

Tesis Doctoral



Resumen de la Tesis Doctoral

Agregando Interfaz Humana Interactiva a Software de Ingeniería

Gonzalo Alberto Farias Castro

Ingeniero en Informática

Departamento de Informática y Automática
Escuela Técnica Superior de Ingeniería Informática
Universidad Nacional de Educación a Distancia

Madrid, 2010

Department Informática y Automática
E.T.S. de Ingeniería Informática

Title Adding Interactive Human Interface
to Engineering Software

Author Gonzalo Alberto Farias Castro

Degree Computer Scientist
Facultad de Ingeniería, Ciencias y Administración
Universidad de la Frontera

Supervisors Sebastián Dormido Bencomo
Francisco Esquembre Martínez

Índice general

1. Resumen	3
1.1. Laboratorios virtuales y remotos	4
1.2. Interactividad y visualización	5
1.3. Sistemas de control en tiempo real	7
1.4. Ambiente de experimentación	8
1.5. Objetivos	9
1.5.1. Objetivo general	9
1.5.2. Objetivos específicos	10
1.6. Contenidos de la Tesis Doctoral	10
1.7. Contribuciones Principales	13
1.7.1. Componentes de software desarrollados	13
1.7.2. Publicaciones	14
1.7.3. Proyectos de investigación	17
2. Conclusiones y Trabajos Futuros	21
2.1. Conclusiones	21
2.2. Trabajos Futuros	25
Bibliografía	29

Capítulo 1

Resumen

Desde las últimas décadas, las tecnologías de la información y comunicación (TICs) han ido presentando un impacto cada vez mayor en la educación. Muchos avances en educación han sido gracias a la evolución de las tecnologías como Internet, servicios web, video conferencias, cursos web, gráficas interactivas, entre otras.

Un informe del año 2001 de la Asociación Americana de la Industria del Software y la Información (SIIA) de los EE.UU. titulado “Redefining Education” concluye que *la tecnología esta redefiniendo la educación*. El informe establece que “el cambio de paradigma en nuestros objetivos y modelos educativos esta recién comenzando, confluendo las soluciones del siglo XXI junto a las infraestructuras del siglo XX y las tradiciones educativas del siglo XIX” (SIIA 2001).

Tal diagnóstico no sólo asume la tremenda transformación que las tecnologías de la información están causando en cada aspecto de la sociedad, sino que también revela los profundos cambios que el paradigma educacional esta enfrentando, el cual, debido a los cambios tecnológicos, se esta transformando, desde un proceso estático, a un proceso dinámico (ver Figura 1.1). Mucho mejor adecuado a las exigencias de una sociedad que

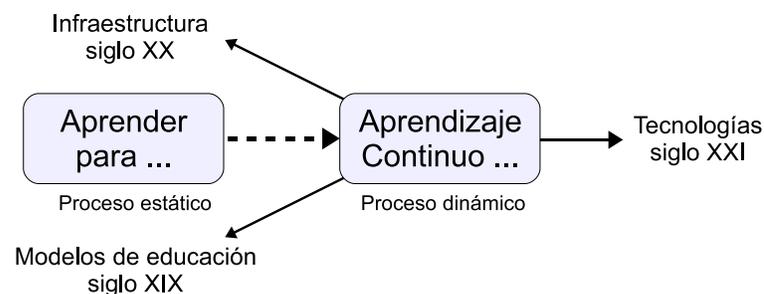


Figura 1.1: Hacia un nuevo paradigma de la educación.

requiere de una enseñanza continua durante toda la vida.

El uso de las TICs proporciona grandes oportunidades a la educación. En los últimos años, la comunidad científica ha hecho un gran esfuerzo para tomar ventaja de estas nuevas posibilidades, mostrando una fructífera investigación en muchas disciplinas tales como la educación de la ingeniería de control (Heck 1999, Dormido 2002, Sánchez 2001, Latchman et al. 2001).

1.1. Laboratorios virtuales y remotos

En la educación del control el impacto de estas tecnologías es aún más grande. La experimentación y prueba de los laboratorios tradicionales son elementos esenciales para los estudiantes, quienes necesitan comprender los conceptos fundamentales desde las perspectivas de la teoría y la práctica. En la literatura se pueden encontrar muchos casos de estudio relacionados con laboratorios tradicionales, ver por ejemplo (Leva 2003, 2004, Wellstead 1983, Chandrasekara & Davari 2004, Spong & Block 1995, Radharamanan & Jenkins 2007).

Sin embargo, los recursos limitados debido a los altos costes asociados con los equipos, el espacio y el personal de mantenimiento, imponen ciertas restricciones en el uso de este material. Por ello, se ha realizado una gran cantidad de investigación para solucionar este problema (Latchman et al. 1999, Gillet et al. 2008, Dormido et al. 2008, Farias et al. 2010, Gomes & Bogosyan 2009, Leva & Donida 2008).

Dos de los resultados más importantes son los denominados *laboratorios virtuales* y *remotos*. Una taxonomía de estos tipos de recursos de aprendizaje puede ser hecha considerando el siguiente criterio (ver Tabla 1.1):

Tipo de recurso indica si el recurso es un equipo real o un modelo de un sistema físico.

Tipo de acceso indica si el recurso y el estudiante están o no en el mismo lugar.

De acuerdo a estos criterios, un laboratorio tradicional consiste en un recurso real con acceso local. Esta configuración implica que el estudiante tiene que desplazarse a la ubicación del equipamiento para ejecutar la experimentación.

Tabla 1.1: Tipos de recursos de aprendizaje

		Recurso	
		Real	Simulado
Acceso	Local	Laboratorio Tradicional	Simulación
	Remoto	Laboratorio Remoto	Laboratorio Virtual

Una simulación consiste normalmente de un modelo computacional de una planta real. Las simulaciones han sido muy populares en los últimos años. Sin embargo, en la educación de la ingeniería del control, las simulaciones deben ser típicamente ejecutadas en un ordenador de la universidad, ya que éstas normalmente requieren de un software de ingeniería especializado y costoso.

La operación remota de un equipo real se denomina comúnmente laboratorio remoto. Este tipo de recurso de aprendizaje ofrece a los estudiantes la manipulación o teleoperación desde sus computadores de las plantas existentes en la universidad. Este tipo de experimentación evita las restricciones de tiempo y lugar de los laboratorios tradicionales.

Finalmente, aquellas simulaciones que pueden ser ejecutadas de forma remota desde los computadores de los estudiantes se denominan comúnmente como laboratorios virtuales o simulaciones distribuidas.

Los laboratorios virtuales y remotos proveen, gracias al acceso remoto, grandes oportunidades a los profesores con el fin de apoyar procesos continuos de aprendizaje por parte de los estudiantes. Esta tesis está enfocada en proveer herramientas a los profesores para facilitar la creación de laboratorios virtuales.

