

DESARROLLO Y CONTROL DE PROCESOS VIRTUALES MEDIANTE MATLAB WEB SERVER

M. Vallés, A. Valera, J.L. Díez,
Dpto. Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia
Camino de Vera 14, 46022 Valencia (Spain)
E-mail: mvalles@isa.upv.es, giuprog@isa.upv.es, jldiez@isa.upv.es

Resumen

Existen áreas de conocimiento donde, por la problemática que ésta trata, puede resultar complicado la impartición de forma no presencial de las asignaturas del área. Un ejemplo representativo de esto lo podemos encontrar en el área de Ingeniería de Sistemas y Automática, puesto que para la realización de las prácticas de laboratorio se necesitan paquetes informáticos específicos y procesos físicos, siendo en ambos casos un equipamiento tan caro que hace imposible que los alumnos dispongan de ellos de forma local.

Para evitar estos problemas el artículo presenta un nuevo concepto: el proceso virtual. Dicho proceso es un conjunto de rutinas que permiten simular de forma animada (mediante fichero formato Audio Video Interleave AVI) el comportamiento, de una forma mucho más realista, de cualquier proceso físico existente en el laboratorio. Esto permite que el alumno asimile mejor una serie de conceptos como la sobreoscilación, tiempo de establecimiento, errores en régimen permanente, etc. De esta forma los alumnos pueden realizar de una forma más intuitiva todas las actividades relacionadas con la implementación del control de procesos: primero puede analizar el sistema para obtener sus características más importantes. Una vez obtenidas puede simular, con los procesos virtuales, la respuesta del proceso a controlar. Después de validar el controlador, los alumnos podrán establecer el control del proceso real (que ha servido de modelo para la generación del proceso virtual) de forma remota.

Para la realización de los laboratorios virtuales basados en los procesos virtuales y de los laboratorios remotos se ha escogido la herramienta Matlab Web Server. La elección está motivada no sólo porque Matlab es el paquete de diseño asistido por computador de sistemas de control más utilizado en el área, sino también porque éste permite de una forma muy simple el acceso de forma remota a las funciones, macros y drivers necesarios para la realización de las prácticas de laboratorios.

Palabras Clave: tele-enseñanza, laboratorio virtual, laboratorio remoto.

1. INTRODUCCIÓN

En los últimos años se ha conseguido, gracias a una mayor incorporación de computadores con conexión a Internet, la incorporación de la tecnología de la información en el proceso de aprendizaje de la enseñanza en general [4] y de la de los sistemas de control automático en particular [3].

Gracias a la evolución tan vertiginosa que se está dando en el desarrollo de las páginas Web y de los navegadores, hoy día pueden darse diferentes niveles de integración de la tecnología a un curso, haciendo posibles tareas tan diversas como: búsqueda de información, comunicación alumno-profesor o entre alumnos, administración de cursos, distribución de material, empleo de multimedia en el aula, desarrollo de tutoriales de autoaprendizaje, creación de laboratorios virtuales (donde la animación y la simulación reemplazan experimentos físicos) y laboratorios remotos (que permiten el ajuste de parámetros y/o la ejecución de experimentos desde un lugar remoto) [6], lo que permite que si se hace de una forma adecuada, la integración de la tecnología de la información con los cursos tradicionales puede servir como un apoyo muy efectivo a la docencia [5].

En trabajos anteriores [7], [8], [9] se ha abordado el diseño de laboratorios virtuales para asignaturas del área de conocimiento de Ingeniería de Sistemas y Automática, trabajos que han permitido minimizar los problemas provocados por la necesidad de tener paquetes informáticos nada económicos para la realización de las prácticas de laboratorio. Sin embargo, se ha detectado que cuando se realiza la simulación del control de los sistemas de forma clásica los alumnos pueden perder el significado físico de cuestiones como si el sistema es subamortiguado o sobreamortiguado, el concepto de sobreoscilación máxima, el tiempo de establecimiento etc.

Para solucionar estos problemas se ha abordado el desarrollo de laboratorios remotos [2], en los que los alumnos tienen que establecer el control de procesos físicos de forma remota. Este control remoto intenta reducir los problemas asociados al precio que estos suelen tener y que resultan generalmente muy caros para tener un gran número de equipos. En estos casos se tiene que limitar el número de usuarios que se conectan simultáneamente al sistema físico, lo que hace que el tiempo de respuesta y espera suelen ser muy grandes.

