



**Universidad de Almería**

**Titulación de Ingeniero en Informática**

Herramientas interactivas basadas  
en tecnología web: aplicación al  
campo de la automática como  
caso de estudio

**Autor:**

Juan Carlos González Rodríguez

**Tutor:**

José Luis Guzmán Sánchez

Almería, febrero 2012

## **AGRADECIMIENTOS**

Deseo mostrar mi agradecimiento a mis padres, mi hermano y mi cuñada por el apoyo incansable que me han ofrecido durante el tiempo que ha durado la realización de este proyecto. También desearía dar las gracias a mi tutor José Luis Guzmán Sánchez por su dirección y orientación, y por la oportunidad que me ha brindado de realizar este proyecto que ha sido una gran motivación.

Por último, quiero dar las gracias a mis compañeros y profesores de la carrera de Ingeniería Informática de la Universidad de Almería. De forma destacada me gustaría mencionar a Ignacio Fernández Sedano, Carlos Rodríguez Contreras y Eugenio Eduardo Villar Fernández



## RESUMEN

Actualmente, en el campo de la educación hay nuevos métodos y técnicas en el aprendizaje y la enseñanza fundamentados en las nuevas tecnologías de la información y la comunicación. Estos métodos permiten a los educadores encontrar técnicas innovadoras para mejorar sus tareas en la enseñanza: herramientas multimedia, sistemas de hipertexto, sistemas interactivos, intercambio de información entre educador y estudiante a través de Internet, acceso a información desde cualquier punto del mundo sin restricciones de tiempo, etc., donde la gran revolución se produjo con la World Wide Web (WWW). Estas nuevas técnicas y métodos han mostrado su influencia, ventajas y desventajas que las nuevas tecnologías de la información y la comunicación han ofrecido a la educación. Sin embargo, dentro del campo de la ingeniería una de las técnicas con mayor impacto fue la aparición de las herramientas interactivas. Aunque la interactividad puede ser definida desde varios puntos de vista, desde el punto de vista de la educación se puede definir como una colección de ventanas gráficas con componentes activos, dinámicos y/o interactivos, los cuales buscan facilitar la comprensión de algunos conceptos. Las herramientas interactivas se han establecido como un complemento excelente para la enseñanza en la ingeniería. Estas nuevas herramientas ayudan a los estudiantes en el aprendizaje de conceptos complejos. Una amplia variedad de herramientas interactivas han sido creadas usando entornos como Easy Java Simulations y Sysquake, presentándose como soporte a la enseñanza de control mostrando conceptos con diferentes niveles de complejidad. Estas herramientas son muy potentes pero están limitadas al ser usadas en cualquier tipo de dispositivos, ya que suelen requerir ficheros ejecutables o máquinas virtuales para que sean ejecutadas. Teniendo en cuenta que el uso de los teléfonos inteligentes y tabletas se está incrementando cada día y está siendo bastante habitual que los estudiantes accedan al material educativo desde estos dispositivos,, es necesario por tanto evaluar nuevas alternativas que permitan el desarrollo de herramientas interactivas para hacer frente a esta desventaja. Este proyecto presenta una solución basada en estándares web, específicamente los de World Wide Web Consortium (W3C). Los estándares W3C definen una plataforma web abierta para el desarrollo de aplicaciones que tienen el potencial sin precedentes de permitir a los desarrolladores construir experiencias ricas en interactividad, impulsadas el manejo de sistemas con gran cantidad de datos, y que son accesibles desde cualquier dispositivo con un navegador web. Este proyecto presenta una solución que consiste en una librería, la cual permite que las herramientas interactivas creadas puedan ser usadas en cualquier dispositivo que contenga un navegador web que implemente estos estándares y permitiendo también la integración de las herramientas con el contenido de cualquier página web. También se presentan en este proyecto varios ejemplos de las capacidades de la librería, mediante la creación de herramientas interactivas de sistemas de control básicos: sistema de primer orden, sistema de segundo orden, efecto de un cero sobre un sistema y un controlador PID. Por otra parte, se muestran otros pequeños estudios como la posibilidad de calcular los datos de los procesos a través de internet o la posibilidad de crear una interfaz gráfica de usuario que utiliza la librería para crear herramientas interactivas.



## ABSTRACT

Currently, in the education field, there are new methods and techniques in learning and teaching based on the new information and communication technologies. These methods allow educators to find innovative techniques to improve their teaching tasks: multimedia tools, hypertext systems, interactive systems, information exchange between teacher and student over the Internet, access information from anywhere in the world without time restrictions, etc., where the greatest revolution was produced by the World Wide Web (WWW). These new techniques and methods have shown the influence, advantages and disadvantages that the new information and communication technologies have offered to education. However, from an engineering education point of view, the biggest impact was with the appearance of interactive tools. Although interactivity can be defined from several points of view, from an education point of view it can be defined as a collection of graphical windows with active, dynamic and/or interactive components, which are intended to explain a few concepts. Interactive tools have been established as an excellent complement to control education. These new tools helping students in learning complex concepts. A wide variety of different interactive tools have been created using environments as Easy Java Simulations and Sysquake. These tools have been presented as a greater support for control education showing concepts with distinct levels of complexity. These tools are very powerful but they are limited to be used in any type of devices, since they require executable files or virtual machines to run. Notice that the use of Smartphones and Tablets is increasing everyday and it is being quite usual that students access to educational material from such devices. Thus, it is necessary to evaluate new alternatives allowing to develop interactive tools to face this disadvantage. This project presents a solution based on web standards, specifically the World Wide Web Consortium (W3C). W3C standards define an Open Web Platform for application development that has the unprecedented potential to enable developers to build rich interactive experiences, powered by vast data stores, that are available on any device having a web browser. Thus, this project presents a solution based on a library, which allows the interactive tools created to be used in any device that contains a web browser and implements those standards, and also allowing the integration of interactive tools with the content of any website. In addition, this project presents several examples of the captivity of the library, by means of the creation of tools for basic control systems: first-order system, second-order system, zero effect, and a PID-based control. On the other hand, it is also briefly shown the possibility of performing the process data calculation through the Internet and the possibility to create a graphical user interface that uses the library to create graphically interactive tools.



---

**INDICE GENERAL**

Agradecimientos .....	1
Resumen.....	3
Abstract .....	5
Indice General .....	7
1 Introducción .....	13
1.1 Contexto del proyecto.....	13
1.1.1 Clasificación general de las NTIC'S .....	14
1.1.2 Clasificación de las NTIC'S en la docencia .....	14
1.1.3 NTIC'S en ingeniería .....	15
1.1.4 Herramientas interactivas.....	16
1.2 Motivación del desarrollo del proyecto .....	18
1.3 Objetivos del proyecto .....	19
1.4 Estructura de la memoria del proyecto fin de carrera.....	21
2 Interactividad .....	23
2.1 Definiciones y clasificaciones .....	23
2.2 Interactividad en la enseñanza .....	26
2.3 Enseñanza del Control Automático basada en la Interactividad .....	28
3 Modelado y control PID .....	31
3.1 Sistema, modelo y simulación.....	31
3.2 Función de transferencia de un sistema dinámico .....	32
3.3 Comportamiento de sistemas dinámicos.....	32
3.3.1 Señales de prueba .....	32
3.3.2 Sistemas de primer orden .....	33
3.3.3 Sistemas de segundo orden .....	35
3.3.4 Análisis dinámico de respuestas subamortiguadas.....	37



## INDICE GENERAL

---

3.3.5	Estabilidad absoluta y relativa.....	38
3.3.6	Efecto de ceros sobre la respuesta .....	39
3.4	Sistemas de control .....	40
3.4.1	Concepto de sistema de control .....	40
3.4.2	Funcionamiento de un sistema de control.....	40
3.4.3	Elementos de un sistema de control.....	41
3.5	Controladores automáticos.....	42
3.5.1	Concepto de controlador automático .....	42
3.5.2	Acciones básicas de control .....	42
	Sistemas de control .....	43
	Acciones de control .....	43
	Discontinuos.....	43
	Control Todo/Nada.....	43
	Control de varias posiciones .....	43
	Continuos .....	43
	Control Proporcional .....	43
	Control Proporcional Integral.....	43
	Control Proporcional Derivativo.....	43
	Control Proporcional Integral Derivativo .....	43
3.5.3	Control Proporcional .....	43
3.5.4	Control Proporcional Integral.....	46
3.5.5	Control Proporcional Derivativo.....	49
3.5.6	Control Proporcional Integral Derivativo .....	51
3.6	Introducción a los sistemas de control en tiempo discreto .....	52
3.7	Descripción de sistemas dinámicos en tiempo discreto .....	54
3.7.1	Ecuación en diferencias.....	54
3.7.2	Transformada z y función de transferencia .....	54
3.7.3	Función de transferencia de sistemas con mantenedor .....	55

---

---

3.8	Respuesta temporal de sistemas en tiempo discreto.....	56
3.8.1	Respuesta temporal de sistemas de primer orden.....	56
3.8.2	Respuesta temporal de sistemas de segundo orden.....	56
3.9	Diseño de controladores en tiempo discreto por conversión de controladores continuos.....	57
3.9.1	Consideraciones generales en la discretización de controladores continuos.....	57
3.9.2	Discretización de la operación integral.....	59
3.9.3	Discretización de la operación derivada.....	60
3.9.4	Discretización de controladores PID.....	62
4	Tecnologías basadas en Web.....	65
4.1	HTML, XHTML y HTML5.....	65
4.1.1	Que es HTML.....	65
4.1.2	Historia de HTML.....	66
4.1.3	Que es XHTML.....	66
4.1.4	Historia de XHTML.....	67
4.1.5	Que es HTML5.....	67
4.1.6	Historia de HTML5.....	67
4.1.7	Mejoras HTML5.....	68
4.1.8	Uso de HTML, XHTML y HTML5 en el proyecto.....	72
4.2	Scripting y DOM HTML.....	75
4.2.1	Que es el Scripting.....	75
4.2.2	DOM HTML.....	75
4.2.3	Uso de Scripting y DOM HTML en el proyecto.....	75
4.3	Gráficos.....	83
4.3.1	SVG.....	84
4.3.2	La API de Canvas.....	84
4.3.3	CSS.....	85
4.3.4	SVG o API Canvas.....	85

---

## INDICE GENERAL

---

4.3.5	Uso de SVG en el proyecto .....	86
4.3.6	Uso de CSS en el proyecto.....	87
4.4	MahtML.....	88
4.4.1	Uso de MathML en el proyecto.....	89
5	Herramientas desarrolladas .....	93
5.1	Análisis y diseño .....	93
5.2	Descripción de clases .....	94
5.2.1	Data .....	94
5.2.2	FirstOrderSystem.....	94
5.2.3	Frame .....	95
5.2.4	Graph.....	95
5.2.5	GraphLegend .....	96
5.2.6	Grid.....	96
5.2.7	HScale .....	96
5.2.8	IButton.....	97
5.2.9	IControl.....	97
5.2.10	ICScaleButton .....	97
5.2.11	IData .....	97
5.2.12	IDot.....	97
5.2.13	IElement .....	98
5.2.14	IHLine.....	98
5.2.15	ILine .....	98
5.2.16	Info .....	98
5.2.17	IValue.....	99
5.2.18	IVLine.....	99
5.2.19	MenuBar.....	99
5.2.20	PIDFOS.....	99
5.2.21	PlaneS.....	100

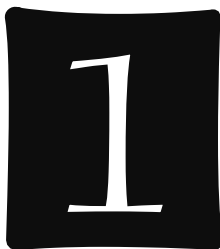
---

5.2.22	Point .....	100
5.2.23	Property.....	100
5.2.24	SecondOrderSystem.....	100
5.2.25	SecondOrderSystemCero .....	101
5.2.26	Step.....	101
5.2.27	System.....	101
5.2.28	Scale .....	101
5.2.29	VScale .....	101
5.3	Diagrama de clases.....	102
5.4	Herramientas desarrolladas.....	103
5.4.1	Sistema de primer orden.....	103
5.4.2	Sistema de primer orden con retardo.....	108
5.4.3	Sistema de segundo orden.....	109
5.4.4	Sistema segundo orden con retardo .....	112
5.4.5	Sistema de segundo orden con el efecto de un cero.....	114
5.4.6	Control PID .....	115
5.4.7	Página web y pruebas .....	117
5.5	Python y AJAX.....	119
5.6	Interfaz gráfica para crear herramientas interactivas.....	120
5.6.1	Descripción de clases .....	120
	IMoveButton .....	121
5.6.2	PropertiesButton.....	121
5.6.3	PropertiesMenu .....	121
5.6.4	SelectList.....	121
5.6.5	SelectListFirstOrderSystem .....	122
5.6.6	ToolBar .....	122
5.6.7	ToolBarItem.....	122
5.6.8	ToolBarItemFirstOrderSystem.....	122

## INDICE GENERAL

---

5.6.9	ToolBarItemStep.....	123
6	Conclusiones y trabajos futuros.....	125
	Referencias.....	127
	Anexo .....	131
	Plantilla CSS MathML .....	131
	Código Python .....	141
	Matlab.py .....	141
	py2xml.py .....	141



## INTRODUCCIÓN

Este capítulo es introductorio al proyecto, en él se explica el contexto del proyecto, la motivación para realizarlo y los objetivos del mismo este.

### 1.1 Contexto del proyecto

La Universidad actual se enfrenta continuamente a nuevos retos, muchos de ellos relacionados con los nuevos tipos de estudiantes, las nuevas tecnologías y las nuevas ideas sobre la enseñanza, el aprendizaje y el significado de la educación universitaria. La informatización es un fenómeno mundial que ha llegado a la Universidad por estar ésta a la vanguardia de las últimas innovaciones científico-técnicas. Actualmente, se puede ver cómo la informática se utiliza en los servicios administrativos (por ejemplo matriculación), servicios de apoyo a la investigación (las hemerotecas, por poner un ejemplo) y últimamente en el proceso de enseñanza (tutorías en línea, cursos virtuales, etc.). La informatización de la Universidad, con la creación de espacios y servicios virtuales a través de la red, pretende mejorar y optimizar el conjunto de sus actividades y funciones. Este primer enfoque necesita diferenciarse de la Universidad Virtual, en la que todos los espacios y todas las comunicaciones entre seres humanos están mediadas por las redes y los computadores.

Los computadores son una herramienta muy presente en las escuelas y universidades de los países desarrollados y se van introduciendo muy rápidamente en las aulas de los países en vías de desarrollo. Se han creado muchas herramientas de aprendizaje basadas en computador, desde juegos didácticos hasta software de enseñanza asistida por computador y software de ayuda para el profesor. Los centros educativos están conectados a Internet e incluso los alumnos de las zonas más remotas tienen acceso a la creciente cantidad de información que antes sólo estaba al alcance de aquellas poblaciones próximas a las bibliotecas municipales y universitarias más completas. Mediante Internet, profesores y alumnos tienen acceso a material curricular, de formación de profesorado y otros materiales de aprendizaje, algunos provistos por sus propias administraciones centrales o estatales y otros suministrados por proveedores privados. Las tecnologías de la información y la comunicación (TIC) se utilizan ampliamente en la enseñanza a distancia y sustituyen a la antigua escuela por correspondencia y a la televisión educativa. En general, la nueva educación a distancia se realiza mediante Internet.

Esta emergente sociedad de la información, sustentada por el uso generalizado de las potentes y versátiles TIC, conlleva cambios que alcanzan todos los ámbitos de la actividad humana. Sus efectos se manifiestan, de manera muy especial, en las actividades laborales y en el mundo educativo, donde todo debe ser revisado: desde la razón de ser de la escuela y demás instituciones educativas, hasta la formación básica que precisamos las personas, la forma de enseñar y de aprender, las infraestructuras y los medios que utilizamos para ello, la estructura organizativa de los centros y su cultura, el rol del profesor, el rol del alumno,...

### 1.1.1 Clasificación general de las NTIC'S

Basándose en la clasificación realizada en [2], si se atiende a los tipos de herramientas existentes, las nuevas tecnologías de la información y la comunicación (nuevas TICs para abreviar) más comunes que se pueden encontrar en la actualidad se pueden clasificar en [15]:

- **Programas informáticos.** Constituyen una herramienta que ayuda a la transformación de la enseñanza y del aprendizaje, porque encaminan, apoyan y complementan cualquier información.
- **Medios audiovisuales.** Bajo esta denominación se agrupan las tecnologías que permiten la realización de una comunicación a distancia, una interacción entre un emisor y un receptor que no están próximos físicamente.
- **Herramientas multimedia.** Se tratan de aplicaciones software que integran el uso de múltiples recursos (textos, sonido, imágenes, vídeo, etc.) dentro de un proceso de comunicación a través del ordenador. Las herramientas multimedia pueden tener diferentes usos, pero es en la instrucción donde tiene una mayor aplicabilidad; su efectividad depende de las relaciones que se establecen entre sus componentes y corresponde al diseñador del software anticipar los niveles de interacción que se han de suscitar. Al hablar de que los programas multimedia interactivos están diseñados en niveles, se hace referencia al hecho de que deben facilitar la navegación a través del programa y responder a las necesidades y características individuales de cada usuario. Así mismo, deben incorporar situaciones de interacción entre el programa y el usuario, de modo que las respuestas que emita éste sobre las actividades multimedia guiarán el desarrollo del programa.
- **Herramientas relacionadas con Internet.** Se refiere a la conexión de los ordenadores con una red de información por todo el mundo, y que pueden utilizarse para adquirir nuevos conocimientos o intercambiar información en diversos formatos como texto, imágenes, música, vídeo, etc.

### 1.1.2 Clasificación de las NTIC'S en la docencia

La clasificación anteriormente dada es igualmente válida dentro del campo de la docencia, pero se pueden resaltar algunos ejemplos de herramientas en los diferentes campos anteriormente citados [15].

- **Programas informáticos.** En el ámbito de la enseñanza, en este grupo se podría incluir cualquier simulador que ayude el aprendizaje del alumnado, o presentaciones en transparencias digitales que facilitan la tarea del docente.
- **Medios audiovisuales.** En el contexto educativo se podrían citar algunos ejemplos como la videoconferencia entre profesores y alumnos separados por grandes distancias, e incluso también se podría considerar a Internet como una herramienta audiovisual de comunicación a distancia.

- **Herramientas multimedia.** En este punto se podrían citar las enciclopedias virtuales que poco a poco están sustituyendo a las más clásicas de papel, los CD o DVD interactivos, los vídeos digitales, tutoriales interactivos que ayuden al alumno en su tarea de aprendizaje, entre otros.
- **Herramientas relacionadas con Internet.** Incluirían tanto las herramientas típicas de Internet, que pueden ser de utilidad para el alumno y al profesor, como programas de enseñanza virtual del tipo WebCT, páginas Web donde tanto el alumno como el docente puedan consultar información, el correo electrónico que ayuda a mantener una enseñanza a distancia, los grupos de noticias, el FTP que permite subir o bajar archivos relacionados con la materia, etc.

### 1.1.3 NTIC'S en ingeniería

Cuando se plantea el efecto que han tenido las TICs en el campo concreto de las titulaciones de Ingeniería, sería necesario ver dicho efecto desde dos puntos de vista: teórico y práctico [15].

Desde un punto de vista teórico, los avances docentes producidos por las TICs a nivel general son totalmente aplicables al campo de la ingeniería. De esta forma y según lo tratado en el punto anterior, se pueden considerar como posibles TICs para ingeniería desde un punto de vista teórico las siguientes:

- **Herramientas de simulación interactivas:** con el fin de mostrar ejemplos prácticos en clase teórica y hacer corresponder la teoría con la práctica.
- **Vídeo-Tutoriales:** vídeos autocontenidos que permiten recibir los contenidos típicos de una clase presencial pero sin restricciones espacio-temporales.
- **Material en línea (online):** sitio web de la asignatura con información complementaria, enlaces, herramientas, etc.
- **Transparencias digitales:** pero no como una mera exposición del texto o fórmulas del contenido teórico, sino como complemento a la teoría y para proporcionar cierto dinamismo. También cuando se utiliza como pizarra digital.

La enseñanza en la ingeniería se caracteriza por ser una disciplina con un fuerte contenido experimental, donde el alumno necesita poner en práctica los conocimientos teóricos adquiridos a lo largo del curso.

Esta componente práctica se ha llevado a cabo tradicionalmente en laboratorios de prácticas. Como es bien conocido, el disponer de un laboratorio bien equipado de elementos para impartir una materia tiene algunos inconvenientes desde el punto de vista temporal y económico: costes en la adquisición de equipos, costes de mantenimiento, coste de personal para puesta en funcionamiento, uso de los dispositivos únicamente en horas lectivas, etc.; y desde el punto de vista de los estudiantes: horario restringido, tiempos de espera cuando



existe material limitado, desplazamientos fuera de horario, etc. Estos aspectos muestran la necesidad de nuevos elementos que permitan cubrir las carencias existentes en los laboratorios tradicionales. Es aquí donde entran en juego los laboratorios virtuales y remotos [9],[30].

Los laboratorios virtuales son herramientas software locales o remotas que, mediante el uso de un modelo y junto con una interfaz de experimentación, simulan los principales aspectos de una planta real, permitiendo al usuario realizar las mismas operaciones que un laboratorio tradicional pero todo ello de forma virtual. Los laboratorios remotos son herramientas que permiten el acceso al equipamiento de un laboratorio real a través de una red. El usuario controla de forma remota sistemas físicos reales mediante una interfaz de experimentación que se encuentra conectada directamente a la planta real. De esta forma es posible explotar el rendimiento de los laboratorios las 24 horas de día, permitiendo una mayor flexibilidad horaria y un menor coste económico (necesitándose menos recursos, puesto que los existentes pueden ser usados durante más tiempo)

Actualmente, existen entornos de desarrollo que permiten implementar este tipo de herramientas de una forma bastante transparente. Ejemplos de tales entornos son Easy Java Simulations [10]y LabVIEW de National Instruments [18].

### 1.1.4 Herramientas interactivas

Otro avance importante en el apoyo a la docencia en Ingeniería ha sido la aparición de las herramientas interactivas. A la hora de estudiar un determinado sistema o comprender las leyes físicas que describen su conducta, los científicos e ingenieros suelen utilizar computadores para calcular y representar gráficamente diferentes magnitudes. Este aspecto queda perfectamente reflejado en el campo del Control Automático donde a menudo se utilizan: respuestas en los dominios del tiempo y la frecuencia, localización de polos y ceros en el plano complejo, diagramas de Bode, Nyquist y Nichols, lugar de las raíces, etc. Con frecuencia, estas magnitudes están relacionadas entre sí ya que constituyen diferentes visiones de una misma realidad. La comprensión de estas relaciones es una de las claves para lograr un buen aprendizaje de los conceptos básicos y permite al alumno estar en disposición de realizar diseños de sistemas de control automático con cierto sentido. A la hora de llevar a cabo el diseño de un sistema es necesario seguir un proceso iterativo compuesto de dos fases (ver

Figura 1). La primera consiste en determinar los principales parámetros del sistema basándose en las especificaciones del problema. La segunda consiste en evaluar los resultados obtenidos y compararlos con las especificaciones. En caso de no cumplir los requisitos, se repetirán de nuevo dichas etapas [9].

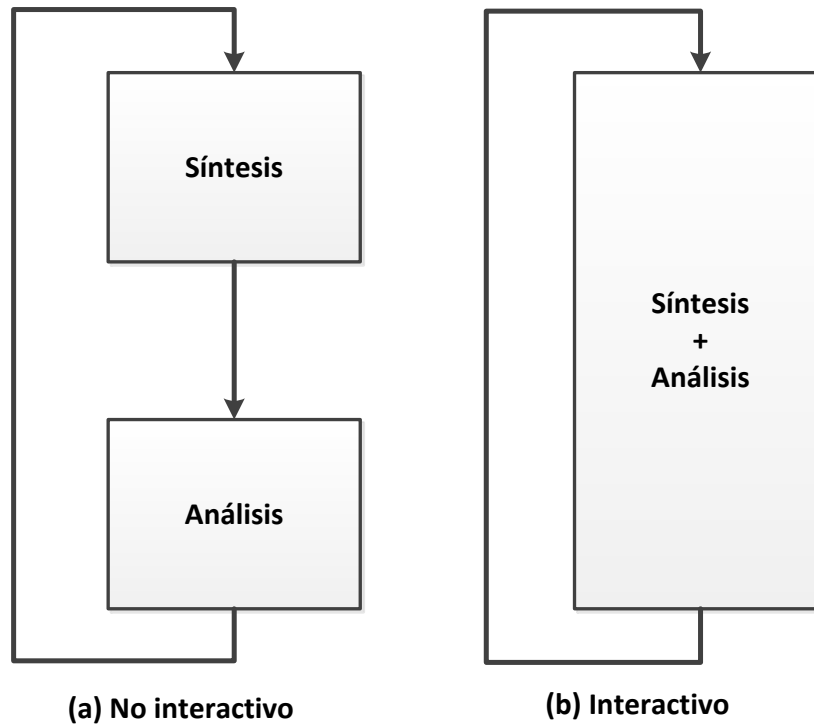


Figura 1. Paradigmas de diseño

En los últimos años se han desarrollado gran cantidad de herramientas basándose en las ideas y conceptos implementados por el Profesor Åström y colaboradores del Instituto Tecnológico de Lund. Estas ideas se basan en los conceptos de gráficas dinámicas y sistemas virtuales interactivos, que fueron introducidos por Wittenmark, donde el principal objetivo de estas herramientas es hacer más activos a los estudiantes aumentando su participación en los cursos de control. Estas nuevas herramientas están basadas en objetos que permiten una manipulación gráfica directa de forma que mientras un objeto está siendo modificado el resto de elementos son actualizados automáticamente, pudiéndose observar, en todo momento y de forma directa la relación existente entre todos ellos. Las herramientas Ictools y CCSdemo [19],[34], desarrollados en el Departamento de Control Automático en el Instituto Tecnológico de Lund, y Sysquake en el Instituto de Automática de la Escuela Politécnica de Lausanne [24], son buenos ejemplos de esta nueva filosofía docente para la enseñanza de control automático.

Gracias a este tipo de herramientas interactivas es posible realizar las fases de síntesis y análisis comentados anteriormente de forma conjunta (figura 1.b), de modo que cuando un parámetro es modificado su efecto se ve reflejado inmediatamente. De esta forma el proceso de diseño se hace realmente dinámico y el alumno percibe el gradiente del cambio del criterio de comportamiento con respecto a los elementos que manipula.

Como se puede ver más adelante este proyecto fin de carrera trata de seguir trabajando en estas líneas presentando el desarrollo de una librería para realizar este tipo de herramientas interactivas.

### 1.2 Motivación del desarrollo del proyecto

Las ideas del control automático son realmente ricas en contenido visual que puede representarse de forma intuitiva y geométrica. Estos contenidos visuales se pueden emplear tanto para presentar tareas como para manejar conceptos y métodos que permitan manipularlos en la resolución de los problemas.

Las ideas básicas del control automático surgen a menudo de situaciones visuales muy específicas y todos los expertos en control reconocen la gran utilidad que supone partir de estas concreciones cuando necesitan manejar los correspondientes objetos abstractos. La percepción humana es fundamentalmente visual y, por lo tanto, no debe sorprender que las ayudas mediante visualizaciones estén tan presentes en el trabajo docente. Con gran frecuencia se utilizan procesos simbólicos, diagramas visuales y otras formas de procesos imaginativos que permiten adquirir, lo que se podría llamar, una cierta intuición de lo abstracto. Estos aspectos intuitivos son probablemente mucho más difíciles de explicitar y asimilar por los estudiantes precisamente porque muy a menudo se encuentran en el sustrato menos consciente de la actividad del especialista [7].

A partir de estas consideraciones generales el computador se puede considerar como una herramienta que permite visualizar y manipular de forma interactiva los objetos que son propios del control automático. El objetivo último es facilitar la comprensión de los conceptos que se tratan de transmitir y enseñar a los estudiantes. Se pueden encontrar en la literatura muchas herramientas interactivas sobre control automático, presentadas como una gran ayuda para la enseñanza, variando desde conceptos básicos hasta los más avanzados [19],[4],[12],[13],[21],[17],[29].

El desarrollo de estas herramientas ha mejorado durante los años, comenzando por las herramientas implementadas en Matlab incluyendo propiedades interactivas, hasta los entornos de programación específicos enfocados en el desarrollo de herramientas interactivas, como es el caso de Easy Java Simulations (EJS). EJS es una herramienta de código abierto que ayuda a crear laboratorios interactivos virtuales y remotos en lenguaje Java. Por otro lado, Sysquake es una herramienta cuyo lenguaje es similar a Matlab, siendo de rápida ejecución y proporciona excelentes facilidades para gráficos interactivos. EJS está llegando a ser muy popular para el desarrollo de laboratorios virtuales y remotos, mientras que Sysquake es el entorno más importante para desarrollar herramientas interactivas en ingeniería de control. Estas herramientas y otras similares son potentes para desarrollar herramientas gráficas interactivas pero tienen una serie de desventajas entre las que están la necesidad de generar archivos ejecutables para la plataforma donde se ejecutan, pueden ser pesadas teniendo que instalar extensiones en los computadores o puede que se necesiten licencias.

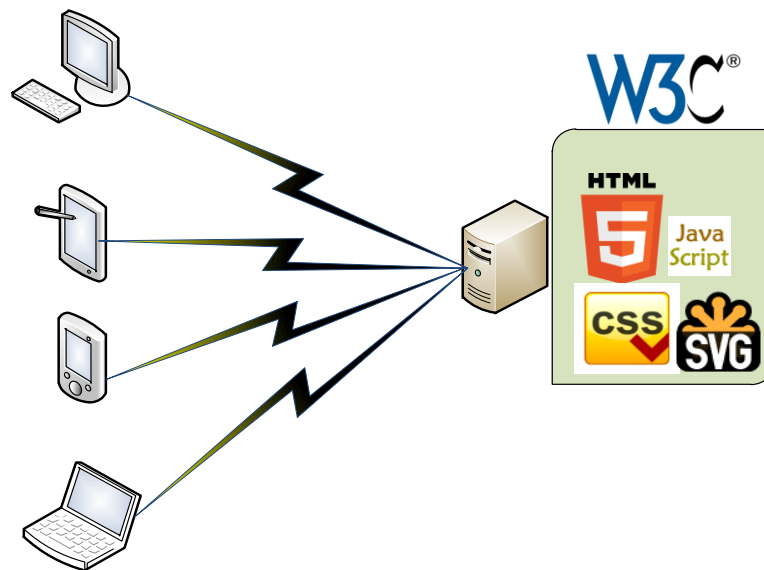


Figura 2. Esquema servidor-cliente usado en la web

En la actualidad existen multitud de herramientas interactivas utilizadas en el ámbito del control automático, donde la gran mayoría de ellas han sido desarrolladas con entornos específicos tipo Sysquake, Easy Java Simulation, etc. El inconveniente de estas herramientas interactivas resultantes es que precisan de la instalación de software adicional y/o de la descarga del archivo ejecutable de la misma para poder ser utilizada. Actualmente, esto se presenta como una desventaja cuando se desea hacer uso de este tipo de herramientas en cualquier dispositivo con acceso a Internet que se encuentra actualmente disponible, tales como Notebooks, computadores, dispositivos móviles, iPhone, iPad, etc. Muchos de estos dispositivos tienen limitada su capacidad y/o potencial y no soportan aplicaciones pesadas y, al mismo tiempo, algunos de ellos pueden tener sistemas operativos específicos no compatibles entre sí. Este hecho hace que o bien las herramientas interactivas desarrolladas se deban implementar para distintos sistemas operativos y distintas plataformas (exceptuando Java en algunos casos) o que se plantee la posibilidad de desarrollar las herramientas en algún tipo de soporte genérico que permita el uso de las mismas de forma transparente, independiente de la plataforma o dispositivo utilizado. Para ello, una solución factible es hacer uso de estándares web, (como los proporcionados por W3C para realizar aplicaciones basadas en web y que soportan la mayoría de los navegadores) para el desarrollo de este tipo de herramientas de tal forma que éstas se puedan utilizar desde cualquier tipo de dispositivo y solamente requiriendo el uso de un navegador web (ver Figura 2).

### 1.3 Objetivos del proyecto

El objetivo de este proyecto consiste en el desarrollo de una librería basada en tecnología web que permita la creación de herramientas interactivas para poder ser utilizadas en cualquier tipo de dispositivo electrónico que posea conexión a Internet y un navegador web. La implementación de esta librería de creación de herramientas interactivas se va a llevar a cabo usando los estándares W3C. Con el fin de mostrar la utilidad de este tipo de herramientas, se tomará el campo de la Automática como caso de estudio, por lo que la librería a desarrollar incorporará la representación matemática de los sistemas más típicos en control automático (sistemas de primer y segundo orden, con efecto de ceros y efecto del retardo, etc.).

Cabe destacar que las herramientas a desarrollar son herramientas a las que se les quiere dotar de carácter puramente interactivo (dicha idea se puede consultar en <http://aer.ual.es/ilm/>). De esta forma, los objetivos fundamentales de este proyecto se resumen en los siguientes puntos:

- *Desarrollo de una librería* de componentes gráficos interactivos que permita representar los elementos visuales básicos de la teoría del control automático. La idea es poder disponer de una librería genérica para poder ser utilizada en el desarrollo de herramientas futuras.
  
- *Desarrollo de un conjunto de cuatro ejemplos de herramientas interactivas:*
  - Sistema de primer orden con y sin retardo: Herramienta que tiene como fin transmitir los conceptos elementales de sistemas de primer orden. Se compondrá de tres gráficas; una representa la entrada en escalón a un sistema de primer orden, la segunda representando la salida del sistema y una tercera que muestra la posición del polo del sistema en el plano complejo  $s$ . Además, contendrá controles para modificar los parámetros característicos del sistema de primer orden y la entrada. También en las gráficas se podrá modificar dichos parámetros que aparecerán representados mediante líneas o puntos de forma interactiva.
  
  - Sistema de segundo orden con y sin retardo. Esta herramienta es similar a la de primer orden pero con los parámetros característicos de un sistema de segundo orden. Compuesta también por tres gráficas que representan los mismos elementos y con controles para los parámetros característicos de un sistema de segundo orden.
  
  - Sistema de segundo orden con un cero. En este caso se ampliaría la herramienta anterior teniendo especial interés en el efecto de un cero en el sistema.
  
  - Control PID. Para esta herramienta la segunda gráfica mostraría la salida de un sistema de primer orden con retardo, pero aplicándole un control PID al sistema, donde se podrían modificar los valores del control para estudiar el problema de seguimiento de referencias y rechazo a perturbaciones
  
- *Desarrollo de un sitio web* que incorpore el conjunto de las herramientas interactivas desarrolladas.
  
- *Validación de resultados en dispositivos móviles tipo iPad.*

## **1.4 Estructura de la memoria del proyecto fin de carrera**

Para una mejor comprensión del documento en este apartado se realiza una síntesis de los capítulos que forman este documento:

- Capítulo 1: Se realiza una introducción del proyecto, explicando el contexto del proyecto, la motivación de su realización y sus objetivos.
- Capítulo 2: Se describe la importancia de la interactividad en la enseñanza y especialmente en el campo del control automático.
- Capítulo 3: Se describen matemáticamente el modelado de los sistemas dinámicos y los controladores de los que se han creado herramientas interactivas implementando sus procesos en la librería.
- Capítulo 4: Aquí se describen las distintas tecnologías (estándares W3C) utilizadas para el desarrollo del proyecto y sus aspectos más importantes para su la implementación del proyecto.
- Capítulo 5: Se describen la forma de desarrollo utilizada y se describen el análisis y diseño de las distintas partes que forman el proyecto.





## INTERACTIVIDAD

Este capítulo está dedicado a describir la importancia de la interactividad en la enseñanza y especialmente en el campo del control automático [16]. En primer lugar se dará una serie de definiciones y clasificaciones. Posteriormente, en los siguientes apartados se irá ubicando la interactividad, primero, en la enseñanza de forma general y, de forma más específica, en la enseñanza de control automático.

### 2.1 Definiciones y clasificaciones

Interactividad, en un marco de trabajo no basado en computadoras, se puede describir como [25]:

**Definición 1.** Alguna acción realizada mutuamente por dos o más objetos, agentes, fuerzas, funciones, etc... la cual permite el intercambio de información entre unos y otros.

La Interactividad es un concepto basado en la bien conocida relación causa-efecto. Un buen fondo para entender la interactividad es observar las tareas diarias realizadas por la gente, donde cuerpo y sentidos perciben continuamente las interacciones del mundo. De forma cotidiana, cuando se conduce, el conductor está respondiendo interactivamente a un entorno circundante y al tráfico. La interacción es un parte aceptada de la vida cotidiana.

Hoy día, es muy común hablar de aplicaciones interactivas, televisión interactiva, teléfonos móviles interactivos, etc., asociando la interacción entre la gente y los dispositivos tecnológicos. La inclusión de este fenómeno en la sociedad ha sido debido a los avances en las nuevas TIC. Este nuevo significado de interactividad viene de su uso en la Ciencia de la Computación, donde la gente interactúa con el ordenador para realizar una tarea específica. Se pueden encontrar algunas definiciones complicadas sobre interactividad, como aquellas relacionadas con el desarrollo de herramientas interactivas o aplicaciones de ordenador basadas en gráficos [31]:

**Definición 2.** Es un compromiso entre los aspectos visuales, la comprensión del usuario, la creatividad, la ayuda técnica y los enlaces directos con la teoría o las ideas subyacentes para las cuales se requiere Interactividad.

En Ciencias de la Computación, la Interactividad es asociada incorrectamente a simples selecciones de menú, objetos a los que se les puede hacer clic, etc. La selección de menú es algo interactivo, pero este hecho no significa que una aplicación que incluye una simple selección de menú sea interactiva.



Los aspectos interactivos no son percibidos principalmente con los ojos sino con el sentido para la calidad interactiva de las cosas. Este sentido se puede considerar como una facultad de los seres humanos que permite percibir, juzgar, imaginar, diseñar y razonar sobre aspectos del comportamiento de nuestro entorno [31]. Por esta razón, el desarrollo de programas interactivos puede ser un poco complicado y subjetivo. La Interactividad envuelve a un amplio rango de disciplinas que incluye la ingeniería del software, inteligencia artificial, matemáticas, sociología, etc. [3]. Una buena descripción para mostrar la dificultad asociada a la Interactividad se describe en la siguiente definición [33]:

**Definición 3.** La Interactividad tiene propiedades 'Gestalt'. De la misma forma que ves una rosa, y no una colección de pétalos; escuchas un tema musical familiar, y no una secuencia de tonos; percibes el comportamiento interactivo de un control de una interfaz gráfica de usuario no como una colección de pares acciones/reacciones, sino como un conjunto completo con significado interactivo.

La inclusión de la Interactividad en una aplicación de ordenador puede ser vista como una cuarta dimensión. La representación gráfica de objetos en 3D es algo muy usual actualmente, permitiendo al usuario observar representaciones gráficas. Este hecho es muy importante, pero desde un punto de vista de la Ciencia de la Computación, se requieren más elementos: ¿qué se puede hacer con este elemento gráfico? ¿qué tipo de información se puede obtener como realimentación? Esto corresponde a la componente de acción, la cual describe las operaciones disponibles usando un elemento específico. Comparativamente, los aspectos interactivos de las interfaces han sido los menos investigados, pero como se comenta anteriormente, este hecho está cambiando y los elementos interactivos son más comunes en la sociedad.

En [9] se puede encontrar una descripción de la evolución del Diseño Interactivo, la cual se asocia al uso de la interactividad en materia de Control Automático. El diseño, con la participación del ser humano, es beneficioso en el sentido de que proporciona un enfoque rápido e intuitivo, lo cual permite al diseñador comprender que está pasando. Desde un punto de vista de diseño de control, la evolución en el uso de gráficos interactivos como ayuda puede ser dividido en 3 fases:

1. Cálculo manual. Se calculaban un conjunto de valores numéricos y los ingenieros de control desarrollaban un conjunto de reglas para dibujar gráficos manualmente.
2. Las computadoras como una herramienta auxiliar. Con la aparición de las computadoras se facilitó la creación de gráficos. Los ingenieros de control tenían la posibilidad de poner a punto los parámetros de diseño mediante prueba y error en un proceso iterativo. Cada iteración es dividida en dos fases. La primera consiste en calcular los parámetros desconocidos del sistema tomando un conjunto de variables de diseño. En la segunda fase se compara el comportamiento del sistema con las especificaciones. Si no coinciden se produce una nueva iteración.

3. Las computadoras como una herramienta interactiva. Es posible mezclar ambas fases en una, de forma que la modificación que se hace en los parámetros produce un efecto inmediato (véase la
4. Figura 1). De esta forma el procedimiento se hace realmente dinámico y el diseñador percibe el gradiente del cambio.

