



Departamento de Posgrados  
Maestría en Sistemas Vehiculares

Implementación de un modelo dinámico para una red de  
comunicación presente en un vehículo eléctrico Kia Soul  
mediante NI-XNET

Trabajo de graduación previo a la obtención del título de  
magister en Sistemas Vehiculares

Autor:

Diego Fernando Rojas Hiedra

Director:

Dr. Efrén Esteban Fernández Palomeque

Cuenca – Ecuador

2019

# **Implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico Kia Soul mediante NI-XNET**

Diego Fernando Rojas Hiedra

Maestrante del Programa en Sistemas Vehiculares de la Universidad del Azuay

[drojas@uazuay.edu.ec](mailto:drojas@uazuay.edu.ec)

Dr. Efrén Esteban Fernández Palomeque

Docente de la Maestría en Sistemas Vehiculares y Director de la investigación

[efernandez@uazuay.edu.ec](mailto:efernandez@uazuay.edu.ec)

## **Resumen**

Desde la primera implementación de módulos electrónicos en vehículos convencionales, la comunicación entre ellos ha sido fundamental, lo que llevo a la creación del protocolo de comunicación CAN (Controller Area Network). Las redes de comunicación presentes en vehículos eléctricos demandan una alta eficiencia y control para el intercambio de información entre los módulos electrónicos. El uso de diversos sistemas de seguridad activa, pasiva y tracción eléctrica generan un constante intercambio de información importante para el correcto funcionamiento del vehículo eléctrico. Este artículo presenta la implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico. El modelo permite analizar las tramas del bus de datos CAN para obtener los identificadores de los módulos de tracción, RPM y velocidad, con el propósito de obtener una herramienta para la validación durante una situación de falla. El objetivo principal es el monitoreo de la red de comunicación para analizar posibles fallos que puedan generarse durante su funcionamiento. Para el desarrollo del modelo y obtención de datos se analiza la configuración de la red de comunicación de un vehículo eléctrico marca Kia Soul modelo 2015 mediante Simulink- Nertwork Vehicle Toolbox, LabVIEW- Ni-Xnet y la tarjeta de adquisición de datos NI-9862.

Palabras Clave: Controller Area Network-CAN, Network, LabVIEW, Simulink, NI-XNET.

## **Implementation of a dynamic model for a communication network in a Kia Soul electric vehicle through NI-XNET.**

Communication between electronic modules in conventional vehicles has been fundamental since its first implementation, leading to the creation of the CAN (Controller Area Network) communication protocol. The communication networks in electric vehicles demand high efficiency and control for the exchange of information between electronic modules. The use of various active, passive and electric traction safety systems generates a constant exchange of important information for the proper functioning of the electric vehicle. This research presents the implementation of a dynamic model for a communication network in an electric vehicle. The model allows to analyze the frames of the CAN data bus to obtain the identifiers for the traction, RPM and speed modules. This was intended to obtain a tool for validation during a failure situation. The main objective is to monitor the communication network and analyze the possible failures that may be generated during operation. For the development of the model and data collection, the configuration of the communication network of a 2015 Kia Soul electric vehicle was analyzed using the Simulink-Network Vehicle Toolbox, LabVIEW-Ni-Xnet and the NI-9862 data acquisition card.

Keywords: Controller Area Network-CAN, Network, LabVIEW, Simulink, NI-XNET.



Translated by  
Ing. Paúl Arpi

# Implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico Kia Soul mediante NI-XNET

Diego Rojas  
Universidad del Azuay  
Maestría en Sistemas Vehiculares  
Cuenca, Ecuador  
drojas@uazuay.edu.ec

Efrén Fernández  
Universidad del Azuay  
Escuela de Mecánica Automotriz  
Cuenca, Ecuador  
Orcid 0000-0003-2188-9446

**Resumen**—Desde la primera implementación de módulos electrónicos en vehículos convencionales, la comunicación entre ellos ha sido fundamental, lo que llevó a la creación del protocolo de comunicación CAN (Controller Area Network). Los adelantos en CAN lo han convertido en el protocolo de comunicación más usado en vehículos actuales, incluyendo los vehículos eléctricos. Las redes de comunicación presentes en vehículos eléctricos demandan una alta eficiencia y control para el intercambio de información entre los módulos electrónicos. El uso de diversos sistemas de seguridad activa, pasiva y tracción eléctrica generan un constante intercambio de información importante para el correcto funcionamiento del vehículo eléctrico. Este artículo presenta la implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico. El modelo permite analizar las tramas del bus de datos CAN para obtener los identificadores de los módulos de tracción, RPM y velocidad, con el propósito de obtener una herramienta para la validación durante una situación de falla. El objetivo principal es el monitoreo de la red de comunicación para analizar posibles fallos que puedan generarse durante su funcionamiento. Para el desarrollo del modelo y obtención de datos se analiza la configuración de la red de comunicación de un vehículo eléctrico marca Kia Soul modelo 2015 mediante Simulink- Network Vehicle Toolbox, LabVIEW- Ni-Xnet y la tarjeta de adquisición de datos NI-9862.

**Palabras Clave**— CAN, NETWORK, LabVIEW, SIMULINK, NI-Xnet.

## I. INTRODUCCIÓN

El protocolo CAN, desarrollado a principio de los 80's por la empresa BOSCH, se ha convertido en el protocolo más fiable para la comunicación entre los sensores y actuadores que pueda poseer un automóvil[1]. A partir del 2008, en EEUU se ordena el uso de CAN para el diagnóstico a bordo de todos los automóviles, dando un giro positivo para los avances tecnológicos en lo que al protocolo se refiere[2]. En tiempos modernos, los vehículos eléctricos están marcando una tendencia en el desarrollo automotriz debido a su baja contaminación y bajo consumo energético. Los vehículos eléctricos requieren de una mayor cantidad de módulos, sensores y actuadores en comparación con un vehículo convencional. Generalmente, la red de comunicación de un vehículo eléctrico que se muestra en la Fig. 1, está conformada por una red de alta velocidad (250 Kbps) donde se encuentran los módulos que necesitan comunicación en tiempo real para una reacción inmediata, esta red de alta velocidad permanece conectada por medio de una compuerta (Gateway) a una red de baja velocidad (125 Kbps) donde se conectan las unidades de control eléctrico[3].