En este trabajo se plantea el desarrollo de **procesos virtuales** para evitar estos problemas. Un proceso virtual es un conjunto de funciones y procedimientos Matlab que simulan, mediante ficheros *avi*, el comportamiento de procesos reales. Si las ecuaciones del proceso virtual se corresponden con las ecuaciones físicas del proceso real se obtendrá un comportamiento simulado prácticamente idéntico al comportamiento real. De esta forma, gracias a la simulación previa con el proceso virtual, se podrá establecer sin peligro el control remoto del proceso físico real.

En este artículo se va a presentar primero la herramienta Matlab Web Server, analizando como se puede configurar dicha herramienta para el desarrollo de laboratorios virtuales. Posteriormente se muestra la programación de procesos virtuales, presentando el proceso virtual de un motor de corriente continua desarrollado para en el laboratorio virtual de control de procesos del Dpto. de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia.

2. MATLAB WEB SERVER

Matlab Web Server es un conjunto de programas que permiten a los programadores de Matlab crear aplicaciones de Matlab accesibles desde la Web. Éste está diseñado para ejecutarse de forma continua en background como un servicio de Windows NT.

La configuración más sencilla consistiría en tener un navegador de Internet en los computadores remotos que acceden a las aplicaciones implementadas con Matlab, las cuales se ejecutarán en la máquina que actúa como servidor, como muestra la figura 1.

Para permitir la ejecución de aplicaciones de Matlab en remoto, Matlab Web Server utiliza los siguientes programas y ficheros [1]:

- *matlabserver*: Se encarga de las comunicaciones entre la aplicación Web y Matlab. Es un servidor TCP/IP multi-hilo que ejecuta los programas de Matlab (ficheros ".m") especificados en el documento HTML. En el fichero de configuración del matlabserver se puede

especificar información como el número de puerto de comunicación y el número máximo de simulaciones que se desean atender

- *matweb.exe*: programa que reside en el servidor HTML. Es un cliente TCP/IP de matlabserver que utiliza la Common Gateway Interface (CGI) para extraer los datos de los documentos HTML y transferirlos a matlabserver.
- *matweb.conf*: fichero de configuración que matweb necesita para conectarse con matlabserver.
- *matweb.m*: llama al fichero ".m" que se desea ejecutar en la aplicación Web

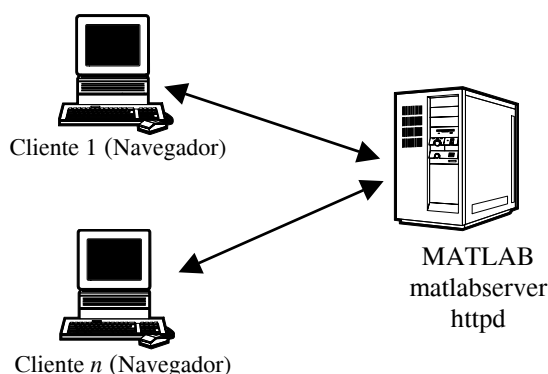


Figura 1: Configuración del laboratorio virtual

En cada aplicación de Matlab se debe especificar el directorio de trabajo de la aplicación y el nombre del host que está ejecutando matlabserver en *matweb.conf*. Opcionalmente, también se puede especificar el número de puerto (que debe coincidir con el asignado en el fichero de configuración de matlabserver), así como el tiempo (en segundos) para el vencimiento de tiempo (timing out).

Los equipos que hacen las peticiones al servidor de Matlab deben tener instalado un buscador Web. Para entornos de PC, los buscadores más utilizados son el Netscape Navigator (www.netscape.com) o el Microsoft Internet Explorer (www.microsoft.com). En el trabajo se ha optado por Internet Explorer versión 5.5.

Las aplicaciones correspondientes al Matlab Web Server serán generalmente una combinación de ficheros de Matlab ".m", código HTML y gráficos, de forma que la creación de este tipo de aplicaciones consistirá en la generación de un documento de entrada HTML para enviar los datos a Matlab, un fichero ".m" para procesar las entradas y calcular las salidas y un documento de salida HTML para mostrar los resultados de Matlab.

