio UDDS [43].

De acuerdo al modelo cuasi-estático, se define un bloque para el vehículo eléctrico híbrido,

Fig. 2.12, que determina el torque y la velocidad angular a partir del ciclo de conducción aplicado el modelo longitudinal.

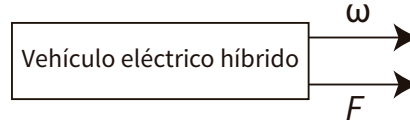


Fig. 2.12. Modelo cuasi-estático de un vehículo eléctrico híbrido.

Posteriormente, se define el modelo cuasi-estático de la transmisión, Fig. 2.13, que obtiene un torque (2.8) y una velocidad angular (2.9) a partir de la relación de engranes (γ).

$$\omega_{out} = \gamma \omega_{in}, \quad (2.8)$$

$$\tau = \gamma F. \quad (2.9)$$

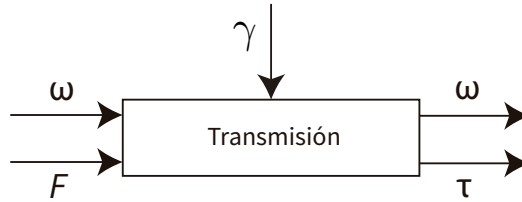


Fig. 2.13. Modelo cuasi-estático de la transmisión.

El torque y la velocidad angular que provee la transmisión se emplea para realizar un mapeo del mapa de eficiencia en el modelo cuasi-estático de un motor/generador eléctrico (2.10) Fig. 2.14.

$$P = \eta(\omega, \tau) \omega \tau. \quad (2.10)$$

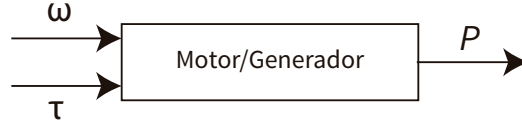


Fig. 2.14. Modelo cuasi-estático de un motor/generador eléctrico.

La potencia que provee el motor/generador ingresa a un modelo cuasi-estático de un convertidor de potencia, Fig. 2.15, que establece un conexión con alguna fuente de energía (2.11) con base a la restricción (2.12):

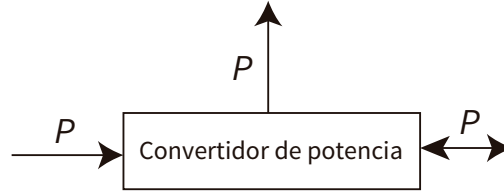


Fig. 2.15. Modelo cuasi-estático de un convertidor de potencia.

$$Fuente = \begin{cases} ESS & \text{si } (P \leq P_{ESS}) \\ ICE & \text{si } (P > P_{ESS}) \end{cases} \quad (2.11)$$

$$(P_{ESS} = P) \text{ or } (P_{ICE} = P). \quad (2.12)$$

El modelo cuasi-estático de un sistema de almacenamiento de energía, Fig. 2.16 (2.13), consiste en una potencia de entrada (P_{in}) para el almacenamiento de energía y una potencia de salida (P_{out}) para el suministro de energía.

$$P_{out} = P_{in} \left(V_{OC} - R_0 I_L - \int \left(\frac{I_L}{C_1} - \frac{V_{R_1 C_1}}{R_1 C_1} \right) dt \right). \quad (2.13)$$

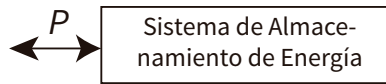


Fig. 2.16. Modelo cuasi-estático de un sistema de almacenamiento de energía.

Considerando la potencia eléctrica que provee el convertidor de potencia, el modelo cuasi-estático de un generador, Fig. 2.17, emplea el método *mirroring* para determinar el torque y la velocidad angular correspondiente.



Fig. 2.17. Modelo cuasi-estático de un sistema de un generador eléctrico.

Posteriormente, el modelo cuasi-estático del motor de combustión interna, Fig. 2.18, realiza el mapeo del torque y velocidad angular de entrada, para determinar el consumo de combustible puntual., referencia a la ecuación

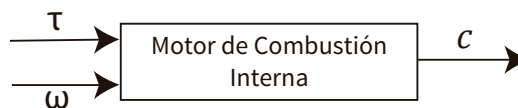


Fig. 2.18. Modelo cuasi-estático de un motor de combustión interna.

Finalmente, se define el modelo cuasi-estático de un vehículo con topología serie, Fig. 2.19, a través de la interconexión de los modelos individuales.

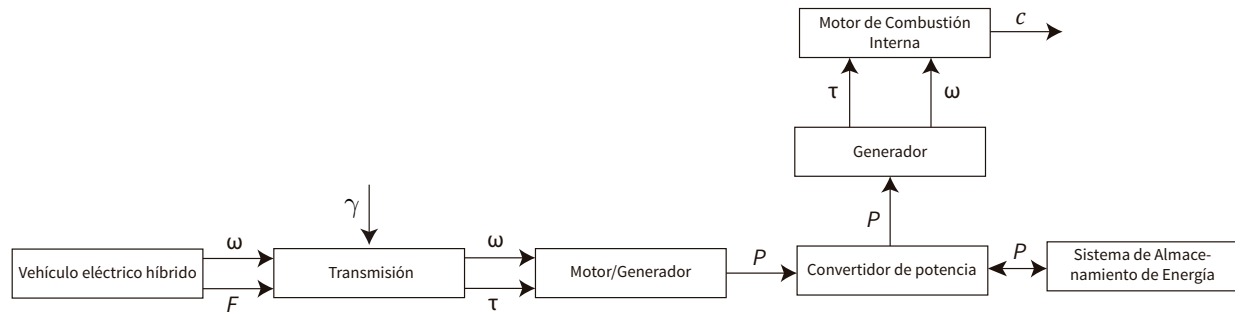


Fig. 2.19. Modelo cuasi-estático de un VEH.

En el presenta capítulo se realizó un análisis del modelo cuasi-estático, que permitirá determinar el consumo de combustible de un vehículo eléctrico híbrido, a partir de una determinada topología y un ciclo de conducción.

Capítulo 3

Diseño de la plataforma de emulación

Con base al modelo cuasi-estático de un vehículo eléctrico híbrido con un tren de potencia con topología serie, se procede a realizar el diseño de una plataforma de emulación que permita determinar el consumo de combustible a partir de parámetros reales. Debido a la complejidad que involucra la plataforma de emulación, se propone fragmentar el sistema en el diseño del sistema embebido, la comunicación y el software, Fig. 3.1.

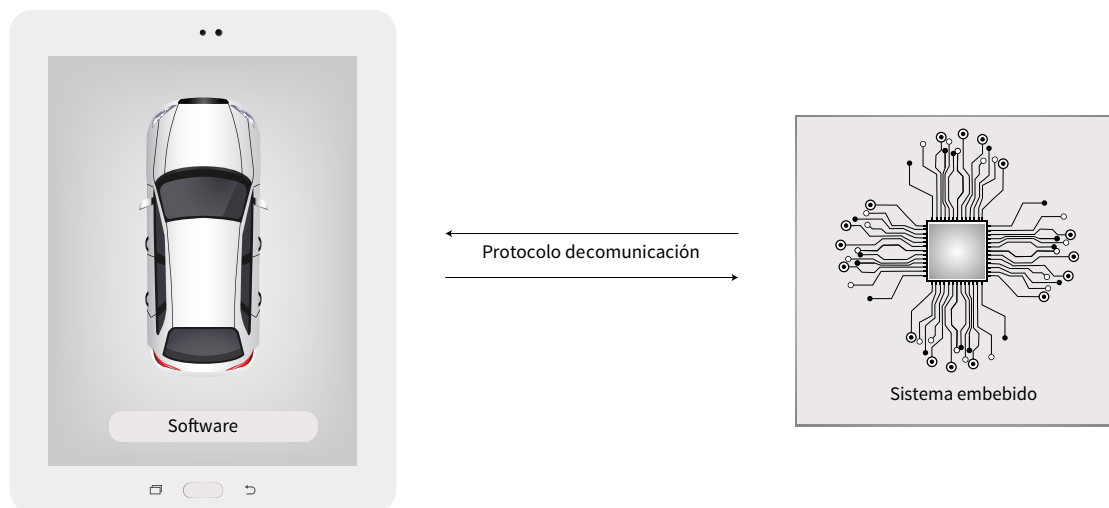


Fig. 3.1. Plataforma de emulación.

El sistema embebido almacenará el modelo matemático de cada elemento del tren de potencia de un vehículo eléctrico híbrido, que serán interconectados con base al modelo cuasi-estático de una topología serie. Además, realizará la emulación del vehículo eléctrico híbrido con el fin de calcular el consumo de combustible a partir de un ciclo de conducción.

Posteriormente, se diseñará un protocolo con el fin de establecer una comunicación entre el software y el sistema embebido, para realizar la configuración de los parámetros del vehículo eléctrico híbrido.

Finalmente, se diseñará un software a través de un patrón de arquitectura de software denominado modelo-vista-controlador (MVC), que será ejecutado en un dispositivo móvil.

3.1. Desarrollo del sistema embebido

Un sistema embebido se define como un sistema electrónico dedicado, que permite ejecutar diversas funciones específicas en tiempo real. Actualmente, se emplean dispositivos programables como microcontroladores o FPGAs (Field Programmable Gate Array), que proveen diversos recursos para el desarrollo de sistemas. Además, un dispositivo programable provee una elevada capacidad de procesamiento y un reducido consumo de energía, lo que permite el desarrollo de sistemas embebidos para una infinidad de propósitos.

La elección del dispositivo programable, dependerá de los requerimientos que deberá cumplir el sistema. El modelo cuasi-estático requiere una ejecución secuencial del modelo matemático de cada elemento del tren de potencia, por ello, se ha decidido emplear un microcontrolador para el diseño del sistema embebido.

Un microcontrolador provee un sistema mínimo formado por un microprocesador, memoria, puestos de entrada/salida (E/S) y periféricos. Además, el microprocesador posee una arquitectura Harvard, Fig. 3.2, que consiste en separar la memoria del programa en una memoria tipo flash y la memoria de datos en una memoria tipo SRAM (Static Random Access Memory).

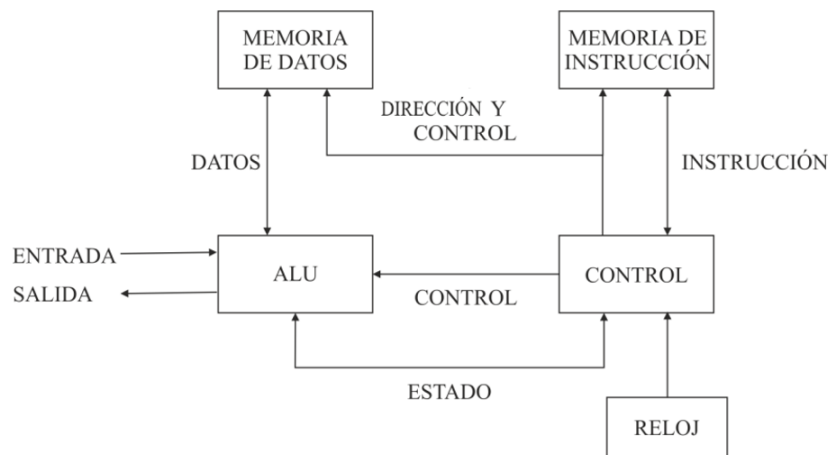


Fig. 3.2. Arquitectura Harvard en un microcontrolador.

Además, la arquitectura del microprocesador permite reducir el tiempo de lectura y escritura a la memoria, lo que incrementa el rendimiento del sistema embebido. Asimismo, la estructura de la arquitectura proporciona la posibilidad de diseñar dispositivos con tamaño reducido, debido a la capacidad de direccionar el acceso a las instrucciones y a los datos de manera independiente.

Debido al procesamiento requerido por el modelo cuasi-estático, se propone emplear el microcontrolador MZ2048ECM144 de la familia PIC32 de la empresa *Microchip*[®]. El microcontrolador provee un microprocesador MIPS32 que opera a una velocidad de reloj de 200MHz, una memoria tipo flash de 2048kB para la memoria del programa y una memoria tipo SRAM de 512kB para los datos. Además, el microcontrolador proporciona módulos DSP (Digital Signal Processor) y un módulo FPU (Floating Point Unit), que permitirán realizar las operaciones de precisión simple y doble que requiere el modelo cuasi-estático.

El microcontrolador requiere de un circuito mínimo, Fig. 3.3, que contempla la alimentación y el circuito de reset.

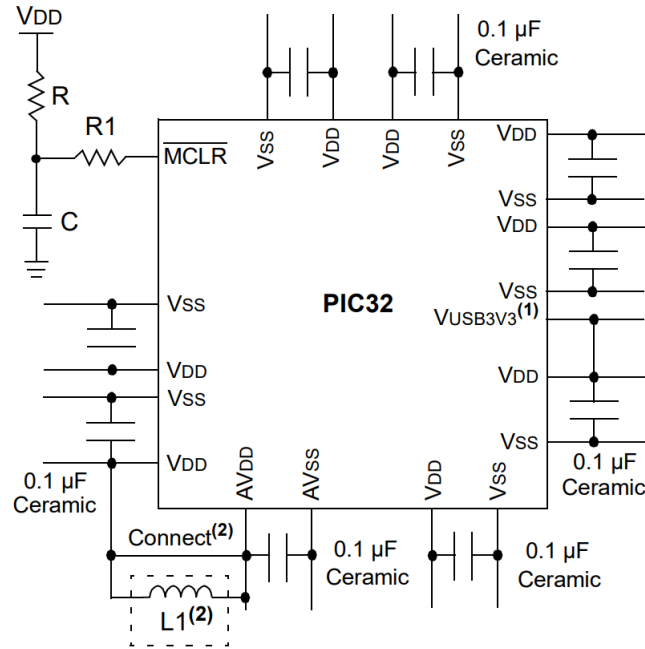


Fig. 3.3. Circuito mínimo para el microcontrolador PIC32MZ2048EFM144.

