

PRÁCTICA 3.

Control en tiempo real por computadora (xPCTarget). Control de movimiento -*Motion Control*- de máquinas eléctricas.

2. 1. Introducción.

Actualmente, se está viviendo un proceso de intenso cambio en el área del control industrial, protagonizado por los denominados PACs (*Programmable Automation Controllers*). Estos dispositivos ofrecen importantes ventajas sobre los clásicos PLCs, fundamentalmente en lo referido a la flexibilidad de programación. No en vano, los sistemas PACs tienen, en casi todos los casos, la estructura interna de una computadora personal, lo que conlleva aprovechar toda la capacidad de programación de esos sistemas. De hecho, pueden programarse exactamente igual que un ordenador personal. Aprovechando esto, hemos implementado un banco de pruebas que permite realizar experimentación remota, a través de redes TCP/IP (Intranet o Internet), sobre algoritmos de control de movimiento de actuadores basados en máquinas eléctricas.

En esta practica, vamos a implementar la programación de un lazo de control RT de posición. Cuando la programación incluye la generación de consignas –trayectorias- se llama también control de movimiento -*motion control*-. En esta práctica se plantea la programación del control de movimiento de un motor *DC*, que forma parte de un sistema industrial de posicionamiento.

Suponemos que nuestra motor DC forma parte de un periférico informático. En dicho contexto, el sistema completo se dedica a mover un elemento de un sitio a otro según una trayectoria predefinida (escáner, impresora, etc.,..).

En cualquier caso, para cada motor implicado en el sistema, la operación se traduce en implementar diversos posicionamientos angulares –trayectorias- que siguen una secuencia determinada, normalmente cíclica –ciclo de producción- y que debe respetar una estricta sincronización temporal –para conformar correctamente el movimiento del sistema completo-, para cada producto o lote de productos.

Finalmente, es evidente que, de la celeridad de los posicionamientos depende la duración de cada ciclo de producción y, por lo tanto, la velocidad completa del sistema. Asimismo, está claro que la exactitud de dichos posicionamientos marcará la calidad del producto final.

Es por todo ello que nos planteamos implementar un banco de pruebas que pueda reproducir, con el mayor realismo posible, el ciclo de producción de los diferentes actuadores implicados en el sistema. Después, estudiaremos el rendimiento del sistema utilizando diversos algoritmos de control, buscando mejorar sobretudo el tiempo de establecimiento y el error en permanente, de modo que pueda ampliarse –en la medida de lo posible, considerando las limitaciones del actuador- la productividad y la calidad final del producto.

2. 2. Hardware.

Utilizaremos para esta practica el mismo motor que en la práctica 2, de modo que nos remitimos al guión correspondiente para la descripción del mismo. Sin embargo, en este caso utilizaremos un amplificador de corriente para actuar sobre el motor. Haciéndolo de este modo, evitamos los problemas del puente en H con las señales de control de pequeña magnitud, por encima y debajo de cero (zona muerta de protección ante cortocircuito del puente). Dichas señales son típicas en la estabilización de los movimientos en los lazos de control.

El amplificador de corriente está implementado en una tarjeta de circuito impreso adjunta, junto con la fuente de alimentación, al dispositivo del motor. De esta manera, solo tenéis que cambiar el conector del motor desde el del puente en H a el conector incluido en dicha tarjeta. Asimismo, debéis conectar la fuente de alimentación incluida en la parte inferior. El esquemático descriptivo del amplificador puede consultarse en la figura 3.1.