En los ficheros ".m" es donde se programarán las ecuaciones que regirán el comportamiento del proceso virtual, así como las instrucciones necesarias

para la generación de las animaciones en formato *avi* que se presentan en el apartado siguiente.

Los usuarios podrán observar y analizar el comportamiento del control del proceso virtual mediante un fichero de animación que contiene la respuesta del sistema. Dichos ficheros *avi* se mostrarán en los documentos HTML de salida.

La estructura de ficheros anteriormente mencionada se muestra en la figura siguiente.

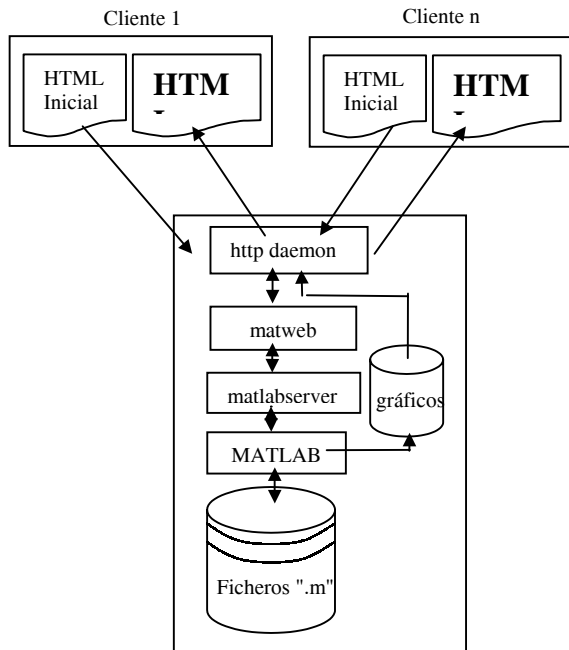


Figura 2: Estructura de ficheros del laboratorio virtual

3 GENERACIÓN DE PROCESOS VIRTUALES CON MATLAB

Matlab dispone, desde su versión 6, de medios (comandos MOVIE2AVI y AVI2MOVIE) para la conversión entre ficheros de película de Matlab (formato *.mov) y ficheros de vídeo con extensión *.avi. Los ficheros *avi* (Audio Video Interleave) son un caso especial de RIFF (Resource Interchange File Format) y, aunque definido por Microsoft, es el formato más común de datos para vídeo en PC con lo que se ha convertido en un estándar “de facto”.

Esto dota a Matlab de la posibilidad, tras la apropiada conversión, de realizar tratamiento de vídeos generados externamente con sus herramientas habituales. Igualmente, se hace así posible la visualización de animaciones generadas con Matlab en cualquier PC en el que se disponga de un reproductor estándar de vídeo, como los conocidos Windows Media Player o RealPlayer.

Adicionalmente, desde la versión 6 de Matlab, se nos permite generar directamente vídeos en formato *avi*, además de la forma habitual de captación de una imagen con FRAME2IM y su posterior inclusión en una película con MOVIE. El inicio de la generación de vídeos en formato *avi* de forma directa se realiza mediante el comando AVIFILE, que se usa con la siguiente sintaxis:

```
>> AVIOBJ=AVIFILE(nombredefichero)
```

Así se crea un objeto AVIFILE con nombre AVIOBJ. Los parámetros de creación del fichero *avi* mediante la función AVIFILE son varios, entre los que cabe destacar:

- FPS. Permite elegir los frames (“fotos”) por segundo que se desean captar de la figura de Matlab (generada, por ejemplo, con el comando PLOT) y que se desea “filmarse”.
- COMPRESSION. Indica el tipo de compresor de vídeo que se desea usar (existen 6 opciones distintas).
- QUALITY. Es un número entre 0 y 100 que define la calidad con que se desea captar la imagen (100 indica calidad máxima).