La alimentación requiere de capacitores en modo bulk para mejorar la estabilidad de la fuente de energía. Por otra parte, el capacitor C requiere ser ajustado para evitar reinicios involuntarios debido a fallas breves.

Con base a los requerimientos del circuito mínimo del microcontrolador, se procede a realizar el diagrama esquemático en un software destinado al diseño asistido por computadora (Computer Aided Design, CAD).

El diagrama esquemático del circuito mínimo del sistema embebido, contempla la alimentación, el circuito de reset y la colocación de unos pines para el protocolo de programación ICSP (In-Circuit Serial Programming).

Posteriormente, se procede a realizar el diseño de la placa de circuito impreso (Print Circuit Board, PCB) a partir del diagrama esquemático, considerando un encapsulado TQFP para el microcontrolador.

Después de realizar el diseño de la placa de circuito impreso para la plataforma de emulación de un vehículo eléctrico híbrido, se procede a modelar el firmware a través del lenguaje de modelado unificado (Unified Modeling Language, UML). Debido a la complejidad del sistema se propone un diagrama UML, que contempla descomponer el firmware de la plataforma en un sistema base y un protocolo de comunicación, Fig. 3.4.

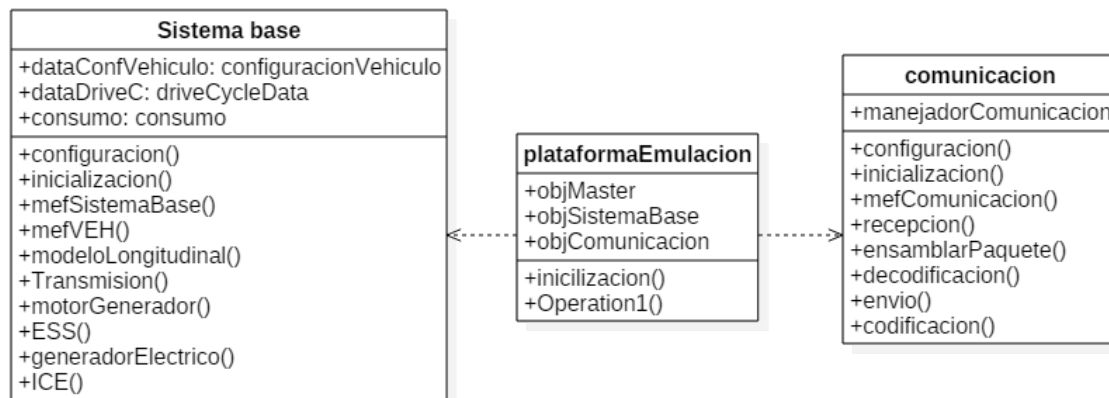


Fig. 3.4. Diagrama UML del firmware.

Posteriormente, se propone un diagrama UML para el diseño del sistema base, Fig. 3.5, que representa las funciones que se implementarán para la emulación del vehículo eléctrico híbrido.

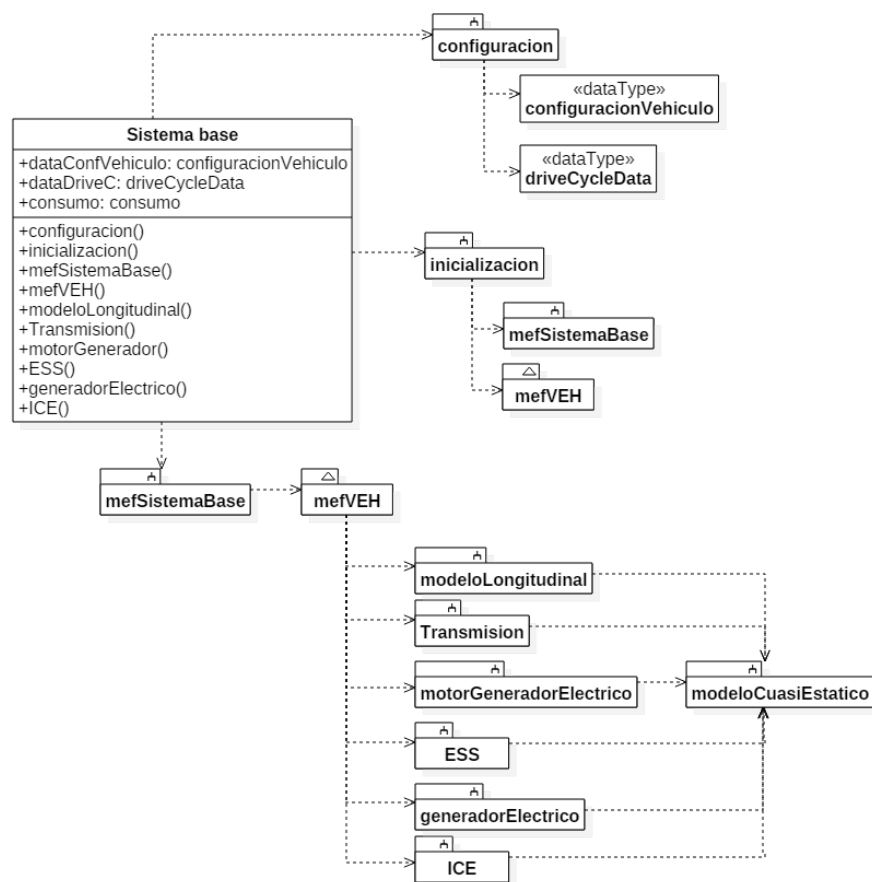


Fig. 3.5. Diagrama UML del sistema base.

El sistema base consta de 4 funciones principales, que son gestionadas a través de una máquina de estados finitos, Fig. 3.6.

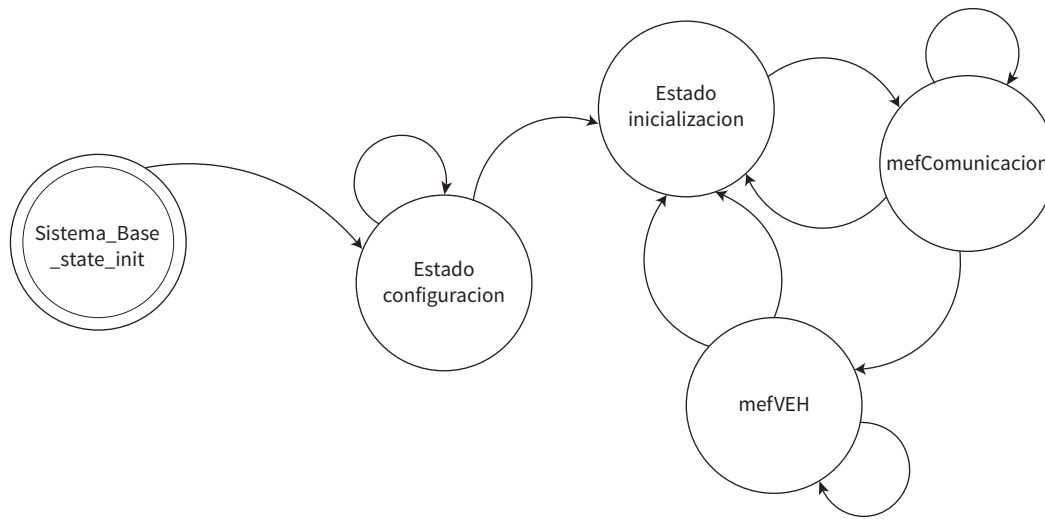


Fig. 3.6. Máquina de estados finitos del sistema base.

Además, el sistema base contempla un estado de configuración destinado a definir características del microcontrolador y un estado para realizar la inicialización de las máquinas de estado.

Debido a la característica secuencial del modelo cuasi-estático de un vehículo eléctrico híbrido, se propone implementar cada elemento del tren de potencia dentro de una máquina de estados finitos, Fig. 3.7.

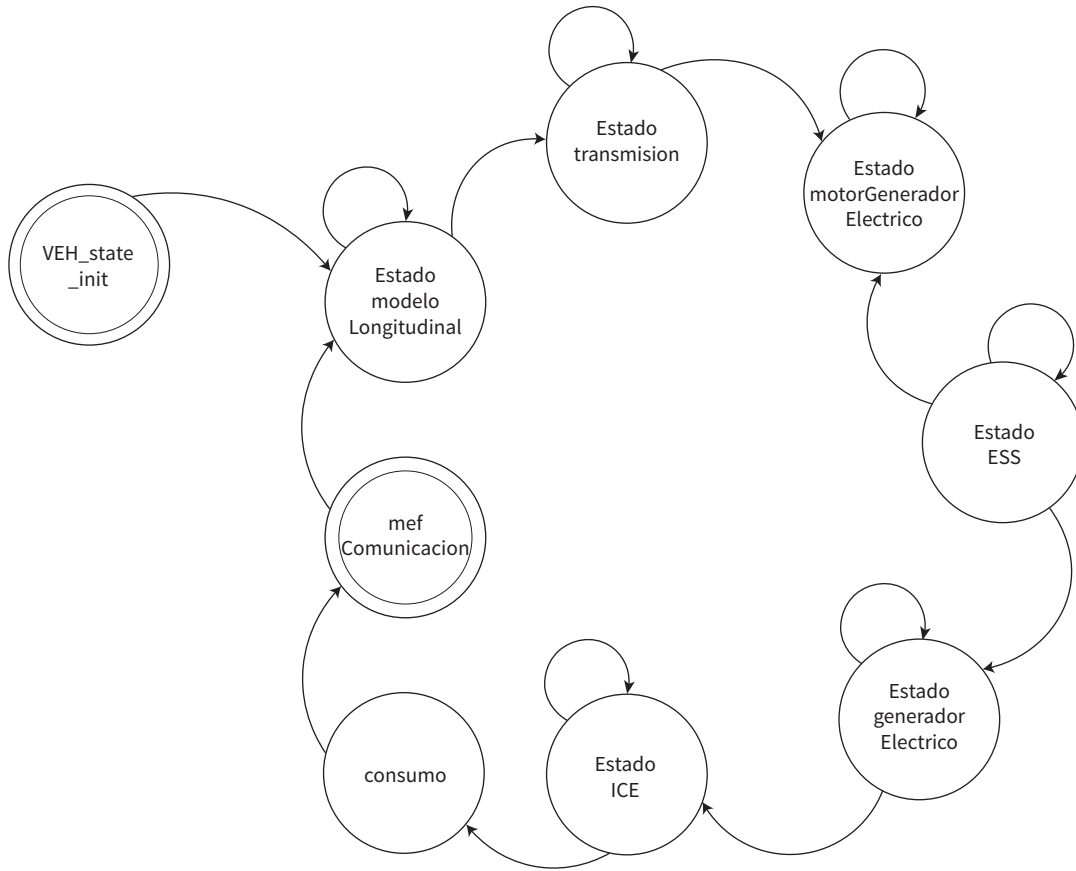


Fig. 3.7. Máquina de estados finitos del modelo cuasi-estático.

Algunos elementos del tren de potencia han sido descritos a través de un modelo matemático que requiere realizar la implementación de una función derivada y una función integral. Por ello, para realizar la derivada se propone emplear el método denominado *valor medio* y el método denominado *cociente de diferencia simétrica*.

La método del valor medio (3.1), permite determinar la derivada a partir del valor de las muestras adyacentes del punto de interés.

$$f'(x) = \frac{f(b) - f(a)}{b - a}, \quad a < x < b. \quad (3.1)$$

Por otro lado, el método del cociente de diferencia simétrica (3.2), permite determinar la

derivada a partir de la frecuencia de muestreo del sistema.

$$f'(x) = f(x + h) - \frac{f(x - h)}{2 * h}. \quad (3.2)$$

Para la función integral se propone emplear el método denominado *regla trapezoidal* (3.3), que realiza una aproximación de la integral a partir de una interpolación lineal.

$$F(x) = (b - a) \cdot \frac{f(a) + f(b)}{2}, \quad a < x < b. \quad (3.3)$$

Además, el método de la regla trapezoidal se caracteriza por establecer una función de grado uno, lo que permite lograr una reducción en el procesamiento de la función.

3.2. Diseño del protocolo de comunicación

Con el fin de establecer una comunicación entre el software y el sistema embebido, se propone un diagrama UML para la implementación de un protocolo de comunicación, Fig. 3.8.

