

Herramientas para la docencia de control digital en grados de ingeniería

Lerma, E.^a, Costa-Castelló, R.^{a,*}, Griñó, R.^a, Sanchis, C.^b

^a*Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)*
Departamento de Ingeniería de Sistemas, Automática e Informática Industrial (ESAI)
Universitat Politècnica de Catalunya (UPC).

^b*MathWorks.*

To cite this article: Lerma, E., Costa-Castelló, R., Griñó, R., Sanchis, C.. 2021. Tools for teaching digital control in engineering degrees. *Revista Iberoamericana de Automática e Informática Industrial* 18, 193-203. <https://doi.org/10.4995/riai.2020.13756>

Resumen

En este trabajo se describen un conjunto de materiales pedagógicos que han sido desarrollados para dar soporte a la docencia del control digital. En particular, se presentan un conjunto de herramientas interactivas y el proyecto *Duino-Based Learning (DBL)* pensado para desarrollar prácticas experimentales de control digital utilizando *Arduino*. Además, se describe el contenido de un curso de control digital, que se imparte en el marco de una titulación de ingeniería industrial, y como se utilizan las diferentes herramientas desarrolladas en su docencia.

Palabras clave: Implementación digital, sistemas lineales, desarrollos curriculares para la enseñanza del control en ingeniería, educación en el ámbito de control utilizando equipos de laboratorio.

Tools for teaching digital control in engineering degrees

Abstract

This work describes a set of teaching materials that have been developed to support the learning of digital control. In particular, it presents a set of interactive tools and the project *Duino-Based Learning (DBL)* that has been designed to develop experimental practices of digital control using *Arduino*. In addition, the paper depicts the contents of a digital control course, taught in the framework of an industrial engineering degree, and how the developed tools are used in its teaching.

Keywords: Digital implementation, linear systems, curricular developments for teaching control in engineering, education in the field of control using laboratory equipment.

1. Introducción

El control automático es una de las disciplinas que tradicionalmente se ha impartido en las titulaciones de ingeniería y, en particular, en la titulación de ingeniería de industrial. En el grado en Ingeniería en Tecnologías Industriales (GETI) que actualmente se imparte en la *Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)* de la *Universitat Politècnica de Catalunya (UPC)*, actualmente se imparten dos asignaturas obligatorias de este ámbito, una de ellas denominada *Dinámica de Sistemas* y la otra *Control Automático*. La primera se impar-

te en segundo curso, mientras que la segunda se imparte en el cuarto y último curso del grado. *Dinámica de Sistemas* introduce a los estudiantes en los conceptos fundamentales del control automático, el modelado, el análisis y el diseño de sistemas de control de tiempo continuo desde un punto de vista entrada-salida. *Control Automático*, objeto de este trabajo, es una clara continuación de la primera pero, esta vez, en un marco de tiempo discreto o, mejor dicho, en el ámbito de los sistemas de datos muestreados. De esta manera, se acerca a los estudiantes a la implementación pues, hoy en día, la gran mayoría de controladores se implementan digitalmente.

*Autor para correspondencia: ramon.costa@upc.edu

Este trabajo presenta las herramientas docentes que se emplean y los contenidos que actualmente se imparten en dicha asignatura. La asignatura *Control Automático* corresponde en el plan de estudios a 6 créditos ECTS. Como en la mayoría de asignaturas del ámbito de la ingeniería, la docencia se divide en sesiones de teoría-problemas y sesiones de trabajos prácticos, estas últimas con un marcado carácter experimental.

Las sesiones de teoría se imparten siguiendo un formato tradicional, donde los diferentes conceptos se imparten mediante clases magistrales utilizando el soporte de transparencias, desarrollos en la pizarra y el soporte de herramientas interactivas. Aunque en ciertos ámbitos el uso de la pizarra tradicional está algo denostado, la experiencia nos indica que este formato sigue siendo uno de los más apreciados por los estudiantes cuando se utiliza de forma ordenada y estructurada.

En la última década, en ciertos ámbitos se ha generalizado el uso de herramientas interactivas como soporte pedagógico. El control automático no ha sido una excepción a ello (Dormido et al., 2005). Con el objetivo de utilizar estas herramientas se desarrolló un libro con las notas del curso y un gran número de herramientas interactivas que pueden ser utilizadas como apoyo durante las clases o como herramientas de autoaprendizaje por parte de los estudiantes (Costa Castelló and Fossas, 2014).

La asignatura *Control Automático* se imparte cuatrimestralmente con entre 220 y 250 estudiantes en cada cuatrimestre. La realización de prácticas experimentales con este volumen de estudiantes requiere de un gran número de puntos de trabajo en el laboratorio y que el coste de cada uno de ellos no sea muy elevado. Desde hace ya casi dos décadas el laboratorio de automática de la ETSEIB está equipado con 10 servomecanismos de corriente continua de LJ Technical Systems. La Figura 1 muestra uno de ellos. Esta planta está compuesta por un motor de corriente continua junto con sus etapas de acondicionamiento de señales y potencia. El módulo contiene los elementos de medida necesarios para cerrar los lazos de control de velocidad, gracias a una dinamo taquimétrica, y de posición, gracias a un potenciómetro circular. Además de esto, el motor está equipado con un freno magnético que se usa como perturbación. Estos equipos se utilizan también para las prácticas de la asignatura *Dinámica de Sistemas*, en ella los lazos de control se cierran mediante electrónica analógica mientras que en *Control Automático* se cierran utilizando un computador.

Anteriormente, en las prácticas de *Control Automático* los estudiantes adquirían los datos usando unas tarjetas AD/DA integradas en los ordenadores del laboratorio. Estas tarjetas eran costosas y utilizaban el bus PCI, que cada vez es menos común, por lo que complican las actualizaciones de los equipos del laboratorio. En comparación, en la última década han aparecido diferentes equipos de bajo coste que pueden fácilmente adaptarse para realizar prácticas de control automático (Barber et al., 2013; Candelas et al., 2015; Sobota et al., 2013). Por ello, se tomó la decisión de cambiar los puestos de trabajo, por dispositivos basados en *Arduino* (Ruiz et al., 2019).

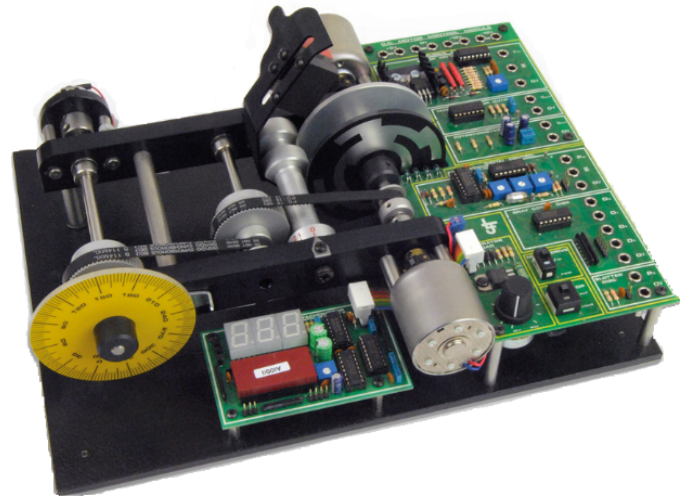


Figura 1: Servomecanismo de corriente continua, fabricado por LJ TechnicalSystems, utilizado en las sesiones de prácticas experimentales

Arduino ofrece una gran gama de modelos y, entre toda la gama, se eligió el *Arduino Due*, debido a las siguientes tres razones:

- Es uno de los pocos modelos equipado con un microcontrolador que incluye periféricos de conversión digital-analógico (DAC) reales.
- Presenta hasta 12 canales de entradas analógicas con una resolución de 12 bits, mientras que otros modelos más conocidos, como el *Arduino Uno* o el *Arduino Mega*, sólo tienen 10 bits de resolución.
- El microcontrolador del *Arduino Due* tiene un alto rendimiento que permite una mayor capacidad de cómputo con lo cual se facilita la implementación de sistemas de control digital.

