



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGIA
ESCUELA DE INGENIERÍA AUTOMOTRIZ

**IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE RED
CAN BASADO EN IDENTIFICADORES PARA UN VEHÍCULO
ELÉCTRICO RENAULT KANGOO ZE.**

Trabajo previo a la obtención del título de:
INGENIERO EN MECÁNICA AUTOMOTRIZ.

Nombre del autor:
Jorge Luis Pinguil Peñaranda.

Nombre del director:
Ing. Efrén Esteban Fernández Palomeque PhD.

Cuenca – Ecuador

2023

DEDICATORIA

Quiero dedicar este trabajo a mi Mamá que me ha enseñado a no rendirme y a superarme cada día. También a mi Papá que ha sido un profesor más para mí durante toda mi carrera universitaria, igualmente me ha enseñado que siempre existe una solución a los problemas. Para mis hermanos que siempre se han preocupado por mí y me han apoyado. Además, para mis abuelos que los quiero mucho y para toda mi familia que siempre están al pendiente de mí.

AGRADECIMIENTOS

Quiero agradecer a la enseñanza y tutoría brindada durante mi trabajo de titulación al Ing. Efrén Fernández Palomeque PhD.

Además, agradezco al Ing. Mateo Chalco tribunal del trabajo. También al Ing. Daniel Cordero y al Ing. Diego Rojas por la ayuda con equipos electrónicos para la toma de datos CAN.

Muchas gracias a la ayuda de mis compañeros durante la conducción del vehículo eléctrico.

De todo corazón gracias al apoyo incondicional de Sammy durante estos años en la Universidad, y a las personas que me han brindado su ayuda en todo este proceso.

IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE RED CAN BASADO EN IDENTIFICADORES PARA UN VEHÍCULO ELÉCTRICO RENAULT KANGOO ZE.

Resumen. El protocolo CAN es un protocolo de comunicación serial orientado al control distribuido en tiempo real, envía y recibe por medio de nodos al Bus CAN identificadores de mensajes. Cada módulo del vehículo eléctrico envía información por medio de señales eléctricas, que cada nodo interpreta en forma de mensajes en una sucesión de bits, en 0 lógico dominante y 1 lógico recesivo. Se transmite la trama de datos orientado siempre al mensaje prioritario, obteniendo cada identificador de mensaje con la ayuda de una tarjeta de adquisición NI-9862 y monitoreado en el programa NI XNET Bus Monitor, al final de cada lectura se guarda en un archivo de texto que se evaluará en archivos CSV. De esta forma se interpreta los bytes del identificador realizando conversiones, usando un byte o una combinación de bits para descifrar el mensaje. Si se aplica este método podremos entender la comunicación CAN y poder generar un diagnóstico con los identificadores encontrados, en este caso de un vehículo eléctrico Renault Kangoo ZE año 2015.

Palabras claves: Identificadores, CAN, bytes, bits.



Ing. Robert Rockwood Iglesias. MSc
Director de la Escuela de Ingeniería Automotriz



Ing. Efrén Fernández Palomeque. PhD
Director de Tesis

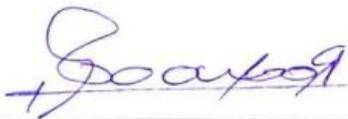


Jorge Luis Pinguil Peñaranda
Autor

IMPLEMENTATION OF A CAN NETWORK MONITORING SYSTEM BASED ON IDENTIFIERS FOR A RENAULT KANGOO ZE ELECTRIC VEHICLE.

Abstract. _ The CAN protocol is a serial communication protocol oriented to distributed control in real-time. It sends and receives message identifiers through nodes to the CAN Bus. Each module of the electric vehicle sends information using electrical signals. Each node interprets in the form of messages in a succession of bits, in logical 0 dominant and logical 1 recessive. The data frame is transmitted always oriented to the priority message, obtaining each message identifier with the help of an NI-9862 acquisition card monitored in the NI XNET Bus Monitor program, at the end of each reading it is saved in a text file that will be evaluated on CSV files. In this way, the bytes of the identifier are interpreted by performing conversions, using a byte or a combination of bits to decrypt the message. If this method is applied, we will be able to understand the CAN communication and be able to generate a diagnosis with the identifiers found, in this case of a Renault Kangoo ZE electric vehicle from 2015.

Keywords: Identifiers, CAN, bytes, bits



Ing. Robert Rockwood Iglesias. MSc
Director of the School Engineering Automotive



Ing. Efrén Fernandez Palomeque. PhD
Thesis Director



Jorge Luis Pinguil Peñaranda
Author

Translated by:



Jorge Pinguil

Índice

DEDICATORIA	2
AGRADECIMIENTOS	3
Resumen.	9
Abstract. _	9
I. INTRODUCCIÓN	9
II. METODOLOGÍA USADA	10
III. IDENTIFICADORES	11
POSICIÓN DEL PEDAL DEL ACELERADOR	12
VELOCIDAD DEL VEHÍCULO	13
CORRIENTE DE LA BATERÍA DE TRACCIÓN	13
VELOCIDAD DEL MOTOR ELÉCTRICO	14
ESTADO DE CARGA DE LA BATERÍA DE TRACCIÓN (SOC)	15
ESTADO DE SALUD DE LA BATERÍA DE TRACCIÓN (SOH)	16
IV. DISCUSIÓN	16
V. CONCLUSIONES	19
VI. REFERENCIAS	19

Índice de tablas

Tabla 1. Trama de datos Renault Kangoo ZE	11
Tabla 2. Conversión de bytes para interpretar datos	11
Tabla 3. Mensajes CAN de ID 0x385	12
Tabla 4. Bytes de ID 0x385.....	12
Tabla 5. Mensajes CAN de ID 0x504	13
Tabla 6. Bytes de ID 0x504.....	13
Tabla 7. Mensajes CAN de ID 0x509	13
Tabla 8. Bytes de ID 0x509.....	13
Tabla 9. Mensajes CAN de ID 0x644	14
Tabla 10. Bytes de ID 0x644	14
Tabla 11. Mensajes CAN de ID 0x1620	15
Tabla 12. Bytes de ID 0x1620	15
Tabla 13. Bytes de ID 0x1624	16
Tabla 14. Bytes del SOH actual	19

Tabla de figuras

Figura 1. Estado dominante y recesivo	9
Figura 2. Trama de mensajes CAN de Volkswagen Passat B6 2005	10
Figura 3. Pasos de la metodología propuesta	10
Figura 4. Configuración NI XNET Bus Monitor	10
Figura 5. Trama de datos del EV	11
Figura 6. Filtrado de datos CAN en archivo CSV	11
Figura 7. Posición del pedal del acelerador durante ruta	12
Figura 8. Velocidad del EV durante ruta	13
Figura 9. Corriente de la batería de tracción durante ruta	14
Figura 10. Revoluciones por minuto del EV durante ruta	15
Figura 11. Porcentaje estado de carga durante ruta	15
Figura 12. Porcentaje estado de salud EV	16
Figura 13. Tarjeta de adquisición de datos NI 9862 conectado al EV monitoreado por programa XNET Bus Monitor	17
Figura 14. Bits de relleno de ID 0x394.....	17
Figura 15. Posición de pedal acelerador en modo de conducción ECO y normal.....	17
Figura 16. Velocidad EV en modo de conducción ECO y normal	17
Figura 17. Rpm de EV en modo de conducción ECO y normal	17
Figura 18. Consumo de corriente de batería en modo de conducción ECO y normal	18
Figura 19. Funcionamiento Económetro de EV	18
Figura 20. Ventana de seguridad de las celdas	18
Figura 21. Algoritmo de funcionamiento de una BMS	18
Figura 22. SOC en dos modos de conducción y porcentaje de km por recorrer con estado de carga	18
Figura 23. Cluster activado modo de conducción ECO	18
Figura 24. Trama de datos de ID 0x1620 con enchufe de carga conectado	19
Figura 25. Tiempo hasta completar la carga del EV con enchufe de carga conectado	19
Figura 26. SOH anterior y actual	19

IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE RED CAN BASADO EN IDENTIFICADORES PARA UN VEHÍCULO ELÉCTRICO RENAULT KANGOO ZE.