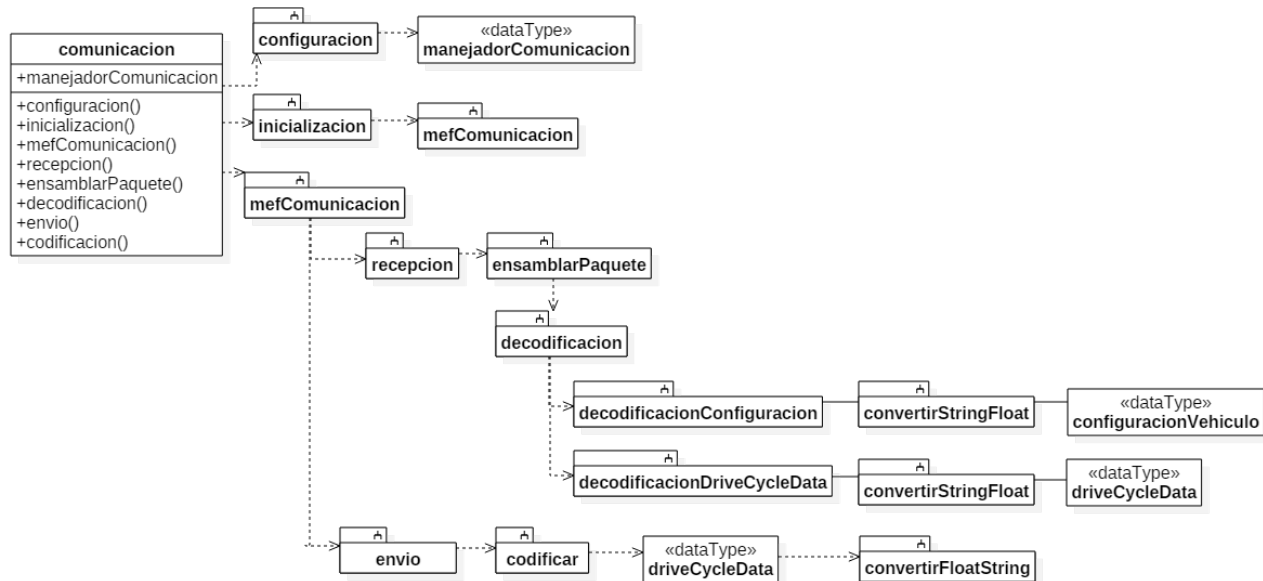


Fig. 3.8. Diagrama UML del protocolo de comunicación.

El protocolo de comunicación empleará una interfáz RS232, que realiza una recepción serial de un paquete de 8 bits. Por ello, se ha diseñado un algoritmo, Fig. 3.9, que permite obtener datos reconstruidos a partir de la decodificación de datos recibidos a través de la interfáz.

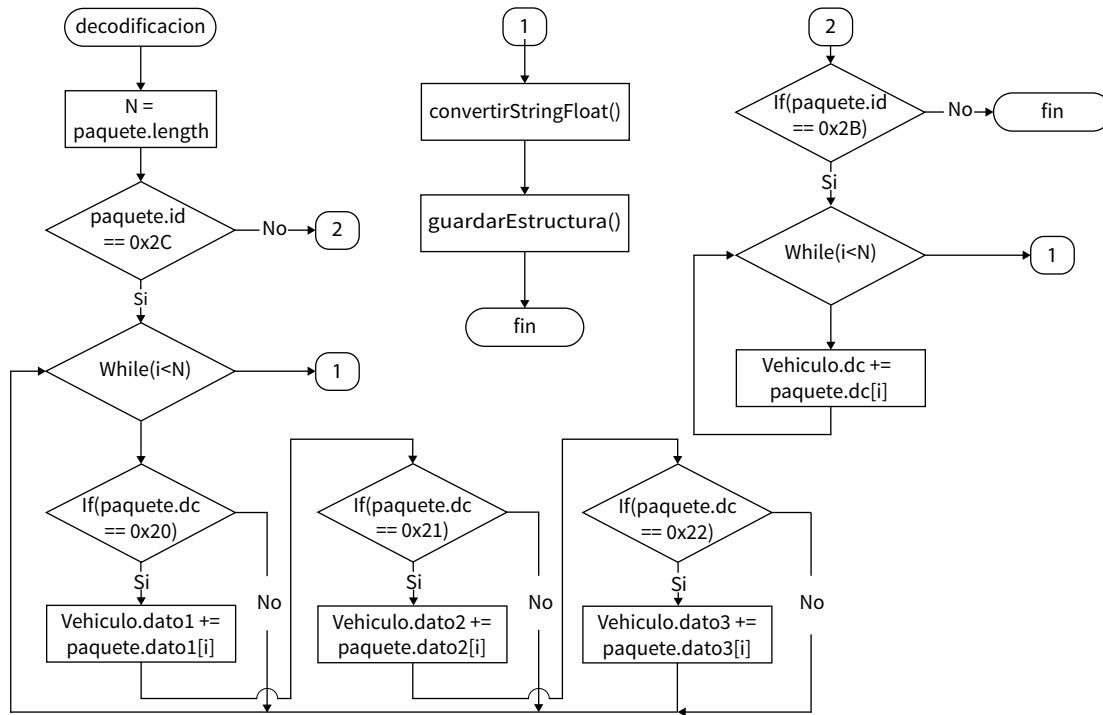


Fig. 3.9. Algoritmo para la decodificación de la interfáz RS232.

Además, algunos datos deben transformarse para ser procesados posteriormente, por ello, se propone un algoritmo que obtiene un dato de tipo flotante a partir de un dato decodificado, Fig. 3.10.

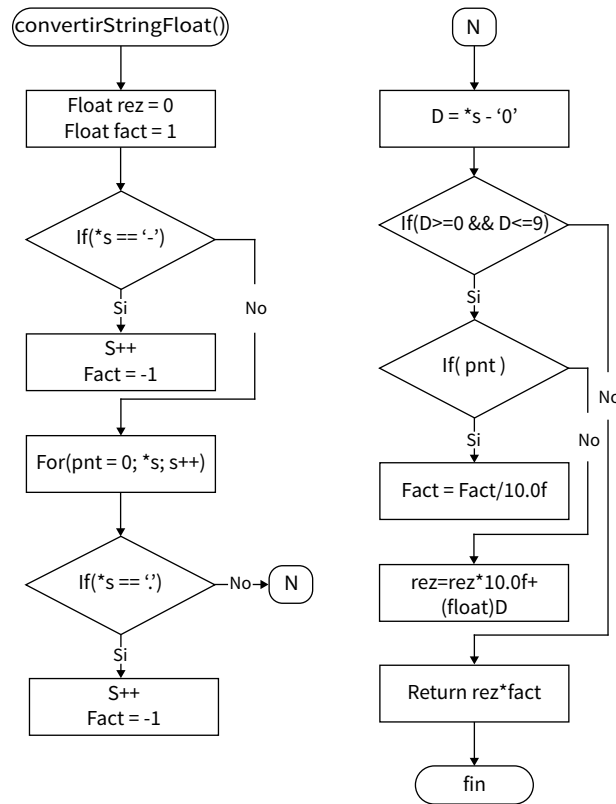


Fig. 3.10. Algoritmo para la transformación de datos.

Posteriormente, se procede a evaluar la información recibida en la interfáz, para determinar si existen parámetros de configuración del vehículo eléctrico híbrido. Se ha diseñado una estructura de datos, Fig. 3.11, con el fin de proveer un modo eficiente para acceder a los parámetros de configuración.

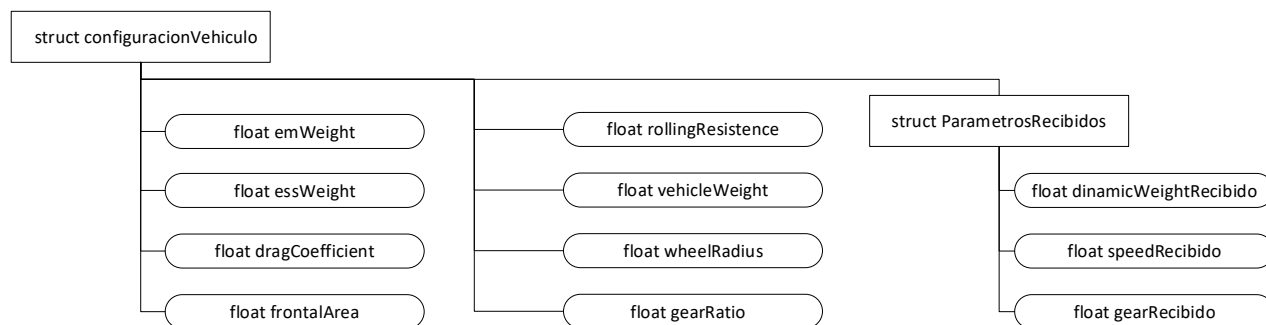


Fig. 3.11. Estructura de datos de configuración.

3.3. Software para la plataforma de emulación

La plataforma de emulación de un vehículo eléctrico híbrido requiere de la implementación de software, que permita realizar la configuración de los elementos del tren de potencia. Por ello, se propone el diseño de un diagrama UML, Fig. 3.12, para modelar la etapa de software.

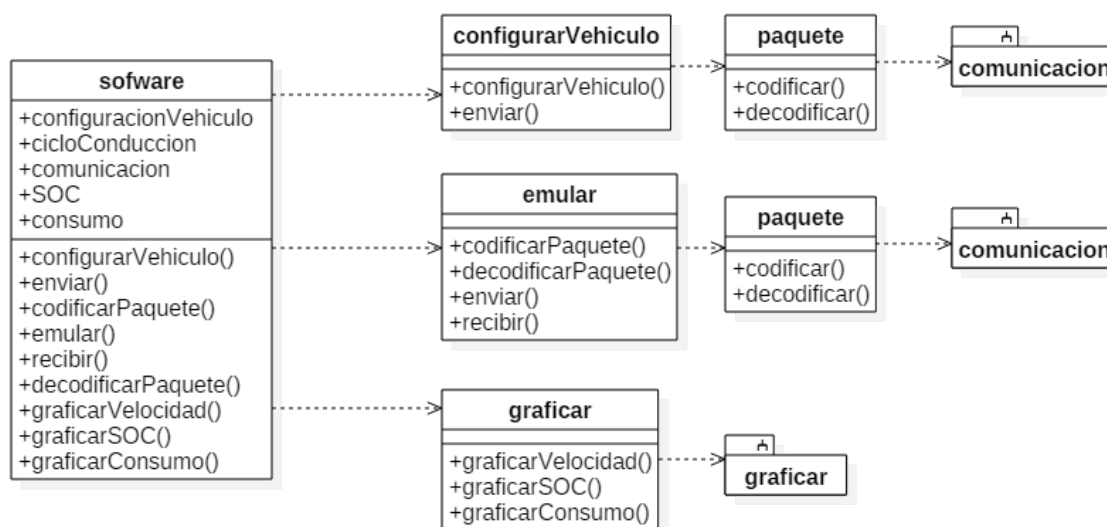


Fig. 3.12. Diagrama UML del software.

Debido a la interacción de un usuario con el software, se propone emplear un patrón de arquitectura de software denominada *Modelo-Vista-Controlador (MVC)*.

El modelo MVC permite separar los datos de la lógica de negocio de una aplicación de interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. El patrón MVC propone la construcción de tres componentes: el modelo, la vista y el controlador. Además, el patrón permite la reutilización de código y separación de conceptos, con el fin de reducir el tiempo de desarrollo de aplicaciones, facilitar el mantenimiento y la reutilización de las funcionalidades de la aplicación.

Con base al diagrama UML del software, se procede a definir el modelo MVC, Fig. 3.13.

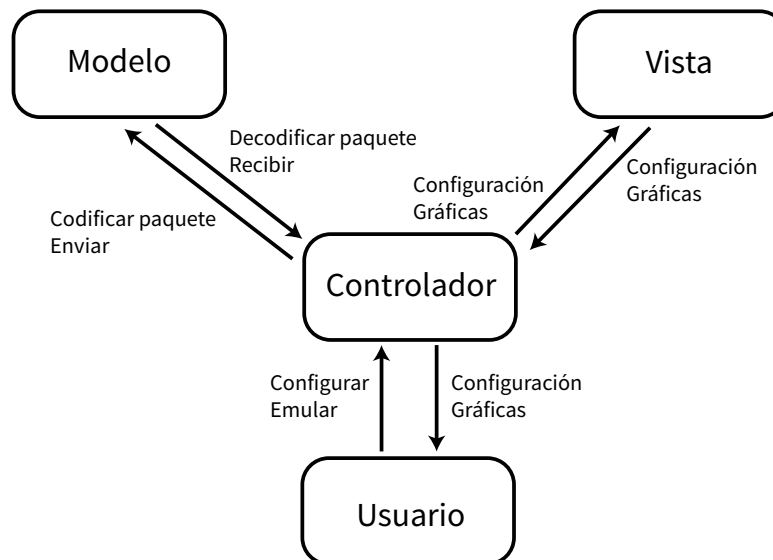


Fig. 3.13. Patrón MVC del software.

El modelo del patrón MVC, Fig. 3.14, representa los datos que serán almacenados y gestionados por el controlador.