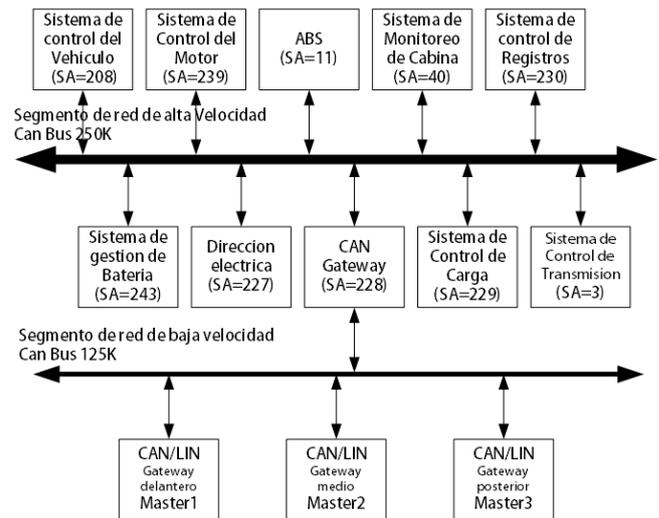


Fig. 1. Red de protocolo CAN.

Para la lectura de las tramas de datos CAN, existen en el mercado diversos dispositivos que no proporcionan una información detallada del evento ocurrido en el vehículo eléctrico.

¿Cómo podríamos obtener una información más detallada de la trama de datos que proporcionan los módulos que integran la red de comunicación CAN?

A través de Simulink se desea implementar un modelo de red con protocolo CAN presente en un vehículo eléctrico marca Kia Soul con el propósito de analizar los datos en diferentes condiciones de operación.

En los vehículos eléctricos existen una gran cantidad de módulos de control, sensores y actuadores, por esta razón, la complejidad de la red de comunicación se incrementa. Debido a este gran aumento de componentes, se han desarrollado los siguientes estudios:

En el artículo presentado en [3], aplica las reglas del protocolo SAE J1939 a un bus eléctrico puro con resultados fiables, pero se limitan a los temas de fallas que pueden ocurrir en la transmisión de datos y no contempla fallas en el mal funcionamiento de los componentes del vehículo.

En [4] se emula los códigos de falla de un scanner básico basándose en los identificadores de módulos sin examinar a profundidad los datos CAN.

En el artículo [5] se basan en códigos de falla generados por cierto modulo y los complementan con los códigos predeterminados por el sistema ODB sin analizar los bits de datos que proporciona el protocolo CAN.

En el artículo [6] se pone énfasis en el hardware y la comunicación con el protocolo SAE J1939 y sus reglas pre establecidas para la transmisión de datos y códigos de fallas, dejando de lado detección de fallas en ciertos desperfectos que no contemplan las reglas del protocolo establecido.

El presente artículo contribuye con la implementación de un modelo dinámico de red CAN en automóvil eléctrico con el propósito de verificar fallas en función de los ID's. El artículo se organiza en las siguientes secciones: en la sección II se realizará una revisión de las topologías de redes CAN en vehículo eléctrico, en la sección III se presenta la metodología para el desarrollo del modelo dinámico, en la sección IV se presenta la implementación del modelo propuesto, en la sección V se presenta la validación y comprobación y, por último, en la sección VI se presenta las conclusiones.

## II. TOPOLOGÍA DE REDES CAN EN EL VEHÍCULO ELÉCTRICO

La cantidad de módulos electrónicos implementados en un vehículo eléctrico supera en gran medida a los módulos que posee un vehículo convencional de gama media. En la Fig. 2 podemos observar los ID's con sus respectivas tramas de datos CAN de un vehículo eléctrico Kia Soul 2015 contando con 67 módulos conectados a la red.

ID	Time Stamp	Length	Data
0x18	119,589	8	00 00 00 60 00 00 20 00
0x34	119,219	8	00 00 00 00 00 00 00 00
0x50	119,579	4	00 00 00 00
0x110	119,669	8	E0 3C 20 09 00 00 00 00
0x120	119,519	4	00 00 00 00
0x153	119,693	8	00 00 01 FF FD 07 40 44
0x160	119,693	8	00 03 00 02 00 00 00 01
0x164	119,692	8	00 08 00 00 00 00 05 0D
0x165	119,689	8	FC 67 80 00 00 00 02 19
0x1F1	119,683	8	00 00 8E E0 08 8D C0 08
0x200	119,689	8	00 A8 2B 10 1E BB 3C 0A
0x201	119,690	8	81 43 FF 24 01 00 00 18
0x202	119,690	8	42 F5 00 00 F5 00 1F 5D
0x220	119,693	8	0F 04 21 04 00 00 D8 0F
0x291	119,689	8	F5 00 C5 04 FE F2 F2 00
0x2A1	119,690	8	00 00 00 00 00 00 00 00
0x2A2	119,693	8	05 00 00 00 00 00 00 00
0x2B0	119,688	5	F9 FF 00 07 32
0x2C0	119,682	8	3D 00 00 00 00 00 00 00
0x433	119,682	8	00 00 08 21 00 00 3C 00
0x47F	119,684	8	00 08 F2 00 00 00 00 00
0x480	119,694	8	37 02 30 02 3A 02 35 02
0x4B1	119,694	8	20 AB C1 40 00 00 00 CC
0x4F0	119,678	8	00 27 5E 00 00 D7 9F 00
0x4F1	119,618	8	C8 CE 00 00 00 00 80 00
0x4F2	119,678	8	20 27 30 A0 00 00 00 AF
0x510	119,629	8	00 00 00 00 00 00 00 00
0x517	119,520	8	00 00 00 00 00 00 00 00
0x51A	119,619	8	00 11 00 00 00 00 00 00
0x523	119,690	8	8C 3C 43 4B 4C 49 00 00
0x570	119,668	8	00 00 21 23 00 00 00 00
0x579	119,676	8	00 00 00 00 00 00 00 00
0x57A	119,676	8	00 00 00 00 00 00 00 00
0x57B	119,676	8	E1 01 00 00 00 00 00 00
0x57F	119,648	8	80 00 F4 00 40 00 00 00
0x581	119,648	8	00 00 00 00 00 00 00 00

Fig. 2. Módulos de un vehículo Kia Soul Eléctrico 2015.