Jorge Pinguil
Universidad del Azuay
Escuela de Ingeniería Mecánica Automotriz
Cuenca, Ecuador
jorgeluis@es.uazuay.edu.ec

Efrén Fernández
Universidad del Azuay
Escuela de Ingeniería Mecánica Automotriz
Cuenca, Ecuador
efernandez@uazuay.edu.ec

Resumen. El protocolo CAN es un protocolo de comunicación serial orientado al control distribuido en tiempo real, envía y recepta por medio de nodos al Bus CAN identificadores de mensajes. Cada módulo del vehículo eléctrico envía información por medio de señales eléctricas, que cada nodo interpreta en forma de mensajes en una sucesión de bits, en 0 lógico dominante y 1 lógico recesivo. Se transmite la trama de datos orientado siempre al mensaje prioritario, obteniendo cada identificador de mensaje con la ayuda de una tarjeta de adquisición NI-9862 y monitoreado en el programa NI XNET Bus Monitor, al final de cada lectura se guarda en un archivo de texto que se evaluará en archivos CSV. De esta forma se interpreta los bytes del identificador realizando conversiones, usando un byte o una combinación de bits para descifrar el mensaje. Si se aplica este método podremos entender la comunicación CAN y poder generar un diagnóstico con los identificadores encontrados, en este caso de un vehículo eléctrico Renault Kangoo ZE año 2015.

Palabras claves: Identificadores, CAN, bytes, bits.

Abstract. _ The CAN protocol is a serial communication protocol oriented to distributed control in real time, it sends and receives message identifiers through nodes to the CAN Bus. Each module of the electric vehicle sends information by means of electrical signals, which each node interprets in the form of messages in a succession of bits, in logical 0 dominant and logical 1 recessive. The data frame is transmitted always oriented to the priority message, obtaining each message identifier with the help of an NI-9862 acquisition card monitored in the NI XNET Bus Monitor program, at the end of each reading it is saved in a text file that will be evaluated on CSV files. In this way, the bytes of the identifier are interpreted by performing conversions, using a byte or a combination of bits to decrypt the message. If this method is applied, we will be able to understand the CAN communication and be able to generate a diagnosis with the identifiers found, in this case of a Renault Kangoo ZE electric vehicle from 2015.

Keywords: Identifiers, CAN, bytes, bits.

I. INTRODUCCIÓN

La comunicación del protocolo CAN bus fue desarrollado por la empresa BOSCH en 1985 [1], CAN por sus siglas en inglés (Controller Area Network) es un protocolo de comunicación serial orientado a control distribuido en tiempo real. Es decir, todos los nodos de la red son capaces de enviar y recibir datos, así como solicitar información de cualquier otro nodo de la red [2]. Los vehículos en la actualidad cuentan con una gran cantidad de nodos que son los encargados de intercambiar la información de diversos módulos del automóvil, así como el motor, clúster, ABS, bolsas de aire, control de tracción, baterías, entre otros [3].

Los nodos intercambian información en forma de señales eléctricas mediante un Bus de dos cables CAN_H (high) y CAN_L (low), por lo tanto, existe dos estados de señales, en estado recesivo los dos canales tienen un voltaje igual de 2.5V y en estados recesivo existe una diferencia de 1V como se observa en la Fig. 1, estas señales al intercambiar información entre nodos se convierten en código binario para interpretar el mensaje [3].

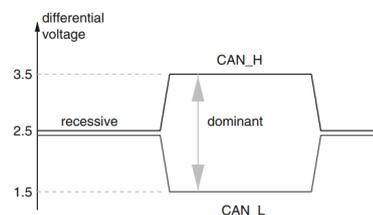


Fig. 1 Estado dominante y recesivo.

Es primordial definir los niveles eléctricos que usan para decodificar la información recibida, utilizando los conceptos de bit dominante y recesivos. El bit dominante es un 0 lógico presente en algún nodo de la red y es nombrado bit dominante debido a que hace desaparecer los 1 lógicos en la red, mientras que, un bit recesivo es un 1 lógico [2].

Los vehículos eléctricos están ganando mucha aceptación debido a su baja contaminación y un consumo energético bajo. Estos vehículos eléctricos necesitan más módulos, sensores y actuadores, comparándolos con un vehículo convencional, por consiguiente, trabaja con una red de alta velocidad de 250Kbps para la comunicación de módulos que necesitan

control inmediato y una red de baja velocidad de 125 Kbps conectado ambas redes por medio de un Gateway que conectan las unidades de control eléctrico [4].

La comunicación de los nodos de la red se realiza mediante mensajes en una sucesión de bits y una serie de campos para la comunicación entre nodos. Estos mensajes permiten identificar al nodo, indicando el principio y final del mensaje y mostrar los datos. Los mensajes son recibidos por todos los nodos, pero este protocolo CAN se oriente en el mensaje prioritario mas no en el destinatario, entontes solo usará los que necesite. En un caso de transmisión de dos o más mensajes simultáneamente, la prioridad tendrá el identificador (ID) más bajo [5].

CAN transmite los llamados trama de datos como se observa en la Fig. 2 que consisten en un campo de prioridad que cuenta con el ID del mensaje, además se encarga que el mensaje prioritario sea siempre el nodo con el ID menor, la longitud (length) nos indica el número de bytes que contiene el campo de datos, campo de datos (payload) donde se encuentran los datos del mensaje, una suma de verificación CRC (checksum) seguida de un indicador de reconocimiento que enviará un mensaje llamado redundancia cíclica junto al ID de mensaje desde un nodo emisor a un nodo receptor que lo interpretará y de haber un problema lo marcará como un mensaje erróneo y ACK (acknowledgment) indicará si la trama de datos ha sido recibida [2][6].

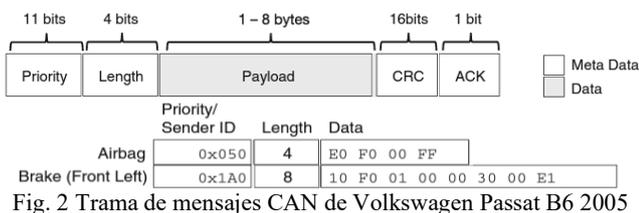


Fig. 2 Trama de mensajes CAN de Volkswagen Passat B6 2005 [6].

El trabajo pretende aportar con una información dinámica para el estudio del protocolo CAN basándose en sus identificadores como parte fundamental en un vehículo eléctrico marca Renault Kangoo ZE año 2015, la poca información y el acceso privado a la base de datos de identificadores de cada nodo abre paso a una serie de dudas. La recopilación de datos se realizará mediante una tarjeta de adquisición NI-9862 que ayudará al monitoreo de los mensajes de identificadores durante algunos recorridos en el EV (Vehículo eléctrico), que posteriormente se desarrollará una metodología para diferenciar cada ID y evaluar la data que contendrá una serie de bytes en un sistema hexadecimal que será deducido y explicado. La información pretende abrir el campo de estudio en la red CAN en los vehículos y el uso de sus identificadores para la comunicación de alta velocidad entre sus nodos.

II. METODOLOGÍA USADA

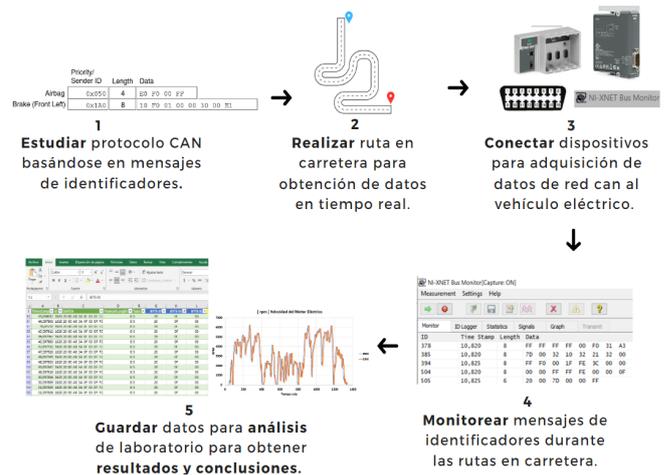


Fig. 3 Pasos de la metodología propuesta.

Se realizó varias vueltas de la misma ruta dentro de la ciudad de Cuenca, Ecuador en diferente horario, distinto porcentaje de batería, y distinto tipo de conducción Económico (ECO) y normal. Para la obtención de los datos CAN del vehículo eléctrico es necesario la configuración de la tarjeta de adquisición de datos NI-9862, conectado al vehículo por su puerto de OBD2 y monitoreado en el programa NI- XNET Bus Monitor a un Baud Rate de 500 Kbaud como se observa en la Fig. 4 y 5 con el EV en movimiento. Durante los recorridos el monitoreo es primordial ya que los valores en diferentes bytes hacen cambios progresivos como por el ejemplo el byte 04 del ID 0x385 del pedal del acelerador, a medida que se pisa el pedal aumenta el valor hexadecimal del byte, por ende, hay que estar atentos a los cambios que se producen en cada byte de cada ID en los recorridos.