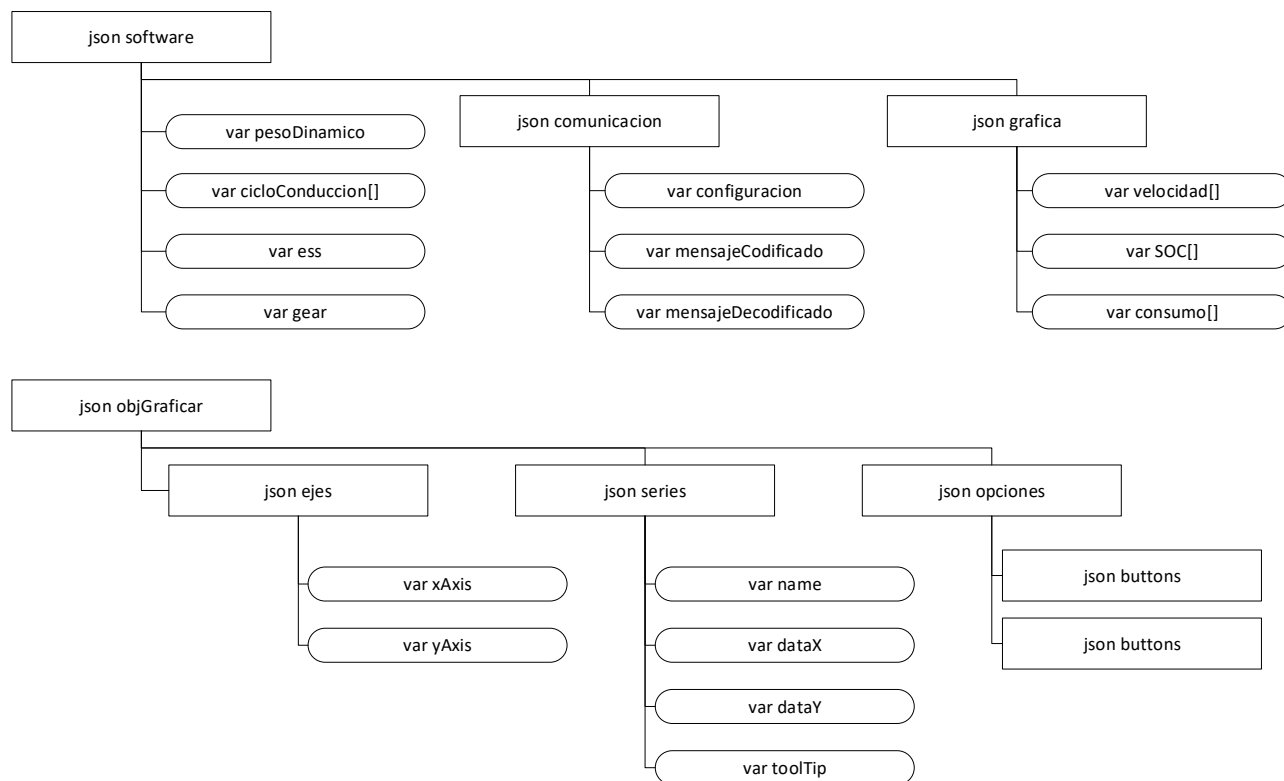


Fig. 3.14. Modelo del patrón MVC.

Por otra parte, el controlador del patrón MVC, Fig. 3.15, realiza la gestión y procesamiento de los datos. Además, realiza la gestión de la vista, que son presentadas al usuario del sistema.

CONFIGURACIÓN		COMUNICACIÓN	
<pre>configurarVehiculo: function(json vehicle){ { emular: function(json packageData, function codificar() {</pre>		<pre>comunicacion.onopen = function () {= comunicacion.onmessage = function (event) {= comunicacion.onerror = function () {= comunicacion.onclose = function (event) {= comunicacion.registerReadCallback(function success(packageDataIn, function decodificar(){= function error(event.error){= });</pre>	
VISUALIZACIÓN			
<pre>graficaVelocidad: function() { charts.stockChart('container_velocidad', { rangeSelector: {= title: {= series: [{= }); }</pre>	<pre>graficaSOC: function() { charts.stockChart('container_SOC', { rangeSelector: {= title: {= series: [{= }); }</pre>	<pre>graficaConsumo: function() { charts.stockChart('container_consumo', { rangeSelector: {= title: {= series: [{= }); }</pre>	

Fig. 3.15. Controlador del patrón MVC.

Finalmente, se procede a realizar la implementación del diseño del sistema embebido, el protocolo de comunicación y el software. Además, se realizará una validación del sistema de emulación de un vehículo eléctrico híbrido a través de un modelo cuasi-estático desarrollado en simulink.

En el presente capítulo se realizó el diseño la plataforma de emulación de un vehículo eléctrico híbrido, considerando la etapa del sistema embebido, el protocolo de comunicación y el software.

Capítulo 4

Implementación de la plataforma de emulación

Después de realizar el diseño de la plataforma de emulación de un vehículo eléctrico híbrido, se procede a realizar la implementación del sistema embebido, el protocolo de comunicación y el software. Con el objetivo de realizar la validación de la plataforma, se realizó la simulación del modelo cuasi-estático a través del software Simulink.

4.1. Construcción y pruebas de funcionamiento de la plataforma de emulación

Con el objetivo de implementar el modelo cuasi-estático de un vehículo eléctrico híbrido, se realizó la fabricación de la placa de circuito impreso, Fig. [4.1](#).

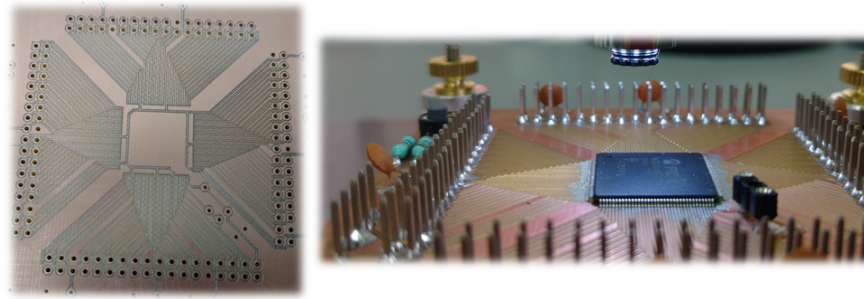


Fig. 4.1. Placa de circuito impreso del sistema embebido.

Posteriormente, empleando el entorno de desarrollo del microcontrolador PIC2MZ2048ECM144, se procede a realizar la configuración de la fuente de reloj para el sistema base.

Se contempla emplear un oscilador interno y un lazo de seguimiento de fase (Phase-Locked Loop, PLL) para definir una frecuencia de reloj de 200MHz para el microprocesador y 100MHz para los módulos, que serán empleados para el protocolo de comunicación.

Posteriormente, se implementó un manejador principal para realizar la instanciación del objeto manejador del sistema, que tendrá la función de controlar la máquina de estados del sistema base (Fig. 3.6).

Después de realizar la implementación de la máquina de estados, se procede a inicializar el puerto de comunicación y definir un manejador del módulo USART (Universal Asynchronous Receiver-Transmitter). A través del manejador se obtiene un acceso al buffer, datos y la configuración del módulo. La configuración del módulo se realizó de acuerdo a los parámetros de la Tabla 4.1.

Tabla. 4.1. Configuración del módulo USART

USART Driver Instance 0	
Driver implementation	Dinamic
USART Module ID	USART_ID_1
Baud Rate	230400
Operation mode	DRV_USART_OPERATION_MODE_NORMAL
Baud Rate Generator Clock	100000000
Line control	DRV_USART_LINE_CONTROL_8NONE1
Handshake mode	DRV_USART_HANDSHAKE_NONE
Power state	SYS_MODULE_POWER_RUN_FULL

Además, se realizó la implementación de un manejador para realizar una instancia de un módulo denominado timer. El manejador permitirá el acceso a un registro de tiempo, que permitirá determinar tiempo de ejecución del modelo cuasi-estático dentro del sistema embebido. La configuración del módulo timer se realizó de acuerdo a los parámetros de la Tabla 4.2.

Se realizó la inicialización de las interrupciones del sistema, con el fin de establecer una interrupción destinada al módulo timer. La interrupción tendrá la función de permitir el acceso al registro de timer, que es empleado para realizar el cálculo de tiempo de procesamiento interno en el sistema embebido.

Posteriormente, se realizó la definición de las variables del modelo cuasi-estático de un vehículo eléctrico híbrido y la implementación de las máquinas de estado. Con base al diseño propuesto, el sistema se encuentra monitoreando el buffer de recepción del módulo USART, que tendrá una comunicación con el software implementado.

Con base al modelo MVC diseñado, se realizó la implementación de la vista de configuración del vehículo eléctrico híbrido, Fig. 4.2.

Tabla. 4.2. Configuración módulo timer

TMR Driver Instance 0	
Driver implementation	Static
Timer Module ID	TMR_ID_2
Interrupt priority	INT_PRIORITY_LEVEL4
Clock Source	DRV_TMR_CLKSOURCE_INTERNAL
Prescale	TMR_PRESCALE_VALUE_256
Operation mode	DRV_TMR_OPERATION_MODE_32_BIT
Time period	47

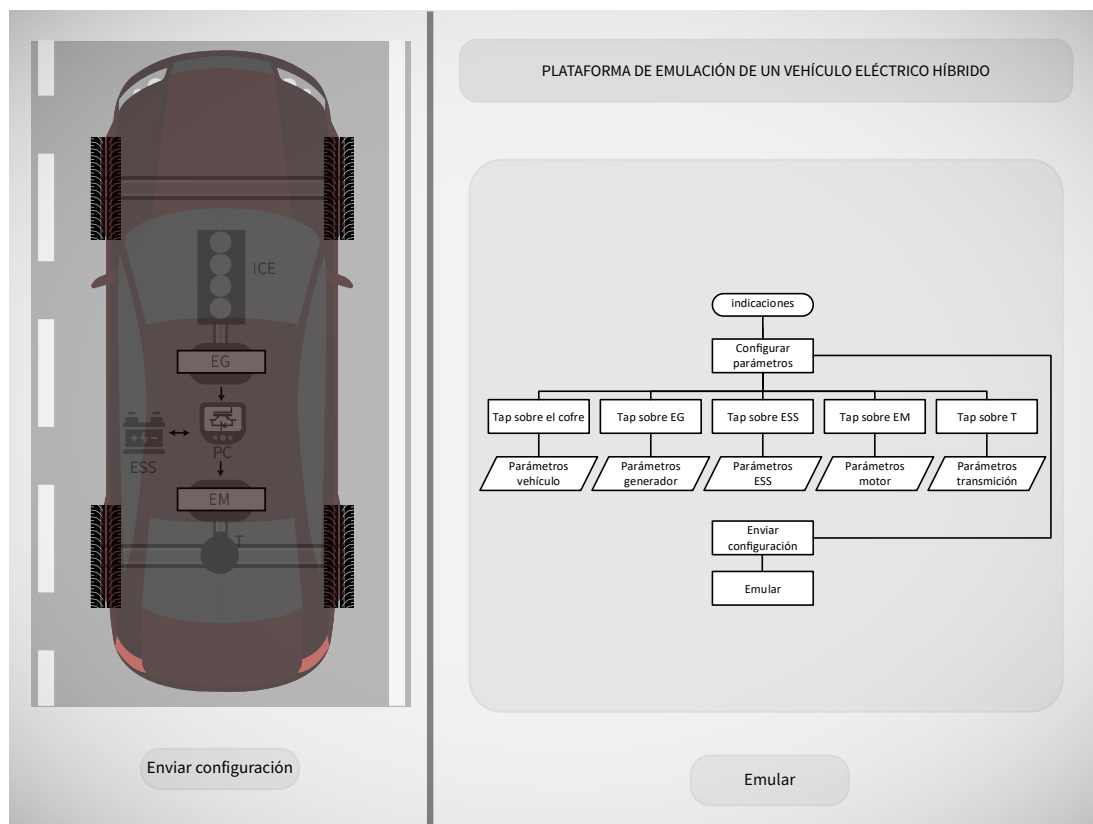


Fig. 4.2. Vista para la configuración de los parámetros del tren de potencia.

Debido a la necesidad de establecer diversos parámetros en los elementos del tren de potencia de un vehículo eléctrico híbrido, se realizó la implementación de una vista para los parámetros del vehículo.

Por otra parte, se implementó una vista para los parámetros de la transmisión, que consiste en definir la relación de engranes, considerando una caja manual de 5 velocidades.

Considerando un tren de potencia de un vehículo eléctrico híbrido con topología serie, se realizó la implementación de una vista para definir los parámetros del motor eléctrico y el generador eléctrico.

Posteriormente, se realizó la implementación de una vista para la configuración de los parámetros del sistema de almacenamiento de energía, donde es posible definir la carga máxima y descarga mínima.

Además, se realizó la implementación de una vista para la configuración del motor de combustión interna, que consiste en definir la potencia de ralenti.

Al realizar la emulación del tren de potencia de un vehículo eléctrico híbrido, el sistema base envía los resultados a través del protocolo de comunicación para ser visualizados en el software. Debido a ello, se realizó la implementación de una vista para los resultados de la emulación, Fig. 4.3, que permite observar la velocidad del ciclo de conducción empleado, el estado de carga del sistema de almacenamiento de energía y el consumo de combustible.

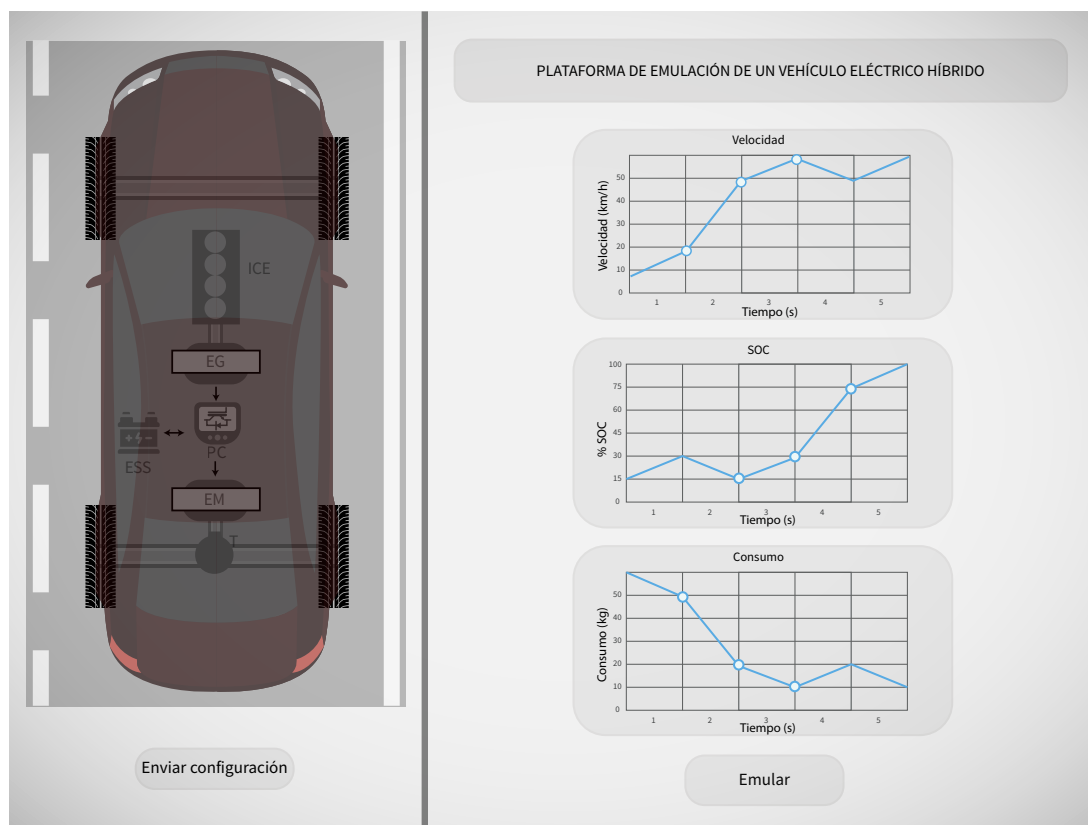


Fig. 4.3. Resultado de la emulación de un vehículo eléctrico híbrido.