En la Fig. 3 se observa los ID's de los módulos que posee un vehículo Kia Rio Xcite 2011 contando con tan solo 8 módulos que conforman la red CAN. La diferencia en cantidad de módulos de control entre un vehículo eléctrico y uno convencional es muy significativa.

Timestamp	ID	Name	Length	Data
46.062287	A0		8	00 78 00 00 10 00 00
46.062039	2A0		8	00 00 89 00 00 00 00 00
46.061322	316		8	31 ED 00 00 F5 28 00 00
46.061555	329		8	D4 AB 5E 06 00 20 FF 10
46.056508	370		8	00 20 00 00 00 00 00 00
46.066027	43F		8	00 40 00 FF 58 00 00 00
46.056258	440		8	00 00 00 04 00 00 00 00
46.061791	545		8	4E 00 00 77 E8 05 28 06

Fig. 3. Módulos de un Vehículo Kia Rio Xcite 2011.

La topología de una red CAN de un vehículo eléctrico es por lo general en paralelo de alta velocidad[2]. Como se observa en la Fig. 4, se divide en cuatro partes: DriveCAN, PowerCAN, ChassisCAN y DashCAN. DriveCAN posee el MCU (Motor Controller Units) y el VCU (Vehicle Controller Units). PowerCAN contiene todo el sistema de alimentación, el cual aborda el BMS (Battery Management System), el CCS (Charger Controller System) y la VCU. ChassisCAN se

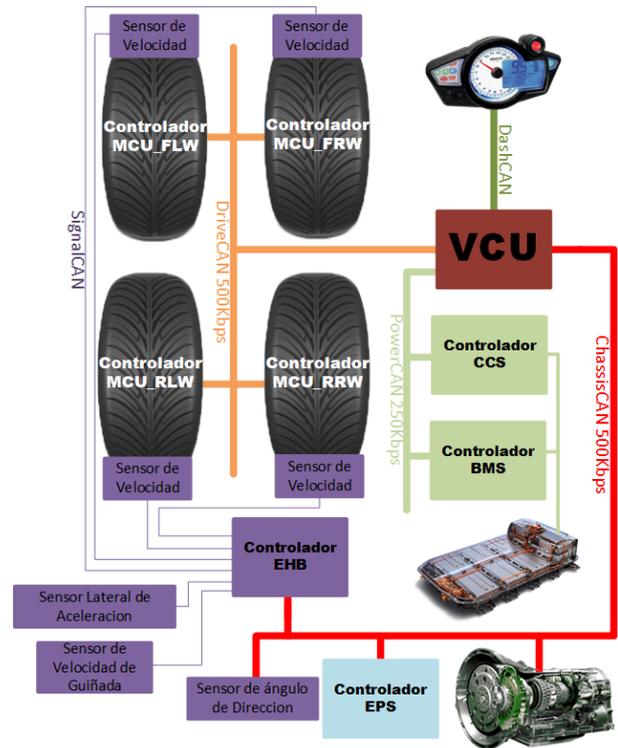


Fig. 4. Topología de red de un vehículo eléctrico.

enfoca en el control del sistema del chasis, como la EPS (Electronic Power Steering System) y el EHB (Electronic Hydraulic Brake System). DashCAN es el encargado de los mensajes del sistema a través del tablero electrónico[7].

Dentro de un Automóvil eléctrico, como se observa en la Fig. 5, se encuentran distintos tipos de redes, ya sea de diferente velocidad de transmisión o distinto protocolo de comunicación. La comunicación entre este tipo de redes es fundamental para el confort y seguridad. Estas topologías comparten información a través de un GATEWAY que permite la comunicación entre los módulos que se encuentran en estas redes[8].

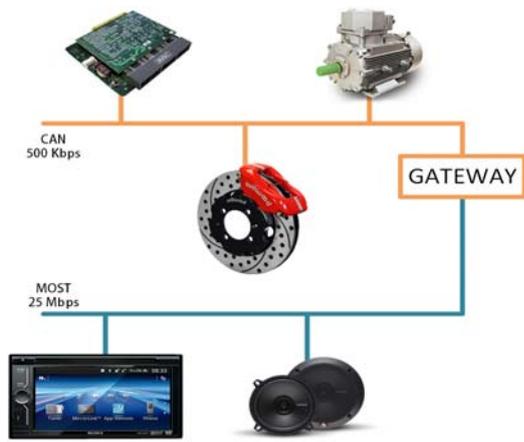


Fig. 5. Conexión de dos redes mediante un gateway.

### III. METODOLOGÍA DEL MODELO DINÁMICO

Para el análisis de los datos CAN obtenidos en el vehículo eléctrico Kia Soul 2015, se ha tomado en cuenta dos escenarios, el primero denominado “sobre la marcha”, el cual nos servirá en gran manera para la validación del modelo



Fig. 6. Metodología del Modelo Dinámico.

dinámico, y un segundo escenario, “laboratorio”, donde se analizarán los datos de manera meticulosa sin necesidad de esforzar al vehículo. En la Fig. 6 se presentan las herramientas que se usaron en los dos escenarios.

#### A. Modelo de Laboratorio

Para el análisis en laboratorio, los datos se obtienen a través de la tarjeta NI-9862 con la aplicación NI-XNET Bus Monitor, la cual nos permite guardar todas las tramas de datos en un archivo de texto. Desde Excel se importa el archivo de texto generando una hoja de cálculo delimitado por comas con la estructura de la tabla I, este archivo se usa para el análisis en LabVIEW.

TABLA I. ESTRUCTURA DE LA TABLA USADA EN LABVIEW

Timestamp	CSV File Struct for LabView			
	ID	Payload	Length	Name
0,001633	291	00 00 00 00 FE F2 F2 00	8	S
0,001875	153	00 00 02 FF FD 07 40 47	8	S
0,002119	160	20 29 01 02 00 00 00 0A	8	S
0,002369	164	00 08 00 00 00 00 06 0E	8	S
0,002619	1F1	00 00 00 00 00 00 00 00	8	S
0,00286	200	00 A8 2C 10 00 BB 3C 13	8	S
0,003108	201	81 00 00 00 00 00 00 81	8	S
0,00335	202	42 00 00 00 00 00 1F 5D	8	S
0,003584	220	0E 04 E8 03 88 00 16 10	8	S
0,003835	2A1	00 00 00 00 00 00 00 00	8	S
0,004075	2A2	04 00 00 B8 03 00 00 1D	8	S

Para el análisis en Simulink, se necesita crear a partir del archivo csv, una estructura compatible con los bloques de CAN Replay a través de programación en MATLAB generando un archivo de extensión mat. En la tabla II se muestra los campos que conforman la estructura que se usa en Simulink.

