

Control y programación de un cargador de baterías para drone

COMELLI, Román⁽¹⁾; ALBA, Daniel^(1,*); VAZQUEZ SIEBER, Andrés⁽²⁾; MORZAN, Renzo⁽¹⁾; JUNCO, Sergio⁽¹⁾

Escuela de Ingeniería Electrónica; Facultad de Ciencias Exactas, Ingeniería y Agrimensura; Universidad Nacional de Rosario

⁽¹⁾LAC, Laboratorio de Automatización y Control; ⁽²⁾AEP, Área de Electrónica de Potencia
{rcomelli,avazquez,rmorzan,sjunco}@fceia.unr.edu.ar

^(*)Ing. Daniel Alba in memoriam, integrante del equipo de desarrollo de este proyecto, fallecido el 16 de abril de 2024.

Palabras claves: Modelado, simulación y control; Model-, Software-, Processor-in-the-Loop; Diseño de firmware

INTRODUCCIÓN

Los progresos en tecnologías relacionadas con el control, las baterías y las cámaras, junto con la disminución de los costos, han contribuido a que los drones se conviertan en dispositivos ampliamente utilizados y difundidos en diversos sectores. En este contexto, la empresa local TECSO decidió enfocarse en el ámbito de la seguridad, proponiéndose desarrollar un drone de vigilancia para el cual, mediante el trabajo conjunto del LAC y el AEP, la Facultad de Ciencias Exactas, Ingeniería y Agrimensura tuvo la responsabilidad de diseñar, construir y probar un cargador de baterías que cumpliera con ciertos requisitos específicos. Aquí se presenta lo hecho para dicho proyecto, puntualmente en el área de control y programación.

MODELADO Y CONTROL

La topología elegida fue la buck-boost de cuatro llaves por poder reducir y elevar tensión continua eficientemente, por permitir un flujo bidireccional de potencia y porque disponíamos de un diseño de referencia basado en la misma. La Figura 1 muestra el modelo promediado de dicha topología (detalle del bloque “4 Switch Buck-Boost”) así como también la estructura del controlador: lazos anidados de corriente y tensión. Múltiples simulaciones se realizaron a partir de este modelo para validar el ajuste de los controladores, incluyendo respuesta al escalón en cada uno de los 4 modos de operación, procedimiento de carga completo, contrastaciones con modelo conmutado y perturbaciones de entrada. Un detalle adicional a mencionar sobre el controlador de corriente es que el mismo requiere una inicialización de su variable de estado y de su modo para funcionar correctamente. Esto se hace a partir de mediciones iniciales de tensión de entrada y salida.