Aunque las placas basadas en *Arduino* presentan su propio entorno de programación, en la actualidad es posible programar dichas placas mediante generación automática de código desde MATLAB/Simulink (Lerma et al., 2018). Ello es de gran interés para la realización de las prácticas de control automático, pues permite a los estudiantes centrarse en los conceptos de control y no en los temas vinculados a la codificación del microcontrolador. Para abordar la transición a este nuevo entorno de prácticas se definió el *Duino-based learning*. Este proyecto contó con el apoyo de *Mathworks* y ha permitido crear un conjunto de elementos *software* y *hardware* de soporte para la realización de prácticas experimentales de control automático. Basado en los desarrollos de este proyecto, se han creado los entornos de prácticas utilizados actualmente para realizar las sesiones experimentales de la asignatura *Control Automático*.

El trabajo se organiza de la siguiente manera: la sección 2 describe las características de las principales herramientas interactivas desarrolladas; la sección 3 describe el proyecto *Duino-based learning*, sus elementos y componentes; la sección 4 describe los contenidos de la asignatura *Control Automático* y como se han integrado las diferentes herramientas en la docencia; finalmente, la sección 5 presenta las conclusiones del trabajo.

2. Aplicaciones interactivas para la docencia del control automático

La interactividad ha demostrado ser una herramienta muy interesante pedagógicamente hablando y en el ámbito del control automático se han desarrollado un gran número de aplicaciones interactivas que pueden servir de apoyo a los docentes durante sus sesiones magistrales o a los estudiantes durante su proceso de aprendizaje (Díaz et al., 2019; Guzmán et al., 2016). Por otra parte, empieza a ser habitual la aparición de libros que utilizan este tipo de herramientas como soporte (Longchamp, 2006; Guzmán et al., 2012; Costa Castelló and Fossas, 2014). En este sentido, se han desarrollado diferentes herramientas que, actualmente, son utilizadas para la docencia del control digital y han sido incorporadas en un libro que puede ser utilizado para seguir los contenidos de la asignatura (Costa Castelló and Fossas, 2014).

Las herramientas interactivas han sido desarrolladas por el equipo de profesores con el apoyo de algunos estudiantes usando el *software* Sysquake (Piguet, 2018) que ha facilitado notablemente su creación y mantenimiento.

Dado que el objetivo de las aplicaciones es dar apoyo a la docencia y al proceso de autoaprendizaje, se ha optado por un diseño minimalista en que cada aplicación se centra en un único concepto y está desprovista de elementos ornamentales. Todos los elementos que aparecen en la aplicación son interactivos y, por tanto, el usuario puede interactuar con ellos para analizar las diferentes causas-efectos que se dan en el concepto analizado. La lista de aplicaciones desarrolladas, clasificadas por temática, es la siguiente:

- Muestreo y discretización de sistemas y señales.
 - Teorema del muestreo y concepto de recubrimiento espectral
 - Reconstrucción de señales.
 - Relación entre el plano s y el plano z .
 - Discretización de sistemas: transformada z , transformada bilineal y aproximación de Euler.
- Análisis de los sistemas de tiempo discreto.
 - Respuesta temporal.
 - Lugar geométrico de las raíces.
 - Criterio de Nyquist.
- Diseño de sistemas de control.
 - Controlador de avance de fase. Diseño del controlador de avance en el campo frecuencial. Diseño del controlador de avance de fase en el lugar geométrico de las raíces.
 - Controlador por retraso de fase. Diseño del controlador por retraso de fase en el campo frecuencial. Diseño del controlador de retraso de fase en el lugar geométrico de las raíces.
 - Controlador PID. Diseño de los controladores PID mediante asignación de los polos de lazo cerrado.
- Aspectos tecnológicos.

- Efecto de la saturación de la acción integral (*windup*) y su corrección.
- Conmutación entre controladores.
- Efectos de la cuantificación.

Todas las aplicaciones desarrolladas están disponibles gratuitamente en <https://sites.google.com/site/sicotedi>.

A modo de ejemplo, la Figura 2 muestra la vista principal de cuatro de las herramientas interactivas desarrolladas. La Figura 2 (a) muestra la herramienta correspondiente al teorema del muestreo. En esta herramienta se puede observar una señal sinusoidal continua y su versión muestreada en el tiempo en la figura superior y, abajo, el espectro correspondiente. El usuario puede modificar interactivamente el período de muestreo y ver el efecto del cambio de manera inmediata. El efecto del recubrimiento espectral también puede visualizarse fácilmente tanto en el dominio del tiempo como en el dominio de la frecuencia.

La Figura 2 (b) muestra la herramienta que analiza la relación entre el plano s y el plano z para un muestreo periódico por impulsos. La herramienta muestra los polos en el plano s y sus equivalentes en el plano z , pudiéndose modificar estos polos de forma interactiva. Los lugares geométricos más relevantes se muestran en ambos planos y también se muestra la respuesta temporal correspondiente a los polos analizados.

La Figura 2 (c) muestra la respuesta en el tiempo de un sistema de tiempo continuo en lazo cerrado y su versión equivalente muestreada. El efecto del período de muestreo y los cambios de parámetros se pueden visualizar con facilidad.

La Figura 2 (d) muestra el lugar geométrico de las raíces de un sistema de tiempo discreto y su respuesta temporal. Cabe destacar que tanto los polos/ceros de la planta como la ganancia proporcional pueden modificarse arbitrariamente. La herramienta también incluye sugerencias para analizar el lugar de las raíces como, por ejemplo, mostrar el centroide, los puntos singulares y las asíntotas.

3. Duino-Based Learning (DBL)

La puesta en marcha de asignaturas que utilizan el aprendizaje basado en proyectos suele requerir un esfuerzo muy elevado y suele ser complicado empezar los proyectos desde cero. Pese a ello, vale la pena, pues este tipo de aprendizaje es una excelente manera de que los estudiantes pongan en práctica todas las técnicas aprendidas en el aula y puedan ser conscientes de las diferencias entre la teoría y la práctica. Cabe decir que una forma de reducir el esfuerzo es disponer de desarrollos preliminares de base que faciliten la puesta en marcha del proyecto final. Esta problemática se nos hizo evidente cuando abordamos migrar el laboratorio docente de la ETSEIB de una plataforma basada en PC y placas AD/DA a una arquitectura de bajo coste basada en plataforma *Arduino* y fue la semilla del proyecto *Duino-Based Learning*.