4.2. Simulación del vehículo híbrido en Simulink

Con el objetivo de validar la plataforma de emulación de un vehículo eléctrico híbrido se realizó el modelo cuasi-estático en el simulador *Simulink*, Fig. 4.4.

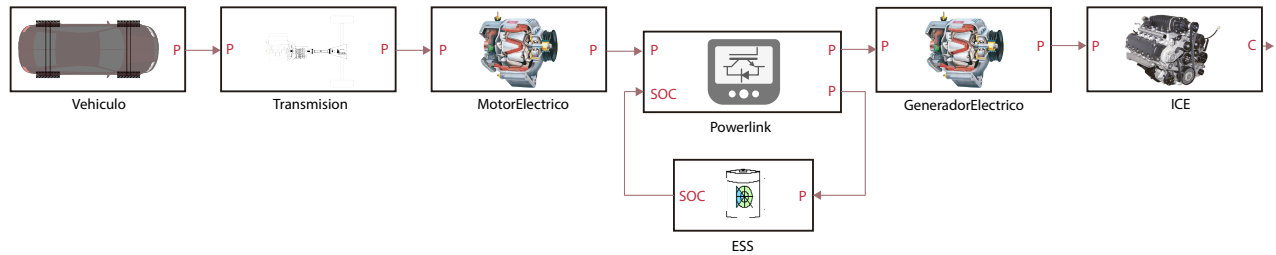


Fig. 4.4. Modelo cuasi-estático en simulink.

Los parámetros empleados en el modelo cuasi-estático, Tabla 4.3, permitirán determinar el comportamiento del tren de potencia de un vehículo eléctrico híbrido.

Tabla. 4.3. Parámetros del vehículo híbrido real para el modelo cuasi-estático

Peso motor eléctrico	100kg	Peso del ESS	150kg
Coeficiente de arrastre	0.621	Área frontal	7.5m ²
Resistencia de rodadura	0.009	Peso del vehículo	1500kg
Radio de las llantas	0.53775m	Potencia de ralenti	29321.5314W
Eficiencia del generador	90	Eficiencia del motor	93
Capacidad de la batería	2Ahr	Voltage nominal	360V
Celdas en serie	100	Límite superior de SOC	80 %
Límite inferior de SOC	20 %	SOC inicial	80 %
Celdas en serie	100	LHV	43400000J/kg
Radio de gear	[4.25 1.54 1 0.75 0.65 5]		

Además, se propone emplear un ciclo de conducción transitorio (Fig. 4.5), para determinar el consumo de combustible.

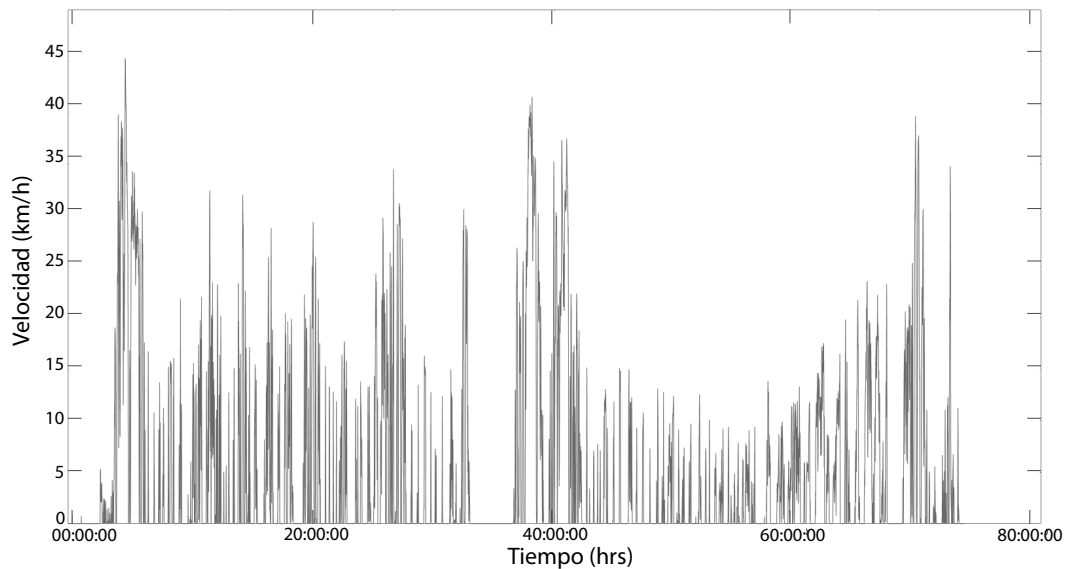


Fig. 4.5. Ciclo de conducción.

Se realizó la simulación del vehículo eléctrico híbrido, considerando el solver denominado ode45.

Con base al ciclo de conducción, se realizó la simulación de la dinámica longitudinal del vehículo eléctrico híbrido empleando el ciclo de conducción (Apéndice II), con el fin de obtener la fuerza (Fig. 4.6) y la velocidad angular (Fig. 4.7).

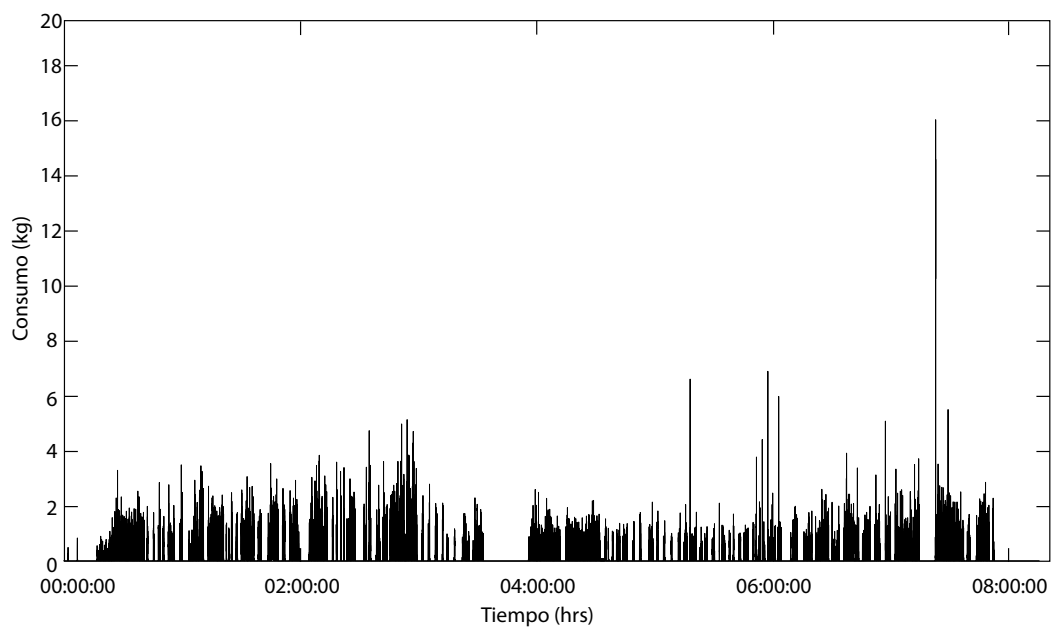


Fig. 4.6. Fuerza de salida de la dinámica longitudinal del vehículo.

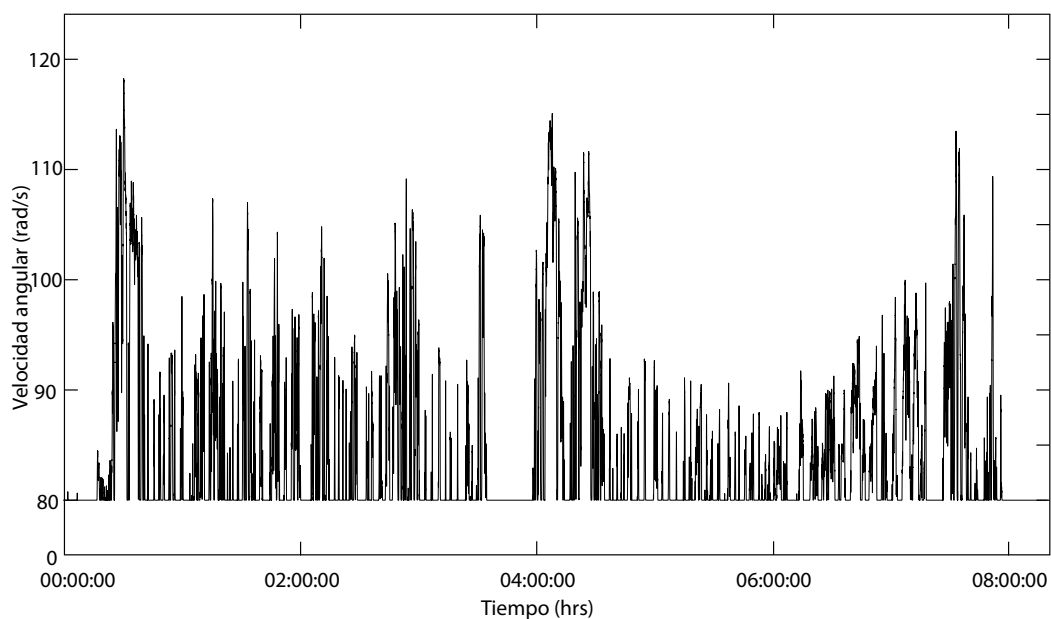


Fig. 4.7. Velocidad angular de salida de la dinámica longitudinal del vehículo.

Posteriormente, se utilizó la fuerza y velocidad angular del vehículo eléctrico híbrido en el modelo cuasi-estático de la transmisión (Apéndice II), para determinar una potencia de salida (Fig. 4.8).

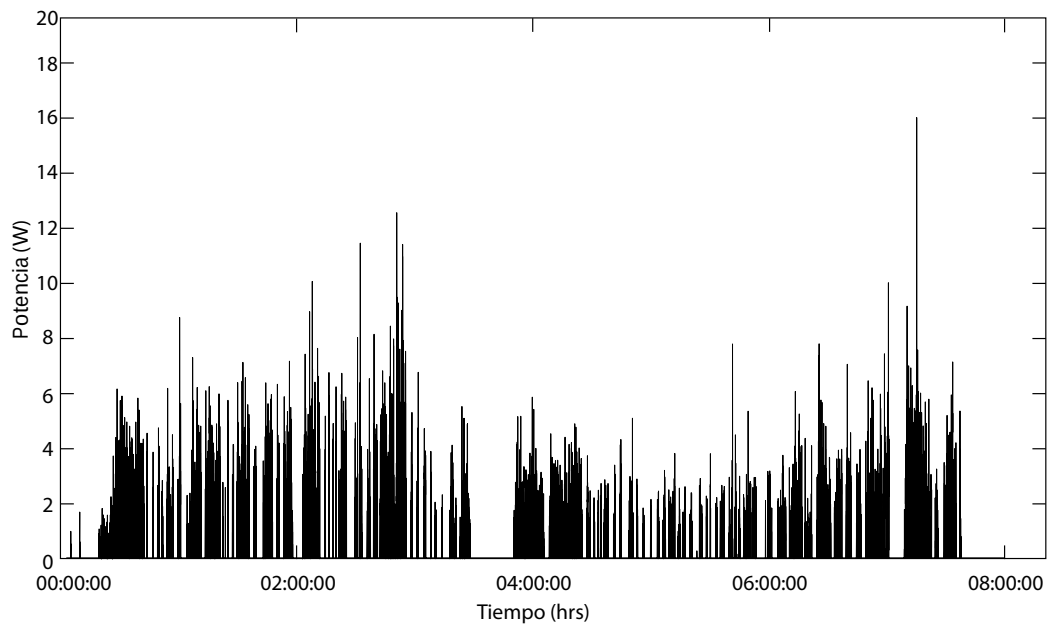


Fig. 4.8. Simulación de la potencia de salida de la transmisión.

Se utilizó la potencia de salida de la transmisión para simular el comportamiento del motor eléctrico (Apéndice II), y determinar la potencia de salida (Fig. 4.9).

