



Trabajo de título USM - INSA Lyon
Simulación y control de un vehículo eléctrico
Prof. Jaime Espinoza
Resumen ejecutivo

Sebastián Zagal Rojas
ROL USM: 201941004-0

Universidad Técnica Federico Santa María
Departamento de Ingeniería Mecánica

24 de octubre de 2024

Índice

1. Introducción	3
2. Desarrollo	3
2.1. Prototipo y esquema geométrico del vehículo	3
2.2. Puesta en marcha del simulador 3D de control	6
2.3. Modelo de control final	7
3. Conclusiones y pistas de mejora	8

Índice de Figuras

1. Prototipo en el estado en que comenzó esta iteración del proyecto.	3
2. Prototipo final con los cambios incorporados.	4
3. Esquema representando el principio de Ackermann.	5
4. Esquema de entradas y salidas desde el computador embebido.	6
5. Simulación 3D en tiempo real del prototipo con control de velocidad y viraje.	7
6. Modelo Simulink de control del vehículo.	7

1. Introducción

Como parte de mis estudios en el programa de doble diploma entre la Universidad Técnica Federico Santa María y el Institut National des Sciences Appliquées de Lyon (INSA Lyon), realicé un trabajo de investigación en ingeniería de una duración de aproximadamente 3 meses para optar al diploma de ingeniero francés. El proyecto que aquí se presenta es la segunda parte de una iniciativa cuyo objetivo último es obtener un vehículo eléctrico autónomo. Si bien dicha meta es bastante ambiciosa, el proyecto ha sido dividido en iteraciones, de las cuales la primera fue en 2019. En ella solo se alcanzó a diseñar el primer prototipo consistente de una placa de metal, un computador embebido y cuatro ruedas con sus correspondientes motores, baterías y sensores (ver Figura 1). El presente trabajo se enmarca en la segunda iteración del proyecto, en la que resulta necesario enfocarse en definir y hacer viable un control del prototipo existente que pueda realizarse de manera remota.

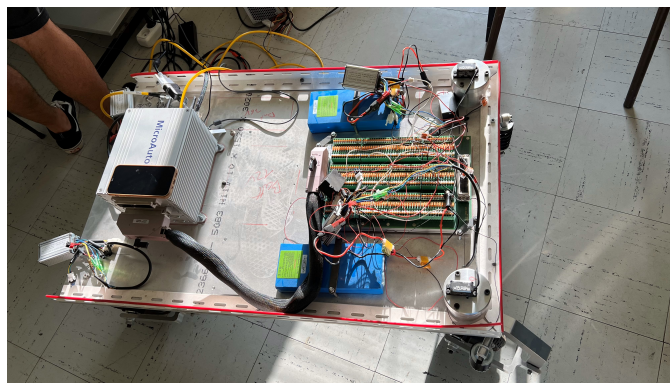


Figura 1: Prototipo en el estado en que comenzó esta iteración del proyecto.

De esta forma, nuestro objetivo es, a partir del prototipo físico previamente desarrollado, introducir mejoras en la electrónica y crear una estructura de control en bucle abierto para controlar el sistema de manera remota y, así, facilitar la siguiente iteración del proyecto. Para esto, se trabaja en la programación del computador embebido, una MicroAutoBox II de la compañía dSPACE, a través del software Simulink de MATLAB. Asimismo, se incorporan placas de desarrollo en vez de los antiguos controladores de torque con los que contaba el vehículo, para poder efectuar el procesamiento de datos y enviar las señales a cada motor; y además, se establece una simulación 3D en tiempo real del vehículo en conjunto con la puesta en marcha del control en bucle abierto en el prototipo modificado.