Las problemáticas que se han abordado en este proyecto han sido fundamentalmente dos: la primera fue desarrollar todo el *hardware* necesario para hacer compatibles las plantas de prácticas con las plataformas *Arduino* de trabajo y, la segunda, desarrollar un conjunto de programas base a partir de los cuales

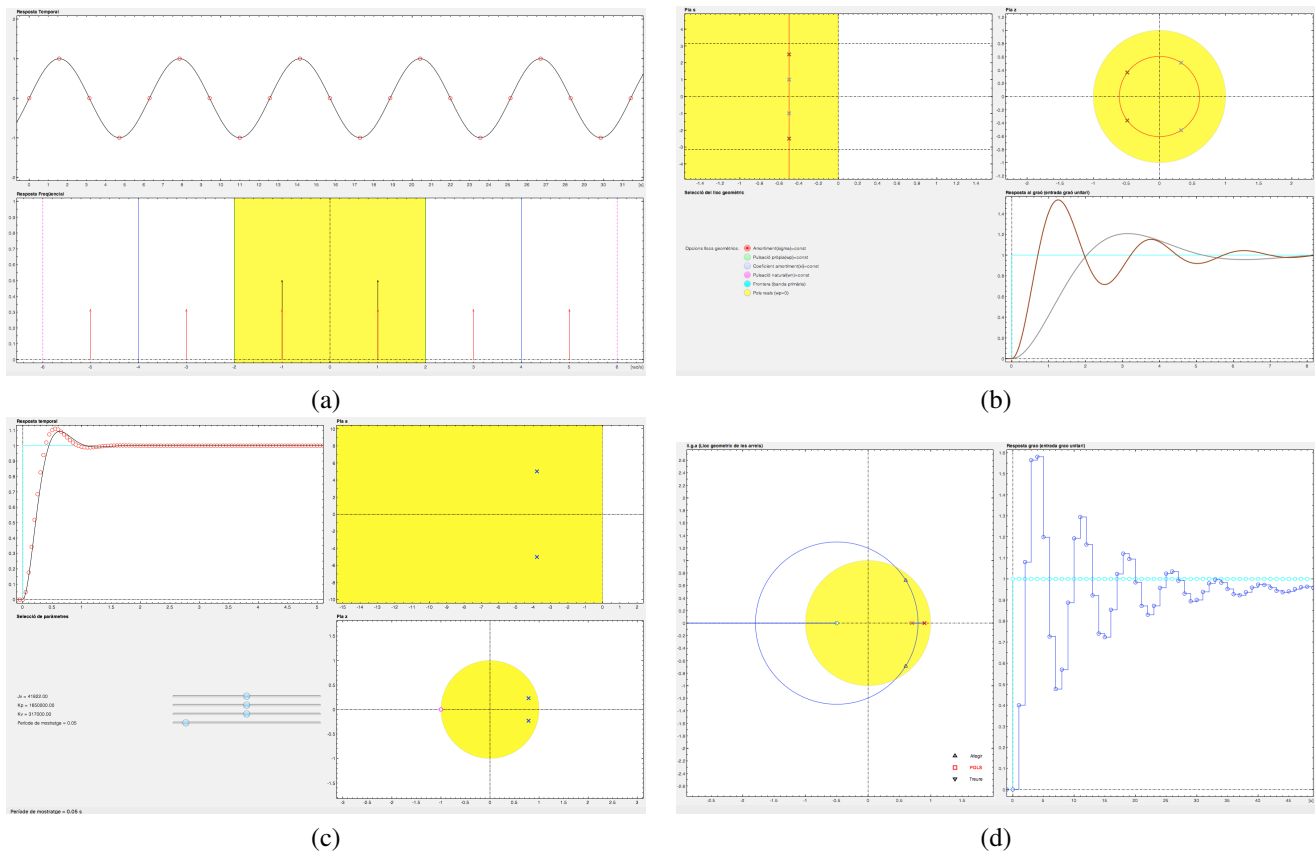


Figura 2: Vista de la pantalla principal de algunas de las aplicaciones interactivas desarrolladas para la docencia del control digital (Costa Castelló and Fossas, 2014).

fuera fácilmente desarrollables proyectos de mayor envergadura. Todo ello proporciona un conjunto de herramientas básicas para proyectos que puede ser útil para cualquier educador o estudiante.

Además de lo mencionado anteriormente, el proyecto contiene instrucciones de compilación para todas las configuraciones, secuencias de comandos en vivo de MATLAB con ejercicios (MATLAB live scripts), modelos de Simulink y vídeos tutoriales.

La Figura 3 muestra el diseño de la página web que contiene toda la información. Como puede verse, la página web está dividida en cinco secciones:

- **Duino-Based Learning (DBL):** Contiene una descripción del proyecto, sus objetivos y alcance.
- **About us:** Presenta a los autores del proyecto y ofrece mecanismos de contacto.
- **Download:** Contiene los enlaces que permiten descargar todos los ejemplos desarrollados en el proyecto y un enlace a un repositorio de GitHub que contiene todos los códigos fuente.
- **Equipment:** Este apartado explica toda la instrumentación utilizada para llevar a cabo los proyectos y cómo están los diferentes elementos conectados entre sí. La sección 3.1 contiene una descripción más detallada de estas cuestiones.
- **Projects:** En esta sección se pueden encontrar cinco proyectos que pueden usarse como semilla para el desa-

rollo de prácticas de laboratorio en el ámbito del control digital. Cada uno de ellos contiene instrucciones para construir plantas, ejercicios y vídeos tutoriales. La sección 3.3 describe estos contenidos con más detalle.

The screenshot shows the homepage of the Duino-Based Learning (DBL) website. The header includes the title 'Duino-Based Learning (DBL)' and navigation links for 'ABOUT US', 'DOWNLOAD', 'EQUIPMENT', and 'PROJECTS'. Below the header, there are language selection buttons for 'ENGLISH', 'CASTELLANO', and 'CATALÀ'. The main content area features the DBL logo and the text 'DUINO-BASED LEARNING'. A paragraph explains the project's goal: 'We believe experimentation is key in learning. Project-Based Learning is hard to implement from scratch and hence we aim to provide a starting point for any educator or learner. DBL is a project to develop freely redistributable materials to conduct educational lab projects with MATLAB, Simulink, Arduino and low-cost plants.' Below this, there is a link to the GitHub repository: 'https://mathworks.com/arduino'. A note states: 'A working copy of MATLAB is required. An Arduino board can be acquired at https://www.arduino.cc.' The 'Projects' section is partially visible, starting with 'The practices of the subject Computer Control consist in the study of a servomechanism of angular positioning (LJ Technical Systems) controlled by a PC. The laboratory sessions P1 and P2 focus on the experimental analysis of the temporal and frequency responses of the system respectively. Sessions P3 and P4 are dedicated to the design of PID controllers. Finally, in practice P5 controllers will be designed in the frequency domain.'

Figura 3: Página de inicio de DBL: <https://duinobasedlearning.github.io>.

Cabe destacar que, en el entorno DBL diseñado, el cierre del lazo de control se efectúa en el *hardware Arduino* y, por tanto, los controladores diseñados se ejecutan en su microcontrolador. De esta forma, se aseguran las prestaciones de tiempo real y la

comunicación con el entorno Simulink para visualizar los resultados se puede realizar a diferente periodo de muestreo. La generación del código ejecutable para el microcontrolador se hace automáticamente usando las herramientas de generación automática de código que incluyen Simulink y MATLAB.

3.1. Adaptación del hardware

Uno de los primeros problemas que uno se encuentra en el momento de conectar una placa *Arduino* con un dispositivo concreto es que los rangos de tensiones de los dos elementos no son los mismos. Por ello, es habitual tener que desarrollar circuitos electrónicos para compatibilizar dichos voltajes.

Con el objetivo de hacer compatibles los rangos de tensión de las E/S del *Arduino Due* y los del módulo *LJ Technical Systems*, se creó un *shield* (ver la Figura 4) encargado de acondicionar dichas tensiones. Esta placa está compuesta por diferentes circuitos viéndose los principales en la Figura 5. Dichos circuitos pueden, fácilmente, adaptarse a otros niveles de tensión seleccionando apropiadamente los componentes pasivos (resistencias y condensadores). Se podría decir que se ha creado un *shield* universal para el *Arduino Due* con el fin de adaptar rangos entre el microcontrolador y la planta a controlar. Los correspondientes esquemáticos de los circuitos pueden ser descargados directamente de la página web del proyecto.