TABLA II. ESTRUCTURA DE LA TABLA USADA EN SIMULINK

Field	Table Struct for Simulink				
	Name	Value	Size	Bytes	Class
1	Data	8x"# of samples taken" uint8	8x1	# of samples taken	uint8
2	Error	1x"# of samples taken" uint8	1x1	# of samples taken	uint8
3	Extended	1x"# of samples taken" uint8	1x1	# of samples taken	uint8
4	ID	1x"# of samples taken" uint32	1x1	# of samples taken	uint32
5	Length	1x"# of samples taken" uint8	1x1	# of samples taken	uint8
6	Remote	1x"# of samples taken" uint8	1x1	# of samples taken	uint8
7	Timestamp	1x"# of samples taken" double	1x1	# of samples taken	double

El modelo propuesto de laboratorio nos ahorró tiempo de uso del vehículo, lo cual implica un ahorro en transportación y consumo eléctrico.

#### B. Modelo sobre la marcha

La metodología usada para el análisis de datos CAN con el vehículo en funcionamiento se basa en la configuración de la tarjeta de adquisición de datos NI-9862 directamente en la programación de LabVIEW y Simulink. En la Fig. 7 se observa la configuración de la adquisición de una trama de datos por la interface CAN1 que corresponde a la tarjeta NI-9862 a una velocidad de 500Kbps en LabVIEW.

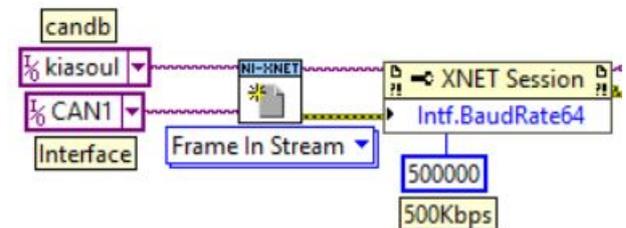


Fig. 7. Adquisición de datos CAN en LabVIEW.

En la Fig. 8, se observa el bloque de configuración de adquisición de datos CAN en Simulink con las mismas características que en el programa anterior.

# NI-XNET 9862 CAN/HS CAN1 Bus speed: 500000

## CAN Configuration

Fig. 8. Adquisición de datos CAN en Simulink.

### IV. IMPLEMENTACIÓN DEL MODELO DINÁMICO

#### A. Análisis en Laboratorio

Para iniciar con el estudio de datos CAN, fue necesaria la configuración de la tarjeta de adquisición de datos NI-9862 a través de la aplicación NI-XNET Bus Monitor, en la cual se configura la interface CAN1 a un baud Rate de 500 Kbaud y los parámetros como se observa en la Fig. 9.

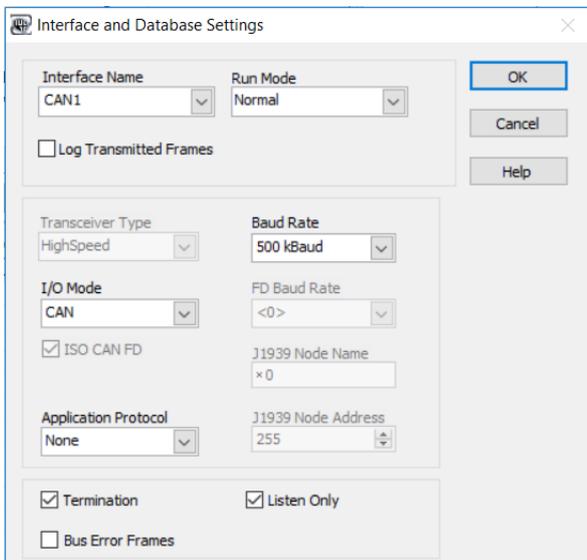


Fig. 9. Configuración de NI-XNET bus monitor.

Al finalizar la sesión, se obtiene un archivo de texto con todos los campos que conforman la trama de datos CAN, a partir de este archivo, se genera un documento delimitado por comas con el que comenzaremos a trabajar.

Para el análisis en LabVIEW, los datos se manejan como un arreglo obtenidos desde un spreadsheet delimitado por “;”, este arreglo es transformado a un clúster para obtener las filas ordenadas en: Time Stamp, ID y Data. De esta forma podemos analizar las tramas CAN modificando el tiempo de adquisición de datos (Baud rate). Al analizar los datos induciendo a cambios de estado en el funcionamiento del vehículo, logramos identificar el modulo que contiene los bits de velocidad, RPM y el módulo de tracción eléctrica. Los datos correspondientes al ID 4F2, en el byte 1, corresponde a la velocidad en una escala 1:2. El modulo que contiene los datos de RPM corresponde el ID 1F1, los bytes 6 y 7 los organizamos de MSB (More Significant Bit) a LSB (Low Significant bit) para obtener el valor de las rpms en decimal, toda esta programación se puede ver en la Fig. 10.

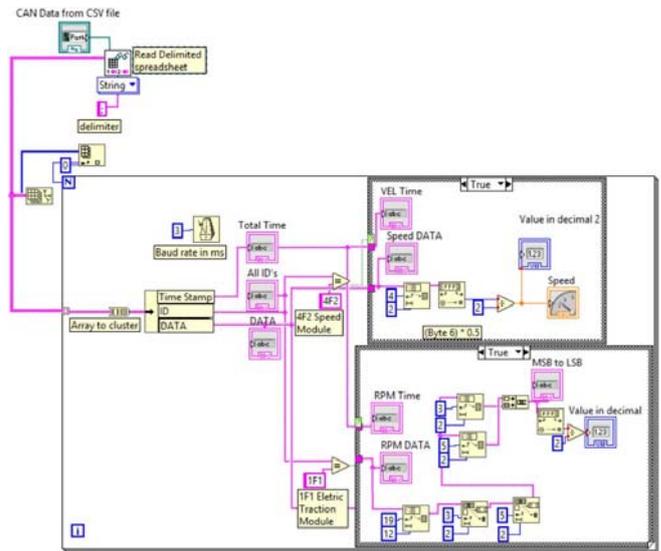


Fig. 10. Análisis de la trama de datos CAN en LabVIEW.