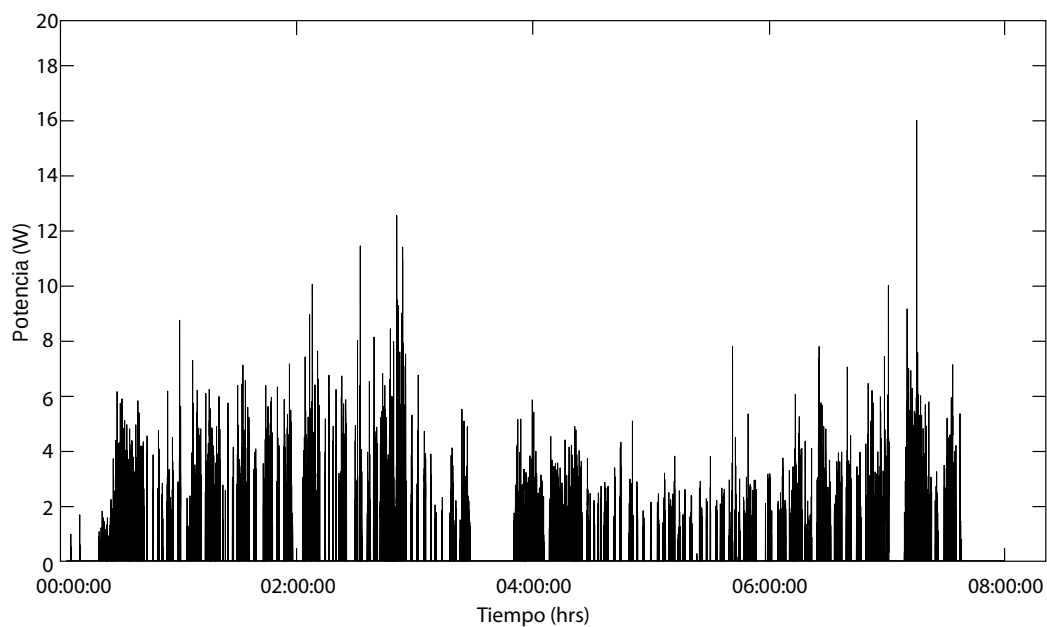


Fig. 4.9. Simulación del convertidor de potencia hacia el generador.

Considerando el funcionamiento del convertidor de potencia (Ec. 1.9) (Apéndice II), se determinó el comportamiento del estado de carga del sistema de almacenamiento de energía (Fig. 4.10) (Apéndice II), logrando un estado de carga final de 29.907 %.

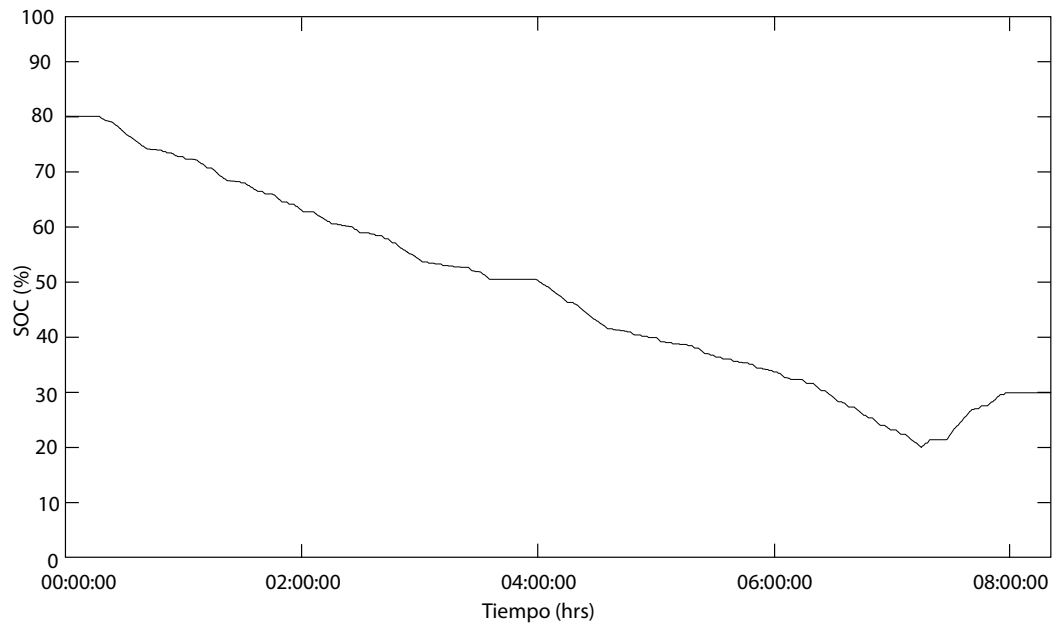


Fig. 4.10. SOC del sistema de almacenamiento de energía (Simulación).

Durante los estados que el sistema de almacenamiento de energía no fue capaz de suministrar energía al tren de potencia, el convertidor de potencia empleará el motor de combustión interna como fuente de energía. Por ello, al realizar la simulación del modelo cuasi-estático de un vehículo eléctrico híbrido se logró determinar el comportamiento del consumo de combustible (Fig. 4.11) (Apéndice II), alcanzando un consumo final de combustible de 19.438kg/s.

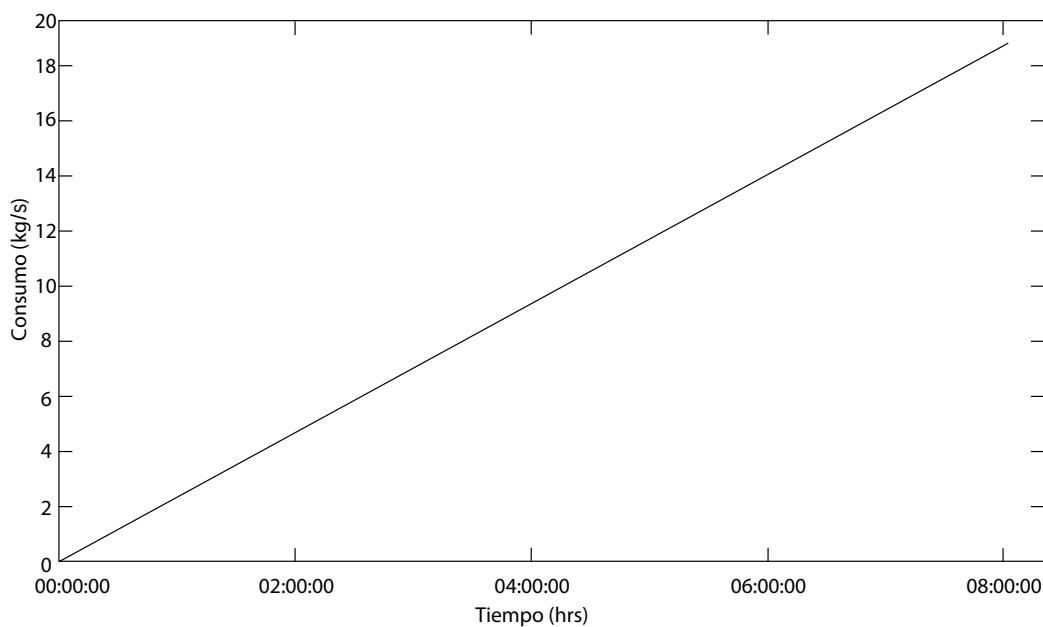


Fig. 4.11. Consumo de combustible (Simulación).

Por otra parte, se definieron los parámetros y el ciclo de conducción en la plataforma, posteriormente, se realizó la emulación del modelo cuasi-estático del tren de potencia de un vehículo eléctrico híbrido empleando la derivada del valor medio (Fig. 4.12) y la derivada del cociente de diferencia simétrica (Fig. 4.13).

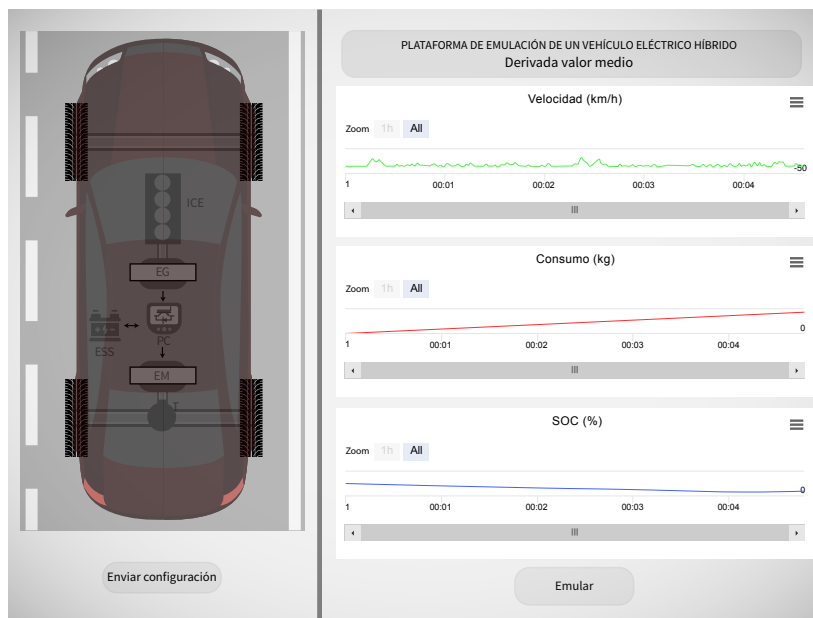


Fig. 4.12. Emulación a través de la derivada del valor medio.

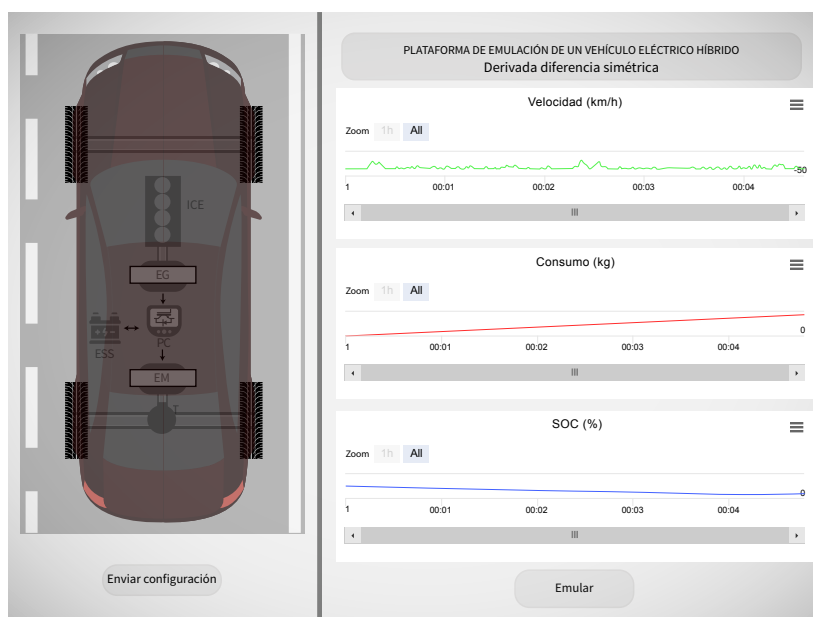


Fig. 4.13. Emulación a través de la derivada del cociente de diferencia simétrica.

La emulación a través de la derivada del valor medio se realizó en 5.2109 segundos, logrando un estado de carga final del sistema de almacenamiento de energía de 29.767 % (Fig. 4.14) y un consumo de combustible de 19.297kg/s (Fig. 4.15).

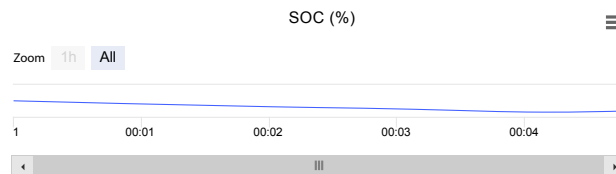


Fig. 4.14. SOC del sistema de almacenamiento de energía (Emulación a través de la derivada del valor medio).

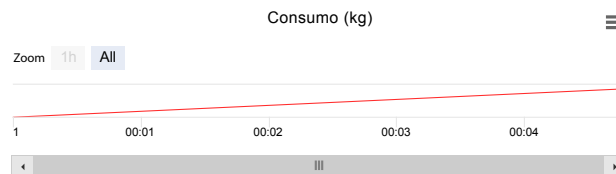


Fig. 4.15. Consumo de combustible (Emulación a través de la derivada del valor medio).

Por otra parte, la emulación a través de la derivada del cociente de diferencia simétrica se realizó en 5.2652 segundos, logrando la respuesta de estado de carga final del sistema de almacenamiento de energía de 29.618 % (Fig. 4.16) y un consumo de combustible de 19.199kg/s (Fig. 4.17).

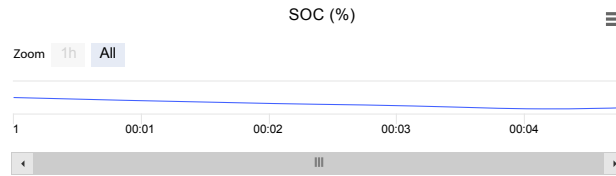


Fig. 4.16. SOC del sistema de almacenamiento de energía (Emulación a través de la derivada del cociente de diferencia simétrica).

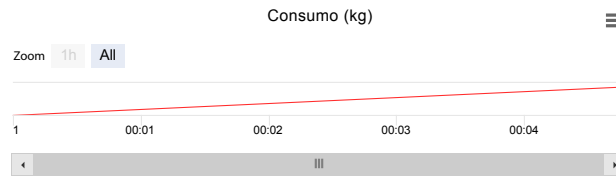


Fig. 4.17. Consumo de combustible (Emulación a través de la derivada del cociente de diferencia simétrica).

Se utilizó un analizador lógico para medir los tiempos en que la plataforma de emulación tarda en procesar la maquina de estados del vehículo, así como los tiempos que tarda al entrar en cada uno de los subsistemas, Fig. 4.18.

El tiempo total que la plataforma tomó para realizar la emulación del ciclo de conducción de 8hrs. fue de 5.210s. El core del microcontrolador está configurado para trabajar a 200Mhz, es decir, un ciclo de instrucción del microcontrolador tarda 20ns.