1.2. Interactividad y visualización

Dos aspectos relevantes de las simulaciones utilizadas en la enseñanza del control son la *interactividad* y la *visualización* (Dormido, Dormido-Canto, Dormido, Sánchez & Duro 2005, Sánchez et al. 2005). Tradicionalmente en simulaciones de ingeniería de control, los análisis típicos de la respuesta de los sistemas vienen de las características de las señales de salida tales como la forma de onda y el periodo (Uran & Jezernik 2008, Wu et al. 2008). Así el análisis de la respuesta del sistema no es explícita ni intuitiva, es

decir, no son en realidad comprendidas por las personas de una forma natural. Sin una adecuada visualización, las simulaciones pueden convertirse en un elemento educativo difícil de estudiar. Además, mucho del análisis realizado es “fuera de línea” o de una forma estática. Esto es, las señales son obtenidas y observadas solamente cuando la simulación ha terminado, con lo que los estudiantes raramente interactúan, cambiando parámetros o entradas, mientras la simulación se está ejecutando. Este rol pasivo de los estudiantes frena el proceso de aprendizaje considerablemente.

Un uso adecuado de la interactividad y la visualización ayuda a minimizar estos dos problemas. Los profesores pueden utilizar la capacidad gráfica de los computadores modernos para agregar un nivel sofisticado de elementos visuales a las simulaciones, con el fin de producir objetos de aprendizaje más intuitivos y naturales. Esta capa de interactividad y visualización avanzada se conoce comúnmente como interfaz humana.

La interacción en línea (o al vuelo) proporciona a los estudiantes la posibilidad de modificar algunas entradas o parámetros del sistema mientras se está ejecutando la simulación. Esto permite a los estudiantes comprender de manera más sencilla y rápida las relaciones entrada/salida, así como apreciar el grado de influencia que algún parámetro tiene en la respuesta global del sistema.

A pesar de que la importancia de la interactividad es aceptada por la comunidad de educación en ingeniería, su uso en las simulaciones no es la norma. La principal razón puede deberse al hecho de que el desarrollo de simulaciones interactivas no es una tarea sencilla desde el punto de vista de la programación. Los profesores de ingeniería, quienes no son normalmente expertos programadores, pueden verse en problemas cuando intentan agregar interactividad o visualización avanzada a simulaciones de ingeniería existentes. La presencia de diferentes lenguajes, técnicas de programación, protocolos de redes, entre otras, sólo complica más las cosas.

Una segunda fuente de frustración es la falta de reusabilidad. Típicamente, cada software de ingeniería tiene su propio marco de trabajo para añadir a las simulaciones interfaces gráficas de usuario (GUIs). Sin embargo, estas GUIs no pueden utilizarse con otro software de ingeniería o incluso con otras simulaciones desarrolladas con el mismo software de ingeniería. Esta carencia de modularidad de las interfaces humanas es una característica común de una gran cantidad de software de ingeniería.

En esta tesis se tratan los dos problemas comentados anteriormente. La tesis in-

introduce un enfoque sistemático para comunicar software de ingeniería con lenguajes de programación de propósito general. La solución propuesta, denominada **interoperate approach**, permite a los autores diseñar e implementar de forma separada la interfaz gráfica de usuario y la simulación de ingeniería, conectándolas a través de un protocolo estándar. Esta modularidad permite el uso de herramientas de autor que faciliten la creación de interfaces gráficas junto con un software de ingeniería para la simulación de un sistema. Esto además promueve la reutilización tanto de la simulación como de la interfaz gráfica de usuario. El enfoque propuesto es una generalización de un trabajo previo descrito en (Sánchez et al. 2002, 2004, 2005) para conectar MATLAB con aplicaciones Java. El nuevo enfoque propuesto en esta tesis permite manipular, desde aplicaciones Java, software de ingeniería estándar tales como MATLAB, Simulink, Scilab y Sysquake (Dormido, Esquembre, Farias & Sánchez 2005, Farias et al. 2010, Farias, Keyser, Dormido & Esquembre 2009, Fabregas et al. 2010).

1.3. Sistemas de control en tiempo real

Además de establecer un enfoque general válido para cualquier software de ingeniería, la tesis considera con especial detalle la simulación de los sistemas de control en tiempo real. Estos sistemas están sujetos a un reciente interés debido a que, contrariamente al diseño tradicional, una nueva metodología de análisis considera los aspectos de tiempo real y control de forma conjunta. Esta nueva perspectiva ofrece un interés atractivo en sí misma, pero además los sistemas de control en tiempo real proporcionan un caso de estudio perfecto para el enfoque de comunicación propuesto gracias a la disponibilidad de TrueTime.

TrueTime es un simulador gratuito, basado en MATLAB/Simulink, de sistemas de control en tiempo real y en red que ha sido desarrollado en la Universidad de Lund desde el año 1999. TrueTime proporciona principalmente dos tipos de bloques, el TrueTime Kernel y el TrueTime Network. Estos bloques simulan el comportamiento de un computador y la red en un sistema de control en tiempo real. Los profesores pueden entonces construir simulaciones en Simulink conectando la dinámica de la planta, modelada mediante bloques Simulink, con las entradas y salidas de los bloques TrueTime Kernel y TrueTime Network.

Debido a que TrueTime es una librería de MATLAB, la interfaz creada en esta tesis para comunicar MATLAB y Java fue directamente utilizada para agregar interactividad y visualización a las simulaciones diseñadas con TrueTime. Los laboratorios virtuales de los sistemas de control en tiempo real desarrollados mostraron que el enfoque *interoperate approach* funciona perfectamente.

Las características especiales de los sistemas de control en tiempo real, las cuales causan un gran número de eventos necesarios para ejecutar la simulación en TrueTime, introdujeron la necesidad de añadir métodos particulares al enfoque de comunicación. Tales métodos son también especialmente adecuados para los modelos Simulink que presentan un elevado número de eventos, proporcionando un incremento sustancial en la velocidad de simulación (Farias, Årzén, Cervin, Dormido & Esquembre 2009, Farias et al. 2007).

La necesidad de una licencia de MATLAB para crear y utilizar las simulaciones de tiempo real siempre acarrea inconvenientes, especialmente cuando las simulaciones deben ser distribuidas a los estudiantes para su análisis. Para minimizar este inconveniente, en esta tesis se ha desarrollado una librería Java que reproduce parte de la funcionalidad del modelo de tarea de TrueTime. Esta librería Java, llamada JavaTrueTime (JTT), no implementa toda la funcionalidad que TrueTime proporciona, sino que se centra en proporcionar a los instructores los elementos mínimos requeridos para la construcción efectiva de simulaciones pedagógicas de sistemas de control en tiempo real (Farias, Cervin, Årzén, Dormido & Esquembre 2009, Farias et al. 2008). La tesis muestra cómo desarrollar laboratorios virtuales en Java mediante el uso de JTT. Estos ejemplos pueden ser utilizados en cualquier curso introductorio de sistemas de control en tiempo real.

