

Control a Baja Velocidad de una Rueda con Motor de Accionamiento Directo mediante Ingeniería Basada en Modelos

Antonio J. Muñoz-Ramírez, Jesús M. Luque Bedmar, Jesús M. Gómez-de-Gabriel,
Anthony Mandow, Javier Serón, Alfonso García-Cerezo
Universidad de Málaga, Andalucía Tech, Dpto. de Ingeniería de Sistemas y Automática
aj@uma.es

Resumen

Los motores de corriente continua sin escobillas (BLDC) con accionamiento directo suponen una solución compacta para la tracción en vehículos eléctricos, si bien requieren detectar la posición del rotor, habitualmente mediante un codificador angular de efecto Hall dentro del mismo motor. No obstante, la ausencia de reductora y a la dificultad de añadir un codificador angular de precisión suponen un reto para lograr un control preciso a baja velocidad, especialmente si se hace uso de controladoras industriales donde las posibilidades de programación son limitadas. Este trabajo propone aplicar una estrategia de ingeniería basada en modelos (MDE) para el control a baja velocidad de una rueda con motor BLDC de accionamiento directo. En particular, se presenta la solución para un caso de estudio basado en hardware de bajo coste que incluye una tarjeta Arduino Due, una controladora Roboteq HBL2360 y un interfaz de comunicación de bus CAN. La solución MDE basada en Simulink ofrece simplicidad conceptual, capacidad de adaptación a nuevas especificaciones de diseño usando herramientas de diseño avanzadas y generación de código automática. El artículo ofrece resultados experimentales obtenidos sobre el sistema real.

Palabras clave: Ingeniería basada en modelos, MDE, Simulink, control baja velocidad, motor BLDC, Bus CAN, Arduino Due

1. INTRODUCCIÓN

La ingeniería basada en modelos (MDE) permite construir sistemas mediante herramientas de modelado de alto nivel. Los lenguajes de modelado de dominio específico permiten generar código de manera automática, evitando una fase que consume tiempo, recursos y puede introducir errores [2]. Sus beneficios han sido aplicados recientemente a problemas relacionados con la automática, como el diseño de sistemas con brazos manipuladores [3], el diseño de vehículos submarinos no tripulados [7] o como herramienta en prácticas de robótica en estudios de ingeniería [6][5]. El uso de lengua-



Figura 1: El vehículo terrestre no tripulado Rambler incorpora un motor BLDC en cada rueda.

jes de modelado permite introducir el modelo del proceso a controlar como parte del diseño del controlador y ser usado para simulación y ajuste de parámetros. En trabajos anteriores hemos investigado el uso de esta estrategia para la generación automática de código de tiempo real con Simulink de placas Arduino en aplicaciones robóticas [4].

Los motores de corriente continua sin escobillas (BLDC) con accionamiento directo (*direct drive*) suponen una solución compacta para la tracción en vehículos eléctricos [11]. Estos motores requieren de un controlador electrónico para alimentar las bobinas y conseguir su rotación, lo cual se realiza en muchos casos mediante la lectura de las señales de un juego de sensores Hall que obtienen la posición del rotor.

El vehículo terrestre no tripulado Rambler (ver Figura 1) posee cuatro ruedas con motor BLDC de accionamiento directo. El control de velocidad de cada una de las ruedas se realiza mediante una controladora HBL2360 de Roboteq. Las controladoras se conectan mediante un bus CAN para la coordinación de los movimientos del vehículo. Cada rueda incorpora un codificador angular de posición de efecto Hall con una resolución de 138 cuentas por vuelta.

Con esta configuración surgen problemas de bloqueo cuando el motor no produce suficiente par

para generar el movimiento de la rueda. Esta situación se puede originar por fricciones a baja velocidad, sobre todo en el inicio del movimiento del vehículo, o a partir de accidentes del terreno. En estas situaciones de bloqueo se pueden producir acciones de control inadecuadas debido a una interpretación errónea de las lecturas de los sensores Hall. La solución de este y otros problemas prácticos hace necesaria una capacidad de diseño y ajuste del control con mayor flexibilidad que la que ofrecen los controladores comerciales.