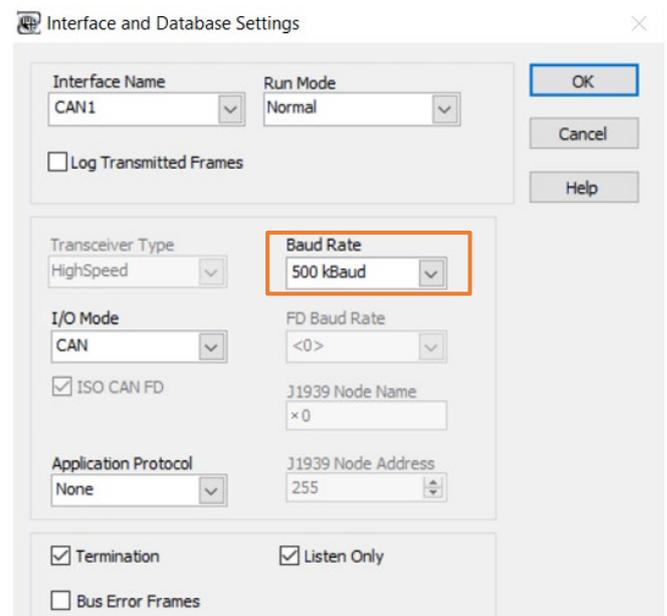


Fig. 4 Configuración NI XNET Bus Monitor.

El vehículo eléctrico Renault Kangoo Z.E año 2015 consta de 42 Identificadores que intercambian información en su red, en la Fig. 5 se observa la trama de datos con la cantidad de identificadores que contiene el vehículo eléctrico.

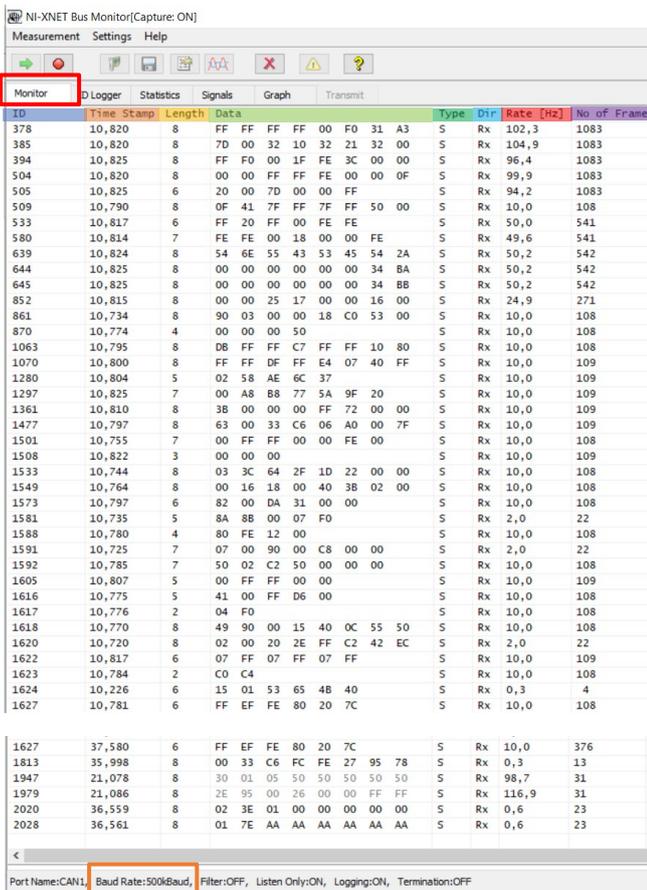


Fig. 5 Trama de datos del EV.

El programa NI-XNET Bus monitor cuenta con la opción “Monitor” como se observa en el recuadro rojo de la Fig. 5, que será el encargado de la lectura, envío y recepción de los mensajes de la red CAN del EV. En la Fig. 5 se observa las siguientes columnas: ID correspondiente al identificador del mensaje, el tiempo de interfaz XNET de recepción de datos, la longitud de bytes del campo de datos, tipo de trama de datos en este caso “S” estándar esto quiere decir que el ID del mensaje es de 11 bits y consta de hasta 2048 valores diferentes [2], Dir es la recepción (Rx) de datos, Rate (Hz) la frecuencia de envío y la columna final el número de tramas enviados.

Al final de cada recorrido se obtendrá un archivo de texto con la trama de datos CAN, para ser evaluados en un archivo CSV. Al tener una lectura de datos a alta velocidad con tiempo de interfaz de 0,1s con vueltas de recorridos alrededor de 23min se necesita evaluar individualmente cada ID como por la gran cantidad de datos como se observa en la Fig. 6. El archivo de texto al final del recorrido se guardará con las siguientes columnas: Tiempo, ID, Longitud y Data.

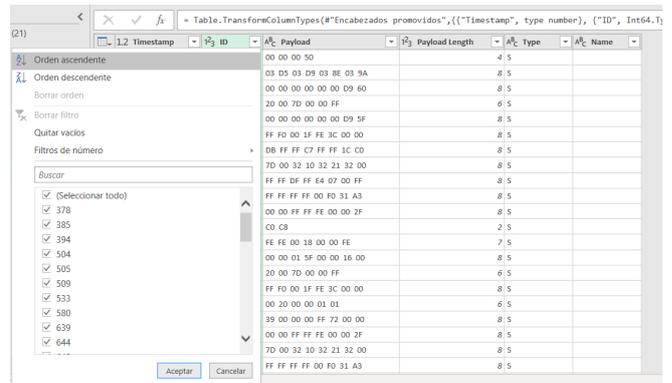


Fig. 6 Filtrado de datos CAN en archivo CSV.

De esa forma se inicia la búsqueda de IDs importantes presentes en el EV, los bytes del campo de datos son expresados en sistema hexadecimal, por lo tanto, separamos cada byte dependiendo su longitud en distintas columnas como se observa en la Tabla 1 y 2, para posteriormente ser transformados a un sistema decimal y ser expresados en valores entendibles. En ocasiones dos bytes trabajan en conjunto por ese motivo es necesario ser transformados en sistema binario, y de esta manera saber cuál de los 8 bits de cada byte está en un estado dominante enviando información a los nodos del vehículo eléctrico.

Tiempo [s]	ID	Longitud	Data
0,03168	0x1620	8	02 00 20 2D FF C2 32 EC
46,03062	0x1620	8	02 00 20 2D FF C2 22 EC
70,530027	0x1620	8	02 00 2C 2C FF C2 22 F0
813,51319	0x1620	8	02 00 2C 21 FF C1 92 DC
1401,999497	0x1620	8	02 00 2C 1D FF C1 72 BC

Tabla 1. Trama de datos Renault Kangoo ZE.

Data ID: 0x1620									
Sist.	Tiempo [s]	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Hex.	0.03168	02	00	20	2D	FF	C2	32	EC
Bin.	0.03168	0001	0000	0010 0000	00101101	1111 1111	1100 0010	0011 0010	1110 1100

Tabla 2. Conversión de bytes para interpretar datos.

III. IDENTIFICADORES

Los archivos de base de datos CAN que se guardan al final de cada recorrido son archivos de texto que contienen la información de marcos y señales del EV, cada red contiene un archivo base de datos único [7], las marcas de vehículos restringen el acceso a sus IDs por seguridad, además cada marca de vehículo define sus propios ID de mensaje lo que dificulta su localización [8]. Por lo tanto, haremos uso de la metodología propuesta para la localización de los identificadores. Al ser un vehículo eléctrico necesita más módulos, sensores o actuadores que un vehículo convencional, el vehículo eléctrico Renault Kangoo Z.E 2015 posee 42 ID de mensajes encargados de intercambiar información entre los nodos de la red CAN, al ser una cantidad considerable nos centraremos en algunos importantes.

Cada byte posee 8 bits que representa un rango hexadecimal de 00 a FF, es decir 255 valores distintos en decimal. Entonces según la metodología separamos cada byte, bit o combinación de bytes que durante el monitoreo en carretera visualizando cambios con patrones identificables, y realizar la evaluación en un archivo CSV mediante gráficos de dispersión contra el tiempo de cada byte individualmente para identificar patrones así mismo reconocibles. La combinación de bytes transmitidos por la red CAN por lo general deben ser interpretados en valores convencionales para ser entendidos, por lo que el uso de fórmulas de conversión para los datos hexadecimales es muy necesario.

Iniciamos con localización de los siguientes ID de mensaje presentes en el EV Renault Kangoo Z.E 2015:

POSICIÓN DEL PEDAL DEL ACELERADOR

Durante un recorrido de 1402,33805s se obtuvo alrededor de 140239 datos del ID 0x385, el pedal del acelerador al estar en constante funcionamiento tendrá un identificador de baja denominación para ser un mensaje prioritario. En el ID 0x385 se visualiza cambios progresivos en el byte 04 al momento de presionar el pedal del acelerador, como se observa en la Tabla 3 durante un recorrido en el VE.