1.4. Ambiente de experimentación

Un tercer tópico considerado en esta tesis es la generación de experimentos. El objetivo último de la construcción de una simulación es ejecutar con ella experimentos. Un experimento consiste básicamente en extraer información de las salidas de un modelo por medio de la excitación de sus entradas. Típicamente, las herramientas de simulación de ingeniería poseen un lenguaje específico (tipo *script*) para crear experimentos, los cuales

normalmente son muy básicos y no explotan todas las capacidades que las tecnologías modernas ofrecen. Algunas características deseables, tales como agregar nuevos eventos en tiempo de ejecución o comparar sobre un mismo eje de coordenadas las respuestas de dos instancias de una simulación, no están actualmente soportadas aún en todas las herramientas de simulación estándar.

La tesis propone un ambiente de experimentación básico, un conjunto de herramientas que permita a los profesores un amplio y muy flexible uso de sus simulaciones (y quizás de otras creadas por sus colegas) mediante la manipulación de éstas en la misma forma que los programadores manipulan las clases y objetos en los lenguajes orientados a objetos. Las características claves ofrecidas por el ambiente de experimentación propuesto son discutidas en detalle en la presente tesis.

Con el objetivo de probar su validez, el ambiente de experimentación propuesto ha sido implementado en la herramienta de autor denominada Easy Java Simulations. Los autores pueden por tanto utilizar este ambiente de experimentación para construir toda la funcionalidad requerida para ejecutar experimentos con sus laboratorios virtuales (Esquembre et al. 2007). Algunos ejemplos de uso son proporcionados para demostrar la potencialidad de la implementación realizada.

Estos tres problemas comentados, agregar interactividad y visualización a simulaciones de ingeniería, desarrollar simulaciones interactivas de sistemas de control en tiempo real, y el uso de un ambiente de experimentación son analizados extensamente con el fin de proporcionar a los educadores de ingeniería herramientas de autor que aprovechen algunas de las grandes oportunidades que las modernas tecnologías de la información y comunicación ofrecen.

1.5. Objetivos

A continuación se describen el objetivo general y los objetivos específicos de la tesis doctoral.

1.5.1. Objetivo general

El objetivo general de esta tesis es proporcionar herramientas de autor a profesores que faciliten el diseño y la construcción de simulaciones de ingeniería orientadas a la

enseñanza. Desde un muy amplio rango de temas a ser considerados, esta tesis se enfoca en la enseñanza de la ingeniería del control. Sin embargo, muchos de los resultados obtenidos en esta tesis pueden ser aplicados fácilmente a otras áreas de ingeniería.

1.5.2. Objetivos específicos

A continuación se mencionan los objetivos específicos definidos para esta tesis:

- Diseñar un enfoque para agregar interfaces humanas interactivas a las simulaciones creadas con un software de ingeniería.
- Implementar librerías para manipular, local y remotamente, software de ingeniería, ampliamente conocidos, desde programas escritos en lenguaje Java.
- Proporcionar herramientas de autor para facilitar la creación de simulaciones interactivas de sistemas de control en tiempo real.
- Definir e implementar un ambiente avanzado para ejecutar experimentos sofisticados con simulaciones.

Estos objetivos son desarrollados en cada parte de la tesis tal como se describen en la siguiente sección.

1.6. Contenidos de la Tesis Doctoral

Este trabajo se divide en tres partes principales. La primera, introduce las características claves de un enfoque para interoperar software de ingeniería con el fin de crear simulaciones interactivas. Después de esto, una implementación del enfoque es descrita en detalle. La primera parte finaliza con un uso altamente sofisticado de las librerías desarrolladas.

La segunda parte esta relacionada con la creación de simulaciones educacionales e interactivas de sistemas de control en tiempo real. Las librerías desarrolladas en la primera parte son utilizadas para construir las simulaciones junto con una toolbox de MATLAB denominada TrueTime. Además en esta segunda parte se desarrolla una librería Java para la creación de simulaciones interactivas de sistemas de control en tiempo real.

La parte final de la tesis trata de la especificación e implementación de un ambiente de experimentación para manipular de forma sofisticada simulaciones existentes.

Una discusión más amplia de cada capítulo se presenta a continuación.

Capítulo 2

El capítulo describe las claves del enfoque **interoperate approach**. La idea es proporcionar una forma genérica para agregar interfaces humanas interactivas a simulaciones que son ejecutadas sobre software de ingeniería. El enfoque ayuda a los autores a enfrentar la creación de simulaciones interactivas dividiendo el proceso completo en dos etapas independientes. La primera etapa implica la descripción del modelo utilizando un software de ingeniería adecuado. La segunda se centra en el desarrollo de la interfaz interactiva, la cual proporciona a los usuarios de la simulación una forma de interactuar a un alto nivel con la simulación. El enlace entre la interfaz de usuario y la simulación de ingeniería se lleva a cabo utilizando un conjunto de métodos, permitiendo alcanzar, desde la interfaz interactiva, un control completo sobre la simulación de ingeniería.

Capítulo 3

Siguiendo la descripción del enfoque de interoperación presentado en el capítulo anterior, un conjunto de métodos son implementados para manipular software de ingeniería. El resultado final es un grupo de librerías Java que permite a los autores manipular de una misma forma distinto software de ingeniería tales como MATLAB, Scilab y Sysquake. En este capítulo además se describe un programa Java, denominado **JIM** server, que soporta el enfoque **interoperate approach** para el control remoto de simulaciones generadas con MATLAB/Simulink.

Capítulo 4

Una vez que las librerías para el control de software de ingeniería son discutidas, este capítulo se centra en la creación de interfaces de usuario interactivas. Este importante tópico podría ser una tarea extremadamente difícil, especialmente para aquellos profesores de ingeniería que no son expertos programadores. Por esto, la herramienta de simulación Easy Java Simulations, la cual ha sido diseñada para ayudar en la construcción de interfaces interactivas en Java, es presentada. Esta herramienta puede utilizar directamente las librerías desarrolladas en los capítulos previos con el fin de manipular las simulaciones de ingeniería desde una interfaz de usuario interactiva. Aunque este uso

directo de Easy Java Simulations ayuda enormemente al proceso de construcción de las simulaciones requeridas, una integración más profunda con las librerías, desarrolladas en el Capítulo 3, en Easy Java Simulations es presentada como manera de facilitar aún más la manipulación de software de ingeniería.

Capítulo 5

La integración de sistemas de tiempo real y la teoría de control ha sido tratada recientemente por la comunidad científica. El *co-diseño* de sistemas de control y de tiempo real requiere de nuevas herramientas. Una de ellas, es la toolbox, basada en MATLAB/Simulink, denominada TrueTime (M. Ohlin & Cervin 2007). La herramienta ofrece gran flexibilidad y potencialidad para ejecutar un diseño y análisis profundo mediante la simulación de sistemas de control en tiempo real. Sin embargo, las simulaciones obtenidas utilizando la toolbox carecen de la interactividad y visualización aconsejada para la enseñanza. De este modo, este capítulo toma los resultados de los capítulos anteriores para agregar las interfaces humanas interactivas a las simulaciones desarrolladas en TrueTime. El trabajo en este capítulo va un paso más allá y presenta una librería Java, denominada **JTT** y basada en TrueTime, la cual es específicamente diseñada para el desarrollo de simulaciones con propósitos pedagógicos. La librería no soporta por el momento toda la funcionalidad de Truetime, pero permite a los autores la creación sencilla de una gran variedad de simulaciones interactivas de sistemas de control en tiempo real.

