



**Mondragon
Unibertsitatea**

Faculty of
Engineering

Plataforma HIL para la validación del control de un motor DC

Master en Sistemas Inteligentes de Energía

ÍNDICE

<u>1</u>	<u>PROPÓSITO DEL DOCUMENTO</u>	<u>2</u>
<u>2</u>	<u>PLATAFORMA MODEL IN THE LOOP</u>	<u>2</u>
2.1	DESCRIPCIÓN DE LA PLATAFORMA	2
2.2	RESULTADOS DE SIMULACIÓN	3
<u>3</u>	<u>PLATAFORMA HARDWARE IN THE LOOP</u>	<u>7</u>
3.1	DISEÑO HARDWARE	7
3.2	DISEÑO SOFTWARE	7
3.2.1	ESTRUCTURA DE LA SIMULACIONES EN TIEMPO REAL	7
3.2.2	ADAPTACIÓN DE LA SIMULACIÓN MIL	8
3.2.3	MEDICIÓN DEL TIEMPO DE EJECUCIÓN	10
3.2.4	MONITORIZACIÓN DE LAS VARIABLES	12
<u>4</u>	<u>VERIFICACIÓN DE LA PLATAFORMA HIL</u>	<u>14</u>
<u>5</u>	<u>ANEXO. PROGRAMACIÓN DE LAUNCHPAD F28379D MEDIANTE MATLAB/SIMULINK</u>	<u>16</u>

1 PROPÓSITO DEL DOCUMENTO

El objetivo de este documento es mostrar el desarrollo y la implementación de una plataforma HIL para un patinete eléctrico.

La Figura 1 muestra un esquema simplificado del sistema a simular.

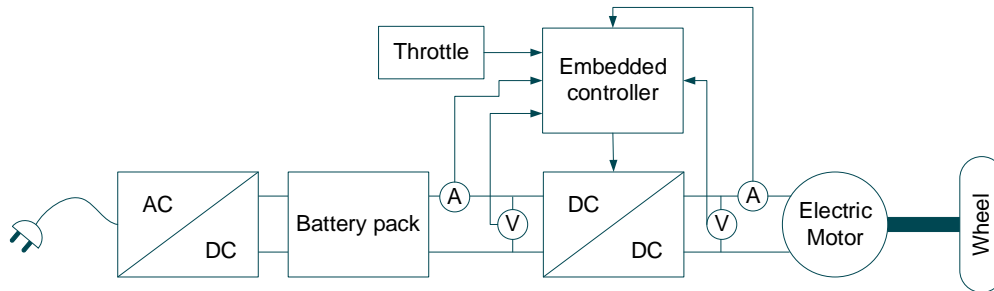


Figura 1 Estructura del patinete eléctrico

Se han abordado tanto el desarrollo hardware como el desarrollo software.

En cuanto al hardware, la plataforma HIL está compuesta por dos tarjetas Texas F28379D. Una actúa como controlador y otra emula la planta en tiempo real (no se consideran la batería y el convertidor AC/DC).

Para el desarrollo software, se ha partido de una simulación convencional MIL que se ha transformado para su ejecución en tiempo real.

Para que los resultados de la HIL se acerquen lo máximo posible a la MIL se tienen en cuenta diferentes factores que pueden tener influencia en la simulación en tiempo real en las tarjetas, como por ejemplo: el offset de las entradas y salidas analógicas, asegurar el orden correcto de los cálculos en la simulación en tiempo real, el reseteo de los integradores, o el tiempo de ejecución.

Una vez adaptada la simulación HIL, mediante una simulación externa que monitoriza los datos desde el PC (HOST) se han podido recoger los datos necesarios para comparar los resultados con los de la simulación MIL.

2 PLATAFORMA MODEL IN THE LOOP

2.1 Descripción de la plataforma

El desarrollo de la plataforma HIL ha partido de una simulación MIL del control de un motor DC usado para un patinete eléctrico. En la Figura 2 se puede ver la estructura de la simulación.

Electric Scooter Digital Twin prototype (medium fidelity)

Medium fidelity model is used to design the control strategy of the motor.
Detailed motor model and average chopper model.

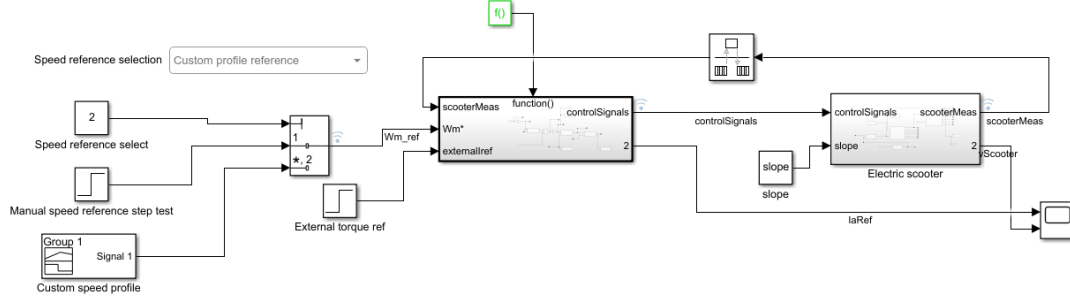


Figura 2 Estructura del patinete eléctrico

La planta está compuesta por un subsistema que simula el comportamiento de un motor DC, para después obtener el par generado y calcular la velocidad que logrará el patinete eléctrico simulando el modelo mecánico de este.

En el caso del motor DC y el sistema mecánico su comportamiento se describe mediante las ecuaciones (1), (2), (3) y (4):

$$\frac{dI_a}{dt} = \frac{V_L}{L_a} = \frac{V_a - \omega \cdot K_e - R_a \cdot I_a}{L_a} \quad (1)$$

$$T_{em} = I_a \cdot K_t \quad (2)$$

$$T_m = T_{em} - \omega \cdot K_f - \omega \cdot D \quad (3)$$

$$\frac{dv_{scooter}}{dt} = \frac{2}{m_{total}} \cdot \left(T_m - \frac{n_{gear}}{eff_{gear} \cdot R_{wheel}} - F_{drag} - F_{hc} - F_{rolling} \right) \quad (4)$$

El control usa la corriente medida en la planta para compararla con la referencia; y con el error conseguido, hacer un control mediante un controlador PI. Para obtener el ciclo de trabajo (δ) del chopper, se utiliza un modelo medio que se representa mediante la ecuación (5):

$$\delta = \frac{V_a}{V_{bat}} \quad (5)$$

2.2 Resultados de simulación

De cara a verificar el funcionamiento de la MIL se han realizado diversas simulaciones. Las siguientes figuras (Figura 3, Figura 4, Figura 5, Figura 6, Figura 7, Figura 8) muestran el control de corriente y la evolución de la velocidad del scooter. En este caso se ha simulado el control de corriente sin ningún control de velocidad.