A partir de la comparación de datos de velocidad y RPM, como se ve en Fig. 11, encontramos el ID correspondiente al módulo de tracción eléctrica que nos da valores del voltaje de entrada del inversor.

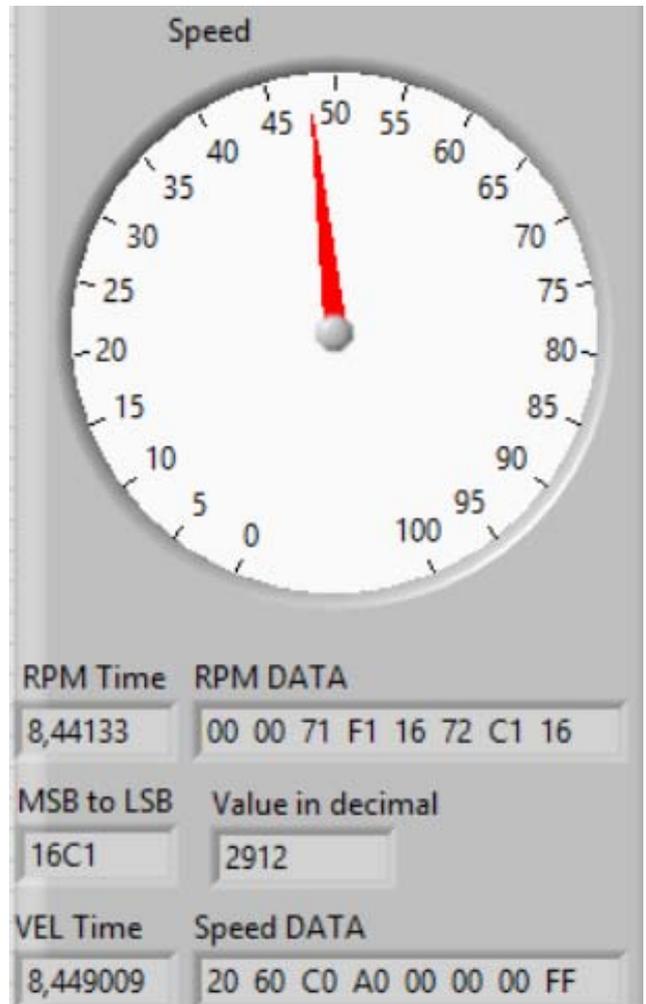


Fig. 11. Panel frontal en LabVIEW.

Para el análisis de datos en Simulink, se estructuró una tabla compatible con el bloque de programación CAN Replay con la programación en MATLAB que se observa a continuación:

```
[Extended] = uint8(xlsread('Extended.xlsx'));
[Length] = uint8(xlsread('Length.xlsx'));
[Remote] = uint8(xlsread('Remote.xlsx'));
[Error] = uint8(xlsread('Error.xlsx'));
[ID] = uint32(xlsread('ID.xlsx'));
[Timestamp] = xlsread('Timestamp.xlsx');
[Data] = uint8(xlsread('Data.xlsx'));
Extended=(Extended. ');
Length=(Length. ');
Remote=(Remote. ');
Error=(Error. ');
ID=(ID. ');
Timestamp=(Timestamp. ');
Data=(Data. ');
Msgsev=struct('Extended',Extended,'Length',Length...
    ,'Remote',Remote,'Error',Error,'ID',ID,'Timestamp',...
    'Timestamp','Data',Data);
save('KiaSoulEV','Msgsev');
```

Una vez creada la tabla de datos, se usa una base de datos CAN que podría obtenerse del fabricante, en este caso se la creó desde cero con los datos obtenidos en el análisis de la trama de datos CAN con las herramientas usadas en este estudio.

Para la creación de una base de datos CAN se utilizó la herramienta NI-XNET Database Editor que nos facilita la creación de un clúster que contenga los datos de los módulos de tracción eléctrica, RPM y velocidad en tres diferentes estructuras. Estas estructuras contienen sus respectivas señales ya identificadas en el estudio, esta red se puede apreciar en la Fig. 12.

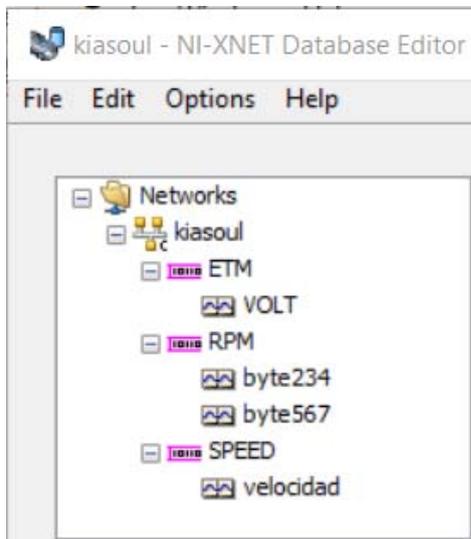


Fig. 12. Estructura de una base de datos CAN.

Cada estructura, como se puede observar en la Fig. 13, tiene sus propiedades. Lo único que cabe recalcar en esta sección es que el ID se lo debe colocar en decimal.

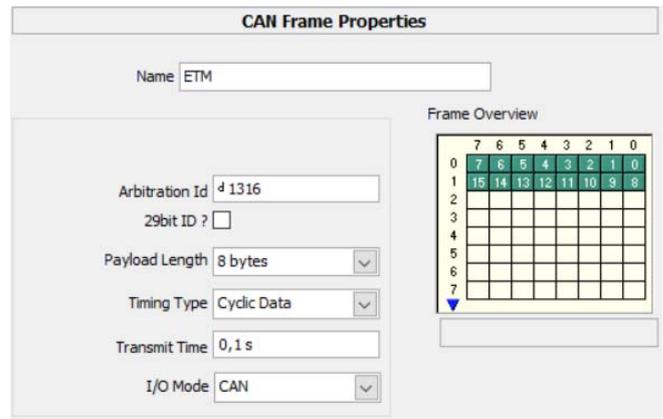


Fig. 13. Propiedades de la trama de datos CAN.

Dentro de las señales, se debe configurar los parámetros que contengan los bits de datos, como ejemplo, en la Fig. 14, se observa la velocidad, que es una señal estática duplicada, razón por la cual usamos un valor de 0.5 en el factor de escala y como el vehículo no rebasa los 200 Km/h, usamos un valor máximo de 200 con unidades de Km/h. Se configura la posición del bit de inicio y el número de bits que conforman el valor de los datos obtenidos, además, se configura el signo de los datos y orden de los bytes.