Tiempo [s]	ID	Longitud	Data
0,000958	0x385	8	7D 00 32 10 32 21 32 00
347,423065	0x385	8	7D 00 38 57 39 20 32 00
726,534452	0x385	8	7D 00 46 6A 49 20 32 00
1010,298399	0x385	8	7D 00 2C 34 2D 20 37 00
1402,338805	0x385	8	7D 00 32 10 32 21 32 00

Tabla 3. Mensajes CAN de ID 0x385.

Entonces constará con los siguientes detalles para su conversión en valores entendibles:

ID 0x385

Primero ubicamos el Byte 04 en la trama de datos de la Tabla 4 y tomamos todos los datos para trazarlos en una gráfica de dispersión contra el tiempo de ruta.

Data ID: 0x385									
Sist.	Tiempo [s]	Byte 07	Byte 06	Byte 05	Byte 04	Byte 03	Byte 02	Byte 01	Byte 0
Hex.	0,03168	7D	00	32	10	32	21	32	00
Bin.	0,03168	0111 1101	0	0011 0010	0001 0000	0011 0010	0010 0001	0011 0010	0

Tabla 4. Bytes de ID 0x385.

Tomamos en cuenta los siguientes detalles durante el inicio del recorrido antes de presionar el pedal, durante el recorrido y al final. Obteniendo los datos de la Tabla 3 usaremos los valores hexadecimales y los interpretamos de la siguiente manera:

- Posición inicial pedal= 10 Hexadecimal= 16 Decimal
- Posición máxima= FD Hexadecimal= 253 Decimal
- Límites posición de pedal: 0% - 100% - 125%

Los valores iniciales antes de presionar el pedal nos indican después de su conversión hexadecimal a decimal valores iniciales desde 16 siendo el único valor de menor denominación, por lo tanto, podemos restar 16 al byte 04 para interpretarlo como un porcentaje inicial igual a 0%. Al verificar los datos con la posición del pedal al máximo FD es igual a 253 y si lo restamos no nos da un valor en porcentaje de 125, por lo que es necesario dividir todos los valores para 189,6 para obtener valores en porcentaje entre 0 (0%), 1 (100%) y 1,25 (125%). Entonces nos regiremos de la siguiente fórmula.

Fórmula usada:

$$\%pedal = \frac{byte04 - 16}{189,6}$$

Entonces graficamos todos los datos para tener una noción de todas las posiciones que ha tomado el pedal del acelerador durante todo el recorrido como se observa en la Fig.7 y de igual manera con la ayuda de un OBD2 marca KlavKarr 210 nos aseguramos que la gráfica según la fórmula propuesta se encuentra correcta.

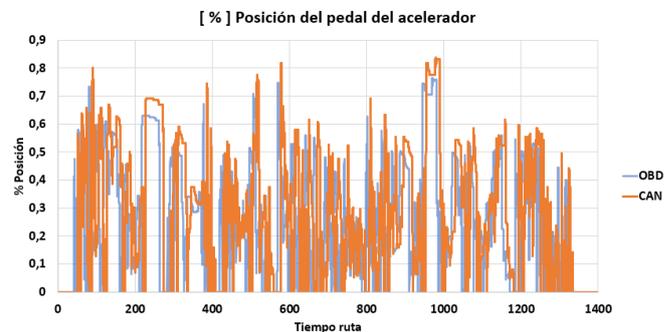


Fig. 7 Posición del pedal del acelerador durante ruta.

De esta manera podemos tomar los datos del byte 04 de los datos de la Tabla 3 e interpretarlos con la gráfica de dispersión de la Fig. 7 obteniendo que la posición del pedal durante el recorrido han sido las siguientes:

Tiempo: 0,000958s

Hex=10; Dec=16

$$\%pedal = \frac{16 - 16}{189,6} = 0\%$$

Tiempo: 347,423065s

Hex=57; Dec=87

$$\%pedal = \frac{87 - 16}{189,6} = 0,37 = 37\%$$

Tiempo: 726,534452s

Hex=6A; Dec=106

$$\%pedal = \frac{106 - 16}{189,6} = 0,47 = 47\%$$

VELOCIDAD DEL VEHÍCULO

En el mismo recorrido durante el mismo intervalo de tiempo se tomó alrededor de 140239 datos, al ser un ID de baja denominación envía y recepta un gran número de tramas con mayor frecuencia de envío. En el byte 02 del ID 0x504 tiene un avance continuo de sus valores hexadecimales, por lo que se toman los datos según la Tabla 5 de este identificador para podernos apoyarnos de un gráfico referencial que nos muestra cambios identificables de velocidad.

Tiempo [s]	ID	Longitud	Data
0,001209	0x504	8	00 00 FF FF FE 00 00 0F
169,427405	0x504	8	00 00 FF FF FE 35 A0 2F
398,162164	0x504	8	00 00 FF FF FE 1A E0 2F
870,461381	0x504	8	00 00 FF FF FE 27 60 2F
1128,845587	0x504	8	00 00 FF FF FE 46 E0 2F

Tabla 5. Mensajes CAN de ID 0x504.

Detalles de identificador:

ID 0x504

Localizamos el ID en la trama de datos separándolos en sus diferentes bytes según su longitud, como se observa en la Tabla 6 para posteriormente graficarlos en un gráfico contra el tiempo.

Data ID: 0x504									
Sist.	Tiempo [s]	Byte 07	Byte 06	Byte 05	Byte 04	Byte 03	Byte 02	Byte 01	Byte 0
Hex.	169,427405	00	00	FF	FF	FE	35	A0	2F
Bin.	169,427405	0	0	1111 1111	1111 1111	1111 1110	0011 0101	1010 0000	0010 1111

Tabla 6. Bytes de ID 0x504.

Al momento de conversión de datos para verificar si necesitamos alguna compensación solo fue necesario el cambio de valores hexadecimales a decimales del byte 02 para encontrar la velocidad directamente en km/h, teniendo este tipo de gráfica como se observa en la Fig. 8 con las siguientes velocidades durante el recorrido en carretera. Para asegurarnos que la conversión de datos está en lo correcto se traza una gráfica contra el tiempo del OBD2 para comparar.

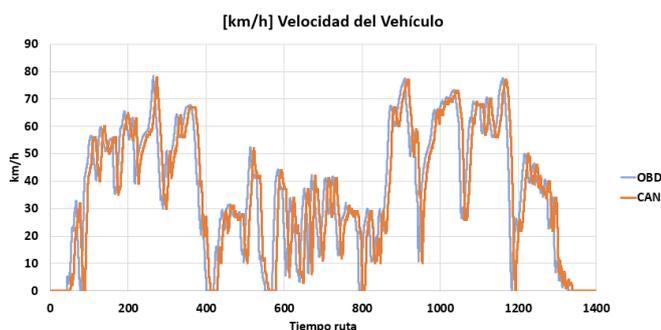


Fig. 8 Velocidad del EV durante ruta.

Así tomamos los datos del byte 02 de la Tabla 5 y los interpretamos con la Fig. 8 para verificar las velocidades del vehículo eléctrico durante diferentes tiempos.

Tiempo: 0,001209s

Hex=00; Dec=00

Velocidad = 0km/h

Tiempo: 169,427405s

Hex=35; Dec=53

Velocidad = 53km/h

Tiempo: 1128,845587s

Hex=46; Dec=70

Velocidad = 70km/h

CORRIENTE DE LA BATERÍA DE TRACCIÓN

A diferencia de los IDs de menor denominación los identificadores con mayor valor recibirán y enviarán datos con menor frecuencia, en el mismo recorrido los datos recopilados por este ID 0x509 son de 14025 datos a comparación de los anteriores. En este caso en la corriente de la batería de tracción el byte más significativo fue el byte 01 con un valor inicial de 50 hexadecimal como se observa en la Tabla 7 en el tiempo inicial 0,001691s

Tiempo [s]	ID	Longitud	Data
0,001691	0x509	8	21 41 7F FF 7F FF 50 00
169,697892	0x509	8	0A 41 7F FF 7F FF 3F 00
273,395487	0x509	8	0A 41 7F FF 7F FF 68 00
831,882783	0x509	8	0A 41 7F FF 7F FF 4D 00
1216,573848	0x509	8	0A 41 7F FF 7F FF 62 00

Tabla 7. Mensajes CAN de ID 0x509.

Detalles del identificador:

ID 0x509

Entonces ubicamos el byte 01 en la Tabla 8 y lo separamos dependiendo su longitud en valores hexadecimales como binarios según la tabla para evaluar los datos en una gráfica contra el tiempo que permita interpretar su significado.

Data ID: 0x509									
Sist.	Tiempo [s]	Byte 07	Byte 06	Byte 05	Byte 04	Byte 03	Byte 02	Byte 01	Byte 0
Hex.	273,395487	0A	41	7F	FF	7F	FF	68	00
Bin.	273,395487	1010	0100 0001	0111 1111	1111 1111	0111 1111	1111 1111	0110 1000	0

Tabla 8. Bytes de ID 0x509.