Un aspecto importante es como conocer el grado de interactividad de una aplicación. En [10] se extraen algunas características interactivas gracias al estudio de varios programas de enseñanza. A continuación, se describen algunas recomendaciones de diferentes autores para estudiar las características interactivas de una aplicación dada:

- Una **respuesta inmediata** significa que el usuario es capaz de acceder a la información (un gráfico, texto, o un vídeo) con un clic de ratón.
- Acceso no lineal a la información. A la habilidad de personalizar la respuesta al nivel de conocimiento del usuario se le llama **adaptabilidad**.
- No hay interactividad sin **realimentación**, el usuario debe conocer si se ha conseguido el objetivo. Las respuestas del tipo “correcto” o “incorrecto” no son adecuadas ya que el usuario necesita saber cómo mejorar y progresar.
- La adaptabilidad y la realimentación deberían tener en cuenta un **número suficiente de opciones**.
- El **grain-size** hace referencia a la cantidad de tiempo requerido antes de permitir al usuario interrumpir o iniciar una acción. La interactividad se compromete si el usuario tiene que esperar demasiado tiempo antes de que la interacción se dé, por ejemplo por un vídeo o por créditos.
- El **control del alumno** implica dar algún grado de responsabilidad al usuario. Si el usuario puede controlar algunos aspectos del proceso como el ritmo y la secuencia de la enseñanza, entonces la motivación y el aprendizaje probablemente aumenten. No se recomienda el control total del alumno.

Otro punto de vista consiste en estudiar los niveles de interactividad que pueden ser presentados en una aplicación específica. En [28] se identifican tres niveles de interactividad:

- **Reactivo**, ofrece muy poco control al usuario sobre el contenido, con las opciones de control del programa y la realimentación.
- **Coactivo**, ofrece control al usuario para secuencia, ritmo y estilo.
- **Proactivo**, permite al usuario controlar tanto la estructura como el contenido.

Por otro lado, en [20] se identificaron cinco niveles de interactividad enfocados a la implicación del usuario con el programa. Éstas son:

- Modalidad de la respuesta del alumno.
- La naturaleza de la tarea.
- El nivel de procesamiento.
- El tipo de programa.
- El nivel de inteligencia en el diseño.

Finalmente, se puede hacer una última clasificación de los aspectos interactivos basándose en el hecho de que los conceptos interactivos no son eventos mutuamente exclusivos, sino elementos que puedan ser integrados para proporcionar transacciones de enseñanza completas. Además, la aplicación de estas interacciones no sólo depende de las habilidades de los diseñadores y desarrolladores, sino también en la medida en que las interacciones son independientes o consecuentes [31]. Un buen contexto de estos conceptos son los siguientes niveles y funciones de interactividad: Interactividad de objetos, Interactividad lineal, Interactividad jerárquica, Interactividad de apoyo, Interactividad de actualización, Interactividad de construcción, Interactividad reflectiva, Interactividad hiperenlazada, Interactividad contextual no inmersiva e Interactividad virtual inmersiva.

## 2.2 Interactividad en la enseñanza

El uso de la Interactividad en la enseñanza implica las diferentes definiciones de la sección anterior. En una clase típica, el profesor debe motivar al estudiante y aumentar su participación, intentando obtener realimentación del estudiante como describe la definición 1 mostrada anteriormente. En una clase no interactiva el profesor explicará conceptos y los alumnos escucharán literalmente.

Otra inclusión de la interactividad en la enseñanza viene dada por el punto de vista descrito en la definición 3. Esto se conoce también como multimedia interactiva o programas interactivos. Son un tipo nuevo de programas de ordenador que mezclan las capacidades multimedia, junto con características interactivas. Consiste en herramientas interactivas con un alto nivel de abstracción permitiendo a los profesores mostrar conceptos teóricos de una forma más fácil, y a los estudiantes entender mejor los conceptos teóricos observando sus aplicaciones prácticas.

Una de las tareas más importantes de los profesores en ingeniería es transmitir no solamente la estructura lógica y formal de la disciplina sino, además, la estrategia y los aspectos intuitivos [37]. Este último aspecto es probablemente el más difícil de hacer explícito y asimilar por los estudiantes. Las herramientas interactivas son consideradas un gran estímulo para el desarrollo de la intuición de los estudiantes e intentan “desmitificar” conceptos matemáticos teóricos a través de la visualización para ejemplos escogidos especialmente [8].

Una herramienta interactiva es esencialmente una colección de ventanas gráficas que son manipuladas solamente usando el ratón. Si el estudiante cambia algún elemento activo en la ventana, se vuelve a recalcular y a presentar de forma automática. De esta manera, se percibe cómo las modificaciones afectan al resultado. Las herramientas interactivas no son efectivas solamente en presentaciones de conceptos teóricos en clase, sino que también son beneficiosas para ampliar la experiencia del estudiante en tareas de análisis y diseño.

Como las herramientas interactivas son fáciles de crear y desplegar, proporcionan un medio para el prototipado rápido y la prueba de ideas teóricas. En particular, las herramientas interactivas se pueden utilizar para investigar precisamente lo que es necesario hacer para que los estudiantes conozcan los conceptos teóricos. De esta manera, la virtud de la sencillez llega a ser un hecho en la investigación del aprendizaje sobre el diseño y la utilización de este tipo de herramientas [8].

En resumen, las razones principales por las cuales las herramientas interactivas son útiles para la enseñanza son las siguientes [11]:

- Facilitan el aprendizaje enfocado al alumno, permitiendo la elección en las vías para el aprendizaje y la velocidad a la que se introduce el nuevo material.
- Se pueden abordar varios estilos y modalidades de aprendizaje proporcionando una rica variedad de enfoques de enseñanza.
- Motivan la interacción, experimentación y el aprendizaje cooperativo de los estudiantes.
- Frecuentemente, los estudiantes trabajan juntos en proyectos.
- Facilitan argumentos o temáticas de aprendizaje en caso de que se pueda tejer una vía para la exploración en torno a una dinámica de conceptos particular.
- Promueven el punto de vista constructivista del aprendizaje.

A continuación se muestran algunos esquemas de enseñanza para identificar problemas donde puede ser adecuado el uso de las herramientas interactivas [11]:

- Material difícil de visualizar como procesos microscópicos.
- Material en 3D, que difícilmente se puede representar en medios 2D como son los libros pizarras.
- Procesos dinámicos, los cuales requieren una comprensión de las relaciones entre los objetos que se mueven.

- Material que abarca contextos generales, cuando una serie de ideas deben estar vinculados para obtener una comprensión de la totalidad, no sólo de las partes.
- Simulaciones de procesos complejos o costosos, donde la comprensión puede ser obstaculizado por detalles del proceso mecánico, o donde no hay posibilidad de utilizar un equipo real.

### 2.3 Enseñanza del Control Automático basada en la Interactividad

Para diseñar sistemas técnicos o, simplemente, para entender las leyes físicas que describen su comportamiento, los científicos e ingenieros suelen utilizar los ordenadores para calcular y representar gráficamente diferentes magnitudes. En ingeniería de control, entre estas cantidades se incluyen: tiempo y frecuencia de respuestas, polos y ceros en el plano complejo, diagramas de Bode, Nyquist y Nichols, etc. Con frecuencia, estas magnitudes están estrechamente relacionadas y constituyen diferentes visiones de una misma realidad simple. La comprensión de estas relaciones es una de las claves para conseguir un buen aprendizaje de los conceptos básicos y permitir a los estudiantes diseñar con precisión los sistemas de control [7].

Las ideas, conceptos y métodos de Control Automático son muy ricos en contenido visual, ya que estos pueden ser representados de manera intuitiva y geométrica. Estos contenidos visuales pueden ser utilizados para la presentación de tareas y manejo de conceptos y métodos, y manipulados para la solución de problemas. Los especialistas de control tienen imágenes visuales, formas intuitivas de percibir los conceptos y métodos que son sumamente importantes para llevar a cabo eficazmente su labor creativa y el dominio del campo en el que trabajan. Haciendo uso de imágenes visuales y de la intuición, son capaces de relacionar las constelaciones de hechos que son a menudo muy complejas, y los resultados de sus teorías de una manera extremadamente versátil y variada. Asimismo, a través de estas importantes redes, son capaces de elegir naturalmente y sin esfuerzo, las estrategias más eficaces para atacar y resolver los problemas a los que se enfrentan [9].

El primer sentido usado por la gente es la vista. Esto no es nada sorprendente ya que el apoyo visual está presente diariamente en nuestro trabajo. Expertos en control a menudo hacen uso de procesos simbólicos, diagramas visuales y otras formas de imaginación en sus procesos de trabajo y adquieren, lo que se podría llamar, una intuición de lo abstracto. De este modo, la visualización parece ser algo muy natural, tanto en los orígenes de control automático como en el descubrimiento de nuevas relaciones entre objetos matemáticos y también, por supuesto, en la transmisión y comunicación del conocimiento sobre control. Una de las tareas importantes para los profesores de ingeniería de control es la de transmitir a los estudiantes no sólo la estructura formal y la lógica de la disciplina, sino también, y sin duda con mucho más énfasis, los aspectos estratégicos e intuitivos de la materia. Estos aspectos estratégicos e intuitivos son probablemente mucho más difíciles de hacer explícitos y de asimilar para el estudiante [9].

El grado de apoyo visual varía considerablemente de un análisis a otro. Lo que para una persona es útil, para otra persona puede ser un obstáculo. Según [32], el control de un diseñador (profesor, ingeniero o estudiante) debe cumplir con el siguiente procedimiento, con el fin de diseñar de forma adecuada un sistema de control de una planta [29]:

1. Estudiar de la planta a controlar y obtener la información inicial.
2. Modelar la planta y simplificar el modelo, si es necesario.
3. Analizar el modelo resultante; determinando sus propiedades.
4. Decidir que variables deben ser controladas.
5. Seleccionar la configuración de control.
6. Decidir el tipo de controlador a usar.
7. Decidir sobre especificaciones de rendimiento, basándose en el conjunto de objetivos de control.
8. Diseñar un controlador.
9. Analizar los resultados y si las especificaciones no son satisfechas, modificar o bien las especificaciones o el tipo de controlador.
10. Simular el sistema de control resultante, en un ordenador o en una planta piloto.
11. Repetir desde el paso 2, si es necesario.
12. Elegir hardware y software e implementar el controlador.
13. Probar y validar el sistema de control y sintonizar el controlador en línea, si es necesario.

Se puede observar que si el diseño del sistema de control lleva a unos resultados insatisfactorios de las simulaciones (paso 10), se forzaría a repetir los pasos desde el 2 hasta el 10 analizándolos uno a uno. El enfoque previo al diseño de un sistema de control usando herramientas tradicionales puede ser considerado como un enfoque no interactivo (ver Figura 1(a)). Esto es debido al hecho de que no es posible conocer las consecuencias de las decisiones tomadas durante el proceso de diseño hasta que estamos muy cerca del final (paso 11). Esta situación produce que se tengan que repetir los mismos pasos, siendo ésta la consecuencia de que no haya un enlace en tiempo real entre el diseño y el análisis [29].

Si hubiera tal conexión en tiempo real o vínculo entre las decisiones adoptadas durante la fase de diseño y los resultados obtenidos en la fase de análisis, ambas fases podrían fusionarse en una. En este nuevo enfoque, las decisiones adoptadas en algunos pasos mostrarían al vuelo las diferencias entre los resultados de la simulación y las especificaciones de control fijadas en el

paso 1. No sería necesario iterar tantas veces tratando de ajustar los parámetros. Este propósito de fusionar análisis y diseño puede ser un hecho proporcionando Interactividad en el desarrollo de nuevas herramientas para la enseñanza de control. Por lo tanto, éste sería un método interactivo (ver Figura 1 (b)).

En sistemas altamente interactivos hay varias ventanas gráficas que se actualizan inmediatamente, reflejando el valor tanto de cualquier elemento activo, como de sus restricciones. Esto permite un comportamiento reactivo entre todos los componentes activos del sistema, escondiendo la base matemática subyacente. A lo largo de los años, se han desarrollado muchas herramientas para la enseñanza de control incorporando los conceptos de imágenes dinámicas y sistemas interactivos virtuales. Hoy día hay una nueva generación de paquetes de software, basados en objetos que admiten una manipulación directa del gráfico y que son actualizados automáticamente, de modo que la relación se mantiene continuamente. Los primeros programas fueron útiles, pero su implementación requería un gran esfuerzo, lo que limitaba su uso en la educación. Los avances en ordenadores y software han hecho que la implementación sea más sencilla. Matlab se ha utilizado en varios proyectos, obteniendo dos resultados exitosos como ICTools [19] y CCSdemo [34] desarrollados en el Departamento de Control Automático en el Instituto de Tecnología de Lund. Sin embargo, los programas muy dependientes de la versión lo que hace que el soporte y los futuros desarrollos sean engorrosos. Yves Piguet en el Instituto Federal de Tecnología de Lausana (EPFL) ha desarrollado un entorno de programación similar a Matlab, denominado Sysquake, que tiene un alto soporte para gráficos interactivos [24]. Se han desarrollado proyectos basados en Sysquake en EPFL y en otros lugares (véase [13][14][1][6]).

Por lo tanto, el papel de esta nueva experiencia de aprendizaje interactivo en el plan de estudios de ingeniería de control es doble [7] :

- proporcionar un nuevo método para impartir material de clase por el cual introducir los conceptos de ingeniería de control de sistemas vía paquetes interactivos y,
- proporcionar una nueva oportunidad para innovar en trabajos de laboratorio donde los estudiantes puedan analizar, diseñar y modificar sistemas de ingeniería de control vía herramientas interactivas.



## MODELADO Y CONTROL PID

En este capítulo se realiza una revisión de la teoría de modelado y control que ha sido necesaria para la implementación de la librería y la creación de las herramientas interactivas que tiene lugar en este proyecto. Se empieza con el modelado de sistemas dinámicos básicos destacando las características más interesantes para el desarrollo del proyecto, se continúa describiendo los sistemas de control haciendo hincapié en el sistema de control PID y finalmente describiendo los sistemas de control en tiempo discreto. Esta teoría ha sido obtenida en su mayor parte a partir de [27].

### 3.1 Sistema, modelo y simulación

El diseño de un sistema de control para un determinado proceso requiere conocer el comportamiento dinámico del mismo. Para ello hay que diseñar e implementar un modelo del sistema que consiste en la descripción matemática de las relaciones dinámicas existentes entre las variables a controlar y las variables de entrada, tanto perturbaciones como variables de control. La principal ventaja de la utilización de un modelo radica en que permite la evaluación del resultado de una decisión del mundo real sin llegar a tomar efectivamente la misma. Disponiendo de un modelo dinámico se puede analizar y evaluar varias cuestiones claves en el diseño de un sistema de control: emparejamiento de variables para formar lazos de control, límites de la acción de control y evaluación de la dificultad de control (signo y sensibilidad de la respuesta, velocidad de respuesta, forma de la respuesta y sensibilidad a cambio en el proceso).

Con los términos modelización y simulación se designa a l conjunto de actividades asociadas con la construcción de modelos de sistemas del mundo real y su simulación en un computador (ver Figura 3):

- Un *sistema* es la parte del mundo real por la que se muestra interés, compuesta por partes interrelacionadas.
- Un *modelo* es una representación matemática de un sistema que contiene un conjunto de instrucciones para generar datos del comportamiento del sistema a estudiar.
- La *simulación* relaciona modelos y computadoras. Se refiere al proceso de imitación de aspectos importantes del comportamiento del sistema mediante el diseño, construcción y experimentación con el modelo del sistema. Este proceso es similar al de experimentación que llevan a cabo los científicos en un laboratorio con el que pretenden aumentar la comprensión de alguna teoría para su validación y empleo posterior.



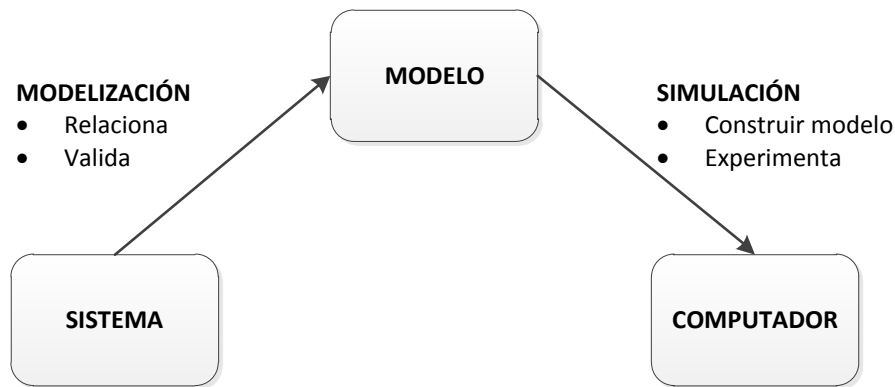


Figura 3. Sistemas, modelos y simulación

### 3.2 Función de transferencia de un sistema dinámico

Si las condiciones iniciales de un sistema representado por una ecuación diferencial son nulas, cuando se le aplica la transformada de Laplace existe una relación entre la entrada y la salida del sistema que se conoce como función de transferencia. La función de transferencia de un sistema es la relación que existe entre la transformada de Laplace de la salida con respecto a la transformada de Laplace de la entrada. Generalmente se denota por  $G(s)$  y se describe por el diagrama de bloques mostrado en la Figura 4 y por la siguiente ecuación:



Figura 4. Función de transferencia

$$Y(s) = U(s) \cdot G(s) \Rightarrow G(s) = \frac{Y(s)}{U(s)}$$

Al denominador de la función de transferencia de un sistema se le denomina polinomio característico, siendo el orden del sistema el grado de este polinomio. A las raíces del polinomio característico se les conoce como polos, y a las del numerador de la función de transferencia como ceros.

### 3.3 Comportamiento de sistemas dinámicos

#### 3.3.1 Señales de prueba

Para determinar la respuesta de un sistema, se utiliza una serie de señales de prueba y en función de la respuesta a estas señales de prueba, se pueden deducir las principales propiedades del sistema. Las señales de prueba más comunes son las que se muestran en la Figura 5.

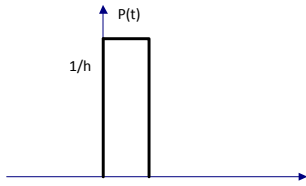
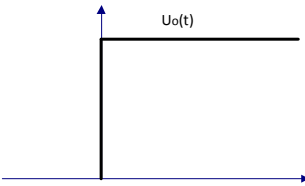
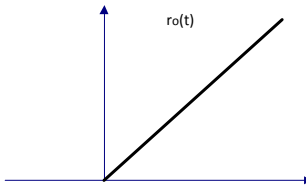
Impulso	Escalón	Rampa
$\delta(t) = \begin{cases} \infty & , t = 0 \\ 0 & , t \neq 0 \end{cases} = \lim_{h \rightarrow \infty} P(t)$	$u_o(t) = \begin{cases} 0 & , t < 0 \\ U & , t \geq 0 \end{cases}$	$r_o(t) = \begin{cases} 0 & , t < 0 \\ t & , t \geq 0 \end{cases}$
		

Figura 5. Principales señales de prueba

La más típica de las entradas es el escalón, que se puede implementar por ejemplo, mediante la apertura instantánea de válvulas de un sistema de calefacción que se mantiene durante un periodo de tiempo, por lo que es muy habitual estudiar la respuesta de sistemas a un escalón de amplitud, siendo su transformada de Laplace la siguiente:

$$u(t) = \begin{cases} 0, & t < 0 \\ U, & t \geq 0 \end{cases} \Rightarrow Laplace \Rightarrow U(s) = \frac{U}{s}$$

### 3.3.2 Sistemas de primer orden

En función de la ecuación diferencial que modelo el sistema o del orden del polinomio característico de la función de transferencia se pueden clasificar a los sistemas. Si la ecuación diferencial es de primer orden (tiene una sola derivada primera) o el orden del polinomio característico es 1, se dice que el sistema es de primer orden cuya forma general es (ver Figura 6):



Figura 6. Sistema de primer orden

$$\tau \frac{dy(t)}{dt} + y(t) = k \cdot u(t) \Rightarrow Laplace \Rightarrow \tau \cdot s \cdot Y(s) + Y(s) = k \cdot U(s)$$

$$\Rightarrow G(s) = \frac{Y(s)}{U(s)} = \frac{k}{1 + \tau s}$$

Como se puede observar, el orden del polinomio característico es uno.

Utilizando la función de transferencia de un sistema de primer orden, se puede observar que la respuesta es una función exponencial, tal y como se demuestra a continuación:

$$Y_{rp}(s) = \frac{k}{1 + \tau s} U(s) = \frac{k}{1 + \tau s} \frac{U}{s} = \frac{A}{s} + \frac{B}{1 + \tau s} = \frac{kU}{s} - \frac{kU\tau}{1 + \tau s} = kU \left( \frac{1}{s} - \frac{1}{\frac{1}{\tau} + s} \right)$$
$$\Rightarrow \text{Laplace inversa} \Rightarrow Y_{rp}(s) = kU \left( 1 - e^{-\frac{t}{\tau}} \right)$$

Gráficamente:

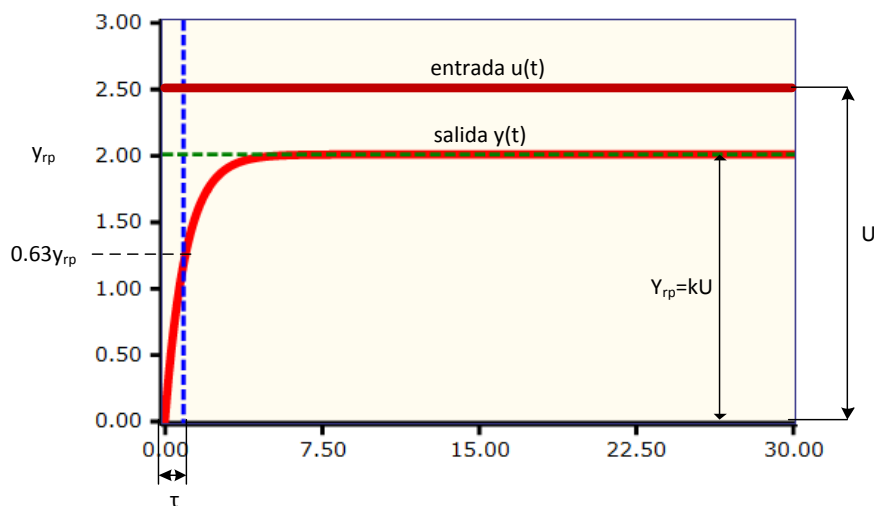


Figura 7. Respuesta de un sistema de primer orden a un escalón

Como se puede observar en la Figura 7, esta respuesta viene determinada por dos parámetros,  $k$  y  $\tau$ , que poseen un significado físico:

- $k$  se denomina ganancia estática del sistema, que proporciona una medida de amplitud de la respuesta del sistema en función de la amplitud de la entrada al mismo. Es el cociente entre el cambio experimentado por la salida una vez alcanzado el régimen permanente final ( $kU$ ) y el cambio efectuado en la variable de entrada (magnitud del escalón,  $U$ ).
- $\tau$  se denomina constante de tiempo, que es el tiempo que tarda el sistema en alcanzar el 63% del valor final de la señal de salida desde que comienza a variar el sistema frente a la entrada en escalón. Es una medida de la rapidez de respuesta del sistema a una señal escalón.

Por otra parte, la respuesta de un sistema a una determinada señal puede considerarse como la suma de dos componentes:

- Régimen permanente. Respuesta del sistema cuando alcanza un comportamiento constante o periódico.
- Régimen transitorio. Respuesta inicial del sistema hasta que alcanza el régimen permanente.

### 3.3.3 Sistemas de segundo orden

Cuando la ecuación diferencial que describe al sistema tiene una derivada doble o el polinomio característico de la función de transferencia es de orden 2, se dice que el sistema es de segundo orden. Existen dos tipos de sistemas de segundo orden:

- Sistemas de segundo orden constituidos por dos sistemas de primer orden en serie, como por ejemplo dos depósitos de acumulación de líquido en serie.
- Sistemas de segundo orden intrínsecos, que son aquellos en los que su dinámica inherente es de segundo orden, como por ejemplo una válvula neumática.

La función de transferencia general de un sistema de este tipo viene dada por:

$$\frac{d^2y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_2 y(t) = bu(t) \Rightarrow \text{Laplace} \Rightarrow s^2 Y(s) + a_1 s Y(s) + a_2 Y(s) = b U(s)$$

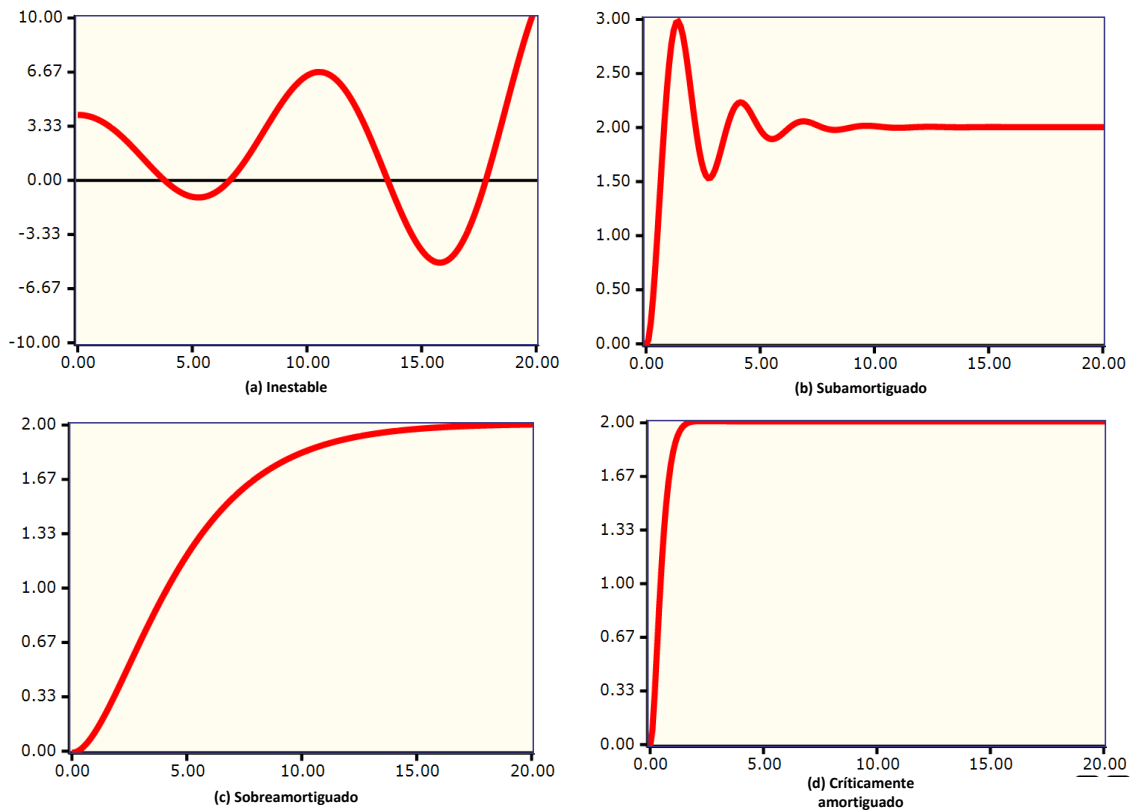
$$\Rightarrow G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s^2 + a_1 s + a_2}$$

Como se puede observar, el orden del polinomio característico es dos.

La respuesta de un sistema de segundo orden a un escalón puede ser de dos tipos:

- No acotada. Cuando el sistema no alcanza un punto de equilibrio después de aplicarle una señal de entrada. Los sistemas que exhiben este tipo de comportamiento aun cuando la entrada aplicada no es divergente se llaman inestables. En la (Figura 8 (a)).se puede observar la respuesta de un sistema inestable cuando se le aplica un escalón unidad como señal de entrada.
- Acotada. El sistema llega a un equilibrio cuando se le aplica una señal de entrada. En función de las características del transitorio del sistema existen tres tipos:
  - Sobreamortiguado. Respuesta sin oscilaciones (Figura 8 (c)).

- Críticamente amortiguado. Máxima velocidad sin oscilaciones (Figura 8 (d)).
- Subamortiguado. Respuesta con oscilaciones (Figura 8 (b)).



**Figura 8. Tipos de respuestas a un escalón de un sistema de segundo orden**

Al igual que ocurre con los modelos de primer orden, existen unos parámetros que definen los modelos de segundo orden:

- Ganancia estática,  $k$ . El significado es similar que en un sistema de primer orden.
- Frecuencia natural no amortiguada,  $\omega_n$ . Corresponde a la frecuencia con la que oscilaría el sistema si no existiera amortiguamiento.
- Coeficiente de amortiguamiento,  $\delta$ . En función de su valor se puede deducir si el sistema es inestable, críticamente amortiguado, sobreamortiguado o subamortiguado (en este caso además su valor determina la amplitud de las sobreoscilaciones).
  - Si  $\delta \leq 0$  entonces el sistema es inestable.
  - Si  $0 < \delta < 1$  entonces el sistema es subamortiguado y presenta sobreoscilaciones.

- Si  $\delta = 1$  entonces el sistema es críticamente amortiguado.
- Si  $\delta > 1$  entonces el sistema es sobreamortiguado.

Para identificar estos parámetros en base a la función de transferencia o la ecuación diferencial que describe al sistema hay que utilizar las siguientes ecuaciones:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b}{s^2 + a_1s + a_2} = \frac{k\omega_n^2}{s^2 + 2\delta s + \omega_n^2} \Rightarrow \begin{cases} b = k\omega_n^2 \\ a_1 = 2\delta\omega_n^2 \\ a_2 = \omega_n^2 \end{cases}$$

### 3.3.4 Análisis dinámico de respuestas subamortiguadas

El régimen transitorio de un sistema de control práctico presenta con frecuencia una respuesta subamortiguada con oscilaciones previas a alcanzar el estado estacionario. El análisis de este tipo de respuestas es común realizarlo en base a un conjunto de características (ver Figura 9):

- Tiempo de subida: es el tiempo requerido para que la respuesta del sistema pase del 10% al 90% de su valor final en régimen permanente (o bien desde el 0 al 100%).
- Sobreoscilación máxima: máximo valor de pico de la respuesta del sistema. Su cálculo se realiza en % como:

$$SO(\%) = \frac{y_{max} - y_{\infty}}{y_{\infty}} \cdot 100\%$$

Donde  $y_{\infty}$  representa el valor alcanzado en régimen permanente e  $y_{max}$  la máxima amplitud de la respuesta.

- Tiempo de pico: tiempo que transcurre hasta alcanzar la máxima sobreoscilación.
- Tiempo de establecimiento: tiempo necesario para que la salida del sistema se encuentre en un rango  $([-x, x])$  típicamente del 3% ó 5% alrededor del valor final.

Las características presentadas anteriormente poseen una relación directa con los parámetros que caracterizan a un sistema de segundo orden de la siguiente forma.

$$SO(\%) = 100e^{-\delta\pi/\sqrt{1-\delta^2}}$$

$$t_e = \frac{3}{\delta\omega_n}$$

$$t_p = \frac{\pi}{\omega_d}$$

$$t_s = \frac{\pi - \varphi}{\omega_d}$$

$$\delta = \cos\varphi$$

$$\omega_d = \omega_n \sqrt{1 - \delta^2}$$

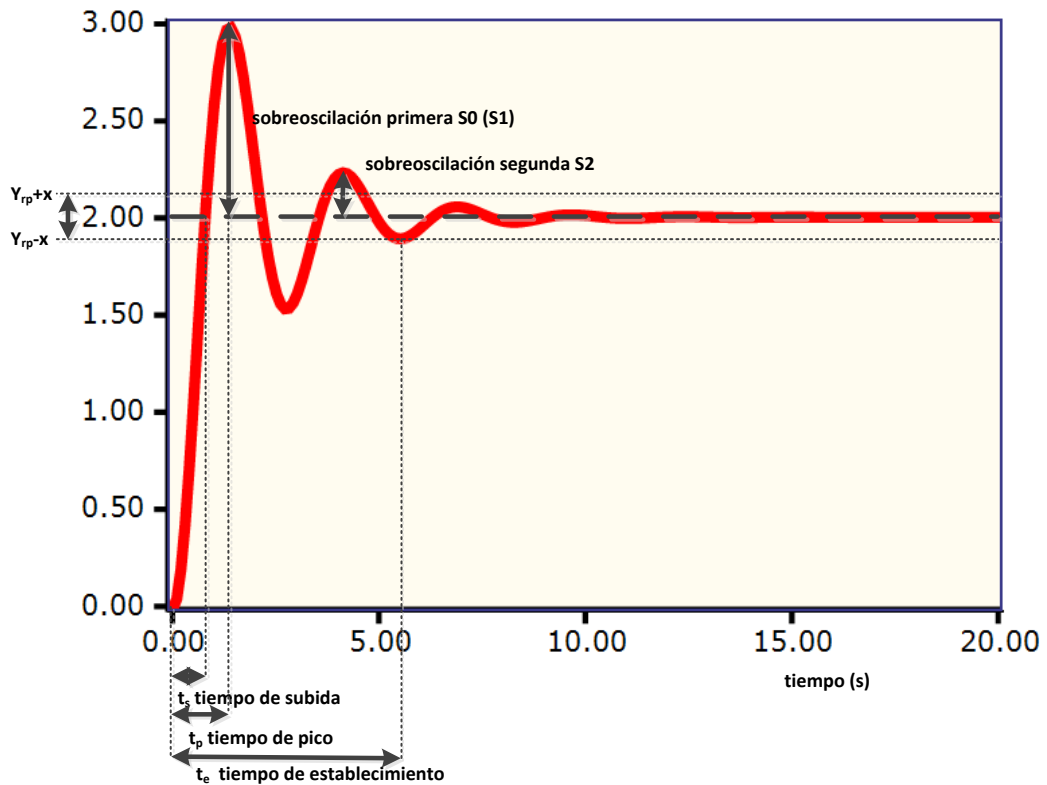


Figura 9. Parámetros temporales característicos de respuesta subamortiguada

### 3.3.5 Estabilidad absoluta y relativa

Haciendo uso de lo estudiado en los apartados anteriores, es posible conocer a priori una serie de características que definen el comportamiento dinámico de un determinado sistema. Sin embargo, al trabajar con sistemas dinámicos la principal característica que se debe conocer es si el sistema es estable o inestable.

Se entiende como estabilidad absoluta a la medida que permite conocer si un sistema es estable o inestable, y estabilidad relativa, a aquella que da una indicación sobre cómo de estable o inestable es el sistema. A partir de la función de transferencia de un sistema. La estabilidad absoluta del mismo se conoce calculando las raíces del polinomio característico. Si todas las raíces poseen parte negativa el sistema es estable, y si alguna de ellas tiene parte real

positiva el sistema es inestable. Dentro de la estabilidad, un sistema será menos estable (relativamente hablando) cuanto más se acerquen sus raíces al eje imaginario ( $0 \pm xj$ ). La Figura 10 muestra un conjunto de ejemplos donde se puede ver mediante la observación de la respuesta a un escalón la variación de la estabilidad del sistema en función de la localización de los polos del mismo.

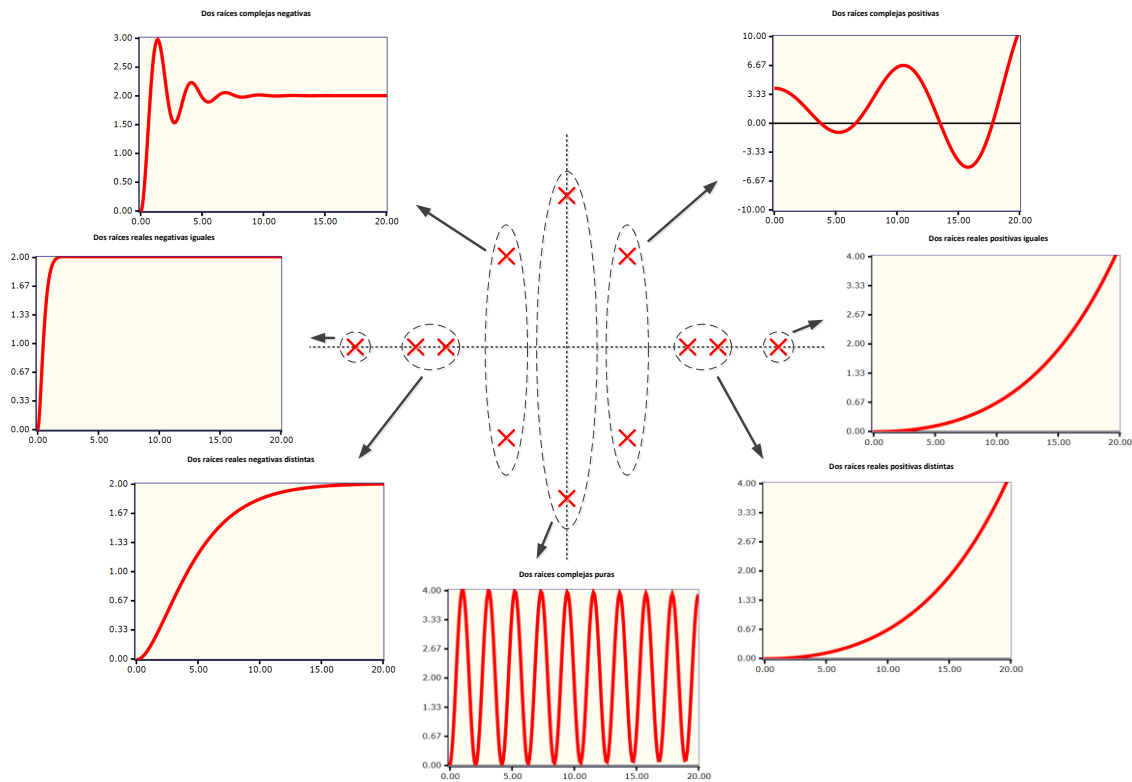


Figura 10. Análisis de estabilidad en función de la localización de los polos

### 3.3.6 Efecto de ceros sobre la respuesta

Los ceros afectan en un sistema al valor de la ganancia y a la forma de respuesta transitoria, pero no a la estabilidad. La respuesta a la entrada de un sistema con un cero en  $s = -c$ , se obtiene sumando a la respuesta del sistema sin cero su derivada multiplicada por un factor  $1/c$ .

$$G(s) \left( \frac{1}{c} s + 1 \right) = G(s) + \frac{1}{c} s \cdot G(s)$$

Según la posición del cero se tendrán comportamientos diferentes.

- Con  $c > 0$ , se adelanta la respuesta. No se producen oscilaciones si la respuesta sin cero no la tiene, pero puede producir sobrepico. En este caso el cero se encuentra en la parte real izquierda del plano  $s$  (Figura 11).



- Con  $c < 0$ , se produce una respuesta inversa inicialmente también llamada fase no-mínima. De esta forma el cero se encuentra en la parte real derecha del plano  $s$  como se puede apreciar en la Figura 11.

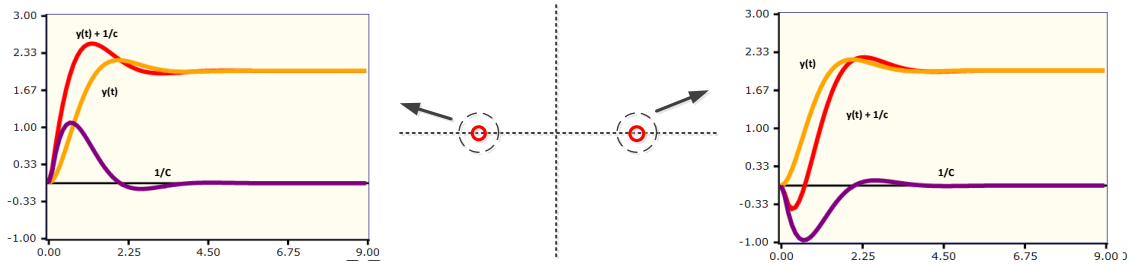


Figura 11. Análisis de efecto de ceros en función de su localización

### 3.4 Sistemas de control

#### 3.4.1 Concepto de sistema de control

Como se ha descrito previamente, un sistema es una combinación de elementos que interactúan. Controlar una variable consiste en mantener su valor dentro de unos límites deseados o preestablecidos; por tanto, un sistema de control es aquél cuyo objetivo es mantener una o varias variables dentro de unos límites prefijados, con un comportamiento conocido. En la teoría de control, la planta es cualquier objeto físico que deba controlarse y proceso es la operación o la variable a controlar.