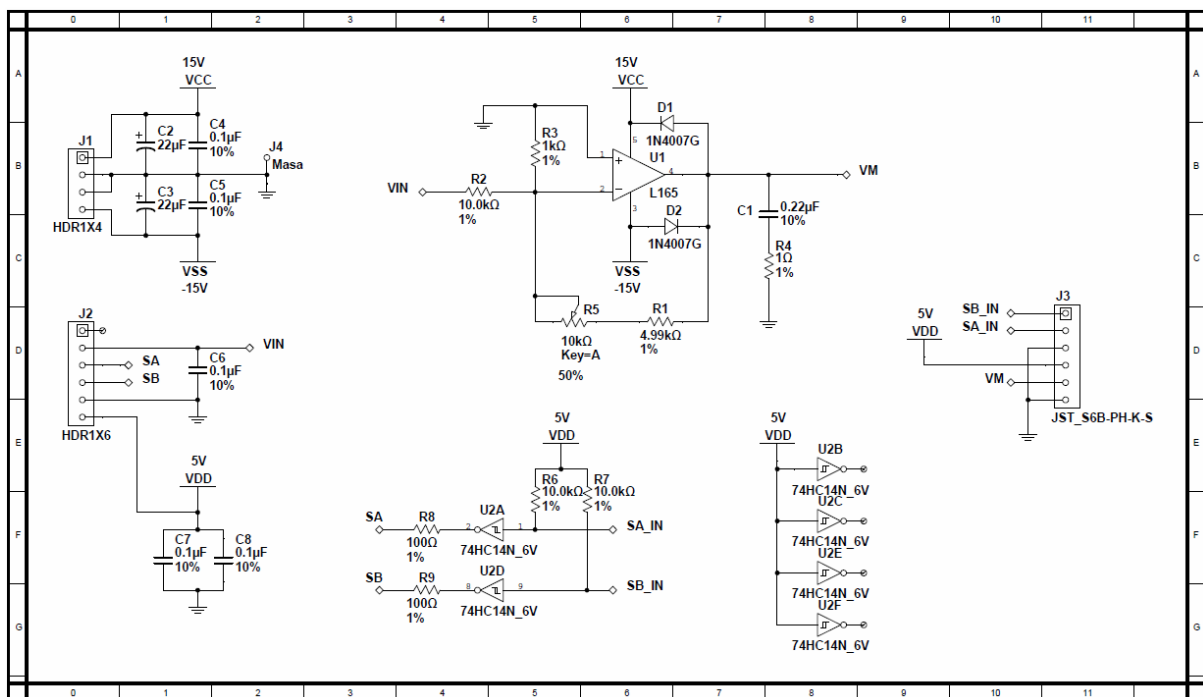


Figura 3.1. Esquemático del amplificador de corriente -con el permiso de Enrique Blanco-.

2. 3. Implementación del control RT de posición de un motor DC.

Esta práctica se plantea como una secuencia de análisis-síntesis. En la primera parte, se implementa un control de posición, proporcionado por el profesor, para su cuidadoso análisis. Posteriormente, deben proponerse alteraciones o mejoras al control mediante el uso exclusivo de los recursos de programación contenidos en el entorno *MathWorks*.

2.3.1. Implementación y análisis.

En este caso, la implementación inicial se limita a crear los ejecutables RT, a partir de los esquemas de *Simulink* proporcionados por el profesor, enviarlos al *xPCTarget* correspondiente, realizar la ejecución de los experimentos de control y recoger las señales de interés para su posterior análisis.

El lazo de posición de partida, así como la generación de la consigna de posición durante un ciclo de producción de 50 segundos se implementa en base al esquema “Control_Mov_PI_Digilent.mdl” y que está disponible en *Moodle*. Podemos ver dicho esquema en la figura 3.2.