En este sentido, resulta de interés la utilización de sistemas embebidos. En [12] se utiliza una placa microcontroladora para realizar control avanzado sobre una controladora industrial de Roboteq. Sin embargo, el modelado en Simulink tan sólo ha sido empleado para la fase de estudio y simulación del sistema.

En este artículo se aborda el control de un sistema físico que replica una de las cuatro ruedas del vehículo Rambler. Con este fin se propone una solución en la que se utilizan herramientas MDE para construir un controlador de velocidad para movimientos de precisión seguros. Para ello se utiliza un microcontrolador de tipo Arduino con una interfaz de comunicaciones CAN, que envía las referencias de par al HBL2360. Se incluye un modelo del motor, cuyos parámetros se han identificado para simulación y ajuste del controlador.

En la sección 2 se describen los componentes del sistema de control. A continuación, la sección 3 ofrece una visión global de la metodología propuesta. La sección 4 detalla el proceso de identificación de los parámetros de la planta. En la sección 5 se describe el proceso de diseño del sistema de control. Por último, se incluyen secciones dedicadas a los experimentos y las conclusiones.

2. DESCRIPCIÓN DEL SISTEMA

2.1. Características generales

En la Figura 2 se muestra al sistema real utilizado como caso de estudio para la solución propuesta. Este sistema constituye una réplica de uno de los cuatro sistemas rueda/motor del vehículo Rambler. El sistema está compuesto por los siguientes elementos:

- Rueda todoterreno con cubierta de tacos de 21 pulgadas de diámetro.
- Motor BLDC de accionamiento directo encasado en la llanta con codificador digital de efecto Hall.



Figura 2: Fotografía del sistema experimental, compuesto por la rueda, la controladora industrial HBL2360 y una placa Arduino Due dotada de un interfaz para comunicaciones con bus CAN.

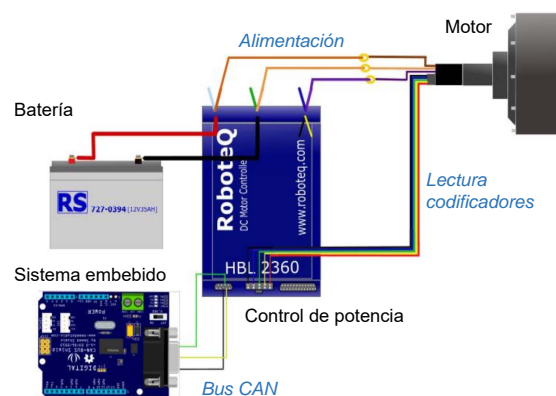


Figura 3: Esquema del conexionado del sistema de control junto con el motor.

- Controladora industrial para dos motores BLDC Roboteq HBL 2360.
- Placa microcontroladora Arduino Due.
- Interfaz de comunicaciones de Arduino para bus CAN de Seedstudio CAN Bus Shield v1.2.
- Baterías. La controladora soporta tensiones entre 12 V y 60 V.

En la Figura 3 aparece el esquema de conexionado de todos los componentes del sistema.

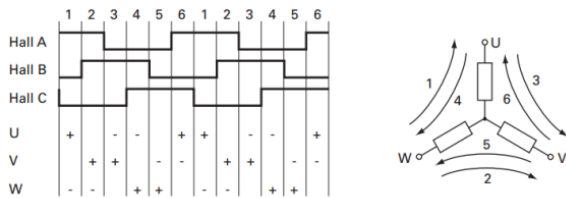


Figura 4: Activación de los sensores digitales de efecto Hall desfasados 120° para bobinas BLDC [9].

2.2. Motor BLDC de accionamiento directo

En los motores BLDC se sustituyen las escobillas responsables del cambio de polaridad por un sistema electrónico, el cual alimenta las bobinas secuencialmente para provocar el giro. Este sistema requiere detectar la posición del rotor para determinar la bobina a alimentar, para lo que se suele contar con sensores Hall.