Cuando se pretende controlar un proceso como la temperatura interior de un invernadero, la planta está sujeta a perturbaciones que son variables que tienden a afectar el valor de salida del sistema pero no pueden ser manipuladas. Si la perturbación se genera dentro del sistema se le denomina interna (p.e. efecto del suelo sobre la temperatura del aire), mientras que si se produce fuera de él, se le denomina externa (p.e. viento, lluvia, etc.). Estas perturbaciones obligan a estar permanentemente observando el proceso y a actuar constantemente sobre el mismo para corregir las desviaciones que se produzcan, siendo esta misión el principal objetivo del sistema.

#### 3.4.2 Funcionamiento de un sistema de control

Un sistema de control se puede representar mediante la Figura 12.

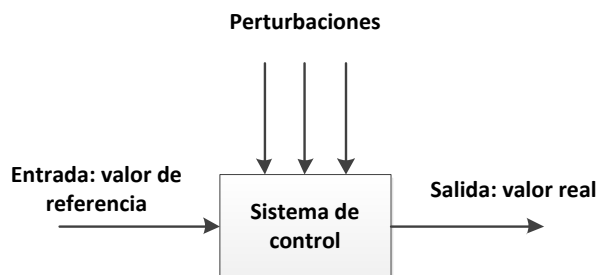


Figura 12. Representación de un sistema de control

Todo sistema de control tiene una entrada denominada consigna, que es el valor de referencia, o valor “deseado”, utilizada para enviar órdenes a fin de que el valor real de la variable a controlar sea igual a éste, o tan parecido como sea posible en presencia de perturbaciones o cambios en el valor de referencia. Por tanto, las acciones básicas que debe realizar todo sistema de control son:

- Observar. Medir la variable a controlar.
- Comparar con un valor de referencia deseado.
- Actuar para llevar el parámetro la variable de salida al valor deseado.

Como se ha podido observar, en un sistema de control hay que considerar las siguientes variables:

- Variable a controlar que es la que se desea mantener en un valor deseado, por ejemplo, la temperatura que se genera en el interior de un invernadero.
- Consigna o valor de referencia que es el valor deseado para la variable a controlar. En numerosas referencias bibliográficas y equipos de control se le denomina por su nombre en inglés Setpoint (SP).
- Variables manipulada o de control que es la variable que se emplea para compensar o corregir el efecto de las perturbaciones, por ejemplo, la apertura de la ventilación de un invernadero para controlar su temperatura.
- Variables de perturbación que son aquellas que afectan a la variable a controlar y no se pueden manipular, por ejemplo, el viento en el exterior del invernadero, la radiación solar, etc. Sobre estas variables no se puede actuar ya que su valor está impuesto por el mundo exterior.

### 3.4.3 Elementos de un sistema de control

Un sistema de control en lazo cerrado comprueba la señal de salida y decid si el nivel de la señal real de salida corresponde al de la señal deseada, o si el nivel real de la señal ha de ser modificado para conseguir el objetivo fijado. También se les conoce con el nombre de sistemas retroalimentados o realimentados. El diagrama de bloques de un sistema de control en lazo cerrado es el mostrado en la Figura 13.

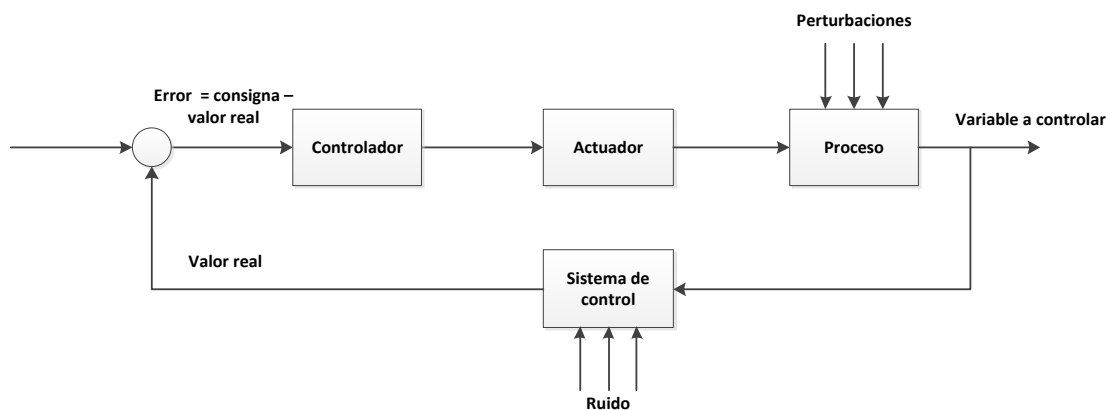


Figura 13. Sistema de control en bucle cerrado

Como se observa, el sistema tiende a mantener una relación preestablecida entre la salida y una entrada de referencia (consigna), comparándolas y utilizando la diferencia, señal de error, como medio de control. En presencia de perturbaciones, este control tiende a reducir la diferencia entre el valor deseado y el valor real de la salida.

### 3.5 Controladores automáticos

#### 3.5.1 Concepto de controlador automático

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (valor deseado), determina el error y produce una señal de control que reducirá el error a cero, o a un valor muy pequeño, sin necesidad de supervisión por parte de un humano. La forma por la que el controlador produce la señal de control, se denomina acción de control. Los procesos presentan tres características básicas que deben ser consideradas al automatizarlos:

- Cambios en la variable controlada debido a alteraciones en las condiciones del proceso. Se pueden producir por dos razones:
  - Cambios en la consigna.
  - Perturbaciones.
- Tiempo necesario para que la variable del proceso alcance un nuevo valor de equilibrio al ocurrir un cambio de carga.
- Estabilidad.

Dependiendo de cómo se consideren estos tres parámetros habrá que utilizar una u otra acción de control.

#### 3.5.2 Acciones básicas de control

Existen cuatro acciones básicas de control que se utilizan por separado o en combinación. Estas acciones: Todo/Nada, Proporcional, Derivativo e Integral.

Combinando estas acciones surgen los seis tipos básicos de control. En base a estas acciones se pueden clasificar los sistemas de control, tal y como se muestra en la siguiente tabla:

Sistemas de control	Acciones de control
Discontinuos	Control Todo/Nada
	Control de varias posiciones
Continuos	Control Proporcional
	Control Proporcional Integral
	Control Proporcional Derivativo
	Control Proporcional Integral Derivativo

### 3.5.3 Control Proporcional

En los casos en que se requiera una acción de control más suave que con un controlador discontinuo, puede utilizarse un controlador proporcional, que genera una señal de control proporcional al error. Actúa como un amplificador de ganancia  $K_p$ . Su acción se representa por:

$$u(t) = K_p \cdot e(t) + u_0$$

Donde  $u_0$  es la salida del controlador cuando el error es cero. La curva característica de la salida del controlador proporcional en función del error se muestra en la

Figura 10. La salida de un controlador se expresa a menudo como un porcentaje del rango completo de posibles valores donde el 0% representa la salida mínima del controlador  $u_{min}$ , y el 100% la máxima  $u_{max}$ .

$$u(\%) = \frac{u(t) - u_{min}}{u_{max} - u_{min}} \cdot 100$$

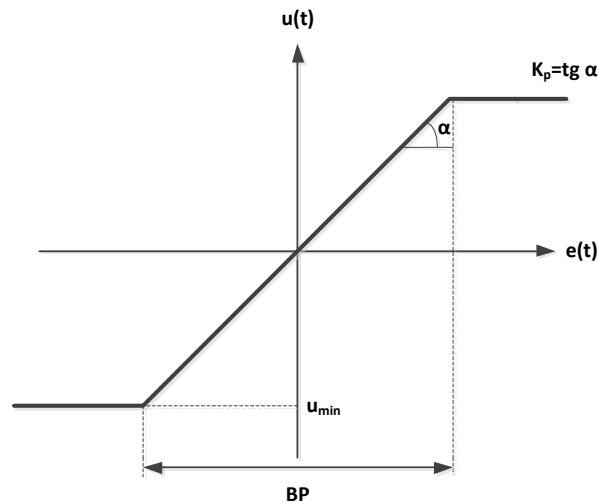


Figura 14. Control Proporcional

Como se puede observar las características más importantes de un controlador proporcional son:

- Ganancia ( $K_p$ ) que es la cantidad por la que se multiplica la señal de error para obtener la señal de salida.
- Banda proporcional ( $BP$ ) que es el porcentaje de variación de la variable controlada necesario para provocar que el elemento final de control recorra sus dos valores extremos. Bandas proporcionales “anchas” implican una lenta aproximación al punto de consigna y bandas relativamente estrechas implican un comportamiento agresivo del controlador consecuencia de la celeridad de su respuesta frente a la inercia de los sistemas, que puede llevar a inestabilidad en determinadas situaciones como consecuencia de la saturación de los actuadores (puesto que éstos tienen un rango limitado de actuación).

Matemáticamente, estos parámetros están relacionados por la siguiente expresión:

$$BP = \frac{100}{K_p}$$

Por ejemplo, se puede utilizar un controlador proporcional para controlar la altura de un depósito de agua donde su nivel puede variar entre 0 y 9 metros, entrando un caudal constante de líquido,  $Q_e(t) = q_e$ . Si la altura requerida (consigna) es de 5.0 metros y el controlador cierra completamente una válvula de salida cuando el nivel de agua baja a 4 metros y la abre completamente cuando asciende a 6 metros, el controlador necesario es el siguiente (ver Figura 11):

- Cuando el error es de -1 metro la salida del controlador debe estar abierta al 100%.
- Cuando el error es de 1 metro la salida del controlador debe estar cerrado al 0%.
- La banda proporcional debe abarcar un error de altura de -1 metro a 1 metro, es decir, 2 metros y como  $K_p = -100/BP = -50$ . El signo negativo es debido a la naturaleza del proceso.

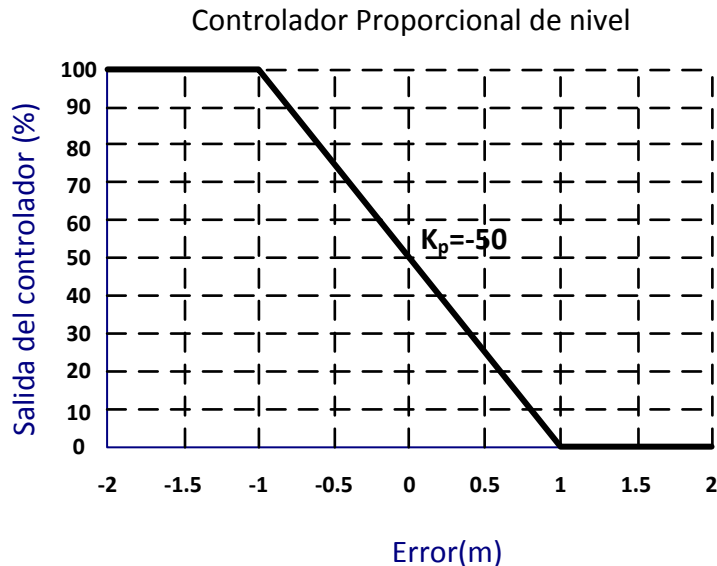


Figura 15. Control proporcional de nivel

La ecuación que describe al controlador sería:

$$u(t) = -50 \cdot e(t) + 50$$

Generalmente, si se desea modificar la consigna de un sistema de control proporcional, no se puede llevar el error a cero, sino que se suele producir un error permanente denominado desplazamiento, o error de offset o error de régimen permanente, consistente en la desviación permanente

De la variable controlada respecto a su consigna al alcanzar el estado estacionario, siendo éste uno de los principales inconvenientes de los controladores proporcionales.

Cuando la señal de salida del sistema final difiere del punto de consigna se denomina error de offset. El error de régimen permanente puede reducirse disminuyendo el valor de la banda proporcional, es decir, aumentando la ganancia  $K_p$ , pero la banda no puede ser tan estrecha como se desee, ya que el sistema daría señales de control de gran amplitud, saturando los actuadores y tendiendo a comportarse como un controlador todo/nada, pudiendo hasta inestabilizar al sistema. Con valores excesivamente bajos de  $K_p$  el sistema evoluciona con lentitud y aumenta el error en régimen permanente. Como se puede observar en la Figura 16, a mayor  $K_p$ , el error va disminuyendo y el sistema es más rápido, pero aparecen más oscilaciones (en el límite, se conseguiría eliminar el error en régimen permanente con  $K_p = \infty$ ).

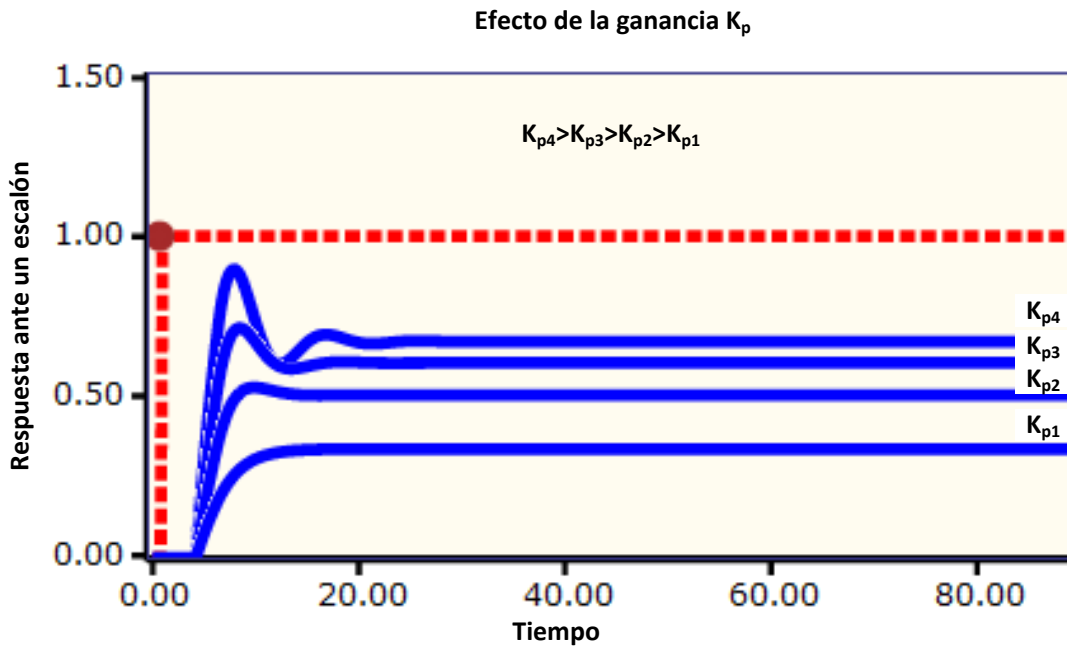


Figura 16. Efecto de la ganancia  $K_p$  de un controlador proporcional

### 3.5.4 Control Proporcional Integral

#### 3.5.4.1 Acción integral

En un controlador que emplea una acción de control integral, la señal de control se modifica a una velocidad proporcional a la señal de error; es decir, si la señal de error es grande, la señal de control se incrementa con gran rapidez; si es pequeña la señal de control se incrementa con lentitud. Este proceso puede representarse matemáticamente por:

$$\frac{du(t)}{dt} = \frac{1}{\tau_I} e(t) \text{ es decir, } u(t) = \frac{1}{\tau_I} \int_0^t e(t) dt$$

Denominándose tiempo integral a  $\tau_I$  (también se suele usar la nomenclatura  $T_I$  ó  $T_i$ ).

Por la propia definición de integral, son sistemas con “memoria”, ya que la salida en un instante dependerá del comportamiento anterior desde 0 hasta t. La Figura 17 muestra la acción de un controlador integral ante un error permanente y constante:

- Cuando la salida del controlador es constante, el error será cero.
- Cuando la salida del controlador varía a una velocidad constante, el error debe tener un valor constante.

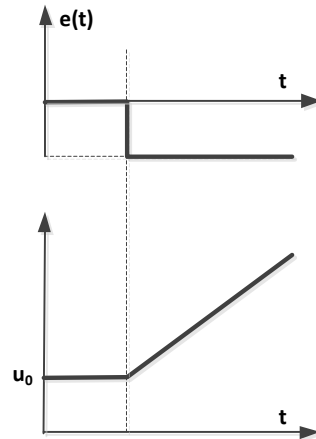


Figura 17. Controlador integral frente a error constante en régimen permanente

Hasta el momento en que se produce el error, el valor de la integral es constante, ya que  $u(t) = u_0$ . Cuando el error es constante, el área bajo la gráfica crece con el tiempo. Puesto que el área aumenta a una velocidad constante, la salida del controlador aumenta, también, a una velocidad constante.

Como se puede observar, siempre que exista error, la acción integral hará cambiar la salida del controlador para corregirlo, por lo que este controlador tiende a cancelar los errores en régimen permanente. Esta es la única ventaja de este tipo de controlador. Porque su utilización produce una respuesta más oscilatoria. Una acción integral corta implica que el controlador tratará de corregir relativamente pronto la desviación entre la consigna y el valor real. En un proceso con inercia, podría causar una oscilación inestable del sistema. Sin embargo, una acción integral relativamente larga es más segura pero implica lentitud de la respuesta. Como se muestra en la Figura 18, cuanto menor sea  $\tau_I$  más rápida e intensa será la acción de control pero aumenta las oscilaciones en el sistema.

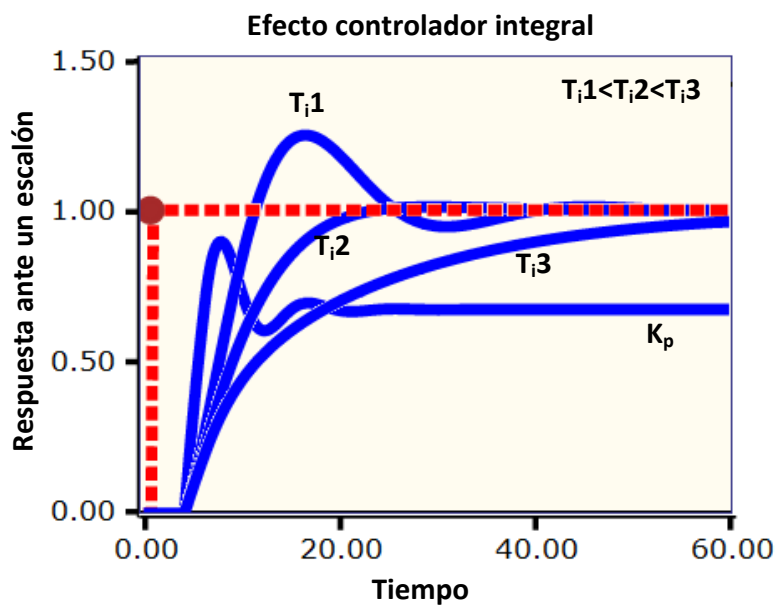




Figura 18. Efecto de acción integral

3.5.4.2 Controlador Proporcional Integral

Como se observa en la Figura 18 en la que se compara distintos controladores integrales con uno proporcional, la acción de control proporcional genera un desplazamiento respecto al estado estacionario, de forma que la variable a deseada no alcanzará el valor deseado. Por otra parte, se puede observar que aunque la acción integral anula el error por si misma, la respuesta del sistema es demasiado lenta si no se admiten comportamientos muy oscilatorios. Por estas razones se combinan las dos acciones obteniéndose un controlador Proporcional Integral o PI que representa las ventajas de cada uno de ellos. Se representa por:

$$u(t) = K_p e(t) + \frac{K_p}{\tau_I} \int_0^t e(t) dt$$

donde  $\tau_I$  regula la ganancia del integrador, y  $K_p$  ajusta el integrador y la ganancia proporcional. A la inversa del tiempo integral,  $1/\tau_I$  se le conoce como repeticiones por minuto, equivalente al número de veces que la acción integral repite el efecto de la acción proporcional ( $K_p = 1$ ) en un minuto. Además, la acción integral se utiliza para determinar de forma automática el valor correcto de  $u_o$  (ver Figura 19).

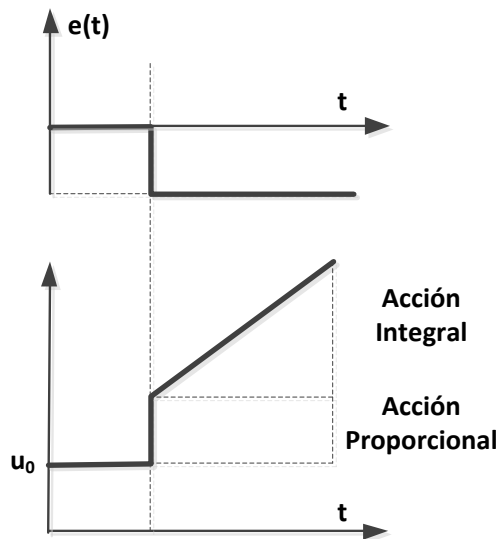


Figura 19. Controlador PI ante error permanente

Como se puede observar en la Figura 20, con un controlador PI se consigue evitar el error de offset o de régimen permanente, pero con una velocidad mayor que con un controlador I, con solo una acción integral. A mayor  $K_p$  del controlador PI, más rápido será el sistema, pero aparecen más oscilaciones.

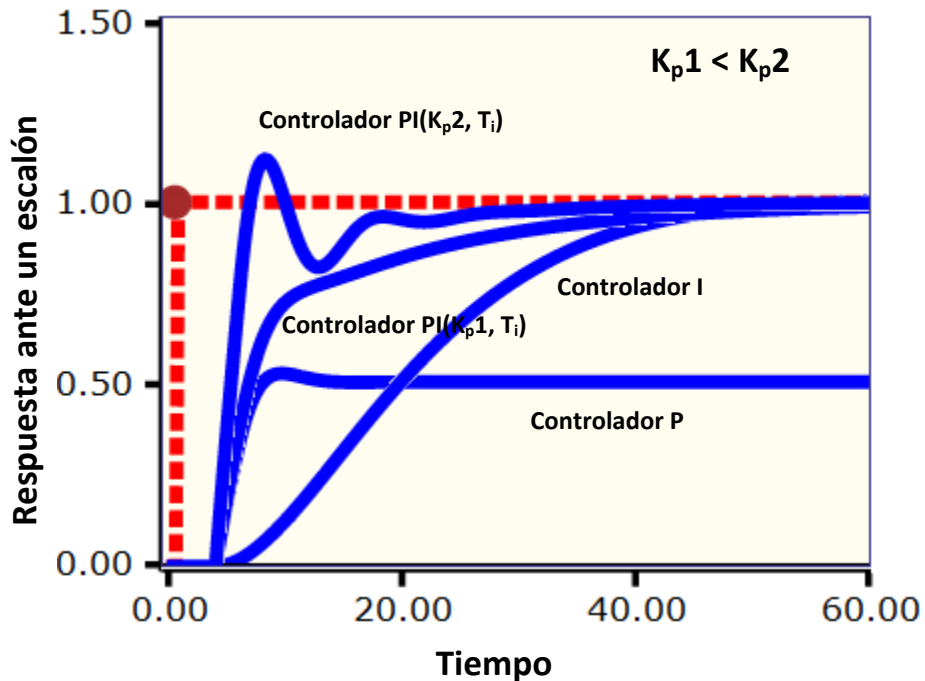


Figura 20. Efecto de los parámetros de un controlador PI

### 3.5.5 Control Proporcional Derivativo

#### 3.5.5.1 Acción Derivativa

En el control derivativo, el cambio en la salida del controlador es proporcional a la relación de cambio con respecto al tiempo, de la señal de error:

$$u(t) = \tau_D \cdot \frac{de(t)}{dt} + u_o$$

Al parámetro  $\tau_D$  ( $T_D$  ó  $T_d$ ) se le denomina tiempo derivativo.

Por el contrario, como se puede observar en la Figura 21, este tipo de controladores no responden a las señales de error en régimen permanente, ya que su derivada es cero, Por esta razón no se utilizan individualmente, sino que hay que combinarlos con alguna otra acción. El efecto de la acción de control derivativo es anticipar los cambios de error, y proporcionar una respuesta más rápida a esos cambios. La velocidad rápida de respuesta inherente al control derivativo permite al sistema salvar el tiempo de retardo en los procesos rápidos y estabilizarse en corto periodo de tiempo, especialmente cuando el error o la carga cambian constantemente. Cuando se tiene valores excesivamente estrechos de la acción derivada (poco tiempo de anticipo) no se realiza una buena predicción y no se eliminan los errores residuales.

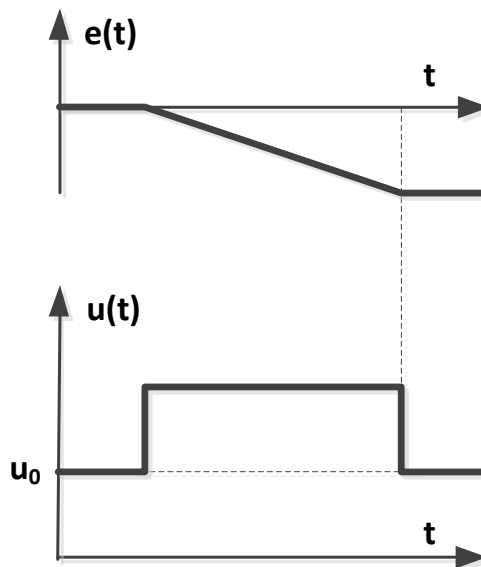


Figura 21. Respuesta de la acción derivativa frente a un error variable

Por el contrario, cuando se dispone de valores excesivamente altos de la acción derivativa, se puede producir saturación de los actuadores. Un valor adecuado de la acción derivativa será aquél que consigue la consigna con el mínimo número de oscilaciones.

### 3.5.5.2 Controlador Proporcional Derivativo

La acción de control derivativo proporciona una señal de control proporcional a la velocidad de cambio de la señal de error. Como se ha comentado en el apartado anterior, puesto que el controlador no genera ninguna salida a menos que el error cambie, no se suele utilizar como acción única. Se combina con una acción proporcional (Controlador PD) para eliminar los errores constantes, resultando una señal de control del siguiente tipo:

$$u(t) = K_p e(t) + K_p \tau_D \cdot \frac{de(t)}{dt} + u_0$$

$\tau_D$  representa el tiempo con que la acción derivativa se anticipa al efecto de la acción proporcional sobre el elemento final de control. La salida del controlador puede variar cuando hay un error cambiando constantemente. Hay un cambio inicial rápido en la salida del controlador a causa de la acción derivativa seguida de un cambio gradual debido a la acción proporcional. Por ejemplo, si se controla un sistema térmico en el que la temperatura cambia en forma de rampa, el elemento final de control se anticipará en dos segundos a la reacción que tendría normalmente por una acción proporcional únicamente (ver Figura 22).

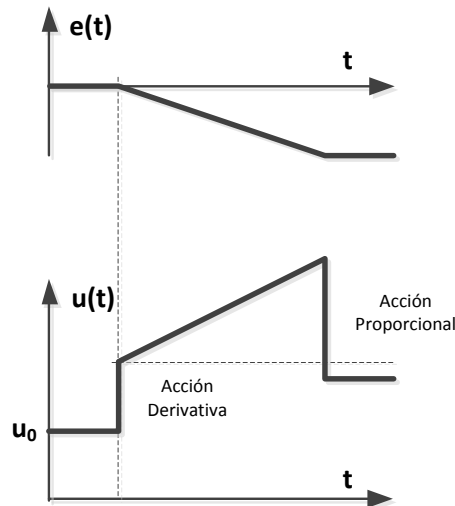


Figura 22. Control PD frente a un error variable

En teoría, la acción derivativa tiene un efecto positivo estabilizador sobre la dinámica del sistema de control, haciendo que se amortigüen las oscilaciones en las respuestas, o bien, que se pueda elevar la ganancia proporcional del controlador y con ello la velocidad, sin que se incremente las oscilaciones.

### 3.5.6 Control Proporcional Integral Derivativo

Si se combinan las tres acciones de control descritas en los apartados anteriores, surge el controlador PID, que no presenta error de desplazamiento y reduce la tendencia a las oscilaciones siempre que se seleccionen bien sus parámetros característicos. Se representa por la siguiente ecuación:  $u(t) = K_p e(t) + \frac{K_p}{\tau_I} \int_0^t e(t) dt + K_p \tau_D \cdot \frac{de(t)}{dt}$

Este control es el más general, y con toda probabilidad es el tipo de controlador más utilizado, ya que permite una óptima explotación de las características de los tres tipos de control. Se puede crear una respuesta de salida que sigue estrechamente la señal de entrada, sin intervalos o retardos en procesos lentos y rápidos, incluyendo aquéllos en los que la carga varía constantemente. Se puede considerar como un controlador proporcional, que dispone de un control integral para eliminar el error de desplazamiento y un control derivativo para reducir retardos de tiempo. En esta combinación:

- El control proporcional da forma a la curva de respuesta de la variable controlada, produciendo más salida cuanto mayor sea el error.
- El control integral disminuye el tiempo de existencia del error, produciendo más salida cuanto más tiempo perdure el error.
- El control derivativo disminuye el tiempo durante el que cambia el error, prediciendo, con antelación el valor del cambio del error, y frecuentemente, reduce el error ejecutando por adelantado las correcciones oportunas, produciendo más salida cuanto más rápidamente se produce el error (adelanto de la respuesta).

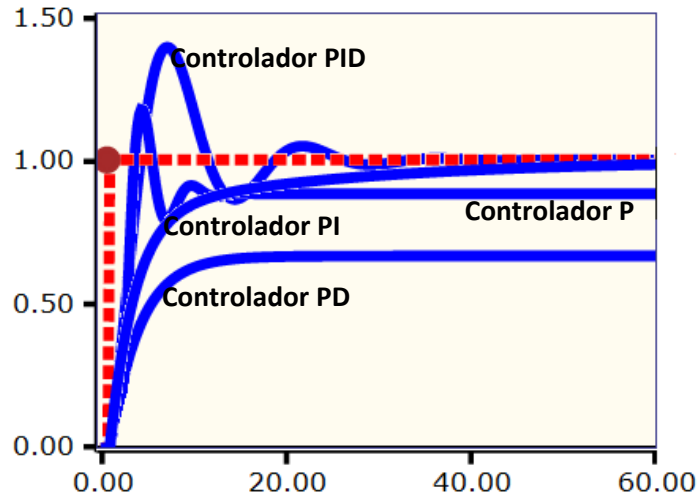


Figura 23. Comparación de controladores P, PI, PD y PID

Como se puede observar en la Figura 23 en la que se comparan los distintos tipos de controladores, se pueden realizar los siguientes comentarios:

- En general (dependerá del tipo de sistema que se esté controlando) solamente los controladores con acción integral son capaces de eliminar los errores en régimen permanente.
- Para grandes valores de  $K_p$ , la respuesta del sistema es más rápida pero presenta oscilaciones que se pueden reducir con una acción derivativa.

### 3.6 Introducción a los sistemas de control en tiempo discreto

Un computador es un sistema discreto, por lo que el diagrama de bloques básico de un sistema de control que lo utilice como controlador es el mostrado en la Figura 24.

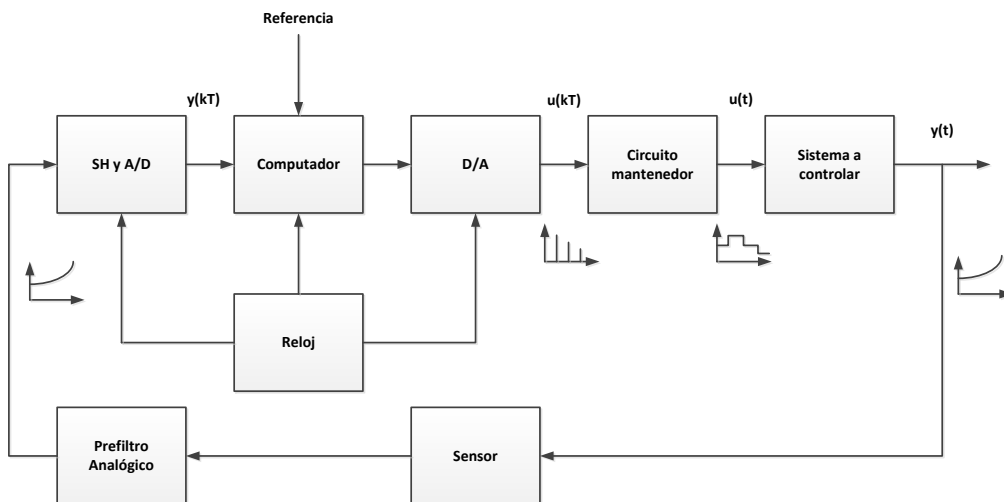


Figura 24. Sistema de control por computador

La salida del sistema a controlar es una señal en tiempo continuo,  $y(t)$ , que es medida por el sensor y sometida generalmente a un prefiltro analógico. Esta señal se convierte a digital cada  $T$  segundos (periodo de muestreo) mediante un circuito de muestreo y mantenimiento (SH, cuya denominación inglesa es Sample and Hold) y el convertidor analógico/digital (A/D), de forma que se obtiene una señal codificada en binario,  $y[kT]$ . En cada instante de muestreo se debe convertir un número codificado en una señal física de control, la cual normalmente es una señal en tiempo continuo por la propia naturaleza de los sistemas de actuación. El convertidor digital/analógico (D/A) y el circuito de retención convirtiendo la secuencia de números en una señal continua por secciones  $u(t)$  que controla la dinámica  $y(t)$  del sistema. El reloj en tiempo real del computador sincroniza todos los eventos. Como se puede observar, son necesarios los siguientes sistemas adicionales que no se utilizan en el control en tiempo continuo:

- Muestreador y mantenedor. Se trata de un circuito que recibe como entrada una señal analógica y mantiene dicha señal en un valor constante (valor medido) durante un tiempo específico.
- Convertidor analógico/digital. Es un dispositivo que convierte una señal analógica en una señal digital, generalmente codificada numéricamente en binario. Normalmente el sistema Muestreador/mantenedor es una parte integrada en un convertidor A/D comercial. Es necesario como interfaz entre un componente analógico y uno digital.
- Convertidor digital/analógico. Es un dispositivo que convierte una señal digital (datos codificados numéricamente) en una señal analógica. Es necesario como interfaz entre un componente digital y uno analógico. Comercialmente se integra con un circuito mantenedor que mantiene constante esa señal analógica hasta que cambia en el siguiente instante de muestreo.
- Prefiltro analógico. Este dispositivo se sitúa entre el sensor y el convertidor A/D y su función es reducir el ruido de alta frecuencia de los componentes de la señal analógica para evitar que el ruido cambie a una frecuencia menor debido al proceso de muestreo. A este problema se le denomina aliasing o el enmascaramiento de frecuencia que consiste en que al reconstruir la señal muestreada, las frecuencias mayores que la mitad de la frecuencia de muestreo aparecen como frecuencias más bajas.

### 3.7 Descripción de sistemas dinámicos en tiempo discreto

#### 3.7.1 Ecuación en diferencias

La dinámica de un sistema continuo se describe utilizando ecuaciones diferenciales. El comportamiento dinámico de un sistema digital o en tiempo discreto se describe con ecuaciones en diferencias. Por ejemplo, una ecuación en diferencias de segundo orden se representa como:

$$y[(k + 2)T] - a_1y[(k + 1)T] + a_0y[kT] = bu[(k + 2)T]$$

Donde  $y[kT]$  representa la salida del sistema en el instante de muestreo  $kT$ . Las ecuaciones en diferencias son muy fáciles de programar. La anterior ecuación se puede programar reorganizando la ecuación de modo que la incógnita a resolver sea  $y[(k + 2)T]$ :

$$y[(k + 2)T] = a_1y[(k + 1)T] - a_0y[kT] + bu[(k + 2)T]$$

Para resolver esta ecuación de forma recursiva, sólo hace falta conocer el valor de la entrada  $u$  y las condiciones iniciales  $y(0)$  e  $y(1)$ .

La relación entre una ecuación en diferencias y una ecuación diferencial puede observarse a través de un ejemplo. Supóngase la ecuación diferencial de primer orden:

$$\frac{dy(t)}{dt} + ay(t) = u(t)$$

Si las medidas de las variables del sistema se toman espaciadas  $T$  segundos, entonces la ecuación diferencial se puede aproximar como:

$$\frac{y[(k + 1)T] - y[kT]}{T} + ay[(k + 1)T] = u[(k + 1)T]$$

Si se despeja  $y[(k + 1)T]$  resulta:

$$y[(k + 1)T] = \frac{1}{1 + aT}y[kT] + \frac{1}{1 + aT}u[(k + 1)T]$$

Es evidente que el pasar de una ecuación diferencial a una en diferencias está condicionado por la aproximación que se haga de la derivada primera, segunda, etc.

#### 3.7.2 Transformada z y función de transferencia

La transformada de Laplace puede utilizarse para resolver sistemas de ecuaciones diferenciales ordinarias en tiempo continuo, pues permite pasar de una ecuación diferencial a una ecuación algebraica. Un método operacional equivalente para la resolución de sistemas de ecuaciones

en diferencias de tipo lineal en tiempo discreto, es el método pasado en la transformada Z. Dada una secuencia de números  $\{x(k)\}$ , se define la transformada Z como la transformada de Laplace de una secuencia. Expresando la secuencia como:

$$\{x_k\} = \sum_{k=-\infty}^{\infty} x_k \delta(t - kT)$$

Donde  $\delta(t)$  es la delta de Dirac, la cual no es una función en el sentido analítico del término, sino que su significación viene dada por las propiedades que tiene (esto se estudia con rigor usando la Teoría de las Distribuciones). En la expresión anterior, representa un tren de pulsos que modula los valores de la secuencia. Por tanto, la transformada z se calcula como la transformada de Laplace de la secuencia anterior, y vendrá dada por:

$$L[\{x_k\}] = \sum_{k=-\infty}^{\infty} x_k e^{-skT} = \sum_{k=-\infty}^{\infty} x_k e^{-k} = Z[\{x_k\}] \quad \text{donde } z = e^{sT}$$

La transformada z definida así se conoce como transformada z unilateral, y en ella se supone que la secuencia  $\{x(k)\} = 0$  para todo  $k < 0$ .

Al igual que la transformada de Laplace, la transformada z tiene propiedades de linealidad traslación real, traslación compleja, teorema del valor inicial, teorema del valor final, convolución real, etc.

### 3.7.3 Función de transferencia de sistemas con mantenedor

A la hora de obtener la función de transferencia en Z de un sistema continuo que se va a controlar utilizando un computador, hay que tener en cuenta que el dispositivo mantenedor de orden cero (MOC) tiene una influencia en la respuesta dinámica del sistema. Como su propio nombre indica, el dispositivo mantenedor lo que hace es fijar la señal de control durante un periodo de muestre a la que tiene al inicio del periodo. En ese sentido, la función de transferencia en s de un dispositivo mantenedor puede establecerse como la de un escalón unitario menos la de un escalón unitario retrasado un periodo de muestreo  $G_{MOC}(z) = \frac{1-e^{-sT}}{s}$ .

Nótese que el numerador de la función de transferencia del mantenedor corresponde en dominio Z a  $(1 - z^{-1})$ , dado que  $z^{-1}$  opera como un retardo de un periodo de muestreo. Analíticamente, la función de transferencia de un sistema dinámico incluyendo el mantenedor de orden cero (equivalente de la planta en tiempo discreto), se calcula siguiendo el siguiente procedimiento:

1. Calcular la transformada inversa de Laplace de la función de transferencia  $G(s)/s$ :

$$g'(t) = L^{-1} \left[ \frac{G(s)}{s} \right].$$

2. Muestrear la función continua  $g'(t)$  y obtener la secuencia:  $g'_k = g'(kT)$ .



3. Calcular la transformada z de la secuencia obtenida  $G'(z) = Z[\{g'_k\}]$ .

La función de transferencia del equivalente en tiempo de muestreo de la planta se obtiene multiplicando  $G'(z)$  por  $(1 - z^{-1})$ :  $G(z) = (1 - z^{-1})G'(z)$ , que es equivalente a  $G(z) = (1 - z^{-1})Z\left(\frac{G(s)}{s}\right)$ .

### 3.8 Respuesta temporal de sistemas en tiempo discreto

Se van a tratar someramente en este apartado los aspectos relacionados con la respuesta temporal de sistemas en tiempo discreto. Los desarrollos son similares a los realizados para sistemas de tiempo continuo, por lo que no se entrará en detalles en muchos aspectos.

La respuesta temporal de sistemas en tiempo discreto se puede obtener resolviendo la ecuación en diferencias o bien utilizando la transformada Z y haciendo uso de tablas de inversas de la transformada. Un aspecto fundamental en la discretización y la obtención de la respuesta temporal es la selección de un periodo de muestreo adecuado para obtener reconstruir la señal medida de forma adecuada en el computador y enviar señales de control con la frecuencia adecuada a las características dinámicas del proceso.

