

Jornadas de Automática

Vehículo híbrido FOX como plataforma experimental para control

Echedey Aguilar Hernández^{a,*}, Ascención Zafra Cabeza^a, Carlos Bordons Alba^a

^aDepartamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Camino de los Descubrimientos, s/n, 41092 Sevilla, España.

To cite this article: Aguilar Hernández, E., Zafra Cabeza, A., Bordons Alba, C. 2026. FOX Vehicle as an Experimental Platform for Control. Jornadas de Automática, 47. <https://doi.org/10.17979/ja-cea.2026.47.13837>

Resumen

Este trabajo presenta la actualización del vehículo eléctrico híbrido FOX como plataforma experimental para ensayos de control embebido, gestión energética, comunicaciones, supervisión HMI y validación de arquitecturas de hardware y software. La plataforma original estaba basada en una arquitectura distribuida PC/104, con una unidad de control sobre QNX y un supervisor independiente sobre Windows XP/LabVIEW. La actualización propuesta sustituye dicha configuración por una ECU industrial ATC8110 basada en Linux RT. Esta arquitectura integra la adquisición de señales del vehículo, la comunicación con el BMS y los controladores de motor, la selección y ejecución de algoritmos de control, el envío de consignas a los actuadores, la operación mediante una HMI desarrollada en Python/Qt y el registro de datos experimentales. Además, incorpora una lógica de arranque seguro, una estructura software modular y flujos de integración continua mediante GitHub Actions. La plataforma resultante proporciona una base flexible para continuar con futuras pruebas de control, diagnóstico, gestión energética y validación experimental en vehículo.

Palabras clave: vehículos híbridos y eléctricos, control embebido, sistemas de tiempo real, redes de comunicación en vehículo, interfaces hombre-máquina, supervisión, adquisición de datos

FOX Vehicle as an Experimental Platform for Control

Abstract

This work presents the upgrade of the FOX hybrid electric vehicle as an experimental platform for embedded control testing, energy management, communications, HMI supervision, and validation of hardware and software architectures. The original platform was based on a distributed PC/104 architecture, with a control unit running on QNX and an independent supervisor running on Windows XP/LabVIEW. The proposed upgrade replaces this configuration with an ATC8110 industrial ECU based on Linux RT. This architecture integrates vehicle signal acquisition, communication with the BMS and motor controllers, selection and execution of control algorithms, command transmission to the actuators, operation through an HMI developed in Python/Qt, and experimental data logging. In addition, it incorporates safe start-up logic, a modular software structure, and continuous integration workflows using GitHub Actions. The resulting platform provides a flexible basis for continuing future tests on control, diagnostics, energy management, and experimental validation in the vehicle.

Keywords: hybrid and electric vehicles, embedded computer control systems, real-time systems, in-vehicle communication networks, man-machine interfaces, supervision, data acquisition

*Autor para correspondencia: eaguilar1@us.es

1. Introducción

Los vehículos eléctricos experimentales constituyen plataformas de gran interés para el desarrollo y la validación de sistemas de control, supervisión y gestión energética. A diferencia de un banco de ensayos puramente simulado, un vehículo experimental permite trabajar con sensores reales, actuadores eléctricos, almacenamiento energético, comunicaciones distribuidas, restricciones físicas de seguridad e interfaces de operación. Por ello, este tipo de plataformas resulta especialmente útil para validar progresivamente arquitecturas hardware y software antes de abordar ensayos avanzados en condiciones reales.

El vehículo FOX es un prototipo eléctrico híbrido desarrollado en la Universidad de Sevilla como plataforma de investigación para el estudio de estrategias de control, gestión energética y distribución de par. La Figura 1 muestra una vista general del vehículo utilizado como plataforma experimental.



Figura 1: Vista general del vehículo FOX utilizado como plataforma experimental.

Su configuración de cuatro motores eléctricos independientes, uno por rueda, permite actuar de forma diferenciada sobre el par aplicado en cada rueda, lo que lo convierte en una plataforma adecuada para ensayos de control de tracción, estabilidad y reparto de par. Además, trabajos previos han contribuido al desarrollo del FOX mediante la supervisión y parametrización de estrategias de control (García, 2019), el análisis de estrategias de tracción y estabilidad (Ramos, 2019), y la propuesta de esquemas orientados a conducción autónoma (Gálvez, 2019).