Una vez realizada la configuración de nuestra base de datos, se exporta la información a un archivo dbc haciendo click derecho sobre la cabecera del clúster y eligiendo “Export to DBC”.

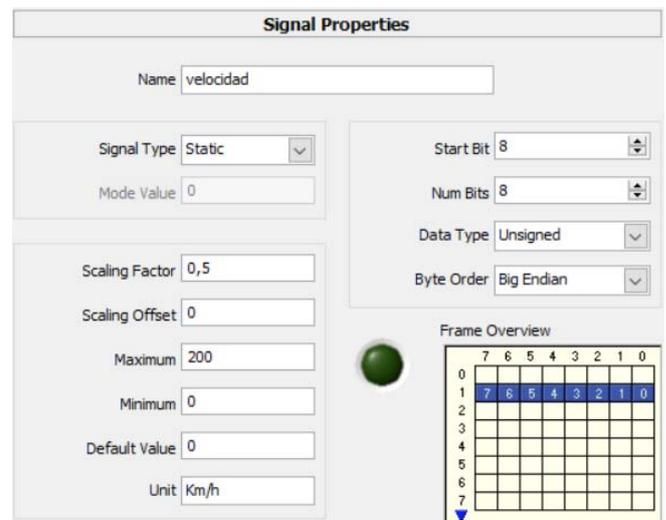


Fig. 14. Propiedades de las señales de la trama de datos CAN.

A partir de la creación de la tabla compatible con el bloque CAN Replay y la elaboración de la base de datos, se crea un canal virtual para el envío de la trama CAN a los bloques CAN Receive. Esta programación se puede observar en la Fig. 15.

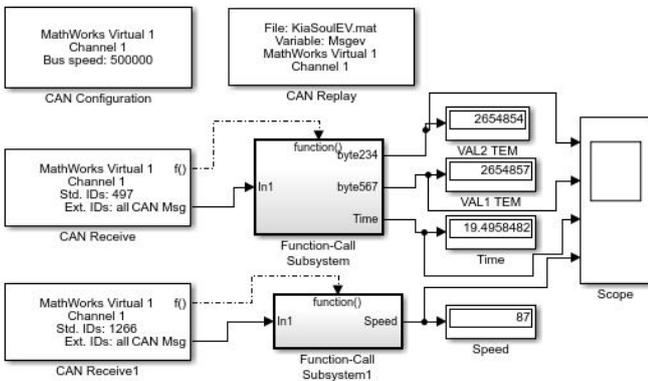


Fig. 15. Análisis de datos CAN en Simulink.

Simulink genera la conversión directa según la base de datos CANdb que se ingrese, para este caso, se ingresa nuestra base de datos generada en el bloque CAN Unpack dentro del Function-Call Subsystem. Esta programación se observa en la Fig. 16, que corresponde al módulo de tracción eléctrica y la comparación de niveles de voltaje.

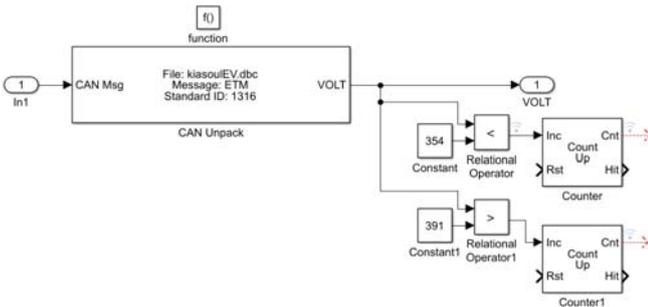


Fig. 16. Obtención de los valores del voltaje de entrada del inversor.

### B. Análisis sobre la marcha

El Software de análisis de la trama de datos CAN sobre la marcha, facilita la comprensión del comportamiento del vehículo, para desarrollar el modelo de detección de fallas dinámico. En la Fig. 17 se observa la programación en LabVIEW, y la configuración de la tarjeta de adquisición y la obtención de los datos con bloques de la librería NI-XNET.

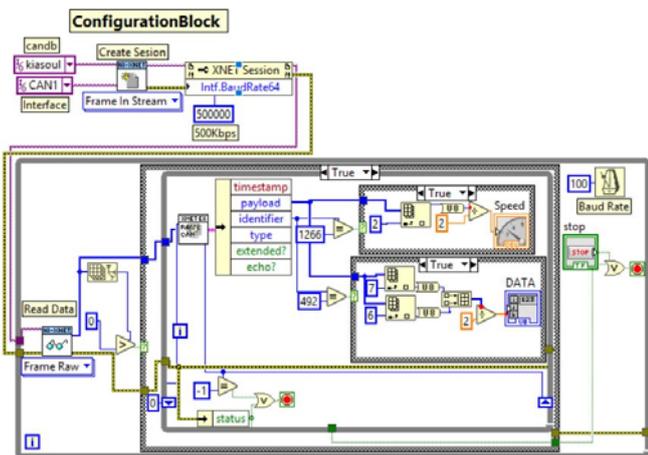


Fig. 17. Programación para lectura de datos CAN en LabVIEW.

En la Fig. 18 se observa la programación del modelo en Simulink. El uso de la base datos kiasoulEV.dbc también es usada en esta programación. En este caso, se configura directamente la tarjeta de adquisición de datos NI-9862 como la interface, facilitando así la programación.

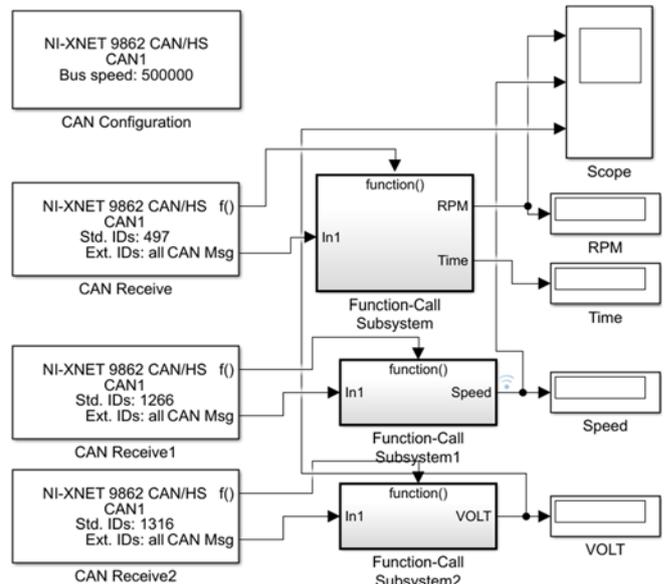


Fig. 18. Programación para lectura de datos CAN en Simulink.