#### 3.8.1 Respuesta temporal de sistemas de primer orden

La función de transferencia de un sistema de primer orden en tiempo discreto teniendo en cuenta el mantenedor de orden cero viene dada por:

$$G(s) = \frac{k}{1 + \tau s} \rightarrow G(z) = \frac{k \left(1 - e^{-\frac{T}{\tau}}\right) z^{-1}}{\left(1 - z^{-1} e^{-\frac{T}{\tau}}\right)} = \frac{b z^{-1}}{1 - a z^{-1}} = \frac{b}{z - a}$$

$$a = e^{-\frac{T}{\tau}}; b = k(1 - a)$$

La respuesta en escalón es de tipo exponencial, como la de un sistema de primer orden. El sistema será estable si  $|a| < 1$ .

#### 3.8.2 Respuesta temporal de sistemas de segundo orden

La función de transferencia de un sistema de segundo orden en tiempo discreto teniendo en cuenta el mantenedor de orden cero viene dada por:

$$G(s) = \frac{K \omega_n^2}{s^2 + 2\delta \omega_n s + \omega_n^2} \rightarrow G(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2}$$

El sistema será estable si el módulo de las raíces del denominador es menor que 1.

### 3.9 Diseño de controladores en tiempo discreto por conversión de controladores continuos

Existen un gran abanico de métodos de diseño de controladores en tiempo discreto que se pueden agrupar en dos clases fundamentales: métodos que calculan el controlador en tiempo discreto a partir de diseños continuos o bien métodos de diseño directo en el plano Z. Cada una de estas clase incluyen diversos métodos, pero a nosotros nos interesa el diseño por conversión de controladores continuos que es el que se ha implementado en las herramientas interactivas.

#### 3.9.1 Consideraciones generales en la discretización de controladores continuos

En este apartado se va a describir el diseño de controladores en tiempo discreto a partir de un controlador continuo, concretamente un PID, diseñado a partir del modelo continuo del proceso. Aproximando su ecuación diferencial característica mediante una ecuación en diferencias discreta.

Si se desea controlar mediante un controlador PID un sistema descrito por una función de transferencia  $G(s)$ , para calcular la función de transferencia del controlador  $C(z)$  hay que realizar los siguientes pasos:

1. Obtener un PID continuo  $C(s)$  a partir de las especificaciones de entrada (ver Figura 25), verificando que se cumplen. Éste queda descrito por sus parámetros característicos  $K_p, T_i$  y  $T_d$ .

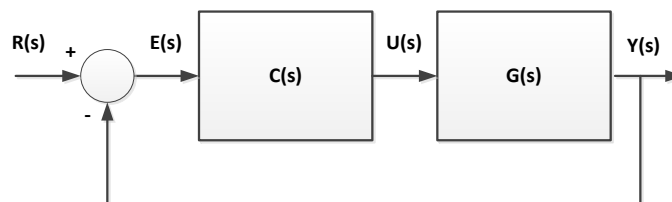


Figura 25. Sistema de control en tiempo continuo

2. Discretizar  $C(s)$  con alguna metodología, obteniendo  $C(z)$ . En este caso, queda descrito por  $K_p, T_i$  y  $T_d$ , además del periodo de muestreo  $T$ . En un primer paso, se sustituye el controlador en tiempo continuo por el de tiempo discreto (ver Figura 26).

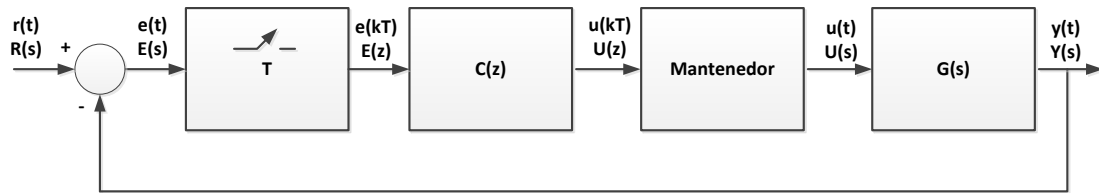


Figura 26. Sustitución del controlador en tiempo continuo por el controlador en tiempo discreto

En un segundo paso, habría que modificar el sistema para que la consigna, el error y la acción de control fueran secuencias, obteniéndose el esquema fundamental del sistema de control por computador, tal y como se muestra en la Figura 27.

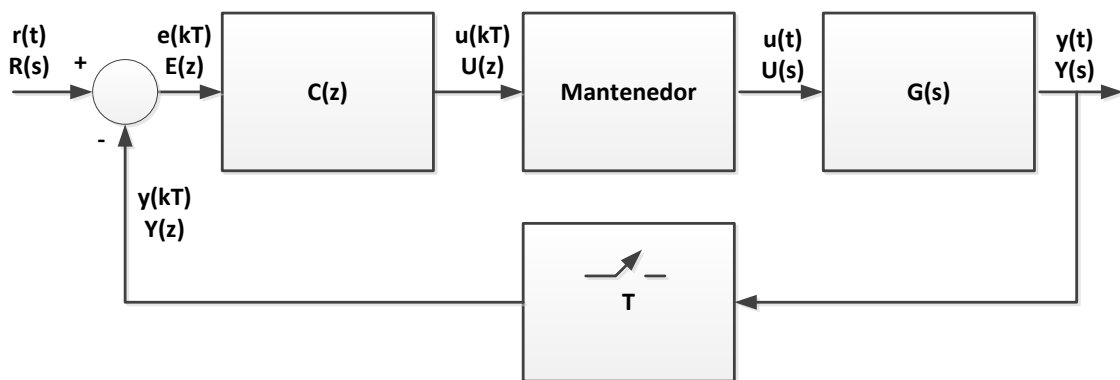


Figura 27. Esquema general de un sistema de primer de control por computador

Si se parte de un controlador continuo tipo PID no interactivo, descrito por la siguiente ecuación diferencial:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \rightarrow C(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

Para discretizarlo, se pueden utilizar varios métodos. Uno de ellos es el de sustitución de operadores que consiste en la sustitución de los operadores de un PID por operadores discretos equivalentes que se obtienen por métodos de análisis numérico. Con este método hay que realizar un nuevo estudio de estabilidad en función de la sustitución empleada.

La acción proporcional tiene una implementación discreta directa ya que multiplica cada muestra de la señal de error por un valor de la ganancia  $K_p$ , es decir:

$$u_p(t) = K_p(r(t) - y(t)) \rightarrow u_p[kT] = K_p(r[kT] - y[kT]) \rightarrow C_p(z) = K_p$$

En los siguientes apartados se estudiarán los métodos numéricos empleados para la implementación de los otros dos operadores necesarios: acción integral y acción derivada.

### 3.9.2 Discretización de la operación integral

Sea un sistema cuya salida es la integral de la entrada, es decir, un sistema cuya función de transferencia  $C_I(s) = 1/s$ , como el que se muestra en la Figura 28.

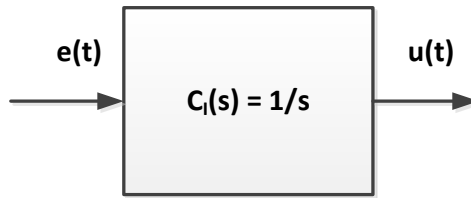


Figura 28. Integrador

La integral de una determinada señal entre dos instantes de tiempo coincide con el área de la superficie que encierra hasta el eje horizontal, por lo que para aproximar esta operación basta con aproximar el cálculo del área. Esto, generalmente, se puede realizar de tres formas posibles (aproximación rectangular hacia delante, aproximación rectangular hacia atrás y aproximación trapezoidal), pero a nosotros solo nos interesa la aproximación trapezoidal que es la que se ha implementado en el proyecto.

En la aproximación trapezoidal se utiliza una aproximación de forma que el área se calcula como sumas sucesivas de trapecios tal y como se puede observar en la Figura 29(a). A esta aproximación se le denomina regla de Tustin o aproximación bilineal. Según esta figura, el área en el instante  $kT$  será igual a la comprendida por el trapecio entre los instantes  $(k-1)T$  y  $kT$  (área del triángulo entre los instantes  $(k-1)T$  y  $kT$  más el área  $e[(k-1)T]T$  correspondiente al rectángulo inferior, como se muestra en la Figura 29(b)), resultando:

$$u[kT] = \sum_{n=1}^k \left\{ T \frac{e[nT] - e[(n-1)T]}{2} + T e[(n-1)T] \right\} = \sum_{n=1}^k T \frac{e[nT] + e[(n-1)T]}{2}$$

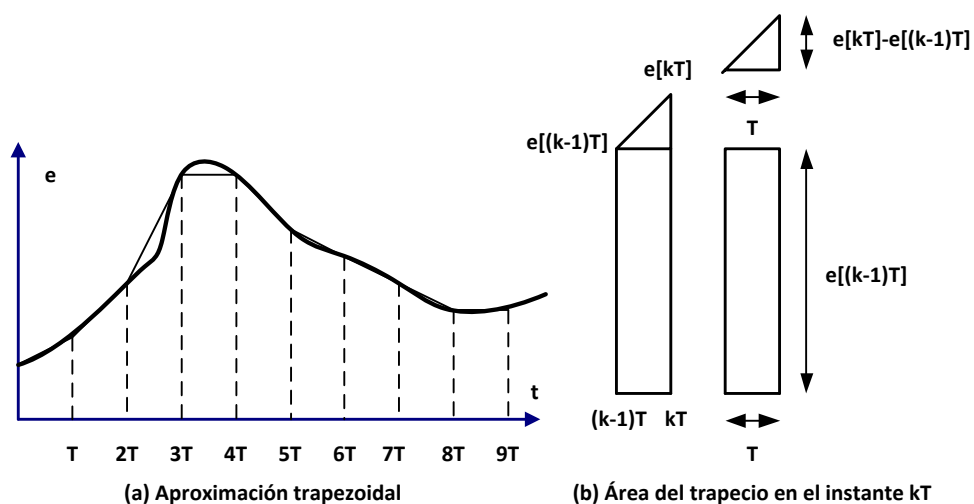


Figura 29. Aproximación trapezoidal

Se observa que en el instante  $kT$  el área encerrada por la curva hasta ese momento coincide con la suma del área encerrada hasta el instante anterior  $(k-1)T$  junto con la del nuevo trapecio, es decir:

$$u[kT] = \sum_{n=1}^k T \frac{e[nT] + e[(n-1)T]}{2} = u[(k-1)T] + T \frac{e[kT] + e[(k-1)T]}{2}$$

Si se aplica la transformada Z a la expresión resulta:

$$U(z) = z^{-1}U(z) + T \frac{E(z) + z^{-1} E(z)}{2} \rightarrow (1 - z^{-1})U(z) = \frac{T}{2} (1 + z^{-1}) E(z)$$

Por lo que la función de transferencia en tiempo discreto del sistema será:

$$C_I(z) = \frac{U(z)}{E(z)} = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} = \frac{T}{2} \frac{z + 1}{z - 1}$$

En este caso, la relación entre s y z resulta ser:

$$s \approx \frac{T}{2} \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{T}{2} \frac{z - 1}{z + 1}$$

### 3.9.3 Discretización de la operación derivada

La operación derivada en el instante de tiempo  $kT$  se debe realizar aproximando la pendiente de la curva del error en ese instante y, al igual que en la operación integral, se puede utilizar varios métodos. Por ejemplo, se puede aproximar la derivada de la señal de error por la pendiente de la recta formada entre las muestras  $(k-1)T$  y  $kT$  de la señal de error discretizada.

Sea una sistema cuya salida es la derivada de la entrada, es decir, un sistema cuya función de transferencia es  $C_D(s) = s$ , como el que se muestra en la Figura 30. Si se observa la Figura 31, la derivada de una señal  $e(t)$  en el instante de tiempo  $kT$  se define como la tangente del ángulo  $\beta$ , es decir:

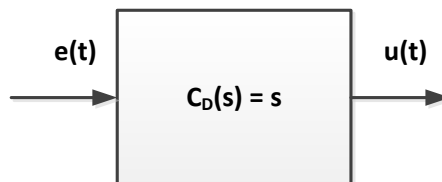


Figura 30. Derivador

$$tg(\beta) = \left. \frac{d e(t)}{dt} \right|_{t=kT}$$

Por otra parte, el ángulo que forma la recta que une dos muestras consecutivas con respecto a la horizontal viene dado por:

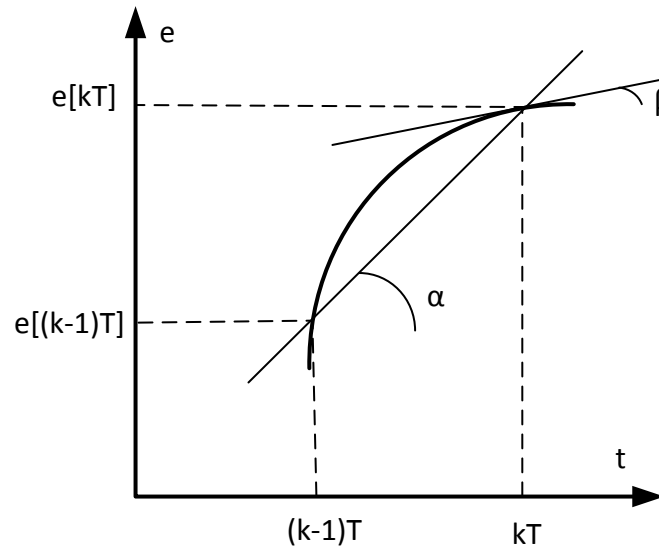


Figura 31. Aproximación de la operación derivada

$$tg(\alpha) = \frac{e[kT] - e[(k-1)T]}{T}$$

Si  $T$  es muy pequeña, tendiendo a cero, se puede determinar que:

$$tg(\alpha) \approx tg(\beta)$$

Por lo tanto:

$$u[kT] = \left. \frac{d e(t)}{dt} \right|_{t=kT} = \frac{e[kT] - e[(k-1)T]}{T}$$

Si se aplica la transformada  $Z$  a la expresión resulta:

$$C_D(z) = \frac{U(z)}{E(z)} = \frac{1 - z^{-1}}{T} = \frac{1}{T} \frac{z - 1}{z}$$

Por lo tanto, si  $C_D(s) = s$  se puede decir que con esta aproximación, la relación entre  $s$  y  $z$  es:

$$s \approx \frac{1 - z^{-1}}{T} = \frac{1}{T} \frac{z - 1}{z}$$

### 3.9.4 Discretización de controladores PID

Sea el controlador PID descrito por su ecuación diferencial característica:

$$u(t) = K_p \left[ e(t) + \frac{1}{\tau_I} \int_0^t e(t) dt + \tau_D \cdot \frac{de(t)}{dt} \right]$$

Para discretizarlo, se utiliza el método de sustitución de operadores de forma que se sustituye la variable  $s$  por su correspondiente relación con  $z$  en función de la aproximación utilizada. Por tanto, el primer paso es aplicar la transformada de Laplace a la ecuación anterior, obteniéndose:

$$C(s) = \frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

Utilizando las ecuaciones desarrolladas en apartados anteriores que nos proporcionan la relación entre  $s$  y  $z$ ;

$$s \approx \frac{T}{2} \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{T}{2} \frac{z - 1}{z + 1} \quad s \approx \frac{1 - z^{-1}}{T} = \frac{1}{T} \frac{z - 1}{z}$$

Operador integral trapezoidal    Operador derivada

Sustituyendo estas expresiones en la función de transferencia del controlador PID, se obtiene:

$$C(s) = \frac{U(s)}{E(s)} \approx K_p \left( 1 + \frac{1}{T_i} \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} + T_d \frac{1 - z^{-1}}{T} \right)$$

Operando y agrupando los términos en función de un polinomio en  $z^{-1}$ , se alcanza la siguiente expresión, que es la función de transferencia de un controlador PID en tiempo discreto:

$$C(s) = \frac{U(s)}{E(s)} \approx \left[ \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{1 - z^{-1}} \right]$$

donde:

$$q_0 = K_p \left( 1 + \frac{T}{2T_i} + \frac{T_d}{T} \right)$$

$$q_1 = K_p \left( -1 + \frac{T}{2T_i} - 2 \frac{T_d}{T} \right)$$

$$q_2 = K_p \left( \frac{T_d}{T} \right)$$





# 4

## TECNOLOGÍAS BASADAS EN WEB

Este proyecto se ha realizado utilizando tecnologías basadas en Web y utilizando estándares creados por W3C. El World Wide Web Consortium (W3C), es una comunidad internacional que desarrolla estándares abiertos para asegurar el mantenimiento de la Web. Su misión es potenciar el crecimiento de la web mediante el desarrollo de protocolos y guías. Esto hace posible el acceso a la web en cualquier lugar, tiempo y dispositivo que sea usado, desde los computadores tradicionales hasta teléfonos y otros dispositivos móviles.

En las siguientes subsecciones veremos los distintos estándares desarrollados por W3C que se han utilizado en la realización del proyecto haciendo una breve descripción de su función, viendo su evolución y describiendo más detalladamente ciertas características que son más interesantes desde el punto de vista de la ejecución de este proyecto [35].

### 4.1 HTML, XHTML y HTML5

Casi con toda seguridad, el protocolo más usado en la Web es HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y los datos que principalmente transporta están usando HTML (HyperText Markup Language, Lenguaje de Etiquetado de HiperTexto), en alguna de sus distintas especificaciones. En este apartado se introducen las distintas especificaciones de este lenguaje [39][43][40].

#### 4.1.1 Que es HTML

HTML es el lenguaje para describir la estructura de las páginas Web dándole a los autores las capacidades de:

- Publicar documentos online con cabeceras, textos, tablas, listas, fotos, etc.
- Obtener información online vía enlaces de hipertexto, con un click de ratón.
- Diseñar formularios para dirigir transacciones con servicios remotos, para el uso en búsqueda de información, realizar reservas, encargar productos, etc.
- Incluir vídeos, audio, y otras aplicaciones directamente en los documentos.

HTML permite establecer la estructura visual del proyecto, creando marcos donde se dibujaran gráficos, añadir textos, y crear botones, cuadros de texto y deslizadores (sliders) para manipular las herramientas interactivas.

La mayoría de la gente está de acuerdo que los documentos HTML deberían funcionar igual de bien independientemente de los navegadores o la plataforma utilizados. Esto permite interoperabilidad a bajo coste entre los proveedores de contenidos haciendo que estos tengan que desarrollar solo una versión del mismo documento.

Cada especificación de HTML ha intentado reflejar un gran consenso entre los actores de la industria para que la inversión realizada por los proveedores de contenidos no sea en vano y sus documentos no dejen de ser interpretados correctamente en un corto periodo de tiempo.

HTML ha sido desarrollado con la visión de que todo tipo de dispositivos pudieran ser capaces de usar la información de la Web: computadores personales con pantallas gráficas de distinta resolución y profundidad de color, teléfonos y computadores con alto o bajo ancho de banda.

### 4.1.2 Historia de HTML

HTML fue originalmente desarrollado por Tim Berners-Lee en el CERN (European Organization for Nuclear Research, Organización Europea para la investigación Nuclear), y popularizado por el navegador Mosaic desarrollado en el NCSA (National Center for Supercomputing Applications, Centro Nacional de Aplicaciones de Supercomputación). Durante el curso de 1990 alcanzó su plenitud con el crecimiento explosivo de la Web. Durante este tiempo, HTML fue extendido de muchas formas. La web dependía de páginas Webs de autores y vendedores compartiendo las mismas convenciones para HTML. Esto motivó el trabajo conjunto en especificaciones para HTML.

HTML 2.0 (Noviembre de 1995) fue desarrollado bajo el auspicio del IETF (Internet Engineering Task Force, Grupo Especial sobre Ingeniería de Internet) para codificar las prácticas comunes a finales de 1994. HTML+ (1993) y HTML 3.0 (1995) proponían versiones mucho más ricas de HTML. A pesar de que estas versiones lideraron la adopción de nuevas características, nunca recibieron consenso en las discusiones de los estándares. Los esfuerzos del World Wide Web Consortium's HTML Working Group para codificar las prácticas comunes en 1996 dió como resultado HTML 3.2 (Enero de 1997). Siguiéndola HTML4, que se completó en 1998.

### 4.1.3 Que es XHTML

XHTML es una variante de HTML que usa la sintaxis de XML (eXtensible Markup Language, Lenguaje de Etiquetado eXtensible). XHTML tiene los mismos elementos que las variantes HTML, pero su sintaxis es ligeramente diferente. A causa de que XHTML se ajusta a las especificaciones de XML, es posible usar otras herramientas XML con XHTML. Otra ventaja respecto a HTML es que si un documento es transmitido con tipo MIME HTML (el cual es text/html), entonces éste será procesado como un documento HTML por el navegador web. En cambio si es transmitido con tipo MIME XML (el cual es application/xhtml+xml), entonces es tratado como un documento XML por el navegador web y será analizado sintácticamente por un procesador XML. Los autores están de acuerdo que un procesamiento para XML y HTML difiere; Por ejemplo, un documento en donde no se cierran todas las etiquetas, como puede ser un salto de línea (<br>), se mostrará correctamente en un navegador si se indica que es de tipo HTML. Sin embargo, si se indica que el documento es XML, le detendrá el renderizado ante la primera etiqueta que no haya sido cerrada correctamente.

#### 4.1.4 Historia de XHTML

HTML4 fue completado en 1998, en ese momento, los miembros de W3C decidieron parar de evolucionar HTML y empezar a trabajar sobre un equivalente basado en XML, llamado XHTML. Estos esfuerzos empezaron con una reformulación de HTML4 en XML, conocido como XHTML 1.0, el cual no añadió ninguna característica nueva excepto la serialización, y fue completada en 2000. Después de XHTML 1.0, W3C se concentró en cambiarlo para hacer más fácil que otros grupos de trabajo se extendieran a XHTML, bajo el nombre de Modularización XHTML. En paralelo con esto, el W3C también trabajó en un nuevo lenguaje que no fue compatible con los anteriores lenguajes HTML y XHTML, llamándose XHTML2.

#### 4.1.5 Que es HTML5

En un futuro HTML5 reemplazará a DOM2 HTML, HTML4, y XHTML1. HTML5 es aún un borrador pero ya está siendo implementado por la mayoría de los principales navegadores web. Véase [5].

#### 4.1.6 Historia de HTML5

La idea de que la evolución de HTML debería ser reabierta fue probado en un taller del W3C en 2004, cuando algunos de los principios que sustentan el trabajo de HTML5, fueron presentados al W3C conjuntamente por Mozilla y Opera. La proposición fue rechazada sobre la base de que la proposición entraba en conflicto con la dirección tomada anteriormente para la evolución de la Web; el personal y afiliados de W3C votaron por continuar desarrollando un reemplazo basado en XML.

Poco después de eso, Apple, Mozilla, y Opera anunciaron conjuntamente su intento para continuar trabajando en el esfuerzo bajo el amparo de una nueva sede llamada WHATWG (Web Hypertext Application Technology Working Group). Se creó una lista de correo pública y se colgó un borrador en la página web de WHATWG. Los derechos de autor fueron modificados posteriormente para ser propiedad conjunta de los tres proveedores, y para permitir la reutilización de la especificación.

El WHATWG fue basado en algunos principios básicos, en particular, que las tecnologías necesitaban ser compatibles con versiones anteriores, que las especificaciones e implementaciones deben coincidir incluso si esto significa cambiar la especificación en vez de las implementaciones, y que las especificaciones tienen que ser lo suficientemente detalladas para que las implementaciones puedan lograr la completa interoperabilidad sin necesidad de usar ingeniería inversa uno del otro.

El anterior requerimiento en particular requería que el alcance de la especificación de HTML5 incluya lo que había sido anteriormente especificado en tres documentos separados: HTML4, XHTML1, y DOM2 HTML. Esto también significó incluir detalles mucho más significativos de los que anteriormente habían considerado la norma.

En 2006, el W3C, finalmente mostró un interés en participar en el desarrollo de HTML5 y, en 2007, formó un grupo de trabajo dedicado para trabajar con la WHATWG en el desarrollo de la especificación de HTML5. Apple, Mozilla, y Opera permitieron al W3C publicar la especificación

bajo los derechos de autor de W3C, mientras mantenían una versión con una licencia menos restrictiva en el sitio de WHATWG.

### 4.1.7 Mejoras HTML5

#### 4.1.7.1 Sintaxis

HTML5 define una sintaxis HTML que es compatible con documentos publicados en HTML4 y XHTML1 en la Web, pero no son compatibles con las características más esotéricas SGML de HTML4, como el procesamiento de instrucciones y el marcado recortado que no son soportadas por la mayoría de agentes de usuario. Los documentos que usan la sintaxis HTML son casi siempre servidos con el tipo de medio text/html.

HTML5 también define unas normas de análisis detalladas para esta sintaxis la cual es ampliamente compatible con las implementaciones populares. Los agentes de usuario deben usar estas normas para recursos que tienen el tipo de medio text/html. Este es un ejemplo de documento conforme a la sintaxis HTML:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
```

La otra sintaxis que puede ser usada para HTML5 es XML. Esta sintaxis es compatible con documentos e implementaciones XHTML1. Los documentos que usan esta sintaxis necesitan ser servidos con el tipo de medio XML y los elementos necesitan ser puestos en el espacio de nombres <http://www.w3.org/1999/xhtml>, siguiendo las reglas establecidas por las especificaciones XML. Debajo se muestra un ejemplo de un documento que esta creado con la sintaxis XML de HTML5:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Example document</title>
  </head>
  <body>
    <p>Example paragraph</p>
  </body>
</html>
MathML y SVG
```

La sintaxis de HTML5 permite el uso de elementos MathML y SVG para que sean usados dentro de un documento. Se ve de forma más fácil mediante un ejemplo:

```
<!doctype html>
<title>SVG in text/html</title>
<p>
  A green circle:
  <svg><circle r="50" cx="50" cy="50" fill="green"/></svg>
</p>
```

Combinaciones más complejas son posibles. Por ejemplo, en un elemento SVG se pueden anidar elementos MathML, HTML, o ambos, quedando dentro de este fragmento SVG que está a su vez, dentro de HTML.

#### 4.1.7.2 Nuevos elementos

Se han introducido los siguientes elementos para mejorar la estructura: *section*, *article*, *aside*, *hgroup*, *header*, *footer*, *nave*, *figure* y *figcaption*.

También hay otros elementos: *track*, *embed*, *mark*, *progress*, *meter*, *time*, *ruby*, *rt*, *rp*, *bdi*, *wbr*, *canvas*, *command*, *details*, *summary*, *detalist*, *keygen* y *output*.

Además, se han introducido los elementos *video* y *audio* para contenido multimedia. Ambos ofrecen una API con la que los autores de aplicaciones pueden realizar sus propias interfaces, pero también hay una forma de activar una interfaz de usuario proporcionada por el agente de usuario. El elemento *source* es usado junto con estos elementos si hay múltiples fuentes de origen disponibles de distintos tipos.

El atributo *type* de los elementos de input tiene ahora los nuevos valores siguientes: *tel*, *search*, *url*, *email*, *datetime*, *date*, *month*, *week*, *time*, *datetime-local*, *number*, *range*, *color*.

#### 4.1.7.3 Nuevos atributos

HTML5 ha introducido algunos atributos nuevos para varios elementos que ya formaban parte de HTML4:

- Los elementos *a* y *area* ahora tienen un atributo *media* por consistencia con el elemento *link*.
- El elemento *area*, por consistencia con los elementos *a* y *link*, ahora también tiene los atributos *hreflang*, *type* y *rel*.
- El elemento *base* puede ahora tener un atributo *target* también, principalmente para consistencia con el elemento *a*.
- El elemento *meta* tiene ahora un atributo *charset*, puesto que este es ya ampliamente utilizado y permite una buena forma para especificar la codificación de caracteres para el documento.

- Un nuevo atributo *autofocus* puede ser especificado sobre los elementos *input*, *select*, *textarea* y *button*.
- Un nuevo atributo *placeholder* puede ser especificado sobre los elementos *input* y *textarea*. Esto representa una pista para ayudar al usuario con los datos de entrada.
- El nuevo atributo *form* para los elementos *input*, *output*, *select*, *textarea*, *button*, *label*, *object* y *fieldset* permite a los controles ser asociados con un formulario. Estos elementos pueden ser ubicados en cualquier lugar de la página, no solo como descendientes del elemento *form*, y seguir estando asociados con un *form*.
- El nuevo atributo *required* aplicado a los elementos *input*, *select* y *textarea*. Este indica que el usuario tiene que rellenarlos con un valor para poder enviar el formulario.
- El elemento *fieldset* ahora permite el atributo *disable* por el cual, los elementos que son descendientes de este se deshabilitan y el atributo *name* el cual puede ser usado para acceder por *script*.
- Los elementos *input* tiene algunos atributos nuevos para especificar restricciones: *autocomplete*, *min*, *max*, *multiple*, *pattern* y *step*.
- El elemento *textarea* también tiene dos nuevos atributos, *maxlength* y *wrap* los cuales controlan la longitud máxima de la entrada y el comportamiento de la línea de envoltura presentada, respectivamente.
- El elemento *form* tiene el atributo *novalidate* que puede ser usado para desactivar la validación en el formulario al enviarse.
- Los elementos *input* y *button* tienen los nuevos atributos *formaction*, *formenctype*, *formmethod*, *formnovalidate* y *formtarget*. Si están presentes, estos sustituyen a los atributos *action*, *enctype*, *method*, *novalidate* y *target* del elemento *form*.
- El elemento *menú* tiene dos nuevos atributos: *type* y *label*. Estos permiten al elemento ser transformado en un menú, tal y como se encuentra en la interfaces de usuarios típicas como las ofrecidas por menús contextuales en conjunto con el atributo *contextmenu*.
- El elemento *style* tiene un nuevo atributo *scoped* el cual puede ser usado para activar el ámbito de las hojas de estilos.
- El elemento *script* tiene un nuevo atributo llamado *async* que influye en el *script* a la hora ser cargado y ejecutado.

- El elemento *html* tiene un nuevo atributo llamado *manifest* que señala hacia un manifiesto en la caché de la aplicación que puede ser usada en conjunto con la API para aplicaciones Web offline.
- El elemento *link* tiene un nuevo atributo llamado *sizes*. Este puede ser usado en conjunto con *icon* para indicar el tamaño del icono referenciado. Permitted de esta forma iconos con distintas dimensiones.
- El elemento *ol* tiene un nuevo atributo llamado *reversed*. Cuando esta presente, este indica que la lista está en orden descendente.
- El elemento *iframe* tiene tres nuevos atributos llamados *sandbox*, *seamless* y *srcdoc* los cuales permiten contenido en caja de arena (sandboxing content).

Algunos atributos de HTML4 ahora se pueden aplicar a todos los elementos. Estos son llamados atributos globales: *accesskey*, *class*, *dir*, *id*, *lang*, *style*, *tabindex* y *title*.

Hay también algunos atributos globales nuevos:

- El atributo *contenteditable* indica que el elemento es una área editable. El usuario puede cambiar el contenido del elemento y manipular el etiquetado.
- El atributo *contextmenu* puede ser usado para apuntar hacia un menú de contexto ofrecido por el autor.
- La colección *data-\** de atributos definidos por el autor. Los autores puede definir cualquier atributo que deseen, mientras lleve el prefijo *data-* para evitar confusiones con futuras versiones de HTML. El único requerimiento sobre estos atributos es que no pueden ser usado por extensiones de agentes de usuario.
- Los atributos *draggable* y *dropzone* pueden ser usados junto con la nueva API *drag & drop*.
- El atributo *hidden* indica que un elemento no es aún relevante, o no lo es más.
- La colección de atributos *role* y *aria-\** los cuales pueden ser usados para instruir la tecnología de asistencia.
- El atributo *spellcheck* permite sugerir si el contenido puede ser usado por el comprobador de ortografía o no.



HTML5 también crea todos los atributos de manejadores de eventos existentes en HTML4, los cuales tienen la forma *onevent-name*, los atributos globales y añade algunos atributos de manejadores de eventos nuevos para los nuevos eventos definidos.

#### 4.1.8 Uso de HTML, XHTML y HTML5 en el proyecto

HTML se ha usado para crear los distintos elementos que forman las herramientas interactivas que se pueden crear con la librería implementada. En el proyecto se ha utilizado HTML5 con la sintaxis de HTML, como ya se ha comentado en secciones anteriores HTML5 incorpora novedades a HTML4, aunque solo se han añadido un tipo de elemento de HTML5 como se mostrara más adelante; el resto de elementos utilizados ya aparecían en HTML4. Ya que, durante la implementación de este proyecto, el estándar HTML5 es solo un borrador y se ha estado implementado en los distintos navegadores probados como se muestra en Figura 32 no se ha querido añadir más elementos de este nuevo estándar al ser probable que fueran modificados o eliminados [5].

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
3 versions back	6.0: 19%	6.0: 84%	13.0: 89%	3.2: 23%	11.0: 56%	3.2: 27%		10.0: 27%	
2 versions back	7.0: 19%	7.0: 84%	14.0: 89%	4.0: 43%	11.1: 58%	4.0-4.1: 40%		11.0: 53%	2.1: 26%
Previous version	8.0: 23%	8.0: 89%	15.0: 89%	5.0: 66%	11.5: 67%	4.2-4.3: 43%		11.1: 63%	2.2: 35%
Current	9.0: 52%	9.0: 89%	16.0: 89%	5.1: 78%	11.6: 71%	5.0: 70%	5.0-6.0: 14%	11.5: 63%	2.3: 41% 3.0: 61%
Near future	9.0: 52%	10.0: 89%	17.0: 89%	5.1: 78%	12.0: 74%				4.0: 66%
Farther future	10.0: 79%	11.0: 89%	18.0: 89%	6.0: 78%	12.1: 74%				

Figura 32. Soporte de los navegadores a HTML5 durante Diciembre 2011

Un caso específico referente a los cambios en la implementación de los elementos de HTML5 con el funcionamiento de la etiqueta *input* al establecer el valor *range* al atributo *type*. Se dio el caso de que al cambiar versión de un navegador (Chrome), el actualizado automático del valor que se había implementado no funcionaba correctamente. Debido al caso anteriormente comentado y al de otros elementos no implementados, se decidió añadir los elementos HTML5 esenciales.

Todos los elementos de HTML han sido añadidos mediante scripting, utilizando la interfaz DOM. Este método es comentado en la siguiente subsección. De modo general se realiza mediante la siguiente línea de código.

```
this.div = document.createElement("div");
```

El funcionamiento de esta función también se comentará más adelante. Usando la función *createElement* se han creado algunos de los elementos HTML utilizados en el proyecto, mientras, otros se han creado usando las etiquetas tradicionales. A continuación se muestra los elementos HTML utilizados en el proyecto:

- **!DOCTYPE**. Esta declaración se encuentra al principio del documento HTML5, no es una etiqueta, es una instrucción para el navegador sobre que versión de HTML se está utilizando.
- **html**. Esta etiqueta le indica al navegador que el documento es del tipo HTML, siendo, al mismo tiempo el elemento raíz de un documento HTML.
- **head**. El elemento head es el contenedor de todos los elementos incluidos en la cabecera del documento. Aquí se pueden incluir scripts, instrucciones al navegador de donde encontrar la hoja de estilos, ofrecer información referente al documento, etc. En este elemento se añade, por ejemplo, la librería creada o la hoja de estilos.
- **script**. Esta etiqueta se suele utilizar para definir scripts del lado del cliente, como Javascript. Puede contener instrucciones o referencias a ficheros externos. En las herramientas interactivas se utiliza para cargar la librería creada y, por otra parte, comandos para crear la herramienta interactiva utilizando dicha librería.
- **link**. La etiqueta define la relación entre el documento y un recurso externo. Se suele utilizar para enlazar las hojas de estilos. Este elemento se ha utilizado al crear las herramientas interactivas para el funcionamiento de MathML (véase sección 4.4).
- **body**. Este elemento define el cuerpo del documento. El cuerpo contiene todo el contenido de un documento HTML, siendo, en este caso, las gráficas, controles o elementos interactivos que forman las herramientas interactivas.
- **div**. Esta etiqueta normalmente se utiliza para definir una división o sección en un documento HTML y es, a menudo, utilizado para agrupar bloques de elementos y/o darles un estilo común. En este caso, se ha utilizado para crear cada tipo de elemento que forman la herramienta interactiva, así como subelementos que forman los distintos elementos si fuera necesario.
- **input**. Este elemento es utilizado por el usuario para introducir información. El tipo de campo de entrada puede variar dependiendo del atributo *type*. Un elemento *input* puede ser un campo de texto, un botón, un deslizador, etc. En la librería se han utilizado dos tipos; un campo de texto para introducir valores y deslizadores para variar el valor de las variables características de los sistemas implementados por las herramientas interactivas.
- **label**. Esta etiqueta define un rótulo para una etiqueta *input*. Este rótulo no muestra nada especial al usuario. Sin embargo, este permite una mejora de la usabilidad para los usuarios de ratones porque, si el usuario realiza clic sobre el texto, este cambia el control. Este elemento solo se ha utilizado con los controles de las variables donde se han conectado con los elementos *input* que manejan dichas variables.

Aunque parezcan pocos elementos, esto es debido a que muchos otros elementos que forman las herramientas interactivas que se pueden crear con esta librería dependen de otros

---

estándares, como SVG y MathML, desarrollados por W3C y que se comentan en secciones posteriores.

En la Figura 33 se muestra una captura de una herramienta interactiva donde se ha destacado en distintos colores los distintos elementos HTML que se han utilizado. No se han marcado todos ellos para no saturar la imagen con colores y, de esta forma, que se puedan ver y entender mejor. Se han destacado las tres etiquetas HTML utilizadas con distintos colores:

- En color verde se muestran las etiquetas *div* que forman los distintos elementos de la herramienta; controles de las variables interactivas, ecuación del sistema, la cabecera de las gráficas, el marco que contiene las gráficas, las gráficas, las escalas de las gráficas y los botones que modifican las escalas.
- En color azul se muestran las etiquetas *input* que, como se comentó se utilizan de dos formas distintas según el atributo *type*. En este caso son o tipo *range* o sin ningún tipo, esto es, como un simple campo de texto.
- Y en color naranja se muestran las etiquetas *label*.

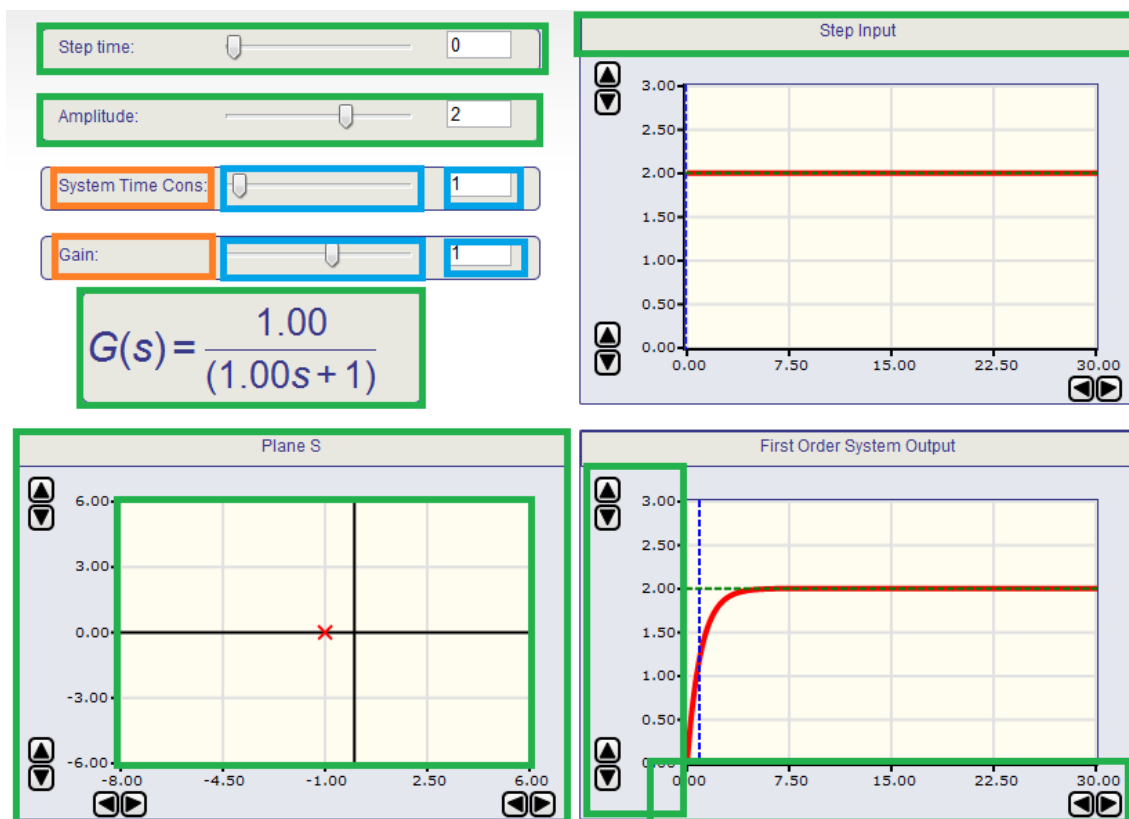


Figura 33. Herramienta interactiva resaltando elementos HTML

## 4.2 Scripting y DOM HTML

Aunque el lenguaje más comúnmente usado de scripting (ECMAScript, más ampliamente conocido como JavaScript) es desarrollado por Ecma, W3C ha definido la gran mayoría del resto de las APIs creadas disponibles en los navegadores [36][38].

