

## Sistema Avanzado de Prototipado Rápido para Control en la Educación en Ingeniería para grupos Multidisciplinares

Antonio Flores Caballero\*, Dorin Sabin Copaci, Álvaro Villoslada Peciña, Dolores Blanco Rojas, Luis Moreno Lorente

*Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, Avda. Universidad nº 30, 28911, Leganés, Madrid, España.*

### Resumen

Con el objetivo de alcanzar resultados satisfactorios en la enseñanza y puesta en práctica de cursos semestrales de ingeniería de control, en los que la presencia de alumnos de distintas disciplinas es más que notoria, se torna necesario recurrir a altos niveles de abstracción en la programación de los sistemas de control. Este alto nivel de abstracción procede del uso de un sistema de prototipado rápido para control de carácter avanzado, que permite recurrir a funcionalidades que no habían sido previstas en ningún entorno de prototipado rápido para control disponible con anterioridad. El carácter avanzado del sistema brinda soluciones desde el más alto nivel de abstracción, el denominado diseño basado en modelos, para las intrincadas relaciones necesarias entre la ingeniería de control y la informática en tiempo real, permitiendo que los alumnos puedan centrar su esfuerzo en el desarrollo del algoritmo de control, la identificación de sistemas y el modelado de plantas físicas en lugar de preocuparse por las tediosas tareas de gestión y configuración a bajo nivel de la arquitectura hardware que están empleando. Gracias a este alto nivel de abstracción, que cubre el espectro abarcado por funcionalidades de muy bajo nivel y funcionalidades de muy alto nivel, el manejo del sistema propuesto se encuentra al alcance de audiencias multidisciplinares. El sistema avanzado de prototipado rápido para control se está empleando para cursos semestrales así como en multitud de Tesis de Máster y Doctorales.

### Palabras Clave:

Educación en control, Herramientas docentes y laboratorios en automática, CAD/CAM y CIM.

### 1. Introducción

Los sistemas de automatización junto con los dispositivos electrónicos de carácter médico y para control del automóvil son las áreas que presentan la mayor utilización de sistemas controladores embebidos (EE Times Group, 2011), (EE Times Group, 2013). Este continuo crecimiento en la demanda de dispositivos embebidos de control provoca que sean cada vez más potentes en términos computacionales y versátiles en sus capacidades de entrada/salida (I/O) y también más sencillos de operar. A pesar de estos avances en la tecnología, enseñar teoría de control a audiencias multidisciplinares sigue representando un gran reto. Debido al origen y a las características curriculares de este tipo de alumnos, que pueden presentar conocimientos diferentes. Alumnos que cursan ingeniería mecánica o eléctrica, bioingeniería, matemática, física... que generalmente carecen del conocimiento y destrezas en electrónica y programación en lenguajes textuales como C, necesarios para acometer la tarea de diseñar e implementar un sistema de control por vías

tradicionales. A esto hay que añadirle el corto espacio de tiempo disponible para la realización de estos cursos de ingeniería de control. Son estos los motivos que llevaron a buscar una solución eficiente al problema de enseñar un amplio contenido en teoría de control e implementación en tiempo real. La solución es emplear una metodología de trabajo diferente, denominada diseño basado en modelos (Mosterman et al., 2004) (MBD, del inglés *Model Based Design*), aplicándola a un sistema de prototipado rápido para control (RCP), entendiendo éstos como el conjunto indisoluble de software y hardware, lenguaje de programación gráfico y el controlador electrónico. MBD es una técnica de programación basada en gráficos en la que todos los asuntos relativos a la configuración y planteamientos de bajo nivel, que representan una gran carga de trabajo, son obviados. El lenguaje gráfico puede utilizarse para modelar matemáticamente la planta a controlar, llevar a cabo simulaciones así como poder diseñar el controlador y desplegarlo en el hardware de control a través de la generación automática de código fuente y la auto compilación del mismo; de forma que pueda monitorizarse y controlarse la planta física objeto de estudio mediante diferentes técnicas de la teoría de control, tales como control empleando tiempo discreto o haciendo uso de una aproxima-

\* Autor en correspondencia.

Correo electrónico: [afcaball@ing.uc3m.es](mailto:afcaball@ing.uc3m.es) (Antonio Flores Caballero)

ción mejorada al tiempo continuo y otras características avanzadas que serán descritas más adelante, que constituyen la base software de un sistema avanzado de prototipado rápido para control (ARCP, del inglés *advanced rapid control prototyping*). En este artículo se ha empleado el soporte para microcontroladores de 32 bits de muy bajo coste y altas prestaciones como base hardware del sistema ARCP. Aunque la herramienta software descrita en este artículo soporta otras arquitecturas tales como Raspberry Pi 2 y PlayStation 2. Respecto al software de diseño basado en modelos se ha empleado Matlab/Simulink®. Este artículo está referido a lenguajes de programación basados en gráficos cuya intención es la de programar sistemas de control y/o describir modelos matemáticos, existen más pero sus intenciones son diferentes (Gordon, 2009).

## 2. Objetivos

La finalidad del sistema presentado en este artículo es la de enseñar de la forma más eficiente la teoría de control sobre plantas reales a alumnos que se corresponden con una audiencia multidisciplinar. Se pretende que dicha audiencia pueda manejar de forma sencilla, a nivel de programación, conceptos propios de los sistemas de control bajo el paradigma de tiempo real, como son la ejecución multitarea, ejecución en tiempo discreto y aproximado a tiempo continuo y la ejecución asíncrona mediante interrupciones, entre otros.

Persiguiendo este propósito se ha diseñado un sistema RCP personalizado de carácter avanzado, que permita no sólo poner en práctica la teoría de control, sino también el plantear y organizar la ejecución del algoritmo de control en un entorno multitarea empleando el mismo soporte del lenguaje gráfico con que se programa el algoritmo. Este sistema ARCP provee a los alumnos de las herramientas hardware y software vitales para reducir en gran medida los tiempos de implementación, debido a que permite obviar la configuración de los periféricos de I/O a bajo nivel y depurar un programa orientado a un sistema embebido. Con esta avanzada herramienta los alumnos tienen la oportunidad de profundizar en su conocimiento de la teoría de control, eliminando la necesidad de programar en base a texto. Gracias a estos beneficios, resulta posible plantear complejos algoritmos de control y probarlos en plantas reales, así como organizar intrincados procesos de ejecución multitarea cuando así se requiera. Objetivos que pueden ser acometidos en tiempos reducidos gracias a disponer del sistema ARCP.

El empleo de un sistema RCP personalizado también persigue otros fines, no directamente relacionados con el objetivo de la enseñanza. En el mercado se encuentran disponibles sistemas RCP. Son sistemas caros de adquirir y de mantener, ligados a licencias de renovación anual. Con el paso de los años, el soporte del lenguaje gráfico para el sistema RCP acaba siendo desechado por el fabricante, en favor de algún nuevo RCP, perdiendo de esta forma todo el valor invertido en la adquisición del antiguo RCP. Estos problemas, pueden no serlo para las infraestructuras educativas de algunas naciones, en cambio en otras se tornan en problemas de vital importancia.

Los sistemas RCP comerciales presentan otros inconvenientes además del económico. Son sistemas diseñados para brindar

soporte sólo para funcionalidades de carácter muy básico, permiten gestionar los periféricos de I/O del hardware de control y crear una única tarea en tiempo real empleando una ejecución en tiempo discreto, sin posibilidad de utilizar una aproximación al tiempo continuo o paso de ejecución variable.

De igual manera, los sistemas RCP comerciales soportados por Matlab®, obvian la importancia que tiene comprender el funcionamiento y sincronización de las interfaces de comunicación. Hacen que un mismo modelo de programación gráfica realice las funciones de controlador en tiempo real y de programa de monitorización de datos, impidiendo que el alumno pueda discernir qué programa se está ejecutando en el sistema RCP y qué programa se está ejecutando en su ordenador.

Estudiando los antecedentes al sistema ARCP expuesto en el presente artículo, encontramos la utilización de sistemas RCP como medios para la educación en ingeniería de control que hacen uso de Matlab/Simulink® (Behrens et al., 2010; Kim, 2011; Duma et al., 2012; Wahyudi et al., 2005; Chen et al., 2010; Luis M. et al., 2005) o bien hacen uso de LabVIEW® (Gómez de Gabriel et al., 2011). Pero todos ellos presentan las mismas limitaciones, ya expuestas recientemente en este mismo apartado, limitaciones que pueden resumirse recurriendo a la cita de forma literal ya expuesta en el análisis de sistemas RCP y tarjetas controladoras de bajo coste realizado en (Valera et al., 2014). Artículo que refleja la limitaciones de la práctica totalidad de sistemas RCP disponibles; el soporte del lenguaje gráfico *“resulta muy limitado tanto en el acceso a bajo nivel del hardware como en la versatilidad de la programación”*, empleando palabras textuales del artículo citado. Generalmente el lenguaje gráfico tan sólo otorga acceso a funcionalidades muy básicas, que no son capaces de cubrir la demanda en lo que respecta a versatilidad de programación, que surge al trabajar en una aplicación de control de carácter complejo, en el que se suele terminar recurriendo a añadir manualmente programación textual en lenguaje C, tanto para aportar características de bajo y de muy alto nivel, ya sean generalistas o específicas, que no suelen estar disponibles en el lenguaje en base a gráficos. Debido a lo anterior muchos autores suelen descartar tempranamente el lenguaje de programación basado en gráficos, recurriendo a diversos lenguajes de programación basados en texto, perdiendo las ventajas en ahorro de tiempo, esfuerzo y capacidad de abstracción al programar en pro de la versatilidad.