La arquitectura original del vehículo estaba formada por una unidad de control electrónica, o ECU, basada en QNX, y un supervisor independiente ejecutado sobre Windows XP Embedded con una interfaz desarrollada en LabVIEW. Esta configuración permitió realizar pruebas funcionales sobre el vehículo, pero presenta limitaciones asociadas a la obsolescencia tecnológica, la dificultad de mantenimiento, la separación entre control y supervisión, y la limitada flexibilidad para incorporar nuevas herramientas de adquisición, registro y análisis de datos.

En este contexto, el presente trabajo describe la actualización hardware y software del vehículo FOX con el objetivo de convertirlo en una plataforma experimental más flexible, mantenible y escalable. La actualización se basa en la sustitución de la arquitectura distribuida original por una nueva ECU

industrial con Linux RT, capaz de centralizar la adquisición de señales, la comunicación con los subsistemas del vehículo, la selección y ejecución de algoritmos de control, el envío de consignas a los actuadores, la supervisión mediante una interfaz hombre-máquina, o HMI, y el registro de datos experimentales. El trabajo se centra en la definición y documentación de una arquitectura experimental que permita realizar futuros ensayos de control, diagnóstico, comunicaciones, supervisión y gestión energética.

Las principales contribuciones de este trabajo son: i) la descripción de la arquitectura original del vehículo FOX y sus limitaciones; ii) la actualización de la plataforma hardware mediante una ECU industrial; iii) la reorganización modular del software embarcado considerando los distintos algoritmos de control; iv) la integración de una nueva HMI desarrollada en Python/Qt; y v) la definición de un flujo de registro y análisis de datos orientado a futuros ensayos experimentales.

2. Sistema original embarcado

El vehículo FOX utiliza una arquitectura de tracción integral mediante cuatro motores eléctricos independientes, uno instalado en cada rueda. Estos motores son de tipo *brushless*, con una potencia nominal de 7 kW cada uno y un par máximo aproximado de 80 Nm. Esta disposición permite modificar de forma independiente el par aplicado en cada rueda, haciendo posible el estudio de estrategias de vectorización de par, control de tracción y control de estabilidad.

La fuente de energía principal del vehículo es una batería de litio-ferrofosfato, formada por 24 celdas agrupadas en seis paquetes de cuatro celdas. La batería alimenta los motores eléctricos a través de los convertidores correspondientes, mientras que el sistema de gestión de baterías, o BMS, supervisa variables como el estado de carga y la tensión de las celdas. La plataforma está concebida como sistema híbrido y puede integrar una pila de combustible de hidrógeno y otros elementos de almacenamiento energético, aunque numerosas pruebas documentadas se han realizado empleando exclusivamente el sistema de baterías. El vehículo dispone además de frenos de disco en las cuatro ruedas, accionados mecánicamente por el conductor e independientes de los motores eléctricos. El sistema de control original se dividía en dos unidades de computación embebidas basadas en el estándar PC/104, situadas tras el asiento del copiloto. La ECU constituía el elemento central de control del vehículo. Su función era adquirir las señales procedentes de los sensores, procesar la información disponible y ejecutar la estrategia de control seleccionada, generando las consignas correspondientes para los actuadores del vehículo.

Su hardware incluía una placa Diamond Systems PCM-3370, tarjetas de adquisición PCM-3718 HO para señales analógicas y digitales, tarjetas de salidas y E/S, fuente de alimentación y módulos de comunicación CAN (Cortés, 2017). La ECU ejecutaba QNX 6.3.2, un sistema operativo de tiempo real empleado para las tareas de adquisición, comunicación y control.

La Figura 2 muestra de forma esquemática la distribución de los principales elementos del sistema original instalado en el FOX, incluyendo la ECU, el BMS, la HMI, los sensores, los

motores y las principales líneas de comunicación y alimentación.