Se emplearon 3 señales para analizar los tiempos, la señal A0 representa los tiempos que tardó el microcotrolador en procesar cada uno de los subsistemas, la señal A1 representa el tiempo total y la señal A2 representa el tiempo de emulación del modelo cuasi-estático por cada valor del ciclo de conducción.

El tiempo (A2) para procesar un dato del ciclo de conducción en el modelo completo del vehículo es de 18us. Donde el subsistema del convertidor de potencia es el que demora un

tiempo mayor con respecto a los demás subsistemas, e tiempo que le toma al microcontrolador ejecutar el modelo del convertido es de 11.65us.

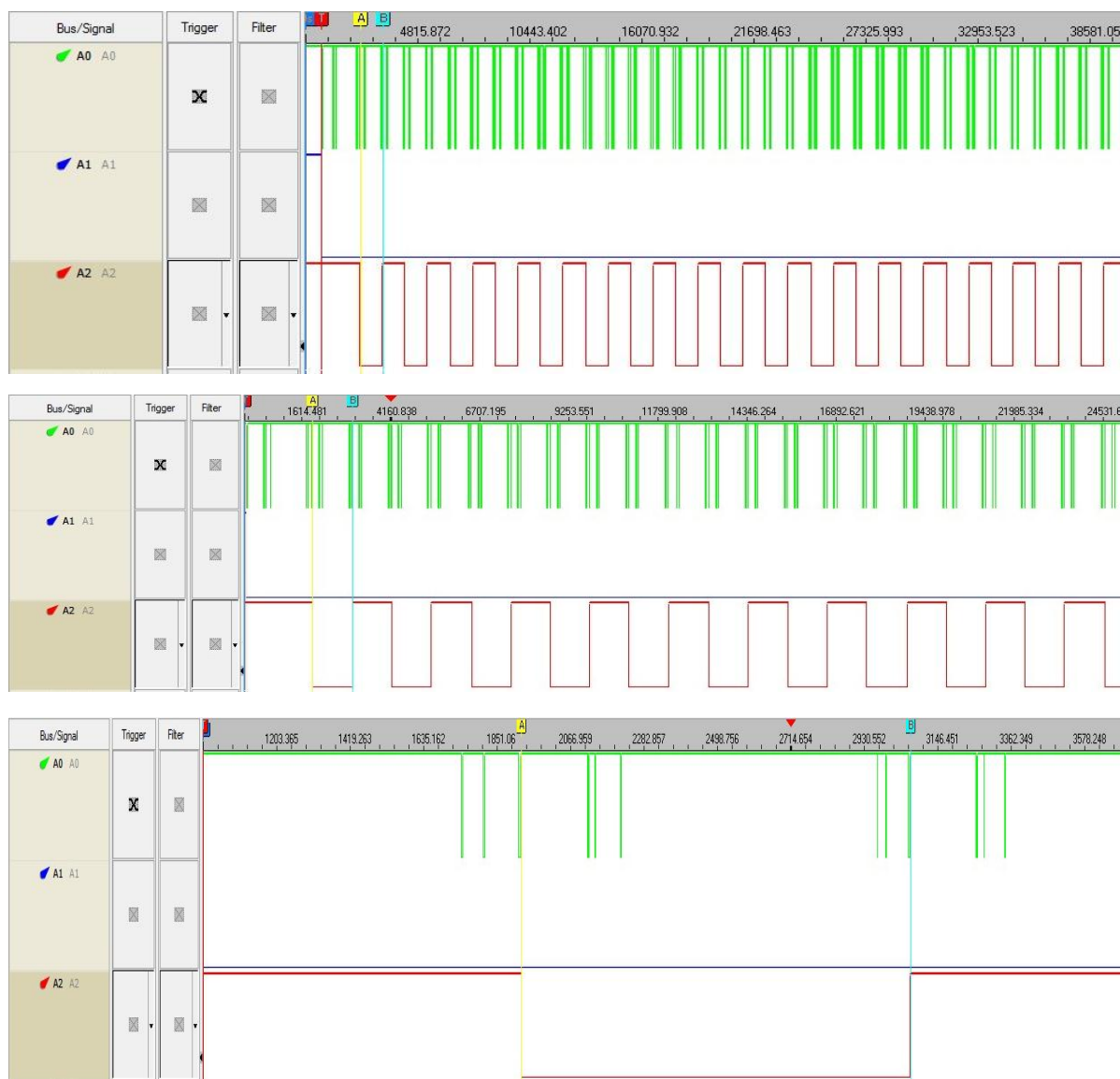


Fig. 4.18. Análisis de tiempos del sistema embebido.

4.3. Validación del funcionamiento de la plataforma de emulación

Finalmente, se realizó la comparación de los resultados obtenidos mediante la simulación y emulación, para ello, se empleó el estado de carga del sistema de almacenamiento de energía (Fig. 4.19) y el consumo de combustible (Fig. 4.20).

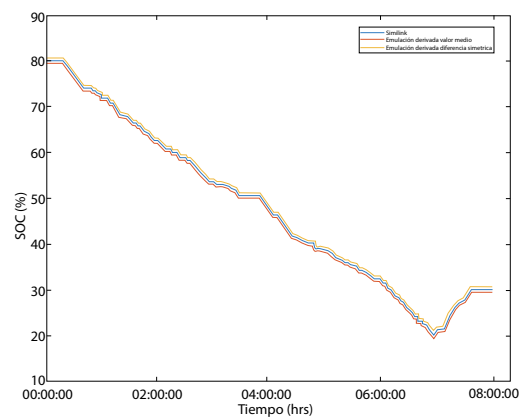


Fig. 4.19. Comparación del SOC del sistema de almacenamiento de energía.

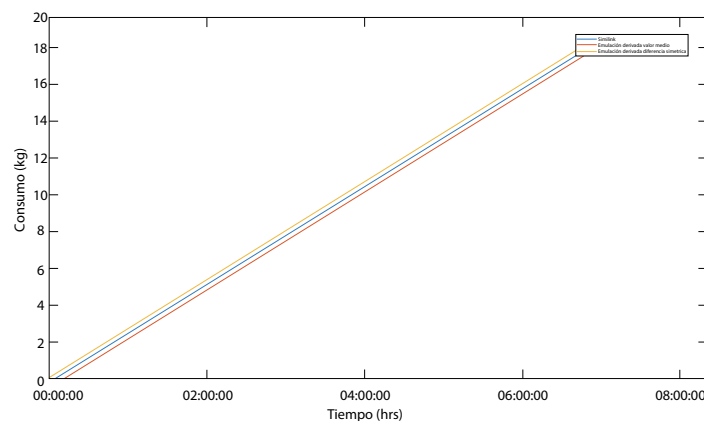


Fig. 4.20. Comparación del consumo de combustible.

En resumen, es posible realizar una comparación del tiempo de procesamiento, estado de carga final del sistema de almacenamiento de energía y el consumo, Tabla 4.4.

Tabla. 4.4. Resultados

	Tiempo (s)	SOC (%)	Consumo (kg/s)
Simulación	230.7	29.907	19.438
Emulación (Derivada del valor medio)	5.210	29.767	19.297
Emulación (Derivada de diferencia simétrica)	5.265	29.618	19.199

Los resultados muestran, que la emulación a través de la derivada del valor medio logró una diferencia de 0.724 % con respecto a la simulación, además, el tiempo de procesamiento fue 44 veces más rápido que la simulación.

En el presente capítulo se realizó la implementación y validación de la plataforma de emulación de un vehículo eléctrico híbrido. Cabe resaltar que la emulación se realizó empleando la derivada del valor medio y la derivada del cociente de diferencia simétrica.

Conclusiones

El presente trabajo de tesis se desarrolló una plataforma de emulación de un vehículo eléctrico híbrido, lo que implicó lo siguiente:

- El diseño e implementación de la plataforma de emulación, que consiste en un sistema embebido, un protocolo de comunicación y software.
- El diseño e implementación de un sistema embebido, realizado en una placa de circuito impreso para el microcontrolador PIC32MZ2048ECM144 y el modelo cuasi-estático de un vehículo eléctrico híbrido.
- La implementación de dos métodos para la realización de la función derivada, el método del valor medio y el de diferencia simétrica. Además un método para la función integral, que son empleados en el modelo cuasi-estático de un vehículo eléctrico híbrido.
- La implementación de protocolo de comunicación entre el software y el sistema embebido, considerando un algoritmo para la decodificación en el buffer de recepción del módulo USART.
- El diseño de un software a partir del patrón de arquitectura de software *Modelo-Vista-Controlador*. Además, el software fue implementado en un dispositivo móvil, logrando una gran flexibilidad de la plataforma de emulación.

- Finalmente, se realizó una validación de la plataforma de emulación de un vehículo eléctrico híbrido, realizando una comparación con la simulación del modelo en el software *Simulink*. Se comprobó el correcto funcionamiento y se logró reducir el tiempo de ejecución del modelo cuasi-estático.

Los resultados de la implementación muestran, que la plataforma de emulación arroja los resultados de SOC de sistema de almacenamiento y consumo de combustible con el error de 0.724 % comparados con datos de simulación.

Aunado a las ventajas de plataforma de emulación, tales como consideración de los parámetros reales a partir de un vehículo específico, cambio de ciclos de conducción y rapidez de procesamiento, además de ser el software amigable e intuitivo lo que hace una opción viable para su implementación en diseño y estudio de los vehículos eléctricos híbridos.

Appendices

.1. Firmware

Pseudocódigo de implementación de derivada de diferencia simetrica en el firmware

```
1 function derivatte_2_point_symmetric_difference_quotient(fxh, fx_h
    , time)
2     lim=time
3     while(lim>epsilon)
4         df = ((fxh) - fx_h)/(2*h);
5         lim--
6         fxh -= lim
7         fx_h -= -lim
8         h -= lim
9     return df
```

Pseudocódigo de implementación de derivada del valor medio en el firmware

```
1 function derivatte_mean_value(fa, fb, a, b, time)
2     lim=time
3     while(lim>epsilon)
4         dfc = (fb - fa)/(b - a)
```



```

5         lim —
6         fb -= -lim
7         fa -= lim
8         b -= -lim
9         a -= lim
10    return dfc

```

Pseudocódigo de implementación de la integral del metodo trapezoidal en el firmware

```

1  function integral_trapezoidal_rule_1 (valueA , valueB , fa , fb , h ,
    time)
2      for (i=1; i<=h; i++)
3          valueA = valueB;
4          valueB = valueA + (i*h);
5          iFf += (valueB-valueA) * ( (fa + fb) / 2.0 );
6      return iFf

```

Pseudocódigo de implementación de la dinámica longitudinal del vehículo en el firmware

```

1  function vehicleModelLongitudinalDynamic ()
2      vehicleForceA = 0.5 * varAirDensity * vehicleParam.frontalArea
        * vehicleParam.dragCoefficient * pow(dataPic.speedPIC , 2)
3      vehicleForceR = (vehicleParam.vehicleWeight + pic .
        dynamicWeightPIC + vehicleParam.emWeight + vehicleParam .
        essWeight) * gravity * vehicleParam.rollingResistence
4      vehicleAngularSpeed = (pic.speedPIC / vehicleParam.wheelRadius
        )
5      if ( pic.speedPIC >= 0 )
6          torque = (0.1) * abs( (((vehicleParam.vehicleWeight + pic .

```

```

dynamicWeightPIC + vehicleParam.emWeight + vehicleParam.
essWeight) * derivatte_mean_value(pic.speedPIC,
speedPICOLD, time) ) + (vehicleForceA + vehicleForceR)) *
vehicleParam.wheelRadius )
7 speedPICOLD = pic.speedPIC

```

Pseudocódigo de implementación de la transmisión en el firmware

```

1 function transmision()
2     if( pic.gearPIC >= 1 && pic.gearPIC <= 6 )
3         outAngularSpeed = (vehicleAngularSpeed * vehicleParam.
            gearRatio[ pic.gearPIC ])
4         outTorque = vehicleForceT * vehicleParam.gearRatio[ pic.
            gearPIC ])

```

Pseudocódigo de implementación del motor/generador eléctrico en el firmware

```

1 function electricMotor()
2     if (outPowerElectricMotor >= 0)
3         powerMotorGenerador = (eficienciaMotor()) *
            outPowerElectricMotor
4     else
5         powerMotorGenerador = (eficienciaGenerador()) *
            outPowerElectricMotor

```

Pseudocódigo de implementación del tablas de busqueda (lookup tables) de eficiencia del motor eléctrico en el firmware

```

1 function integer eficienciaMotor(torque, angularSpeed, index)
2     if(*index)

```

```
3         *index—
4         return eficienciaEtaActual
5     else
6         *eficienciaEtaActual = *index + 3
7         return eficienciaEta[*torque+1, *angularSpeed]
```

Pseudocódigo de implementación del convertidor de potencia en el firmware

```
1 function powerConverter()
2     estadosPowerConverter()
3     powerconverterActualPower = powerMotorGenerador
4     actualEssSOC = outESSbatteryPackSOC
5     vehicleOperationMode = powerconverterActualPower
6     essMaxPower = 2 * (batteryCapacity) * (batteryNominalVoltage *
        numberSerieCell);
7     if(metodo == 1)
8         if( (powerlinkOperationMode == 1) && (essBatteryState == -1)
        && (powerconverterActualPower <= essMaxPower) )
9             essPower = -powerconverterActualPower
10            icePower = 0
11        else
12            if( (powerlinkOperationMode == 1) && (essBatteryState ==
        -1) && (powerlinkPower > essMaxPower) )
13                outESSPower = -essMaxPower
14                outICEPower = powerconverterActualPower -
        essMaxPower
15        else
16            if( (powerlinkOperationMode == 1) && (
```

```
    essBatteryState == 1) )
17     essPower = 0
18     icePower = powerconverterActualPower
19 else
20     if( (powerlinkOperationMode == 2) && (
        essBatteryState == 1) && (powerlinkPower <=
        essMaxPower) )
21         essPower = powerconverterActualPower
22         icePower = 0
23     else
24         if( (powerlinkOperationMode == 2) && (
            essBatteryState == 1) && (powerlinkPower >
            essMaxPower) )
25             essPower = essMaxPower
26             icePower = powerconverterActualPower -
                essMaxPower
```