Capítulo 6

En este capítulo se especifican los elementos básicos para realizar experimentos con una simulación. La manipulación de las simulaciones mediante el ambiente de experimentación incluye algunas carecterísticas interesantes proporcionadas por los lenguajes y computadores modernos. Los elementos requeridos por los experimentos son implementados en Easy Java Simulations con el fin de obtener un avanzado ambiente de experimentación. Al final del capítulo se presentan algunos ejemplos sencillos con el fin de destacar las principales características de la implementación. Aspectos particulares de los experimentos, desde el punto de vista pedagógico, son también discutidos en este capítulo.

Capítulo 7

El capítulo discute las principales conclusiones y trabajos futuros de las líneas de investigación descritas en la tesis.

1.7. Contribuciones Principales

Las contribuciones durante la formación doctoral son divididas en dos aspectos principales: Componentes de Software y Publicaciones.

1.7.1. Componentes de software desarrollados

Los resultados concretos de esta tesis doctoral incluyen el diseño de protocolos, de librerías de programación, de la creación un servidor, y de laboratorios virtuales y remotos. Los resultados más importantes son resumidos a continuación.

- **Protocolos:** Quizás la contribución más importante de esta tesis consiste en la definición de un protocolo de comunicación para implementar el enfoque **interoperate approach**. Este protocolo esta basado en experiencias acumuladas durante el desarrollo de la tesis. El protocolo permite la interoperación de software de ingeniería estándar con lenguajes de programación de propósito general para facilitar la creación de simulaciones interactivas avanzadas. El protocolo que soporta el ambiente de experimentación descrito en el Capítulo 6 es una segunda contribución de esta tesis.
- **Librerías:** Hay dos librerías interesantes a destacar: **JIMC** and **JTT**. La primera puede ser utilizada por profesores para manipular desde programas Java el software MATLAB/Simulink de acuerdo a las especificaciones requeridas por el enfoque de interoperación descrito en el Capítulo 2. La segunda librería fue desarrollada para facilitar la creación de simulaciones interactivas para la enseñanza de sistemas de control en tiempo real. La librería esta basada en la toolbox de MATLAB TrueTime.
- **Programas:** El programa más interesante desarrollado en esta tesis se denomina **JIM** server. Este programa, descrito en el Capítulo 3, permite la utilización re-

mota desde programas Java de las simulaciones desarrolladas con el software de ingeniería MATLAB/Simulink.

- **Laboratorios:** Muchas simulaciones fueron desarrolladas durante el periodo de doctorado, entre otras, puede destacarse las simulaciones de ejemplo utilizadas para demostrar la manipulación de MATLAB, Scilab y Sysquake desde Easy Java Simulations. Otras simulaciones interesantes son descritas en el capítulo 5, las cuales pueden ser utilizadas como parte de un curso introductorio de sistemas de tiempo real. Cabe destacar además, el laboratorio remoto desarrollado para ejecutar control en red de un servo motor utilizado en un curso introductorio de control en la Universidad de Gante.

1.7.2. Publicaciones

Durante la tesis doctoral varios artículos han sido publicados en revistas especializadas y conferencias internacionales. Muchos de los trabajos han sido obtenidos como resultado directo de esta tesis, otros, sin embargo, han sido desarrollados en colaboración con diferentes grupos de investigación.

Publicaciones en revistas especializadas

Los siguientes artículos han sido publicados en revistas y están directamente relacionados con la tesis doctoral:

- G. Farias, K. Årzén, A. Cervin, S. Dormido, F. Esquembre (2010) *Teaching Embedded Control Systems*, International Journal of Engineering Education (accepted for publication).
- G. Farias, R. De Keyser, S. Dormido, F. Esquembre (2009) *Developing Networked Control Labs: A Matlab and Easy Java Simulations Approach*, IEEE Transactions on Industrial Electronics, (accepted, DOI: 10.1109/TIE.2010.2041130).
- N. Duro, R. Dormido, H. Vargas, S. Dormido-Canto, J. Sánchez, G. Farias, F. Esquembre, S. Dormido (2008) *An Integrated Virtual and Remote Control Lab: The Three-Tank System as a Case Study*, Computing in Science and Engineering Magazine, Vol. 10, Issue 4, pp:50-58. Ed: IEEE.

- H. Vargas, J. Sánchez, N. Duro, R. Dormido, S. Dormido-Canto, G. Farias, S. Dormido, F. Esquembre, C. Salzmann, and D. Gillet. (2008) *A Systematic Two-Layer Approach to Develop Web-based Experimentation Environments for Control Engineering Education*, Intelligent Automation and Soft Computing, Vol. 14, Num. 4, pp:505-524, ISSN: 1079-8587.
- R. Dormido, H. Vargas, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, F. Esquembre, S. Dormido (2008) *Development of a Web-based Control Laboratory for Automation Technicians: The Three Tank System*, IEEE Transaction on Education, Vol. 51, Num. 1, pp: 35-44.

Publicaciones en revistas especializadas en revisión

Los siguientes artículos son el resultado del trabajo desarrollado en la tesis doctoral que aún están bajo proceso de revisión en revistas especializadas.

- G. Farias, A. Cervin, K. Árzén, S. Dormido, F. Esquembre (2009) *Java Simulations of Embedded Control Systems*, Submitted to Real-Time Systems (Springer).
- E. Fabregas, G. Farias, S. Dormido-Canto, S. Dormido, F. Esquembre (2010) *A Practical Approach for Remote Interaction with a Real Plant*, Submitted to Computer & Education (Elsevier).
- J. Sánchez, S. Dormido-Canto, G. Farias, S. Dormido, F. Godoy (2010) *Understanding Automatic Control Concepts by Playing Games*, Submitted to International Journal of Engineering Education.

Publicaciones en conferencias

Los siguientes artículos han sido publicados en conferencias, nacionales e internacionales, principalmente relacionadas a la ingeniería del control. Debido al alto número de publicaciones, sólo las más importantes son descritas.

- G. Farias, R. De Keyser, S. Dormido, F. Esquembre (2009) *Building Remote Labs Using Easy Java Simulation and Matlab*, The European Control Conference 2009, August 23-26, 2009, ISBN: 978-963-311-369-1, Budapest, Hungary.