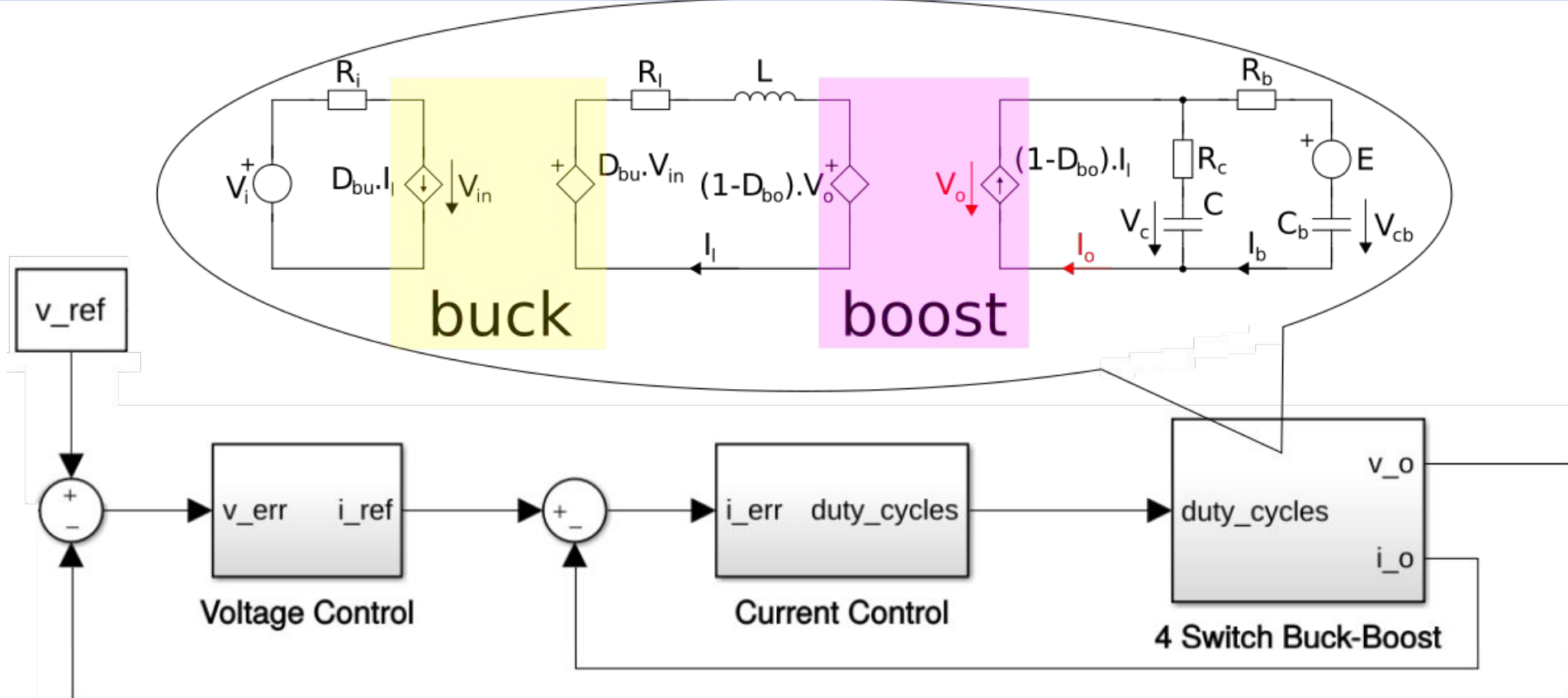


Figura 1. Modelo con controlador.

PROGRAMACIÓN

Para desarrollar el firmware del microcontrolador, se analizaron varias arquitecturas diferentes hasta converger a una basada en los patrones de diseño de Objetos Activos [1] y Máquinas de Estado Jerárquicas (HSMs) [2]. En esencia, el sistema funciona a partir de un conjunto de máquinas de estado finito que corren independientemente en sus propios hilos de ejecución y se comunican mediante eventos. La principal HSM se encarga de la gestión general de la operación del cargador, lo que involucra el arranque, el normal funcionamiento y la detención. Otras HSMs se encargan del manejo de la comunicación, de los pulsadores y de las protecciones por software, entre otras cosas. Cada HSM tiene una prioridad y, en función de eso, un scheduler las ejecuta.

TESTEO ASISTIDO POR SIMULACIÓN: MiL, SiL, PiL

Distintas técnicas de testeo asistido por simulación fueron utilizados para este desarrollo. El descripto previamente al hablar de modelado y control, mediante el uso del programa Simulink en una computadora, se denomina Model-in-the-Loop (MiL). Se emplearon también técnicas llamadas Software-in-the-Loop (SiL) y Processor-in-the-Loop (PiL). Para la primera de estas, el firmware se implementó en el lenguaje a utilizar en el microcontrolador pero se ejecutó también en una computadora aunque por fuera de Simulink con lo que solo el sistema físico corría dentro de dicho entorno. A diferencia de esto, al hacer PiL, el control así como también las HSMs que hacen a la operación del equipo se ejecutaban efectivamente en el microcontrolador, es decir, en la plataforma de cómputo final que utilizamos para el cargador de baterías. El sistema físico, al igual que antes, se simulaba en la computadora nuevamente con Simulink. La utilización de estas metodologías, especialmente la de PiL, permitió la familiarización temprana con el microcontrolador y sus limitaciones que eran la baja velocidad y la ausencia de Unidad de Punto Flotante. Hay que destacar que esto último involucró la necesidad de utilizar punto fijo y realizar muchos ajustes de escalas, para lo cual trabajar directamente con el microcontrolador fue de gran ayuda.