Como detalle característico de este ID convertimos todos los valores hexadecimales a valores decimales demostrando que es una gráfica de corriente comparándolo con el OBD2, pero con un dato característico que está al revés, e inicia con un valor hexadecimal igual a 50 que es igual a 80 en decimal. Por lo tanto, necesitamos que inicie desde un valor inicial

igual a 0A ya que necesitamos valores positivos como negativos, negativo cuando se acelera y positivo cuando se deja de presionar el pedal del acelerador, por lo que restamos los valores decimales menos 80 para tener diferentes signos en los valores. Pero la gráfica sigue estando al revés por lo que multiplicamos todos los valores decimales del byte 01 restados menos 80 por menos 1, la gráfica comienza a tener sentido, pero los valores no se ajustan aún por lo que una compensación será necesaria multiplicando la ecuación por 2,75 para obtener una gráfica real como se observa en la Fig. 9. Entonces nos regiremos de la siguiente fórmula.

Fórmula usada:

$$A = [(byte\ 01 - 80)(-1)](2,75)$$

Entonces hacemos uso de la fórmula propuesta para realizar la gráfica contra el tiempo como se observa en la Fig. 9 de la corriente de la batería de tracción, y verificamos si la fórmula usada se ajusta a los valores correctos comparándola con valores tomados del OBD2.

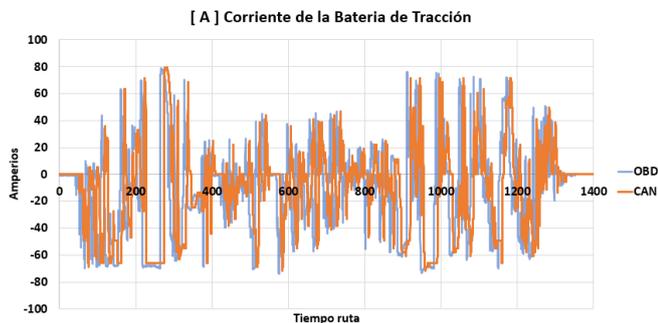


Fig. 9 Corriente de la batería de tracción durante ruta.

De esta forma tomamos los datos de la Tabla 7 y la interpretamos con la Fig. 9 usando la fórmula propuesta con los diferentes tiempos para encontrar los valores de amperaje.

Tiempo: 0,001691s

Hex=50; Dec=80

$$A = [(80 - 80)(-1)](2,75) = 0A$$

Tiempo: 169,697892s

Hex=3F; Dec=63

$$A = [(63 - 80)(-1)](2,75) = 46,75A$$

Tiempo: 273,395487s

Hex=68; Dec=104

$$A = [(104 - 80)(-1)](2,75) = -66A$$

VELOCIDAD DEL MOTOR ELÉCTRICO

En este identificador reconocemos las revoluciones por minuto del motor eléctrico, el ID de las rpm sería el valor 0x644 con una longitud de 8 bytes. Toma alrededor de 70118

datos un recorrido estimado de 23min, los cuales hemos plasmado en la Tabla 9 diferentes tiempos para descifrar los valores hexadecimales a decimales y tener valores entendibles en unidades de rpm. En este caso las rpm del motor son una combinación de bytes.

Tiempo [s]	ID	Longitud	Data
0,008396	0x644	8	00 00 00 00 00 00 F4 7A
209,267924	0x644	8	2E 22 2E 2D 16 69 D3 83
457,367079	0x644	8	12 7D 12 71 08 F7 48 DF
925,325694	0x644	8	33 92 33 9E 19 06 AE E9
1336,124277	0x644	8	02 A0 02 B8 01 4B EA 18

Tabla 9. Mensajes CAN de ID 0x644.

Detalles del identificador:

ID 0x644

Tenemos dos bytes importantes en este ID, al momento de graficar cada byte del identificador se confirma que el byte 07 y el byte 05 como se observa en la Fig. 10 tienen una gráfica identificable con valores parecidos a la velocidad, pero con valores mucho menores. Por lo que se necesitaría del uso de una conversión para interpretarlo como unidades de revoluciones por minuto.

Data ID: 0x644									
Sist.	Tiempo [s]	Byte 07	Byte 06	Byte 05	Byte 04	Byte 03	Byte 02	Byte 01	Byte 0
Hex.	457,367079	12	7D	12	71	08	F7	48	DF
Bin.	457,367079	1 0010	111 1101	1 0010	0111 0001	1000	1111 0111	0100 1000	1101 1111

Tabla 10. Bytes de ID 0x644.

Entonces tomamos el byte 07 que representa los valores más significativos de la velocidad del motor eléctrico, y lo multiplicamos por 256 para obtener un factor de escala que nos permita incrementar los valores para ajustar la curva. Podemos decir que multiplicar los valores por 256 en este tipo de fórmulas de conversión de datos CAN es por el hecho de que el byte es la unidad de medida básica de la red CAN. Entonces para poder combinar ambos bytes sumamos el byte 05 con los valores menos significativos, de esta manera obtenemos valores más grandes en la conversión. Al obtener los valores combinados de ambos bytes dividimos para 4 para ajustar la curva en la gráfica contra el tiempo y multiplicamos por 1,55 para que se ajuste correctamente como se observa en la Fig. 10. Por lo tanto, nos regimos de la siguiente fórmula.

Fórmula:

$$rpm = \left[\frac{(byte\ 07)(256) + byte\ 05}{4} \right] (1,55)$$

Hacemos uso de la fórmula planteada para realizar una gráfica contra el tiempo donde se ajustan los valores como se observa en la Fig. 10 comparándolas con un OBD2.

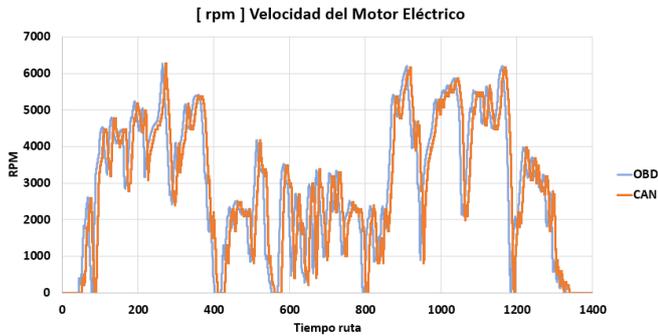


Fig. 10 Revoluciones por minuto del EV durante ruta.

Tomamos diferentes puntos dentro de la gráfica para verificar su uso.

Tiempo: 209,267924s

Byte 07: Hex=2E; Dec=46 Byte 05: Hex=2E; Dec=46

$$rpm = \left[\frac{(46)(256) + 46}{4} \right] (1,55) = 4581rpm$$

Tiempo: 209,267924s

Byte 07: Hex=33; Dec=51 Byte 05: Hex=33; Dec=51

$$rpm = \left[\frac{(51)(256) + 51}{4} \right] (1,55) = 5079rpm$$

ESTADO DE CARGA DE LA BATERÍA DE TRACCIÓN (SOC)

En este identificador es uno de los más importantes de la batería tracción, tomando alrededor de 2806 datos, pero dando varios tipos de información en este ID de mensaje. En la Tabla 11 se toma varios puntos durante el tiempo para demostrar los siguientes detalles de la ruta.

Tiempo [s]	ID	Longitud	Data
0,03168	0x1620	8	02 00 20 2D FF C2 32 EC
46,03062	0x1620	8	02 00 20 2D FF C2 22 EC
70,530027	0x1620	8	02 00 2C 2C FF C2 22 F0
813,51319	0x1620	8	02 00 2C 21 FF C1 92 DC
1401,999497	0x1620	8	02 00 2C 1D FF C1 72 BC

Tabla 11. Mensajes CAN de ID 0x1620.

Detalles del identificador:

ID 0x1620

En este caso a diferencia de los demás identificadores será el que mayor uso de sus bytes tendrá, separando cada byte de la Tabla 11 para poder encontrar valores en sistema decimal entendibles. Como parte característica de este ID transformaremos los bytes que nos da el programa como se observa en la Tabla 12 en sistema binario, usaremos los 8 bits de cada byte o combinaremos dos bytes usando ciertos bits para descifrar en valores entendibles que mensaje envía al Bus este identificador.

Data ID: 0x1620									
Sist.	Tiempo [s]	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Hex.	0,03168	02	00	20	2D	FF	C2	32	EC
Bin.	0,03168	0001	0000	0010 0000	00101101	1111 1111	1100 0010	0011 0010	1110 1100

Tabla 12. Bytes de ID 0x1620.

La información que brinda este ID es la siguiente:

Byte 07: Enchufe de carga.