El sistema ARCP propuesto en este artículo elimina completamente todos estos defectos y limitaciones. Por medio de nuevos mecanismos, que modifican internamente a Matlab®, implantados en el proceso de generación de código fuente, que permiten otorgar soporte basado en lenguaje gráfico a las funcionalidades que hacen al sistema tan versátil como programarlo en lenguaje C. (Flores-Caballero, 2014). En cambio LabView® no permite añadir soporte para hardware personalizado que exija modificar o cambiar completamente los mecanismos internos con los que funciona el entorno MBD.

## 3. Controladores seleccionados para el ARCP

En la actualidad los dispositivos de control más demandados consisten en microcontroladores (MCU) con unidad central

de proceso (CPU) de 32 bits, que cuentan con una gran variedad de periféricos I/O, posibilitando de esta manera que la integración de los sistemas resulte más sencilla y cuente con el menor número de componentes añadidos (EE Times Group, 2013). A menor número de componentes mayor fiabilidad presenta el sistema electrónico de control. Aproximadamente la mitad de los proyectos de control demandan velocidades del reloj de la CPU entre 1 y 99 MHz (EE Times Group, 2013), el 26 % demandan una velocidad del reloj por encima de 100 MHz y una pequeña cuantía de los mismos exige más de 1 GHz. Basándose en estas demandas, el sistema ARCP ha sido diseñado en torno a MCU de 32 bits. Concretamente la familia STM32F40x (siendo la 'x' un número del 1 al 7) del fabricante ST Microelectronics®, cuya placa de evaluación cuesta tan sólo 15 euros y el MCU NRF51x22 (siendo 'x' unos de los número 4,8 o 9) del fabricante Nordic Semiconductor®. El MCU STM32F4 cuenta con una CPU de 32 bits a 168 MHz y unidad de cálculo en coma flotante de precisión simple, multitud de interfaces I/O, contando con la potencia computacional necesaria para desarrollar complejos algoritmos de control. El MCU NRF51x22 cuenta con una CPU también de 32 bits corriendo a 16 MHz. Las interfaces de I/O presentes en el MCU STM32F4 se muestran en la tabla 1. Mientras que el MCU NRF51x22 provee de menor variedad de interfaces I/O, pero incluye una interfaz inalámbrica basada en Bluetooth 4.0, por un coste montado en placa de 6 euros. El sistema ARCP aquí presentado cuenta actualmente con soporte para arquitecturas más potentes a nivel computacional tales como Raspberry Pi 2, PlayStation 2 y el MCU STM32F7; pero no serán objeto de estudio en este artículo.

Tabla 1: Lista de periféricos I/O disponibles en el MCU STM32F4

| Periféricos Entrada/Salida                      |
|-------------------------------------------------|
| 1 x Universal Serial Bus 2.0 (USB)              |
| 2 x Controller Area Network 2.0B (CAN)          |
| 5 x Asynchronous Receiver/Transmitter (UART)    |
| 16 x 12 Bits Analog to Digital (A/D)            |
| 2 x 12 Bits Digital to Analog (DAC)             |
| 31 x Pulse Width Modulation Outputs (PWM)       |
| 12 x Pulse Width Modulation Inputs (PWM-IC)     |
| 3 x Serial Peripheral Interface (SPI)           |
| 2 x Interintegrated Circuit Communication (I2C) |
| 40 x General Purpose Digital I/O (GPIO)         |
| 14 x General Purpose 16 Bits Timers             |

También fueron objeto de estudio los controladores basados en puertas lógicas reconfigurables, *Field Programmable Gate Arrays*, cuyas siglas FPGA representan el nombre comúnmente utilizado. Estos dispositivos ofrecen altas cotas de rendimiento a costa de sacrificar periféricos de I/O, facilidad de uso y flexibilidad, esto último debido a la gran especialización necesaria para poder implementar en ellos un controlador eficiente (MathWorks, 2012), (Vouzis et al., 2009).

Recientemente, National Instruments® ha lanzado al mercado un sistema RCP orientado a su uso como herramienta educativa, denominado MyRIO®, pero en este sistema persisten las limitaciones y desventajas explicadas en el apartado anterior. Resulta costoso de adquirir y mantener; y su soporte en lenguaje gráfico es limitado, no resulta posible añadir funcionalidades personalizadas o más avanzadas.

#### 4. Creando soporte hardware para Simulink®

El sistema ARPC consta de un componente hardware y de un componente software. El software es el soporte para poder programar el hardware haciendo uso del lenguaje gráfico de Matlab/Simulink®. La figura 1 muestra el aspecto del entorno de trabajo de Simulink®. Este entorno de trabajo completamente visual es capaz de generar todo el código fuente textual que equivale, en cuanto a comportamiento, al modelo de programación descrito mediante gráficos.

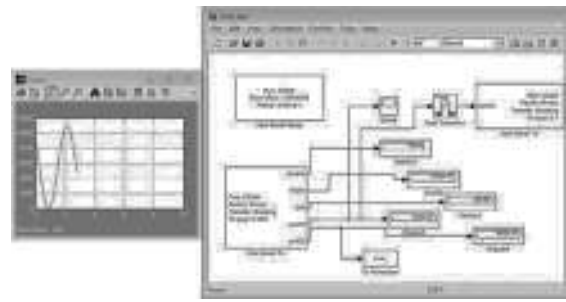


Figura 1: Entorno de programación basado en gráficos de SIMULINK®

La elección de Matlab/Simulink® como entorno de diseño basado en modelos (MBD) responde a los siguientes criterios:

Permite añadir soporte personalizado para cualquier hardware deseado, así como modificar, complementar o reemplazar los mecanismos internos de generación de código fuente. También el entorno (MBD) de Simulink® provee de las herramientas, denominadas toolboxes, necesarias para simular sistemas eléctricos, mecánicos, hidráulicos/neumáticos y otras más muy útiles en el trabajo propio de la ingeniería de control. Las citadas toolboxes se tornan muy interesantes cuando se emplean escenarios de simulación en los que interviene el hardware controlador, denominándose a este tipo de simulaciones como pruebas con el hardware en el bucle, conocidas por su denominación inglesa 'Hardware in the Loop' (HIL). En cambio el entorno MBD de LabVIEW® carece de muchas de estas toolboxes de modelado y simulación, o incluso de algunas otras que pudieran resultar equivalentes.

La descripción completa, acerca de cómo crear soporte hardware personalizado y que modifique/reemplace los mecanismos de generación de código por otros más acordes a las necesidades del sistema, se encuentra muy alejada del alcance y objetivos de este artículo. No obstante, se anotará el camino correcto a seguir de una manera breve, pero concisa:

En primer lugar, es necesario un conocimiento profundo de la arquitectura hardware a integrar en Simulink®. En segundo lugar se deben conocer cómo funcionan los mecanismos internos

tanto de Simulink® como de las herramientas de generación automática de código asociadas. Tercero, hacen falta destrezas magistrales de programación en lenguaje C. A continuación, cada nueva funcionalidad que desee integrarse como un nuevo bloque de programación gráfica de Simulink® necesitará una descripción de su comportamiento de I/O programada con la interfaz de programación de aplicaciones (API) de Matlab®. Cada versión de Matlab®, tanto bianual como de 32 o 64 bits presenta diferencias en esta API que han de ser tenidas en cuenta. Una vez definido el comportamiento de las I/O del bloque, se han de programar las opciones de configuración de su máscara desplegable, en lenguaje Matlab®. La detección de errores de configuración y selección de opciones disponibles en el bloque gráfico de soporte hardware se programa con la API de lenguaje C anterior. La información y aspecto visual que presenta el bloque gráfico se programa en lenguaje Matlab®. Y por último el código fuente generado por el bloque gráfico está controlado por el lenguaje denominado "Target Language Compiler"(TLC), que orquesta la generación automática del código fuente en base a las opciones disponibles en los bloques y a cómo se han conectado éstos entre sí.

*Dar vida a una nueva toolbox de soporte hardware, con funcionalidades avanzadas o diferentes de las comunes incluidas en Matlab/Simulink®, no es tarea al alcance de cualquiera, dados los conocimientos necesarios. En cambio, utilizar la toolbox para diseñar complejos controladores se encuentra al alcance de un público multidisciplinar, pues hacer uso del sistema ARCP no exige conocimiento alguno acerca de cómo programar a bajo nivel la arquitectura o sobre cómo funciona Matlab® en su fuero interno.*

En la Universidad Carlos III de Madrid, en el RoboticsLab, en el transcurso de la citada tesis doctoral (Flores-Caballero, 2014), se han creado varias toolboxes de programación gráfica de funcionalidades muy avanzadas, que aportan una versatilidad y robustez en la programación equiparable a la que se obtendría utilizando el lenguaje C. Gracias a la toolbox sólo se ha de recurrir a Simulink® para organizar complejos sistemas de control en tiempo real con capacidad multitarea con interrupciones y demás cualidades del hardware. Las toolboxes del sistema ARCP reciben el nombre de Beyond Control Blockset.