## V. VALIDACIÓN DEL MODELO DINÁMICO

Para la validación del modelo dinámico, se usa el escenario de en marcha en Simulink. Se conecta la tarjeta al vehículo como se muestra en la Fig. 19, ocasionando una falla con una herramienta de diagnóstico conectada en paralelo a la tarjeta de adquisición de datos.

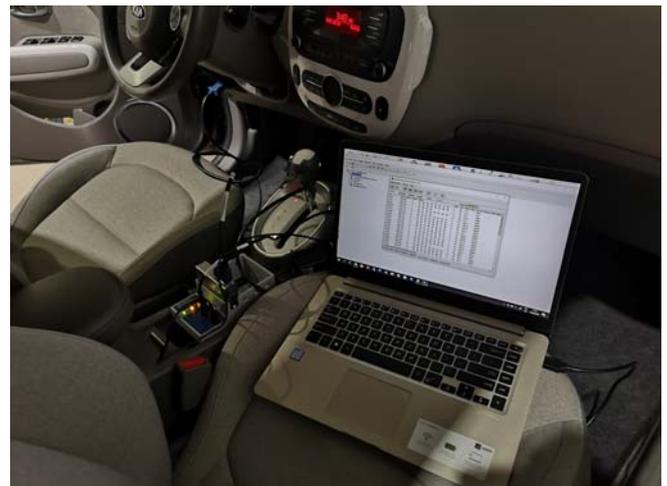


Fig. 19. Adquisición de la trama de datos CAN.

El voltaje se compara con los valores de velocidad y RPM como se muestra en la Fig. 20. El voltaje varía entre 354-390V en funcionamiento normal. Se tomó una muestra de una ruta corta, donde se alcanzó los 100 km/h para verificar que el rango de voltaje antes valorado son los valores de normal funcionamiento de las baterías.

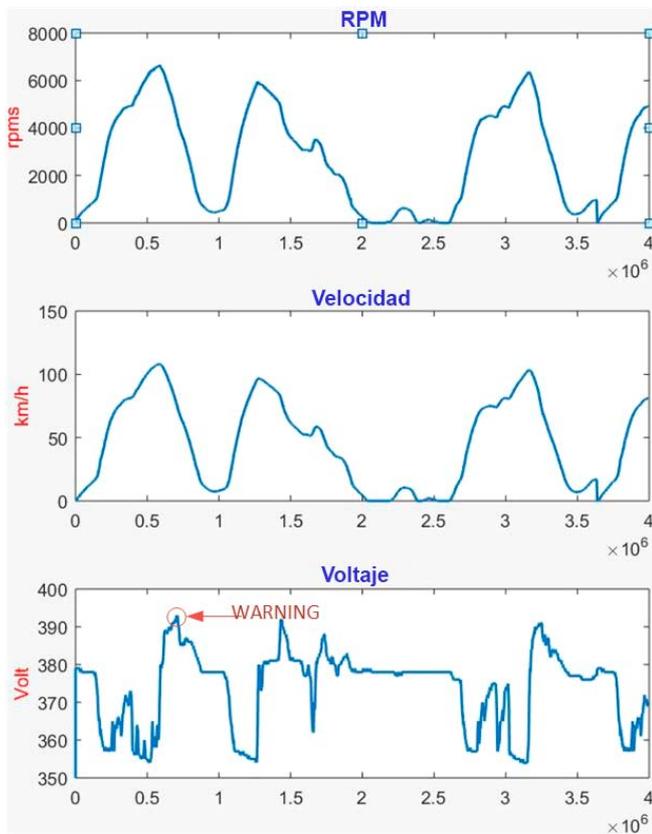


Fig. 20. Evento de warning en niveles de voltaje.

Un indicador visual que consiste en un cuadro de advertencia que aparece frente al panel del programa nos advierte de un posible alto voltaje. En la Fig. 21, se puede observar el funcionamiento de este indicador de alto voltaje al ocurrir un evento de fuera de rango de los niveles normales.

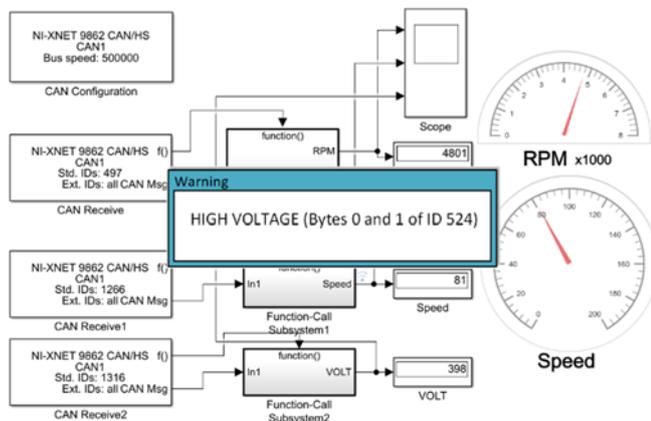


Fig. 21. Bloques de programa en un evento de alto voltaje.

Un evento de nivel bajo de voltaje se muestra en la Fig. 22. Se ha simulado un error en el voltaje de entrada del inversor al generar una caída brusca de voltaje desde una herramienta de diagnóstico.