El motor utilizado en el presente trabajo es un motor BLDC trifásico de corriente continua con 23 polos magnéticos y dos juegos independientes de sensores Hall. Cada juego consta de tres sensores Hall desfasados 120° que proporcionan una resolución de $2,6^\circ$. En la Figura 4 se ilustra la secuencia de activación de las bobinas para producir el giro junto con las señales digitales de los tres sensores Hall.

2.3. Control de Potencia

Para el control de potencia se adopta la controladora Roboteq HBL2360 para motores BLDC [8]. Este dispositivo utiliza la información de los sensores Hall para calcular la velocidad y la posición de los motores. Esta controladora acepta comandos recibidos por USB, RS232, una cadena de pulsos, interfaz CAN y entrada analógica (0 - 5V). Asimismo, dispone de dos canales para manejar dos motores. El interfaz CAN proporciona una velocidad de bus de hasta 1 Mbit/s. En particular, el protocolo MiniCAN se adecua a las exigencias de velocidad de transmisión de datos del bucle de control. La controladora se configura desde un PC a través de puertos USB y RS232 mediante una aplicación proporcionada por el fabricante.

La controladora HBL2360 permitiría implementar soluciones de control simples a través de un PID integrado. Sin embargo, en este trabajo se utilizará únicamente como control de potencia y acondicionamiento de la señal del codificador.

2.4. Sistema Embebido

El algoritmo de control de velocidad se pretende implementar en un sistema embebido consistente en una placa Arduino Due [1], basada en un microcontrolador Atmel SAM3X8E de 32 bits. Dispone de 54 pines de entradas/salidas digitales, de las cuales 12 pueden ser utilizadas como salidas de modulación por ancho de pulso (PWM). También tiene 12 entradas analógicas con resolución de 12 bits, 4 puertos serie (UART) y trabaja con un oscilador de 84 MHz. Dispone de todas sus entradas digitales con la función de interrupción.

La tensión de trabajo de la placa Arduino Due desde el punto de vista de las entradas y salidas es de 3.3 V. Cada pin puede suministrar de 3 mA - 15 mA, dependiendo del pin, o recibir de 6 mA - 9 mA, dependiendo del pin. Estos pines también poseen una resistencia de *pull-up*, que será útil para la lectura directa del codificador.

Si bien este sistema embebido posee comunicación con el bus CAN de forma nativa, la electrónica de adaptación de señales a la de la controladora es más costosa que un interfaz completo para Arduino como la CAN Bus Shield de *Seedstudio*.

3. CONTROL CON ESTRATEGIA MDE

En este trabajo se propone una estrategia MDE para el diseño del bucle de control del sistema en Simulink de tal forma que permita de forma simultánea la simulación en el PC, la programación del sistema embebido y la sintonización del controlador según las especificación de diseño deseadas. Dicho diseño se muestra en la Figura 5 en donde se aprecia la capacidad de admitir consignas de control a través del puerto serie del sistema embebido.

Para la realización de esta estrategia se proponen los siguientes pasos:

1. Implementación de un interfaz Simulink con el proceso real.
2. Identificación de un modelo lineal del proceso a partir de datos de entrada/salida.
3. Sintonización del control PID.
4. Incorporación de elementos no lineales en el control para solucionar el problema del bloqueo.

Todos estos aspectos se tratan en las siguientes secciones.