### 4.2.1 Que es el Scripting

Un script es un código de programa que no necesita pre-procesamiento (por ejemplo compilación) antes de ser ejecutado. En el contexto de los navegadores Web, el scripting usualmente se refiere a un código de programa escrito en JavaScript que es ejecutado en un navegador cuando una página es descargada, o en respuesta a un evento lanzado por el usuario.

El Scripting puede hacer las páginas Web más dinámicas. Por ejemplo, sin recargar una nueva versión de una página esta podría permitir modificaciones en el contenido de la página, o permitir al contenido para ser añadido o enviado desde esa página. El primer ejemplo ha sido denominado DHTML (Dynamic HTML, HTML Dinámico), y el último AJAX (Asynchronous Javascript and Xml, Javascript Asíncrono y Xml).

Bajo estos, los scripts, cada vez más, permiten a los desarrolladores crear un puente entre el navegador y la plataforma en la que están ejecutándose, haciendo posible, por ejemplo, crear páginas Web que incorporan información desde los entornos de usuario, tales como localización actual, detalles de la libreta de direcciones, etc.

Esta interactividad adicional hace a las páginas Web comportarse como las aplicaciones de software tradicional. Estas páginas Web son a menudo llamadas aplicaciones Web y puede estar disponibles directamente tanto en el navegador en forma de página Web, como pueden ser empaquetadas y distribuidas como un Widget.

### 4.2.2 DOM HTML

La interfaz más básica de scripting es la desarrollada por W3C llamada DOM (Document Object Model, Modelo de Objetos del Documento), la cual permite a programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de los documentos. Las especificaciones DOM forman el núcleo del DHTML.

Aunque existen otras interfaces para modificar el contenido, el uso de DOM, por parte del usuario y de disparadores de eventos, permiten construir interfaces de usuario mucho más ricas. Un número de interfaces más avanzadas han sido estandarizadas, por ejemplo XMLHttpRequest (componente principal de la interfaz AJAX) hace posible la carga adicional de contenido desde la Web sin tener que descargar un nuevo documento completo.

### 4.2.3 Uso de Scripting y DOM HTML en el proyecto

Para el desarrollo de la librería se ha utilizado el lenguaje ECMAScript (JavaScript), creándose una serie de clases que se explicarán en la sección 5.2. Además, se ha utilizado la interfaz de DOM HTML y, como veremos en la sección 4.2.3.5, también se utilizó la interfaz AJAX,

probando así su funcionamiento. En esta sección solo se destacarán aquellos elementos de estas interfaces cuyo conocimiento resulta interesante conocer para uso en la creación de la librería.

#### 4.2.3.1 Nodos y el árbol de DOM

Antes de continuar explicando cómo se ha utilizado el scripting en la implementación del proyecto se debería saber cómo estructura DOM un documento HTML y sus elementos DHTML. Como es bien sabido, HTML, es un lenguaje de etiquetado. Por lo tanto, DOM, para poder moverse por las etiquetas, representa estas etiquetas en una estructura típica de árbol con nodos. Esta estructura de árbol tiene siempre como nodo raíz el documento representado por la propiedad *document*.

#### 4.2.3.2 JavaScript un Lenguaje prototipado

Un aspecto a tener en cuenta de JavaScript es que es un lenguaje prototipado. Éste es el único mecanismo que permite la herencia en JavaScript entre dos objetos. Resulta interesante conocer esto ya que en la sección 5.1 se verá la forma de crear herencia en la implementación de la librería. Veamos algunos aspectos importantes del prototipado en JavaScript.

- **La cadena de prototipos:** Un objeto puede opcionalmente tener un prototipo, que apunta a otro objeto por medio de la propiedad interna *Prototype*. El valor de este propiedad puede ser obtenida por medio del método *Object.getPrototypeOf()*. El objeto prototipo puede, a su vez, tener un prototipo. La secuencia de objetos que son prototipos de cada uno es llamada cadena de prototipos (ver Figura 34).
- **Búsqueda de propiedades:** Si JavaScript no puede encontrar una propiedad en un objeto, mantiene la búsqueda en el prototipo, siguiendo por el prototipo del prototipo, etc. Y siempre la propiedad *this* apunta al principio de la cadena de prototipos.

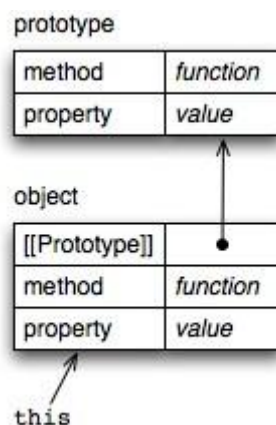


Figura 34. Esquema prototipado JavaScript

### 4.2.3.3 Creación de elementos y asignación de atributos

Para la creación de los distintos elementos que forman las herramientas interactivas en la librería se han utilizado principalmente dos funciones:

- **createElement.** Esta función crea un elemento del tipo especificado. Nótese que la instancia devuelta implementa la interfaz *Element*, así que los atributos pueden ser directamente especificados en el objeto devuelto. Si hay atributos conocidos con valores por defecto, los nodos *Attr* que los representan son automáticamente creados y añadidos al elemento. La instrucción tiene la siguiente forma:

```
Element createElement(in DOMString tagName)
```

- Parámetro:
  - *tagName* del tipo *DOMString*: Es el nombre del tipo de elemento a instanciar. Si se especifica que el documento es XML, entonces se deben mayúsculas y minúsculas. Si el documento es HTML, el parámetro podría ser permitido en cualquier caso, pero debe ser mapeado por su forma canónica en mayúsculas por la implementación DOM.
- Valor devuelto. *Element*, un nuevo objeto *Element* con el atributo *nodeName* con el valor *tagName*, y *localName*, *prefix*, y *namespaceURI* con valor *null*.

En la librería, se ha utilizado esta función para crear los distintos elementos HTML (*div*, *input*, *label*). Veamos como ejemplo la creación de un control de una variable interactiva. Para ello se crea, primero, un elemento *div* y se guarda en una variable local del objeto que lo crea.

```
this.div = document.createElement("div");
```

Dentro de este *div*, que será el contenedor, se añadirán un *label* y los elementos *input* que modifican el valor de la variable. Estos se crean de la siguiente forma:

```
control = document.createElement("label");
control = document.createElement("input");
```

- **createElementNS.** Crea un elemento a partir de nombre cualificado y de un URI de espacio de nombres dados. Los elementos HTML implementados por DOM son los únicos que no necesitan ser implementados por este método. La instrucción tiene la siguiente forma:

```
Element createElementNS(in DOMString namespaceURI, DOMString
qualifiedName)
```

- Parámetros:
  - *namespaceURI* del tipo *DOMString*. El URI del espacio de nombres elemento a crear.
  - *qualifiedName* del tipo *DOMString*. El nombre cualificado del tipo de elemento a instanciar.
- Valor devuelto. *Element*, un nuevo objeto *Element* con el atributo *nodeName* con el valor *qualifiedName*, el atributo *namespaceURI* con el valor de parámetro *namespaceURI*, el atributo *prefix* con el valor de *prefix* del *qualifiedName* o null si no existe, el atributo *localName* con el valor del *localName* obtenido del *qualifiedName* y el valor del atributo *tagName* es *qualifiedName*.

En la librería, esta función se ha utilizado para crear los distintos elementos SVG y MathML, los cuales no son implementados por DOM y, por tanto, es necesario usar esta función para obtenerlos. Por ejemplo para crear las gráficas se realizan las siguientes acciones:

```
this.svg = 'http://www.w3.org/2000/svg';  
this.svgframe = document.createElementNS(this.svg, 'svg');  
this.ceroline = document.createElementNS(this.svg, 'line');
```

Se puede observar que le URI del espacio de nombres es “http://www.w3.org/2000/svg”, el cual es guardado en una variable que luego se pasa como parámetro al método *createElementNS*. También se puede ver que se crea un elemento *svg*, que es el raíz para poder representar cualquier elemento SVG, y luego se crea un elemento de tipo *line*.

Una vez creados los distintos elementos, a estos se le asignan sus atributos para asignarles una posición en el documento, un estilo u otros atributos que los caracterizan. Para ello, se ha utilizado el método *setAttribute*.

- **setAttribute**. Esta función añade un nuevo atributo. Si el atributo ya existe actualmente en el elemento su valor es cambiado por el valor pasado por parámetro. Este valor no será convertido al tipo del atributo. El método tiene la forma que se muestra a continuación

```
void setAttribute(in DOMString name, DOMString value)
```

- Parámetros:
  - *name* de tipo *DOMString*. Es el nombre del atributo a crear o modificar.

- value de tipo *DOMString*. Es el valor que será asignado.

Siguiendo el ejemplo anterior con SVG veamos que atributos se añaden o modifican usando el método *setAttribute*:

```

this.svgframe = document.createElementNS(this.svg, 'svg');
this.svgframe.setAttribute('width', (this.width)+'px');
this.svgframe.setAttribute('height', (this.height)+'px');
this.ceroline = document.createElementNS(this.svg, 'line');
this.ceroline.setAttribute("style", "fill:none;stroke:black;stroke-width:2");
this.ceroline.setAttribute("transform", "translate(0,"+(this.height)+") scale(1,-1)");

```

Se puede ver cómo, una vez creado el elemento *svg*, se le añade un ancho (*width*) y un alto (*height*), mientras que, a la línea creada, se le añade un estilo haciendo que la línea sea de color negra y con un ancho dos, seguido de una transformación de la línea que consiste en trasladarla y realizar un escalado. El estilo se le da utilizando el estándar CSS que se comenta en la sección 4.3.3, mientras que la transformación forma parte del estándar de SVG.

Una vez creados los elementos y asignados sus atributos, los elementos deben ser añadidos al documento. Se puede añadir un elemento al documento directamente al nodo raíz *document* o a uno de sus hijos. Para añadirlos se utiliza el método *appendChild*:

- **appendChild**. Este método añade el nodo *newChild* al final de la lista de hijos del nodo sobre el que se ejecuta el método. Si *newChild* ya está en el árbol, este es eliminado primero.

```
Node appendChild(in Node newChild)
```

- Parámetro:
  - *newChild* del tipo *Node*. El nodo que será añadido.
- Valor devuelto. *Node*, el elemento *Node* que ha sido añadido.

Siguiendo con el ejemplo con elementos SVG, terminaríamos la secuencia de crear el elemento, añadirle atributos y añadirlo al documento tal y como se muestra a continuación:

```

this.svg = 'http://www.w3.org/2000/svg';
this.svgframe = document.createElementNS(this.svg, 'svg');
this.svgframe.setAttribute('width', (this.width)+'px');
this.svgframe.setAttribute('height', (this.height)+'px');

this.div.appendChild(this.svgframe);

this.ceroline = document.createElementNS(this.svg, 'line');
this.ceroline.setAttribute("style", "fill:none;stroke:black;stroke-width:2");

```



```
this.ceroline.setAttribute("transform", "translate(0,"+(this.height)+") scale(1,-1)");  
  
this.svgframe.appendChild(this.ceroline);
```

Se puede ver que, una vez creado el elemento del tipo *svg*, este se añade a la variable *this.div* que es un elemento *div* y, justo después de crear y asignarle los atributos al elemento de tipo *line*, se añade como hijo al elemento de tipo *svg* creado anteriormente.

### 4.2.3.4 Manejo de eventos en DOM

En esta sección se explica cómo funcionan los eventos en DOM y que funciones existe para manipularlos.

#### 4.2.3.4.1 Captura de eventos

La captura del evento es un proceso por el cual un manejador de eventos registrado en un ancestro del objetivo del evento puede interceptar el evento de un tipo dado antes de que sea recibido por el objetivo del evento. La captura opera desde el principio del árbol, generalmente *Document*, y baja, hasta el objetivo, realizando el camino simétrico opuesto al realizado por la propagación de eventos. Este camino se define antes de que se envíe el evento. Si se modifica el árbol de elementos durante el procesado, el flujo del evento que se procesara se basa en el estado inicial del árbol.

Si se desea parar la propagación de un evento se puede realizar la llamada al método *stopPropagation()* de la interfaz *Event*. Este no permitirá que el evento siga siendo enviado, aunque los escuchadores de eventos adicionales registrados en el mismo nivel continuarán recibiendo el evento.

La captura de eventos solo permite interceptar aquellos eventos que son objetivo de elementos descendientes. No permitiendo interceptar eventos con objetivos de los superiores, hermanos o descendientes de hermanos. El evento capturado no se especifica para un simple objetivo, se especifica para un tipo específico de evento. Una vez especificado, el capturador de eventos intercepta todos los eventos del tipo especificado que tenga como objetivo cualquier descendiente del capturador.

#### 4.2.3.4.2 Propagación del evento

Es el proceso por el cual un evento se propaga hacia arriba a través de sus ancestros después de ser manejado por el objetivo del evento.

#### 4.2.3.4.3 Cancelación del evento

Algunos eventos son especificados como cancelables. Para estos eventos, la implementación DOM generalmente tiene una acción por defecto asociada con el evento. Un ejemplo de esto es un hiperenlace en un navegador web. Cuando el usuario hace click sobre el hiperenlace la acción por defecto es generalmente activar este hiperenlace. Antes del procesamiento de estos eventos, la implementación debe comprobar los demás manejadores de eventos registrados para recibir el evento y despachar el evento a estos manejadores. Estos manejadores entonces tienen la opción de cancelar la acción implementada por defecto o

permitir que la acción por defecto proceda. En el caso del hiperenlace en un navegador, cancelar la acción dará como resultado la no activación del hiperenlace.

La cancelación se puede realizar mediante la llamada del método *preventDefault* del prototipo *Event*. Si uno o más manejadores de eventos llama al método *preventDefault* durante cualquier fase del flujo del evento la acción por defecto será cancelada.

#### 4.2.3.4.4 Registro de escuchadores de eventos

La interfaz *EventTarget* es implementada por todos los Nodos en una implementación la cual soporta el Modelo de Eventos DOM. Por tanto, esta interfaz se puede obtener usando los métodos específicos de llamada sobre una instancia de la interfaz *Node*. La interfaz permite el registro y la eliminación de manejadores de eventos sobre un objetivo de eventos y el envío de eventos hacia estos objetivos de eventos. De la interfaz nos interesan los siguientes eventos que son los utilizados en la implementación de la librería.

#### Función **addEventListener**

Esta función permite el registro de manejadores de eventos sobre un objetivo de eventos. Si un manejador de eventos es añadido a un objetivo de eventos mientras se está procesando un evento, está no será lanzado por la acción actual pero podría ser lanzado durante la etapa posterior del flujo de evento, como la fase de propagación. Si múltiples e idénticos manejadores de eventos son registrados sobre el mismo objetivo de eventos con los mismos parámetros la duplicación de instancias son descartadas. Estos no producen que el manejador de eventos sea llamado dos veces ya que estos fueron descartados y no necesitan ser eliminados con el método *removeEventListener*.

```
void      addEventListener(in DOMString type,  
                          in EventListener listener,  
                          in boolean useCapture);
```

Tiene los siguientes parámetros:

- *Type* del tipo *DOMString*. El tipo de evento que está siendo registrado.
- *Listener* del tipo *EventListener*. Es una interfaz implementada por el usuario la cual contiene los métodos que serán llamados cuando los eventos ocurran.
- *useCapture* del tipo boolean. Si es true, *useCapture* indica que el usuario desea iniciar la captura. Después de iniciarse la captura, todos los eventos del tipo especificado serán enviados al manejador de eventos registrado antes de ser enviado a cualquier otro objetivo de eventos por debajo de ellos en el árbol. Aquellos eventos que son propagados hacia arriba a través del árbol no lanzarán los manejadores de eventos designados para usar captura.

No devuelve ningún valor.

### Función `removeEventListener`

Este método permite la eliminación de los manejadores de eventos del objetivo del evento. Si un manejador de eventos es eliminado de un objetivo de eventos mientras se está procesando un evento, este no será lanzado por las acciones actuales. Un manejador de eventos nunca podrá ser llamado después de ser eliminado. La llamada a `removeEventListener` con argumentos que no identifican ningún escuchador de eventos actualmente registrado sobre el objetivo de eventos no tendrá efecto.

```
void      removeEventListener(in DOMString type,  
                              in EventListener listener,  
                              in boolean useCapture);
```

Tiene los siguientes parámetros:

- *type* del tipo *DOMString*. Especifica el tipo de evento del *EventListener* que será eliminado.
- *Listener* del tipo *EventListener*. Indica el escuchado de eventos que será eliminado.
- *useCaputre* del tipo boolean. Especifica si el manejador de eventos que será eliminado fue registrado como un manejador capturador o no. Si un manejador fue registrado dos veces, una con captura y otra sin, cada una debe ser eliminada separadamente. La eliminación de un manejador capturador no afecta a una versión no capturadora del mismo manejador, y viceversa

No devuelve un valor.

#### 4.2.3.5 *Uso de Ajax*

El uso de Ajax en la librería se ha limitado a la obtención de datos desde un servidor remoto. Este proceso se ha realizado, de forma experimental, para comprobar si era factible el funcionamiento de esta herramienta de obtención de datos y la interactividad que proporcionaba. Los resultados de esta prueba se comentará con más detalle en el apartado 5.4.7. Esta sección se limitará a comentar el funcionamiento del scripting utilizado que hace uso de Ajax.

Se ha utilizado para obtener los datos el objeto *XMLHttpRequest* cuando no se trataba de un navegador distinto a Internet Explorer, puesto que éste utiliza el objeto *ActiveXObject*., cambiando también los comandos y parámetros utilizados. Por otra parte, aunque el acrónimo AJAX de Asynchronous JavaScript and XML, no se tiene que utilizar XML, y podría utilizarse JSON. Aún así, para no tener que utilizar una nueva librería adicional se optó por utilizar únicamente XML. El método siguiente es parte de la codificación de la librería, donde se muestra el uso de Ajax.

```
this.xmlHttpPost = function(strURL,query) {
```

```

var xmlhttpReq = false;
var self = this;
// Mozilla/Safari
if (window.XMLHttpRequest) {
self.xmlhttpReq = new XMLHttpRequest();
}
// IE
else if (window.ActiveXObject) {
self.xmlhttpReq = new ActiveXObject("Microsoft.XMLHTTP");
}
self.xmlhttpReq.open('POST', strURL, true);
self.xmlhttpReq.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
self.xmlhttpReq.onreadystatechange = function() {
if (self.xmlhttpReq.readyState == 4) {
txtDoc=self.xmlhttpReq.responseText;

if (window.DOMParser){
parser=new DOMParser();

xmlDoc=parser.parseFromString(txtDoc,"text/xml");
}
// IE
else{
xmlDoc=new
ActiveXObject("Microsoft.XMLDOM");
xmlDoc.async="false";
xmlDoc.loadXML(txtDoc);
}
xml = xmlDoc;
}
}
self.xmlhttpReq.send(query);
}

```

### 4.3 Gráficos

La Web es algo más que texto e información, es también un medio por el cual expresar la creatividad artística, la visualización de datos, y la optimización de la presentación de información para diferentes audiencias con diferentes necesidades y expectativas. El uso de gráficos en sitios Web mejora la experiencia de los usuarios, y W3C tiene algunas tecnologías diferentes y complementarias que trabajan conjuntamente con HTML y el scripting. Estas herramientas proporcionan a los creadores de páginas y Aplicaciones Web las herramientas que estos necesitan para ofrecer la mejor representación posible de su contenido.

Los gráficos Web son representaciones visuales usadas sobre sitios Webs para mejorar o permitir la representación de una idea o sentimiento, con el fin de llegar a los usuarios del sitio Web. Los gráficos pueden entretener, educar, o impactar emocionalmente al usuario, y son cruciales para dar claridad de ilustración, y facilidad de uso a las interfaces.

Como ejemplos de gráficos se incluyen mapas, fotografías, diseños y patrones, árboles de familia, diagramas, planos arquitectónicos o de ingeniería, gráficos de barras o circulares, tipografías, esquemas, líneas de arte, diagramas de flujo, y muchas otras formas de imágenes.

Los diseñadores gráficos tienen muchas herramientas y tecnologías a su disposición, y W3C proporciona muchos de los formatos de fondo que pueden ser usados para la creación de contenidos en la plataforma web abierta.

Los diferentes casos de uso de gráficos exigen soluciones diferentes, por lo tanto hay diferentes tecnologías disponibles. Las fotografías son mejor representadas con PNG, mientras que las líneas de arte interactiva, visualización de datos, e incluso los interfaces de usuarios necesitan del poder de SVG o la API Canvas. CSS existe para mejorar otros formatos como HTML o SVG. WebCGM satisface las necesidades para la ilustración y documentación técnica en muchas industrias.

En este caso nos interesa las capacidades que nos ofrece SVG y la API de Canvas para realizar la interfaz de usuario y CSS para mejorar el contenido de la aplicación tanto de los elementos HTML como de los elementos creados con SVG, la API de Canvas o MathML.

### 4.3.1 SVG

SVG (Scalable Vector Graphics, Gráficos Vectoriales Escalables) es parecido a HTML pero para gráficos. Es un lenguaje de etiquetado para describir todos los aspectos de una imagen o aplicación Web, desde la geometría de las formas, el estilo de los textos y las formas, las animaciones, hasta las presentaciones multimedia incluyendo vídeo y audio. Es completamente interactivo e incluye una secuencia de comandos DOM como la animación declarativa (vía la especificación SMIL). Es compatible con un amplio rango de características visuales tales como gradientes, opacidad, filtros, recorte y enmascarado.

El uso de SVG permite gráficos totalmente escalables, suavizados y reutilizables, a partir de simples gráficos para mejorar páginas HTML, hasta gráficos y visualización de datos totalmente interactivos, juegos e imágenes estáticas de alta calidad independientes. SVG es nativamente compatible con la mayoría de los navegadores modernos, y está ampliamente disponible en dispositivos móviles. La mayoría de las herramientas de dibujo de gráficos vectoriales importan o exportan SVG. Además puede ser generado desde el lado de cliente o del lado del servidor con lenguajes de scripting [42].

### 4.3.2 La API de Canvas

La API de Canvas es una tecnología de scripting del lado del cliente que permite la creación o alteración de imágenes de trama (bitmaps) ricas. Usa métodos de programación basado en vectores para crear las formas, gradientes y otros efectos gráficos y, al no tener DOM, puede ejecutarse muy rápidamente. Scripters dedicados pueden desarrollar juegos o incluso aplicaciones llenas de características usando la API Canvas, solo o integrados con HTML o SVG. Es compatible nativamente con la mayoría de los navegadores modernos e incluso en algunos dispositivos móviles.

### 4.3.3 CSS

CSS (Cascading Style Sheets) es un lenguaje que describe la presentación de páginas Webs, incluyendo colores, diseño, y fuentes. Permite adaptar la presentación a distintos tipos de dispositivos, tales como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y puede ser usado con cualquier lenguaje de etiquetado basado en XML como SVG. La separación entre HTML y CSS hizo más fácil el mantenimiento de sitios, compartiendo hojas de estilo entre páginas, y páginas a medida de distintos entornos. Esto se conoce como la separación entre la estructura y la presentación [37].

### 4.3.4 SVG o API Canvas

Se realizaron unos pequeños ejemplos iniciales de la herramienta usando SVG y Canvas de forma separada para comparar sus características a la hora de representar gráficamente datos, tales como su facilidad a la hora de implementar la librería para la creación de herramientas interactivas o el resultado final de las herramientas creadas (ver Figura 35).

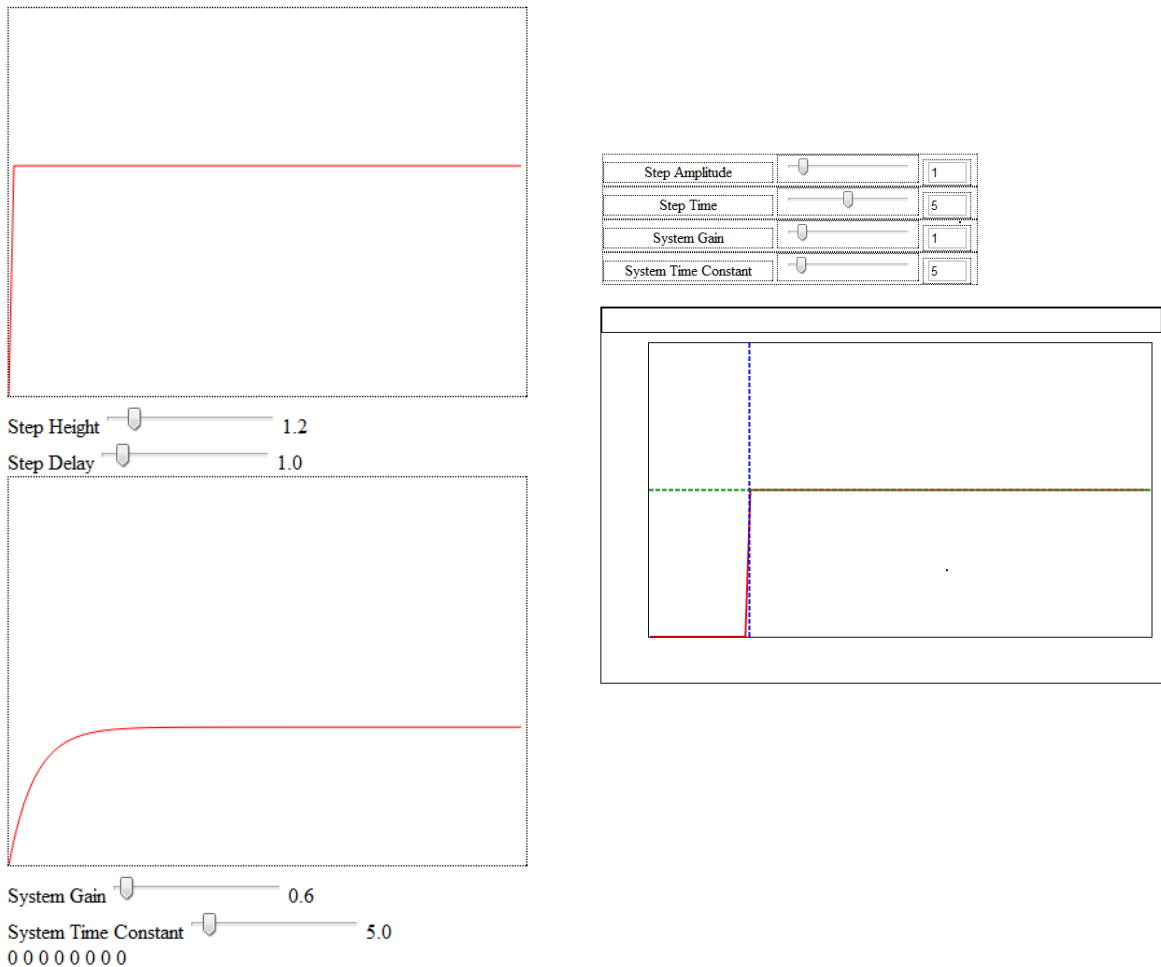


Figura 35. Comparativa API Canvas (izquierda) y SVG (derecha)

Se puede observar que el resultado es similar quitando las diferencias de colores y contenido, etc., obteniéndose un aspecto muy similar. Finalmente se escogió usar SVG debido a las siguientes razones:

- Trabajo con eventos. Con SVG se pueden asignar manejadores de eventos a cada elemento de los elementos gráficos representados de una forma fácil mientras que con Canvas este trabajo se debería hacer de forma manual y programarlo. Esto hace que trabajar con eventos en SVG sea más sencillo.
- SVG vectorial. SVG es vectorial como se ha comentado anteriormente haciendo que los elementos que se muestran sean escalables y tenga un mejor aspecto una vez representados. Esto en su contra hace que SVG sea más costoso de representar para el dispositivo, pero actualmente hasta los dispositivos más pequeños tienen suficiente potencia para manejarlos sin dificultad.
- SVG ya implementado. Como ya se comentó antes, SVG es compatible con la mayoría de navegadores modernos y dispositivos móviles, mientras que Canvas es una tecnología reciente y no está ampliamente soportada.

Aunque se escogió la utilización de SVG, esta elección no quiere decir que no se pueda utilizar Canvas. Para ciertos aspectos de las herramientas interactivas se podría utilizar Canvas, según el dispositivo, para que fuera más eficiente a la hora de ejecutarse en dispositivos con poca potencia, pero para no complicar la implementación de la librería se descartó esta posible mejora.

### 4.3.5 Uso de SVG en el proyecto

SVG se ha utilizado en el proyecto principalmente para la representación gráfica de los datos generados dinámicamente y los elementos interactivos que se representan en las gráficas, pero también se ha utilizado para la creación de los iconos de los botones. Para la creación de los distintos elementos se ha utilizado el método *createElementNS* de DOM, utilizando como nombre cualificado <http://www.w3.org/2000/svg>, tal y como se puede ver en las siguientes líneas de código:

```
this.svg = "http://www.w3.org/2000/svg";  
this.svgframe = document.createElementNS(this.svg, "svg");
```

Cada elemento (líneas, círculos, poli-líneas, etc.) que se representa en SVG debe de añadirse sobre un elemento del tipo *svg*. Por lo tanto después del código anteriormente descrito en esta sección, se podría añadir una polilínea utilizando el método *appendChild* de DOM de la siguiente forma:

```
this.line = document.createElementNS(this.svg, "polyline");  
this.svgframe.appendChild(this.line);
```

Para la realización de la librería se han utilizado los siguientes elementos de SVG, creándose tal y como se acaba de mostrar:

- **svg.** Es el elemento raíz de cualquier documento svg. Los demás elemento svg deben tener como ancestro un elemento svg. Es por esa razón que se ha utilizado en la librería para crear cualquier elemento que requiriera SVG.
- **line.** Este elemento define un segmento de una línea que empieza en un punto y termina en otro. Teniendo como atributos las coordenadas de los punto inicial y final. Este elemento se ha utilizado para representar valores interactivos modificables en uno de los ejes del plano, creando líneas horizontales y verticales, y también para la representación de los ejes del plano donde se representarán los datos.
- **circle.** Este elemento define un círculo basado en un punto central y un radio teniendo como atributos a estos. Aunque este elemento define círculos se ha utilizado para definir puntos rellenando el interior con el mismo color del borde del círculo. Este punto se ha utilizado para representar valores interactivos que pueden ser modificables en los dos ejes del plano o que requerían que su representación fuera circular.
- **polyline.** Este elemento define un conjunto de segmentos de líneas rectas conectados, normalmente se utiliza para definir formas abiertas. Tiene como atributo el conjunto de puntos que se conectarán mediante líneas. Con este elemento se han representado conjuntos de datos en gráficas de líneas. Depende, por tanto, del tamaño donde se muestra la polilínea y de la cantidad de puntos el que se vean picos al representar los datos.
- **path.** Representa el contorno de una forma que puede ser rellenada, trazada o usada como línea guía o una combinación de cualquiera de las tres. Tiene como atributos los datos del camino de la forma. Este elemento se ha utilizado para la creación del grid de las gráficas, ya que nos permite el actualizarlas utilizando un único elemento SVG en vez de crear un elemento line por cada línea del grid.
- **rect.** Este elemento define un rectángulo alineado con los ejes del plano del elemento al cual pertenece. Tiene como atributos un punto inicial, el ancho y el alto del rectángulo y el radio de redondeo de las esquinas. Este elemento se ha utilizado para crear un marco para las leyendas de los datos de las gráficas.
- **text.** Este elemento define un texto. Tiene como atributos el punto inicial donde se representara entre otros atributos. Se utiliza para crear texto dentro de un elemento SVG como es el caso del texto de la leyenda o de las escalas.

Para añadir los atributos a los distintos elementos SVG se ha utilizado el método *setAttribute*.

#### 4.3.6 Uso de CSS en el proyecto

Para el uso de CSS dentro de la librería creada en este proyecto se ha utilizado el método *setAttribute* sobre el atributo *style*. El atributo *style* está destinado a modificar el estilo del elemento mediante instrucciones CSS. Esta forma de modificación del CSS mediante el atributo



*style* permite cambiar el estilo de forma dinámica de una forma sencilla. CSS también tiene otros métodos propios de modificación dinámica del estilo de elementos pero en un principio se ha optado por la utilización del método *setAttribute* de DOM ya que es más adecuada a la forma en la que se ha realizado el trabajo.

Un ejemplo de código CSS es el siguiente:

```
math {line-height:1.3em; text-indent:0;}
```

En este ejemplo se define la altura de una línea y el sangrado de un texto. Este mismo código se podría usar mediante la función método *setAttribute* de la siguiente forma:

```
setAttribute("style"," line-height:1.3em;text-indent:0;");
```

Obteniéndose en ambos casos el mismo resultado pero con instrucciones distintas.

Un posible cambio futuro es la asignación de clases CSS para los elementos definidos en la librería creada para crear herramientas interactivas. Con esto se conseguiría que se pueda utilizar un fichero CSS para cambiar el estilo de los elementos de una herramienta interactiva de forma sencilla. Dando de esta forma, la posibilidad de cambiar los estilos utilizando solo CSS y no teniendo que utilizar instrucciones en JavaScript. Este aspecto no se ha realizado ya que es algo opcional y no fundamental para el funcionamiento de la librería o de las herramientas creadas con ésta, siendo solo una mejora en la accesibilidad a la creación de herramientas interactivas.

Aunque no se haya utilizado CSS de forma tradicional en general dentro de la librería, si que se ha utilizado para poder usar una aproximación de MathML mediante la solución provisional ofrecida por W3C para navegadores que no cumplen el estándar MathML. Esta solución provisional consiste en una hoja de estilos CSS que emula aproximadamente el comportamiento que debería tener MathML.

## 4.4 MahtML

MathML (Mathematics Markup Language, Lenguaje de Marcado Matemático) es una tecnología Web para el almacenado y transporte de formulas matemáticas. Este puede ser usado para incluir formulas en documentos (por ejemplo, en ensayos o artículos) y para el intercambio de datos entre software matemático [41].

MathML es un lenguaje para expresar formulas matemáticas. Está basado en XML y es compatible con un amplio rango de productos software.

W3C publicó la primera versión de MathML en 1998. Fue actualizado y ampliado en 2003. La tercera versión, la cual incluye, entre otras cosas, el soporte de lenguajes escritos de derecha a izquierda, matemáticas elementales, e integración de los diccionarios de contenido de OpenMath, llegó en 2010.

Conceptualmente, MathML tiene dos partes: presentación y contenido. Estas forman un lenguaje único y es fácil de mezclar, pero se concentran en distintos objetivos: la parte de la presentación expresa la visión en dos dimensiones de una fórmula y es más adecuada para publicaciones, tales como artículos y libros; la parte del contenido se concentra en la semántica y es más adecuada como un formato de intercambio entre aplicaciones para ingeniería, estadística, etc.

El etiquetado de una fórmula simple podría ser como éste:

```
<msqrt> <mi> x </mi> <mo> + </mo> <mfrac>
<mi> a </mi> <mi> b </mi> </mfrac> </msqrt>
```

#### 4.4.1 Uso de MathML en el proyecto

Como ya se comentó anteriormente la mayoría de navegadores web no implementan este estándar, por lo que para su utilización se ha utilizado una plantilla de estilos CSS que emula el comportamiento de MathML. Esta plantilla de estilos se ha obtenido en la especificación del estándar y se muestra en el anexo Plantilla CSS MathML. Pero esta plantilla solo hace que se visualice de forma correcta en los navegadores, para crear los elementos del estándar MathML se ha hecho de la misma forma que si los navegadores implementaran este estándar. Por tanto, una vez que un navegador implemente este estándar solo habría que dejar de usar esta hoja de estilos. En la Figura 36 se puede ver la diferencia entre usar o no la plantilla CSS en el momento de la implementación de la librería.

Figura 36. Comparativa de uso de plantilla CSS para MathML.

Para la creación en la librería de los distintos elementos que forman parte de las herramientas interactivas se ha utilizado el método *createElementNS* de DOM utilizando como nombre cualificado <http://www.w3.org/1998/Math/MathML> como se utilizó de forma similar para los elementos del estándar SVG.

```
this.math =
document.createElementNS("http://www.w3.org/1998/Math/MathML", "math");
```

Además estos elementos se deben anidar lo cual se ha realizado con el método *appendChild* de DOM.

```
this.math.appendChild(this.mrow1);
```

Una vez creados los elementos, se añade el contenido de estos modificando su atributo *textContent*. Este atributo está definido para cualquier elemento de un documento HTML.

```
this.mi1.textContent = "x";
```

Los elementos MathML utilizados en la librería son los siguientes:

- *math*. Este elemento identifica el nivel más alto o la raíz de una instancia MathML.
- *mrow*. Este elemento se utiliza para diferenciar grupos de cualquier número de subexpresiones horizontalmente.
- *mfrac*. Este elemento forma una fracción de dos subexpresiones.
- *mo*. Este elemento se utiliza para añadir operadores, corchetes o separadores.
- *mi*. Este elemento se utiliza para añadir identificadores.
- *mn*. Este elemento se utiliza para añadir números.
- *msup*. Este elemento añade un superíndice a una base.

Utilizando estos elementos en distintos órdenes se han creado las fórmulas que muestran las distintas herramientas interactivas.

Como se comentó anteriormente, generalmente los principales navegadores no implementan el estándar MathML por lo que se utilizó una hoja de estilos CSS, pero aun así esto puede dar problemas. Por ejemplo, se producía un mal funcionamiento al ver las herramientas con el navegador Safari o el navegador por defecto de los sistemas operativos IOS que llevan los dispositivos portátiles de Apple. Concretamente se producía al crear un paréntesis "(" como un elemento *mo*, llegando a la solución de crearlo como un elemento *mn*. Esta solución resuelve el funcionamiento errático haciendo que se cambiase un mal funcionamiento de la herramienta interactiva, por un funcionamiento normal de la herramienta con una peor representación gráfica del paréntesis, pero que aun así legible. Este problema se dio durante el periodo de implementación de la librería en distintas versiones, por lo que el estado de este problema puede cambiar.

También se produjeron otros problemas como que los superíndices se muestran justo encima de la base en lugar de a su lado derecho. Un principio de solución para el navegador Chrome fue las siguientes líneas de código ya que este problema no se produce por ejemplo en el navegador Firefox.

```
if (navigator.userAgent.indexOf("Chrome")!=-1){  
  document.styleSheets[0].addRule("msup",  
  "width:"+this.mi3.textContent.length*0.8+"em");  
}
```

Pero viendo que la solución era pobre y que el trabajo para corregir estos detalles era costoso se decidió no hacer hincapié en estos problemas y esperar a que la futura implementación del estándar en los navegadores los arregle.





## HERRAMIENTAS DESARROLLADAS

Este capítulo describe la metodología empleada en el desarrollo y como ha sido estructurada de la librería y las herramientas interactivas. También se divide y se describen las distintas partes del proyecto. Esta división consiste en: la librería, las herramientas interactivas, uso de AJAX y Python y la interfaz gráfica para crear herramientas interactivas.

### 5.1 Análisis y diseño

Para el desarrollo de la librería primero se creó el núcleo de la librería a la misma vez que se implementaba la herramienta interactiva que representa el análisis de un proceso de un sistema de primer orden, siendo está la más sencilla de las herramientas creadas. El núcleo consiste en las clases básicas para poder representar gráficamente e interactuar con un proceso. Después se fueron desarrollando las demás herramientas y se fue mejorando el núcleo si era necesario añadiendo más funcionalidades o cambiando las ya existentes. Aparte se añadió otras funcionalidades, cuando se vio necesario, para mejorar las herramientas interactivas o el mismo funcionamiento de la librería, por ejemplo para su funcionamiento en distintos dispositivos. También se trato de crear una interfaz de usuario gráfica para la creación de herramientas interactivas con el uso de la librería implementada, esto se comenta con más detalle en la sección 0.