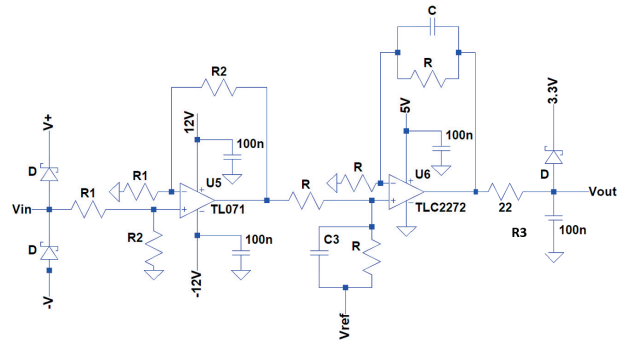
La Figura 6 muestra la estación de trabajo completa en su configuración actual. Toda la información sobre los diferentes pasos seguidos y los componentes está también disponible en la página web del proyecto.



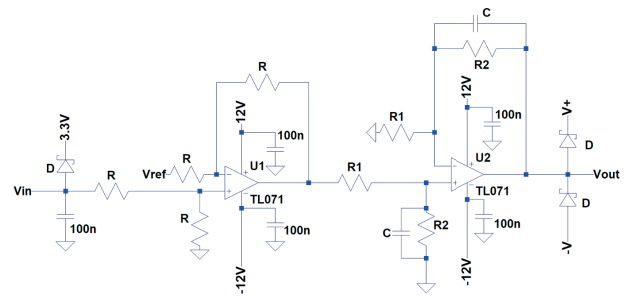
Figura 4: Arduino Due junto con la placa de acondicionamiento de señal.

3.2. Análisis de las prestaciones de tiempo real

Como paso previo a implementar los controladores es conveniente analizar las prestaciones de tiempo real que ofrece la placa de control *Arduino* y la hoja de simulación en Simulink. Este aspecto es relevante puesto que se pretenden utilizar los bloques *Scope* de Simulink como elementos de visualización de las señales relevantes (referencia, salida y señal de control) del sistema en lazo cerrado real. A tal efecto, se distinguen dos periodos de muestreo diferentes: el periodo de muestreo de visualización, T_g , de valor pequeño, para que las representaciones gráficas en los *Scopes* sean lo más fieles posible a las señales reales de tiempo continuo, y el periodo de muestreo del sistema de control digital, T_s , que será un múltiplo entero de T_g a fin de simplificar la codificación.



Acondicionamiento de la señal para la entrada del convertor A/D.



Acondicionamiento de la señal para la entrada del convertor D/A.

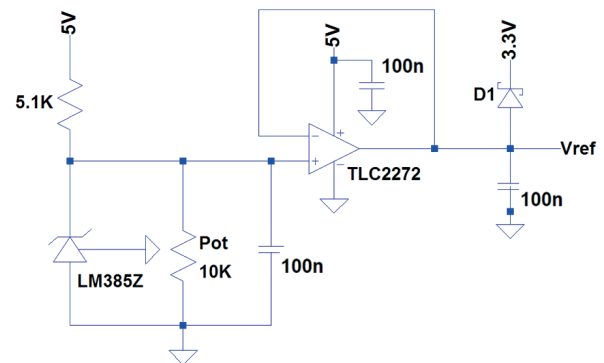


Figura 5: Circuitos acondicionadores de señal

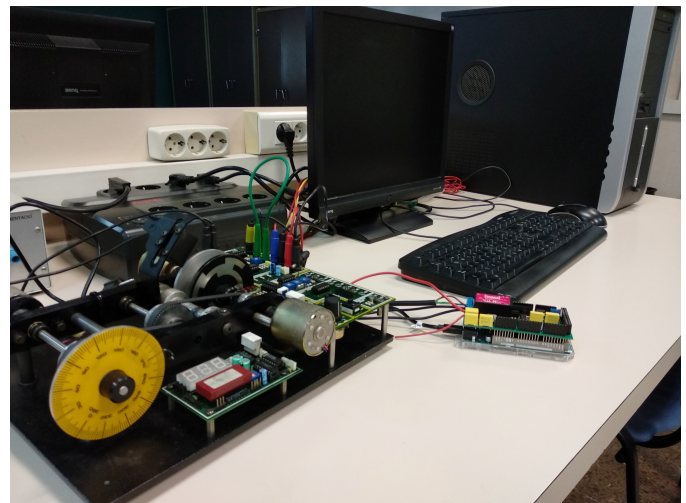


Figura 6: Una de las estaciones de trabajo del laboratorio de control automático.

Para caracterizar dichas prestaciones se ha diseñado una aplicación sencilla que genera una señal sinusoidal de 10 Hz (frecuencia fundamental mayor que las que pueden aparecer en el sistema real) y se verifica el comportamiento con distintos períodos de visualización T_g . En la Figura 7 se muestra el esquema de bloques Simulink para llevar a cabo el experimento: la señal sinusoidal sale por el convertor D/A y la salida física de este convertor se conecta con la entrada física del convertor A/D, comparándose, de forma gráfica, la lectura digital con la señal original. Se empieza con un T_g de 0,001 s y se verifica si hay pérdida de muestras en los *Scopes*. Si las hay, se va incrementado el período de visualización hasta que no se produce pérdida de información.

Como se ha mencionado anteriormente, se utiliza una placa *Arduino Due* que ofrece una resolución de 12 bits en el convertor A/D. No obstante, el bloque *analog input* de la librería Simulink solamente ofrece una resolución de 10 bits, con lo que la señal de entrada y salida tendrían distinta cuantificación y distintos valores para una misma señal original. Para que esto no ocurra, se ha cambiado la codificación interna del bloque *analog input* para poder aprovechar la resolución de 12 bits que ofrece el hardware. La modificación efectuada consiste en llamar, en el código asociado al bloque, a la función *analogReadResolution*, fijando 12 bits, antes de usar *analogRead* si la placa en uso es un *Arduino Due*. De esta forma, los convertidores A/D y D/A operan a 12 bits y la cuantificación de las señales de entrada y salida es la misma.

La Figura 8 muestra las señales obtenidas (en el *Scope* de la Figura. 7) para el caso de $T_g = 1$ ms. Como se puede observar, se producen pérdidas de muestras y, por tanto no puede garantizarse un período de visualización constante. Por otra parte, los rangos de las señales de entrada y salida no son iguales. Esto ocurre porque el rango de tensiones del convertor D/A en el uC de la placa *Arduino Due* no es de 0 V a 3.3 V sino de 0.5 V a 2.82V. Esta característica se corregirá en el diseño del sistema en lazo cerrado.

Para poder hacer un análisis cuantitativo de la variabilidad del período T_g , se ha realizado un histograma representando la distribución de intervalos de muestreo ocurridos en un tiempo de experimentación finito. La Figura 9 muestra que, aproximadamente, la mitad de los períodos corresponden a $T_g = 1$ ms pero la otra mitad se reparten entre 6 y 8 ms. En consecuencia, no se puede seleccionar el valor de $T_g = 1$ ms como período de visualización y debe incrementarse. Tras unas cuantas iteraciones se selecciona como período de visualización $T_g = 5$ ms puesto que, para este valor, la totalidad de los períodos medidos es 5 ms excepto un pequeño *jitter* como puede observarse en la Figura 10. La Figura 11 muestra las señales temporales obtenidas (en el *Scope* de la Figura 7) para el caso de $T_g = 5$ ms. En ellas puede observarse que el período de visualización se mantiene constante no perdiéndose muestras por problemas de comunicación. Por tanto, para el resto del trabajo se asume este período de visualización.