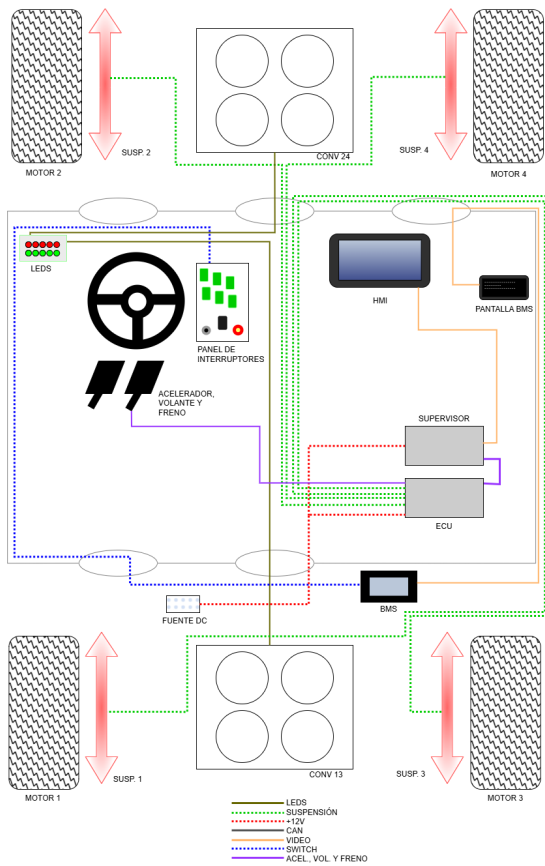


Figura 2: Esquema general del sistema original instalado en el vehículo FOX.

El Supervisor era un segundo ordenador PC/104 dedicado a la interacción con el usuario, la monitorización y la operación del vehículo. Este sistema ejecutaba Windows XP Embedded e incorporaba una interfaz desarrollada en LabVIEW. A través de una pantalla táctil, el copiloto podía visualizar variables de baterías, motores, velocidades, temperaturas y errores, así como modificar parámetros asociados a los controladores durante las pruebas. Esta separación permitía distinguir entre las tareas críticas de control, ejecutadas en la ECU, y las funciones de supervisión y ajuste, realizadas desde el Supervisor.

El vehículo FOX incorporaba además una unidad de medida inercial Microstrain 3DM-GX3-35 con GPS, empleada para obtener aceleraciones, velocidades angulares y posicionamiento. También disponía de potenciómetros para medir el desplazamiento de la suspensión en cada rueda, la posición de los pedales de freno y acelerador, y el ángulo de giro del volante. La comunicación principal entre la ECU, el Supervisor, el BMS y los controladores de motor se realizaba mediante bus CAN, mientras que la IMU se comunicaba con la ECU mediante RS-232.

Esta arquitectura permitió realizar ensayos funcionales sobre el vehículo y validar estrategias de control en una plataforma real. Sin embargo, presenta limitaciones importantes para su evolución como plataforma experimental. La dependencia de hardware PC/104, QNX 6.3.2 y Windows XP Embedded dificulta el mantenimiento y la sustitución de componentes.

Además, la separación física y lógica entre ECU y Supervisor limita la integración de nuevas herramientas de supervisión, registro y análisis de datos. Por último, la dependencia de una HMI desarrollada en LabVIEW reduce la flexibilidad para modificar, desplegar y versionar el software de forma trazable. Estas limitaciones motivan la actualización hardware y software descrita en las siguientes secciones.

3. Actualización hardware y software propuesta

La actualización de la plataforma sustituye la arquitectura distribuida original por una nueva ECU industrial capaz de centralizar la adquisición de datos, la actuación, las comunicaciones, la supervisión y el registro experimental. La plataforma computacional actual se basa en una ATC8110 con Linux RT, equipada con procesador Intel i7, memoria RAM, almacenamiento SSD e interfaces de comunicación y expansión adecuadas para su integración en el vehículo.