Hexadecimal: 02
Binario: 0001 0000 (No Conectado)

Hexadecimal: 20
Binario: 0010 0000 (Cargando EV) 0= no, 1= si

Byte 04: Estado de carga batería

Tiempo: 0,03168s

Hexadecimal: 2D

Decimal: 45

El valor en decimal es el porcentaje de estado de carga 45%

Tiempo: 46,03062s

Hexadecimal: 2D

Decimal: 45%

Tiempo: 70,530027s

Hexadecimal: 2C

Decimal: 44%

Tiempo: 813,51319s

Hexadecimal: 21

Decimal: 33%

Tiempo: 1401,999497s

Hexadecimal: 1D

Decimal: 29%

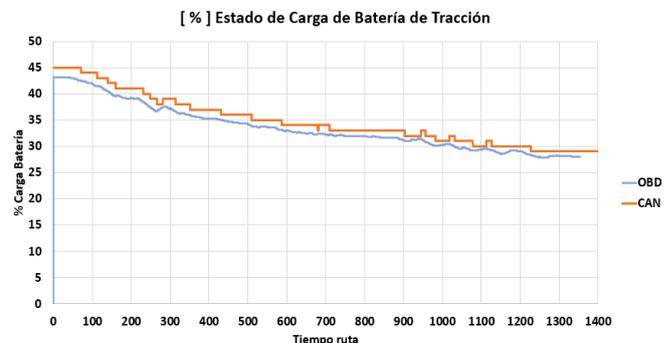


Fig. 11 Porcentaje estado de carga durante ruta.

Byte 03 y 02: Tiempo hasta completar la carga. Byte 03 todos sus bits, byte 02 sus dos primeros bits. No se puede tomar el tiempo hasta completar la carga porque está en movimiento, pero este es un ejemplo real tomado al momento de la carga con 96% de carga de batería.

Ejemplo:

Byte 03= 0C Byte 02= 84 Binario= 1100 1000 01000

Decimal= 50min hasta completar la carga

Byte 02 y 01: Rango de distancia por recorrer con estado de carga

Tiempo: 0,03168s

Hexadecimal: C2 32

Binario: 1100 0010 0011 0010

45%

Decimal: 35km

El valor en decimal es el rango de distancia por recorrer en km con el estado de carga 45%.

Tiempo: 46,03062s

Hexadecimal: C2 22

Binario: 1100 0010 0010 0010

44%

Decimal: 34km

Tiempo: 813,51319s

Hexadecimal: C1 92

Binario: 1100 0001 1001 0010

33%

Decimal: 31km

Tiempo: 1401,999497s

Hexadecimal: C1 72

Binario: 1100 0001 0111 0010

29%

Decimal: 23km

ESTADO DE SALUD DE LA BATERÍA DE TRACCIÓN (SOH)

Reconocemos el identificador del estado de salud en el byte 01, con valor constante en hexadecimal igual a 4D. En el ID 0x1624 al tener un valor constante simplemente separamos los bytes como en la Tabla 13 y transformamos todos los datos a decimal.

Data ID: 0x1624							
Sist.	Tiempo [s]	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Hex.	1261,116299	15	01	53	65	4D	40
Bin.	1261,116299	0001 0101	0001	0101 0011	0110 0101	0100 1101	0100 0000

Tabla 13. Bytes de ID 0x1624.

Teniendo una gráfica como la Fig. 12 con un valor constante de su estado de salud que realizando una conversión de hexadecimal a decimal nos dice, que se encuentra con un porcentaje del 77% desde el inicio de las

pruebas para recopilar datos del VE con la tarjeta de adquisición NI-9862.

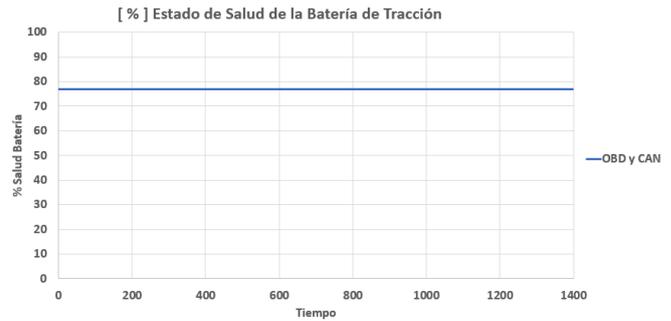


Fig. 12 Porcentaje estado de salud EV.

IV. DISCUSIÓN

En base a los resultados obtenidos podemos dar certeza de la dificultad de encontrar cada ID del vehículo eléctrico, y no solo porque los fabricantes de vehículos no ponen a disposición pública esta información, sino que también el formato estándar CAN no define que identificador de mensaje se asocia con que módulo o componente. En [3] usan una metodología para la obtención de identificadores de mensajes, recaudan IDs de diferentes vehículos y los comparan cada uno de ellos para controlar por CAN Bus un clúster de instrumentos, usan además el programa de lectura de datos NI-XNET Bus Monitor para leer y enviar por mensajes de red al clúster hasta activar algún testigo para encontrar los identificadores. En [4] monitorean la red de comunicación para el análisis de fallos en un vehículo eléctrico mediante identificadores con el uso de los módulos de velocidad del vehículo, velocidad del motor eléctrico y voltaje de la batería de tracción, mediante el mismo programa que en [3] de lectura de datos y una tarjeta de adquisición de datos NI-9862. En [8] al igual que en [3] y [4] usan identificadores para el funcionamiento de un cluster de instrumentos, en este caso usan la ingeniería inversa mediante la adquisición de datos comparándolos en gráficos contra el tiempo en archivos CSV para reconocer patrones identificables para enviar al cluster para su funcionamiento por CAN Bus.

Este trabajo a diferencia de los trabajos citados anteriormente muestra con detalle como los identificadores de mensajes intercambian información a diferentes nodos del vehículo eléctrico, obteniendo la trama de datos gracias a la tarjeta de adquisición NI 9862 y monitoreado por el programa NI-XNET Bus Monitor, que posteriormente serán evaluados en archivos CSV que ayudarán a reconocer patrones reconocibles en gráficos contra el tiempo. Entonces se muestra en el archivo CSV la trama de datos con sus IDs y dependiendo su longitud nos muestra el campo de datos compuesto por dos dígitos hexadecimales que podría tomar de 00 hasta FF valores, es decir 0 a 255 en valores decimales y valores binarios hasta 256 valores distintos.



Fig. 13 Tarjeta de adquisición de datos NI 9862 conectado al EV monitoreado por programa XNET Bus Monitor.

Es primordial interpretar de manera correcta la trama de datos, ya que cada byte tendrá un significado que tendrán que ser interpretados según la metodología para posteriormente convertirlos a valores entendibles por medio de fórmulas de conversión. Hay casos en este trabajo en los cuales un identificador pueda combinar en una misma fórmula dos bytes uno más significativo y uno menos significativo, así como añadir una compensación para ajustar las curvas en las gráficas, así también como dividir toda la fórmula para obtener valores menores. También puedes existir el uso de solo bits o combinación de bits como del ID 0x1620, y bytes en otros IDs que solo necesitan la conversión de un byte a decimal para descifrar la información. Esto dependerá de cada fabricante de automóvil.

Cabe destacar que durante la evaluación de los diferentes Identificadores del vehículo eléctrico se encontraron varios bytes que al ser valores constantes dificultan su significado, entre los que más se repiten son los valores de dos dígitos decimales 00 y FF. Estos se repitían en la mayoría de bytes de cada ID de la trama de datos del EV Renault Kangoo ZE dando a entender que según la teoría existen bytes que funcionan como “relleno de bits”, es decir más de 5 bits del mismo valor que se utilizan esencialmente para señalar errores, un indicador de error transmite a otros nodos para que también detecten el error dejando de transmitir e intentando el envío después [9]. También se dice en [9] que el bit de relleno se utiliza como mecanismo de resincronización para mantener el sincronismo en el todo el envío del mensaje, podemos tener un claro ejemplo en la Fig. 14.

Timestamp	ID	DATOS	Payload length	Type	BITS 07	BITS 06	BITS 05	BITS 04	BITS 03	BITS 02	BITS 01	BITS 00
0.00296	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.01077	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.02084	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.03077	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.04076	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.05643	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.06627	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.07945	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.08825	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.09664	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.10902	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.11643	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.12033	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.13039	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.14644	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.15681	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.16028	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.17641	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.18047	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.19072	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000
0.20056	394	FF 00 1F FE 3C 00 00	8	5	11111111	11110000	00000000	00011111	11111110	00111100	00000000	00000000

Fig. 14 Bits de relleno de ID 0x394.

De esta forma con el uso de la teoría investigada, la metodología y las formas de conversión de datos se encontraron seis identificadores: Posición del Pedal del Acelerador, Velocidad del Vehículo, Velocidad del Motor Eléctrico, Corriente de la Batería de Tracción, Estado de Carga de la Batería de Tracción y Estado de Salud de la Batería de Tracción.