Pseudocódigo de implementación del sistema de almacenamiento de energía en el firmware

```
1 function ess()
2     estadosEss()
3     inputESSPower = outESSPower;
4     v_bwESS_battery_current = -(i_bwESS_power/essBatteryVoltage)
5     essBatteryOpenCircuitVoltage = (((
        p_bwESS_battery_nominalVoltage * p_bwESS_battery_serieCell))
        * interpolacion(dataSOC,dataVOC,essBatterySOC))
6     batteryChargeCapacity = integral_trapezoidal_rule(
        v_bwESS_battery_current , v_bwESS_battery_currentOLD)
```

```
7     v_bwESS_battery_currentOLD = v_bwESS_battery_current
8     essBatterySOC = p_bwESS_battery_initialSOC - ((
        batteryChargeCapacity * (100 / (p_bwESS_battery_capacity *
        p_bwESS_battery_parallelCell))) * (1))
9     essBatteryDOD = 100 - essBatterySOC
10    batteryCellRCBranchCapacitorVoltage = (v_bwESS_battery_current
        / (batteryCellRCBranchCapacitor * (
        p_bwESS_battery_parallelCell / p_bwESS_battery_serieCell)))
11    essBatteryVoltage = (essBatteryOpenCircuitVoltage -
        batteryCellRCBranchCapacitorVoltage - (
        v_bwESS_battery_current * batteryCellInternalResistance)) -
        voltageNegativeReference
```

Pseudocódigo de implementación del ICCE en el firmware

```
1  function ICE()
2      if( inputPower > ralentiPower)
3          icePower = inPower
4      else
5          *icePowerBSFC = bsfc[*torque, angularSpeed]
6          icePower = *icePowerBSFC + ralentiPower
7          consumption = (icePower*dieselLHV)
```

.2. Vistas de software

Vistas del software para la configuración de los parámetros del vehículo



Fig. 21. Vistas de la aplicación.

Referencias

- [1] M. S. Alegre Buj, “Modelado del vehículo eléctrico e híbrido paralelo por medio de MATLAB/SIMULINK y planificación de estaciones de carga mediante sistemas de información geográfica y algoritmos genéticos”, Tesis Doctoral, Departamento de ingeniería eléctrica, electrónica, control, telemática y química aplicada a la ingeniería, UNED, 2017.
- [2] I. García García, “Análisis de factores económicos, tecnológicos y políticos en el futuro mercado del vehículo eléctrico en España mediante dinámica de sistemas”, Tesis Doctoral, Departamento de Ingenierías de Sistemas y Automáticas, Universidad de Valladolid, Valladolid, España, 2013.
- [3] M. C. Raga Arroyo, “Dimensionamiento, modelado e implementación de sistemas de distribución de potencia aplicados a Vehículos Eléctricos Híbridos basados en Pilas de Combustible”, Tesis Doctoral, Departamento de tecnología electrónica, UC3M, Leganés, España, 2015.
- [4] G. Rizzoni and S. Onori, “Energy Management of Hybrid Electric Vehicles: 15 Years of Development at the Ohio State University” IFP Energies nouvelles International Conference, pp. 1–14, Jul. 2014.
- [5] Q. Xu, J. Sun, X. Jiang, and S. Cui, “Comparison Analysis of Power Management used in Hybrid Electric Vehicle based on Electric Variable Transmission”, 11th International Conference on Control, pp. 1–7, 2016.

-
- [6] M. A. Durán Fonseca, “Control de la Operación de un Vehículo Eléctrico con un Uso Eficiente de la Energía”, Tesis Doctoral, Departamento de Ingeniería Electrónica, CENIDET, Morelos, México, 2013.
- [7] F. Naseri, E. Farjah and T. Ghanbari, “An Efficient Regenerative Braking System Based on Battery/Supercapacitor for Electric, Hybrid, and Plug-In Hybrid Electric Vehicles With BLDC Motor,” in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3724-3738, May 2017.
- [8] A. Zia, “A comprehensive overview on the architecture of Hybrid Electric Vehicles (HEV),” 2016 19th International Multi-Topic Conference (INMIC), Islamabad, pp. 1-7, 2016.
- [9] S. D. V. R. Vadlamudi, V. Kumtepli, S. Ozcira and A. Tripathi, “Hybrid energy storage power allocation and motor control for electric forklifts,” 2016 Asian Conference on Energy, Power and Transportation Electrification (ACEPT), Singapore, pp. 1-5, 2016.
- [10] S. Jurkovic, K. M. Rahman and P. J. Savagian, “Electric traction motors for Cadillac CT6 plugin hybrid-electric vehicle,” *IEEE Energy Conversion Congress and Exposition (ECCE)*, Milwaukee, WI, pp. 1-9, 2016.
- [11] Z. Yang, F. Shang, I. P. Brown and M. Krishnamurthy, “Comparative Study of Interior Permanent Magnet, Induction, and Switched Reluctance Motor Drives for EV and HEV Applications,” in *IEEE Transactions on Transportation Electrification*, vol. 1, no. 3, pp. 245-254, Oct. 2015.
- [12] R. Vincent and V. Emmanuel and G. Lauric and G. Laurent, “Optimal sizing of an electrical machine using a magnetic circuit model: application to a hybrid electrical vehicle”, *IET Electrical Systems in Transportation*, vol. 6, no. 1, pp. 27-33, 2016.
-

- [13] X. Wu and S. Han and H. Sun and G. Han, "Efficiency analysis of planetary coupling drive system with dual motors on electric bus", 2015 IEEE International Conference on Mechatronics and Automation (ICMA), pp. 2547-2552, Aug. 2015.
- [14] V. Sezer and M. Gokasan and S. Bogosyan, "A Novel ECMS and Combined Cost Map Approach for High-Efficiency Series Hybrid Electric Vehicles", IEEE Transactions on Vehicular Technology, vol. 60, no. 8, pp. 3557-3570, Oct. 2011
- [15] S. Overington and S. Rajakaruna, "A novel model of internal combustion engine for high efficiency operation of hybrid electric vehicles and power systems," 2014 Australasian Universities Power Engineering Conference (AUPEC), Perth, WA, pp. 1-6, 2014.
- [16] A. I. H. Al-zughaibi and N. F. Jasim, "Control system design for fuel to air ratio in an internal combustion engine with unknown time-delay and fuel fraction impinging manifold's wall," 2016 International Conference for Students on Applied Engineering (ICSAE), Newcastle upon Tyne, pp. 137-141, 2016.
- [17] R. Luís and J. C. Quadrado and J. F. Silva, "Optimal brake specific fuel consumption trajectory for stand-alone variable speed Diesel Gen-Set", 2015 9th International Conference on Compatibility and Power Electronics (CPE), pp. 289-294, June 2015.
- [18] K. Lee and B. An and M. W. Hadi and J. Choi and Y. Song, "Optimized control strategy for hybrid energy system", 2015 9th International Conference on Power Electronics and ECCE Asia (ICPE-ECCE Asia), pp. 2495-2500, Jun. 2015.
- [19] P. H. Bauer and E. Pérez-Bernabeu, "The effect of the operating point choice on fuel economy in series hybrids", 2013 World Electric Vehicle Symposium and Exhibition (EVS27), pp. 1-4, Nov. 2013.

- [20] Q. Jin and G. Wu and K. Boriboonsomsin and M. J. Barth, “Power-Based Optimal Longitudinal Control for a Connected Eco-Driving System”, IEEE Transactions on Intelligent Transportation Systems, vol. 17, pp. 2900-2910, Oct. 2016.
- [21] J. Sharafi, S. Hager, W. H. Moase, P. Dennis, M. J. Brear and C. Manzie, “Fast extremum seeking for optimization of brake specific fuel consumption,” 53rd IEEE Conference on Decision and Control, Los Angeles, CA, pp. 3432-3437, 2014.
- [22] B. Wang, J. Xu, B. Cao, and X. Zhou, “A novel multimode hybrid energy storage system and its energy management strategy for electric vehicles,” Journal of Power Sources, vol. 281, pp. 432 – 443, 2015.
- [23] L. Zhang, X. Xia and F. Barzegar, “Control of a battery/supercapacitor hybrid energy storage system for electric vehicles,” 2017 36th Chinese Control Conference (CCC), Dalian, 2017, pp. 9560-9565. May. 2015
- [24] F. Jin, M. Wang and C. Hu, “A fuzzy logic based power management strategy for hybrid energy storage system in hybrid electric vehicles considering battery degradation,” 2016 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, pp. 1-7, 2016.
- [25] R. Gu, P. Malysz, Deqiang Wang, Weizhong Wang, H. Yang and A. Emadi, “On the design of a direct cell coupled hybrid energy storage system for plug-in hybrid electric vehicles,” 2016 IEEE Transportation Electrification Conference and Expo (ITEC), Dearborn, MI, pp. 1-7, 2016.
- [26] H. Miniguano, C. Raga, A. Barrado, A. Lázaro, P. Zumel and E. Olías, “A comparative study and parameterization of electrical battery models applied to hybrid electric vehicles,” 2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles and International Transportation Electrification Conference (ESARS-ITEC), Toulouse, pp. 1-6, 2016.

- [27] M. Wieczorek and M. Lewandowski, "A mathematical representation of an energy management strategy for hybrid energy storage system in electric vehicle and real time optimization using a genetic algorithm," *Applied Energy*, vol. 192, pp. 222 – 233, 2017.
- [28] M. M. Sabri, K. Danapalasingam, and M. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 1433 – 1442, 2016.
- [29] N. Sulaiman, M. Hannan, A. Mohamed, E. Majlan, and W. W. Daud, "A review on energy management system for fuel cell hybrid electric vehicle: Issues and challenges," *Renewable and Sustainable Energy Reviews*, vol. 52, pp. 802 – 814, 2015.
- [30] K. Itani, A. De Bernardinis, Z. Khatir and A. Jammal, "Energy management of a battery-flywheel storage system used for regenerative braking recuperation of an Electric Vehicle," *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Florence, pp. 2034-2039, 2016.
- [31] X. Zhang, C. C. Mi, and C. Yin, "Active-charging based powertrain control in series hybrid electric vehicles for efficiency improvement and battery lifetime extension," *Journal of Power Sources*, vol. 245, no. Supplement C, pp. 292 – 300, 2014.
- [32] Y. Li, Y. Wang and S. Jiao, "Simulation research on control strategy of single-axle parallel hybrid electric vehicle," *2017 36th Chinese Control Conference (CCC)*, Dalian, pp. 9390-9395, 2017.
- [33] J. Fu, S. Songi, Z. Fu and J. Ma, "Efficiency control to maximize the powertrain efficiency for parallel hybrid electric vehicle," *2017 36th Chinese Control Conference (CCC)*, Dalian, pp. 289-294, 2017.

- [34] R. Cheng and Z. Dong, “Modeling and Simulation of Plug-In Hybrid Electric Powertrain System for Different Vehicular Application,” 2015 IEEE Vehicle Power and Propulsion Conference (VPPC), Montreal, QC, pp. 1-7, 2015.
- [35] T. Letrouvé, A. Bouscayrol, W. Lhomme, N. Dollinger and F. M. Calvairac, “Different models of a traction drive for an electric vehicle simulation”, 2010 IEEE Vehicle Power and Propulsion Conference, Lille, pp. 1-6, 2010.
- [36] A. Jain, T. Nueesch, C. Naegel, P. M. Lassus and C. H. Onder, “Modeling and Control of a Hybrid Electric Vehicle With an Electrically Assisted Turbocharger”, in IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 4344-4358, June 2016.
- [37] G. I. Zamora Gómez, “Efecto de la topología del tren de potencia en la vida de las baterías”, Master thesis, Instituto Potosino de Investigación Científica y Tecnológica, A.C., 2015.
- [38] C. Naxin, L. Fengxia, W. Jian and W. Xiaoxia, “Optimization of HEV energy management strategy based on driving cycle modeling”, 2015 34th Chinese Control Conference (CCC), Hangzhou, pp. 7983-7987, 2015.
- [39] D. Zhao, R. Stobart, G. Dong and E. Winward, “Real-Time Energy Management for Diesel Heavy Duty Hybrid Electric Vehicles”, in IEEE Transactions on Control Systems Technology, vol. 23, no. 3, pp. 829-841, May 2015.
- [40] Y. Wang, S. Niu and W. Fu, “Electrical-Continuously Variable Transmission System Based on Doubly Fed Flux-Bidirectional Modulation”, in IEEE Transactions on Industrial Electronics, vol. 64, no. 4, pp. 2722-2731, April 2017.
- [41] X. Li, S. Kher and S. Lin, “Multi-Domain System Level Behavioral HEV Library Using VHDL-AMS”, 2013 IEEE International Electric Vehicle Conference (IEVC), Santa Clara, CA, pp. 1-5, 2013.

-
- [42] M. F. M. Sabri, K. A. Danapalasingam, M. F. Rahmat and M. R. M. Yusof, “Fuel economy analysis of a through-the-road hybrid electric vehicle”, 2015 10th Asian Control Conference (ASCC), Kota Kinabalu, pp. 1-6, 2015.
- [43] D. F. Torres Moscoso, “Análisis energético de un sistema de propulsión eléctrico alternativo para movilidad en el cantón Cuenca”, Master thesis, Universidad del Azuay, 2016.