Una vez abierto el fichero, para generar la película, el siguiente paso es emplear el comando GETFRAME para hacer la “foto” de una figura deseada y que previamente se ha dibujado:

```
>> Comando_de_dibujo
>> FRAME=GETFRAME
```

Esta foto se debe incluir a continuación dentro del fichero de película, pasando a ser uno de los fotogramas que la van a componer. La forma de hacerlo es mediante el comando ADDFRAME, cuya sintaxis es:

```
>> AVIOBJ=ADDFRAME(AVIOBJ,FRAME)
```

Evidentemente, tanto el comando de dibujo como los comandos GETFRAME y ADDFRAME se suelen usar incluidos en un bucle FOR, de forma que se capten diferentes figuras cuya superposición generará el efecto de una película.

Un ejemplo clarificador e interesante puede ser el que se muestra a continuación, el cual genera el dibujo de una circunferencia que va creciendo desde un punto a un radio predefinido además de cambiar de color, mediante el empleo de todos los comandos anteriores y junto a los conocidos comandos de dibujo:

```
fig=figure;
mov=avifile('ejemplo.avi');
x = -pi:1:pi;
radius = [0:length(x)];
```

```

for i=1:length(x),
    if(mod(i,2)~=0),
        h=plot(sin(x)*radius(i),cos(x)*radius(i));
        axis([-80 80 -80 80]);
    else
        h=plot(sin(x)*radius(i),cos(x)*radius(i),'r');
        axis([-80 80 -80 80]);
    end;
    F = getframe(fig);
    mov = addframe(mov,F);
end;
mov = close(mov);

```

Como se puede observar, una vez se ha finalizado con la generación del vídeo (se han incluido todos las fotos deseadas), se debe cerrar el fichero abierto con:

```
>> AVIOBJ=CLOSE(AVIOBJ).
```

Con el uso de las herramientas que Matlab ofrece y que se han presentado en forma de comandos a lo largo de este apartado se puede conseguir, con su combinación con el Toolbox Matlab Web Server comentado en el apartado anterior, que cualquier animación generada con Matlab sea vista a través de Internet en una página Web. De hecho, es incluso posible que sea el usuario remoto, mediante una página web, el que genere la animación en función de unos parámetros que se le soliciten y que posteriormente observe el resultado en forma de fichero de vídeo.

4. CONTROL DE PROCESOS VIRTUALES

Como se ha comentado anteriormente, disponer de laboratorios remotos para el control de procesos permite que los alumnos puedan realizar prácticas de control sin que estos dispongan de herramientas software que pueden ser costosas ni de procesos que, evidentemente, el alumno es difícil que pueda tener acceso a ellos fuera del laboratorio. Esto, sin embargo, presenta un problema importante debido a que, en un momento determinado, únicamente un alumno podría estar trabajando con el proceso de manera remota (en el caso de que el proceso sea único, y en cualquier caso, limitado al número de procesos disponibles).

Una alternativa sería trabajar, usando un laboratorio virtual, con un programa de simulación que permitiera obtener la respuesta simulada del comportamiento del proceso. Sin embargo, el problema que puede presentar esta solución es el de que el resultado pueda ser demasiado abstracto para el alumno y sea difícil relacionar los resultados devueltos con el funcionamiento real del proceso.

El desarrollo de procesos virtuales, que permita disponer del número de sistemas que se necesite,

unido a un acceso a ellos a través de Internet sería una mejor solución al problema. En este apartado se va a presentar el desarrollo del control de dichos procesos virtuales. De esta forma se ha generado una práctica en la que el alumno puede evaluar el funcionamiento de los controladores diseñados para un proceso virtual.

El primer paso para el desarrollo de esta práctica consiste en la implementación del proceso virtual a controlar. En esta práctica se pretende controlar la posición del eje de un motor virtual de corriente continua. El eje del motor describe una trayectoria circular de forma que, dependiendo de la acción de control que se le proporcione, puede alcanzar cualquier ángulo entre 0 y 2π .

De esta forma, para la creación del proceso virtual, se ha programado la generación un fichero *avi* que muestre la evolución del cursor del motor, consistente en una rueda graduada sobre la que se desplaza un puntero (en rojo) que indica la posición del motor en cada momento imitando el comportamiento del motor real.

La evolución vendrá dada por la ecuación en diferencias de la función de transferencia discreta del sistema. Además, para que se interprete mejor, en el proceso virtual se ha colocado la fotografía del motor de corriente continua real que se ha utilizado como modelo. En concreto se trata del Servomotor SFT 154 de entrenador 33-100, de Feedback.