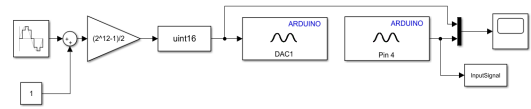


Figura 7: Diagrama de bloques de Simulink para el análisis de la variabilidad en el período de muestreo.

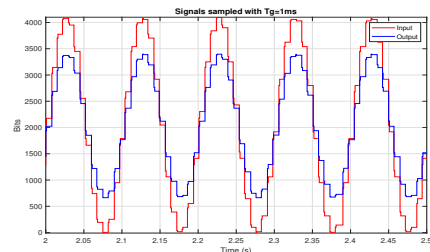


Figura 8: Señales obtenidas con $T_g = 1$ ms.

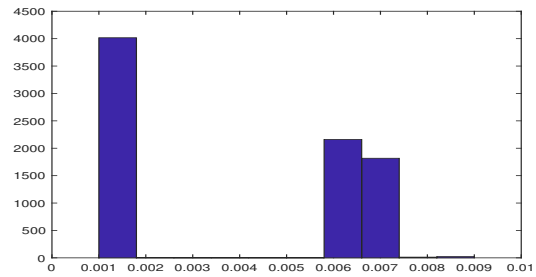


Figura 9: Variación del período de muestreo para $T_g = 1$ ms.

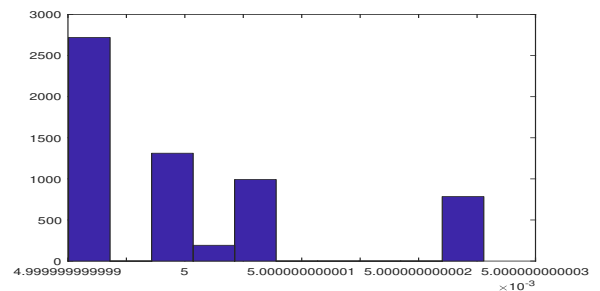


Figura 10: Variación del período de muestreo para $T_g = 5$ ms.

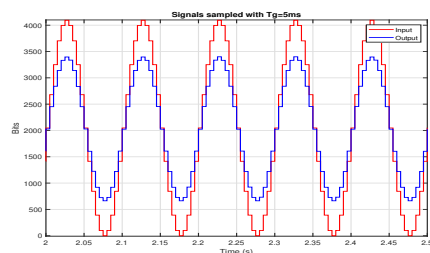


Figura 11: Señales obtenidas con $T_g = 5$ ms.

3.3. Ejemplos de uso

Como base para futuras tareas experimentales, se han desarrollado diferentes ejemplos que pueden utilizarse a modo de plantillas. Para cada uno de ellos, se ofrecen los scripts de MATLAB junto con datos, gráficos y explicaciones. Dado que el objetivo principal de estos ejemplos es mostrar la interacción con entornos físicos, se han desarrollado aplicaciones que interaccionan con los motores de corriente continua, y en particular con los servomecanismos de *LJ Technical Systems* (ver Figura 1).

Como se ha mencionado anteriormente, en la página web hay cinco ejemplos completos diferentes. Estos se han desarrollado en el laboratorio que el departamento de Ingeniería de Sistemas, Automática e Informática Industrial (ESAI) de la UPC tiene en la ETSEIB. La Figura 12 muestra el diseño de la pestaña dedicada a estos proyectos. A continuación se describen estos ejemplos.

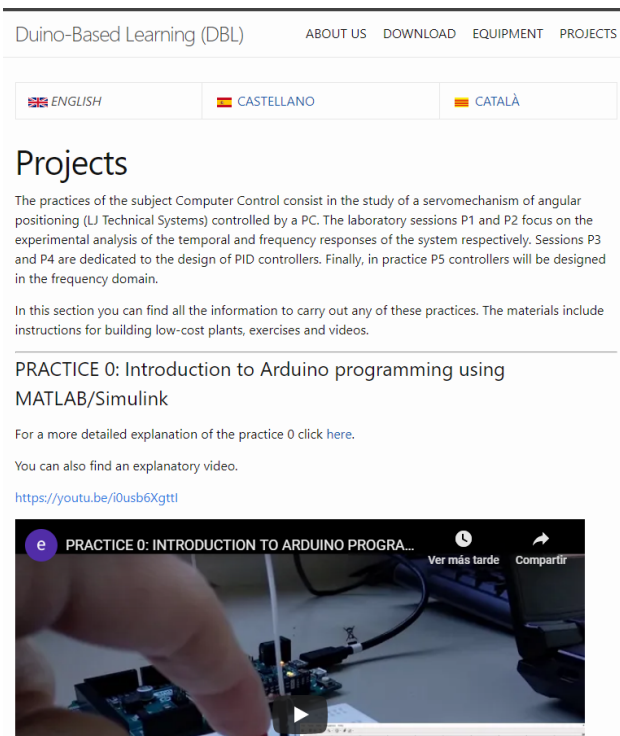


Figura 12: Sección de proyectos de Duino-Based Learning.

3.3.1. Introducción a la programación en Arduino mediante MATLAB/Simulink

El objetivo principal de este primer ejemplo es aprender a configurar Simulink para permitir la conexión y la transferencia de datos entre la placa *Arduino* y Simulink. Junto a esto, se explica el uso de los bloques de entrada y salida analógicos y digitales específicos de *Arduino*. La Figura 13 muestra el gráfico obtenido al conectar un botón al modelo de *Digital Input* de Simulink.

La documentación y el vídeo incluyen los siguientes elementos: *Digital output*, *Digital input*, *Analog output* y *Analog input* que, posteriormente, son necesarios para la realización del resto de prácticas.

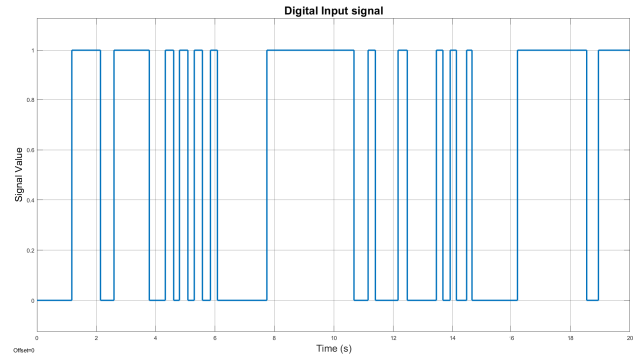


Figura 13: Señal obtenida en uno de los experimentos del primer ejemplo.

3.3.2. Análisis de la respuesta temporal de un sistema de control digital

En este segundo ejemplo, se analiza la respuesta temporal del servosistema de *LJ Technical Systems*. En primer lugar, se obtiene un modelo de comportamiento basado en el estudio de la respuesta temporal de lazo abierto. Luego se cierra un lazo de control proporcional en salida velocidad y se analiza el efecto de la ganancia proporcional en la precisión y estabilidad del sistema en lazo cerrado. También se analiza el efecto del periodo de muestreo en la estabilidad del sistema en lazo cerrado muestreando con diferentes periodos de muestreo. Se sigue la misma pauta para el control de posición. Los casos tratados son los siguientes:

1. Control de velocidad:

- Respuesta de lazo abierto.
- Respuesta de lazo cerrado.
- Evaluación experimental de las prestaciones del sistema en lazo cerrado.

2. Control de posición:

- Respuesta de lazo abierto.
- Respuesta de lazo cerrado.
- Evaluación experimental de las prestaciones del sistema en lazo cerrado.

La Figura 14 muestra la evolución temporal experimental de la velocidad del sistema en lazo cerrado de control de velocidad para diferentes valores del período de muestreo y un mismo valor de la ganancia proporcional.