- G. Farias, A. Cervin, K. Årzén, S. Dormido, F. Esquembre (2008) *Multitasking Real-Time Control Systems in Easy Java Simulations*, Proceedings of the 17th IFAC World Congress 2008, ISBN: 978-1-1234-7890-2, Seoul, Korea.
- G. Farias, S. Dormido, F. Esquembre, H. Vargas, S. Dormido-Canto (2008) *Laboratorio Virtual Para la Enseñanza de Técnicas de Reconocimiento de Patrones*, XIII Latin-American Congress on Automatic Control. Mérida, Venezuela.
- G. Farias, M. Santos, V. López (2008) *Brain Tumour Diagnosis with Wavelets and Support Vector Machines*, 3rd International Conference on Intelligent System and Knowledge Engineering, Proceedings of the 2008 3rd ISKE, IEEE Press, ISBN: 978-1-4244-2197-8, pp: 1453-1459, November 17-19, Xiamen, China.
- F. Esquembre, S. Dormido, G. Farias (2007) *Defining and Performing Experiments in Virtual Laboratories*. International Conference on Engineering Education, ICEE2007. September 3-7. Coimbra Portugal.
- G. Farias, K. Årzén, A. Cervin (2007) *Interactive Real-Time Control Labs with TrueTime and Easy Java Simulations*, In Proceedings of the International Multi-conference on Computer Science and Information Technology, International Workshop on Real Time Software, pp.811-820. Wisla, Poland.
- G. Farias, F. Esquembre, J. Sánchez, S. Dormido, H. Vargas, S. Dormido-Canto, R. Dormido, N. Duro (2006) *Laboratorios Virtuales Remotos Usando Easy Java Simulations y Simulink*, Jornadas de Automática. Almería, España.
- H. Vargas, R. Dormido, N. Duro, J. Sánchez, S. Dormido-Canto, G. Farias, S. Dormido, F. Esquembre (2006) *Heatflow: Un laboratorio basado en web usando Easy Java Simulations y Labview para el entrenamiento de técnicas de automatización*, XII Latin-American Congress on Automatic Control. Bahía, Brasil.
- G. Farias, F. Esquembre, J. Sánchez, S. Dormido, H. Vargas, S. Dormido-Canto, R. Dormido, N. Duro (2006) *Desarrollo de laboratorios virtuales, interactivos y remotos utilizando Easy Java Simulations y Modelos Simulink*, XII Latin-American Congress on Automatic Control. Bahía, Brasil.

- S. Dormido, F. Esquembre, G. Farias, J. Sánchez (2005) *Adding interactivity to existing Simulink models using Easy Java Simulations*, 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05) Seville, Spain.

Publicaciones en revistas especializadas obtenidas en colaboración

Los siguientes artículos son algunas de las publicaciones obtenidas mediante la colaboración con diferentes grupos de investigación. El tema principal de estos artículos es la aplicación de técnicas de reconocimiento de patrones a las bases de datos del reactor experimental de fusión TJ-II del CIEMAT.

- S. Dormido-Canto, G. Farias, J. Vega, R. Dormido, J. Sánchez, N. Duro, H. Vargas, A. Murari, and JET-EFDA Contributors. (2008) *Classifier based on support vector machine for JET plasma configurations*, Review of Scientific Instruments, Vol. 79, pp: 10F326-1/10F326-3, ISSN: 0034-6748.
- S. Dormido-Canto, G. Farias, R. Dormido, J. Sánchez, N. Duro, H. Vargas, J. Vega, G. Ratta, A. Pereira, A. Portas (2008) *Structural pattern recognition methods based on string comparison for fusion database*, Fusion Engineering and Design, Vol. 83, Issue 2-3, pp: 421-424. ISSN: 0920-3796. Ed. Elsevier.
- G. Farias, S. Dormido-Canto, J. Vega, J. Sánchez, N. Duro, R. Dormido, M. Ochando, M. Santos, G. Pajares (2006) *Searching for patterns in TJ-II time evolution signals*, Fusion Engineering and Design, Vol. 81, pp: 1993-1997, ISSN: 0920-3796, Ed. Elsevier.
- G. Farias, R. Dormido, M. Santos, N. Duro (2005) *Image Classifier for the TJ-II Thomson Scattering Diagnostic: Evaluation with a Feed Forward Neural Network*, Lecture Notes in Computer Science. Springer-Verlag, Vol. 3562, Part 2, pp: 604-612, ISSN: 0302-9743.

1.7.3. Proyectos de investigación

La tesis doctoral ha sido desarrollada bajo los siguientes proyectos de investigación.

- Herramientas interactivas para el modelado, visualización, simulación y control de sistemas dinámicos. Referencia: DPI 2004-01804.
- Control de sistemas complejos en logística y producción de bienes y servicios. Referencia: S-0505/DPI/0391.
- Modelado, simulación y control basado en eventos. Referencia: DPI2007-61068.

Las labores principales realizadas en estos proyectos son las siguientes:

- Desarrollo de laboratorios virtuales para la enseñanza del control.
- Diseño de un enfoque homogéneo para agregar interactividad y visualización a simulaciones de ingeniería.
- Desarrollo de la librerías JIMC y el programa JIM server.
- Implementación del enlace local y remoto entre Matlab/Simulink y Easy Java Simulations.
- Implementación del enlace local entre Scilab y Sysquake con Easy Java Simulations.
- Desarrollo de un enfoque para construir laboratorios de control en red con MATLAB.
- Desarrollo del enfoque TrueTime-Easy Java Simulations para crear sistemas de control en tiempo real.
- Implementación de la librería Java TrueTime (JTT) para facilitar la creación de simulaciones pedagógicas de sistemas de control en tiempo real.
- Definición e implementación de un ambiente de experimentación para simulaciones.

La colaboración con otros grupos de investigación ha demandado las siguientes actividades:

- Programación en MATLAB de herramientas propias para aplicar técnicas de reconocimiento de patrones.

- Utilización de técnicas de reconocimiento de patrones tales como Máquinas de vectores soporte y transformada Wavelets.
- Aplicación del reconocimiento estructural de patrones para la búsqueda específica de señales.
- Clasificación de imágenes en el diagnóstico Scattering Thompson.
- Detección automática de diferentes anomalías en señales de fusión.

Capítulo 2

Conclusiones y Trabajos Futuros

A continuación se describen las principales conclusiones y las líneas futuras de investigación a las que ha dado lugar la tesis doctoral.

2.1. Conclusiones

Las tecnologías de la información y comunicación han tenido un gran impacto en la vida moderna, y la educación no es la excepción. Estas tecnologías han proporcionado muchos avances en un creciente número de campos, tales como Internet, servicios web, video conferencias, cursos basados en web, gráficos interactivos, y otros.

La educación en la ingeniería del control se ha visto también afectada de forma positiva con estos avances. Los laboratorios virtuales y remotos están siendo utilizados con mayor frecuencia para incrementar las posibilidades que tienen los estudiantes de interactuar con recursos simulados o reales. Esta interacción ofrece nuevos elementos de aprendizaje sin las típicas restricciones temporales y espaciales de los laboratorios tradicionales.

Aunque la comunidad científica ha realizado una gran cantidad de avances en los laboratorios basados en computadoras aún queda margen de mejora.

Por ejemplo, muchas simulaciones no permiten la interacción de usuario mientras la simulación se está ejecutando, lo cual fuerza a los estudiantes a esperar hasta el final de la simulación para experimentar con el sistema con un nuevo conjunto de parámetros. Otras simulaciones proporcionan sólo sencillos dibujos de algunas salidas del modelo para mostrar el comportamiento del sistema, lo cual puede dificultar el aprendizaje de

algunos estudiantes. Sin embargo, la potencia gráfica de los modernos computadores de hoy en día pueden ayudar al desarrollo de material educativo más natural e intuitivo.