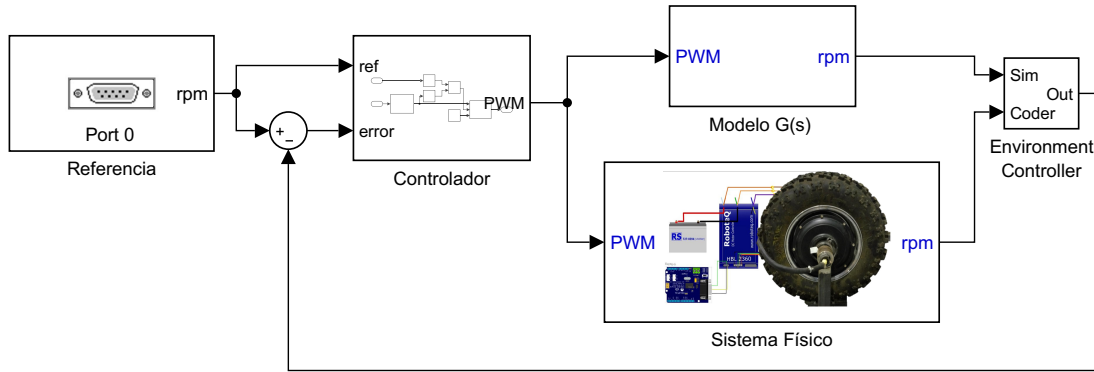


Figura 5: Implementación de la estrategia MDE mediante un modelo Simulink del sistema de control que permite la simulación, la sintonización del controlador y la generación del código para la placa Arduino.

4. IDENTIFICACIÓN DE LA PLANTA

En esta sección se aborda la identificación de la planta la cual consiste en la rueda, el motor BLDC, la controladora industrial HBL2360, y su sistema de actuación (PWM) y lectura de los sensores de efecto Hall dado por la tarjeta controladora Arduino Due.

4.1. Lectura de la velocidad

Como paso inicial en el estudio del sistema hay que realizar la lectura de la velocidad y el envío de la actuación PWM. La activación del motor se puede realizar con una salida PWM (modulación por anchura de pulso) a través de la entrada de RC del HBL2360. En cuanto a la medida de la velocidad, la manera más simple sería a través de las entradas digitales de la placa Arduino enlazadas con rutinas de interrupción con cada flanco, ya sea de subida o de bajada de las señales A, B, o C del codificador. Sin embargo, el resultado de la lectura directa de las señales del codificador y el cálculo de la velocidad midiendo el tiempo entre flancos [10] frente una excitación produce resultados afectados por ruido de alta frecuencia.

Alternativamente, en este trabajo se propone aprovechar el hardware de acondicionamiento de señal de la controladora HBL2360 para extraer la información de la velocidad de la rueda. Esto se puede realizar a través del interfaz bus CAN en conjunción con la excitación del motor, para lo que ha sido necesario programar dos bloques *S-Function Builder* en Simulink con la utilización de un protocolo ligero denominado MiniCAN [9]. El subsistema resultante se observa en la Figura 6.

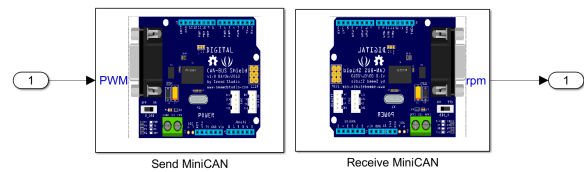


Figura 6: Definición del subsistema Simulink que permite un interfaz con el sistema físico, el cual mediante *S-Function Builder* implementa el envío de la señal PWM a la controladora HBL2360 y la lectura de la velocidad de la rueda a través del bus CAN.

4.2. Identificación del Modelo

La identificación del sistema físico se realiza mediante la herramienta *System Identification Toolbox* de Matlab. Los datos para la identificación consisten en la respuesta del sistema ante una secuencia de escalones PWM que producen una respuesta a baja velocidad (en torno a 23rpm), tal y como se muestra en la Figura 7. Empíricamente se ha comprobado que una función de transferencia dada por dos polos y ningún cero proporciona un buen ajuste (78.01%) con los siguientes parámetros:

$$G(s) = \frac{191,065}{s^2 + 80,739s + 8,051}$$

5. CONTROL A BAJA VELOCIDAD

5.1. Modelado

En el modelo del sistema de control de la Figura 5 se observa como aparecen tanto la interconexión con el Sistema Físico a través del Bus CAN como la función de transferencia que aproxima la