Cada uno de las mejoras citadas anteriormente se ha realizado utilizando las fases clásicas para el desarrollo del software:

1. Análisis: En general para la librería se ha realizado un análisis típico de la ingeniería del software, mientras que para las herramientas interactivas que representan procesos (sistema de primer orden, etc.), además se identifica el conjunto de variables y los elementos interactivos.
2. Diseño: se diseñan las clases necesarias y en caso de existir, se elaboran diseños gráficos de los elementos gráficos.
3. Implementación: se implementa en JavaScript las clases diseñadas de la librería. Para las herramientas interactivas se crea la página web y el *script*.
4. Prueba/entrega: cada nueva función para representar una parte del sistema, o acción interactiva, es probada individualmente mediante una batería de pruebas. Una vez implementada la mejora o nueva funcionalidad del sistema, es entregada al director del proyecto para su revisión.

El conjunto de etapas anteriores se realizan de forma iterativa hasta obtener un prototipo totalmente funcional del sistema con la aprobación del director del proyecto, teniendo éste constante contacto con los resultados obtenidos en cada incremento. El primer incremento o prototipo es un sistema esencial sin componentes interactivas.

Este proyecto se ha intentado analizar, diseñar e implementar intentando hacer una programación orientada a objetos. Pero la orientación a objetos en JavaScript es distinta, como se comento en la sección x JavaScript es un lenguaje Prototipado, es decir la herencia no se realiza entre clases sino que los objetos utilizan las propiedad *prototype* (ver sección 4.2.3.2). Cabe destacar que la forma de herencia mediante prototipos no permite crear herencia múltiple de forma sencilla, es por esto que en vez de usar los prototipos se ha creado la herencia de forma manual. Esta forma consiste, para cada clase que herede, en crear una variable con la clase de la cual hereda y ejecutar esta variable en el *constructor* de la clase. De esta forma la clase heredera obtiene las funciones y propiedades de la que da la herencia. A continuación se muestra el código de la clase *ICScaleButton* que hereda de la clase *IButton* [23],[25].

```
this.extend = IButton;  
this.extend(parent,x,y);
```

Existen otras formas de herencia en JavaScript con sus pros y sus contras pero debido a la poca experiencia en JavaScript y para facilitar la implementación de la librería se opto por esta forma.

Debido a que ahora se usando JavaScript orientado a objetos, ahora en vez usar *funciones* utilizaremos *métodos*.

## 5.2 Descripción de clases

En esta sección se realiza una breve descripción de las clases que han sido diseñadas e implementadas en la librería.

### 5.2.1 Data

Clase abstracta que aglutina todos los métodos y propiedades de las clases que se encargan de crear u obtener llamando a un servicio web los datos de los distintos sistemas y procesos, donde estos datos luego son representados con otras clases. De esta clase hereda las clases *System* y *Step*.

### 5.2.2 FirstOrderSystem

Esta clase hereda de la clase *System* y se encarga de generar u obtener los datos de un sistema de primer orden según los valores que tengan los parámetros que definen el modelo del sistema (constante de tiempo, ganancia y tiempo de retardo), otras variables (tiempo de muestreo y número de muestras) y los datos de la señal de entrada al sistema. Además esta clase se encarga de dar la posibilidad de generar controles y elementos interactivos que se representan en las clases *Graph* y *PlaneS* que permiten modificar el valor de las variables características.

### 5.2.3 Frame

Clase que crea un marco en donde añadir las gráficas, escalas, botones y otros elementos. Básicamente actúa como un contenedor pero esta clase también se encarga de añadir automáticamente los elementos predeterminados al crear los distintos tipos de gráficas. En la Figura 37 se muestra su aspecto si no se le añade ningún elemento.

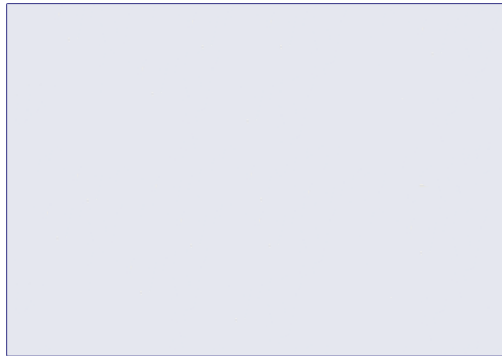


Figura 37. Marco que genera la clase Frame

### 5.2.4 Graph

La clase *Graph* se encarga de crear un plano donde representar los datos añadiendo los ejes y una rejilla (Grid) antes de ocuparse de representar los datos obtenidos a partir de una clase del tipo *Frame* y elementos interactivos procedentes de la clase *Data*. En la Figura 38 se muestra el aspecto de la gráfica obtenida al usar esta clase con los datos de la clase *FirstOrderSystem*.

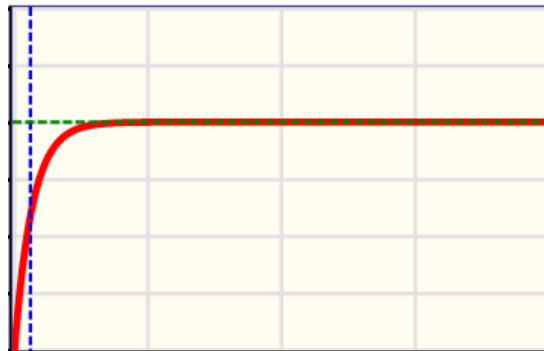


Figura 38. Ejemplo de gráfica que genera la clase Graph

Como se ha dicho anteriormente los datos se obtienen de la clase *Frame* en vez de una de las clases que heredan de la clase *Data* (encargadas de generar los datos). Esto se hace de esta forma para que se puedan elegir que datos se desean representar en dentro de un *Frame* de entre todos los que crea la clase *Data*. Esta selección de datos se realiza de forma automática según el valor que tenga la propiedad *type* de la clase *Frame*.



### 5.2.5 GraphLegend

Clase que crea una leyenda para los datos de una gráfica generada por la clase *Graph*, creándose por defecto en la parte superior izquierda de la gráfica. Para ello accede a los datos que utilizó la clase *Graph* y obtiene el nombre y el color de las clases *IData* que son las que contiene dicha información de los datos. En la figura x se muestra el aspecto que tiene la leyenda generada esta clase.

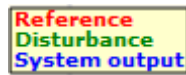


Figura 39. Leyenda que genera la clase GraphLegend

### 5.2.6 Grid

Clase que genera la rejilla (Grid), mostrada en color gris en la Figura 40, que utilizan las clases *Graph* y *PlaneS*. No realiza ninguna función excepto la de mostrar la rejilla, esta clase se creó pensando en el futuro de la librería y en ampliar su funcionalidad. La rejilla tiene tantas divisiones horizontales y verticales como divisiones tiene el escalado horizontal y vertical respectivamente de la gráfica donde se crea.

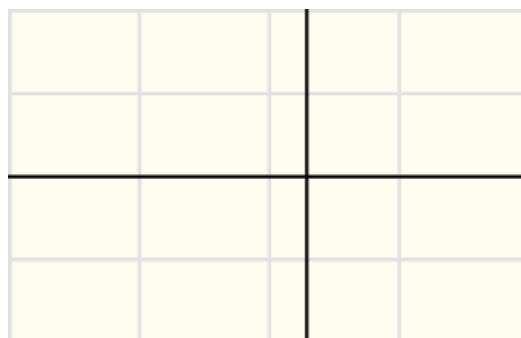


Figura 40. Rejilla que genera la clase Grid

### 5.2.7 HScale

Esta clase crea una escala horizontal y dos botones para ampliar o reducir el límite superior de la escala. El número de la escala es modificable y por defecto el mínimo es cero y el máximo se ajusta al número de muestras que tienen los datos del *Frame* donde se posiciona. El aspecto que presenta la escala generada por la clase se muestra en la Figura 41.

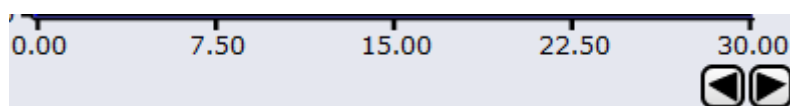


Figura 41. Escala que genera la clase HScale

### 5.2.8 IButton

Clase abstracta que agrupa todos los métodos y propiedades de las clases que se encargan de crear botones sin importar las acciones que realizan. De esta clase hereda la clase *ICScaleButton* y otras clases que se han desarrollado para el futuro de la aplicación como *AutoScaleButton*, *PropertiesButton* y *IMoveButton* que se describen en secciones posteriores.

### 5.2.9 IControl

La clase *IControl* crea un control como el de la Figura 42 partir de un objeto de la clase *IValue* donde el control modifica la propiedad *value* de la clase *IValue*. El control creado se compone de tres partes:



Figura 42. Control que genera la clase IControl

- Una etiqueta con el nombre de la variable que se almacena en la propiedad *name* de la clase *IValue*.
- Un botón deslizador (slider) que modifica el valor de la variable sobre un máximo *IValue.max* y un mínimo *IValue.min*.
- Un campo de entrada donde se puede escribir el valor de la variable.

### 5.2.10 ICScaleButton

Esta clase crea un botones que aumenta o reducen los límites inferior y superior de las escalas creadas por las clases *HScale* y *VScale*. Los botones consisten en un recuadro con fondo gris y bordes negros que tienen una flecha en el interior con una dirección y sentido según su función como se muestran en la Figura 43.



Figura 43. Botones que genera la clase ICScaleButton

### 5.2.11 IData

La clase *IData* consiste en una serie de propiedades que almacenan información sobre los datos generados por las clases de tipo *Data* que es útil para otras clases que trabajan con estos datos y una serie de métodos útiles para trabajar con esta clase.

### 5.2.12 IDot

Esta es una clase que tiene como función representar variables (normalmente la propiedad *value* de la clase *IValue*) en forma de puntos interactivos sobre los distintos elementos gráficos

(clase *Graph* y *PlaneS*). Algunos ejemplos de la forma de la representación de variables creada se muestran en la Figura 44 con una equis que se usa para representar los polos de los sistemas o un círculo para representar los ceros en un plano S.



Figura 44. Elementos gráficos que genera la clase *IDot*

### 5.2.13 *IElement*

Se trata de una clase abstracta que congrega todos los métodos y propiedades de las clases que se encargan de representar variables en forma elementos interactivos sobre gráficos SVG. De esta clase hereda las clases *IDot* e *ILine*.

### 5.2.14 *IHLine*

Esta clase hereda de la clase *ILine* y tiene como función representar variables (normalmente la propiedad *value* de la clase *IValue*) en forma de líneas horizontales interactivas sobre los distintos elementos gráficos (clase *Graph* y *PlaneS*). En la Figura 45 se muestra una de estas líneas.

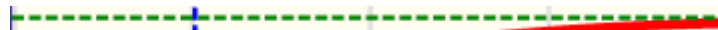


Figura 45. Línea que genera la clase *IHLine*

### 5.2.15 *ILine*

Se trata de una clase abstracta que hereda de la clase *IElement* que congrega todos los métodos y propiedades de las clases que se encargan de crear elementos interactivos en forma de línea sobre gráficos SVG. De esta clase hereda las clases *IHLine* e *ILine*.

### 5.2.16 *Info*

Esta clase se encarga de crear marco que contiene una fórmula matemática utilizando el estándar MathML. Normalmente describe formulas relacionadas con los datos de sistemas de control generados por las clases del tipo *Data*, cambiando la formula dinámicamente al igual que las variables características que forman parte de los sistemas. En la Figura 46 se observa la formula que describe un sistema de primer orden creada a partir de esta clase.

$$G(s) = \frac{0.81}{(7.65s + 1)}$$

Figura 46. Fórmula generada por la clase *Info*

### 5.2.17 IValue

La clase *IValue* realiza el mantenimiento de una variable que es interactiva, normalmente tratándose de una variable perteneciente a una clase *Data*, que se utiliza para generar los datos y que se quiere que sea interactiva. Esta clase se encarga de generar si se desea un control de la variable usando la clase *IControl* y/o el elemento interactivo para modificar el valor de la variable. Además se encarga de llamar a la actualización de la clase *Data*, el control o el elemento interactivo cuando se modifica el valor de la variable.

### 5.2.18 IVLine

Esta es una clase que tiene como función representar variables (normalmente la propiedad *value* de la clase *IValue*) en forma de líneas verticales interactivas sobre los distintos elementos gráficos (clase *Graph* y *PlaneS*). En la Figura 47 se muestra una de estas líneas.



Figura 47. Línea que genera la clase *IVLine*

### 5.2.19 MenuBar

Esta clase genera una cabecera sobre el marco creado por la clase *Frame* y escribe un título en el centro de esta cabecera. Esta clase también está diseñada para que en un futuro se añadan botones con los que se puedan modificar las preferencias del contenido del marco, como colores, fuentes, grosores de líneas, etc. Y que se puedan añadir botones que aporten otras funcionalidades, como lupa, captura, exportación de datos, etc. En la Figura 48 se puede ver el aspecto que tiene la cabecera generada.



Figura 48. Barra generada por la clase *MenuBar*

### 5.2.20 PIDFOS

Esta clase hereda de la clase *FirstOrderSystem* y se encarga de generar u obtener los datos de un sistema de primer orden y de un controlador PID según los valores tengan los parámetros que definen el modelo del sistema y el controlador (constante de tiempo, ganancia, tiempo de retardo, constante proporcional, constante integral y constante derivativa), otras variables (tiempo de muestreo y número de muestras) y los datos de entrada (una entrada en escalón, la referencia y una perturbación en forma de escalón). Además esta clase se encarga de dar la

posibilidad de generar controles y elementos interactivos que se representan en las clases *Graph* y *PlaneS* que permiten modificar las variables características.

### 5.2.21 PlaneS

La clase *PlaneS* se encarga de crear un plano donde representar los datos añadiendo los ejes y una rejilla (*Grid*) antes de representar los elementos interactivos que se deseen de una clase *Data*. En la Figura 49 se muestra el aspecto de la gráfica obtenida al usar esta clase donde se añade el polo de un sistema de primer orden.

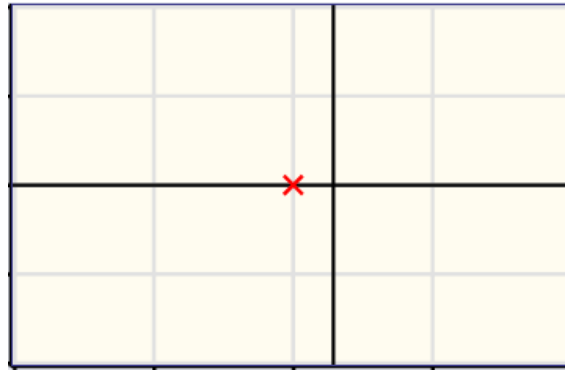


Figura 49. Plano s generado por la clase PlaneS

### 5.2.22 Point

Esta clase representa los puntos donde se almacenan los datos generados por una clase *Data*. Los datos generados por una clase *Data* consisten en una lista (un objeto *Array* de JavaScript) de puntos con coordenadas en los ejes x e y, donde el tamaño de esta lista es el del número de muestras. En un futuro en esta clase se podría añadir la función de representar también gráficamente cada punto y mostrar sus valores.

### 5.2.23 Property

En un principio esta clase se diseñó para que se modifiquen propiedades pertenecientes a las distintas clases que generan elementos visuales en las herramientas. Con esto se pretendía que propiedades como colores, tamaños o el que un elemento este activo o no se crearan con esta clase y esta clase se encarga de crear un control para poder modificar su valor. Pero finalmente esto se ha dejado para una mejora futura de la librería aunque gran parte de esta funcionalidad ya está implementada. Por otra parte se utiliza esta clase para almacenar propiedades que son comunes entre varios objetos de las clases *Data* como es el caso del número de muestras y el tiempo de muestreo para un objeto de la clase *Step* y uno de la clase *FirstOrderSystem*.

### 5.2.24 SecondOrderSystem

Esta clase hereda de la clase *System* y se encarga de generar u obtener los datos de un sistema de segundo orden según los valores que tengan los parámetros que definen el modelo del sistema (constante de tiempo, frecuencia natural no amortiguada, coeficiente de

amortiguamiento y tiempo de retardo), otras variables (tiempo de muestreo y número de muestras) y los datos de la señal de entrada al sistema. Además esta clase se encarga de dar la posibilidad de generar controles y elementos interactivos que se representan en las clases *Graph* y *PlaneS* que permiten modificar el valor de las variables características.

#### 5.2.25 *SecondOrderSystemCero*

Esta clase hereda de la clase *System* y se encarga de generar u obtener los datos de un sistema de segundo orden según los valores que tengan los parámetros que definen el modelo del sistema (constante de tiempo, frecuencia natural no amortiguada, coeficiente de amortiguamiento, tiempo de retardo y cero), otras variables (tiempo de muestreo y número de muestras) y los datos de la señal de entrada al sistema. Además esta clase se encarga de dar la posibilidad de generar controles y elementos interactivos que se representan en las clases *Graph* y *PlaneS* que permiten modificar el valor de las variables características. Esta clase muy parecida a la clase *SecondOrderSystem* quedando pendiente un estudio para la unificación de las dos clases en una o que la clase *SecondOrderSystemCero* herede de la clase *SecondOrderSystem*.

#### 5.2.26 *Step*

Esta clase hereda de la clase *Data* y se encarga de generar u obtener los datos de una señal de entrada en escalón según los valores que tengan los parámetros que definen el modelo de la señal ( tiempo de salto y amplitud) y otras variables (tiempo de muestreo y número de muestras). Además esta clase se encarga de dar la posibilidad de generar controles y elementos interactivos que se representan en las clases *Graph* que permiten modificar el valor de los parámetros que definen la señal.

#### 5.2.27 *System*

La clase *System* es abstracta y hereda de la clase *Data* consiste en una serie de propiedades que almacenan información sobre las clases que generan datos de modelos de sistemas. De esta clase hereda las clases *FirstSystemOrder*, *SecondSystemOrder*, *SecondSystemOrderCero*.

#### 5.2.28 *Scale*

Se trata de una clase abstracta que congrega todos los métodos y propiedades de las clases que se encargan crear escalas en las gráficas. De esta clase hereda las clases *VScale* y *HScale*.

#### 5.2.29 *VScale*

Esta clase crea una escala vertical y cuatro botones para ampliar o reducir los límites superior e inferior de la escala. El número de la escala es modificable y por defecto el mínimo y el máximo se ajustan al rango de los valores generados por la clase *Data* que pertenecen a la gráfica donde se encuentra la escala. El aspecto que presenta la escala generada por esta clase se muestra en la Figura 50.



Figura 50. Escala generada por la clase VScale

### 5.3 Diagrama de clases

Se ha realizado un diagrama de clases para entender mejor la conexión entre las distintas clases. Un resumen de dicho diagrama de clases se muestra en la Figura 51.

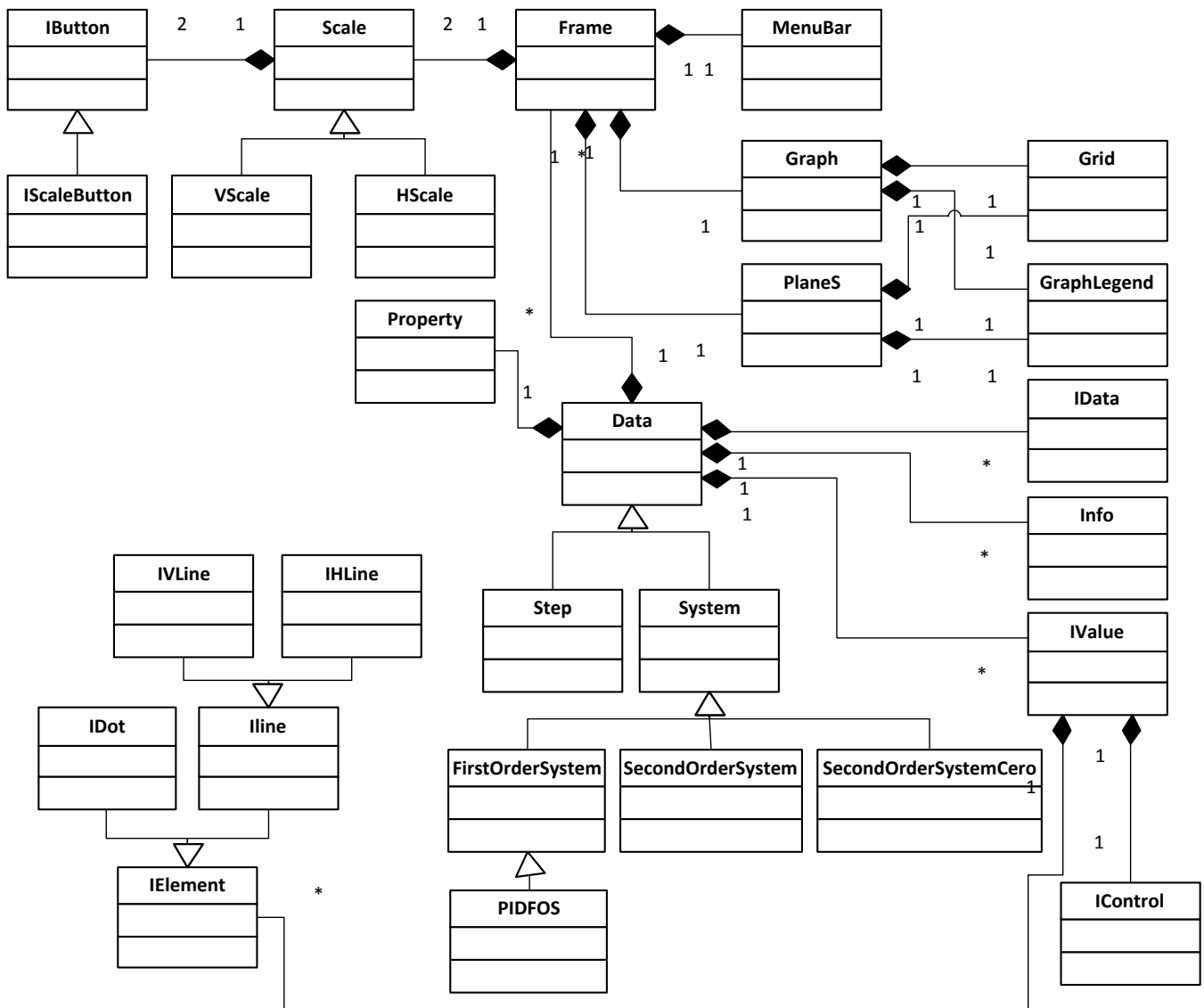


Figura 51. Diagrama de clases.

## 5.4 Herramientas desarrolladas

En esta sección se describen la metodología utilizada para desarrollar las distintas herramientas interactivas propuestas comprobándose el potencial y viéndose de este modo el funcionamiento de la librería implementada. También se documenta el funcionamiento de las distintas herramientas a modo de tutorial.

Para el desarrollo de una nueva herramienta interactiva se usa la librería creando una página HTML normal con algunas elementos específicos. La página debe cargar la librería desarrollada, *interactive\_control.js*, y la plantilla de estilos CSS para la representación matemática de las formulas con MathML, *mathml.css*. Entonces, se crea una función JavaScript que carga las variables principales, elementos gráficos y las capacidades de interacción.

Por defecto para la creación de las herramientas interactivas que se describen a continuación esta función JavaScript se ha cargado al inicio del documento HTML sobre la etiqueta `<body>` con el evento *onload*.

### 5.4.1 Sistema de primer orden

Esta herramienta interactiva describe el análisis de un sistema dinámico de primer orden visto en la sección 3.3.2 con una señal de entrada en escalón. El resultado se muestra en la Figura 52 diferenciándose las siguientes partes:

- Un área situada en la **parte superior izquierda** de la herramienta contiene una serie de controles interactivos, compuestos cada uno de ellos por una etiqueta un botón deslizador y una campo de entrada, que modifican los parámetros principales tanto del proceso de primer orden, la constante de tiempo y la ganancia, como de la señal de entrada en escalón situados, tiempo de escalón y amplitud.
- Una gráfica situada en la **parte superior derecha** que muestra la señal de entrada en escalón donde tanto la línea vertical azul que representa el tiempo de escalón y la línea horizontal verde que representa la amplitud del escalón pueden ser modificados interactivamente pulsando botón izquierdo y arrastrando el ratón. A está gráfica se le puede modificar el escalado pulsando los botones situados en la parte izquierda superior e inferior y la parte inferior derecha.
- Una gráfica situada en la **parte inferior izquierda** muestra el polo del sistema de primer orden representado con una equis roja sobre un plano  $s$ , donde el polo puede ser modificado interactivamente pulsando botón izquierdo y arrastrando el ratón. Además, a está gráfica se le puede modificar el escalado pulsando los botones situados en la parte izquierda superior e inferior y la parte inferior derecha.
- Situado en la **parte inferior derecha** se encuentra una gráfica que representa la salida del proceso de primer orden, donde la ganancia del proceso simbolizada con una línea horizontal de color verde y una línea de color azul vertical que representa la constante de tiempo del proceso pueden ser modificadas por interacción usando el ratón y



pulsando con el botón izquierdo y arrastrando vertical y horizontalmente respectivamente.

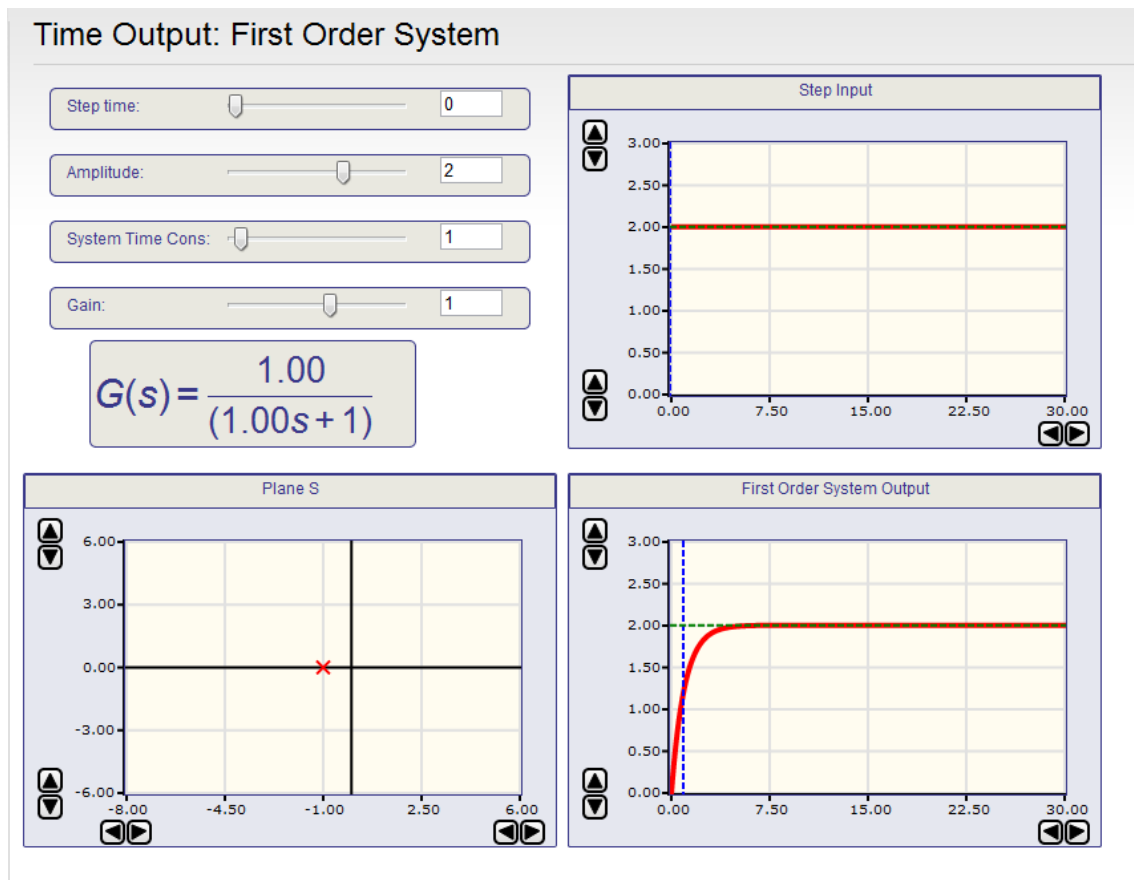


Figura 52. Herramienta interactiva de un sistema de primer orden

A continuación se muestra las instrucciones que se ha utilizado de la librería implementada para generar la herramienta interactiva anteriormente descrita y que se muestra en la Figura 52.

```

(1) <!doctype html>
(2) <html>
(3) <head>
(4) <title>First Order System</title>
(5)
(6) <link rel="stylesheet" type="text/css" href="mathml.css" />
(7) <link href="styles.css" rel="stylesheet" type="text/css">
(8) <script src="interactiv_control.js"
    type="text/javascript"></script>
(9)
(10) <script language="Javascript">
(11) var isiPad = navigator.userAgent.match(/iPad/i) != null;
(12)
(13) function init(){
(14)
(15)     document.onselectstart= function() {return false;}; // para
        mozilla poner en css "-moz-user-select: none;"

```

```

(16) //document.oncontextmenu = function(){return false;};
(17)
(18) length = new Property(this,30,"N° samples");
(19) sampling = new Property(this,0.1,"Sampling");
(20)
(21) step = new Step({text:"Step
Input",length:length,sampling:sampling});
(22) step.generateData();
(23) pframe = new
Frame({data:step,x:420,y:40,width:400,height:280,type:"graph"
,title:"Step Input"});
(24) step.addFrame(pframe);
(25) step.setDefault();
(26)
(27) system = new FirstOrderSystem({text:"First Order
System",input:step,length:length,sampling:sampling});
(28) system.generateData();
(29) systemframe = new
Frame({data:system,x:420,y:340,width:400,height:280,type:"gra
ph",title:"First Order System Output"});
(30) systemframe2 = new
Frame({data:system,x:10,y:340,width:400,height:280,type:"plan
es",title:"Plane S"});
(31) system.addFrame(systemframe);
(32) system.addFrame(systemframe2);
(33) system.setDefault();
(34)
(35) step.addInputof(system);
(36)
(37) length.attachElement(pframe);
(38) length.attachElement(systemframe);
(39) length.updateAttachedElements = function(){
(40)     for(var element in this.attachedElements) {
(41)         this.attachedElements[element].parent.update();
(42)
this.attachedElements[element].graph.calculateFactors();
(43)         this.attachedElements[element].update();
(44)     }
(45) }
(46)
(47) step.steptime.setValue(0);
(48) step.steptime.setMin(0);
(49) step.steptime.setMax(20);
(50) step.steptime.setStep(0.001);
(51) step.steptime.x = 30;
(52) step.steptime.y = 50;
(53) step.steptime.updateControl();
(54)
(55) step.amplitude.setValue(2);
(56) step.amplitude.setMin(-6);
(57) step.amplitude.setMax(6);
(58) step.amplitude.setStep(0.001);
(59) step.amplitude.x = 30;
(60) step.amplitude.y = 100;
(61) step.amplitude.updateControl();
(62)
(63) pframe.vscale.minadd = -1;
(64) pframe.vscale.nintervals.value = 6
(65) pframe.resetScales();
(66)
(67) system.timeconstant.setValue(1);

```

```
(68)    system.timeconstant.setMin(0.001);
(69)    system.timeconstant.setMax(30);
(70)    system.timeconstant.setStep(0.001);
(71)    system.timeconstant.x = 30;
(72)    system.timeconstant.y = 150;
(73)    system.timeconstant.updateControl();
(74)
(75)    system.gain.setValue(1);
(76)    system.gain.setMin(-6);
(77)    system.gain.setMax(6);
(78)    system.gain.setStep(0.01);
(79)    system.gain.x = 30;
(80)    system.gain.y = 200;
(81)    system.gain.updateControl();
(82)
(83)    system.delay.setEnabled(false);
(84)
(85)    system.info.x = 60;
(86)    system.info.y = 240;
(87)    system.info.width = 240;
(88)    system.info.update();
(89)
(90)    systemframe.vscale.maxadd = 1;
(91)    systemframe.vscale.minadd = 1;
(92)    systemframe.vscale.nintervals.value = 6;
(93)    systemframe.resetScales();
(94)
(95)    systemframe2.hscale.maxadd = 5;
(96)    systemframe2.hscale.minadd = -7;
(97)    systemframe2.vscale.maxadd = 4;
(98)    systemframe2.vscale.minadd = -4;
(99)    systemframe2.resetScales();
(100)   }
(101)   </script>
(102)   </head>
(103)   <body onload="init()">
(104)   <div id="right-sidebar">
(105)   <h2>Time Output: First Order System</h2>
(106)   </div>
(107)   </body>
(108)   </html>
```

El código empieza con la típica cabecera de una página HTML5 en las líneas (1) a (4). Seguidamente se los fichero externos necesarios para el funcionamiento de la herramienta, tales como las hojas de estilos CSS, para la utilización de MathML (6), el estilo de la propia página (7) y el fichero JavaScript que implementa la librería (7). Desde las líneas (10) a (135) se crea un script con una función re realiza las acciones necesarias para definir los elementos que serán incluidos en la herramienta interactiva utilizando las clases definidas en la librería implementada y que se ha cargado en la línea (7). Inicialmente se comprueba si el dispositivo utilizado ver la herramienta es un iPad (11), esta instrucción fue necesaria durante un periodo de la implementación debido a las diferencias que se encontraban entre el navegador Safari y el navegador que llevaba IOS (sistema operativo del iPad), no siendo necesaria actualmente y siendo añadida está en la librería en un futuro en caso de ser necesario.

A partir de la línea (12) se encuentra la función que crear la herramienta. Esta empieza con dos instrucciones opcionales (14-15) una de ellas comentada que sirven para que no se seleccione el texto al pulsar con el botón izquierdo y arrastrar del ratón (14) y para que no se muestre el

menú contextual al pulsar con el botón derecho del ratón (15). En las líneas (17-18) se crean dos propiedades, que definen el número de muestras y el tiempo de muestreo respectivamente, que se pasarán como parámetros tanto a la señal de entrada en escalón como al proceso del sistema en primer orden. Las líneas (20-24) se encargan de crear los datos de la señal de entrada en escalón y representarla en una gráfica. Primero se crea el objeto que crea los datos (20) y se generan los datos (21). Después se crea el marco con la gráfica donde se representan los datos (22), se añade el marco a los datos para que se actualicen correctamente (23) y se generan los controles por defecto (24). En las líneas (26-32) se crean todo lo referente al proceso del sistema de primer orden, creándose los datos (26), generando los datos (27), creando el marco con la gráfica que representa la salida del sistema (28), el polo del sistema en otro marco con un plano  $s$  (29) y por último añadiendo los marcos al objeto que genera los datos (30-31) y creando los controles por defecto (32). A continuación se añaden los datos del sistema de primer orden como la salida de datos de la señal en escalón (34) para que cuando se modifiquen los datos de la señal en escalón se actualice el sistema de primer orden.

Una vez creados las distintas gráficas y controles se procede a modificar las propiedades de estos poniéndole límites a sus valores y posicionándolos dentro de la herramienta interactiva. En primer lugar se modifica la propiedad del número de muestras (36-44) añadiéndole los objetos de tipo *Frame* creados anteriormente para que se actualicen cuando se modifique el valor del número de muestras (36-37) y se modifica el método `updateAttachedElements` para que realice unas operaciones necesarias para que tenga el comportamiento deseado al actualizarse el número de muestras (38-44). Estas líneas son temporales y en un futuro se modificara la librería para que realice estas operaciones automáticamente. Seguidamente se modifican los distintos controles de las variables características tanto de la señal de entrada en escalón como del proceso del sistema de primer orden, modificando sus valores iniciales del tiempo de escalón y la amplitud del escalón (46,54), la constante de tiempo y la ganancia (66,74) y modificando también sus valores máximos, mínimos y de paso del botón deslizador (47-49,55-57,67-69,75-77). A cada uno de los controles además se les modifica la posición donde se sitúan (50-51,58-59,70-71,78-79) y por último se llama a una función para que actualice los cambios realizados (52,60,72,80) e cada uno de estos controles respectivamente. Mientras en la línea (82) se desactiva y se oculta el control y los elementos interactivos de las gráficas pertenecientes al tiempo de retardo del proceso del sistema de primer orden. Y para terminar con la configuración de los controles entre las líneas (84) y (87) se modifica la posición donde se muestra y el ancho del cuadro que expone la fórmula del proceso del sistema de primer orden.

Por otra parte entre las líneas (89) y (98) se modifica los valores máximo y mínimo inicial y el número de intervalos del escalado de las gráficas que se sitúan en la parte inferior de la Figura 52 tanto en el lado izquierdo como en el derecho. Ya que por defecto el escalado se ajusta a los valores representados en las gráficas automáticamente.

Una vez finalizadas las instrucciones del script que genera la herramienta interactiva se procede a crear la pagina HTML que contendrá la herramienta (102-106). Esta página tener todo el contenido de una página HTML que se desee teniendo en cuenta que una vez que se carga el script de inicio aparecerán los elementos generados en este en las posiciones concordadas. Para cargar el script se procedido con la utilización del evento *onload* de la etiqueta principal de cualquier documento HTML `<body>` (102).

### 5.4.2 Sistema de primer orden con retardo

Esta herramienta interactiva describe el análisis de un sistema dinámico de primer orden con una señal de entrada en escalón al igual que la anterior pero esta vez añadiéndole retardo al sistema. El resultado se puede ver en la Figura 53 donde se pueden diferenciar cuatro partes pero que solo difieren de la herramienta interactiva descrita en la sección 5.4.1 en lo siguiente:

- Un área situada en la **parte superior izquierda** de la herramienta contiene una serie de controles interactivos, compuestos cada uno de ellos por una etiqueta un botón deslizador y una campo de entrada, que modifican los parámetros principales tanto del proceso de primer orden, la constante de tiempo, la ganancia y el tiempo de retardo, como de la señal de entrada en escalón situados, tiempo de escalón y amplitud.
- Una gráfica situada en la **parte superior derecha** es igual que la gráfica de la herramienta interactiva de un sistema de primer orden.
- Una gráfica situada en la **parte inferior izquierda** derecha también es idéntica que la gráfica de la herramienta interactiva de un sistema de primer orden.
- Situado en la **parte inferior derecha** se encuentra una gráfica que representa la salida del proceso de primer orden, donde la ganancia del proceso simbolizada con una línea horizontal de color verde, una línea de color azul vertical que representa la constante de tiempo del proceso y una línea de color azul vertical que representa el tiempo de retardo del proceso pueden ser modificadas por interacción usando el ratón y pulsando con el botón izquierdo y arrastrando vertical y horizontalmente respectivamente.

Las instrucciones necesarias para la creación de esta herramienta son similares que las utilizadas en la sección 5.4.1. Los cambios realizados respecto a las instrucciones son:

- Se elimina la línea (82), visualizando de esta forma el control y la línea interactiva en la gráfica del tiempo de retardo del sistema.
- Se añaden instrucciones de configuración para el tiempo de retardo del sistema similares a las líneas (74-80).

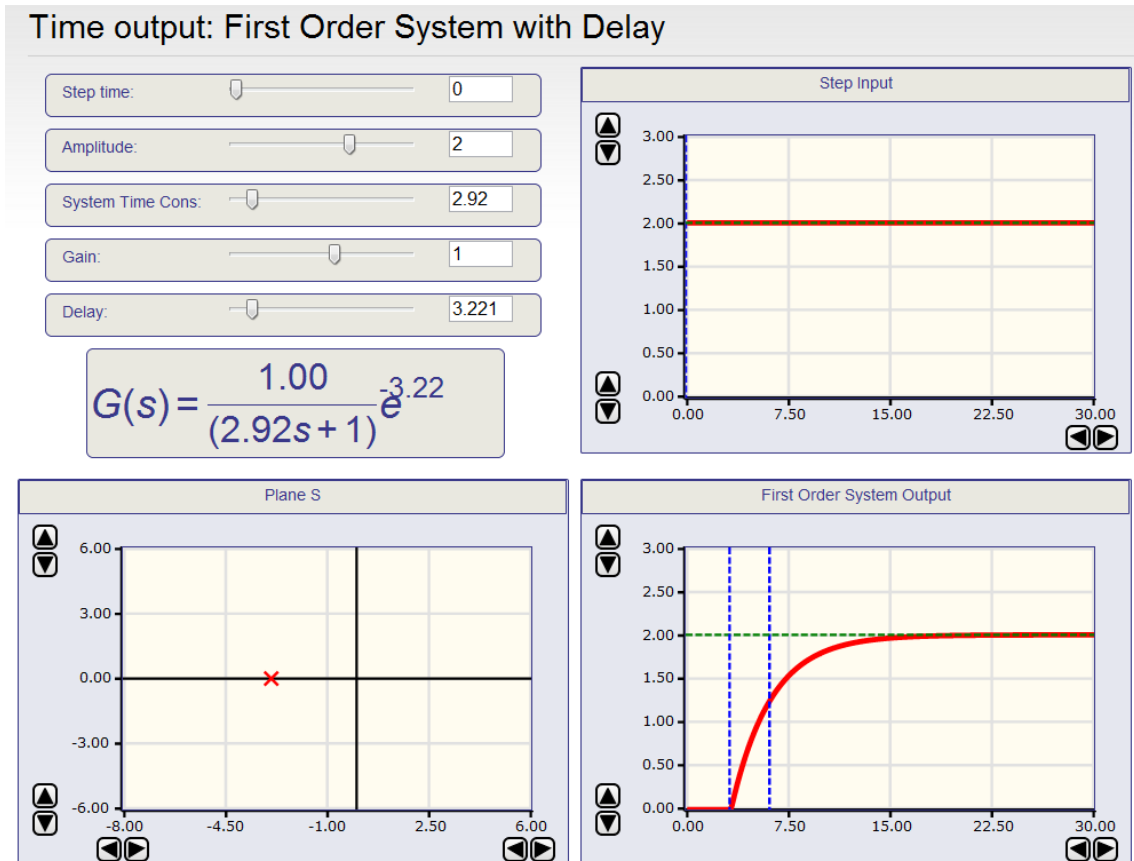


Figura 53. Herramienta interactiva de un sistema de primer orden con retardo

### 5.4.3 Sistema de segundo orden

Se ha creado una herramienta interactiva utilizando la librería creada en este proyecto que describe el análisis de un sistema dinámico de segundo orden con una señal de entrada en escalón. El resultado obtenido se muestra en la Figura 54 y Figura 55 según el valor de los polos, presentado un formato donde se pueden diferenciar cuatro partes:

- Un área situada en la **parte superior izquierda** de la herramienta contiene una serie de controles interactivos, compuestos cada uno de ellos por una etiqueta, un botón deslizador y un campo de entrada, que modifican unos parámetros principales y características del proceso de segundo orden según el valor de los polos como se define en la sección 3.3.5:
  - Si el valor de los polos del sistema es complejo, entonces, se muestran los controles de la ganancia estática, la frecuencia natural no amortiguada y el coeficiente de amortiguamiento. Además se añaden otras características como el tiempo de pico y la sobreoscilación máxima.