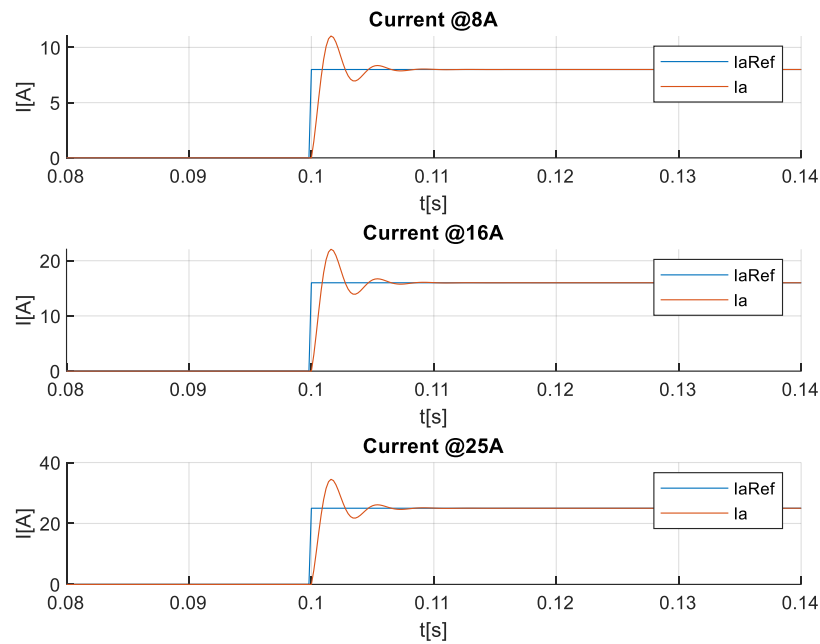


Figura 3 Evolución de la corriente a diferentes referencias

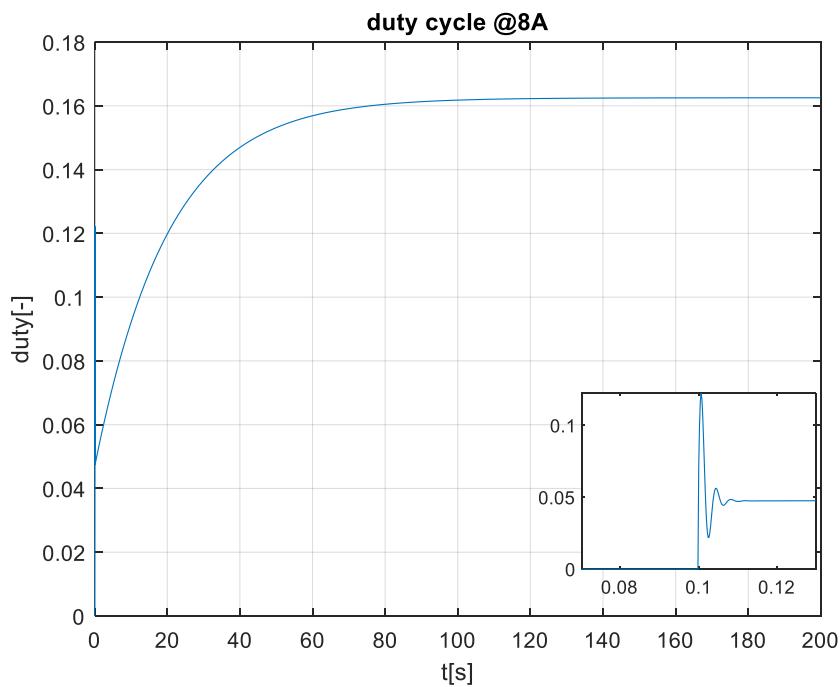


Figura 4 Ciclo de trabajo del motor a 8 A

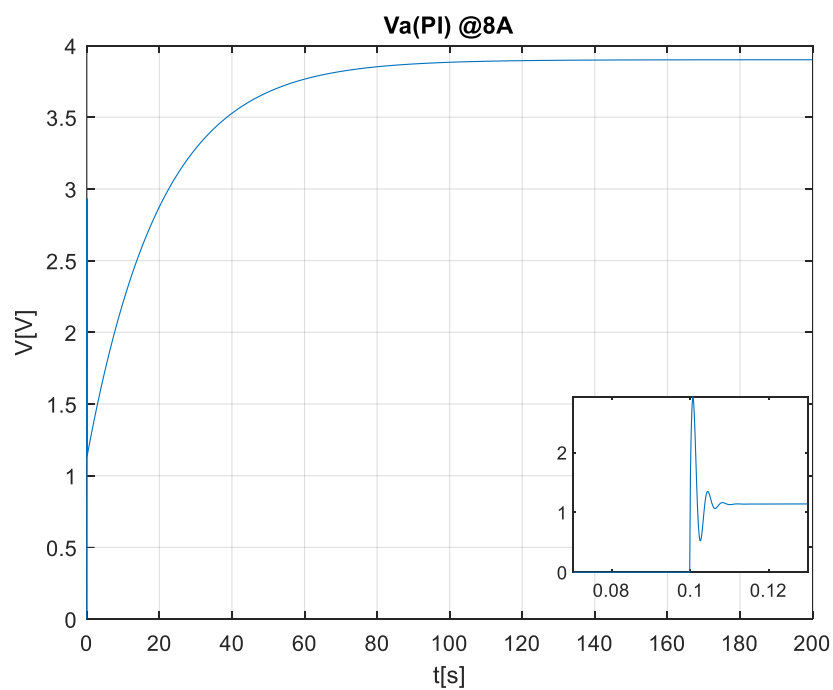


Figura 5 Tensión del motor a 8 A de referencia

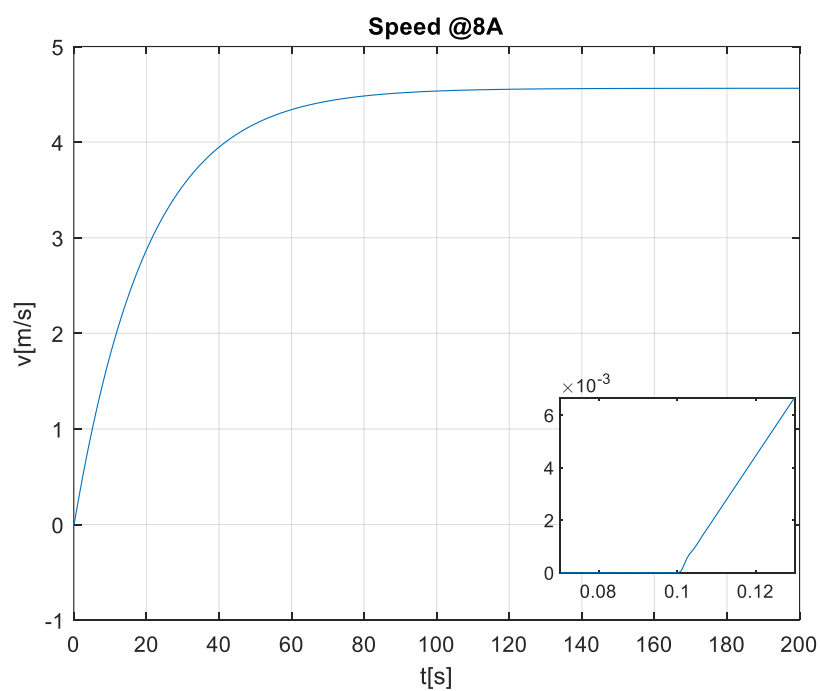


Figura 6 Velocidad del motor a 8 A de referencia

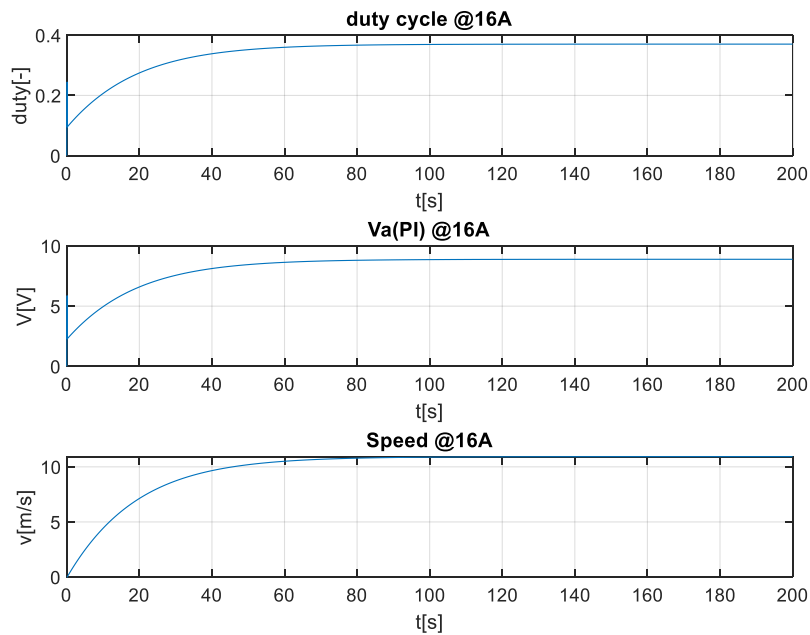


Figura 7 Ciclo de trabajo, tensión y velocidad del motor a 16 A de referencia

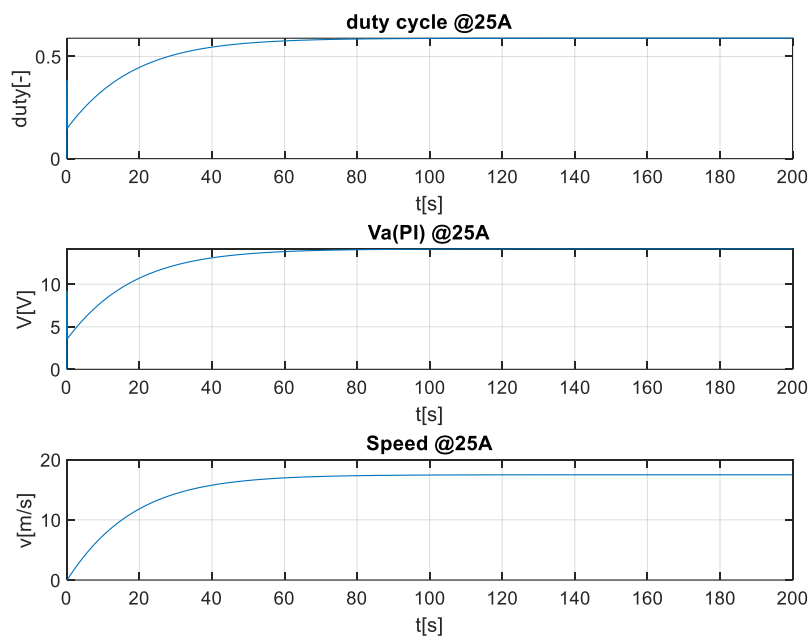


Figura 8 Ciclo de trabajo, tensión y velocidad del motor a 25 A de referencia