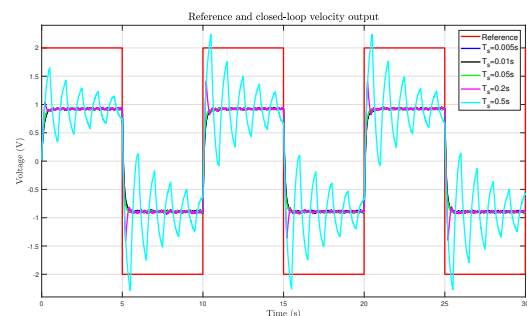


Figura 14: Efecto del período de muestreo para el caso de control de velocidad.

3.3.3. Análisis de la respuesta frecuencial de un sistema de control digital

En esta práctica se analiza la respuesta en frecuencia del servosistema *LJ Technical Systems*. Aquí se construirá un diagrama de Nyquist experimental y se comparará con el teórico (obtenido a partir de la función de transferencia ajustada en la práctica anterior). Junto con esto, se aplicará el criterio de Nyquist para determinar el rango de valores de un controlador proporcional que da lugar a un sistema estable en lazo cerrado y se verificará experimentalmente. Las actividades son:

1. Obtención del diagrama de Nyquist.
2. Obtención experimental del diagrama de Nyquist del sistema de tiempo discreto.
3. Aplicación del criterio de Nyquist.
4. Obtención del diagrama teórico de Bode.

La Figura 15 muestra la comparación entre el diagrama de Nyquist experimental y el teórico.

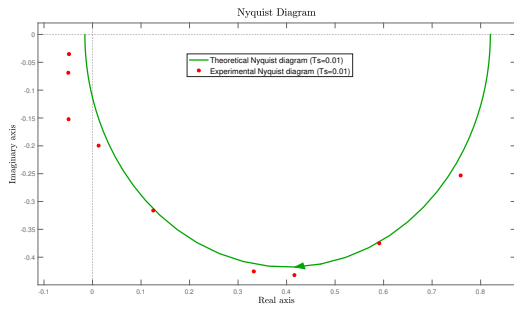


Figura 15: Comparación entre el diagrama de Nyquist ideal y el experimental.

3.3.4. Diseño e implementación de controladores PID

Este ejemplo aborda el diseño de controladores PID. En primer lugar, se diseña un controlador PI digital para el control de velocidad y, posteriormente, se diseña un PID digital para la regulación de la posición.

1. Diseño e implementación de un PI por asignación de polos para salida velocidad.
2. Diseño e implementación de un PID por asignación de polos para salida posición.

3.3.5. Mejora de los controladores PID

Como en la práctica anterior no se tienen en cuenta posibles cancelaciones de ceros y polos o posibles mejoras en la implementación, en esta práctica, se presenta un posible cambio para mejorar la dinámica del sistema en lazo cerrado. Para ello, se modifica la arquitectura actual de los controladores y se diseñan un controlador I-P para velocidad y un controlador I-PD para posición.

1. Diseño e implementación de un controlador I-P por asignación de polos para salida velocidad.
2. Diseño e implementación de un controlador I-PD por asignación de polos para salida posición.

Las Figuras 16 y 17 muestran las respuestas temporales con los controladores implementados en esta práctica en comparación con los resultados obtenidos en la práctica anterior.

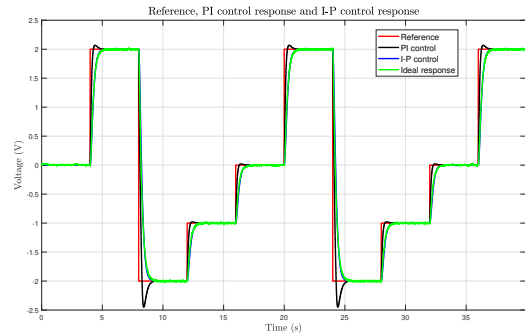


Figura 16: Comparación experimental de los sistemas de control de posición PI e I-PD.

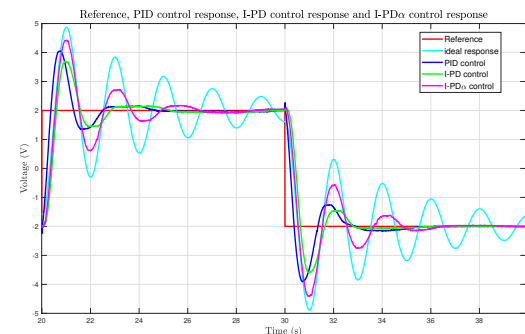


Figura 17: Comparación experimental de los sistemas de control de posición PID, I-PD e I-PD α .

3.3.6. Diseño de controladores en el dominio frecuencial

En este quinto trabajo, se efectúa el diseño de un controlador en el campo frecuencial a fin de modificar el comportamiento del sistema en lazo cerrado. En concreto, se diseña un controlador por avance de fase para que la respuesta del motor en lazo cerrado, para el control de posición, presente un error estacionario del 10% frente a una referencia rampa tal como se aprecia en la Figura 18. Las actividades son:

1. Determinación de los márgenes de ganancia y fase del sistema.
2. Diseño de un controlador de avance de fase.
3. Análisis en simulación.
4. Implementación del controlador de avance de fase.

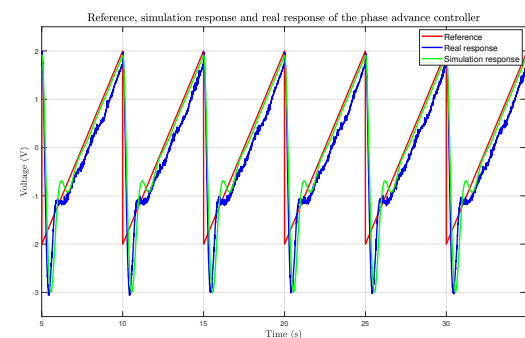


Figura 18: Respuesta del sistema a una referencia rampa con el controlador de avance de fase.

3.3.7. Control adaptativo indirecto

En este último ejemplo, se estudian e implementan dos controladores autoajustables: uno para el control de velocidad y otro para el control de posición. Los parámetros de la planta se estiman mediante el algoritmo de mínimos cuadrados recursivos de Rubio et al. (2019), el cual hace que se actualicen las ganancias del controlador I-P o I-PD dependiendo del ejercicio. La Figura 19 muestra el diagrama de bloques implementado en Simulink para el caso de control de velocidad.

1. Diseño e implementación de un PI adaptativo indirecto por asignación de polos para salida velocidad
2. Diseño e implementación de un PID adaptativo indirecto por asignación de polos para salida posición

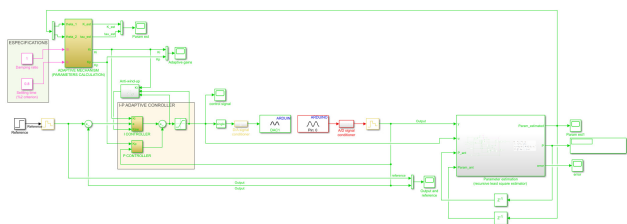


Figura 19: Modelo Simulink implementado para el controlador I-P adaptativo indirecto.

3.4. Repositorio

Todos los modelos implementados en la página web están disponibles gratuitamente en *GitHub* junto a los archivos *Gerber* utilizados para la construcción de la placa de circuito impreso, un archivo .pdf con la explicación de los pasos que se han seguido para su creación y los *Live Scripts* de MATLAB en tres idiomas (inglés, español y catalán). El objetivo de este proyecto, además de ayudar a los estudiantes a comprender los conceptos básicos del control digital, es dar facilidades a los educadores a la hora de idear otras lecciones prácticas y, en general, animar a las personas interesadas a contribuir con nuevas ideas o mejores implementaciones. De esta forma, se podría conseguir una actualización y ampliación continuas de la página web. La página web se puede encontrar en <https://duinobasedlearning.github.io/>.