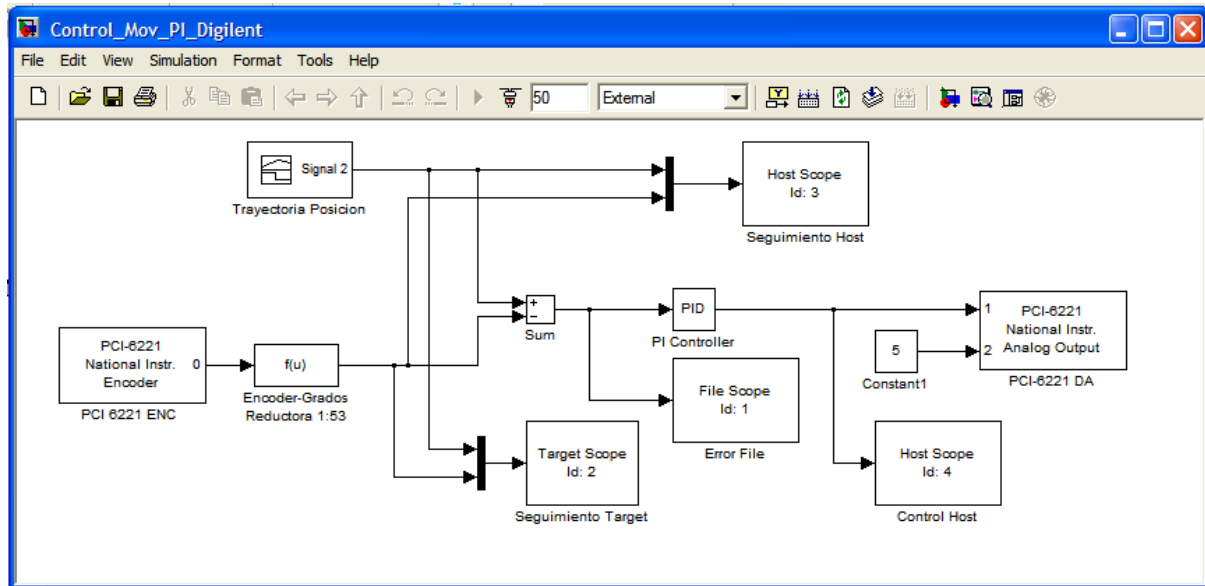


Figura 3.2. Esquema del lazo de control de movimiento.

En dicho esquema tenemos un bloque para convertir la salida del bloque “Encoder” en la posición angular en cada instante y, si es necesario, tener en cuenta la presencia de reductores mecánicos de par –utilizados a menudo para incrementar el par del actuador, “reduciendo” las vueltas que da-. Por otro lado, tenemos el bloque llamado “*Traectoria_Posición*” que es del tipo “*Signal Builder*” de *Simulink* y que nos permite generar una consigna –de posición- que describe el ciclo de producción de nuestro actuador. Podemos observar dicho ciclo en la figura 3.3., donde vemos que enviamos al motor a la posición 350 grados hasta el segundo 5, luego a la posición 0 hasta el 10,... Evidentemente, existen tipos de consignas mucho más sofisticadas que mejoran el comportamiento del actuador y que pueden implementarse también con bloques de *Simulink*.

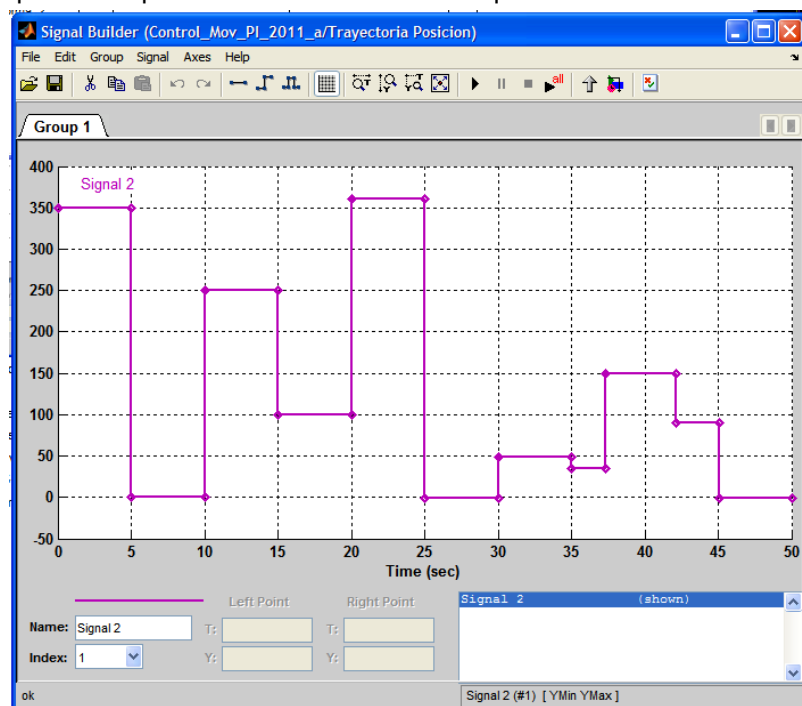


Figura 3.3. “Traectoria” del actuador durante el “ciclo de producción”.