Como se puede comprobar, cada escalón de corriente provoca un aumento exponencial en la velocidad, hasta llegar al régimen permanente que es el resultante de las fórmulas anteriores.

3 Plataforma Hardware in the loop

3.1 Diseño hardware

La plataforma HIL está compuesta por dos tarjetas TI Delfino F28379D LaunchPad simulando una el control y otra la planta, conectadas por sus ADC y DAC. La Figura 9 muestra la estructura general de la plataforma.

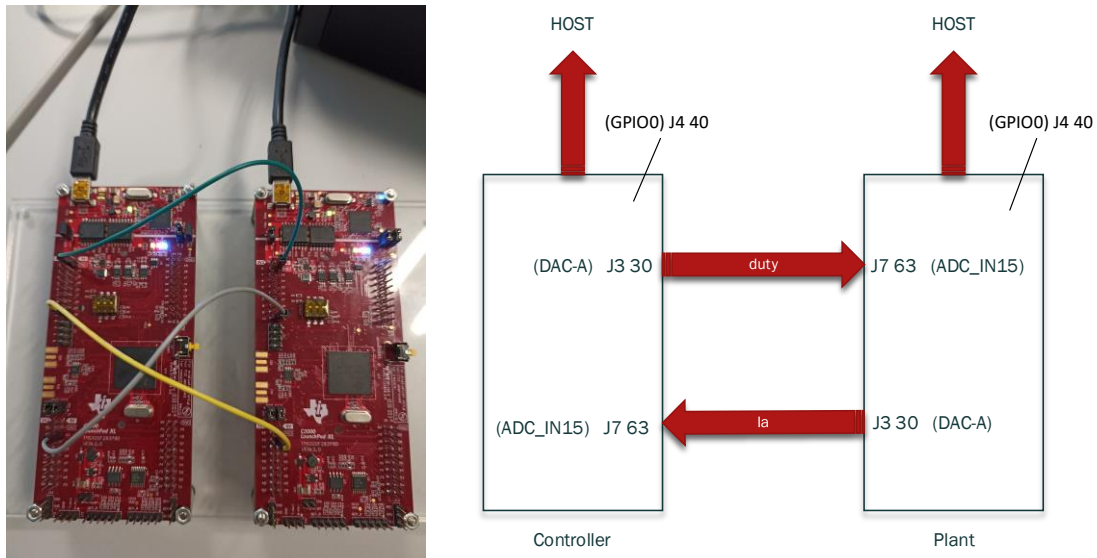


Figura 9 Diseño del Hardware

Lo primero es conectar los GND de las placas para que tengan una misma referencia.