La Figura 3 resume la organización funcional de la nueva ECU. La adquisición de datos y las comunicaciones CAN actualizan un estado compartido del vehículo, que actúa como punto común para los módulos de gestión energética, control, supervisión HMI y registro de datos. Las consignas generadas por los algoritmos de control pasan por una capa de seguridad funcional antes de enviarse a la etapa de actuación.

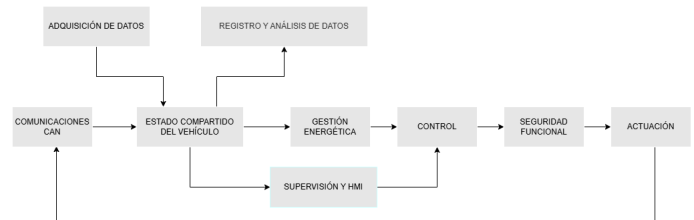


Figura 3: Organización funcional de la ECU ATC8110 con Linux RT.

La comunicación con los principales subsistemas del vehículo se realiza mediante bus CAN a través de módulos EMUC-B2S3 integrados con SocketCAN. Esta solución permite disponer de varios canales CAN 2.0B aislados y facilita el desarrollo de aplicaciones de comunicación sobre Linux. La adquisición de señales analógicas se lleva a cabo mediante una tarjeta PEX-1202L, empleada para la lectura de señales procedentes de pedales, volante, corrientes de los ejes y desplazamientos de suspensión. Por su parte, la tarjeta PEX-DA16 proporciona salidas analógicas para la ruta de actuación, cuya integración final requiere la validación del mapeo físico de señales y de las habilitaciones de seguridad.

La nueva arquitectura reduce la dependencia de hardware y software obsoleto, elimina la separación entre ECU y Supervisor, y proporciona una base más flexible para integrar algoritmos de control, adquisición de datos, supervisión HMI y herramientas de análisis experimental.

El desarrollo software se realiza inicialmente en un ordenador externo, donde se implementan y verifican los distintos módulos de adquisición, comunicación, control, supervisión y registro. Una vez obtenida una versión inicial, el código se compila y se valida mediante flujos de integración continua basados en GitHub Actions. Este proceso permite automatizar

comprobaciones básicas del software antes de su despliegue en la ECU del vehículo.

Tras esta etapa, la versión candidata se transfiere a la ECU ATC8110, donde se ejecutan pruebas de funcionamiento sobre la plataforma experimental. Los resultados de estas pruebas permiten verificar la integración con los subsistemas del vehículo y, en caso satisfactorio, fusionar los cambios con la versión estable del software de control. La HMI desarrollada en Python/Qt se describe con mayor detalle en la Sección 4.

4. Interfaz HMI en Python/Qt

Junto con la actualización de la ECU se desarrolla una nueva interfaz hombre-máquina, o HMI, para la supervisión y operación del vehículo. La interfaz se implementa como una aplicación de escritorio en Python utilizando Qt/PySide6. Su objetivo es sustituir la supervisión basada en LabVIEW por una herramienta modular, mantenible y adaptable a la evolución de la plataforma.

La HMI contempla varias vistas funcionales, entre ellas una vista principal, una vista BMS, una vista de control, una vista CAN, una vista de motores y una vista de errores. El panel principal reúne la información operativa más relevante mediante un velocímetro compacto, indicadores de salud del bus CAN, un resumen de errores, un bloque de tensiones de celdas del BMS y una representación gráfica del vehículo.

Esta representación sintética muestra una vista superior del vehículo y superpone información contextual sobre la dirección, los motores seleccionados y sus velocidades de giro. Este enfoque evita depender únicamente de tablas de datos y facilita la interpretación rápida de la información operativa durante pruebas de banco o ensayos embarcados.

La Figura 4 muestra el panel principal de la HMI en estado de reposo.