2. Desarrollo

2.1. Prototipo y esquema geométrico del vehículo

Para caracterizar el vehículo y saber de qué forma establecer el control remoto, fue necesario caracterizar sus parámetros más importantes y, basándonos en ellos, hacer los cálculos de la geometría para planificar el control. Antes de proceder a esto, se hicieron algunas modificaciones en la distribución del prototipo de la primera iteración, como por ejemplo el cambio de placas de desarrollo por los controladores de torque previamente instalados. Una vez hecho lo anterior (ver Figura 2), los componentes principales del vehículo quedaron de la siguiente forma:

- Una placa de aluminio que actúa como cuerpo/chasis del vehículo cuyas dimensiones son $972 \times 626 \times 50$ [mm³].
- Cuatro ruedas motrices de 200 [mm] (OZO ELECTRIC, n.d.) de diámetro, como aquellas de los scooters eléctricos, con una potencia de 350 [W] y alimentadas con Corriente Continua (CC) en tensiones de 24, 36 o 48 [V]. De ellas, solo se controlan las dos traseras, de modo que las delanteras quedan como ruedas libres.
- Dos servomotores Savox SB-2290SG con un torque de 45 [kg·cm] para generar la dirección del eje frontal, es decir, el giro de las ruedas delanteras.
- Dos baterías de 36 [V] para alimentar el sistema ¹.
- Una MicroAutoBox II, de la compañía dSPACE, que es el computador embebido del sistema y se encarga realizar los cálculos y de administrar las señales que irán a cada motor.
- Cuatro placas de desarrollo dsPICDEM MCLV-2 de Microchip para controlar las ruedas motrices en base a la información de la MicroAutoBox II, así como para comunicar directamente con el control remoto (ver Figura 4).

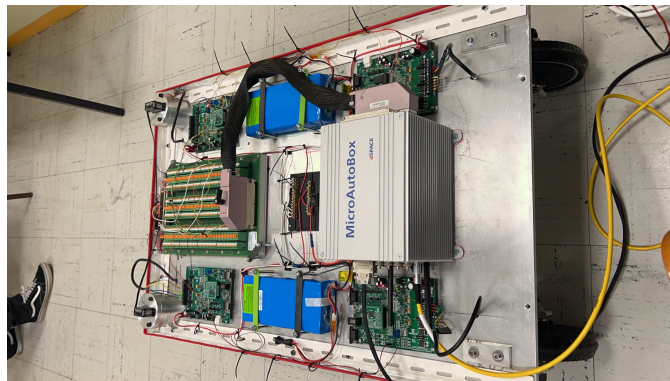


Figura 2: Prototipo final con los cambios incorporados.

De la antigua iteración del proyecto, se contaba con los datos más importantes para definir el movimiento geométrico del vehículo:

- Masa: Supuesta de máximo 100 [kg] (considerando eventuales cargas añadidas, pues sin peso era de alrededor de 63 [kg]).
- Distancia entre ejes: $a_1 a_2 = a_1 + a_2 = 846$ [mm], con $a_1 = a_2 = 423$ [mm].
- Trocha: $2b = 500$ [mm].
- Rango angular de los servomotores: 100° .
- Amortiguadores: Kind Shock KS 260 650 LBS, con una constante de resorte, $k = 114$ [N/mm] y un entre eje de fijación de 165 [mm].

Posteriormente, se busca conocer el radio de giro mínimo teórico, $R_{g-\text{mín}}$, que se ve impuesto

¹Inicialmente eran cuatro baterías, una para cada rueda, pero dos de ellas no se conservaron bien desde la primera iteración del 2019 y ya no funcionaban cuando se retomó el proyecto.

por la trocha y la distancia entre ejes, además del rango de los servomotores de dirección:

$$R_{g-\text{mín}} = \sqrt{a_2^2 + \left(b + \frac{a_1 + a_2}{\tan \beta}\right)^2} = 1,05[\text{m}]$$

Con $\beta = 50^\circ$, el rango angular unilateral de cada servomotor.

De la misma forma, hay que considerar que al momento de maniobrar el vehículo, para que pueda girar sin que ninguna de sus ruedas deslice y de forma fluida, según el principio de Ackermann, se necesitan dos cosas:

1. Mantener la misma velocidad angular respecto a un Centro Instantáneo de Rotación (CIR) en el plano del movimiento del vehículo y;
2. Asegurarse de que las ruedas que están siendo giradas sean tangentes a círculos concéntricos.