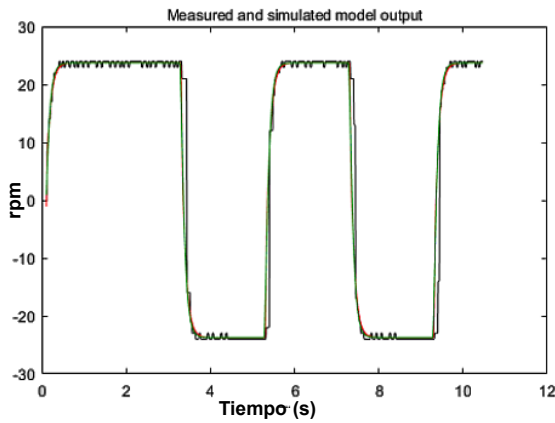


Figura 7: Comparación del señal de salida real del sistema con la del modelo identificado.

dinámica de la planta.

5.2. Sintonización

El modelo propuesto permite la sintonización con la herramienta de Matlab PID Tuner, la cual facilita el ajuste del regulador jugando con especificaciones de diseño tales como la robustez y la rapidez de la respuesta. En la Figura 8 se realiza la sintonización para unas especificaciones de respuesta del sistema de 0.5 s de tiempo de subida y 1 s de tiempo de establecimiento.

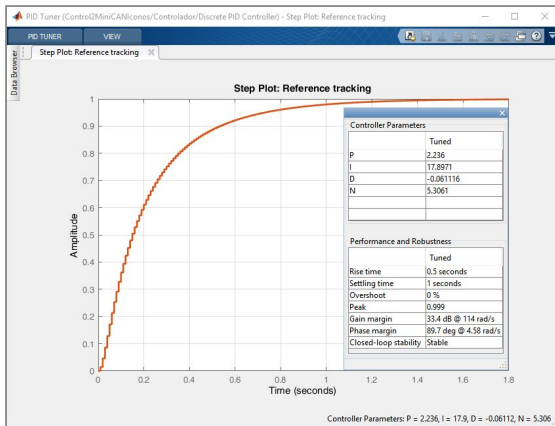


Figura 8: Ventana de la sintonización del controlador del sistema realizada con PID Tuner sobre el propio modelo, donde se obtienen los parámetros del PID en base a cumplir las especificaciones de diseño (en este caso con tiempo de subida 0.5 s y tiempo de establecimiento 1 s)

5.3. El problema del bloqueo

Mediante el modelo de la Figura 5 se permite un rápida sintonización del controlador junto con la aplicación de técnicas de filtrado derivativo, anti-windup y saturación dentro del mismo bloque del

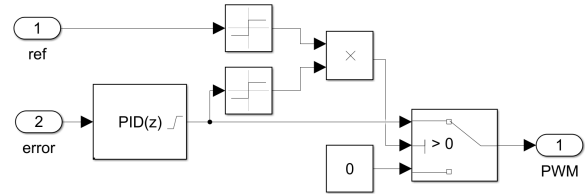


Figura 9: Modelo del controlador con bloque PID y lógica para eliminar retrocesos.

PID discreto. Otro problema que se puede abordar mediante la estrategia MDE surge en la aplicación práctica del control cuando se produce el bloqueo de la rueda.

En caso de bloqueo, se ha observado que uno de los tres sensores Hall se mantiene oscilando en su zona de flanco, lo cual es interpretado erróneamente por el controlador como una lectura de alta velocidad en el sentido de la marcha. Para compensar esta falsa lectura, la acción de control provoca súbitamente un giro a alta velocidad en sentido contrario al de la marcha.

A fin de solucionar este problema, se propone la solución que se ilustra en la Figura 9 para el subsistema de control. Esta solución permite detectar una diferencia de signo entre la referencia y el resultado del algoritmo PID, en cuyo caso se genera una actuación nula.