Para poder utilizar dicho esquema, realizamos **con cuidado** las conexiones eléctricas descritas a continuación:

A. Amplificador de corriente-Tarjeta:

1. DIR (1 de J1-Marca blanca-) sin conectar.
2. EN (2 de J2) a 22 del embornado (salida analógica AO 0).
3. SA (3 de J2) a 42 del embornado (CTR 1 A).
4. SB (4 de J2) a 46 del embornado (CTR 1 B).
5. GND (5 de J2) a 54 del embornado (AO GND).
6. VCC (6 de J2) a 21 del embornado (salida analógica AO 1 fija a 5 V.).

La salida de la señal de control se realiza en este caso directamente a través de la una de las salidas analógicas de la tarjeta NI6221M, dado que ahora se alimenta el motor directamente gracias a l amplificador de corriente. Como podéis observar en la figura 3.2. sacamos 5 V. fijos por la otra salida analógica para conectar a la alimentación de los diodos del encoder (Vcc), en el caso de que haya problemas con la salida fija que proporciona la tarjeta. (bastante habitual ultimamente).

Podemos observar en la figura 3.1. que en el esquema se incluyen varios “*Scopes (xPC)*”, preparados para monitorizar y estudiar las señales de interés del sistema. Una vez compilado el código generado a partir de este esquema y enviado el ejecutable al XPCTarget “Controlador”, obtenemos la respuesta descrita en la figura 3.4 a partir del “Host *Scope*” Id. 3, que monitoriza el seguimiento de la consigna, cuando ejecutamos remotamente un primer experimento de control de movimiento-sin carga-.