Una tarjeta ejecuta la parte de control recibiendo la corriente desde la otra tarjeta con el uso de los ADC y DAC. Definiendo los bloques correspondientes en la simulación, podemos enviar el ciclo de trabajo a la otra planta desde el DAC-A del control y el ADC-IN15 de la planta. A su vez, se se envían varias variables para llevarlos al programa HOST para analizarlos.

La otra ejecuta la simulación de la planta, con los modelos del motor DC y el sistema mecánico. Al recibir el ciclo de trabajo desde la tarjeta de control y hacer todo el proceso, se obtiene la corriente del motor y se hace el mismo puente de DAC y ADC, pero esta vez al revés. Similar al control, se envían varias variables al HOST para comparar los resultados de la planta con los del control.

Después, el programa HOST ejecutado en el PC enviará variables como la corriente de referencia y el valor Reset de los integradores para poder manipular las dos simulaciones en tiempo real desde un mismo modelo. Al obtener las variables internas calculadas por las tarjetas, se puede visualizar la evolución de la corriente o el ciclo de trabajo directamente en el HOST; pudiendo ver las diferencias entre el control y la planta, causadas por factores como el offset.

3.2 Diseño software

3.2.1 Estructura de la simulaciones en tiempo real

Para simular las dos partes en tiempo real, y asegurar el orden correcto de cálculos en las simulaciones, se usa estructura compuesta por subsistemas en serie que necesitan recibir datos de un subsistema anterior para comenzar a funcionar. Tres subsistemas

bastan para lo que se necesita en este caso (ver **¡Error! No se encuentra el origen de la referencia.**Figura 11).

El primero y el último gestionan una señal de cambio de estado para una GPIO, para después observar el tiempo de ejecución entre el primer subsistema y el último mediante un osciloscopio (más detalles en sección 3.2.3). También en el primero se crea la señal que habilita la ejecución del segundo subsistema. De esta forma se asegura que las tareas se ejecutan en secuencia.

El segundo contiene toda la simulación de control/planta explicado anteriormente en la sección 2.1, partido en dos justo donde se obtiene el ciclo de trabajo y la corriente (ver Figura 10). Además, se añaden bloques SCI que envían datos al HOST fuera del subsistema, ya que estos funcionan a tiempos de muestreo distintos y un bloque enable no permite tiempo de ejecución diferentes.

Además, en uno de los modelos se añade en el subsistema final un contador de ciclo que va aumentando por cada ejecución. De esta manera, teniendo el valor del contador en el HOST, se puede calcular el tiempo que lleva ejecutándose cada tarjeta y se pueden mostrar los resultados correctamente. Cabe destacar que la ejecución del HOST no es en tiempo real por lo que las variables monitorizadas a través del HOST se representan con eje de tiempo diferente.

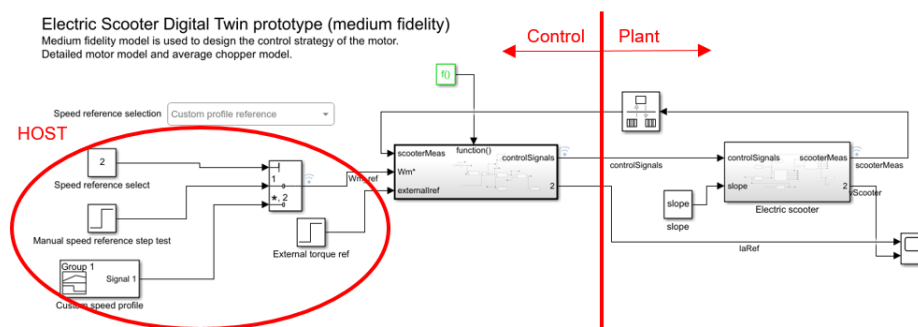


Figura 10 Separación de Planta y Control partiendo de la estructura del patinete eléctrico

3.2.2 Adaptación de la simulación MIL

La simulación MIL se ha tenido que adaptar para su ejecución en tiempo real. La Figura 11, la Figura 12 y la Figura 13 muestran los modelos preparados para su ejecución en tiempo real.