#### 4.1. Beyond Control blockset, origen y funcionalidades

Este blockset nació en torno al año 2011, debido a la necesidad de disponer de una herramienta de programación para controladores de alta eficiencia y gran versatilidad y robustez (similar a lo que sería posible empleando el lenguaje C) que permitiera fusionar en una sola etapa el prototipado en laboratorio y la implementación del controlador final; la eficiencia durante la programación es facilitada por el lenguaje gráfico y la versatilidad y robustez es facilitada por las funcionalidades avanzadas, funcionalidades y características que no se encuentran disponibles en ningún otro blockset de origen comercial o gratuito. Originalmente Beyond Control se concibió como un paquete de ampliación para el blockset *Waijung*, blockset de funcionalidades básicas comercial que cuenta con licencia de

evaluación gratuita para usos no comerciales (Aimagin®, Mayo 2015). *Waijung*, cuyo nombre es la transcripción fonética de 'muy deprisa' en Tailandés, es un blockset que aporta funcionalidades básicas, permite manejar los periféricos de I/O y crear una sola tarea en tiempo real. Pero la necesidad de mejorar la versatilidad en la programación, corregir las imperfecciones de *Waijung*, mejorar el soporte DMA, añadir un nuevo soporte multitarea y poder usar el tiempo libre entre tareas, así como mejorar la aproximación al tiempo continuo y permitir que los mecanismos de generación de código de Matlab/Simulink® pudiesen utilizar 3 áreas diferentes de memoria RAM (algo muy útil en el MCU STM32F4 para optimizar el rendimiento) en lugar de una sola; ha hecho que en el año 2015, en el blockset Beyond Control tan sólo permanezca de *Waijung* el aspecto visual de los bloques gráficos y las máscaras de configuración (sólo para STM32F4 y NRF51, el soporte para Raspberry Pi 2, PlayStation 2 y STM32F7 prescinde completamente de *Waijung*), habiendo cambiado completamente el código fuente que generan, de forma que ya no existan más incompatibilidades entre periféricos o problemas de prioridad entre interrupciones y dicho código sea acorde para dar cabida a las funcionalidades avanzadas. Durante la instalación de Beyond Control también se amplían las funcionalidades de Matlab/Simulink® y se instala todo lo necesario para que la toolbox funcione. Como compiladores de código fuente emplea GNU-ARM (instalado junto con la herramienta) o el comercial Keil Uvision® 5.

Cierto es, que además de los sistemas RCP disponibles en el mercado (tales como XPC Target®, TurnKey® o el soporte de Raspberry Pi® de The Mathworks® o los propios de dSPACE®, por citar algunos), otros investigadores han creado soporte personalizado para diferentes arquitecturas. D. Hercog diseñó su propio soporte para un MCU de 16 bits pensado para control de motores en usos educativos (Hercog et al., 2006). R.Duma presentó una toolbox para una familia de MCU de 32 bits de Renesas® (Duma et al., 2010), mientras que F. Luo (Luo and Huang, 2010) y R.Bartosinski (Bartosinski et al., 2007) hicieron lo mismo para un MCU de 32 bits de Freescale®. C.Rusu se decantó por el MCU TMS320F24x de Texas Instruments® (Rusu et al., 2007). Todas las citadas toolboxes de programación gráfica presentan la limitación de otorgar funcionalidades básicas (manejo de periféricos I/O y creación de una sola tarea en tiempo real), que no permiten disponer de la versatilidad necesaria para llevar a buen término el diseño y desarrollo de muchos sistemas de control. Las funcionalidades avanzadas que hacen versátil la programación con el sistema ARCP se describen a continuación:

#### 4.2. Optimizaciones

Los mecanismos de generación de código fuente han sido modificados para adecuar y optimizar los tiempos de ejecución, de forma que se detecte automáticamente el uso de funciones de procesamiento digital de señal (DSP) y se ubiquen en los espacios de memoria más adecuados en el microcontrolador STM32F4, maximizando la eficiencia del sistema. El MCU STM32F4 posee un área de memoria de 64 KB ultra-rápida; la toolbox Beyond Control ubica el código fuente para que todos los cálculos

en coma flotante así como los cambios de contexto del procesador se realicen utilizando este espacio de memoria. Así mismo también es posible hacer que todos los tipos de datos flotantes de precisión doble, sean reescritos de forma automática como precisión simple. Para el MCU NRF51x22, al no disponer de unidad de cálculo en coma flotante, los tipos de datos son convertidos automáticamente a punto fijo con la precisión más adecuada a los cálculos que se realizan, garantizando un compromiso entre precisión y rendimiento en este MCU.

#### 4.3. Aproximación mejorada al tiempo continuo

Matlab/Simulink® provee de un soporte inicial para generación de código fuente con soporte de tiempo continuo, mediante integradores numéricos. La imprecisión de este soporte radica en que la fuente de alimentación, respecto al transcurso del tiempo, para los bloques en tiempo continuo es provista por la misma base temporal que regula las tareas en tiempo real. Para mejorar la precisión de los integradores numéricos que obtienen el valor de los estados en tiempo continuo, Beyond Control provee de un mecanismo temporal con resolución de microsegundos a dichos integradores numéricos, por lo que cada bloque en tiempo continuo es alimentado con una base temporal mucho más precisa que la provista por defecto en Matlab/Simulink®. De esta manera, se consigue una mejor aproximación al tiempo continuo cuando se prescinde de tareas en tiempo real, de esta forma el programa de control se ejecuta a la máxima frecuencia y los bloques en tiempo continuo conocen con más exactitud el transcurso del tiempo. Aplicando estos nuevos mecanismos sobre una planta que evolucione lentamente (por ejemplo un motor hasta 2000 rpm) junto con un modelo de control en tiempo continuo, se podrá disponer de una buena aproximación al tiempo continuo aún empleando un controlador discreto.

Una tarea ejecutándose con esta aproximación al tiempo continuo tiende a consumir el 100 % de la capacidad computacional disponible, siendo de mucha utilidad al ponerla en práctica durante las lecciones impartidas en clase, como comparación entre éstas y las tareas en tiempo discreto, abriendo la puerta a la oportunidad de iniciar una discusión didáctica en el aula, respecto de las situaciones en las que resulta conveniente emplear tiempo continuo y aquellas en las que no presenta utilidad.

#### 4.4. Multitarea sin sistema operativo

Una característica muy avanzada del sistema ARCP, es la de exprimir los recursos hardware de cada controlador soportado, de forma que se disponga de capacidad de multitarea prioritaria en tiempo real, con capacidad de gestionar dichas tareas, sin necesidad de recurrir a un sistema operativo en tiempo real (RTOS). En el MCU STM32F4 son posibles 16 tareas en tiempo real, y en el MCU NRF51x22 son posibles 4 tareas en tiempo real. En Raspberry Pi 2 son posibles 4 tareas, una por núcleo de la CPU y en PlayStation 2 es posible ejecutar 3 tareas en perfecto paralelismo debido a su arquitectura multiprocesador. El soporte multinúcleo y multiprocesador es un añadido a Matlab/Simulink® que no está presente por defecto.

#### 4.5. Soporte para ejecución con paso variable

Esta es una característica del sistema ARCP, que no ha tenido presencia en ningún otro sistema RCP anterior.

La ejecución de una tarea en tiempo real con paso de ejecución variable permite que la frecuencia a la que se ejecuta dicha tarea se ajuste a las necesidades del control. Permitiendo al didacta explicar en el aula de prácticas la diferencia que presenta en cuanto a control un mismo sistema que una vez evoluciona lentamente y otras en cambio, evoluciona muy deprisa. Comparando la respuesta del controlador funcionando en tiempo real con paso de ejecución fijo frente a la misma respuesta funcionando bajo paso variable, el alumno alcanzará conclusiones por métodos inductivos de deducción.

#### 4.6. Uso del tiempo libre entre tareas

A menudo, en un sistema de control, resulta necesario llevar a cabo operaciones matemáticas de transformación de datos, como transformadas rápidas de Fourier u otras, cuyo tiempo de cómputo es difícil de acotar. Es debido a esto que este tipo de operaciones es preferible llevarlas a cabo aprovechando el tiempo libre que ha quedado al ejecutar todas las tareas en tiempo real, asegurándose de esta manera que las tareas de tiempo real no se vean entorpecidas. El uso del tiempo libre es una característica poco común entre los sistemas RCP disponibles con anterioridad, puede encontrarse también en los sistemas soportados por la toolbox XPC Target® de Matlab®.

#### 4.7. Comunicaciones USB® nativas de alto rendimiento

Para el sistema ARCP, en concreto el MCU STM32F4, se ha diseñado e implementado cuidadosamente una interfaz de comunicaciones que emplea el periférico USB® nativo en una nueva modalidad que rompe la típica barrera de los 64 bytes por paquete de datos; ahora, el sistema ARCP puede enviar y recibir paquetes de hasta 16 KB sin recurrir a interrupciones ni a un uso intensivo de la CPU, empleando un controlador USB que funciona tan sólo con acceso directo a memoria (DMA), eliminando el uso del procesador central en la recepción y transmisión de datos, garantizando un ancho de banda de 12 Mbits/s y tiempos completamente deterministas para la transmisión/recepción.

#### 4.8. Uso de los timers - Determinar carga de la CPU

El sistema ARCP permite medir los tiempos de ejecución de un algoritmo completo, de una tarea en tiempo real, de cada iteración de la tarea en tiempo continuo, del tiempo utilizado o sin utilizar en el uso del tiempo libre, de una parte específica del programa... se pone a disposición del usuario todas las características disponibles en los timers de forma muy sencilla. Con precisión de microsegundos.