4. Aplicación a una asignatura de control digital

Como se ha comentado en la introducción, en el plan de estudios del grado en “Ingeniería en Tecnologías Industriales”(GETI) de la ETS de Ingeniería Industrial de Barcelona (ETSEIB) de la Universitat Politècnica de Catalunya se incluyen dos asignaturas cuatrimestrales obligatorias de control clásico. La primera, *Dinámica de Sistemas*, se halla en segundo curso y tiene una duración de 45 horas lectivas totales introduciendo al alumno al control clásico de sistemas lineales de tiempo continuo con una sola entrada y una sola salida (SISO). La segunda, *Control Automático*, se encuentra en cuarto curso y tiene una duración de 60 horas lectivas y en ella se imparte el control clásico de sistemas lineales SISO de tiempo discreto. Estas 60 horas lectivas se reparten en 48 horas de clases de teoría-problemas y 12 horas de sesiones prácticas en el laboratorio. Concretamente, en las sesiones prácticas de esta asignatura es donde se ha introducido el entorno DBL como proyecto

de renovación de los entornos de trabajo de las plantas experimentales del laboratorio. Cabe destacar que en la asignatura *Control Automático* se matriculan entre 220 y 250 estudiantes por cuatrimestre y que, por tanto, la ocupación del laboratorio docente es notable durante todo el curso.

En las siguientes subsecciones se presentan el programa de la asignatura, la descripción de la bibliografía sugerida y la programación de las sesiones prácticas con el encaje, en ellas, del entorno DBL.

4.1. Programa de la asignatura

El objetivo de la asignatura es introducir a los estudiantes en el control de sistemas SISO lineales de datos muestreados desde un enfoque entrada-salida. Es decir, el típico curso introductorio de control digital de sistemas. Para ello, y siguiendo un enfoque clásico como el que sigue el libro básico de la asignatura (Kuo, 1995), el programa se ha estructurado en seis capítulos: un capítulo se dedica a la introducción a los sistemas de datos muestreados, otro al muestreo en el dominio temporal, tres más al análisis (herramientas matemáticas necesarias, dominio temporal y dominio frecuencial) y un último capítulo al diseño e implementación de controladores digitales. El detalle de estos capítulos se presenta a continuación.

Cap. 1: Introducción al control digital.

1. Sistemas de control digital.
2. Controladores digitales y algoritmos de control.
3. Arquitectura de un sistema de control digital.

Cap. 2: Muestreo de señales.

1. Muestreo y mantenimiento.
2. Conversores A/D y D/A..
3. Estudio matemático del muestreo.
4. Propiedades de la transformada de Laplace de una señal muestreada.
5. Teorema del muestreo.
6. Reconstrucción de señales muestreadas.

Cap. 3: Sistemas de tiempo discreto.

1. Definición de la transformada z .
2. Propiedades de la transformada z .
3. Correspondencia entre plano s y plano z .
4. Cálculo de transformadas z .
5. Cálculo de transformadas z inversas.
6. Funciones de transferencia en z .
7. Diagramas de bloques. Simplificación.
8. Sistemas en lazo cerrado. Función de transferencia de lazo abierto (L) y funciones de transferencia de lazo cerrado (S, T).

Cap. 4: Análisis en el dominio temporal.

1. Respuesta temporal de sistemas de tiempo discreto.
2. Configuración de polos en el plano z y respuesta temporal.
3. Estabilidad. Estabilidad interna.
4. Criterio de estabilidad de Jury.
5. Precisión. Error estacionario y tipo.

6. Comparación de respuestas de sistemas de tiempo continuo y tiempo discreto.
7. El lugar geométrico de las raíces (LGR). Trazado e interpretación del LGR.

Cap. 5: Análisis en el dominio de la frecuencia.

1. Respuesta frecuencial de sistemas de tiempo discreto.
2. Diagramas de Nyquist y de Bode.
3. Criterio de estabilidad de Nyquist.
4. Criterio simplificado de Bode. Márgenes de ganancia y de fase.

Cap. 6: Diseño e implementación de controladores digitales.

1. Algoritmos de control y controladores digitales.
2. Controladores PID digitales.
3. Diseño en el dominio de la frecuencia de controladores por avance y retraso de fase.
4. Diseño en el lugar geométrico de las raíces de controladores por avance y retraso de fase.
5. Diseño en el lugar geométrico de las raíces de controladores PI y PD.
6. Diseño algebraico de controladores digitales: asignación de polos y otras especificaciones, controladores de estructura general y controladores PID.
7. Programación de algoritmos de control.
8. Selección del periodo de muestreo.
9. Efectos de la cuantificación y el tiempo de cálculo.

Es importante destacar que, durante la impartición de la asignatura, se hace un énfasis importante en las técnicas de diseño digital directo en comparación con la aproximación de controladores diseñados en tiempo continuo a tiempo discreto. Por otra parte, la docencia de la asignatura se divide en sesiones de teoría-problemas y sesiones de laboratorio. En las primeras se usa un enfoque combinado: la teoría y los ejemplos y problemas se intercalan en el hilo temporal del programa sin separación en sesiones de teoría y problemas, siendo la docencia a cargo del mismo profesor.

Los estudiantes disponen en el Campus Virtual de la UPC del enlace de acceso a las Herramientas Interactivas para Sistemas de Control de Tiempo Discreto comentadas en la sección 2 a fin de poder experimentar numéricamente y afianzar conceptos cualitativos a partir de ello.

4.2. Bibliografía de la asignatura

El libro “*Digital Control Systems*”, de B. Kuo (1995), es el texto básico para el seguimiento de la asignatura por parte de los estudiantes. Este libro sigue un enfoque clásico para el control de los sistemas lineales SISO de datos muestreados. Este libro se complementa con el libro de texto (Basañez and Caminal, 2002) de problemas resueltos de control digital que incluye un número significativo de problemas de análisis y diseño provenientes de exámenes finales de ediciones anteriores de la

asignatura. Además, los estudiantes tienen acceso a la guía (Basañez et al., 2013) que describe las tareas que se han de realizar en las sesiones experimentales de laboratorio.

Los textos (Phillips et al., 2014; Ogata, 1994) se recomiendan como alternativos al libro básico (Kuo, 1995). Estos libros tienen un contenido muy parecido y siguen un enfoque similar, incluso en la notación, al del libro básico y, por ello, son también adecuados para el seguimiento de la asignatura. Para aquellos estudiantes que desean revisar, o necesitan afianzar, sus conocimientos de control clásico de tiempo continuo se recomienda, además de asistir a las horas de tutoría de los profesores, la lectura de ciertos capítulos del libro (Ogata, 2015). Es importante destacar que el libro (Åström and Wittenmark, 2011) se incluye en la bibliografía de la asignatura por que, aunque tiene un enfoque diferente del que se sigue en el curso, es de gran interés para el análisis y diseño de sistemas de control digital. Es por esta razón que, en el comentario de la bibliografía que se efectúa al inicio del curso, se menciona como texto complementario.

Finalmente, para ciertos aspectos teóricos que pueden ser de interés para determinados estudiantes, se recomiendan los textos fundamentales (Jury, 1958; Tou, 1959). En este sentido, el libro (Jury, 1973) también sirve como referencia para profundizar en la teoría asociada con la transformada z que es la herramienta matemática básica del curso.