Estas características interesantes pueden ser añadidas, como una capa de interfaz humana, a simulaciones de ingeniería de manera de facilitar el proceso de aprendizaje y reducir el tiempo requerido para una comprensión completa de los conceptos fundamentales.

Sin embargo, la creación de los laboratorios basados en computadores con interfaces humanas avanzadas no es una tarea sencilla. Muchas herramientas de ingeniería proporcionan excelentes librerías para construir simulaciones de ingeniería, pero éstas normalmente carecen de herramientas para agregar interfaces humanas avanzadas. Así, la creación de estos laboratorios puede demandar un gran esfuerzo por parte de los profesores, especialmente aquellos que no son expertos en programación de computadores.

Por otra parte, una vez que finalmente un profesor crea una interfaz gráfica de usuario para su simulación, éste podría sentirse frustrado por el hecho de que las interfaces son normalmente incompatibles entre diferentes software de ingeniería.

Esta tesis doctoral se ha centrado en estos problemas, y las principales contribuciones consisten en nuevas herramientas de autor para facilitar la creación de laboratorios virtuales y remotos.

De manera de generar una forma estandarizada para agregar interfaces humanas a modelos existentes, el Capítulo 2 introduce un novedoso enfoque de diseño para crear simulaciones interactivas de ingeniería. El enfoque, denominado **interoperate approach**, divide la creación de laboratorios basados en computador en dos actividades separadas.

Primero, el instructor desarrolla la simulación de ingeniería utilizando un software de ingeniería estándar. Luego, el instructor usa un lenguaje de programación, como Java, para crear la interfaz humana interactiva. Ambos componentes son entonces integrados utilizando un protocolo de comunicación genérico para manipular la simulación de ingeniería desde la interfaz humana.

La tesis describe el protocolo de comunicación requerido para soportar el enfoque **interoperate approach**. El protocolo tiene dos niveles de comunicación para manipular la aplicación externa (la simulación de ingeniería) desde la aplicación cliente (la interfaz humana). El protocolo de nivel alto ofrece a los autores el control de la aplicación externa

en un sencillo, y alto nivel de abstracción, pero aún proporcionando un enlace efectivo entre el modelo de ingeniería y la interfaz humana de la simulación. El protocolo de bajo nivel da a los autores un control total de la aplicación externa, proporcionando un enlace amplio entre ambas partes que puede utilizarse para obtener un alto grado de interactividad y visualización.

El protocolo de alto nivel es comúnmente recomendado para la mayoría de los autores, especialmente para aquellos que no sean expertos programadores o no requieran de un control fino de la comunicación entre la aplicación cliente y la externa. El protocolo de bajo nivel debe ser la opción preferida por los autores que necesiten un control total en todos los pasos de comunicación. Sin embargo, el uso del protocolo de bajo nivel exige un esfuerzo de programación mayor que el requerido por el protocolo de alto nivel.

El protocolo de comunicación permite además el control de simulaciones de ingeniería a través de la red. Así, el enfoque **interoperate approach** puede ser utilizado para crear laboratorios locales y remotos de una forma unificada. De hecho, el estudiante que utiliza un laboratorio virtual diseñado con el enfoque **interoperate approach**, no observa ninguna diferencia entre una versión local y una remota de la misma simulación interactiva, exceptuando los retardos inherentes de la red. La existencia de tales retardos incidió en el desarrollo de dos tipos de enlace remoto para el protocolo de comunicación de alto nivel: Los enlaces remotos síncrono y asíncrono.

En el Capítulo 3 se describe la implementación del enfoque **interoperate approach** para una variedad de software de ingeniería. Las implementaciones siguen un esquema similar para todos los casos dado que la mayoría del software de ingeniería usado proporciona una interfaz para ser manipulado desde un lenguaje externo como C o Java. Mediante esta interfaz, el software de ingeniería es controlado desde una clase Java.

Las clases Java implementadas en esta tesis permiten a los autores manipular aplicaciones como MATLAB y Scilab de acuerdo al protocolo de comunicación diseñado previamente. Debido a que el control de las simulaciones de ingeniería esta ahora estandarizado, las interfaces humanas desarrolladas para una simulación de MATLAB pueden ser reutilizadas sin ninguna modificación por una simulación similar de Scilab.

Algunas de las clases Java implementadas son parte de la librería de código abierto **JIMC**. Los instructores pueden hacer uso de esta librería para crear laboratorios virtuales y remotos utilizando el software MATLAB/Simulink. El servidor de código abierto **JIM**

puede ser además utilizado para soportar la interacción remota con las simulaciones diseñadas en MATLAB/Simulink.

Una vez que la manipulación del software de ingeniería es definida de acuerdo al enfoque **interoperate approach**, los autores pueden hacer uso de librerías Java, tales como **Swing** y **AWT**, para crear la interfaz humana interactiva del laboratorio. Típicamente la construcción de una interfaz gráfica de usuario demanda un gran esfuerzo desde el punto de vista la implementación, especialmente para aquellos instructores inexpertos en programación. Por esta razón, los profesores pueden utilizar algunas herramientas de autor en Java, tales como Easy Java Simulations (EJS), para construir de forma sencilla la interfaz de usuario interactiva.

La creación de interfaces de usuario y el uso del enfoque **interoperate approach** desde EJS es descrito en el Capítulo 4. Este capítulo es dividido en dos partes. La primera parte describe el uso de una librería Java, como JIMC, para establecer la comunicación con un software de ingeniería desde EJS. La segunda parte presenta una nueva versión de EJS que integra todas las clases Java descritas en el Capítulo 3. Los instructores pueden hacer uso de esta versión de EJS para desarrollar, de forma aún mucho más sencilla que antes, los laboratorios interactivos con simulaciones diseñadas en MATLAB, Simulink, Scilab o Sysquake. Ambas partes del capítulo proporcionan ejemplos sencillos del uso de EJS bajo el enfoque **interoperate approach**. El capítulo finaliza con un ejemplo real de un laboratorio de control en red utilizado en un curso de ingeniería de control en la Universidad de Gante en Bélgica.

Luego de la descripción de como añadir interfaces humanas a cualquier simulación de ingeniería, la tesis considera en detalle la simulación de sistemas de control empotrado. Estos sistemas están sujetos a un reciente interés debido a un novedoso enfoque de análisis que considera, contrariamente al enfoque tradicional, los aspectos de tiempo real y de control en forma conjunta. Esta nueva perspectiva proporciona un caso de estudio perfecto para el enfoque **interoperate approach** gracias a la librería TrueTime basada en MATLAB/Simulink.