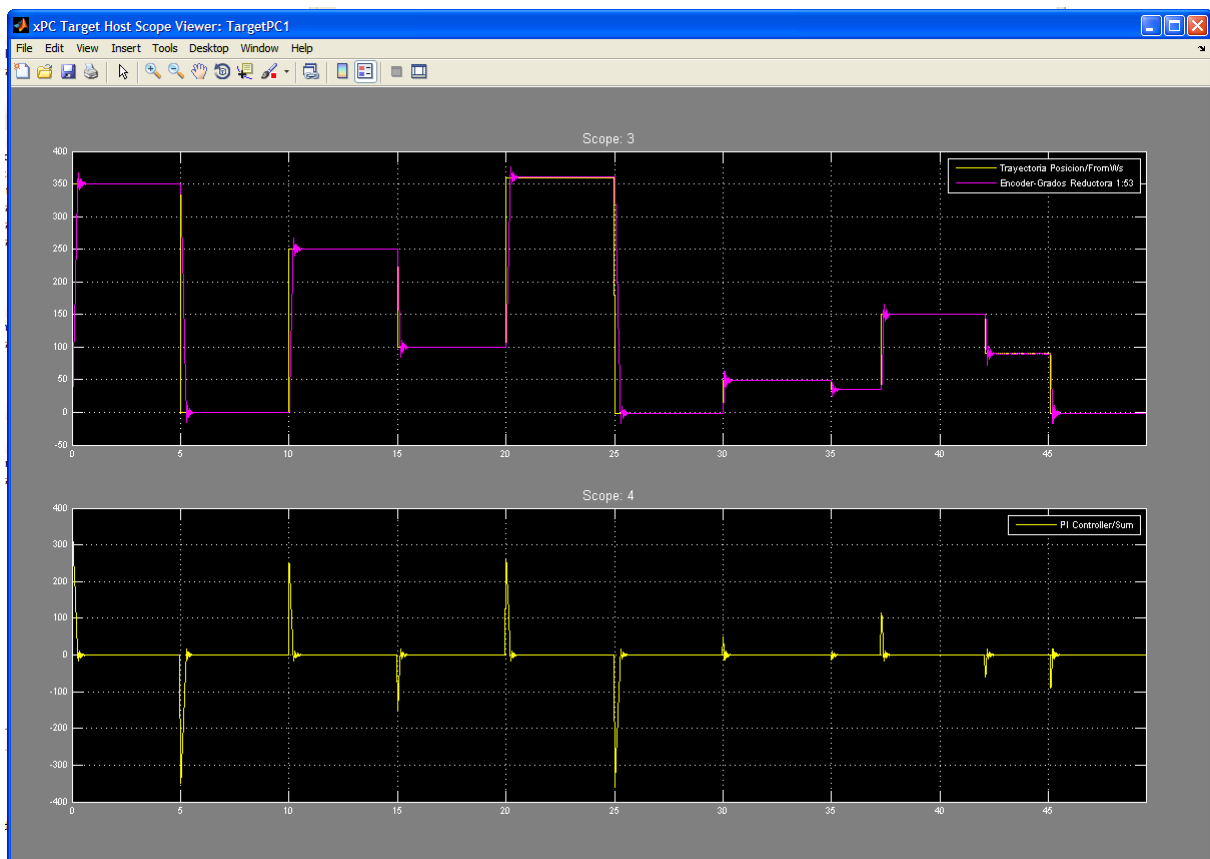


Figura 3.4. Trayectoria, señal de control y posición angular del experimento 1.

Una vez concluida la implementación de este sistema de control de partida, el objetivo es que trabajéis con él para comprender en detalle su funcionamiento en una primera etapa de análisis. Para homogeneizar los experimentos de control que llevaremos a cabo en el laboratorio, vamos a completar unos apartados comunes a todos los grupos. Son 9 experimentos en total, que podéis encontrar planteados en el documento "*GuiaExperimentosPractica3.pdf*".

Tenéis que tener en cuenta la descripción que aparece en este guión y las señales de interés que hemos monitorizado. Podéis alterar la señal de consigna –aunque es importante que todas las consignas acaben en la posición 0, para no perder la posición inicial- para someter al sistema a diversas situaciones de funcionamiento. También puede considerarse cambiar algunos detalles de la programación de los canales de entrada y salida...

Después de esta primera etapa, debéis ser capaces de:

- A. Describir con detalle el funcionamiento de las etapas de entrada y salida.
- B. Describir con detalle el planteamiento del lazo de control.
- C. Estudiar especialmente las características del transitorio (sobretudo, rebose y tiempo de establecimiento) y el error en permanente. ¿puedes medirlos exactamente?
- D. Contestar a las siguientes preguntas. ¿Cómo se hace la conversión de la reductora y para qué?, ¿el rendimiento es el mismo en cada sentido de giro?, ¿es el mismo en cada parte del ciclo de producción?

Una vez que entendemos con detalle cómo funciona el controlador planteado, debemos criticarlo: ¿es bueno el comportamiento dinámico?, ¿son limpias las señales?, ¿por qué? ¿el controlador está bien ajustado?....

Esta etapa es muy importante, ya que abre el camino a posibles mejoras de programación.

2.3.2. Propuesta de mejoras. Etapa de Síntesis.

En esta última etapa, se deben proponer mejoras a la programación de partida. No hay limitaciones, siempre que no requieran cambios en las conexiones eléctricas o en la configuración inicial del equipo.

Si hace falta cambiar las conexiones o alterar la configuración del *driver*, debe justificarse previamente al profesor y recibir su visto bueno y asesoramiento.

Para estudiar y demostrar la bondad de vuestras propuestas utilizaremos los *test* incluidos en la *Guía de experimentos* utilizada en la fase de análisis.