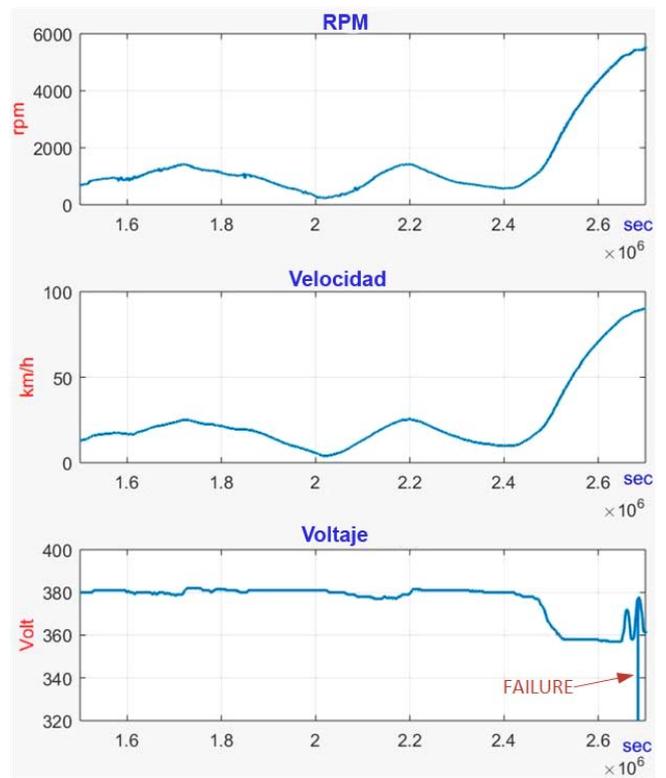


Fig. 22. Nivel de voltaje bajo en la entrada del inversor.

Cuando el error ocurre, la herramienta emite alertas en tiempo real con un cuadro de diálogo, mientras que el análisis gráfico verifica los valores de RPM y velocidad como se muestra en la Fig. 23.

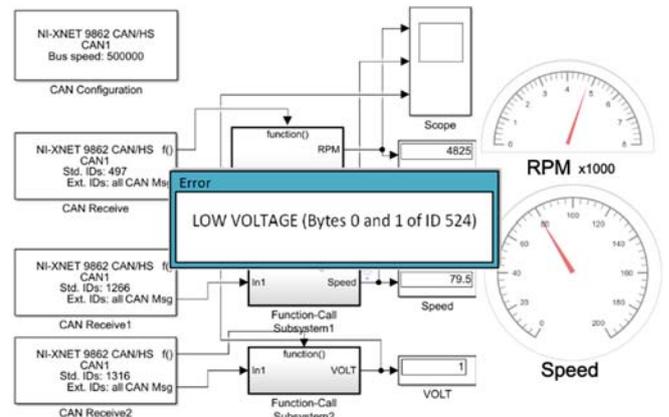


Fig. 23. Mensaje de error en un evento de bajo voltaje.

La herramienta desarrollada en LabVIEW se utiliza para el análisis de la trama de datos CAN del archivo obtenido en las pruebas con automóviles. Esta es la razón por la cual no hay verificación de modelo con el vehículo en marcha. Sin embargo, la herramienta en LabVIEW, para el modelo sobre la marcha, también evalúa el vehículo mientras se está operando.

El modelo dinámico está diseñado para el estudio de varios identificadores que cambian los valores de referencia establecidos en el mismo y con las herramientas desarrolladas en LabVIEW dedicadas a la comprensión de los datos.

Este artículo presenta la validación del modelo dinámico para las comunicaciones en redes presentes en el vehículo Kia Soul Electric con el protocolo CAN.

Es posible obtener toda la información que proporcionan los módulos que integran un vehículo eléctrico Kia Soul a través del protocolo de comunicación CAN por medio de la tarjeta NI-9862 para su posterior análisis con el uso de software que contiene bibliotecas especializadas en el estudio de redes CAN. En este estudio se utilizaron las bibliotecas CAN de Simulink y NI-XNET de LabVIEW.

Se puede realizar un estudio de los datos directamente en el automóvil o en un laboratorio con las tablas y la base de datos generada. Podemos detectar en tiempo real posibles fallas generadas en el módulo de tracción eléctrica, específicamente a través del voltaje de la entrada del inversor. La herramienta obtenida nos permite validar los errores más rápido y presenta soluciones a problemas que las herramientas de diagnóstico normales no suelen detectar.

#### FUTUROS TRABAJOS

Mediante la herramienta implementada se puede generar a futuro diferentes estudios en distintos tipos de redes de vehículos presentes en la actualidad, por ejemplo, se aplicará a vehículos híbridos y de motores de combustión interna. Además, será una poderosa herramienta para estudiantes que desarrollen temas referentes a estudios de comunicación.

#### AGRADECIMIENTO

Este trabajo fue apoyado en parte por la Universidad del Azuay y su Decanato de Investigación con el Proyecto 2018-0188. Además, agradecer a la Universidad Estatal de Cuenca por facilitarnos el acceso a los Laboratorios de Balzay y al vehículo eléctrico Kia Soul.

- [1] M. Nadeem, L. Yuan, W. Ur, A. Rakhio, and S. Siddique, "Diagnostic Tool And Remote Online Diagnostic System For Euro Standard Vehicles," pp. 415–419, 2017.
- [2] E. Esteban and F. Palomeque, "Estudio de la Red CAN y sus Diversas Evoluciones en Vehículos Convencionales e Estudio de la Red CAN y sus Diversas Evoluciones en Vehículos Convencionales e Híbridos , con el Propósito de Realizar un Diagnóstico Acertado Tomando Como Base sus Protocolos," no. November 2014, 2015.
- [3] J. Nan, L. Zai, Z. Wang, and J. Wang, "Bus communication and control protocol using the electric passenger car control system," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2006, vol. 2, pp. 8288–8291.
- [4] J. Camara, T. A. Cerqueira, and V. L. da Silva, "Development of an Automotive Scanner for Educational Application," *IEEE Lat. Am. Trans.*, vol. 15, no. 1, pp. 40–47, 2017.
- [5] L. Y. Wang, L. F. Wang, W. Liu, and Y. W. Zhang, "Research on fault diagnosis system of electric vehicle power battery based on OBD technology," in *2017 International Conference on Circuits, Devices and Systems, ICCDS 2017*, 2017, vol. 2017-Janua, pp. 95–99.
- [6] W. Jianfeng, W. Dafang, and X. Jie, "The design of electric motor car's body network based on CAN-bus distributed control," in *2009 Chinese Control and Decision Conference, CCDC 2009*, 2009, pp. 3712–3717.
- [7] X. Chen, J. Wu, Y. Zhao, and H. T. Bai, "Design of CAN Communication Network with Distributed Control Systems for Pure Electric Vehicle," *Adv. Mater. Res.*, vol. 791–793, pp. 647–651, 2013.
- [8] J. M. Bustillo, "REDES Y MULTIPLEXADOS," pp. 1–22.