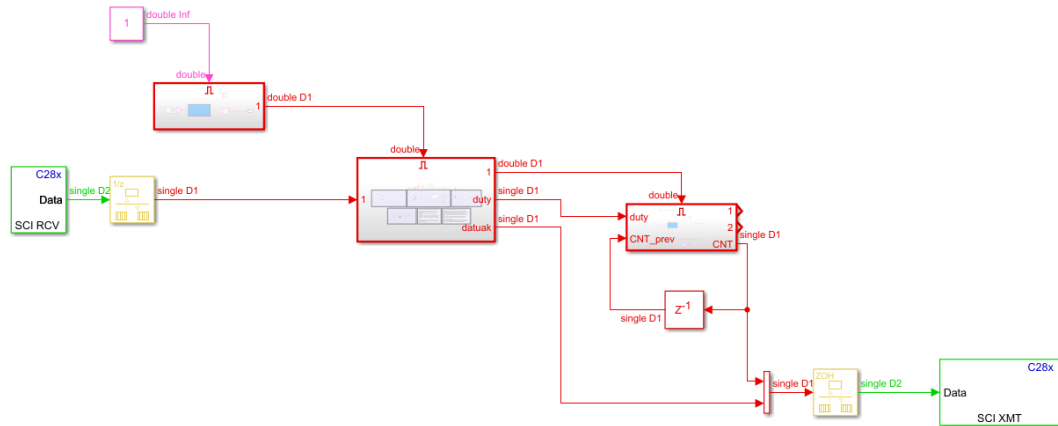


Figura 11 Estructura de la simulación para las tarjetas

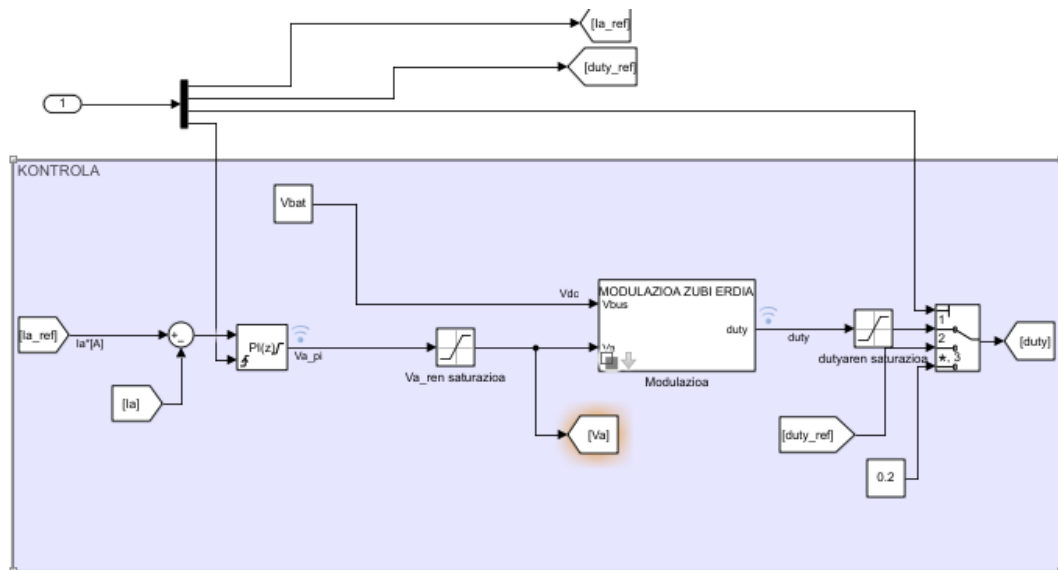


Figura 12 Estructura de control

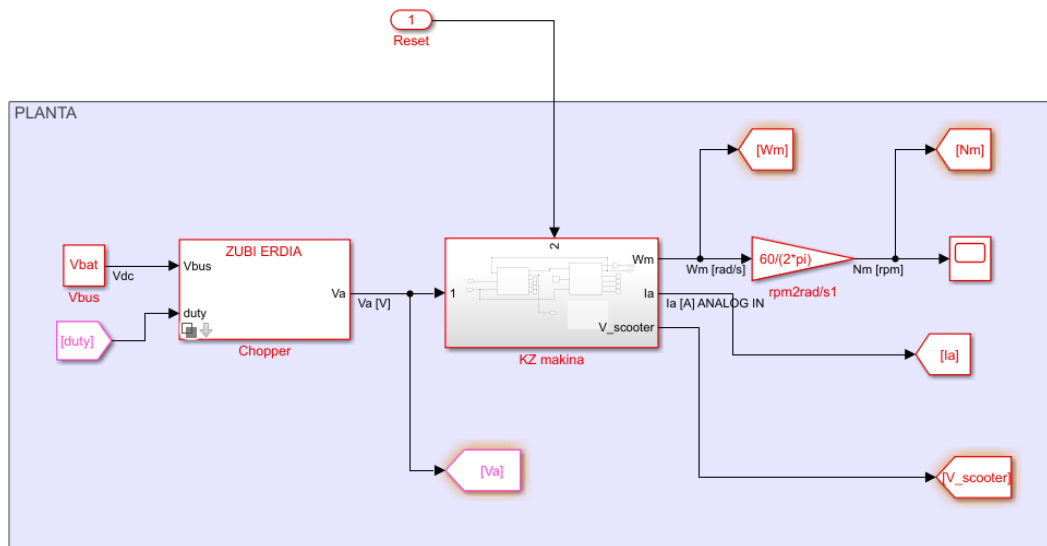


Figura 13 Estructura de planta

En estos modelos las adaptaciones principales han sido las lecturas y escrituras de las señales a través de canales ADC/DAC y el reset de los integradores.

El uso de canales ADC/DAC ha exigido su calibración para evitar errores de medida. La calibración del offset de las IO analógicas se ha hecho mediante una recogida de datos en excel. Teniendo las dos placas; se envía una constante desde el DAC de uno hasta el ADC del otro; y al ver que la desviación era exponencial, se ha decidido hacer lo siguiente: restar el offset del valor en 0 y aplicar una ganancia que disminuya esta subida exponencial, calculando la media de ganancia necesaria para aproximar los datos en un rango de datos recogidos.

Tabla 1 Calibración de las placas

Referencia placa 1	0	50	100	200	500	1000	1500	2048	3000	3900	3934	4095
Valor placa 2	10	61	115	220	529	1040	1557	2118	3107	4061	4095	4095
Offset	10	11	15	20	29	40	57	70	107	161	161	0
Ganancia	0	0,98039216	0,95238095	0,95238095	0,96339114	0,97087379	0,96961862	0,971537	0,96867937	0,96272525	0,9630355	1,00244798

Media	0,96886025
-------	------------

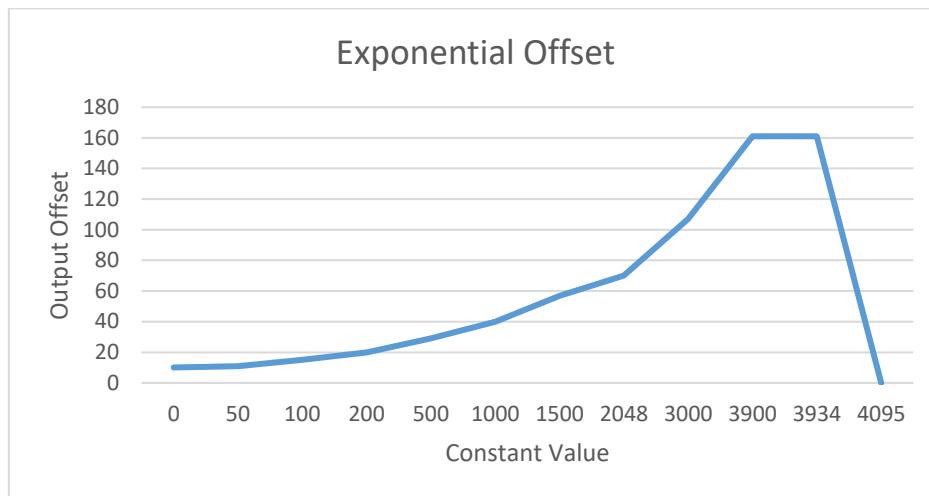


Figura 14 Gráfica del aumento exponencial del Offset Analógico

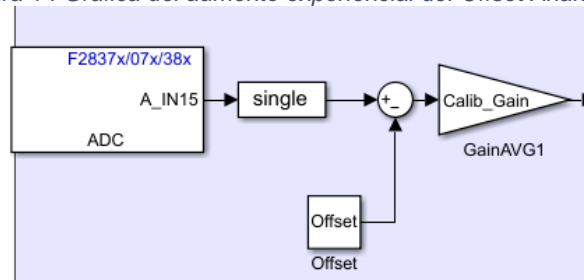


Figura 15 Resta del Offset y ganancia de calibración

Tabla 2 Resultados después de la corrección

Referencia placa 1	0	50	100	200	500	1000	1500	2048	3000	3900	3934	4095
Valor placa 2	0,97	51,35	102,7	204,4	502,8	993,1	1490	2027	20965	3868	3909	3955

Por otro lado, se ha visto que entre ensayo y ensayo se deben volver a inicializar los valores anteriores de los integradores. Si se quieren realizar diferentes ensayos sin tener que volver a cargar y ejecutar los modelos de nuevo, se deben resetear los integradores de los controladores y la planta. Si no, el siguiente ensayo comienza desde el punto donde acabó el anterior y se obtienen resultados erróneos.