El Capítulo 5 discute la creación de laboratorios virtuales de sistemas de control empotrado. El capítulo está dividido en dos partes. La primera describe la creación de simulaciones interactivas que utilizan el enfoque **interoperate approach**. Los profesores pueden crear simulaciones de ingeniería de sistemas de control en tiempo real

mediante TrueTime, y luego utilizar EJS para agregar la interfaz humana interactiva, tal como lo describió el Capítulo 4. La segunda parte del Capítulo 5 muestra la implementación de la librería Java de código abierto , y basada en TrueTime, denominada **JTT**. El principal objetivo de esta librería es proporcionar a los instructores una solución gratuita para desarrollar simulaciones interactivas de sistemas de control en tiempo real dirigidos a la enseñanza. Ambas partes del Capítulo 5 muestran varios ejemplos de uso de las herramientas mencionadas.

La tesis finalmente considera un tercer tópico, esto es la creación de experimentos en laboratorios virtuales. Los experimentos pueden ser usados, por ejemplo, para optimizar parámetros del modelo mediante la ejecución de varias instancias de la simulación. Este problema ha sido considerado desde los inicios de las simulaciones mediante computadores. Sin embargo, la creación de experimentos ha evolucionado muy poco desde entonces. La tesis propone explotar las capacidades de los lenguajes modernos de programación para llevar a cabo experimentos educacionales utilizando simulaciones.

El Capítulo 6 describe un conjunto de elementos requeridos para ejecutar experimentos avanzados. Una implementación de un ambiente de experimentación básico es también descrita en detalle en el capítulo. Este ambiente de desarrollo de experimentos ha sido implementado en una versión de Easy Java Simulations. Los instructores pueden crear experimentos en una nueva sección de EJS llamada **Experimentos**.

Algunos ejemplos de uso son presentados para destacar las principales características del ambiente de experimentación propuesto. Desde el punto de vista educacional, los experimentos pueden ser utilizados, por ejemplo, para comparar en tiempo de ejecución las salidas de simulaciones ejecutadas bajo condiciones iniciales diferentes.

2.2. Trabajos Futuros

Aunque este trabajo ha proporcionado muchos resultados importantes, aún hay un espacio para nuevas mejoras. Los trabajos futuros son posibles en los tres tópicos tratados en la tesis.

El enfoque **interoperate approach** ha probado su validez para cuatro diferentes software de ingeniería. El esquema utilizado en el Capítulo 3 puede ser fácilmente seguido para implementar el protocolo de comunicación para otras aplicaciones externas tales

como Octave, Maple, y Dymola. De hecho, mucho del software de ingeniería presenta modernas interfaces externas que pueden proporcionar un acceso desde programas Java. Esto podría ser interesante para implementar un protocolo de comunicación dedicado para el paquete Scicos de Scilab. Scicos es un simulador de diagramas de bloques similar a Simulink. Así, basado en la experiencia con simulaciones Simulink, un enlace especial podría ser desarrollado también para simulaciones diseñadas en Scicos.

Aunque la implementación del protocolo de comunicación ha sido desarrollada en lenguaje Java, podría ser interesante probar la validez del enfoque **interoperate approach** utilizando otro lenguaje de propósito general tal como C, C++, o C#.

Con respecto a la operación remota del protocolo de comunicación, en esta tesis sólo se ha tratado el desarrollo de los enlaces remotos para MATLAB y Simulink. Otro software de ingeniería, tal como Scilab y Sysquake, podría también soportar una implementación de un enlace remoto de manera similar al esquema descrito en este trabajo.

El laboratorio de control en red descrito en el Capítulo 4 presenta un ejemplo real de un laboratorio remoto. Aún cuando este ejemplo muestra que este tipo de aplicaciones pueden ser creadas utilizando el protocolo de comunicación descrito, todavía se pueden realizar muchas mejoras. Por ejemplo, utilizando un protocolo de red diferenciado (TCP/IP o UDP) para tratar los datos que entran y salen del laboratorio remoto.

En redes de bajo ancho de banda o redes congestionadas, el uso del protocolo TCP/IP puede agregar retardos innecesarios al tráfico de datos que provienen de la planta real. Tales retardos se deben principalmente a los mecanismos de control de flujo y congestión del protocolo TCP/IP. Al contrario, el protocolo UDP puede proporcionar un intercambio más rápido de datos entre el cliente y el servidor, pero a expensas de perder la confiabilidad en la transmisión que el control de flujo ofrece.

Por una parte, el flujo de datos proveniente del servidor remoto transporta datos reales desde la planta. Estos datos son principalmente utilizados por la interfaz de usuario para monitorear el estado de la planta real. Así, en vez de utilizar el protocolo TCP/IP para transportar este tipo de información, el protocolo de comunicación podría usar el protocolo UDP para mejorar el rendimiento del laboratorio remoto.

Por otra parte, el flujo de datos que viaja desde la interfaz de usuario hacia el servidor remoto transporta principalmente datos de control. Esta información puede ser utilizada para modificar los parámetros del controlador ó para enviar la acción de control a la

planta real. Así, la selección del protocolo TCP/IP o el protocolo UDP debería ser hecha considerando si el control de la planta real depende o no de datos sin errores.

El enfoque **interoperate approach** podría ser extendido además para ser usado en otras plataformas tales como los dispositivos móviles, de manera de aprovechar la capacidad computacional de los teléfonos móviles o las nuevas experiencias de usuario proporcionadas por los dispositivos táctiles.

Con respecto a la simulación de los sistemas de control empotrado o en tiempo real, el rendimiento de los enfoques presentados en esta tesis pueden ser optimizados.

El enfoque que combina TrueTime y EJS aún requiere de un gran número de evaluaciones de funciones de cruce por cero para simular los sistemas de control en tiempo real en Simulink. Así, podrían agregarse nuevos métodos al protocolo de comunicación que manipula los modelos Simulink de forma de reducir la cantidad de eventos y por tanto mejorar el rendimiento de la simulación.

Además, el protocolo de comunicación podría considerar un enlace dedicado con la implementación en C++ de TrueTime con el objetivo de acelerar las simulaciones. La implementación de TrueTime sobre Scicos puede también ser analizada con la finalidad de proporcionar una plataforma completa de código abierto para la creación de simulaciones de control en tiempo real.

El enfoque que combina JTT y EJS puede ser extendido para soportar sistemas que simulan comunicaciones por red. Otras funcionalidades, tales como el bloque de batería de TrueTime podría también ser añadido a la librería JTT.

Algunos ejemplos de uso de la funcionalidad para simular sistemas de tiempo real no estrictos podrían ser considerados en el futuro. JTT también podría ser utilizado con la colección de *drivers* de código abierto del proyecto Comedi con el fin de ofrecer una plataforma de código abierto para el desarrollo de laboratorios remotos.

Ambos enfoques, TrueTime y JTT, pueden ser además considerados para generar nuevos laboratorios virtuales de sistemas de control en tiempo real que se utilicen directamente en un curso de control.

El trabajo futuro de experimentos en laboratorios virtuales involucrará principalmente el desarrollo de nuevas aplicaciones.

El ambiente de experimentación podría ser además extendido para crear experimentos en laboratorios remotos. Esta característica es interesante desde un punto de vista

educacional para mostrar los efectos de una selección incorrecta de parámetros de un controlador.