6. RESULTADOS EXPERIMENTALES

Los experimentos realizados demuestran por un lado la validez de la solución propuesta y por otro la robustez frente a perturbaciones. Las pruebas se han realizado con alimentación de batería de 12 V, si bien tanto la controladora como el motor admiten hasta 60 V. De esta manera, se persigue probar el sistema de control en situaciones de bajo par.

6.1. Validación del modelo

El experimento consiste en comparar la respuesta en bucle cerrado de la planta real y del modelo identificado ante una consigna de 100 rpm. El resultado de este experimento se observa en la Figura 10, donde se aprecia bastante similitud en especial en los parámetros de diseño de tiempo de subida y tiempo de estabilización. Este resultado indica que la sintonización automática a partir del modelo identificado resulta adecuada para el control de la planta real.

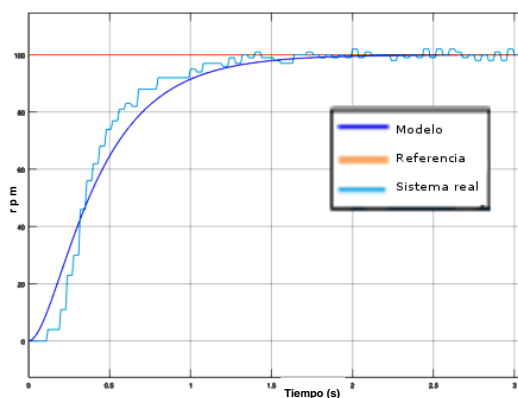


Figura 10: Comparación entre la velocidad del sistema controlado y el modelado frente una referencia de 100 rpm.

6.2. Robustez a perturbaciones

El experimento para determinar la robustez en presencia de perturbaciones se realizó aplicando una referencia continua en el tiempo y frenando manualmente la rueda con una palanca de tal forma que la fuerza ejercida sobre la rueda fuera capaz de bloquearla. Una vez puesta en marcha la rueda y estabilizada su velocidad a la de referencia (50 rpm) se frenó paulatinamente hasta el bloqueo en dos ocasiones, y posteriormente se desbloqueó también de forma pausada. Por último, se realizó una frenada suave y su consiguiente retirada también suave. La gráfica con las señales de salida obtenidas se puede visualizar en la Figura 11. Donde se aprecia que en los tres casos el controlador aumentó la actuación para mantener la referencia deseada.

7. CONCLUSIONES

En este artículo se ha abordado el control a baja velocidad de una rueda con motor de corriente continua sin escobillas (BLDC) de accionamiento directo. Concretamente, se ha ofrecido una solución para un caso de estudio consistente en la réplica de una de las cuatro ruedas del vehículo eléctrico Rambler. Con este fin, se ha propuesto una solución en la que se utilizan herramientas de ingeniería basada en modelos (MDE) a través de Simulink con generación de código automático que se ejecutará en un microcontrolador de tipo Arduino. Esta estrategia requiere identificar la dinámica de la planta para sintonizar del controlador a partir de simulaciones.

Se ha diseñado un controlador PID discreto, que evita las sobreoscilaciones y es robusto ante la presencia de perturbaciones. Asimismo, se ha incorporado al PID una solución específica para detec-

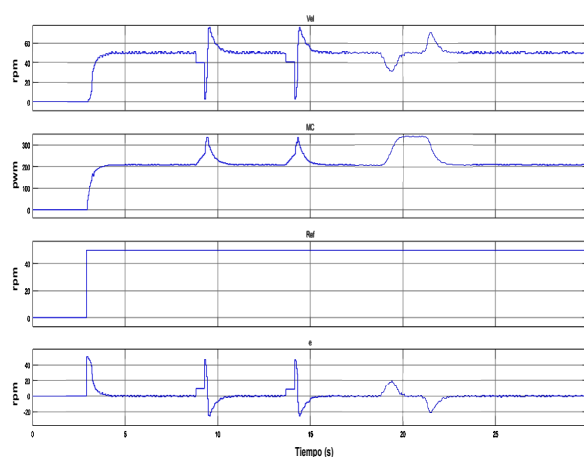


Figura 11: Resultados del sistema controlado frente una entrada en escalón al que se le han aplicado tres perturbaciones externas: dos bloqueos de la rueda para después liberarla, y una frenada y liberación suave. Las gráficas de tiempo por orden de arriba a abajo son: velocidad de la rueda en rpm, actuación en PWM, referencia en rpm, y error cometido en rpm.