4.3. Prácticas de la asignatura

Los trabajos prácticos de la asignatura se llevan a cabo en cuatro sesiones (L1 a L4) de tres horas cada una en el laboratorio docente departamental. La planta experimental que se emplea es un servosistema de cc de la marca *LJ Systems* (10 equipos en el laboratorio docente) y, recientemente, se ha renovado el entorno de trabajo de las plantas experimentales adoptando un subconjunto del entorno DBL. Para la realización de las prácticas los estudiantes de la asignatura se dividen en 16 grupos de entre 13 y 20 personas. Así pues, las sesiones prácticas se pueden hacer de forma cómoda con dos estudiantes por puesto de trabajo. Estas parejas y su correspondiente puesto de trabajo se mantienen durante la realización de las tres primeras sesiones prácticas¹.

El contenido de las sesiones prácticas es el siguiente:

- L1:** Estudio experimental de la respuesta temporal de un sistema digital de control de velocidad y de posición para diferentes periodos de muestreo: análisis de la precisión y la estabilidad. Esta sesión usa las prácticas desarrolladas en el entorno DBL descritas en las subsecciones 3.3.1 y 3.3.2.
- L2:** Estudio experimental y numérico de la respuesta frecuencial del sistema digital de control de velocidad. Análisis de la estabilidad mediante el criterio de Nyquist. Esta sesión usa la práctica desarrollada en el entorno DBL descrita en la subsección 3.3.3.
- L3:** Diseño de un controlador PID para el sistema digital de control de posición y verificación experimental de las prestaciones comparándolas con los resultados obtenidos

¹La cuarta sesión es el examen oral individual de prácticas.

en simulación. Esta sesión usa la práctica desarrollada en el entorno DBL descrita en la subsección 3.3.4.

- L4:** Evaluación de las sesiones prácticas. El alumno realiza un examen oral individual en el laboratorio trabajando en la planta experimental en la que debe efectuar ciertas tareas, obtener resultados y responder a las preguntas del profesor examinador. Usualmente, varios alumnos están trabajando simultáneamente en las tareas del examen, que acostumbra a durar de 20 a 25 minutos, y el profesor va pasando de uno a otro para evaluarlos.

Además, los estudiantes disponen en el Campus Virtual de la UPC del enlace de acceso al repositorio del entorno DBL donde pueden encontrar toda la información que se describe en la sección 3.

5. Conclusiones

En este trabajo se presentan las herramientas didácticas que se utilizan en un curso de sistemas de control de tiempo discreto y los contenidos, bibliografía y sesiones prácticas de dicho curso. Todo el material está disponible en línea y puede ser libremente utilizado.

Las herramientas diseñadas para el trabajo experimental emplean MATLAB/Simulink para interactuar con las plantas experimentales. De este modo, se facilita a los estudiantes la comprensión de los sistemas de control de tiempo discreto y como realizar su control puesto que, al no aparecer aspectos de bajo nivel de programación de microcontroladores, se pueden centrar en los conceptos propios de la teoría de control. Además, los experimentos propuestos están basados en un *hardware* de bajo coste *Arduino* y, por tanto, pueden alentar a las personas interesadas a entrar y progresar en el mundo del control digital.

Agradecimientos

El trabajo de E. Lerma ha sido financiado por la donación de Mathworks UPC-I-01523.

El trabajo de R. Griñó ha sido parcialmente financiado por el Gobierno de España a través de la *Agencia Estatal de Investigación* Proyecto DPI2017-85404-P, la *Generalitat de Catalunya* a través del Proyecto 2017 SGR 872 y la donación de Mathworks UPC-I-01523.

El trabajo de R. Costa-Castelló ha sido parcialmente financiado por el Gobierno de España a través de la *Agencia Estatal de Investigación* Proyecto DOVELAR ref. RTI2018-096001-B-C32

(MCIU/AEI/FEDER, UE) y “María de Maeztu Seal of Excellence to IRI” (MDM-2016-0656), la *Generalitat de Catalunya* a través del Proyecto 2017 SGR 482 y la donación de Mathworks UPC-I-01523.

Referencias

- Åström, K., Wittenmark, B., 2011. *Computer-Controlled Systems: Theory and Design*. Dover Pubs.
- Barber, R., Horra, M., Crespo, J., 2013. Practices using Simulink with Arduino as low cost hardware. *IFAC Proceedings* 46, 250 – 255.
- Basañez, L., Caminal, P., 2002. *Control digital: problemas*. Edicions UPC.
- Basañez, L., Costa, R., Fossas, E., Mañanas, M.A., Puig, V., Riera, J., Rosell, J., Villá, R., 2013. *Treballs pràctics de control digital (en catalán)*. CPDA.
- Candelas, F., García, G., Puente, S., Pomares, J., Jara, C., Pérez, J., Mira, D., Torres, F., 2015. Experiences on using arduino for laboratory experiments of automatic control and robotics. *IFAC-PapersOnLine* 48, 105 – 110.
- Costa Castelló, R., Fossas, E., 2014. *Sistemas de Control en Temps Discret (en catalán)*. Edicions UPC.
- Díaz, J.M., Costa-Castelló, R., Dormido, S., 2019. Closed-loop shaping linear control system design: An interactive teaching/learning approach. *IEEE Control Systems Magazine* 39, 58–74.
- Dormido, S., Dormido-Canto, S., Dormido, R., Sánchez, J., Duro, N., 2005. The role of interactivity in control learning. *Int. J. Eng. Educ.* 21, 1122–1133.
- Guzmán, J.L., Costa Castelló, R., Berenguel, M., Dormido, S., 2012. *Control automático con herramientas interactivas*. Pearson.
- Guzmán, J.L., Costa-Castelló, R., Dormido, S., Berenguel, M., 2016. An interactivity-based methodology to support control education: How to teach and learn using simple interactive tools. *IEEE Control Systems* 36, 63–76.
- Jury, E., 1958. *Sampled-data Control Systems*. John Wiley and Sons.
- Jury, E., 1973. *Theory and Application of the z-Transform Method*. Krieger Pub. Co.
- Kuo, B., 1995. *Digital Control Systems*. Oxford University Press.
- Lerma, E., Griñó, R., Costa-Castelló, R., Sanchis, C., 2018. Implementación de controladores en Arduino mediante Simulink, in: *Actas de las XXXVIII Jornadas de Automática*, Badajoz. pp. 158–164.
- Longchamp, R., 2006. *Comande Numériques de Systèmes Dynamiques. Cours d'Automatique*. Presses Polytechniques et Universitaires Romandes.
- Ogata, K., 1994. *Discrete-time Control Systems*. Prentice Hall.
- Ogata, K., 2015. *Modern Control Engineering*. Pearson India.
- Phillips, C., Nagle, T., Brickley, J., Chakraborty, A., 2014. *Digital Control System Analysis & Design*. Pearson Education Ltd.
- Piguet, Y., 2018. *Sysquake Pro User Manual*. Calerga. Lausanne, Switzerland.
- Rubio, J., García, E., Aquino, G., Aguilar-Ibáñez, C., Pacheco, J., Meda-Campana, J.A., 2019. Mínimos cuadrados recursivos para un manipulador que aprende por demostración. *Revista Iberoamericana de Automática e Informática industrial* 16, 147–158.
- Ruiz, A.P., Rivas, M.A., Harbour, M.G., 2019. Aplicaciones ADA en android con requisitos de tiempo real. *Revista Iberoamericana de Automática e Informática industrial* 16, 264–272.
- Sobota, J., Pisl, R., Balda, P., Schlegel, M., 2013. Raspberry Pi and Arduino boards in control education. *IFAC Proceedings* 46, 7 – 12.
- Tou, J., 1959. *Digital and Sampled-data Control Systems*. McGraw-Hill Inc.