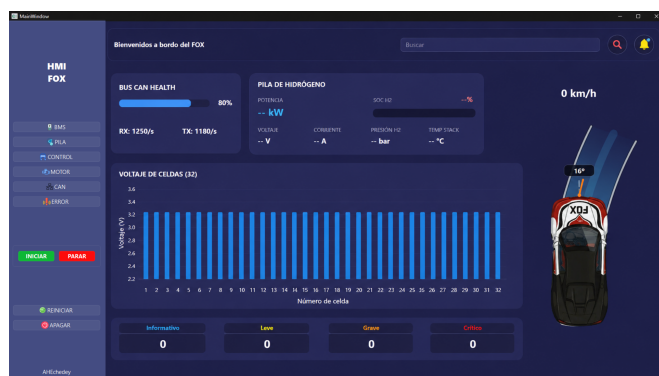


Figura 4: Panel principal de la HMI desarrollada en Python/Qt.

Además de las funciones de supervisión, la HMI permite seleccionar y parametrizar las estrategias de control ejecutadas en la ECU. Desde la vista de control, el usuario puede escoger la estrategia activa, habilitar o deshabilitar la actuación sobre los motores y modificar parámetros asociados a cada controlador antes o durante las pruebas. Este planteamiento mantiene la filosofía del supervisor original, que permitía al copiloto monitorizar el estado del vehículo, seleccionar la estrategia de

control, ajustar sus parámetros principales y decidir sobre qué motores se actuaba durante los ensayos (García, 2019).

Las estrategias consideradas están orientadas al control de tracción y estabilidad del vehículo. En términos generales, emplean variables como la velocidad angular de guiñada, o *yaw-rate*, el ángulo de deslizamiento, la velocidad del vehículo, el ángulo del volante y las velocidades de rueda para calcular acciones correctivas sobre el par aplicado a cada motor (Rajamani, 2012). De este modo, la arquitectura permite ensayar estrategias de reparto de par entre ruedas, control de estabilidad y reducción del deslizamiento longitudinal o lateral.

4.1. Estrategias de control implementadas

La vista de control incluye las estrategias Fuzzy, Proporcional, MPC, Ganancias, PI y Elipsoide. Estas estrategias no actúan todas sobre el mismo aspecto dinámico del vehículo: algunas están orientadas principalmente al control de estabilidad, corrigiendo desviaciones de *yaw-rate* o del ángulo de deslizamiento; otras se emplean como capas de control de tracción, limitando el par aplicado cuando existe riesgo de pérdida de adherencia; y otras combinan ambas funciones mediante reparto de par y limitación del par transmisible.

La HMI no ejecuta directamente estos algoritmos, sino que actúa como interfaz de selección, parametrización y supervisión. La selección realizada se transmite al módulo de control de la ECU, que ejecuta el algoritmo correspondiente y genera las consignas enviadas posteriormente a la etapa de actuación. A continuación se describe brevemente la función de cada estrategia dentro del sistema de control del FOX.

- **Controlador Proporcional** La estrategia *Proporcional* utiliza dos modos de operación para actuar sobre la estabilidad del vehículo. El primero es un modo de maniobrabilidad, asociado principalmente al error de *yaw-rate*, y el segundo es un modo de estabilidad lateral, asociado al ángulo de deslizamiento. Su finalidad es modificar el reparto de par entre las ruedas para corregir situaciones en las que la velocidad angular real del vehículo no coincide con la referencia deseada o cuando aparece un deslizamiento lateral significativo.
- **Controlador mediante Ganancias** La estrategia *Ganancias* calcula directamente los pares que deben aplicarse a cada motor, sin obtener previamente un momento correctivo total sobre el vehículo. Para ello, parte del par demandado por el conductor y lo modifica mediante dos factores: una ganancia de tracción y una ganancia de estabilidad. La ganancia de tracción limita el par cuando la rueda puede perder adherencia, mientras que la ganancia de estabilidad corrige situaciones de subviraje y sobreviraje a partir del error de *yaw-rate*. De este modo, el controlador adapta de forma independiente la actuación de cada rueda según el estado dinámico del vehículo (García, 2019).
- **Controlador PI** La estrategia *PI* proporciona una estructura de control más directa para la corrección de errores dinámicos. Se implementa como un controlador de estabilidad basado en el error entre la velocidad angular real del vehículo

la velocidad angular de referencia. A partir de dicho error de *yaw-rate*, el controlador calcula una variable de actuación que permite distribuir el par total demandado por el conductor entre las cuatro ruedas.