tar situaciones de bloqueo de la rueda. Los resultados experimentales han mostrado un buen comportamiento del sistema controlado ante perturbaciones de bloqueo y en arranque a bajas velocidades.

La continuación de este trabajo pasa por su implantación en el vehículo robótico Rambler, en donde se aplicará la ingeniería basada en modelos en las etapas de identificación y sintonización para sus cuatro ruedas.

Agradecimientos

Este proyecto ha sido parcialmente financiado por el proyecto de Innovación educativa de la Universidad de Málaga PIE 15-180 y por el proyecto CICYT DPI2015-65186-R.

Referencias

- [1] Arduino (2014). Arduino - ArduinoBoardDue.
- [2] Balasubramanian, K., Gokhale, A., Karsai, G., Sztipanovits, J., and Neema, S. (2006). Developing applications using model-driven design environments. *Computer*, 39(2):33–40.
- [3] Estévez, E., Sánchez-García, A., Gámez-García, J., Gómez-Ortega, J., and Satorres-Martínez, S. (2016). A novel model-driven approach to support development cycle of robotic systems. *The International Journal of Advanced Manufacturing Technology*, 82(1-4):737–751.
- [4] Gil Lozano, J. E., Muñoz Ramírez, A. J., Torres, V. L., and Gomez, J. M. (2014). Uso de

- Simulink y Arduino para prácticas de robótica. In *Jornadas de Automática*, pages 3–5, Valencia, Spain. Comité Español de Automática (CEA-IFAC).
- [5] Muñoz-Ramírez, A. J. and Gómez-de Gabriel, J. (2016). Modelar o programar en prácticas de robótica. In *Jornadas de Automática*, pages 1–7, Madrid. Comité Español de Automática (CEA-IFAC).
- [6] Muñoz-Ramírez, A. J., Jesús Fernández-Lozano, J., and Manuel, J. (2015). Ingeniería Basada en Modelos en Prácticas de Robótica. In *Jornadas de Automática*, pages 624–630, Bilbao. Comité Español de Automática de la IFAC (CEA-IFAC).
- [7] Ortiz, F. J., Insaurralde, C. C., Alonso, D., Sánchez, F., and Petillot, Y. R. (2015). Model-driven analysis and design for software development of autonomous underwater vehicles. *Robotica*, 33(8):1731–1750.
- [8] Roboteq (2015). HBL23xx motor controller datasheet.
- [9] Roboteq (2016). RoboteQ Advanced Brushed and Brushless Digital Motor Controllers User Manual.
- [10] Romero, A., Muñoz-Ramírez, A. J., and Gómez-de Gabriel, J. (2015). Realimentación de velocidad con encoders de baja resolución en Simulink. In *Actas de las XXXVI Jornadas de Automática*, pages 215–220.
- [11] Salehifar, M., Arashloo, R. S., Moreno-Eguilaz, M., Sala, V., and Romeral, L. (2015). Observer-based open transistor fault diagnosis and fault-tolerant control of five-phase permanent magnet motor drive for application in electric vehicles. *IET Power Electronics*, 8(1):76–87.
- [12] Viguria, A., Prieto, A., Fiacchini, M., Cano, R., Rubio, F. R., Aracil, J., and Canudas-De-Wit, C. Desarrollo y experimentación de un vehículo basado en péndulo invertido (PPCar). *Revista Iberoamericana de Automática e Informática Industrial*, 3(4):53–62.