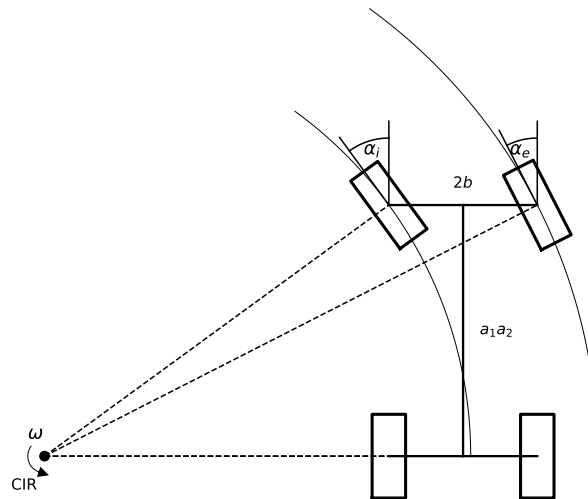


Figura 3: Esquema representando el principio de Ackermann.

Estos dos requisitos se pueden observar de mejor manera en la Figura 3. En ella, podemos identificar que los ángulos de rotación de la rueda externa e interna cumplen con $\alpha_e < \alpha_i$ para que ambas ruedas sean tangentes a circunferencias concéntricas en el CIR; y que, para que no deslice, la velocidad lineal V de cada punto de contacto tiene que ser proporcional a la velocidad angular ω del vehículo alrededor del CIR y a la distancia R entre dicho punto de contacto y el CIR, esto es:

$$V = \omega \cdot R$$

Para el control de los ángulos de rotación de cada rueda, considerando que el ángulo máximo de viraje de cada una de ellas, en cada dirección, es de 50° y que la rueda interna debe rotar más que la externa, el cálculo fue hecho respecto a la interna. De esta forma, se calculó el ángulo de rotación de cada servomotor mediante las siguientes expresiones:

$$\alpha_i = \alpha$$

$$\alpha_e = \frac{\alpha \cdot a_1 a_2 \cdot \sin \alpha}{a_1 a_2 \cdot \cos \alpha + \operatorname{sgn}(\alpha) \cdot 2b \cdot \sin \alpha}$$

Con $\alpha \in [-50^\circ, 50^\circ]$.

Cabe recordar que solo se controlaban las velocidades de las ruedas traseras, y, por tanto, para simplificar los cálculos y el control del vehículo, los valores de velocidad de cada una de ellas se tomaron únicamente en referencia a la rueda externa a la circunferencia instantánea de rotación, dejándonos con la expresión:

$$V_{rueda-externa} = V$$

$$V_{rueda-interna} = V \cdot \frac{a_1 a_2 \cdot \cos \alpha}{2b \cdot \sin \alpha + a_1 a_2 \cdot \cos \alpha}$$

2.2. Puesta en marcha del simulador 3D de control

Primeramente, se estableció un esquema de entradas y salidas desde la MicroAutoBox II (ver Figura 4). En él, podemos ver que los servomotores están ligados directamente al computador, mientras que los motores a CC tienen las placas de desarrollo como intermediarias. El protocolo escogido para comunicar el computador con las placas de desarrollo fue el protocolo CAN (Controlled Area Network), que permite la transmisión de una gran cantidad de información de manera bilateral, continua y confiable. La señal del control remoto es recibida por una de las placas de desarrollo y posteriormente reenviada al computador por el protocolo CAN para que este gestione su redireccionamiento al resto de ruedas y a los servomotores.

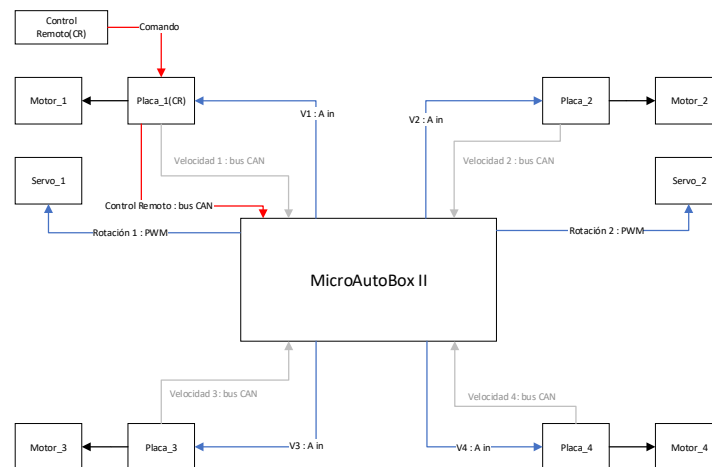


Figura 4: Esquema de entradas y salidas desde el computador embebido.