#### 4.9. Uso completo de las interrupciones

El comportamiento de las interrupciones se define desde el entorno de programación gráfico de Simulink®; tales como fuentes de interrupción externa (EXTI), interrupciones de los timers, interrupciones de las entradas de ancho de pulso modulado (PWM), interrupciones provocadas por los manejadores de

errores y por el perro guardián del sistema. Utilizando el sistema ARPC puede definirse que ha de ejecutarse en caso de error grave del sistema, situación que provoca una interrupción de los manejadores de error; se cuenta con opciones predefinidas como resetear el sistema o devolver al estado inicial todos los periféricos de I/O, o incluso, se permite que el usuario programe lo que desee para estos casos mediante bloques de Simulink® y gestionar completamente la prioridad de ejecución de las interrupciones.

#### 4.10. Sincronización de sistemas de control distribuidos

En el desarrollo de Beyond Control se ha cubierto la necesidad de trabajar con sistemas de control distribuidos, la herramienta permite sincronizar en el tiempo las tareas de tiempo real de distintos controladores, mediante un sencillo bloque gráfico se puede elegir la interfaz de sincronización, e incluso se pueden sincronizar controladores diferentes entre sí. Es una funcionalidad disponible en todas las arquitecturas soportadas.

#### 4.11. Soporte multimedia

Como capacidades, más alejadas de la teoría de control y la implementación de sistemas computacionales en tiempo real; la toolbox del ARPC permite reproducir sonidos en el mezclador software y mostrar una interfaz de usuario en una pantalla LCD en color, todo diseñado expresamente para el MCU STM32F4. Ambas funcionalidades multimedia están implementadas de forma que no perjudiquen el desempeño de las tareas en tiempo real. Son funcionalidades que se emplean en la realización de Tesis de Máster y Tesis Doctorales en la creación de sistemas de control completamente autónomos, que no son monitorizados a través del puerto USB de un ordenador. Ambas funcionalidades también se programan de forma visual mediante Simulink®.

### 5. Dispositivos disponibles en los cursos de control

Para comprender las posibilidades del sistema ARCP como medio educativo en el aula, es conveniente conocer el tipo de sensores, actuadores y plantas físicas disponibles en la escuela.

Gracias a la difusión de la tecnología Arduino, que ha conseguido una mayor aceptación de los sistemas de control embebidos en general, y la rápida evolución y abaratamiento de la microelectrónica de estos últimos años, el mercado provee actualmente de una amplia variedad de sensores de bajo coste y excelentes prestaciones. La tabla 2 muestra una descripción de los sensores disponibles en los laboratorios y sus interfaces de I/O. Aún así, prácticamente cualquier sensor o sistema de actuación tiene cabida de ser conectado al sistema ARCP dadas sus excelentes capacidades de integración.

Los actuadores disponibles cubren desde sistemas lineales como motores de corriente continua, paso a paso y sin escobillas de tres fases, hasta actuadores no lineales como aleaciones con memoria de forma (SMA, del inglés "Shape Memory Alloy") y motores basados en ultrasonidos.

Para fortalecer los conocimientos adquiridos sobre teoría de control, es conveniente poder evaluar distintos controladores

Tabla 2: Lista de sensores disponibles para los cursos de control

| Tipo de Sensor                | Interfaz              |
|-------------------------------|-----------------------|
| Sensor de posición flexible   | Analógico             |
| Sensor posición lineal hall   | I2C, Analógico        |
| Encoder absoluto efecto hall  | SPI, PWM IC           |
| Encoder relativo efecto hall  | Señales en Cuadratura |
| Encoder relativo óptico       | Señales en Cuadratura |
| Encoder absoluto óptico       | SPI (Niveles RS-422)  |
| Giróscopo, Acelerómetro       | I2C                   |
| Barómetro, Brújula            | I2C                   |
| Temperatura, Humedad          | Analógico             |
| Sensor de fuerza              | SPI                   |
| GPS Receiver                  | UART                  |
| NRF51x22, Datos Inalámbrico   | UART                  |
| Ethernet Transceiver          | Ethernet Embebido     |
| CAN Transceiver               | CAN Embebido          |
| RS-232/422/485 Transceiver    | UART                  |
| Mando PlayStation 2           | GPIO                  |
| Sensor distancia ultrasonidos | PWM IC                |
| Sensor distancia infrarrojo   | PWM IC                |

sobre sistemas físicos reales. Éstos introducen efectos no ideales, difíciles de modelar en simulación. De esta forma disponemos de una variedad de situaciones a controlar. Con estos fines se dispone de una diversidad de plantas reales que serán descritas con brevedad a continuación:

#### 5.1. Exoesqueleto ligero-no rígido

Exoesqueleto vestible que proporciona medios de actuación basados en SMA para el codo y la muñeca. La intención de este sistema de cara a su uso educativo es la de evaluar el control para actuadores no lineales afectados por la gravedad y por el tejido blando muscular, que absorbe parte del movimiento.

#### 5.2. Simulador de articulación del codo humano

Se trata de un banco de pruebas formado por dos elementos de acero articulados. Su peso y tamaño derivan de medidas humanas. Este dispositivo, mostrado en la figura 2, como elemento motor cuenta con un motor USM, actuador no lineal. El peso masivo de este dispositivo, permite disponer de un escenario apropiado para estudiar técnicas de compensación de la inercia. Este sistema es ideal para mostrar los beneficios proporcionados por una ejecución del algoritmo mediante el paradigma de tiempo real con paso de ejecución variable, paradigma disponible en nuestro sistema ARCP. Se emplean encoders absolutos y relativos, y sensores de fuerza diseñados en la UC3M como medios de realimentación de información.

#### 5.3. Mano robótica

Un prototipo fabricado en plástico en la propia escuela. Específicamente diseñado para ser actuado mediante fibras SMA



Figura 2: Banco de pruebas de acero a semejanza de un brazo humano. En la imagen, un grupo de estudiantes compara los resultados del control en simulación con los resultados en la realidad.

(Villoslada et al., 2014). Emplea sensores de fuerza en las yemas de los dedos para proporcionar información útil para realizar agarre de objetos. La posición angular de los dedos emplea sensores flexibles. A este dispositivo se le puede añadir un sistema de adquisición electromiográfica (EMG) diseñado en la propia UC3M, de forma que la señal de referencia pueda ser proporcionada por la activación de los músculos del brazo de un alumno. Se trata de un sistema que está empleándose en los estudios de bioingeniería.

#### 5.4. Banco de pruebas de SMA

Este dispositivo consiste en un banco de pruebas en el que controlar varios actuadores no lineales basados en SMA en configuración antagonista. El banco permite controlarlos en posición, velocidad y fuerza; lo que conlleva que sea ampliamente utilizando en los cursos de control como ejemplo de cómo trabajar con sistemas no lineales.

#### 5.5. Motor de Corriente Continua

Planta física que representa uno de los sistemas más utilizados para prácticas en ingeniería de control. En este caso se trata de un sistema portable de reducido tamaño y muy bajo coste. Se emplea un encoder relativo de efecto hall. Representa el ejemplo de planta con actuador lineal disponible en los laboratorios.

#### 5.6. Procesamiento de señales EMG

Dispositivo electrónico de diseño personalizado con excelente relación señal ruido (SNR), llevado a cabo en la UC3M, que permite adquirir señales electromiográficas con el sistema ARCP. Se emplea principalmente para impartir enseñanza sobre teoría de control y procesamiento de señal para estudiantes de bioingeniería. La naturaleza computacionalmente compleja del procesamiento de señal, hacen que este sistema recurra a las funcionalidades avanzadas del ARCP.

### 6. Utilizando el sistema ARCP

El sistema ARCP emplea el mayor nivel de abstracción conocido para programar, hacerlo mediante un lenguaje basado en gráficos. En este caso se trata de un soporte mejorado basado en Matlab/Simulink®. Se pueden emplear bloques gráficos,



Figura 3: Alumno monitorizando la adquisición de datos del sistema EMG.

y diagramas de estados de forma simultánea. El lenguaje textual Matlab® también está soportado, pues es posible integrar código en lenguaje M dentro de un bloque gráfico fácilmente. Cada bloque de programación basado en gráficos, incluyendo los estándares de Simulink® y los propios de la toolbox Beyond Control del sistema ARCP, puede considerarse como una pieza de código modular, el código fuente que generan se encuentra completamente probado y depurado, de forma que se garantiza que programas muy extensos basados en estos bloques modulares funcionarán sin ningún problema.