- Si el valor de los polos del sistema es real, entonces, los controles son distintos, mostrándose en este caso la constante de tiempo de cada uno de los polos.

Mientras que para la señal de entrada en escalón se muestran siempre los mismos controles, tiempo de escalón y amplitud.

Además de los controles se muestra una fórmula que representa el sistema de segundo orden. Esta fórmula también cambia su forma también según el valor de los polos del sistema.

- Una gráfica situada en la **parte superior derecha** que muestra la señal de entrada en escalón donde tanto la línea vertical azul que representa el tiempo de escalón y la línea horizontal verde que representa la amplitud del escalón pueden ser modificados interactivamente pulsando botón izquierdo y arrastrando el ratón. A esta gráfica se le puede modificar el escalado pulsando los botones situados en la parte izquierda superior e inferior y la parte inferior derecha.
- Una gráfica situada en la **parte inferior izquierda** muestra los dos polos del sistema de segundo orden representado con dos equis de color rojo sobre un plano  $s$ , donde los polos pueden ser modificados interactivamente pulsando botón izquierdo y arrastrando el ratón. Se puede observar que ambos polos son simétricos respecto al eje  $x$  mientras las raíces del sistema son complejas pero que cuando son reales el valor en el eje  $x$  de los polos puede ser distinto. Además, a esta gráfica se le puede modificar el escalado pulsando los botones situados en la parte izquierda superior e inferior y la parte inferior derecha.
- Situado en la **parte inferior derecha** se encuentra una gráfica que representa la salida del proceso de segundo orden, esta gráfica es distinta también al igual que la parte superior izquierda según el valor de los polos del sistema:
  - Si los polos son complejas, se muestra en rojo la señal de salida del sistema de segundo orden y los siguientes elementos interactivos: una línea horizontal de color verde que simboliza la ganancia del proceso, una línea azul vertical que representa el tiempo de pico y un punto de color morado que permite modificar la sobreoscilación máxima donde la ganancia del proceso simbolizada con una línea horizontal de color verde y una línea de color azul vertical.
  - Si los polos son reales, se puede observar en color rojo la señal de salida del sistema de segundo orden junto con los siguientes elementos interactivos: Una línea horizontal de color verde representando la ganancia del proceso junto con dos líneas azules verticales que están asociadas a los polos del sistema y un punto de color morado que permite modificar la sobreoscilación máxima que normalmente desaparece al situarse fuera de la gráfica.

Teniendo la posibilidad con los controles, tanto si los polos son complejos o reales, de modificar las variables asociadas usando el ratón, pulsando con el botón izquierdo y arrastrándolo.

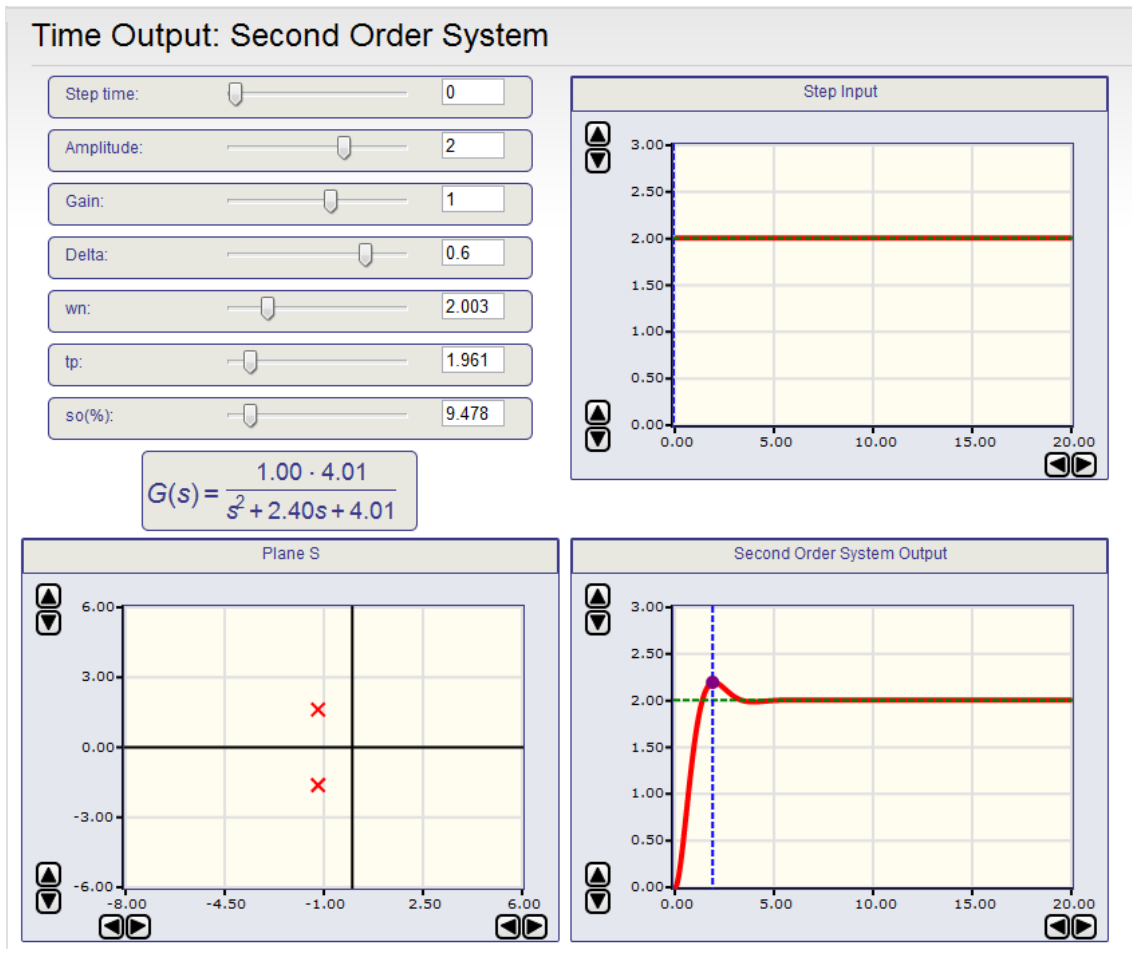


Figura 54. Herramienta interactiva de un sistema de segundo orden con dos polos complejos

Las instrucciones para generar esta herramienta son muy parecidas a las vistas en la sección 5.4.1, con cambios menores como cambiando el nombre de los títulos de las gráficas y las posiciones de los distintos elementos, sabiendo que estos son distintos a los de la sección 5.4.1. Se puede destacar un cambio importante, dándose que en este caso los datos generados son los de un sistema de segundo orden en vez de los de un sistema de primer orden por tanto se sustituye la línea (20) del código de la sección 5.4.1 por la línea que se muestra a continuación

```
(20) system = new SecondOrderSystem({text:"Second Order System
Output",input:step,length:length,sampling:sampling});
```



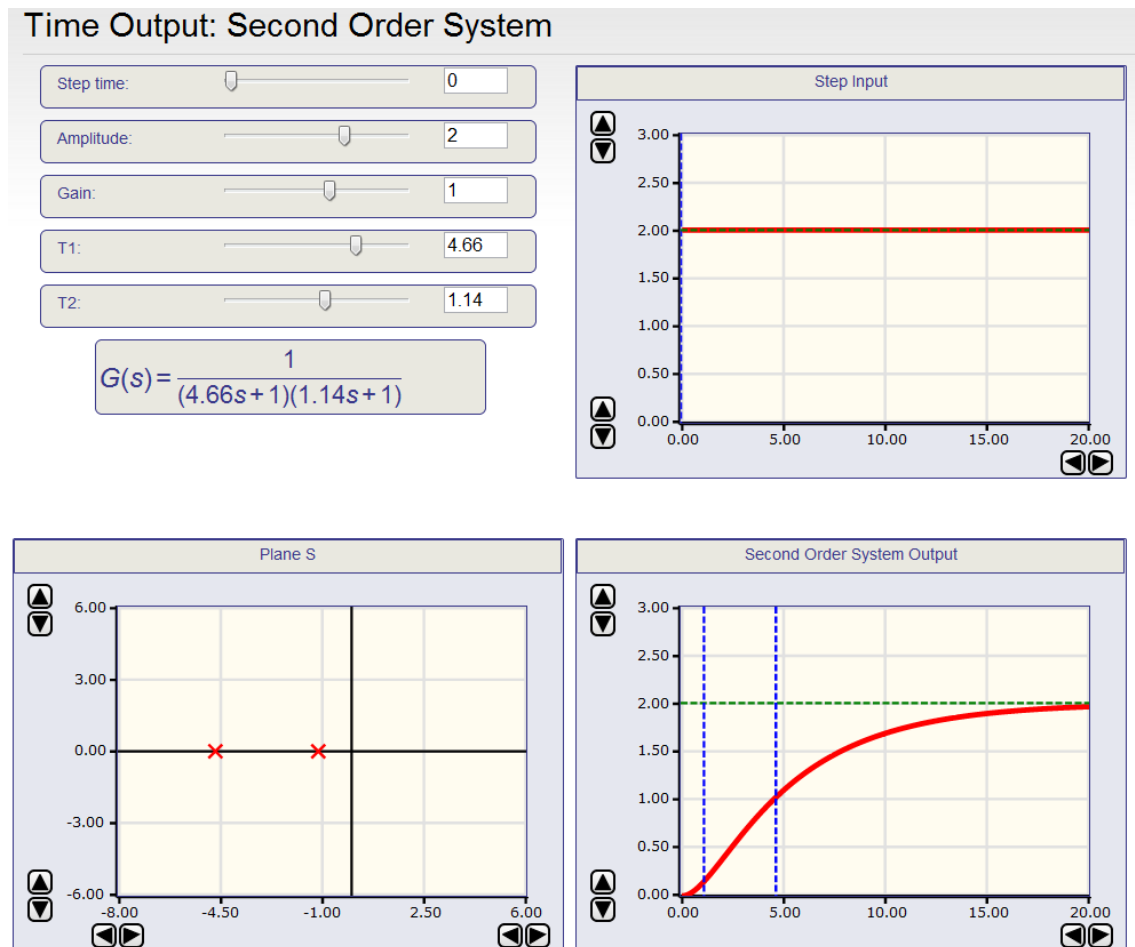


Figura 55. Herramienta interactiva de un sistema de segundo orden con dos polos reales

#### 5.4.4 Sistema segundo orden con retardo

Se ha creado una herramienta interactiva utilizando la librería creada en este proyecto que describe el análisis de un sistema dinámico de segundo orden con retardo con una señal de entrada en escalón. El resultado obtenido se muestra en la Figura 56 donde se puede observar la similitud con el sistema de segundo orden sin retardo, las principales diferencias que se pueden ver son:

- En la **parte superior izquierda** de la herramienta que contiene los controles de las variables interactivas, se añade a los ya descritos en la sección 5.4.3 un control para modificar el tiempo de retardo tanto las raíces del sistema son reales o complejas.
- La gráfica situada en la **parte superior derecha** es igual que la gráfica de la herramienta interactiva de un sistema de segundo orden.
- La gráfica situada en la **parte inferior izquierda** derecha también es idéntica que la gráfica de la herramienta interactiva de un sistema de segundo orden.

- Situado en la **parte inferior derecha** se encuentra una gráfica que representa la salida del proceso de segundo orden, en esta parte se añade una línea azul vertical interactiva que representa al tiempo de retardo, siendo esta modificable arrastrando el ratón cuando se pulsa el botón izquierdo del ratón sobre la línea. Esta línea aparecen tanto si las raíces el sistema son reales o complejas.

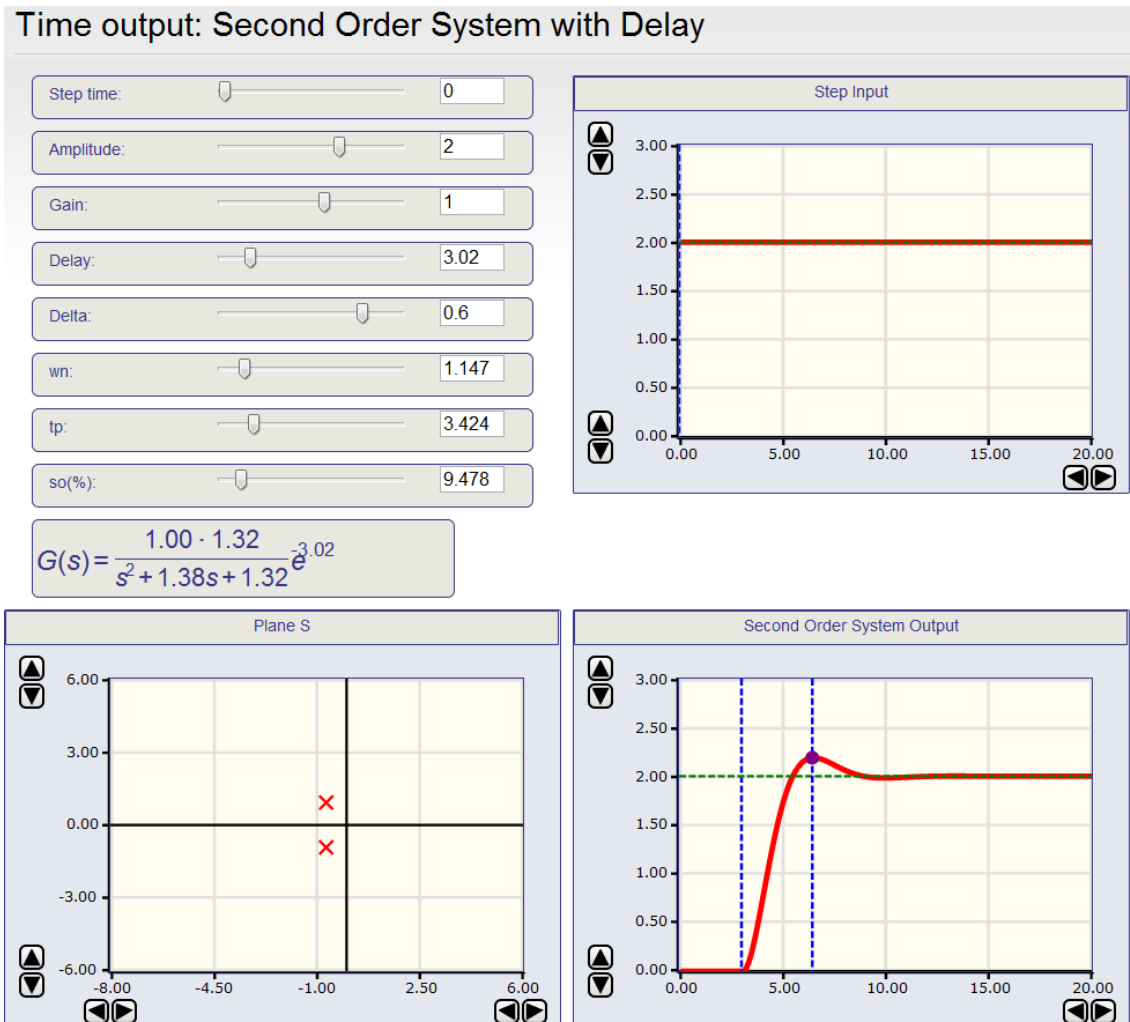


Figura 56. Herramienta interactiva de un sistema de segundo orden con retardo

Las instrucciones necesarias para la creación de esta herramienta son similares que las utilizadas en la sección 5.4.1. Los cambios realizados respecto a las instrucciones son:

- Se elimina la línea (82), visualizando de esta forma el control y la línea interactiva en la gráfica del tiempo de retardo del sistema.
- Se añaden instrucciones de configuración para el tiempo de retardo del sistema similares a las líneas (74-80).

### 5.4.5 Sistema de segundo orden con el efecto de un cero

Se ha creado una herramienta interactiva utilizando la librería creada en este proyecto que describe el efecto que tiene un cero sobre un sistema dinámico de segundo orden sin retardo con una señal de entrada en escalón. El resultado obtenido se muestra en la Figura 57 donde se puede observar la similitud con el sistema de segundo orden sin retardo, las principales diferencias que se pueden ver son:

- En la **parte superior izquierda** de la herramienta se encuentran los controles de las variables interactivas, se añade a los ya descritos en la sección 5.4.3 un control para modificar el valor absoluto del cero.
- La gráfica situada en la **parte superior derecha** es igual que la gráfica de la herramienta interactiva de un sistema de segundo orden.
- A la gráfica situada en la **parte inferior izquierda** se le añade un elemento interactivo, una pequeña circunferencia de color rojo que representa al cero en el plano  $s$ .
- Situado en la **parte inferior derecha** se encuentra una gráfica que representa la salida del proceso de segundo orden sin el efecto del cero en color naranja, en color rojo la salida del proceso de segundo orden con el efecto del cero, y en color morado el efecto del cero que tiene sobre la señal, en esta parte también se añaden los elementos interactivos que están descritos en la sección 5.4.3, pero en este caso se añade un punto de color gris interactivo que modifica el valor del cero cuando se modifica su posición.

Las instrucciones necesarias para la creación de esta herramienta son similares que las utilizadas en la sección 5.4.1. Los cambios realizados respecto a las instrucciones son las mismas que para la sección 5.4.3 pero añadiendo las siguientes líneas para modificar las propiedades del control que modifica el cero:

```
(1)    system.cero.setValue(0.2);
(2)    system.cero.setMin(-20);
(3)    system.cero.setMax(20);
(4)    system.cero.setStep(0.001);
(5)    system.cero.x = 30;
(6)    system.cero.y = 240;
(7)    system.cero.updateControl();
```

## Time Output: Zero effect on Second Order System

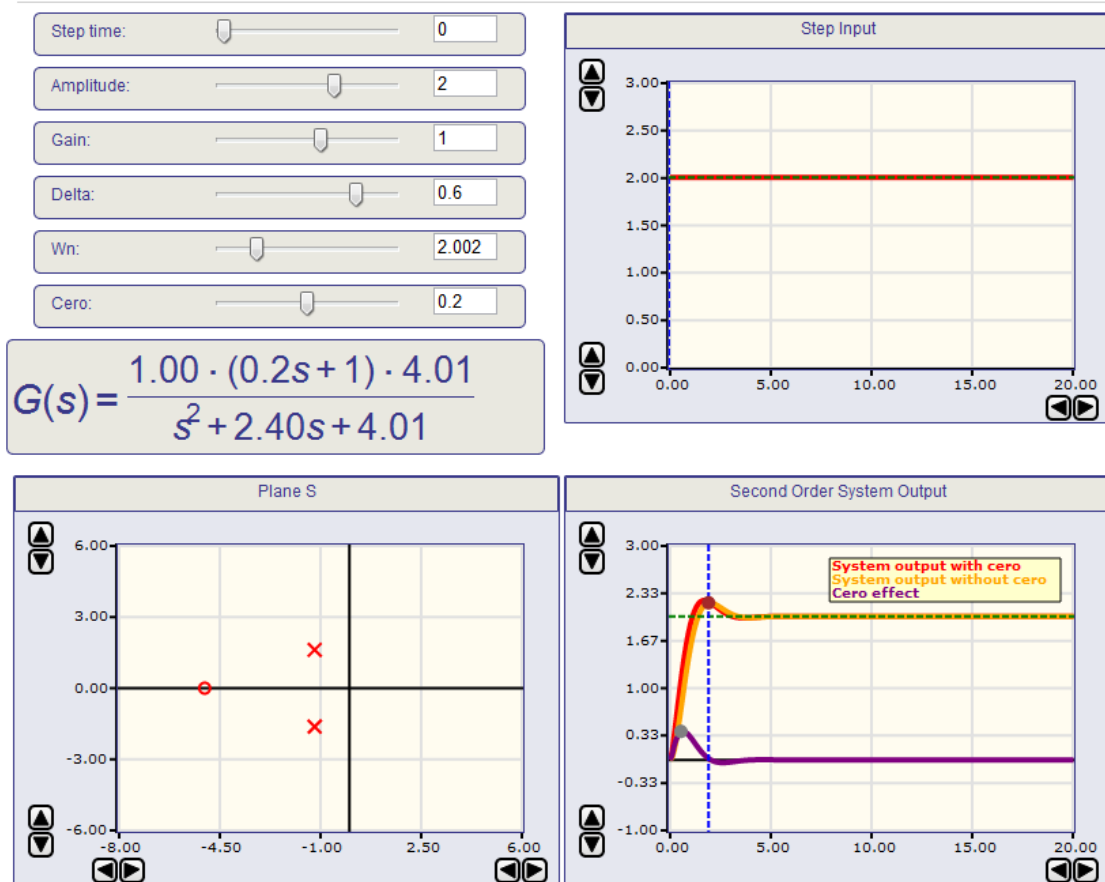


Figura 57. Herramienta interactiva de un sistema de segundo orden con un cero

## 5.4.6 Control PID

Esta herramienta interactiva describe el análisis de un controlador PID sobre un sistema dinámico de primer orden con una señal de entrada en escalón. El resultado se muestra en la Figura 58 diferenciándose las siguientes partes:

- En la **parte superior izquierda** se encuentra un área con los controles interactivos compuestos por una etiqueta un botón deslizador y una campo de entrada que modifican algunos parámetros principales del proceso de primer orden, la constante de tiempo y el tiempo de retardo, y las distintas constantes que forma parte del controlador, constante proporcional, constante integral y constante derivativa. Después de los distintos controles se sitúan dos formulas, una correspondiente al sistema de primer orden y otra al controlador.
- Una gráfica situada en la **parte superior derecha** que muestra la señal de salida del controlador en color morado. Esta gráfica no tienen ningún elemento interactivo pero se puede modificar el escalado pulsando los botones situados en la parte izquierda superior e inferior y la parte inferior derecha.

- Una gráfica situada en la **parte inferior** se encuentra una gráfica donde se representa la salida del proceso de primer orden con una línea en color azul, la señal de referencia con una línea punteada de color rojo y con una línea en color verde la señal de la perturbación. Esta gráfica tiene además dos elementos interactivos, un punto de color marrón que permite modificar el valor de la señal de referencia y un punto amarillo que permite modificar tanto el valor de la señal de la perturbación como el momento en el que afecta al sistema. Como las demás gráficas esta dispone de botones situados en la parte izquierda superior e inferior y la parte inferior derecha que permiten cambiar el escalado de la gráfica.

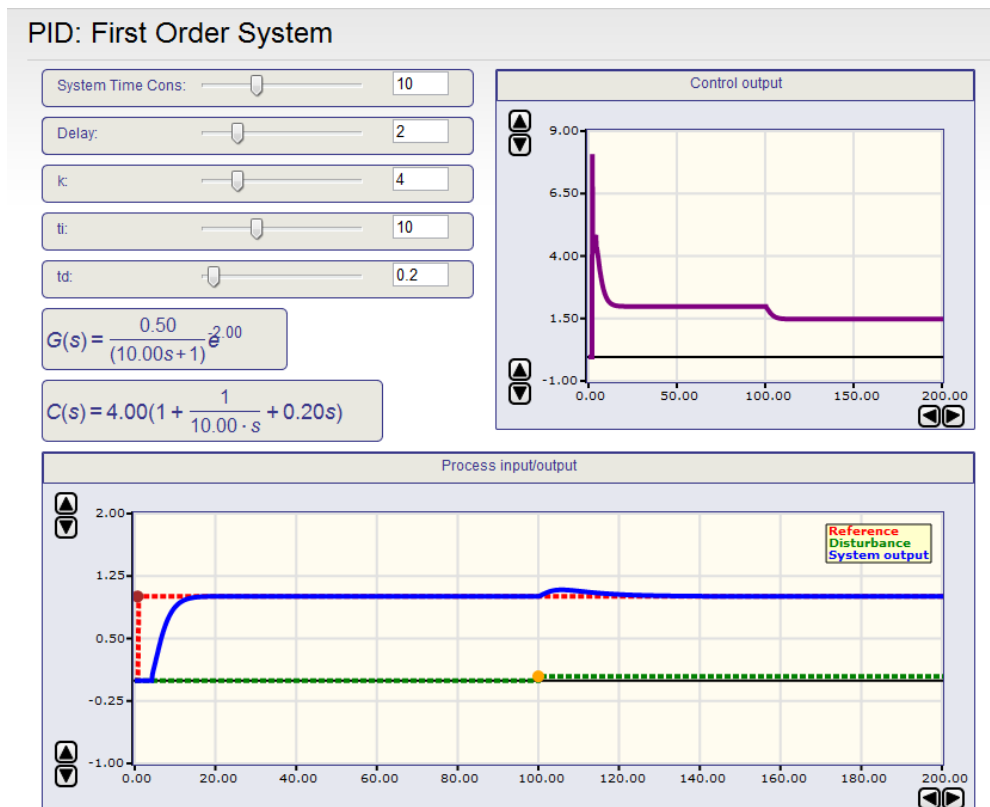


Figura 58. Herramienta interactiva de un controlador PID sobre un sistema de primer orden

Las instrucciones necesarias para generar esta herramienta son muy parecidas a las vistas en la sección 5.4.1, las principales diferencias son:

- En este caso en vez de crear una señal de escalón que es la señal de entrada al sistema, se crean dos señales en escalón, una para la señal de referencia y otra para la perturbación.

```

(1)  step = new
      Step({text:"Reference",length:length,sampling:sampling});
(2)  step.generateData();
(3)
    
```

```
(4) disturbance = new
    Step({text:"Disturbance",length:length,sampling:sampling});
(5) disturbance.amplitude.pushValue(0.05);
(6) disturbance.steptime.pushValue(100);
(7) disturbance.generateData();
```

- Se crea un controlador PID sobre un sistema de primer orden y no un sistema de primer orden.

```
(1) system = new PIDFOS({text:"PID First Order
    System",input:step,length:length,sampling:sampling,disturbanc
    e:disturbance});
(2) system.generateData();
```

#### 5.4.7 Página web y pruebas

Para agrupar todas las herramientas y realizar pruebas en distintos dispositivos se ha creado una página web que se muestran en la **¡Error! No se encuentra el origen de la referencia.** y ue es accesible desde la dirección <http://aer.ual.es/webcontrol/>. En esta página web se compone, de arriba hacia abajo, de una cabecera donde se muestra el título de la página, un menú por el que navegar para seleccionar las distintas herramientas interactivas creadas en este proyecto y la herramienta interactiva que ha sido seleccionada en el menú en sí.

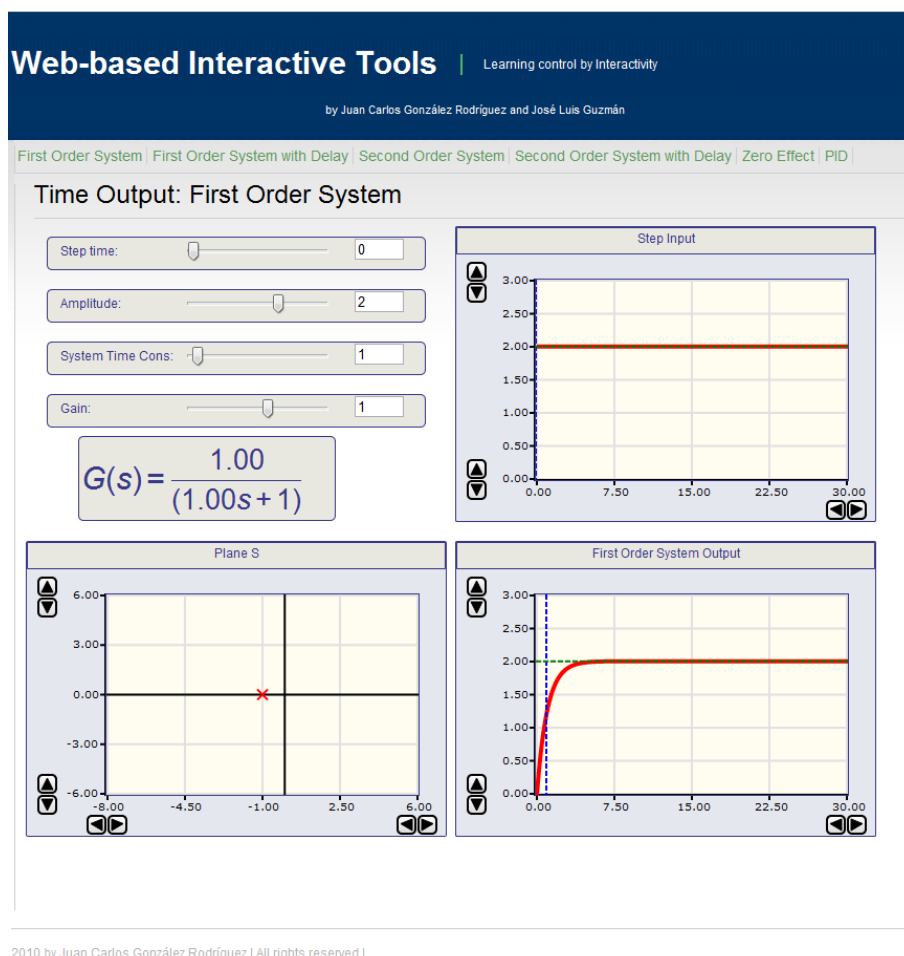


Figura 59. Página web con herramientas interactivas creadas en el proyecto

## HERRAMIENTAS DESARROLLADAS

Se han realizado distintas pruebas en distintos dispositivos, siempre intentando que funcionará perfectamente en un computador personal con un navegador web Chrome, debido a que éste es uno de los que más adelantado tienen la implementación de los últimos estándares W3C, pero también se han probado en otros dispositivos con mayor o mejor resultado como un iPad (Figura 60) o móviles con sistemas operativos Android (Figura 61).

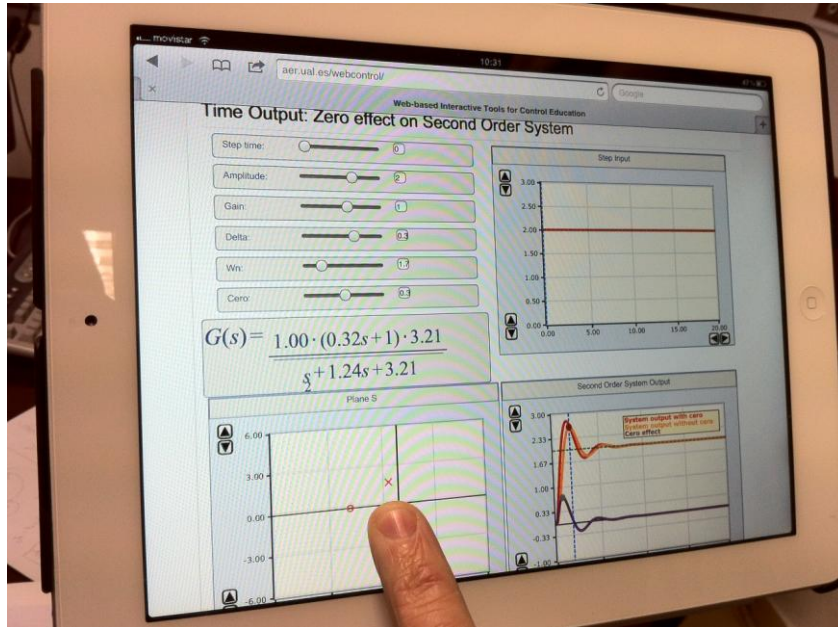


Figura 60. Prueba de herramientas interactivas sobre un iPad

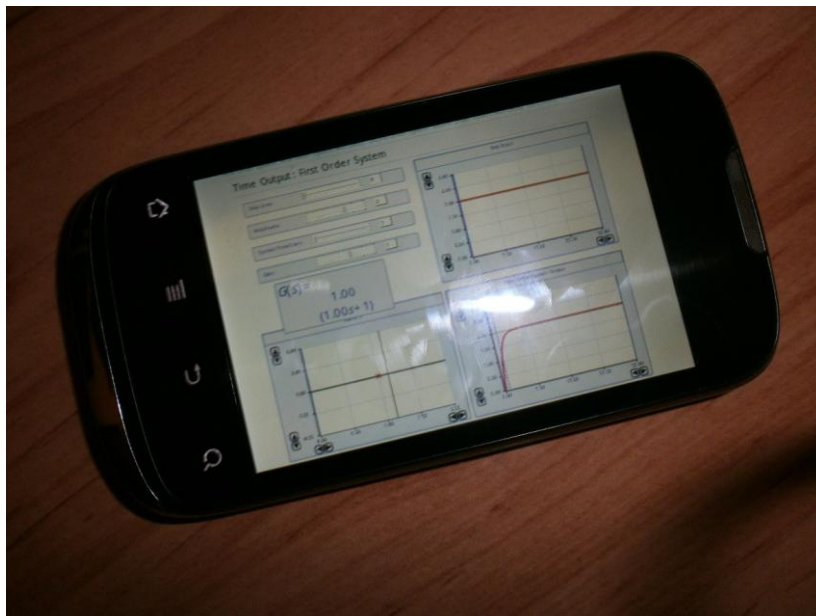


Figura 61. Prueba de herramientas interactivas sobre un móvil con Android

## 5.5 Python y AJAX

Existen procesos que se quiere representar gráficamente es complejo de implementar o los cálculos puedan ser muy costosos para un dispositivo en concreto. Por ello, se ideó una forma de obtener los datos que son representados en las gráficas a través de internet. Esta forma consiste en crear servicios web que realizaran estos cálculos costosos y mediante AJAX realizar peticiones a los servicios web para obtener los datos cuando fuera preciso.

Se realizaron unas pruebas preliminares para obtener datos y decidir su viabilidad. Se optó por utilizar una librería Python que simula el comportamiento de las funciones que tiene Matlab para calcular sistemas de control [22]. Para crear los servicios web se utilizó un sistema Linux instalando un servidor web Apache e instalando el modulo de Python de Apache. Se crearon dos scripts en Python (véase anexo Código Python) y una herramienta interactiva con código AJAX (véase sección 4.2.3.5) para llamar al servicio web. En la Figura 62 se muestra una tabla con el tiempo de las respuestas a cada petición sobre el servicio web. Se puede observar que los tiempos de respuesta no pasan de los 40 ms para un ordenador personal en la misma red que el servidor, téngase en cuenta que estos tiempos pueden ser mayores si se encuentran en redes distintas. Dados estos tiempos se observaba un pequeño efecto de congelado de menos de un segundo de la gráfica al interactuar con la herramienta interactiva.

Name	Method	Status	Type	Size	Time	Timeline	407ms
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	40ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	12ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	37ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	34ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	18ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	16ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	14ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	11ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	28ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	10ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.86KB	11ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	24ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	21ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	13ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	13ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	9ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	20ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	12ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	11ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	9ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	27ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.85KB	21ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.84KB	16ms		
<input type="checkbox"/> matlab.py	POST	200	text/plain	1.84KB	13ms		

Figura 62. Tiempo de respuestas de herramienta interactiva al obtener los datos de un servidor

Debido a que la interactividad se veía afectada al utilizar esta forma de obtención de datos y a que las herramientas a desarrollar dentro de este proyecto no requerían de cálculos costosos ni dificultad en la implementación, se optó por dejar esta parte como un trabajo futuro.



## 5.6 Interfaz gráfica para crear herramientas interactivas

Se decidió desarrollar una interfaz gráfica que utilizará la librería desarrollada para crear las herramientas interactivas. Debido a la amplitud del proyecto se hizo una versión preliminar de esta interfaz. Esta interfaz gráfica consistiría en una barra de herramientas que incluya todos los elementos gráficos incorporados en la librería y que permitirá crear herramientas simplemente arrastrando y dejando sobre un marco de trabajo los elementos gráficos a utilizar. En la Figura 63 se muestra el aspecto de una herramienta interactiva creada con la versión alfa de este entorno, donde en la parte superior en color verde se encuentra la barra de herramientas.

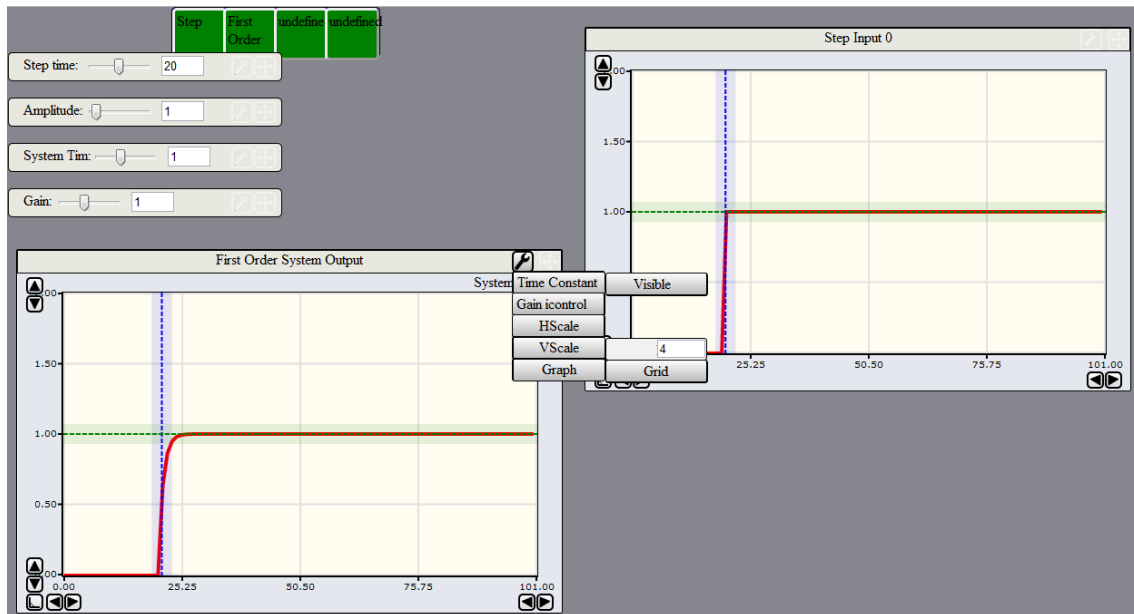


Figura 63. Interfaz gráfica para crear herramientas interactivas

### 5.6.1 Descripción de clases

En este apartado se describen las clases diseñadas para la interfaz gráfica. Debe de tenerse en cuenta que esta parte del proyecto se encuentra en una fase prematura de su desarrollo completo y que se ha planteado dejarlo como trabajo futuro.

#### *AutoScaleButton*

Clase que crea un botón que realiza la acción de auto-escalar la gráfica del *frame* donde se inserta el botón llamando al método por defecto *autoScale* de las escalas vertical y horizontal de la gráfica. Esta clase hereda de *IButton*. En la Figura 64 se muestra el aspecto del botón generado.



Figura 64. Botón generado por clase AutoScaleButton

### IMoveButton

Clase que crea un botón que permite mover el elemento gráfico donde se encuentra, en principio mueven los controles de variables interactivas y los marcos donde se encuentran las gráficas. En la Figura 65 se muestra el aspecto del botón generado.