3.2.3 Medición del tiempo de ejecución

Como se ha explicado anteriormente, en los subsistemas enable primero y último se cambia de estado el GPIO, y al medir con un osciloscopio en el pin correspondiente se puede ver el tiempo que tarda la simulación en empezar y terminar el ciclo, y también se observa cada cuanto se repite este proceso. El tiempo de repetición de ciclos debería ser el Sample Time definido en el Solver para que sea correcto, y lógicamente; cuanto más peso tenga la simulación, más tiempo de ejecución necesitará. Por eso, es importante verificar que el tiempo de ejecución de las tareas no es superior al paso de ejecución impuesto (ver Figura 16)

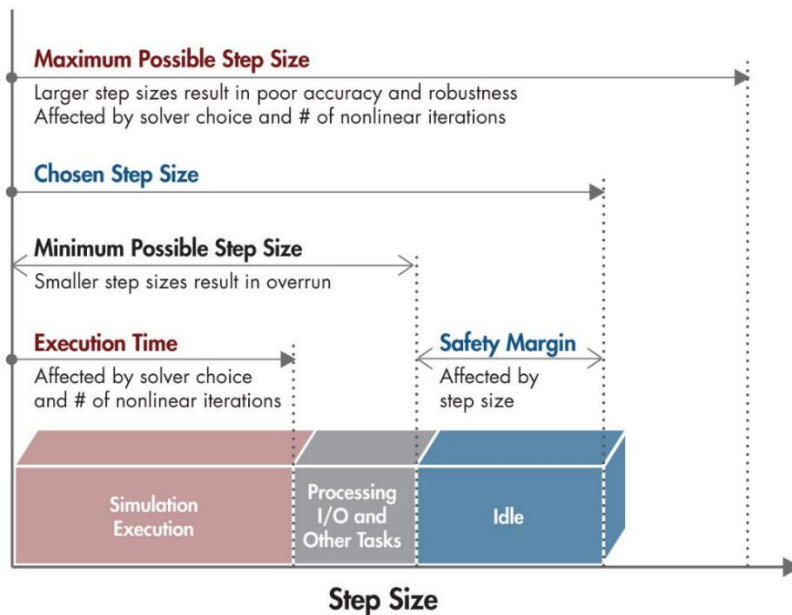


Figura 16 Diagrama de tiempos de ejecución en tiempo real
(<https://www.mathworks.com/company/newsletters/articles/real-time-simulation-of-physical-systems-using-simscape.html>)

A continuación se puede ver la medición del tiempo de ejecución a través de una GPIO y el osciloscopio. En este caso las tareas tardan 2.44 microsegundos en ser ejecutadas (ver Figura 18), que son la suma de los diferentes procesos de la simulación (modelo planta/control, entradas y salidas analógicas, envío de señales al GPIO...). Además, habiendo elegido un tiempo de ciclo mayor; 200 microsegundos (ver Figura 17), se obtiene un idle time de 197.56 microsegundos como margen de seguridad para procesos que no entran en las placas; como los bloques SCI que envían datos al ordenador.

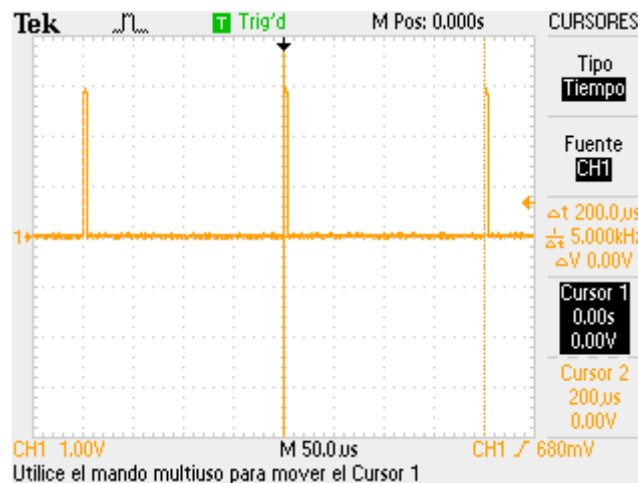


Figura 17 Repetición de ejecuciones (Chosen Step Size)

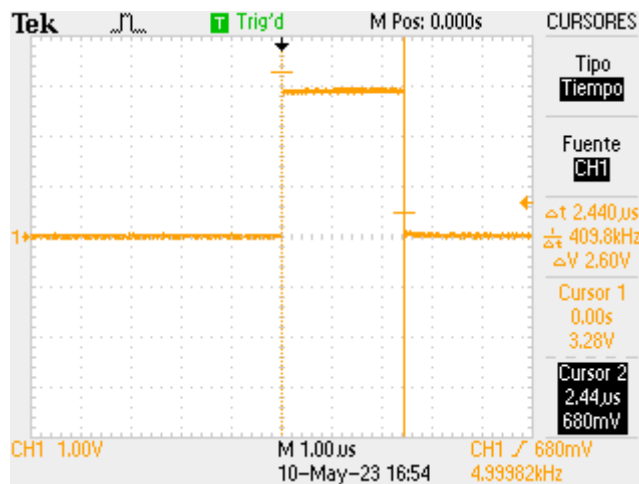


Figura 18 Tiempo de ejecución (Simulation Ejecution Time)

3.2.4 Monitorización de las variables

Para poder visualizar las variables internas de las tarjetas, se utiliza el programa HOST explicado anteriormente; que puede enviar datos con un Serial Send y recibirlos con un Serial Receive. El proceso es el siguiente: los datos enviados por el Serial Send saldrán por el SCI Receive de las simulaciones buildeadas; y así obtienen información para comenzar el ciclo. Los datos conseguidos se envían al Serial Receive del HOST mediante el SCI Transmit de los modelos.