■ Controlador *Fuzzy*

La estrategia *Fuzzy* calcula acciones correctivas a partir de reglas lingüísticas definidas sobre variables dinámicas del vehículo. Este tipo de control permite trabajar con razonamientos similares a los que emplearía un conductor experto, transformando variables de entrada, como errores dinámicos o condiciones de estabilidad, en una acción de control correctiva.

En el caso del vehículo FOX, el controlador basado en lógica difusa está orientado al control de estabilidad. Su función es calcular una actuación que permita corregir la trayectoria del vehículo cuando aparece una desviación entre el comportamiento real y el comportamiento deseado. Para ello, se apoya en la idea de par diferencial: modificar el par aplicado en las ruedas de un lado y del otro del vehículo permite generar un efecto correctivo sobre el giro del vehículo.

■ Controlador basado en Elipsoides

La estrategia *Elipsoide* se emplea como capa de control de tracción. Su finalidad es limitar el par aplicado a cada rueda para que la fuerza transmitida por el neumático no supere la adherencia disponible. Para ello, se considera la relación entre las componentes longitudinal y transversal de la fuerza del neumático, cuyo límite puede representarse mediante una región elíptica denominada elipsoide de tracción. A partir de esta representación, el controlador estima el par máximo transmisible por cada rueda, imponiendo una restricción sobre la componente longitudinal de la fuerza. Dado que este cálculo depende de aproximaciones y estimaciones del estado dinámico del vehículo, la estrategia se complementa con un control proporcional del deslizamiento. De este modo, el par finalmente aplicable queda condicionado tanto por el límite de fuerza longitudinal transmisible como por el error de deslizamiento de la rueda (Yin and Hori, 2009).

■ Controlador MPC

La estrategia *MPC*, o control predictivo basado en modelo, emplea un modelo del vehículo para calcular la acción de control considerando la evolución prevista del sistema. A diferencia de estrategias más directas, el MPC no actúa únicamente en función del error instantáneo, sino que utiliza un horizonte de predicción para anticipar el comportamiento dinámico del vehículo y determinar la acción correctiva más adecuada.

En el esquema de control utilizado para el vehículo FOX (ver Figura 5), el controlador MPC calcula el momento correctivo total necesario a partir de las referencias de *yaw-rate* y *sideslip*. Posteriormente, una capa de vectorización de par, o TVC, reparte ese momento correctivo entre las ruedas. Finalmente, la capa MTTE, *Maximum Transmissible Torque Estimation*, actúa como control de tracción limitando el par aplicado para

evitar la pérdida de agarre de las ruedas (Camacho and Bordons, 2026).

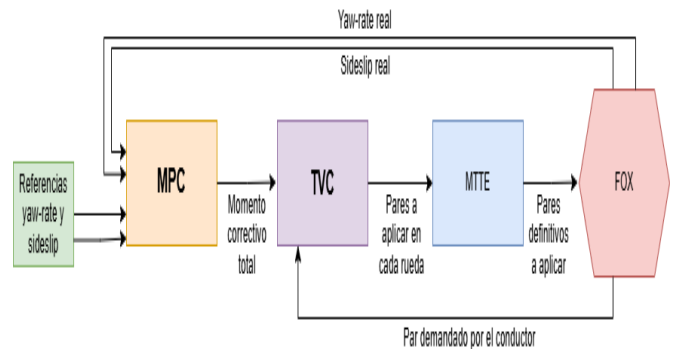


Figura 5: Esquema funcional del controlador basado en MPC.

5. Registro, análisis y posibilidades de ensayo

La arquitectura actualizada incorpora el registro de información experimental como parte del flujo de ensayo. Durante las pruebas, la ECU puede almacenar variables procedentes de sensores, mensajes CAN, consignas de control, estados de batería, alarmas y eventos de operación. Esta información permite reconstruir los ensayos, representar series temporales y analizar el comportamiento del sistema con mayor detalle que durante la operación en tiempo real.