El controlador hardware embebido del sistema ARCP requiere su propio programa basado en gráficos, mientras que el computador que realiza la función de monitorización de datos necesita también su programa gráfico. De esta forma, el programa de control funcionará en el procesador embebido y el computador será sólo la herramienta de desarrollo y visualización. Se evita así tener que recurrir al modo externo de Matlab/Simulink® (Pinzon-Ardila et al., 2011; Enikov and Campa, 2012); posibilitando el funcionamiento de multitud de toolboxes de Matlab® que no pueden funcionar en modo externo, como SimMechanics® por ejemplo. El determinismo en la transmisión/recepción de los datos está garantizado por la cuidadísima interfaz nativa USB® del sistema ARCP, explicada en el apartado 4.2. También, el prescindir del modo externo es beneficioso para el alumnado, de esta forma el alumno puede entender de manera muy clara que en el hardware controlador se encuentra un programa funcionando de forma independiente al ordenador de propósito general, mientras que en este último sólo se está ejecutando un programa que monitoriza los datos adquiridos. Aunque ambos programas estén realizados en Simulink®, al alumno le resulta fácil distinguir que son distintos. Mientras que al emplear sistemas RCP basados en modo externo, la diferencia no sería tan notoria, puesto que un único programa de Simulink® serviría tanto para programar el hardware del RCP como para monitorizar datos, de forma que a muchos alumnos les costaría distinguir ambos programas, siendo muy difícil o impracticable el plantear sistemas de control distribuidos. Puesto que no sería posible separar el controlador de tiempo real del ordenador que monitoriza los datos. El sistema ARCP presentado en este artículo elimina esta limitación, permitiendo que los controladores hardware puedan ser autónomos respecto del monitor de datos.

El uso de la toolbox personalizada está diseñado para resultar cómodo y sencillo para los usuarios, cada bloque gráfico contiene una descripción de su funcionamiento y opciones dis-

ponibles que se muestra en su propia máscara de configuración. Las opciones más complejas y de bajo nivel de cada periférico y del MCU son configuradas automáticamente en base al tiempo de muestreo, tareas deseadas y configuración seleccionada para cada periférico. Una vez que el modelo gráfico de Simulink® está listo para ser probado, el usuario sólo tiene que apretar el botón “build” para que se genere el código fuente de forma automática, se compile y se programe en la ROM del MCU. Todos estos pasos se realizan de forma transparente al usuario, necesitando un tiempo aproximado de 40 segundos. La herramienta de programación visual del sistema ARCP soporta dos MCU diferentes y hasta dos entornos de desarrollo (IDE) distintos: GNU-ARM y Keil Uvision 5®. El primero de ellos representa la opción utilizada para los cursos de teoría de control, y es un entorno gratuito. Las otras opciones ofrecen mejores optimizaciones del código máquina generado. En todas las opciones el código fuente en lenguaje C resultante permanece disponible en el disco duro del computador.

## 7. Trabajando en el aula con un sistema lineal

En la enseñanza de ingeniería de control una práctica habitual es comenzar tratando sistemas que presentan un comportamiento lineal invariante en el tiempo. Éstos pueden ser sistemas continuos en el tiempo y sistemas discretos, donde las formas más comunes de representarlos son: variables de estado, funciones de transferencia, o fracciones parciales. Una forma de dividir todo este contenido consiste en separar los sistemas de tiempo continuo que se enseñan en primer curso de los sistemas de tiempo discreto y espacio de estados que se enseñan en el segundo curso. Las prácticas juegan un rol importante en el proceso de aprendizaje ya que ofrecen la oportunidad de experimentar la aplicación de la teoría de control sobre sistemas físicos reales, de forma que se incentive y se despierte la motivación e interés del alumnado mediante el aprendizaje basado en proyectos (comúnmente denominado por sus siglas, ABP), involucrando a los alumnos en la resolución de un problema real. La figura 4 muestra una de las plantas reales que consiste en un motor de corriente continua, utilizado en prácticas en los grados de ingeniería en electrónica industrial y automática, tecnologías industriales, ingeniería eléctrica y mecánica. Se utiliza tanto para los contenidos prácticos del primer curso de control (control en tiempo continuo), como para el segundo curso de control (control en tiempo discreto). Para el tercer curso de control (control no lineal), se emplea una planta no lineal.

Para poder aplicar los métodos analíticos para el cálculo de controladores PID y espacio de estados, se debe obtener primero la función de transferencia de la planta. System identification toolbox de Matlab® ofrece varios métodos de identificación de sistemas, uno de ellos obtiene la función de transferencia de la planta en tiempo continuo mediante la introducción de estímulos de entrada en bucle abierto. Con este fin se emplean las funcionalidades avanzadas del ARCP que permiten una mejor aproximación al tiempo continuo y la interfaz USB por no suponer ésta última un lastre computacional. Una vez en posesión de la función de transferencia de la planta, se puede obtener el controlador PID por varios métodos:

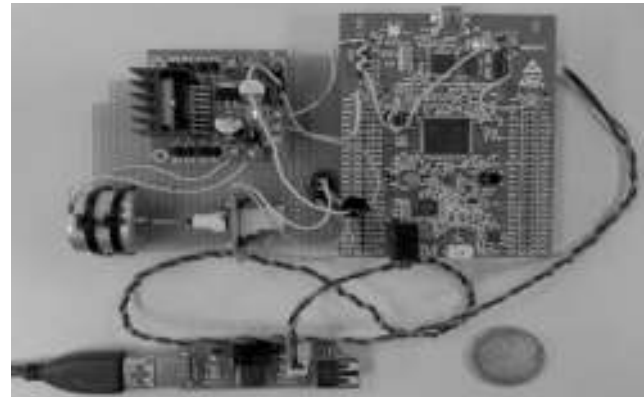


Figura 4: Sistema ARCP junto con el motor DC, encoder relativo y electrónica de potencia. En la parte baja de la figura se encuentra un aislador USB®

### 7.1. Cálculo de reguladores PID discretizados

Partiendo de la función de transferencia del sistema en tiempo continuo se pide diseñar el regulador PD ideal continuo necesario para que el sistema en lazo cerrado cumpla con ciertas especificaciones como sobreoscilación, tiempo de establecimiento y el error en régimen permanente, empleando para ello métodos analíticos aprendidos en la parte teórica de los cursos de control. Una vez hecho esto, se solicita discretizar el regulador por métodos como Euler hacia atrás y así obtener la expresión del regulador PD en función de Z, utilizando el período de muestreo que se considere más adecuado. La parte final de la práctica consiste en implementar el regulador discreto obtenido sobre el sistema físico real y analizar la respuesta midiendo las características en el tiempo del sistema. También, como parte final de la práctica, se pide analizar la carga de la CPU para el controlador discreto que han propuesto y comparar este resultado con el que obtienen al implementar el controlador PD continuo sobre la aproximación al tiempo continuo del ARCP.

Los conocimientos teóricos que implica esta práctica son el análisis en el tiempo necesario para calcular la posición de los polos dominantes en función de las especificaciones exigidas para la planta, el lugar geométrico de las raíces y los controladores de tipo PID a partir del cual se aplican los métodos analíticos para el cálculo del regulador en tiempo continuo del sistema. Por último se utilizan los métodos de discretización para obtener la expresión del regulador PID en tiempo discreto.

### 7.2. Diseño de reguladores por síntesis directa

En este caso se pide diseñar por síntesis directa el regulador que consiga que la función de transferencia en lazo cerrado tenga ganancia unitaria y los polos dominantes se encuentren ubicados dónde se especifica en el enunciado, o se extraiga la posición de éstos de las especificaciones para la respuesta en el tiempo deseada para la planta. Como en el caso anterior el alumno debe utilizar el tiempo de muestreo que considere más adecuado, y tener en cuenta que cómo ha obtenido con anterioridad la función de transferencia en tiempo continuo necesita tener en cuenta el bloqueador que también ha empleado

con anterioridad de cara a obtener el regulador por síntesis directa. Luego para obtener el equivalente discreto de la función de transferencia habrá que discretizar el conjunto bloqueador-función de transferencia. Una vez hecho esto, el alumno implementaría el regulador obtenido sobre el sistema real y analizaría los resultados como las características temporales del sistema controlado. En el transcurso de las prácticas se comparan diferentes casos, como cuando los polos dominantes del sistema sean reales o complejos conjugados.

### 7.3. Control mediante realimentación del estado

A partir de la función de transferencia en tiempo continuo el alumno necesita aplicar los conocimientos sobre la teoría de espacio de estados para controlar la planta del motor. A partir de la función de transferencia debe obtener las variables de estado del motor, con la ayuda de Matlab® puede diseñar el sistema de control de posición mediante realimentación de estados ajustando la ganancia estática necesaria para que el sistema en lazo cerrado cumpla con los requisitos temporales.

## 8. Trabajando en el aula con un sistema no lineal

La planta a utilizar es el banco de pruebas de actuadores SMA. Las fibras SMA son bien conocidas por su no linealidad (Krejci and Kuhnen, 2001; Krulevitch et al., 1996; Dutta and Ghorbel, 2005). Este banco supone la oportunidad didáctica para llevar a cabo el modelado e identificación de un sistema no lineal de forma asistida por computador. Como paso previo a aplicar los conocimientos sobre control para plantas no lineales impartidos en el aula teórica. Se hace uso de las herramientas proporcionadas por Matlab® como System Identification Toolbox. La identificación es llevada a cabo en bucle abierto, mediante la introducción de estímulos de entrada a la planta, mostrada en la figura 6. Dichos estímulos de entrada pueden estar basados en rampas, escalones, entradas aleatorias... la adquisición de datos se realiza a alta frecuencia (2 - 5 KHz) a través del puerto USB® del sistema ARCP, de manera que se pueda reconocer el retardo que presentan los actuadores SMA. Estas capturas de datos junto con los estímulos de entrada a la planta son los datos necesarios para la identificación. La validación del sistema identificado se lleva a cabo comparándolo con el sistema real, la figura 5 ilustra dicha comparación.