Los datos usados fueron tomados en octubre de 2022, al ser un vehículo eléctrico debe sufrir cambios sobre todo en sus

baterías, entonces hemos tomado datos actuales en mayo 2023 con dos tipos de conducción normal y conducción tipo ECO, para poder comparar datos.

Podemos dar certeza que el pedal del acelerador del vehículo eléctrico tiene un máximo de 1,25 es decir 0% a 125% de posición del pedal como se observa en la Fig. 15 un adicional que puede exigir al EV a dar su máximo voltaje, máximo consumo de corriente y aumento de temperatura en la batería de tracción, esto podría afectar al estado de carga con una disminución de la autonomía. Además, para consultar el rendimiento del motor y detener la función ECO, se debe pisar a fondo el pedal del acelerador [10]. En el tipo de conducción ECO y conducción normal, no existe una restricción de cuanto se puede presionar el pedal del acelerador.

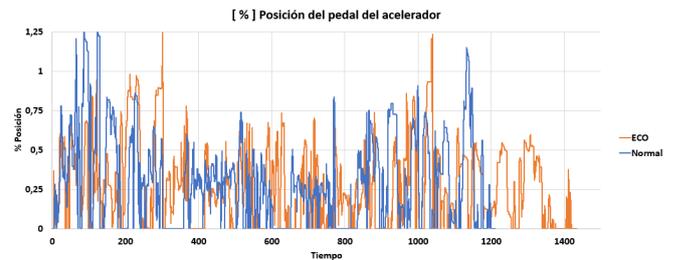


Fig. 15 Posición de pedal acelerador en modo de conducción ECO y normal.

Cuando se activa el modo ECO el vehículo controla la potencia del motor automáticamente reduciendo al máximo el consumo. Según la posición del pedal del acelerador la batería de tracción suministrará al motor energía eléctrica que lo usará el motor para mover el EV [10]. Como podemos observar en la Fig. 15, 16 y 17 diferenciamos las exigencias de potencia al motor eléctrico, en una conducción normal el

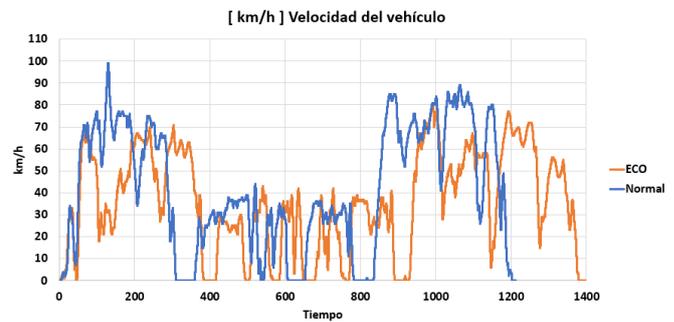


Fig. 16 Velocidad EV en modo de conducción ECO y normal.

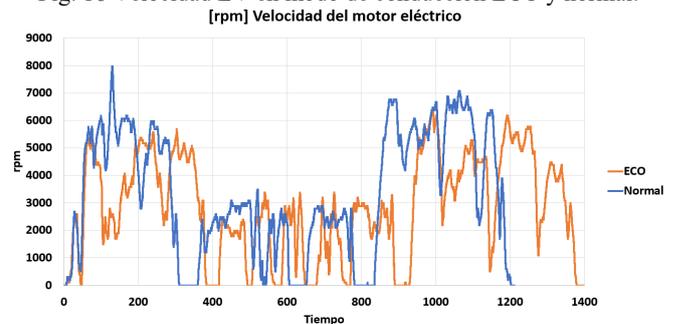


Fig. 17 Rpm de EV en modo de conducción ECO y normal.

EV aumenta la velocidad del vehículo y aumenta la velocidad del motor eléctrico sin restricción alguna, que a diferencia del modo ECO no sobrepasa los 70km/h y las 600rpm como se

observa en la Fig. 16 y 17, además que el consumo de energía se refleja en el Económetro del vehículo. Su funcionamiento dependerá de la energía eléctrica suministrada al motor, y además la corriente eléctrica usada de la batería de tracción como se observa en la Fig. 18 muestra el consumo (-A) y la recuperación de energía (+A). Existirá menor entrega de corriente eléctrica en el modo ECO siendo su máximo de 70 A, que a diferencia de una entrega de alrededor 135 en conducción normal.

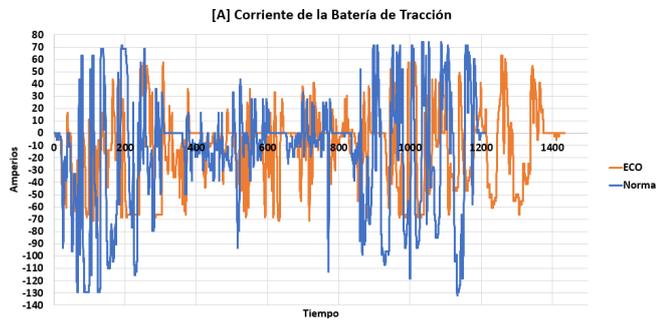


Fig. 18 Consumo de corriente de batería en modo de conducción ECO y normal.

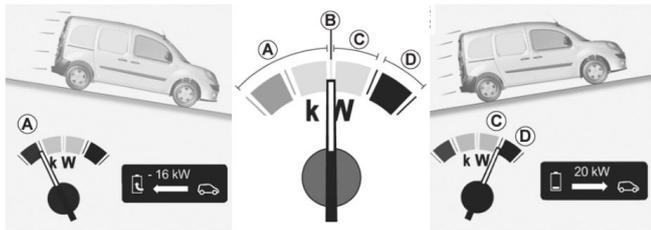


Fig. 19 Funcionamiento Económetro de EV [10].

El motor del vehículo eléctrico tendrá un freno motor importante, según la Fig. 19 en la posición B existe un consumo nulo, en posición A indica que el EV genera energía y recarga la batería, la posición C nos muestra un indicativo de consumo moderado, que a diferencia de la posición D refleja un consumo no aconsejado por el fabricante por seguridad de la batería. Al momento de levantar el pie del acelerador, el motor durante desaceleración genera corriente eléctrica para recargar la batería de tracción [10].

SOC (Estado de carga)

Se denomina estado de carga al porcentaje de capacidad sobrante de la batería de tracción en comparación con su capacidad total. El estado de carga informa cuantos Ah (Amperio hora) se puede extraer de cada celda hasta completar su descarga, parecido al indicador de combustible en los vehículos a combustión. A diferencia del sensor de medición de gasolina, el EV estima su SOC mediante algoritmo por la medición de varios sensores, garantizando la seguridad de cada celda de la batería para no sobrepasar voltaje ni temperatura y cause un accidente [11].

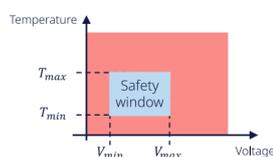


Fig. 20 Ventana de seguridad de las celdas [11].

Los vehículos eléctricos poseen una BMS encargada de monitorear las condiciones de funcionamiento de las celdas de la batería de tracción y mantenerlas en los límites de la ventana de seguridad como se observa en la Fig. 20, de esta forma extendiendo su vida útil. La BMS posee varios sensores de voltaje, temperatura y corriente como en la Fig. 21 que son los encargados de recopilar información de manera periódica para estimar el SOC y SOH, autorizando o no la carga y descarga, potencia máxima y mínima de entrada o salida, y también el balanceo de carga de las celdas [11].

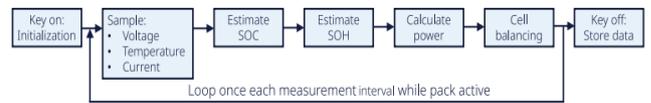


Fig. 21 Algoritmo de funcionamiento de una BMS [11].

En la Fig. 22 se muestra el estado de carga de la batería de tracción en dos modos de conducción, nos muestra en unidades de kilómetros el rango de distancia por recorrer con el estado de carga y además muestra el porcentaje de estado de carga. Se puede notar que, aunque el tiempo de recorrido es mayor en el modo ECO, por las limitaciones que tiene este modo su consumo es menor, con un estado de carga inicial de 89% y una autonomía inicial de 61km durante 23min se ha consumido el 15% de carga de batería y se ha reducido a 53km su autonomía. A diferencia del modo normal inicia con un estado de carga de 64% y una autonomía de 45km, que durante 20 min consume alrededor de 17% de carga de batería y reduce a 35km su autonomía. Puede consumir un poco más que el modo ECO, pero los picos de recuperación de carga son varios por el frenado regenerativo, ayudando mucho a que su descarga no sea tan progresiva, es recomendable por el fabricante siempre realizar una conducción relajada con una conducción constante y evitar acelerar más de lo requerido en subidas.