Con este esquema de comunicaciones listo, se procedió a configurar una simulación del sistema en 3D en Simulink. Para esto, nos basamos en un archivo público de MathWorks, `VehicleDynamics3D.slx`, en el que se simula la dinámica de un vehículo de competición estudiantil mientras se controla la velocidad y el ángulo de viraje con dos perillas virtuales, algo bastante similar a lo que se busca lograr con el control remoto. Luego, para plasmar nuestro vehículo en la simulación, tomamos los archivos CAD de la anterior iteración del

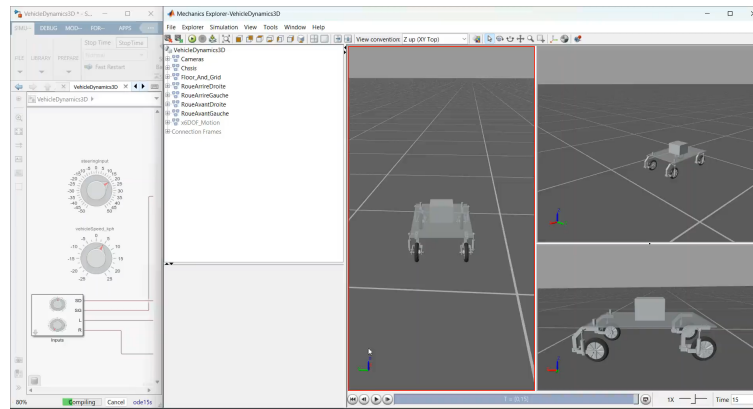


Figura 5: Simulación 3D en tiempo real del prototipo con control de velocidad y viraje.

proyecto en conjunto con nuestras modificaciones y, utilizamos el plugin Simscape Multibody en SolidWorks para crear un modelo Simscape Multibody que representase fielmente al prototipo. Una ventaja de este modelo es que podemos especificar los parámetros de cada unión entre los componentes (resistencia a la fricción, posición de equilibrio y constante de rigidez) y los parámetros de cada componente (masa y densidad) para luego obtener una simulación en 3D, visualmente comprensible, de la cual incluso podemos obtener medidas de sensores instalados virtualmente. De esta manera, se pone en marcha una simulación con control en bucle abierto mediante el ajuste de la velocidad y el ángulo de viraje en tiempo real.

2.3. Modelo de control final

Posteriormente a obtener simulaciones exitosas, se buscó implementar el control en el prototipo físico. Para ello, utilizamos Simulink en conjunto con ControlDesk de dSPACE, para programar así el computador embebido. En este modelo Simulink, y al igual que en el esquema de la Figura 4, consideramos que la entrada a nuestro sistema, es decir, los comandos, vienen desde el control remoto, y son leídos por el sensor de una de las placas de desarrollo y comunicados por protocolo CAN al ordenador embebido. Por ello, se parte con una serie de bloques que se encargan de traducir los comandos de los *joysticks* del control remoto a los valores de ángulo y velocidad deseados para el vehículo. Posteriormente, considerando la velocidad y el ángulo de viraje deseados, se envía a cada rueda motriz el voltaje correspondiente a la velocidad, así como a cada uno de los dos servomotores la señal PWM representando el ángulo de giro. Por último, todo está protegido con elementos que permiten un desempeño más fluido del sistema, como los bloques de saturación y el bloque de retraso de la señal.