En los procesos de identificación de sistemas en bucle abierto, uno de los mejores estímulos de entrada a la planta consiste en una sucesión de entradas en escalón de amplitud creciente hasta un límite, asegurándose que la planta no va a ser forzada. La respuesta frente a una señal del tipo escalón como la mostrada en la figura 5 permite a los procedimientos de identificación de las plantas recurrir a aproximar la constante de tiempo y la ganancia del sistema para cada uno de los escalones. El resultado final para los actuadores SMA suele ser una función de transferencia de segundo orden con un retardo a tener en cuenta. Uno de los métodos de control apropiados para plantas con retardo, ensayados en el aula de prácticas, está basado en Predictores de Smith. Los alumnos calcularán el controlador PID correspondiente para la función de transferencia sin considerar

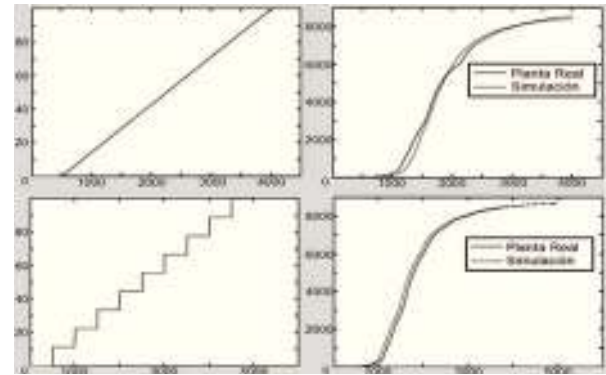


Figura 5: Identificación de los actuadores SMA en bucle abierto. A la izquierda se localizan los estímulos de entrada. A la derecha se muestra el resultado de la identificación asistida por computador comparándola con el sistema real.

los retardos. A continuación, procederán a aplicar la teoría respecto al Predictor de Smith. Por ello puede utilizarse el cero del PID para cancelar el polo del proceso y elegir la ganancia de forma que la constante de tiempo del proceso en bucle cerrado sea sensiblemente menor a la original. Una vez obtenidos los valores a utilizar para el Predictor de Smith, se aplicarán como bucle de realimentación al PID anterior y se podrá comprobar el resultado del controlador en el banco de pruebas.



Figura 6: Banco de pruebas SMA. Arriba una fotografía del dispositivo. Debajo un esquema de su estructura y cada una de sus partes: 1-Sujeción fija. 2-Sujeción móvil. 3-Sensor de desplazamiento lineal. 4-Peso.

La figura 7 muestra el resultado del controlador con Predictor de Smith diseñado por un grupo de alumnos.

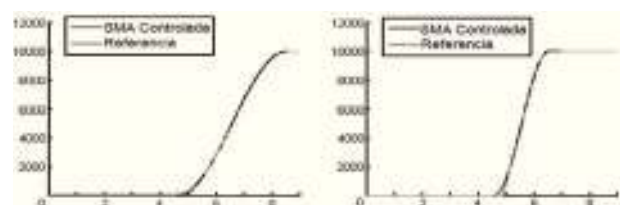


Figura 7: Resultados de un controlador para actuadores no lineales funcionando en el sistema ARCP de la UC3M. El valor del eje y son unidades del sensor, en el eje x se muestra el tiempo en segundos.

## 9. Trabajando en el aula en procesamiento de señal

La funcionalidades avanzadas del blockset utilizado permiten maximizar el rendimiento computacional de las arquitecturas soportadas, en especial el utilizar de forma automática las distintas secciones de memoria RAM disponibles en el MCU STM32F4, dependiendo de los bloques gráficos utilizados, hace que el uso intensivo de etapas de filtrado, transformadas de Fourier y cambios de base repercuta de forma menos negativa en lo que respecta a la carga de la CPU. Este tipo de operaciones presentan por naturaleza unos tiempos de ejecución imprecisos, que pueden depender de la complejidad o aparente aleatoriedad de los datos de entrada. Es por ello que resulta conveniente realizar este tipo de operaciones en un contexto diferente a la tarea de tiempo real, tarea que adquiere los datos y controla la planta física. Este contexto es proporcionado por el uso del tiempo libre entre tareas, que permite realizar las citadas operaciones de procesamiento de señal sin perjuicio de la tarea de tiempo real.

Las funcionalidades avanzadas de Beyond Control han abierto la posibilidad de emplear el MCU STM32F4 para procesamiento de señal en el grado de bioingeniería, a la par que los alumnos emplean la información obtenida tras el procesamiento de señal para actuar sobre una planta física, como puede ser la mano robótica. En las prácticas de las asignaturas de estos planes de estudio en la Universidad Carlos III de Madrid, se ilustra la utilización de las fórmulas de análisis y síntesis respecto del cambio al dominio de la frecuencia y el retorno al dominio del tiempo para un conjunto de datos capturados mediante el ARCP y el sistema de adquisición EMG, así como la realización de operaciones y extracción de propiedades 'ocultas' (si sólo se tuviera en cuenta el dominio temporal) en los datos. Correspondiendo este tratamiento de la información con análisis mediante transformadas de Fourier. Se tienen en cuenta las resultantes tanto para la magnitud como para la fase, pues ambas aportan información relevante procedente de las señales capturadas desde los músculos de los propios alumnos.

En lo que respecta al grado en ingeniería aeroespacial se está barajando la posibilidad de emplear contenidos semejantes mediante Fourier aplicados al análisis estructural, de forma que en el aula de prácticas el alumnado analice las frecuencias de las vibraciones presentes en una estructura real a escala reducida y trabajen en soluciones que las cancelen.

## 10. Tesis de Máster y Tesis Doctorales

Facilitar el trabajo derivado de la participación en proyectos de investigación y facilitar asimismo el desarrollo de las tesis doctorales parejas a dichos proyectos, constituyen el pilar sobre el que se cimentó el concepto de una herramienta de programación de alta eficiencia, gran robustez y avanzadas funcionalidades, que no dependiera de compañías comerciales, cuyo soporte y resolución de problemas pudiera ser tratado en la propia escuela. La herramienta software descrita en este artículo fue originalmente concebida para satisfacer complejas necesidades que surgen en tesis doctorales (Clemente, 2014), integrar éstas en la herramienta y que resulten sencillas de utilizar.

En esta línea existen otras tesis doctorales en curso en la Universidad Carlos III de Madrid y en la Universidad de Toledo que difícilmente podrían llevar a cabo sus etapas experimentales sin contar con un blockset para Simulink® de funcionalidades avanzadas, a este respecto el resto de soluciones existentes no proveen de las soluciones que necesitan estos doctorandos. Y ellos no poseen el conocimiento necesario de programación a bajo nivel y de lenguaje C para poder integrar, aunque tuvieran gran cantidad de tiempo disponible, las funcionalidades que requieren de forma manual en el código autogenerado por otra herramienta de programación gráfica. Respecto a las tesis de Máster, no suelen ser tan complejas y exigentes, pero el soporte multitarea, uso del tiempo libre entre tareas, las optimizaciones para procesamiento de señales y el soporte personalizado están permitiendo que el sistema ARCP se extienda como alternativa de propósito general en la Universidad Carlos III de Madrid. Debido a que el sistema cubre todo el espectro de necesidades, desde sencillas hasta sofisticadas, esto lo hace ideal para su uso con fines de investigación y docentes, pudiendo reemplazar a muchos otros sistemas ya sean comerciales o no.

## 11. Impresión de los alumnos

El acostumbrado efecto rechazo provocado por las soluciones hardware y software empleadas con anterioridad ha desaparecido, debido a la facilidad de uso del sistema ARCP, el bajo coste, que incluso ha permitido que algunos alumnos lleguen a contar con su propia unidad ARCP y la gran facilidad de integración que presentan sus numerosas interfaces I/O. También ha ayudado en ello el reducido tamaño y la naturaleza como sistema embebido, que permite crear sistemas de control autónomos, como robots móviles. El componente hardware del ARCP se muestra en la figura 9.

Los estudiantes están siendo animados a emplear este sistema ARCP para sus proyectos final de carrera, tesis de máster y tesis doctorales; debido a la gran versatilidad en su programación y el alto nivel de abstracción en la misma. Antes de la existencia de este sistema ARCP, casi nadie optaba por emplear el antiguo hardware para llevar a cabo trabajos relacionados con el control. Afortunadamente, esta tendencia ha cambiado completamente en un espacio de tiempo muy reducido, consiguiéndose además que se interesen alumnos de disciplinas muy diversas.

Además del creciente interés mostrado por el alumnado, el impartir cursos de control basándose en un avanzado sistema RCP representa una oportunidad para ganar en experiencia en el diseño de controladores clásicos y modernos, también para enriquecer el conocimiento sobre implementación y organización de sistemas en tiempo real distribuidos al mismo tiempo.

Aparte de estas mejoras detectadas por el profesorado, se ha llevado a cabo una encuesta basada en el test de usabilidad (SUS, del inglés "System usability scale") (Lewis and Sauro, 2009). La población de la encuesta es de 100 estudiantes, procedentes de distintos sistemas educativos, desde no graduados hasta estudiantes cursando su tesis de máster o doctoral. La puntuación media para el sistema ARCP resultante es de 81 de 100, donde la puntuación más alta es mejor. La figura 8 ilustra la distribución de la puntuación.