Los experimentos en laboratorios remotos pueden ser también utilizados para crear algoritmos generalizados que identifiquen el modelo de una planta real.

La combinación del ambiente de experimentación con el enfoque **interoperate approach** y la simulación de sistemas de control en tiempo real puede ser una extensión interesante de llevar a cabo.

Finalmente, en un contexto más general, la API de experimentación descrita puede ser la base para la definición de un lenguaje de experimentación estándar que otras herramientas de simulación y modelado podrían adoptar.

Bibliografía

- Chandrasekara, C. & Davari, A. (2004), Inverted pendulum: an experiment for control laboratory, *in* ‘Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory’, pp. 570 – 573.
- Dormido, R., Vargas, H., Duro, N., Sánchez, J., Dormido-Canto, S., Farias, G., Esquembre, F. & Dormido, S. (2008), ‘Development of a web-based control laboratory for automation technicians: The three-tank system’, *IEEE Transactions on Education* **51**(1), 35–44.
- Dormido, S. (2002), ‘Control learning: Present and future’, *IFAC Annual Control Reviews* **28**, 115–136.
- Dormido, S., Dormido-Canto, S., Dormido, R., Sánchez, J. & Duro, N. (2005), ‘The role of interactivity in control learning’, *The International Journal of Engineering Education: Especial issue on Control Engineering Education* **21**(6), 1122–1133.
- Dormido, S., Esquembre, F., Farias, G. & Sánchez, J. (2005), Adding interactivity to existing simulink models using easy java simulations, *in* ‘Proceedings 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC’05)’, pp. 4163–4168.
- Esquembre, F., Dormido, S. & Farias, G. (2007), Defining and performing experiments in virtual laboratories, *in* ‘Proceedings of the 10th International Conference on Engineering Education, ICEE2007’.
- Fabregas, E., Farias, G., Dormido-Canto, S., Dormido, S. & Esquembre, F. (2010), ‘A practical approach for remote interaction with a real plant’, *Submitted to Computer & Education* .

- Farias, G., Årzén, K. & Cervin, A. (2007), Interactive real-time control labs with true-time and easy java simulations, *in* ‘Proceedings of the International Multiconference on Computer Science and Information Technology, International Workshop on Real Time Software, IMCSIT2007’, pp. 811–820.
- Farias, G., Årzén, K., Cervin, A., Dormido, S. & Esquembre, F. (2009), ‘Teaching embedded control systems’, *International Journal of Engineering Education* . Submitted.
- Farias, G., Cervin, A., Årzén, K., Dormido, S. & Esquembre, F. (2008), Multitasking real-time control systems in easy java simulations, *in* ‘Proceedings of the 17th IFAC World Congress’.
- Farias, G., Cervin, A., Årzén, K., Dormido, S. & Esquembre, F. (2009), ‘Java simulations of embedded control systems’, *Real-Time Systems* . Submitted.
- Farias, G., Keyser, R. D., Dormido, S. & Esquembre, F. (2009), Building remote labs using easy java simulation and matlab, *in* ‘Proceedings of the 10th European Control Conference’.
- Farias, G., Keyser, R. D., Dormido, S. & Esquembre, F. (2010), ‘Developing networked control labs: A matlab and easy java simulations approach’, *IEEE Transactions on Industrial Electronics* . DOI:10.1109/TIE.2010.2041130.
- Gillet, D., El Helou, S., Yu, C. M. & Salzmänn, C. (2008), Turning web 2.0 social software into versatile collaborative learning solutions, *in* ‘Advances in Computer-Human Interaction’, pp. 170 –176.
- Gomes, L. & Bogosyan, S. (2009), ‘Current trends in remote laboratories’, *IEEE Transactions on Industrial Electronics* **56**(12), 4744–4756.
- Heck, B. (1999), ‘Special report: Future directions in control education’, *IEEE Control Systems Magazine* **19**(5), 35–58.
- Latchman, H., Salzmänn, C., Gillet, D. & Bouzekri, H. (1999), ‘Information technology enhanced learning in distance and conventional education’, *Education, IEEE Transactions on* **42**(4), 247 –254.
- Latchman, H., Salzmänn, C., Gillet, D. & Kim, J. (2001), ‘Learning on demand—a hybrid synchronous/asynchronous approach’, *Education, IEEE Transactions on* **44**(2), 17 pp.

- Leva, A. (2003), ‘A hands-on experimental laboratory for undergraduate courses in automatic control’, *Education, IEEE Transactions on* **46**(2), 263 – 272.
- Leva, A. (2004), An experimental laboratory on control structures, *in* ‘American Control Conference, 2004. Proceedings of the 2004’, Vol. 4, pp. 3227 – 3232.
- Leva, A. & Donida, F. (2008), ‘Multifunctional remote laboratory for education in automatic control: The crautolab experience’, *Industrial Electronics, IEEE Transactions on* **55**(6), 2376 – 2385.
- M. Ohlin, D. H. & Cervin, A. (2007), *TrueTime 1.5 Reference Manual*, Department of Automatic Control.
- Radharamanan, R. & Jenkins, H. (2007), Robot applications in laboratory-learning environment, *in* ‘Proceedings of the Thirty-Ninth Southeastern Symposium on System Theory’, pp. 80 – 84.
- Sánchez, J. (2001), Un nuevo enfoque metodológico para la enseñanza a distancia de asignaturas experimentales. Análisis, diseño y desarrollo de un laboratorio virtual y remoto para el estudio de la automática a través de Internet, PhD thesis, Universidad Nacional de Educación a Distancia (UNED).
- Sánchez, J., Dormido, S. & Esquembre, F. (2005), ‘The learning of control concepts using interactive tools’, *Computer Applications in Engineering Education* **13**(1), 84–98.
- Sánchez, J., Dormido, S., Pastor, R. & Esquembre, F. (2004), Interactive learning of control concepts using easy java simulations, *in* ‘Proceedings of the IFAC Workshop Internet Based Control Education IBCE’04’.
- Sánchez, J., Morilla, F., Dormido, S., Aranda, J. & Ruiperez, P. (2002), ‘Virtual and remote control labs using java: a qualitative approach’, *Control Systems Magazine, IEEE* **22**(2), 8 – 20.
- SIIA (2001), Trends report 2001: Trends shaping the digital economy, Technical report, The software & Information Industry Association.
- Spong, M. & Block, D. (1995), The pendubot: a mechatronic system for control research and education, *in* ‘Proceedings of the 34th IEEE Conference on Decision and Control’, pp. 555 – 556.

Uran, S. & Jezernik, K. (2008), ‘Virtual laboratory for creative control design experiments’, *Education, IEEE Transactions on* **51**(1), 69 –75.

Wellstead, P. E. (1983), ‘The ball and hoop system’, *Automatica* **19**(4), 401 – 406.

Wu, M., She, J.-H., Zeng, G.-X. & Ohyama, Y. (2008), ‘Internet-based teaching and experiment system for control engineering course’, *Industrial Electronics, IEEE Transactions on* **55**(6), 2386 –2396.