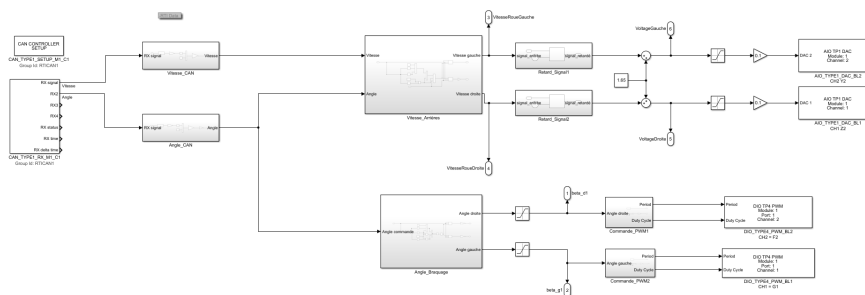


Figura 6: Modelo Simulink de control del vehículo.

El ensayo sobre el prototipo real fue exitoso, pues el control de la velocidad y la dirección funcionó como esperado. Sin embargo, el ordenador embebido debió estar conectado a la corriente todo el tiempo y, por ende, el rango del vehículo se vio limitado a la longitud del cable. Por otra parte, los servomotores fueron dañados en un ensayo previo en que solo se probó la dirección y, en el momento del ensayo de control, la dirección funcionó solo parcialmente.

3. Conclusiones y pistas de mejora

El proyecto mayor en el que se enmarcó este trabajo tiene como objetivo final concebir un vehículo eléctrico autónomo. Para avanzar hacia ello, en esta ocasión nos centramos en establecer un algoritmo de control remoto para dicho vehículo. Para establecerlo se realizaron los siguientes avances en términos generales:

- La modificación del prototipo, al cual se le cambiaron y redistribuyeron componentes para facilitar los desarrollos futuros.
- El cálculo, en base a la geometría de este, para determinar las relaciones entre los ángulos y velocidades de cada rueda y de esta forma implementar el control en bucle abierto.
- La simulación en 3D del control en tiempo real en Simulink y la comprobación del algoritmo en el prototipo con resultados exitosos.

Pese a haber logrado el objetivo principal del proyecto, controlar el vehículo a control remoto, existen pistas de mejora claramente identificadas:

- Las pruebas de control sobre el prototipo fueron exitosas, pero requerían que el computador embebido siguiese conectado a la corriente. Por lo anterior, el alcance de movimiento del vehículo estuvo limitado al largo del cable. Esto puede ser solucionado agregando una batería, en un espacio que ya está designado, para alimentar el computador. Sin embargo, al momento de realizar la investigación, no se contaba con la batería del voltaje adecuado y, utilizar una de las baterías de los motores era arriesgado, pues se encontraba justo sobre el límite superior del voltaje aceptado el dispositivo.
- Los servomotores utilizados y definidos en la primera iteración del proyecto no fueron correctamente escogidos. Por ello, en los ensayos, la unión entre el ensamblaje de la rueda y el chasis se rodó, y la dirección no funcionó como esperado. Para solucionar este problema se deben buscar nuevas alternativas de motores para la dirección de la rueda y rehacer el diseño de esta.
- Aunque el control con bucle abierto resulta satisfactorio como una primera iteración, la utilización del protocolo CAN y de las placas de desarrollo permitirá, en una futura iteración, realizar el control mediante bucle cerrado a través de la recopilación de la velocidad real de las ruedas.

Referencias

Brossard, Jean-Pierre. 2006. *Dynamique du véhicule : modélisation des systèmes complexes*. Collection des sciences appliquées de l'INSA de Lyon mécanique. Lausanne: Presses polytechniques et universitaires romandes.

CSS Electronics. 2024 (Jul). *Can bus explained - a simple intro*.

dSPACE. 2024. *MicroAutoBox II - User documentation*.

MathWorks Student Competitions Team. 2024. MATLAB and Simulink Racing Lounge: Vehicle Modeling with Simscape Multibody. *MATLAB Central File Exchange*.

Microchip. *DsPICDEM™ MCLV-2 Development Board user's guide*.

Oi, Toru, Shinyama, Katsuyoshi, & Fujita, Shigetaka. 2011. Electrical properties of heat-treated poly-lactic acid. *IEEJ Transactions on Fundamentals and Materials*, **131**(5), 395–400.

OZO ELECTRIC. *Electric motor 200mm scooter 8" 250W 350W*.