Para que este proceso se haga correctamente es importante comprobar en qué puertos COM están las tarjetas y la cantidad de datos que se envían o reciben. Dado que el HOST se ejecuta en un ordenador y hay dos puertos configurados (uno por tarjeta), para cargar los programas primero se debe de conectar un puerto USB que conecte a la tarjeta que recibirá el control. Después de cargar el modelo de control en la placa, se desconecta el puerto USB (ya que el programa queda cargado en la placa) y se conecta la placa que llevará la planta y se carga. Para poder ver activas a las dos placas a la vez; una vez conectados los ADC/DAC y los GND físicamente, se ejecuta el programa HOST mientras los dos puertos están conectados al ordenador (la tarjeta de control tendrá un COM y el de la planta otro).

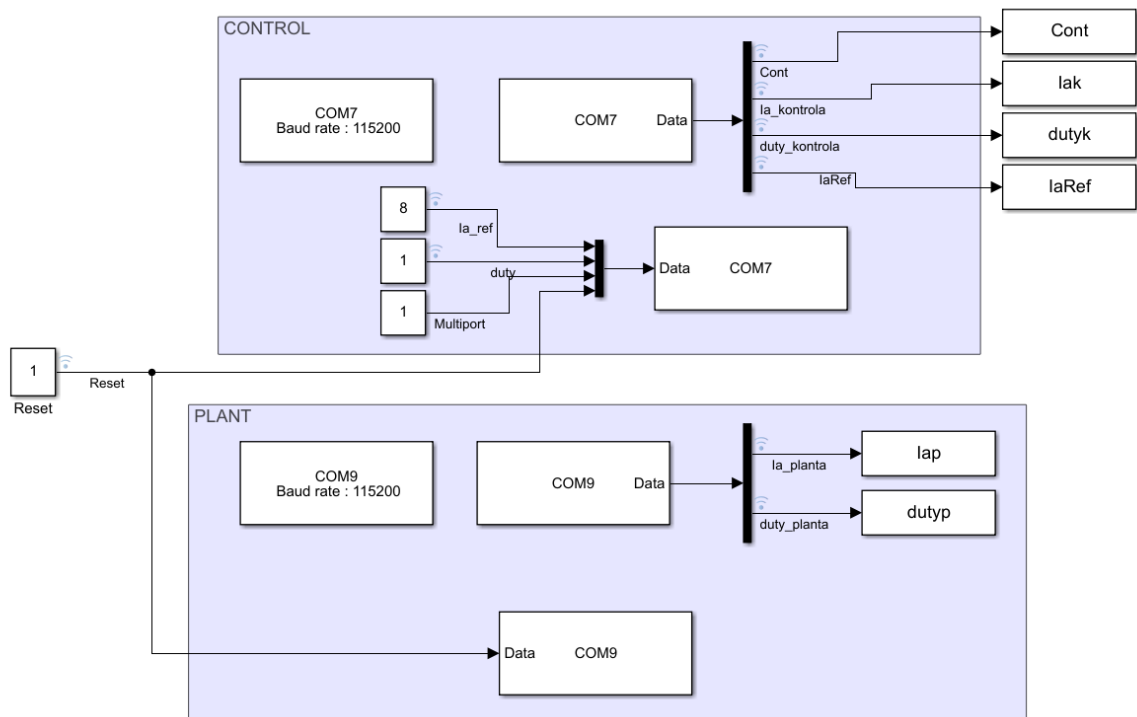


Figura 19 Estructura del HOST

Se ha comprobado que esta configuración de HOST permite monitorizar las variables de cada tarjeta y compararlas, pero como se ha explicado en la sección 3.2.1, el eje de tiempos para las variables en tiempo real no es correcto. El HOST no se ejecuta en tiempo real y las variables internas de las tarjetas representadas en el eje de tiempos del HOST muestran un comportamiento que no es real.

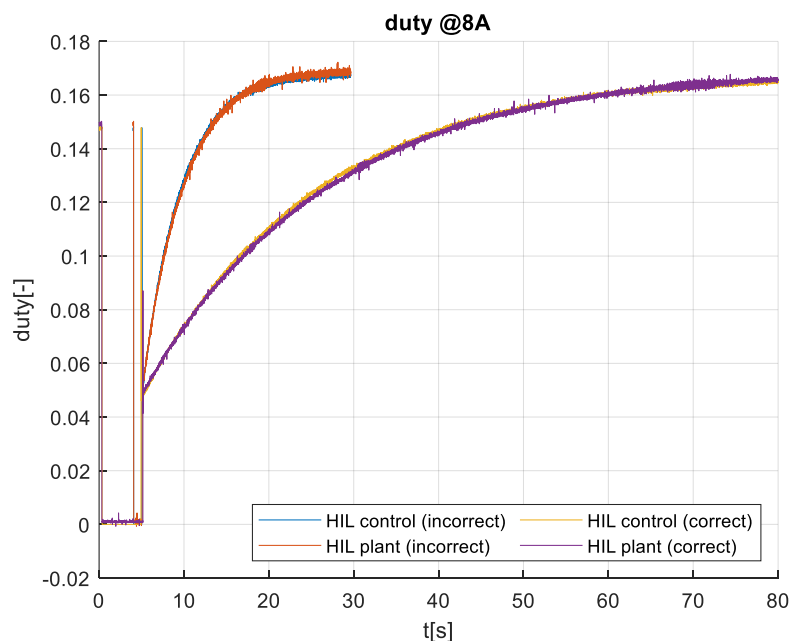


Figura 20 Ejes de tiempo diferentes para la misma variable

Para solucionar este problema se han probado alternativas como:

- El uso de Simulink Desktop Real time.
- El envío de [información en paquetes y la configuración del Simulation pacing](#).
- El modo [external de Simulink](#).
- La implementación de un contador de ciclos

En ninguno de los casos se han obtenido buenos resultados, por lo que se ha optado por implementar manualmente el contador mencionado anteriormente. Así se han monitorizado los resultados de la sección 4.

4 Verificación de la plataforma HIL

La plataforma HIL se ha verificado contra los resultados de la plataforma MIL. Se han recogido tanto la respuesta del ciclo de trabajo del osciloscopio como los datos del control y de la planta para compararlos con la simulación MIL. Comparando la subida del duty cycle en diferentes referencias de corriente, se obtienen los siguientes resultados:

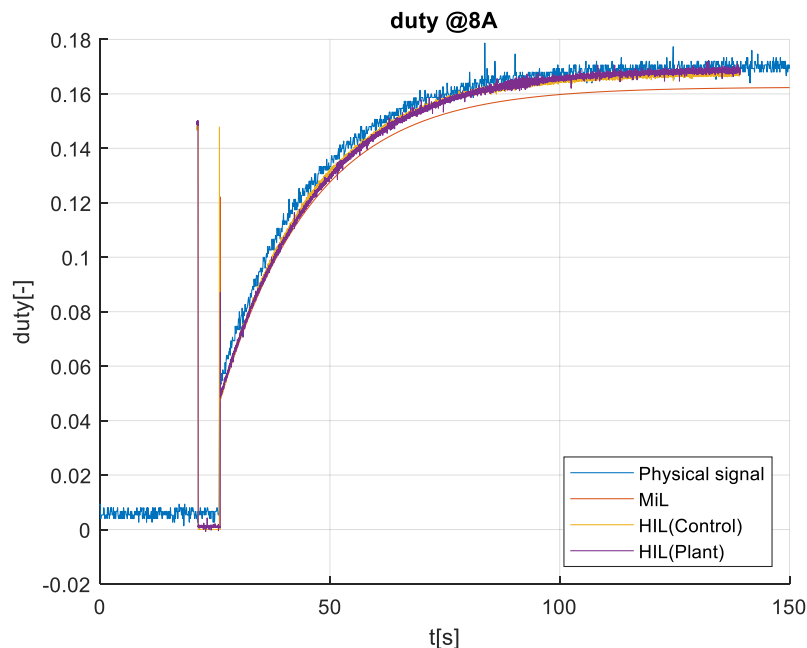


Figura 21 Resultados del Ciclo de trabajo a 8 amperes de referencia

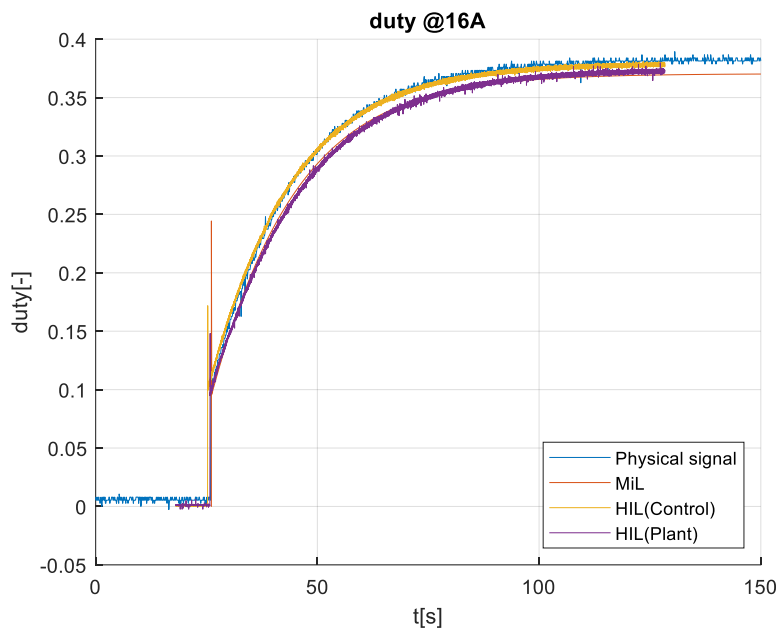


Figura 22 Resultados del Ciclo de trabajo a 16 amperes de referencia

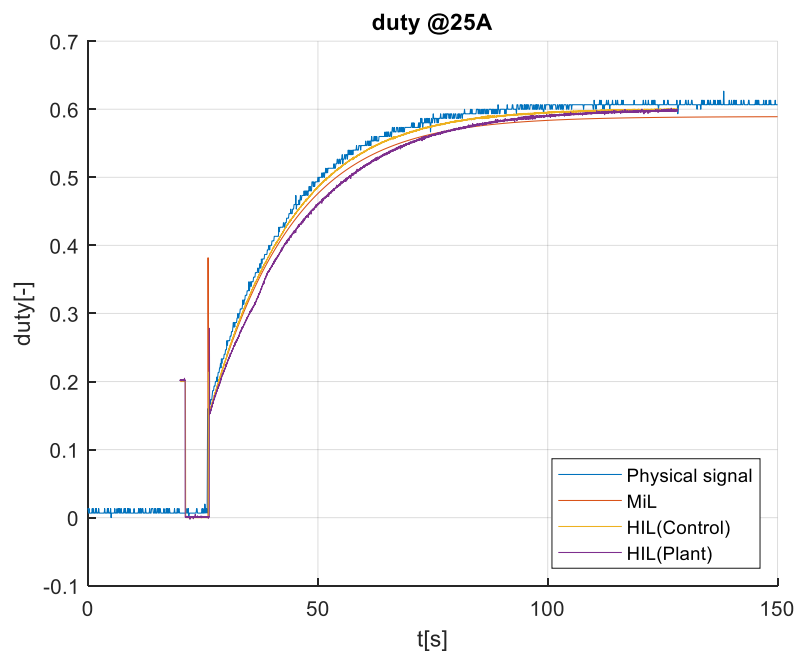
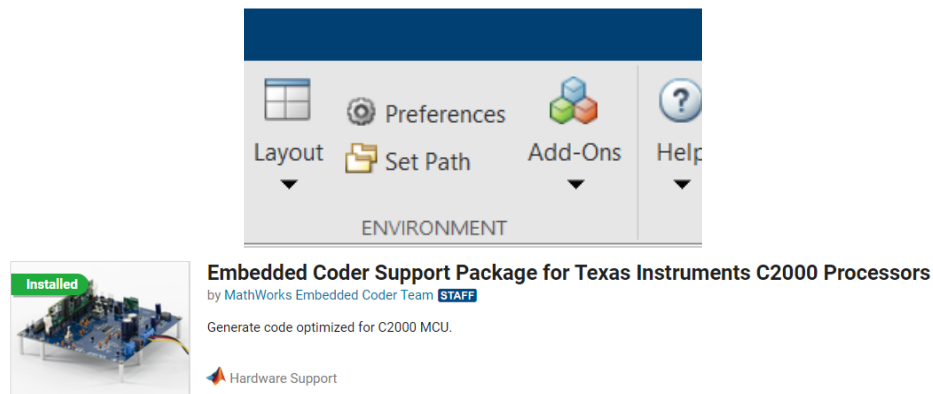


Figura 23 Resultados del Ciclo de trabajo a 25 amperes de referencia

En estos resultados se observa que el aumento exponencial es bastante parecido con un error en régimen permanente. Esto es debido a factores negativos como el offset, o el fallo de linealidad. En el régimen permanente, en general, las señales de HIL son más altas que el MiL, cosa que afectará después a otros datos como la velocidad; pero para la aplicación real son fiables ya que el error es de pocos decimales.

5 Anexo. Programación de LaunchPad F28379D mediante Matlab/Simulink

Instalar el programa necesario en Simulink: Getting Started with Embedded Coder Support Package for Texas Instruments C2000 Processors.



Al instalarlo, la aplicación pedirá que configures el programa. Para ello, hay que instalar varios programas de Texas Instruments y hacerse una cuenta. El programa mismo indica los pasos a seguir.