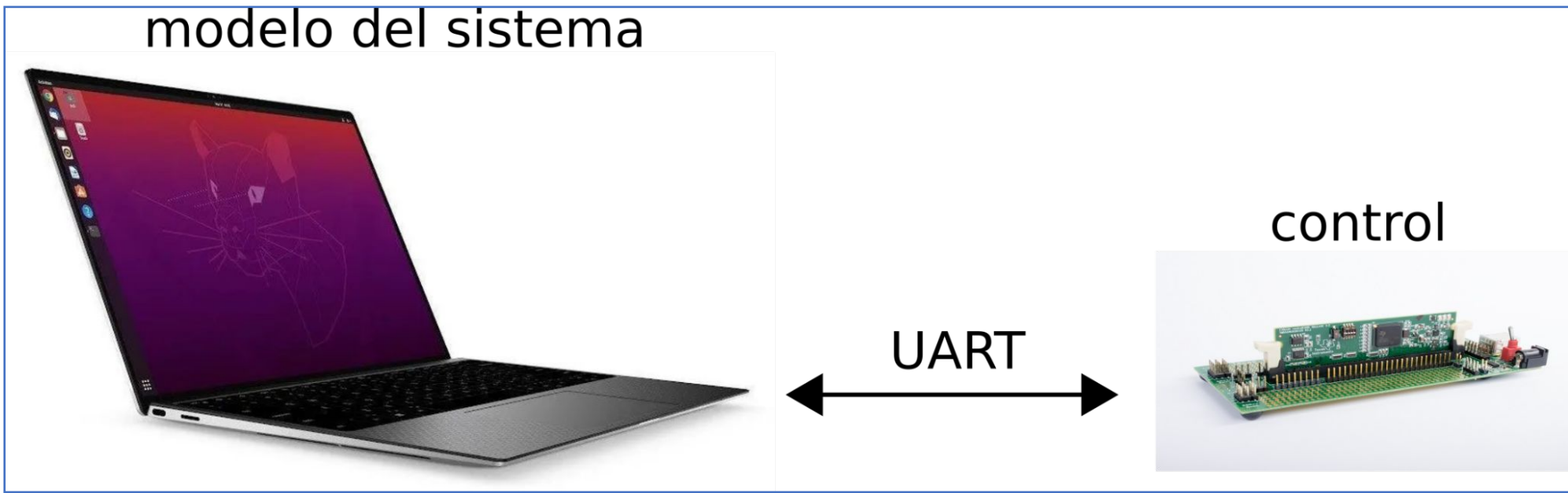


Figura 2. Esquema processor-in-the-loop.

CONCLUSIONES

Se desarrolló y verificó con éxito un cargador basado en un convertidor buck-boost de 4 llaves, incluyendo el diseño de su controlador, la implementación de un firmware orientado a eventos y su validación mediante simulaciones y ensayos experimentales. Tras resolver problemas de hardware detectados durante las pruebas, el sistema mostró un desempeño satisfactorio tanto en régimen dinámico como estático. Como trabajo futuro, se prevé evaluar la eficiencia del sistema en operación normal y su comportamiento ante condiciones adversas como la desconexión súbita de la batería.

REFERENCIAS

[1] Schmidt, D. C., Stal, M., Rohnert, H., & Buschmann, F. (2013). *Pattern-oriented software architecture, patterns for concurrent and networked objects*. John Wiley & Sons.
[2] Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3), 231-274.