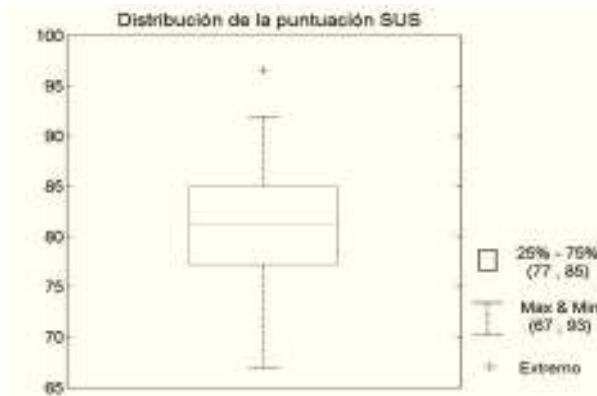


Figura 8: Distribución de la puntuación para el test SUS

El resultado muestra lo que se había detectado, el sistema ARCP presenta una gran facilidad de uso y versatilidad de programación. El uso realizado tanto por los alumnos de la Universidad Carlos III de Madrid como por el resto de escuelas e instituciones que ya utilizan el sistema refleja una gran robustez del software y del hardware, al no haberse encontrado nunca por parte de los alumnos, escuelas e instituciones errores en la generación del código ni en la compilación en los estadios actuales de la herramienta. Huelga decir que en estadios anteriores, los fallos en versiones anteriores localizados por otras escuelas e instituciones de los que hemos sido informados, han ayudado a hacer la herramienta más estable. Por otra parte, el resultado del test SUS refleja una falta de conocimiento y destrezas iniciales en el manejo de Simulink®, sobretodo con datos en forma de vectores y al plantear controladores multitarea o que hagan uso del tiempo libre entre tareas. Esta falta de destrezas necesita ser tratada mediante más clases prácticas a modo de introducción al manejo de Simulink®, no como herramienta de simulación, sino como herramienta de programación de sistemas informáticos. De forma que la curva de aprendizaje en el manejo del sistema ARCP no presente picos abruptos, debido al desconocimiento de las nuevas funcionalidades aportadas a Simulink®.



Figura 9: Comparación entre el tamaño del sistema ARCP para el MCU STM32F4 y el MCU NRF51x22

## 12. Impresión de otras escuelas e instituciones

Actualmente existen otras escuelas e instituciones que emplean Beyond Control para unos u otros propósitos, dentro del ámbito de la investigación y/o docencia. Son centros e instituciones que se mostraron interesados desde el principio, en el año 2011, cuando Beyond Control no era más que un pequeño añadido a otro blockset. Estos centros han servido como evaluadores y localizadores de fallos en la herramienta software, así como para proponer la inclusión de nuevas funcionalidades. Por tanto el blockset Beyond Control se emplea en la Universidad Carlos III de Madrid, donde se ha desarrollado y se sigue desarrollando (Flores-Caballero, 2014), la Universidad Politécnica de Madrid, la Universidad de Toledo (dónde surgió la solicitud de mejorar el soporte de tiempo continuo) y la Universidad de Almería. Como centros extranjeros la Universidad Técnica de Mittelhessen en el campus de Friedberg/Hessen en Alemania, para impartir docencia en ingeniería de control. También en el centro de estudios militares de ingeniería de sistemas automatizados y autodirigidos, en Polonia, impartiendo docencia en ingeniería de control.

## 13. Ventajas del sistema ARCP

Antes de la existencia del actual sistema ARCP de la UC3M, se disponía de una variedad de hardware que se utilizaba en la puesta en práctica de los cursos de control. El primero de ellos en utilizarse fue la placa de expansión para ordenador denominada PMAC® del fabricante Delta Tau®. Esta controladora presenta serios inconvenientes, es muy cara de adquirir y mantener, y sólo permite controladores basados en PID para control de motores con encoders relativos. La variedad de plantas a controlar se torna muy limitada, además su soporte para Matlab/Simulink® es muy simple y poco versátil, por lo que la programación de forma gráfica se vuelve incompetente en cuanto se piensa en una planta algo compleja.

Otras posibilidades de las que se dispuso de forma posterior al sistema anterior fueron tarjetas de adquisición de datos como PCL-818 PCLAB de Advantech® o KUSB-3100 de Keithley®, sistemas menos caros en su adquisición, aunque adolecen de disponer de un soporte de programación gráfica muy sencillo y que no presenta versatilidad, y de requerir la utilización del modo externo de Simulink®, lo que inhabilita pruebas HIL con diversas toolboxes de Matlab®.

En cambio, y tal como se ha mostrado en el apartado 4.2, el sistema ARCP propuesto en este artículo, presenta una versatilidad en su programación a través del lenguaje gráfico equiparable a la que proporciona el lenguaje C; además de poner al alcance de Simulink® la organización de una ejecución multitarea en tiempo real, con soporte mejorado en la aproximación al tiempo continuo, y con una capacidad de integración sin precedentes al permitir acceder desde el lenguaje gráfico a todos los aspectos disponibles en el hardware, con lo que no sólo es más fácil plantear una gran variedad de plantas físicas con las que trabajar, sino que también resulta mucho más atractivo para los alumnos en la consecución de los objetivos en sus tesis de máster y doctorales, pues ya no se encuentran limitados

por sus conocimientos de programación en lenguaje C y micro-electrónica, o por el soporte del lenguaje gráfico proporcionado por algún fabricante. En el momento en que surja una nueva necesidad, ésta puede integrarse como nueva parte constituyente del lenguaje gráfico, así como en el caso de localizarse un error o incompatibilidad de la herramienta, ésta puede subsanarse pues no se depende de terceras compañías en su desarrollo. La herramienta software elimina los factores limitantes a la hora de diseñar sus sistemas de control embebidos, por complejos que sean, tanto a nivel de aplicación de la teoría de control como a nivel del número de tareas y modelos de ejecución que necesiten.

Por otra parte, las ventajas técnicas aportadas por la herramienta, se traducen a menudo en nuevas posibilidades en el aula, como la posibilidad de entrenar el procesamiento de señal, de manera real en una planta. También permite explicar mediante refuerzo experimental la jerarquía en la ejecución de un sistema de múltiples tareas, prioridades de las mismas y cómo afecta esto al conjunto global de la planta.

#### 14. Conclusiones

El artículo presenta un sistema avanzado de prototipado rápido para control (ARCP) de muy bajo coste y altas prestaciones, diseñado para cubrir las necesidades y lagunas funcionales que el resto de sistema RCP existentes no pueden satisfacer. La enorme versatilidad en la programación en base a un lenguaje gráfico, proporcionada por el componente software del ARCP, permite que se pueda mantener un alto nivel de abstracción durante la programación en todo momento, incluyendo toda situación y sistema de control que se imagine, incluso cuando se necesite recurrir a funcionalidades de muy bajo nivel u otras de muy alto nivel nunca antes contempladas para ser utilizadas desde un lenguaje basado en gráficos. Esto ha permitido, que no sólo el sistema ARCP se haya mostrado a sí mismo como una excelente herramienta para utilizar en el aula, sino que también es una herramienta muy versátil para todo tipo de tesis de máster y doctorales en las que se necesite recurrir a un hardware de control embebido.

El uso de un lenguaje de programación basado en gráficos, con el que no se requiere conocimiento alguno sobre la arquitectura hardware para programarla de forma eficiente; cuyas funcionalidades son muy potentes, equiparables al lenguaje de programación C, ha permitido que grupos multidisciplinares de alumnos puedan participar en el desarrollo de cursos semestrales sobre ingeniería de control y en trabajos fin de carrera, máster y doctoral; poniendo en práctica lo aprendido sobre diversas plantas físicas, elevando de esta forma la motivación de los alumnos por participar.

El bajo coste del componente hardware del sistema ARCP, 15 euros para el basado en MCU STM32F4 y 6 euros para el MCU NRF51x22, junto con el soporte software personalizado, creado en el departamento de Ingeniería de Sistemas de la propia Universidad Carlos III de Madrid, ha eliminado los inconvenientes de utilizar sistemas de prototipado rápido para control (RCP) de carácter más convencional y funcionalidades muy básicas de origen comercial, como los propios de dSPACE®

(Busch and Lambeck, 2010; Ballagi and Futas, 2004) or xPC Target® basados en estándar PC-104 (Wahyudi et al., 2005; Chen et al., 2004). Las funcionalidades avanzadas del sistema ARCP como el poder ejecutar modelos con una mejor aproximación al tiempo continuo, el poder organizar esquemas de ejecución multitarea en tiempo real, y posibilitar el acceso y programación al gusto y detalle de las fuentes de interrupción, hacen del ARCP una herramienta de versatilidad sin precedentes, con las que resulta sencillo plantear innovadores escenarios a nivel docente, en los que haya que atender al planteamiento de un sistema de control y a la ejecución multitarea al mismo tiempo, pues ambos son conceptos íntimamente ligados. También, con el sistema ARCP, resulta posible acometer complejas y enrevesadas tesis doctorales sobre control de plantas físicas de comportamiento profundamente no lineal.

Estas nuevas posibilidades docentes abiertas por el sistema ARCP representan un incremento notable en la riqueza curricular del alumnado, además de un notable incremento del interés al poder aplicar el concepto de aprendizaje basado en proyectos en una amplia variedad de escenarios.