Figura 65. Botón generado por la clase IMoveButton

### 5.6.2 PropertiesButton

Clase que crea un botón que despliega el menú generado por la clase *PropertiesMenu*. En la Figura 64 se muestra el aspecto del botón generado.



Figura 66. Botón generado por la clase PropertiesButton

### 5.6.3 PropertiesMenu

Esta clase genera una lista de las propiedades que son modificables de un elemento según las instrucciones del método *getProperties*. En la Figura 67 se muestra el aspecto del menú generado.

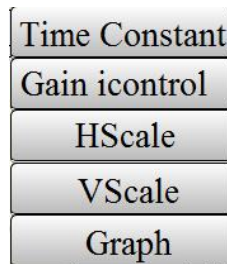


Figura 67. Menu generado por la clase PropertiesMenu

### 5.6.4 SelectList

Se trata de una clase abstracta para crear listados de elementos de clase *Data*. Su principal función es seleccionar un elemento de una lista generada.

### 5.6.5 SelectListFirstOrderSystem

Esta clase hereda de la clase *SelectList* y genera un listado de los objetos del tipo *Data* creados que pueden ser seleccionados para ser los datos entrada de un sistema de primer orden. En la Figura 68 se muestra el aspecto del listado generado.

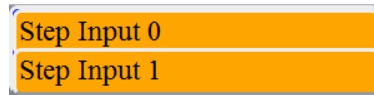


Figura 68. Lista generada por la clase SelectListFirstOrderSystem

### 5.6.6 ToolBar

Esta clase genera una barra de herramientas con elementos de la clase *ToolBarItem*. En la Figura 69 se muestra el aspecto de la barra de herramientas generada.

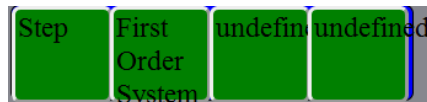


Figura 69. Barra de herramientas generada por la clase ToolBar

### 5.6.7 ToolBarItem

Clase abstracta para crear elementos que se incluirán en la barra de herramientas creada por la clase *ToolBar*.

### 5.6.8 ToolBarItemFirstOrderSystem

Esta clase hereda de la clase *ToolBarItem* y genera un elemento (botón) que se incluirán en la barra de herramientas creada por la clase *ToolBar*. Esta clase contiene las acciones para crear todos los elementos gráficos que pueden ser obtenidos a partir de un sistema de primer orden. En la Figura 70 se muestra el aspecto del elemento generado.

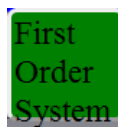


Figura 70. Elemento de la barra de herramientas generado por la clase ToolBarItemFirstOrderSystem

### 5.6.9 **ToolBarItemStep**

Esta clase hereda de la clase *ToolBarItem* y genera un elemento (botón) que se incluirán en la barra de herramientas creada por la clase *ToolBar*. Esta clase contiene las acciones para crear todos los elementos gráficos que pueden ser obtenidos a partir de una señal de entrada en escalón . En la Figura 70 se muestra el aspecto del elemento generado.



**Figura 71.** Elemento de la barra de herramientas generado por la clase *ToolBarItemStep*





## CONCLUSIONES Y TRABAJOS FUTUROS

En este proyecto se ha presentado una librería capaz de generar herramientas interactivas que puedan ser ejecutadas en distintos tipos de dispositivos a través de internet. Las capacidades interactivas de las herramientas creadas son similares a las de otros entornos de desarrollo de este tipo de herramientas conocidos, tales como Sysquake y EJS, pero con la ventaja de que estas nuevas herramientas interactivas puede ser ejecutadas solo usando un navegador web y sin el uso de maquinas virtuales o ficheros ejecutables. El uso de los estándares W3C para el desarrollo de la librería hace que sea compatible con la mayoría de los navegadores web del mercado, por lo que este aspecto hace que las herramientas interactivas creadas tenga una gran capacidad para ser incorporadas en cualquier página web. Por tanto, estas herramientas se podrían incorporar en cualquier lugar, por ejemplo, en una página web de un curso junto con las explicaciones teóricas, junto con una videoconferencia o un chat donde los alumnos puedan intercambiar sus opiniones. Además, gracias a los estándares utilizados es posible utilizar dichas herramientas desde cualquier dispositivo móvil, que era uno de los objetivos principales de este trabajo.

Como se ha mencionado en este documento se han iniciado el desarrollo de algunas partes de la librería o elementos externos a esta que se han dejado como trabajos futuros debido a la cantidad de trabajo que requieren. A continuación, se describen los trabajos futuros a partir de lo desarrollado en este proyecto.

- Obtención de datos mediante AJAX. Aunque la librería funcionan correctamente para el desarrollo de herramientas interactivas básicas, se presentan limitaciones para el desarrollo de herramientas más avanzadas con cálculos matemáticos complejos. Un trabajo futuro será el mejorar la obtención de datos mediante AJAX conectando con un servidor que realice estos cálculos usando otras librerías Python (sección 5.4.7), creando servicios web que permitan obtener los datos de sistemas complejos enviando los parámetros del sistema.
- Interfaz gráfica de usuario para la creación de herramientas. Esta herramienta está enfocada a la docencia en la ingeniería, por tanto, un profesor de una ingeniería que no tenga conocimientos en informática le resultaría dificultoso utilizar la librería para crear una herramienta interactiva. Por este motivo es una buena idea crear una interfaz gráfica de usuario que facilite la utilización de la librería para crear herramientas interactivas. En la sección 0 se puede ver una muestra del aspecto que tendría la interfaz gráfica.
- Mejora de la librería. Como resultado de este proyecto se ha obtenido la versión 1 de la librería, quedando para el futuro tareas como ampliar sus funcionalidades, optimizar sus procesos internos, mejorar su estructuración y añadir nuevos procesos y simulaciones orientados al control automático.



---

**REFERENCIAS**

- [1] J. Aracil, S. Dormido, F. Gordillo, y S. Dormido Canto. An interactive tool for introductory nonlinear control systems education. En 17th IFAC World Congress, Barcelona (España), 2002.
- [2] O.E Blanco. Estrategias de evaluación que utilizan los docentes de la carrera de Ingeniería Básica Integral de la Universidad de los Andes-Táchira. Tesis Doctoral, Universidad Rovira i Virgili, 2003.
- [3] P. Booth. An Introduction to Human-Computer Interaction. Lawrence Erlbaum, 1989.
- [4] D. Bucci, J. Sánchez, S. Dormido, P. Mullhaupt, y D. Bonvin. Interactive 3d Simulation of control systems: The spidercrane as a case study. En 44th IEEE Conference on Decision and Control and European Control Conference, Sevilla, 2005.
- [5] <http://caniuse.com>
- [6] S. Dormido, J. Aranda, y J.M. Díaz. Interactive computer-aided control design using quantitative feedback theory: the problem of vertical movement stabilization on a highspeed ferry. *International Journal of Control*, 78(11):813\_825, 2005.
- [7] S. Dormido. Control learning: present and future (plenary lecture). En Proceedings of the 17th IFAC World Congress, Barcelona (España), 2002.
- [8] S. Dormido. The role of interactivity in control learning (plenary lecture). En Proceedings of the IFAC Symposium on Advances in Control Education ACE03, Oulu (Finlandia), 2003.
- [9] S. Dormido. Control learning: Present and future. *Annual Reviews in Control*, 28(1):115\_136, 2004.
- [10] F. Esquembre. Easy Java Simulations 3.1. <http://fem.um.es/Ejs/>, 2002.
- [11] R.S. Gilliver, B. Randall, , y Y.M. Pok. Learning in cyberspace: shaping the future. *Journal of Computer Assisted Learning*, 14(14):212\_22, 1998.
- [12] J. L. Guzmán, K. J. Åström, S. Dormido, T. Hägglund, M. Berenguel, y Y. Piguet. Interactive learning modules for pid control. *IEEE Control System Magazine*, 28(5):118\_134, 2008.
- [13] ] J. L. Guzmán, S. Dormido, y M. Berenguel. Interactive teaching of constrained generalized predictive control. *IEEE Control Systems Magazine*, 25(2):79\_85, 2005.
- [14] J. L. Guzmán, Y. Piguet, T. Hägglund, S. Dormido, y K. J. Åström. Interactive learning modules for pid control. En Proceedings of the 7th IFAC Symposium on Advances in Control Education ACE06, Madrid (España), 2006.
- [15] J. L. Guzmán, F. Rodríguez, J. D. Álvarez, M. Berenguel, y S. Dormido. Ntic's en el ámbito de la ingeniería: un ejemplo de puesta en práctica. En Jornadas sobre la influencia de las



## REFERENCIAS

---

- nuevas tecnologías de la información y la comunicación en el campo docente en estudios de ingeniería, Almería (España), 2005.
- [16] J. L. Guzmán. Interactive Control System Design. Tesis Doctoral, Universidad de Almería, 2006.
- [17] B.S. Heck. Enhancing classical control education via interactive gui design. IEEE Control System Magazine, 19(3):35\_58, 1999.
- [18] National Instrument. LabVIEW Distance Learning Solution Guide. Universidad de Almería, <http://www.ni.com/pdf/academic/us/>, 2003.
- [19] M. Johansson, M. Gäfvert, y K. J. Åström. Interactive tools for education in automatic control. IEEE Control Systems Magazine, 18(3):33\_40, 1998.
- [20] D. Jonassen. Instructional Designs for Microcomputer Courseware. Lawrence Erlbaum, 1988.
- [21] J.P. Keller. Interactive control system design. Control Engineering Practice, 14:177\_184, 2006.
- [22] H.Ordiales. <http://h.ordia.com.ar/blog/index.php?entry=entry061127-202822>
- [23] L. Perera. <http://laktek.com/2011/02/02/understanding-prototypical-inheritance-in-javascript/>
- [24] Y. Piguet. SysQuake: User Manual. Calerga S'arl, <http://www.calerga.com/doc/>, 2004.
- [25] RAE. Real Academia Española de la Lengua. <http://www.rae.es/>.
- [26] A. Rauschmayer. <http://www.2ality.com/2011/06/prototypes-as-classes.html>
- [27] F. Rodríguez, M. Berenguel, J.L. Guzmán, Arahal, M.R.; 2004; Prácticas de control por computador; Universidad de Almería; 302 pp
- [28] D.M. Rhodes y J.W. Azbell. Designing interactive video instruction professionally. Training and Development Journal, 39(12):31\_33, 1985.
- [29] J. Sánchez, S. Dormido, y F. Esquembre. The learning of control concepts using interactive tools. Computer Applications in Engineering Education, 13(1):84\_98, 2005.
- [30] J. Sánchez. 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. Tesis Doctoral, Universidad Nacional de Educación a Distancia (UNED), 2001.
- [31] R. Sims. Interactivity: A forgotten art? <http://www2.gsu.edu/~wwwitr/docs/interact/>, 1995.
- [32] S. Skogestad y I. Postlethwaite. Multivariable Feedback control: Analysis and Design. Wiley, 1996.

- [33]D. Svanaes. Understanding Interactivity: Steps to a Phenomenology of Human-Computer Interaction. Tesis Doctoral, NTNU, Trondheim (Noruega), 2000.
- [34]B. Wittenmark, H. Häglund, y M. Johansson. Dynamic pictures and interactive learning. IEEE Control Systems Magazine, 18(3):26\_32, 1998.
- [35] World Wide Web Consortium (W3C). <http://www.w3.org/>
- [36]World Wide Web Consortium (W3C). Core. <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>
- [37]World Wide Web Consortium (W3C). CSS. <http://www.w3.org/TR/CSS2/>
- [38] World Wide Web Consortium (W3C). Events. <http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/>
- [39]World Wide Web Consortium (W3C). HTML. <http://www.w3.org/TR/1999/REC-html401-19991224/>
- [40]World Wide Web Consortium (W3C). HTML5. <http://www.w3.org/TR/html5-diff/>
- [41]World Wide Web Consortium (W3C). MathML. <http://www.w3.org/TR/2003/REC-MathML2-20031021/>
- [42]World Wide Web Consortium (W3C). SVG. <http://www.w3.org/TR/SVG11/>
- [43]World Wide Web Consortium (W3C). XHTML. <http://www.w3.org/TR/2010/REC-xhtml-basic-20101123/>



## ANEXO

## Plantilla CSS MathML

```

@namespace "http://www.w3.org/1998/Math/MathML";
math
  {line-height:1.3em;
  text-indent:0;}
math[display="block"]
  {display:block;
  text-align:center;
  page-break-inside:avoid;}
mfrac
  {display:inline-table;
  white-space:nowrap;
  border-collapse:collapse;
  text-align:center;
  /*Para chrome antes vertical-align:0.9em;*/
  vertical-align:-0.9em;
  margin:0 2px;
  font-size:1em;}
mfrac > *
  {line-height:1.3em;
  font-size:0.9em;}
mfrac > *:first-child
  {display:inline-table;
  vertical-align:text-bottom;}
mfrac > * + *
  {border-top:solid thin;
  display:table-row;}
mfrac[linethickness="0"] > * + *
  {border-top:none;}
mfrac[linethickness="2"] > * + *, mfrac[linethickness="thick"] > * + *
  {border-top:solid medium;}
mfrac[numalign="left"] > *:first-child, mfrac[denalign="left"] > * + *
  {text-align:left;}
mfrac[numalign="right"] > *:first-child, mfrac[denalign="right"] > * + *
  {text-align:right;}
msub, msup, msubsup, mmultiscripts
  {display:inline-table;
  line-height:0.4em;}
msubsup, msup, mmultiscripts
  {margin-top:0.4ex;
  table-baseline:2;}
msubsup, msub, mmultiscripts
  {margin-bottom:0.4ex;}
msubsup, msup
  {direction:rtl;}
msub > *
  {display:table-row;}
none
  {content:"\A0";}

```

```

msubsup > *, msup > *
/*Para chrome antes {display:table-row;
direction:ltr;
text-align:left;};*/
{display: table-header-group;
direction:ltr;
text-align:left;}
mmultiscripts > *
{display:none;}
mmultiscripts > *:first-child, mmultiscripts > mprescripts + *
{display:table-row;}
mmultiscripts > mprescripts + * + *
{display:table-header-group;}
msub > *:first-child:after, msub > * + *:before,
msubsup > *:first-child:before, msup > *:first-child:before,
mmultiscripts > *:first-child:before
{display:table-cell;
content:"\A0";}
msubsup > * + * + *, msup > * + *
/*Para chrome antes {display:table-header-group;};*/
{display:table-caption;
text-align: right;}
msub > * + *, msup > * + *, msubsup > * + *, munder > * + *, mover > * + *,
munderover > * + *,
mmultiscripts > * + *
{font-size:0.7em;}
munder, munderover, mover
{display:inline-table;
margin:1px;
text-align:center;}
munder > *, munderover > *, mover > *
{display:table-row;}
mover > * + *, munderover > * + * + *
{display:table-header-group;}
mover, munderover
{table-baseline:2;}
msqrt
{display:inline-block;
margin:1px 0 1px 22px;
border-top:solid 1px;
border-left:groove 2px;
padding:2px 5px 0 0;}
msqrt:before
{display:inline-block;
vertical-align:bottom;
content:"";
width:22px;
height:14px;
background-repeat:no-repeat;
margin:0 3px 0 -22px;}
msqrt:before, mroot > * + *:after

```

```

        {background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABYAAAAOCAYAA
AArMezNAAAArkIEQVR42mNgIAzkgJiRgYpAGlgnAfFvIF4JxKyUGsgOxKVA/ASI5wCxHRBv
A+K9QMxHrqGRQLweiDOBmBdJnAWI5wPxeSAWJ8VAPSBOWJTAuo6gPguEKsQY6gQE
POQ4lh8IH4JxCYMNABhQPwOiN1pYbgzEL8C4hhaGG4AxM+BuJgWhisB8W0g7qOF4SJA
fBqll1Ejl6EDbiDeAcQ7SUXIRAGQa5cA8RkGGGoFeACG1GGdqegJfAAAAAEIFTkSuQmCC");}
mroot
    {display:inline-table;
    direction:rtl;}
mroot > *
    {display:table-cell;
    direction:ltr;
    text-align:left;}
mroot > *:first-child
    {border-top:solid 1px;
    border-left:groove 2px;
    padding:2px 5px 0 3px;}
mroot > * + *
    {vertical-align:bottom;
    text-align:right;
    font-size:0.7em;
    line-height:1em;}
mroot > * + *:after
    {display:block;
    content:"";
    width:22px;
    height:14px;
    margin-right:-1px;
    margin-left:auto;}
mfenced
    {display:inline-table;
    border-collapse:separate;
    border-spacing:0.2ex 0;
    white-space:nowrap;
    margin:1px;}
mfenced > *:first-child
    {display:table-row;}
mfenced > * + *
    {display:none;}
mfenced > *:before, mfenced > *:after
    {display:table-cell;
    content:"\A0";
    background-repeat:no-repeat;
    background-size:100% 100%;}
mfenced > *:before
    {background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABQCAYAA
AAZQFV3AAAABtEIEQVR42s2ZK0hEURCGr8UiPhYVRJAFoxssFtsafDXRKLogNpNNDDazTSxi
MYhoMdtWsAibtKhssCj4ALWIoM4s58AwdB7gD3+45eO8z5z/FgWfnfEFAWoTz4tHCJhC
dsQDBGxKXBd3EbBI8a14kIBtiF+pMdsS/4pnCNic+Ed8SMAq4g/xM7HWetIEaFdrROtOEu
ySgE0kmHqB2FKNBLtP3yGtmtatRWGd4scEexF3RIHrpnV7xGRcG+BsFDZuYO/i9ihw3wC
Picn4NMDFKHDJwNS9UeCBgTWJ2W0a4GkUVnbd3YwCaw44TY6fOnyQ3hjYE3FufRngV

```

```

RQ45Lp7FgVWHXA3Clyhl8y2A4ZvtyMHnlwCzx1wLAq8cMDRKLdHgMNR4J0DlqPABwcs
RYFvDtgdBX47YPHvgHiX8UnBlw2+sPGthx8O+PGFH7D4FVA4t4EsKv0bxi16FlilqvFjCyzm8
4FShJbEfr6Rox58V+MNHhT7NVPjjUYU+b1X4AxyPCFRoiKHCYxYVGGrloVGVCG/TVGjcl4
UGkll0ZJqFhrpZaOychQbjWWH0b7co9nPBqvX74w/Kju1j59IjnwAAAABJRu5ErkJggg==")
;}
mfenced > *:after
    {background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABQCAyAA
AAZQFV3AAABl0IEQVR42s3ZPyhFYRjH8ddiEa5cJaVbd2SwWGwM/m1yR3FKNpNNBpvZl
ossBonFbKMs6k4s6A4Wyp/CloXfqXd4etbnOzj1Wz+97zvn3+ekIFJV6UvwM6TMK20k2q
9sZxx7upRzZYpEB5RbZYI+p6/KOonOKL/KJokeKD/KHAWWY/RZ+VCGKbTIXS8/VIVCLzN6
TIGNDJaZIMBySt5nsEIN0VXTyhUC7FBeMviodBLormnlGgHOGvCaANuVd4OOEeiRAfclC
MGAn8TH6TVgmUWiS0D7hPgiQFbBLjhul2LgtMOLKJg1YHie3wy4A0BXhnmKwFL2qnr9
mAU3HHgOD10lqNg4cCtKDjpwMMoOOrAsyg44sCLKFh3YDMK1hx4FwV7HPgQBbsd+E
bMZwt+/zsQ7zL+UfBhgw9sfOrhiwO+fOELL4F4JsUvo3iGz16FMEPS/hxDj9w4kdi9NCOX
yvw96NcMvj/j1Fr2A4yUCtliBl1nwQhBaqkKLaXi5Dy1loiVTtKiLp3Rwjhaukd/LqC/P/4A
wHbtY0GeJ20AAAASUVORK5CYII=");}
mfenced[open="[" > *:before
    {border-style:solid;
border-width:1px 0 1px 1px;
background-image:none;}
mfenced[close="]" > *:after
    {border-style:solid;
border-width:1px 1px 1px 0;
background-image:none;}
mfenced[open="\2016" > *:before
    {border-style:double;
border-width:0 0 0 3px;
background-image:none;}
mfenced[close="\2016" > *:after
    {border-style:double;
border-width:0 3px 0 0;
background-image:none;}
mfenced[open="\27e6" > *:before
    {border-style:double;
border-width:2px 0 2px 3px;
background-image:none;}
mfenced[close="\27e7" > *:after
    {border-style:double;
border-width:2px 3px 2px 0;
background-image:none;}
mfenced[open="|" > *:before
    {border-style:solid;
border-width:0 0 0 1px;
background-image:none;}
mfenced[close="|" > *:after
    {border-style:solid;
border-width:0 1px 0 0;
background-image:none;}
mfenced[open="\230a" > *:before
    {border-style:solid;
border-width:0 0 1px 1px;

```

```

background-image:none;}
mfenced[close="\230b"] > *:after
{border-style:solid;
border-width:0 1px 1px 0;
background-image:none;}
mfenced[open="\2308"] > *:before
{border-style:solid;
border-width:1px 0 0 1px;
background-image:none;}
mfenced[close="\2309"] > *:after
{border-style:solid;
border-width:1px 1px 0 0;
background-image:none;}
mfenced[open="{}"] > *:before
{content:"\A0\A0";
background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABQCAYAA
AAZQFV3AAABuklEQVR42rWZuy9EURCHj0c8Qta7pRMh0WgUqBQqG4leNGRL2VZpO6
GiYobHf+DhliSf8AjFBikmwiNRNZvsmdjtM53J/naL+fec+7MnLkhpMeoWBZDoi1F1C12R
EW0pK6qQ1yIY+AJQ5M4ES+inxCWru2UCFmXeBWfokAIN+PqMkJmG1GNwnlCWlyyL9
FJCI+i8lqQNcfNMoe2IZyNMmOJPHvGGCHMouW7NQE04iYKHwlZazwqJjwjhJPu/R0Swk
Un3CKEa05YJhOCsUoI95ywSAhPnXCGEF464QQhvHbCEUJ4SwwvnXCQED47YQ8hrNLcd
ydEUIfNkRwFJ3sjhONoe4IF5zWgBD61LVBCPFdb3GaSpUNil8o3CVWV4myu1Bv45J317
6QBzGdKlsRT6FeP9r/I7DPqU8MizmxLnpTVoQL8UfOfVNyOTa5HGz808slOeDpC0+weAk
IAS5FngZxQs93orgzRLezuENJ94S4007fq3ALz741Qy/POLXW/wCboGOCCzwlQY+ZsEHQ
fioyglDpoWQw7gPH0j6RFRQnyo689kiRLig/HGBmGj+0agPxd8/Pn98QMoXcSZw6bNQw
AAAABJRU5ErkJggg==");}
mfenced[close="{}"] > *:after
{content:"\A0\A0";
background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABQCAYAA
AAZQFV3AAABoklEQVR42sWZvUpDQRBGxx+iosS/2GoXJIKNjYVapbAyCPbBRrEMaS1Nj1
ppo1gEO30HQZGU+glxooWgQkC0ESR+iwOZBzjghdMe9t7dnZ39rplZRkyJDZE38OkTNXE
gRkxjubgRg5RwQryKC9FDSXdER1QpYVZ8iTcxTEnrPspdSIh0YZuaoCHx7dISNcqGC88o4b
4L0+T0EsJ1FyaWCeFsECJrMhWOHxfWqe/45MJ7SnjlwrSE+gnhafiO84RwLwjXCGE1CLcI4
WYQIoWiFIRHhHApCC8J4VwQ3hLCmSC8o4VNQpgLwhYhHA3CF1rYJoSZIPygKk4ngDzvQ
ZglhNdBWCCeJOG4SggrBpewBeseVsfUxBy68FNMEsLU1j24tEaNclE8+o4pUNIB+ztnnkXZ
wGdMblsVMS3GfZv+vxB9ZXR8GWDLmx86+HFAS9felHFjwD0kMKPUfygx1sRvFnC2zm
84cRbYrxpx68V+MUHv5rhl0f8eotfwNGIAA8x8JgFD4LwqAoNO/C4Dw8k0cgUD3XR2BkP
xrHoHvu5gP7++AUnfMSZodLlnwAAAABJRU5ErkJggg==");}
mfenced[open="\27e8"] > *:before, mfenced[open="\2329"] > *:before
{background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABZCAYAA
AA+TwS/AAAB50lEQVR42sZPyiFURiHj7+Lrkhks0goEqtFWSilW8oqg5GULExWk5K6g26Z
rFI3i24Go8kkkWQwSPkTA673lzOcvu7kfZSfnvTe77vnO+c834hsORgX+gjZR2WIVI4T8oa
LGukcNYyTgoPLfWUbNiyR1ZXtOQpWbvl0dJECfVmDyhZneXWMkclZywfIjZKeGlPuzKt2Y
plkRLuRmEXldMze70cUdWtxurWqalyE4UDhDafZzfUcMtRuEnlBqNMGSWEhSi7t9R4Za
2W1ygsENWtJMod9MpQLddR9mRp9Aqnk+r2ieEeJ8JZr6zf8hVl75Zmr3Anqa7klbVYXhLh
gle4nMg+LZ0emVbCVSI89VY3lCgqcWK7OMoluz2y3mSqKOf6rYz1W14ZJq4zxnhIE4IJH
deKfKRUA45RFOZGtKmedYysgews+2+St6MINFKXqq26oy3GmP8K6KsPVfVYg/Q/wt/8k8
xFcKvpb/5GuDfw8F/sUW6J4i8F0P35cFenIQ+NIGoKcvGZ8PBXqCffgZG78FhAdfUwR+kw

```



```

oBvusJ/DYqygG8Lwv8Ro/3HATaFRF430agnSWB974E2p0TeP8Q73AktAcr8C6xKAawjy3
wTrtA/wUI/G8F/j9FzNNC/J+U6KOFuW87Lt0vtp6NAwAAAABJRU5ErkJggg==");}
mfenced[close="\27e9"] > *:after, mfenced[close="\232A"] > *:after
  {background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAABQAAABZCAYAA
AA+TwS/AAABYElEQVR42s2ZPSjFURiH/z4XkZvIZpFQJFaLsIDqdkvdVQYjqZuFyWq6JXUH3
TJZpW4WYWA0mSS6yWCQ8hGDr9+pdzid7vY+g1PP+vSe//89H+97sizLOjN4jNDCCdFHS
5dp4aZol4WzokgKW8UxPe0DMUkKC6JKCjvEs+glpUf2x7GxJO5FCyXsEV9ikYzyTjYTwIXa
2scGQMm3CejvBQf9k2RsWVRbIDCMRPWYrS6MwMBeu6Y8lwSTpswME4Im8SjCstUIB
UTvosclZyPpl0ihO3ixYR3opmQHkZR5glhMRKeEslu8WnChzFKSGtRIHuEcCUSvolur7BffEf
SdSLKi0h4ayvJNUqRMLDgFQ4mwhNi2leRMKTQsFe4nUS56xVOJcJXS3zXqCfSNa+wnAiv
vSk0kwgDcx5hOfafEmHNO+1qlgwpNOQR5htMu+wR5hoIH/5VhOg3xP8ynof4SkHXMr7
b4PshumPjZwp+6qHnMn5zwO826OOLvx/iN1j0jo1XAXidglZSeK2HV6N4vYxW9HjPAe+K
oH0bvLOE975CmmDdObx/iHc40R4s3iXG+9hopx1/C8BfK/D3FPTFB3+Twl/N0He9P0bv3
S+iamc2AAAAAElFTkSuQmCC");}
mfenced[open=""] > *:before, mfenced[close=""] > *:after
  {content:normal;}
mover > * + mo[fence="true"], munder > * + mo[fence="true"]
  {content:"\A0";
  line-height:1ex;
  background-size:100% 100%;
  background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAAFAAAAAUCAYAA
AAa2LrXAAABxEIEQVR42u2YsUsCURzHnxIncmJlTtYwkeDi4pBNdu6J0C4tRmPc2phb2KS
L0iBt9T8EhtJWP2CFDoInCFKLiNe3+gaH1ekdZ6S8L3yWd+/d7/v7+t5xpD/WwqRcqhlU
VQa0TKoXaillANlwCHJyDjsaQs0wRjPckxqhFZADnSBZhrXOJbjHCKoyF2VBFiQBC/AAM/AZ
5rr45jBOUWuSflewWkNYR5EwR4bOgEfcA2q4B7UwRNocRf1GMRvPIL4D7XivGa1tscaL
das00OVngr0mKXnKHv4E82BBI9TBTyA/oiG7DAA50C18KByzDFun32UmFvCfbqihZACLw
A3UXT73TALSIBYxCz4SvGNSXeo+OyN509p5iBbanc6l2L3dIAN6AMTvnrbHbDoNoiAdRIGi2
SSn2aKqU7YVD9CTyl61Oi5zB4aFru6yyzUcQx4eHN9alvXwBlIg00xm9+nCntLs9fa0CNKZz
Yeq+N6xcmvfCbsAv8MhjWu/MygkwMzVtTWAFAHWiDiZHFrwUTVJDZtJlVwHwxDy5BS
OY0UiFmlf8a2BCfb/hS9pRjdmifeGUetuVldmJVZuFYH9nJv8yds3kDn8bEmYaXZFsAAAA
SUVORK5CYII=");}
munder > * + mo[fence="true"]
  {background-
image:url("data:image/png;base64,iVBORw0KGgoAAAANSUgAAAFAAAAAUCAYAA
AAa2LrXAAABzEIEQVR42u2Xv0tCURzFrXlPRbEyW2uLSHBxccimBqdEaJcWozFcG3MLm3
RRGqSt/ogEEf7B6zQQVBBkFoEsfPsCA8zX+9m0Y974Lnc773f7znvPrIPCCe0oSSrXYrKg
dpDbPbA3aVhWXZmZ1YB2mVh2Wlmd1QGxAbfCoXU/mYVcY46AG3oAkOgVfI9EzeZtN
kVp7xCu5wCQbgGRTBDnD949BczKDITAbMyPneAhtlgRYn6/RAGZyCONgQf/Pqo9FbnF7
L9D7KocVsbB/ZzA2OQcewgZE+qIFrUAAn3HwfxMAWCIA14gcLRPviEEZ1/lb6AfYUY48p9l
yghxo9TfLaYRZuMYaLHo+dipnQRvcgDw4AiELfYW4Js892jPurUXPsWmvq1XNgQifnv6fc
Dd2xD+LfgLOTJ60m3P6M6zbo5civUXo9Vs0D4JgFyR51LPgCpRABVTBA2jwdeiaGLoH4Q
m1wvxt2touazRYs8oeSuwpyx6T7DlID79S+rVgE0RpKAeeGMQjcBjmOjg24Jwc10S5h7p2
Ucvi9Ybf4es0Uopjac5RMpF+qupgkdQ5pmRBCXBAEioOOW0TJUmtEiVJLREISWnih393v
wCZoSzSn592GQAAAABJRU5ErkJggg==");}
menclose
  {display:inline-table;
  border-collapse:separate;
  border-spacing:0.4ex 0;}
menclose[notation="top"]
  {border-top:solid thin;}

```

```

menclose[notation="bottom"]
    {border-bottom:solid thin;}
menclose[notation="right"]
    {border-right:solid thin;}
menclose[notation="left"]
    {border-left:solid thin;}
menclose[notation="box"]
    {border:solid thin;}
menclose[notation="horizontalstrike"]
    {text-decoration:line-through;}
mtable
    {display:inline-table;
    line-height:1.5em;
    text-align:center;
    vertical-align:middle;}
mtr
    {display:table-row;}
mtd
    {display:table-cell;
    padding:0 0.5ex;}
mtable[columnalign="left"], mtr[columnalign="left"], mtd[columnalign="left"]
    {text-align:left;}
mtable[columnalign="right"], mtr[columnalign="right"], mtd[columnalign="right"]
    {text-align:right;}
mtable[rowalign="top"] mtd, mtable mtr[rowalign="top"] mtd,
math mtable mtr mtd[rowalign="top"]
    {vertical-align:top}
mtable[rowalign="bottom"] mtd, mtable mtr[rowalign="bottom"] mtd,
math mtable mtr mtd[rowalign="bottom"]
    {vertical-align:bottom}
mtable[rowalign="center"] mtd, mtable mtr[rowalign="center"] mtd,
math mtable mtr mtd[rowalign="center"]
    {vertical-align:middle}
mtable[frame="solid"]
    {border:solid thin;}
mtable[frame="dashed"]
    {border:dashed thin;}
mtable[rowlines="solid"], mtable[rowlines="dashed"],
mtable[columnlines="solid"], mtable[columnlines="dashed"]
    {border-collapse:collapse;}
mtable[rowlines="solid"] > mtr + mtr
    {border-top:solid thin;}
mtable[rowlines="dashed"] > mtr + mtr
    {border-top:dashed thin;}
mtable[columnlines="solid"] > mtr > mtd + mtd
    {border-left:solid thin;}
mtable[columnlines="dashed"] > mtr > mtd + mtd
    {border-left:dashed thin;}
mspace[linebreak="goodbreak"]:before
    {content:"\200B";
    white-space:normal;}
mspace[linebreak="newline"]:before, mspace[linebreak="indentingnewline"]:before

```

```

        {content:"\000A";
        white-space:pre;}
mspace[width]:before
        {content:normal;}
mspace[width="verythinmathspace"]
        {padding:0 0.05em;}
mspace[width="thinmathspace"]
        {padding:0 0.08em;}
mspace[width="mediummathspace"]
        {padding:0 0.11em;}
mspace[width="thickmathspace"]
        {padding:0 0.14em;}
mspace[width="verythickmathspace"]
        {padding:0 0.17em;}
mo[largeop="true"]
        {font-size:1.3em;
        vertical-align:-0.1ex;}
mo[form="infix"], * + mo
        {padding:0 0.3ex;}
mo[form="prefix"]
        {padding:0 0 0 0.5ex;}
mo[form="postfix"]
        {padding:0 0.5ex 0 0;}
mo[fence="true"], mo[separator="true"]
        {padding:0;}
mi[mathvariant="bold"], mi[mathvariant="bold-italic"], mi[mathvariant="bold-sans-
serif"],
mi[mathvariant="sans-serif-bold-italic"],mn[mathvariant="bold"],
mn[mathvariant="bold-italic"],
mn[mathvariant="bold-sans-serif"], mn[mathvariant="sans-serif-bold-
italic"],mo[mathvariant="bold"],
mo[mathvariant="bold-italic"], mo[mathvariant="bold-sans-serif"],
mo[mathvariant="sans-serif-bold-italic"],
ms[mathvariant="bold"], ms[mathvariant="bold-italic"], ms[mathvariant="bold-sans-
serif"],
ms[mathvariant="sans-serif-bold-italic"],mtext[mathvariant="bold"],
mtext[mathvariant="bold-italic"],
mtext[mathvariant="bold-sans-serif"], mtext[mathvariant="sans-serif-bold-italic"]
        {font-weight:bold;
        font-style:normal;}
mi[mathvariant="monospace"],
mn[mathvariant="monospace"],mo[mathvariant="monospace"],
ms[mathvariant="monospace"],mtext[mathvariant="monospace"]
        {font-family:monospace;
        font-style:normal;}
mi[mathvariant="sans-serif"], mi[mathvariant="bold-sans-serif"],
mi[mathvariant="bold-sans-serif"],
mi[mathvariant="sans-serif-italic"], mi[mathvariant="sans-serif-bold-
italic"],mn[mathvariant="bold-sans-serif"],
mn[mathvariant="sans-serif"], mn[mathvariant="bold-sans-serif"],
mn[mathvariant="sans-serif-italic"],

```

```

mn[mathvariant="sans-serif-bold-italic"], mo[mathvariant="sans-serif"],
mo[mathvariant="bold-sans-serif"],
mo[mathvariant="bold-sans-serif"], mo[mathvariant="sans-serif-italic"],
mo[mathvariant="sans-serif-bold-italic"],
ms[mathvariant="sans-serif"], ms[mathvariant="bold-sans-serif"],
ms[mathvariant="bold-sans-serif"],
ms[mathvariant="sans-serif-italic"], ms[mathvariant="sans-serif-bold-italic"],
mtext[mathvariant="sans-serif"],
mtext[mathvariant="bold-sans-serif"], mtext[mathvariant="bold-sans-serif"],
mtext[mathvariant="sans-serif-italic"],
mtext[mathvariant="sans-serif-bold-italic"]
    {font-family:sans-serif;
    font-style:normal;}
mi, mi[mathvariant="italic"], mi[mathvariant="bold-italic"],mi[mathvariant="sans-
serif-italic"],
mi[mathvariant="sans-serif-bold-italic"],mn[mathvariant="italic"],
mn[mathvariant="bold-italic"],
mn[mathvariant="sans-serif-italic"], mn[mathvariant="sans-serif-bold-
italic"],mo[mathvariant="italic"],
mo[mathvariant="bold-italic"],mo[mathvariant="sans-serif-italic"],
mo[mathvariant="sans-serif-bold-italic"],
ms[mathvariant="italic"], ms[mathvariant="bold-italic"],ms[mathvariant="sans-serif-
italic"],
ms[mathvariant="sans-serif-bold-italic"],mtext[mathvariant="italic"],
mtext[mathvariant="bold-italic"],
mtext[mathvariant="sans-serif-italic"], mtext[mathvariant="sans-serif-bold-italic"]
    {font-style:italic;}
mi[mathvariant="normal"], mn[mathvariant="normal"], mo[mathvariant="normal"],
ms[mathvariant="normal"], mtext[mathvariant="normal"]
    {font-style:normal;}
ms:before, ms:after
    {content:"\0022"}
ms[lquote]:before
    {content:attr(lquote)}
ms[rquote]:after
    {content:attr(rquote)}
mphantom
    {visibility:hidden}
merror
    {outline:solid thin red}
merror:before
    {content:"Error: "}
mrow
    {white-space:nowrap;}
math[display='block']
    {display:block;
    margin:1em 0 1em 3em;}
mstack, mlongdiv
    {display:inline-table;
    font-family:monospace;}
mstack
    {text-align:right;

```

```
border-collapse:collapse;}
mstack[align='top'], mlongdiv[align='top']
    {vertical-align:top;}
mstack[align='bottom'], mlongdiv[align='bottom']
    {vertical-align:bottom;}
mstack[align='center'], mlongdiv[align='center']
    {vertical-align:middle;}
msline
    {display:block;
    border-bottom:solid thin;}
mstack > *, mlongdiv > mn:first-child
    {display:table-row;}
mlongdiv > *:first-child + *
    {display:table-cell;}
mlongdiv > *:first-child + * + *
    {border-top:solid thin;}
mlongdiv > *:first-child:before
    {display:table-cell;
    content:'\a0';}
mlongdiv > *:first-child + *:after
    {content:'\a0';}
mlongdiv > *
    {display:block;}
mscarry
    {display:none;}
maction > * + *
    {display:none;}
maction[actiontype="tooltip"]:focus > * + *,
maction[actiontype="tooltip"]:hover > * + *
    {position:fixed;
    display:block;
    top:0;
    left:0;
    background-color:InfoBackground;
    color:InfoText;
    padding:0.5ex;
    border:solid 1px;}
annotation, annotation-xml
    {display:none;}
```

## Código Python

### Matlab.py

```
#!/usr/bin/python
from pylab import arange
from scipy.signal import lti
from scipy.signal import lsim
from py2xml import polyline2xml
from mod_python import apache
from numpy.ma.core import max
def index(req):
    xmin = 0;
    xmax = 100;
    system = lti([float(req.form['k']),[float(req.form['tau']),1])
    time_array = arange(xmin,xmax,1)
    input_array = time_array-time_array+1
    [time_array,output_array,state_array] = lsim(system,input_array,time_array)
    xml =
    polyline2xml(time_array,output_array,int(req.form['width'])/xmax,int(req.form['height
    '])/10)
    #req.form['w']

    return xml;
```

### py2xml.py

```
def array2xml(array):

    xml_data_list = []
    xml_data_list.append("<main>")
    for i in range(len(array)):
        xml_data_list.append("<d%s>%s</d%s>" % (i, array[i],i))
    xml_data_list.append("</main>")
    xml = ".join(xml_data_list)
    return xml;

def polyline2xml(x_array,y_array,xscale,yscale):

    xml_data_list = []
    xml_data_list.append("<main>")
    xml_data_list.append("<data>")
    for i in range(len(x_array)):
        xml_data_list.append("%d,%.2f " % (x_array[i],(y_array[i])))
    xml_data_list.append("</data>")
    xml_data_list.append("<polyline>")
    for i in range(len(x_array)):
        xml_data_list.append("%d,%.2f " %
(x_array[i]*xscale,(y_array[i])*yscale))
    xml_data_list.append("</polyline>")
    xml_data_list.append("</main>")
    xml = ".join(xml_data_list)
    return xml;
```