El registro de datos facilita la comparación entre distintas configuraciones de control y la depuración del software embarcado. En particular, los datos registrados pueden integrarse con herramientas de visualización como Grafana para analizar la evolución temporal de las variables relevantes y comparar resultados entre pruebas. De esta forma, se puede observar en la Figura 6 las variables registradas sobre los 4 motores en un ensayo real usando Grafana.

La plataforma actualizada habilita varias líneas de ensayo futuras. En primer lugar, permite validar estrategias de control longitudinal y de tracción sobre una arquitectura de cuatro motores eléctricos independientes. En segundo lugar, proporciona una base para integrar progresivamente funciones de control de estabilidad y reparto de par, utilizando variables dinámicas ya disponibles en la plataforma, como las procedentes de la IMU/GPS. Finalmente, la integración progresiva de la pila de hidrógeno permitirá estudiar estrategias de gestión energética híbrida, coordinando batería, BMS, demanda de tracción y restricciones operativas de la pila (Quintero, 2015).

6. Conclusiones

Este trabajo ha presentado la actualización del vehículo FOX como plataforma experimental para ensayos de control embebido, supervisión, comunicaciones y gestión energética. Para ello, se ha descrito la arquitectura original basada en una ECU PC/104 con QNX y un Supervisor PC/104 con Windows XP/LabVIEW, identificando sus principales limitaciones en términos de mantenimiento, ampliación, integración de nuevas herramientas y análisis de datos.



Figura 6: Visualización en Grafana de variables registradas durante un ensayo sobre los cuatro motores del FOX.

La actualización propuesta sustituye dicha arquitectura distribuida por una ECU industrial ATC8110 con Linux RT. Esta configuración permite integrar la adquisición de señales, la comunicación con los subsistemas del vehículo, la ejecución de algoritmos de control, la supervisión HMI y el registro de datos experimentales en una plataforma más modular y mantenible.

Asimismo, la nueva HMI desarrollada en Python/Qt y el flujo de registro de datos proporcionan una base flexible para la operación, depuración y análisis posterior de ensayos. La plataforma resultante refuerza el papel del FOX como banco experimental para validar estrategias de control, diagnóstico, comunicaciones y gestión energética en vehículo.

Como líneas futuras se plantea completar la validación de las interfaces de comunicación y actuación, consolidar de forma progresiva el subsistema de hidrógeno y realizar ensayos experimentales en banco y sobre el vehículo.

Agradecimientos

Este trabajo forma parte del proyecto de I+D+i PID2022-142069OB-I00 (REGREENH2), financiado por MICIU/AEI/10.13039/501100011033 y por los fondos FEDER, Unión Europea, y forma también parte del proyecto AGERAR PLUS, financiado por FEDER-POCTEP.

Referencias

- Camacho, E. F., Bordons, C., 2026. Model Predictive Control. Third Edition. Springer.
- Cortés, P. M., 2017. Estudio preliminar del diseño de una nueva unidad de control del motor para el vehículo fox. Trabajo fin de grado, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, España, tutor: Francisco Gavilán Jiménez.
- García, J. C. F., 2019. Control de la dinámica de vehículos eléctricos con motores en las ruedas. Trabajo fin de grado, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, España, tutores: Carlos Bordons Alba y Juan José Márquez Quintero.
- Gálvez, M. M., 2019. Esquema de control para conducción autónoma: aplicación al vehículo fox. Trabajo fin de máster, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, España, tutor: Carlos Bordons Alba.
- Quintero, J. J. M., 2015. Desarrollo de sistema de control y supervisión de una pemfc para su integración en sistemas híbridos de energía. Trabajo fin de máster, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, España, tutor: Carlos Bordons Alba.
- Rajamani, R., 2012. Vehicle Dynamics and Control, 2nd Edition. Springer.
- Ramos, I. R., 2019. Control de tracción y estabilidad para un vehículo eléctrico con motores en las ruedas. Trabajo fin de grado, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Sevilla, España, tutores: Miguel Ángel Rídao Carlini y Juan José Márquez Quintero.
- Yin, D., Hori, Y., 2009. A novel traction control without chassis velocity for electric vehicles. World Electric Vehicle Journal 3 (2), 282–288.