#### English Summary

##### Advanced Rapid Control Prototyping system on engineering education for multidisciplinary student groups

#### Abstract

In order to achieve successful results in the control courses for a multidisciplinary audience, a high abstraction layer at programming is required. This abstraction is provided by a custom developed rapid control prototyping system that enables the usage of model based design techniques to control real plants. Model based design is focused on the research and development of the control algorithm, system identification and plant modelling. The usage of the proposed embedded rapid control prototyping system eliminates completely the need for low level input/output control hardware configuration and debugging, making the system capable to be used by the intended multidisciplinary audience, since model based design focuses on design and analysis techniques and shows the relationships between control theory and real controller implementation issues. Several controllable real plants are ready to be used by students, from Bachelor, Master Thesis and Ph.D. degrees. They come from many engineering fields including mechanical, electrical... Also the common drawbacks of commercial rapid control prototyping systems are avoided thanks to the customized low cost rapid control prototyping system presented in this article.

**Keywords:** Control Engineering Education, Teaching Tools and Automation Laboratories, CAD/CAM and CIM.

#### Agradecimientos

Los autores expresan su deseo de agradecimiento por el soporte económico a los proyectos HYPER-Consolider Ingenio

2010, al proyecto RoboCity2030-II-CM (S2009/DPI-1559) y al proyecto RoboHealth. Este artículo está dedicado a la memoria de Valeriu Baciú, su tesis de máster hizo posible la existencia del banco de pruebas para fibras SMA, siendo también precursor de la mano robótica actuada por SMA.

## Referencias

- Aimagin®, Mayo 2015. Opciones de licenciamiento del blockset waijung. In: [https://www.aimagin.com/waijung\\_licenseing\\_options?\\_\\_store=default](https://www.aimagin.com/waijung_licenseing_options?__store=default).
- Ballagi, A., Futas, J., 2004. Teaching and building motion control systems with rapid control prototyping and hardware-in-the-loop methods. In: EPE-PEMC 2004 11th International Power Electronics and Motion Control Conference. pp. 4–514 vol.4.
- Bartosinski, R., Hanzalek, Z., Struzka, P., Waszniowski, L., March 2007. Integrated environment for embedded control systems design. In: Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International. pp. 1–8.  
DOI: 10.1109/IPDPS.2007.370338
- Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Balle, J., Herold, T., Telle, A., Noll, T., Hameyer, K., Aach, T., 2010. MATLAB meets LEGO mindstorms. A freshman introduction course into practical engineering. IEEE Trans. Edu. 53 (2), 306–317.  
DOI: 10.1109/TE.2009.2017272
- Busch, C., Lambeck, S., Sept 2010. Rapid control prototyping of a controller for a experimental helicopter set-up used in control engineering courses. In: Control Applications (CCA), 2010 IEEE International Conference on. pp. 338–343.  
DOI: 10.1109/CCA.2010.5611063
- Chen, C.-H., Tsai, H.-L., Tu, J.-C., Sept 2004. Robot control system implementation with rapid control prototyping technique. In: Computer Aided Control Systems Design, 2004 IEEE International Symposium on. pp. 278–283.  
DOI: 10.1109/CACSD.2004.1393889
- Chen, X., Gong, X., Zhou, H., Xu, Z., Xu, Y., Kang, C., 2010. An economical rapid control prototyping system design with Matlab/Simulink and TMS320F2812 DSP. In: IMECS 2010.
- Clemente, A. M., 2014. Modelado y control de sistemas no lineales de tipo sma. Ph.D. thesis, Universidad Carlos III de Madrid.
- Duma, R., Dobra, P., Sita, I., Petreus, D., Sept 2012. Rapid control prototyping toolbox for the stellaris lm3s8000 microcontrollers. In: Education and Research Conference (EDERC), 2012 5th European DSP. pp. 198–202.  
DOI: 10.1109/EDERC.2012.6532254
- Duma, R., Petreus, D., Sita, V., Dobra, P., Rusu, A., May 2010. Rapid control prototyping toolbox for renesas m32c87 microcontroller. In: Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on. Vol. 1. pp. 1–6.  
DOI: 10.1109/AQTR.2010.5520902
- Dutta, S., Ghorbel, F., April 2005. Differential hysteresis modeling of a shape memory alloy wire actuator. Mechatronics, IEEE/ASME Transactions on 10 (2), 189–197.  
DOI: 10.1109/TMECH.2005.844709
- EE Times Group, 2011. 2011 embedded market study. Tech. rep.
- EE Times Group, 2013. 2013 embedded market study. Tech. rep.
- Enikov, E., Campa, G., Nov 2012. Mechatronic aeropendulum: Demonstration of linear and nonlinear feedback control principles with matlab/simulink real-time windows target. Education, IEEE Transactions on 55 (4), 538–545.  
DOI: 10.1109/TE.2012.2195496
- Flores-Caballero, A., 2014. Sistema avanzado de prototipado rápido para control en exoesqueletos y dispositivos mecatrónicos. Ph.D. thesis, Universidad Carlos III de Madrid.
- Gómez de Gabriel, J., Mandow, A., Fernández-Lozano, J., García-Cerezo, A., 2011. Using LEGO NXT mobile robots with LabVIEW for undergraduate courses on mechatronics. IEEE Trans. Edu. 54 (1), 41–47.  
DOI: 10.1109/TE.2010.2043359
- Gordon, S. N. J., 2009. Computer aided software design via inference and constraint propagation. Integrated Computer-Aided Engineering, IOS Press 16, 181–291.  
DOI: 10.3233/ICA-2009-0312
- Hercog, D., Curkovic, M., Jezernik, K., Oct 2006. Dsp based rapid control prototyping systems for engineering education and research. In: Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE. pp. 2292–2297.  
DOI: 10.1109/CACSD-CCA-ISIC.2006.4776997
- Kim, Y., 2011. Control systems lab using a LEGO mindstorms NXT motor system. IEEE Trans. Edu. 54 (3), 452–461.  
DOI: 10.1109/TE.2010.2076284
- Krejci, P., Kuhnen, K., May 2001. Inverse control of systems with hysteresis and creep. Control Theory and Applications, IEEE Proceedings 148 (3), 185–192.  
DOI: 10.1049/ip-cta:20010375
- Krulevitch, P., Lee, A., Ramsey, P., Trevino, J., Hamilton, J., Northrup, M., Dec 1996. Thin film shape memory alloy microactuators. Microelectromechanical Systems, Journal of 5 (4), 270–282.  
DOI: 10.1109/84.546407
- Lewis, J., Sauro, J., 2009. The factor structure of the system usability scale. In: HCII 2009, San Diego CA, USA.
- Luis M., J., Puerto, R., Óscar Reinoso, Fernández, C., Neco, R., 2005. Recolab: Laboratorio remoto de control utilizando matlab y simulink. Revista Iberoamericana de Automática e Informática Industrial [RIAI] 11 (4), 64–72.  
DOI: <http://dx.doi.org/10.1016/j.riai.2014.09.002>
- Luo, F., Huang, Z., July 2010. Embedded c code generation and embedded target development based on rtw-ec. In: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. Vol. 5. pp. 532–536.  
DOI: 10.1109/ICCSIT.2010.5563906
- MathWorks, 2012. Implementing MATLAB and Simulink Algorithms on FPGAs.  
URL: [MathWorks FPGA Seminar with Xilinx.pdf](http://mathworks.com/seminar/xilinx.pdf) downloadable at MathWorks Website
- Mosterman, P., Sztipanovits, J., Engell, S., March 2004. Computer-automated multiparadigm modeling in control systems technology. Control Systems Technology, IEEE Transactions on 12 (2), 223–234.  
DOI: 10.1109/TCST.2004.824280
- Pinzon-Ardila, O., Angel, L., Useche, M., 2011. xPC target an option for position control of robotic manipulators. In: Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC). pp. 1–6.  
DOI: 10.1109/LARC.2011.6086795
- Rusu, C., Radulescu, M., Balan, H., Sept 2007. Embedded toolbox for f24x dsb target microcontroller. In: Electrical Machines and Power Electronics, 2007. ACEMP '07. International Aegean Conference on. pp. 556–559.  
DOI: 10.1109/ACEMP.2007.4510564
- Valera, A., Soriano, A., Vallés, M., 2014. Plataformas de bajo coste para la realización de trabajos prácticos de mecatrónica y robótica. Revista Iberoamericana de Automática e Informática Industrial [RIAI] 11 (4), 363–376.  
DOI: <http://dx.doi.org/10.1016/j.riai.2014.09.002>
- Villoslada, A., Flores-Caballero, A., Copaci, D., Blanco, D., Moreno, L., Nov 2014. High-displacement fast-cooling flexible shape memory alloy actuator: Application to an anthropomorphic robotic hand. In: Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on. pp. 27–32.  
DOI: 10.1109/HUMAN0IDS.2014.7041333
- Vouzis, P., Kothare, M., Bleris, L., Arnold, M., Sept 2009. A system-on-a-chip implementation for embedded real-time model predictive control. Control Systems Technology, IEEE Transactions on 17 (5), 1006–1017.  
DOI: 10.1109/TCST.2008.2004503
- Wahyudi, M., Salami, M., Albagul, A., 2005. Development of a microcontroller-based control system with a hardware-in-the-loop (HIL) method for control education using MATLAB/simulink/xPC Target. International Journal on Engineering Education 21 (5), 846–854.

## Apéndice A.

Los autores del presente artículo han tenido a bien proporcionar un enlace de descarga, que contiene imágenes de la herramienta software descrita, dado que éstas no cabrían en el artículo: <https://mega.nz/#!fk52UZTY>