El proceso virtual se complementa con dos gráficas animadas más que muestran la evolución de la posición el motor y de la acción de control. Estas dos señales evolucionan sincronamente con el cursor del motor. Con esto los alumnos pueden analizar de una forma directa y simple si la respuesta del sistema es amortiguada o subamortiguada, el significado de la sobreoscilación máxima del sistema, etc. Además, también se pueden analizar como afecta la dinámica de control o el tipo de controlador a los valores de la acción de control.

En la figura 3 se muestra el aspecto que tiene el motor virtual desarrollado con la subdivisión de la zona de dibujo de las distintas señales.

Una vez disponible el proceso virtual, se pueden elaborar las distintas actividades que conformarán la o las prácticas del laboratorio virtual. Para ello primero se tiene que diseñar una página web de entrada que permita al alumno introducir los datos necesarios para la realización de la práctica. En la figura 4 se muestra la página programada para la práctica propuesta. En ella se proporciona al alumno una breve descripción del objetivo de la práctica, así como del significado de los campos en los que ha de introducir los datos y los resultados que debe obtener.

En este caso el alumno debe introducir la referencia que se pretende seguir, los coeficientes correspondientes al numerador del controlador discreto diseñado por el alumno y los coeficientes del denominador.

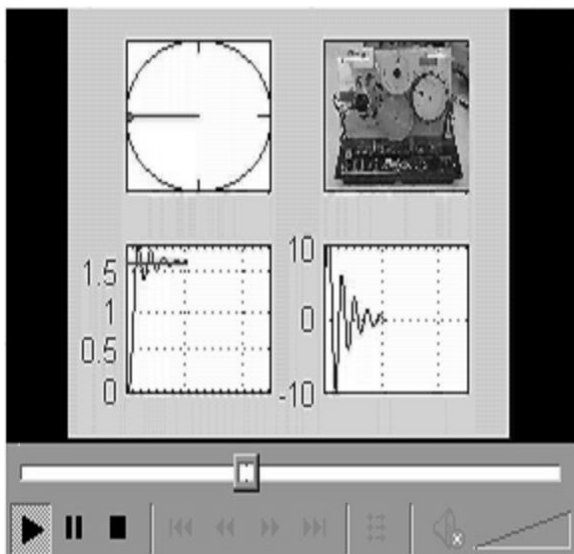


Figura 3. Detalle del resultado de la práctica.

Figura 4. Página de introducción del controlador

El paso siguiente para la elaboración de la práctica es generar una función que recoja la información que introducirá el alumno en la página Web de entrada y se la pase a Matlab. Para ello se hace uso de la herramienta Matlab Web Server porque simplifica mucho esta operación. En los trabajos [7], [9] se describe en detalle cómo se lleva esto a cabo.

A partir de los parámetros suministrados por el usuario, Matlab puede ejecutar la función de control como se hace de forma clásica. En esta práctica se simula el funcionamiento de un bucle cerrado de control de la posición del motor virtual, proporcionando como se ha dicho en un fichero

formato *avi* la evolución del cursor del eje del motor, la posición de éste y la acción de control generada por el regulador.

Este fichero *avi* es el parámetro que devuelve la función de Matlab a la página web que mostrará los resultados de la práctica y cuyo aspecto se puede ver en la figura siguiente:

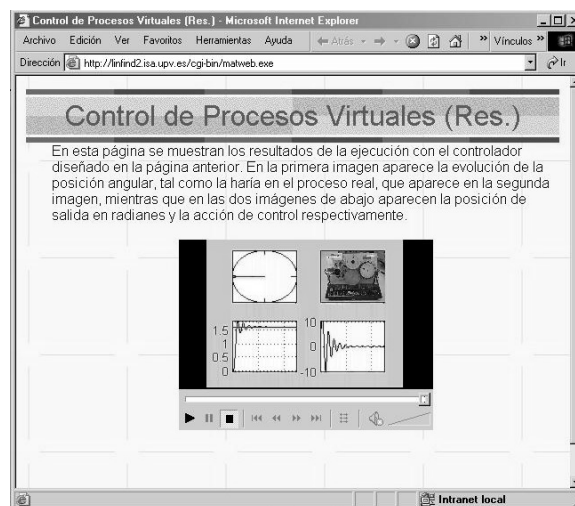


Figura 5. Página de resultados

Tal como se puede apreciar en la figura, en el fichero *avi* que recoge el resultado de la práctica se muestra en la esquina superior izquierda la posición final que ha alcanzado el cursor del motor (en este caso sobre $\pi/2$, ya que la posición 0 se ha hecho que coincida con la marca de segmento superior, tal como ocurre en el motor real), así como la fotografía del motor real. En la parte inferior se presenta la referencia y la posición del proceso virtual así como la evolución de la acción de control necesaria para llevarlo a la posición de referencia.

Con esta página, el alumno podrá ejecutar tantas veces como quiera el fichero *avi*, lo que le permitirá de una forma rápida e intuitiva analizar el comportamiento del controlador.

5. CONCLUSIONES

En el artículo se ha presentado como generar ficheros de animaciones con formato *avi* mediante Matlab. Programando adecuadamente el sistema e incorporando las ecuaciones físicas que rigen el comportamiento del sistema se pueden generar lo que se ha definido como procesos virtuales.

La utilización de estos procesos virtuales permite ampliar y complementar laboratorios virtuales para la realización de prácticas de Ingeniería de Sistemas y Automática.

Para mostrar la potencia y utilidad del trabajo propuesto se ha desarrollado una práctica para

establecer el control de posición de un motor virtual de corriente continua. Este proceso virtual proporciona el cursor del motor, la posición y la acción de control de forma animada, lo que hace más intuitivo y simple analizar el comportamiento de la respuesta del control del proceso virtual.

Agradecimientos

Los autores desean expresar su agradecimiento al Vicerrectorado de Coordinación Académica y Alumnado y al Instituto de Ciencias de la Educación de la Universidad Politécnica de Valencia por su apoyo de este trabajo, bajo el proyecto de innovación docente PID-12081C.

Referencias

- [1] (1999) Matlab Web Server user's guide. The MathWorks, Inc.
- [2] Díez, J.L., Vallés, M., Valera, A., Navarro, J.L., (2001) "Remote Industrial Process Control with Matlab Web Server", *IFAC-Internet Based Control Education IBCE2001*. Universidad Nacional de Educación a Distancia (Madrid), Diciembre (2001).
- [3] Olivares, M., Díez, J.L., (2000) "Estado Actual de la Docencia ISA en Internet", *I Jornadas de Trabajo de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática, EIWISA 2000*. Universidad Politécnica de Valencia, Mayo, ISBN 84-7721-963-X.
- [4] Paproch, Kenneth., (1998). Distance Learning: The Ultimate Guide London. Sage Publications.
- [5] Poindexter S.E., B.S. Heck, (1999) "Using the Web in your Courses: What can you do? What should you do?", *IEEE Control Systems Magazine*, Vol. 19,
- [6] Valera, Á., Vallés, M., Tornero, J., (2000) "LABCONROB: Laboratorio Remoto de Control en Tiempo Real de Sistemas Robotizados", *XXI Jornadas de Automática* Universidad de Sevilla, Septiembre, 2000. ISBN 84-699-3163-6
- [7] Valera, A., Vallés, M., Díez J.L., (2001) "Generación de un Laboratorio Virtual para Prácticas de Control con Matlab Web Server", *II Jornadas de Trabajo de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática, EIWISA 2001*. Universidad Nacional de Educación a Distancia (Madrid), Abril (2001).
- [8] Vallés, M., Valera, Á., Díez J.L., Grau, V., (2001), "Uso de la Herramienta Matlab Web Server para la Realización de Prácticas de Laboratorio Vía Internet", *Intermedia'01*. Universidad Politécnica de Valencia, Mayo (2001). ISBN 84-9705-049-5
- [9] Vallés, M., Díez, J.L., Valera, A., Navarro J.L., (2001) "Setting up a Virtual Matlab Control Laboratory", *IFAC-Workshop EDCOM2001: Education in Automatic Control*. Hohannes Kepler University Linz (Austria), Septiembre (2001)