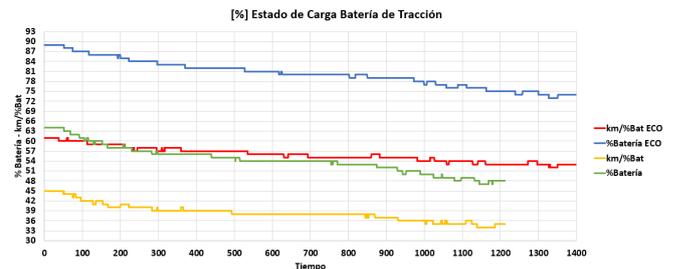


Fig. 22 SOC en dos modos de conducción y porcentaje de km por recorrer con estado de carga.

Podemos observar en la Fig. 23 en el cluster de instrumentos ya se recorrido durante varios minutos, de esta forma podemos dar certeza de la carga de batería y autonomía de la Fig. 22 en el modo de conducción ECO, que se puede observar que está activado en la parte derecha del cluster.



Fig. 23 Cluster activado modo de conducción ECO.

Además, el ID 0x1620 consta con el byte 07 que es el encargado de identificar el enchufe de carga se encuentra conectado, así como se describió en el apartado de Identificadores cuando el byte 07 sus dos dígitos hexadecimales se encuentran en 20 es decir 0010 bits, el EV se encuentra cargando. Podemos observar en la Fig. 24 en la trama de datos del identificador que además de del byte 07, en una combinación de bits del byte 03 y 02 muestra el tiempo hasta completar la carga.

Timestamp	ID	DATOS	Payload Length	Type	BYTE 02	BYTE 03	BYTE 07	BITS 02	BITS 03	BITS 07
44.098047	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
44.597963	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
45.09792	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
45.597912	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
46.097967	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
46.597963	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
47.097933	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
47.597926	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
48.097949	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
48.597853	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
49.097847	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
49.597919	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
50.097856	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
50.597869	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
51.097839	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
51.597838	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
52.097818	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
52.597829	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
53.097787	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
53.597832	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
54.097821	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
54.597806	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	
55.097778	1620	20 00 AD 5A 0F 03 DF FC	8 S	20	0F	03	00100000	00001111	00000011	

Fig. 24 Trama de datos de ID 0x1620 con enchufe de carga conectado.

Podemos afirmar entonces que la combinación de bits del ID 0x1620 brinda la información del tiempo hasta completar la carga de la batería de tracción, en el cluster de instrumentos de la Fig. 25 se observa el tiempo faltante junto al archivo CSV que combina los bits.



Fig. 25 Tiempo hasta completar la carga del EV con enchufe de carga conectado.

SOH (Estado de salud)

Las baterías de un EV tienen procesos de envejecimiento producto del uso y tiempo, puede ocurrir la degradación por varios motivos, pero dependerá en parte de los materiales de fabricación de la batería. Puede ocurrir el envejecimiento de la pérdida de capacidad definitiva durante el almacenamiento por altas temperaturas provocando corrosión y pérdida de litio. Además, un SOC elevado durante el almacenamiento produce una mayor degradación, se produce cuando ocurre la carga y descarga, dependerá también del modo de uso del vehículo. Para evaluar el envejecimiento de las celdas de la batería, el uso común es SOH (Estado de Salud) que es estimado por la BMS periódicamente. El envejecimiento de las celdas produce un desequilibrio de la cantidad de carga de cada celda, la BMS interpreta este problema equilibrando cada celda descargando la celda con mayor SOC, por tal motivo puede producir la sensación de que el vehículo eléctrico se descarga más rápido [11].

Data ID: 0x1624							
Sist.	Tiempo [s]	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Hex.	1261,116299	15	01	53	65	49	40
Bin.	1261,116299	0001 0101	0001	0101 0011	0110 0101	0100 1001	0100 0000

Tabla. 14 Bytes del SOH actual.

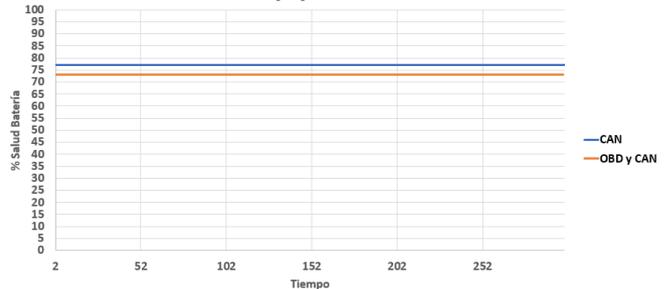


Fig. 26 SOH anterior y actual.

Esto se ve reflejado en el vehículo eléctrico Renault Kangoo ZE del año 2015, desde que se realizó la toma de datos en octubre de 2022 hasta la fecha que se tomaron nuevos datos mayo de 2023 se ha notado un cambio del estado de salud del 77% a 73% de SOH como se observa en la Fig. 26, notando el constante uso del EV y las exigencias al vehículo. Podemos reconocer los valores obtenidos ya que fueron tomados y comparados por un OBD 2 marca KlavKarr 210 y una tarjeta de adquisición de datos CAN NI 9862.

V. CONCLUSIONES

Este artículo presenta una información dinámica del uso de identificadores para su comunicación en el CAN Bus de un vehículo eléctrico Renault Kangoo Z.E año 2015. Con el uso de la tarjeta de adquisición de datos NI 9862, el monitoreo de la trama de datos en el programa NI XNET Bus monitor y la evaluación de los datos en archivos CSV, se demostró el uso de algunos IDs y el uso de sus bytes en el campo de datos el intercambio de información en sus respectivos nodos. Además, el uso de combinación de bytes, bits y fórmulas de conversión para demostrar mediante una comparación con un OBD 2 que la información plasmada es la correcta. Además, la interpretación de cada ID encontrado para dar una evaluación del vehículo eléctrico, para demostrar sus tolerancias y exigencias en dos modos de conducción ECO y normal, encontrando al final un estado de salud desde la fecha de inicio 77% hasta el momento 73% dando a entender que los parámetros de uso del EV no son los adecuados por eso su degradación constante.

VI. REFERENCIAS

- [1] Hernandez Soria JE. Vulnerability assessment of Electronic Control Unit (ECU) of automotive system through OBD-II port and CAN bus. 2017.
- [2] Pacheco Hernández JEP. Monitoreo de hábitos de manejo por medio de una red CAN automotriz [Internet]. 2011 [citado el 16 de junio de 2023]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/pacheco_h_je/
- [3] Feliciano Fuentes LG. Identificación y control de parámetros de clúster de instrumentos automotriz mediante Red Can. noviembre de 2019 [citado el 16 de junio de 2023]; Disponible en: <https://repositorioinstitucional.buap.mx/handle/20.500.12371/4738>
- [4] Rojas Hiedra DF. Implementación de un modelo dinámico para una red de comunicación presente en un vehículo eléctrico Kia Soul mediante NI-XNET [Internet] [masterThesis]. Universidad del Azuay; 2019

[citado el 16 de junio de 2023]. Disponible en: <http://dspace.uazuay.edu.ec/handle/datos/9564>

- [5] Robert Bosch GmbH, editor. Bosch Automotive Electrics and Automotive Electronics: Systems and Components, Networking and Hybrid Drive [Internet]. Wiesbaden: Springer Fachmedien; 2014 [citado el 16 de junio de 2023]. Disponible en: <http://link.springer.com/10.1007/978-3-658-01784-2>
- [6] Nürnberger S, Rossow C. – vatiCAN – Vetted, Authenticated CAN Bus. En: Gierlichs B, Poschmann AY, editores. Cryptographic Hardware and Embedded Systems – CHES 2016. Berlin, Heidelberg: Springer; 2016. p. 106–24. (Lecture Notes in Computer Science).
- [7] National Instruments. Descripción general de la red de controladores de área (CAN) [Internet]. [citado el 16 de junio de 2023]. Disponible en: <https://n9.cl/va1fy>
- [8] Staggs J. How to Hack Your Mini Cooper: Reverse Engineering CAN Messages on Passenger Automobiles. En 2013 [citado el 16 de junio de 2023]. Disponible en: <https://n9.cl/ndujg>
- [9] Davis R. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. Real-Time Systems. el 6 de febrero de 2007; 35:239–72.
- [10] Renault S.A.S. Kangoo Z.E Manual de Utilización. Francia: 2015
- [11] Rincon Vija F. Enabling robust wireless communication for battery management systems in electric vehicles [Internet] [phdthesis]. Ecole nationale supérieure Mines-Télécom Atlantique; 2022 [citado el 16 de junio de 2023]. Disponible en: <https://theses.hal.science/tel-04009502>