

JMat – Herramienta remota de cálculo y multiusuario para el aprendizaje basado en problemas usando Matlab

JMat – Multiuser and remote calculation software tool for the problem based learning using Matlab

Bladimir Bacca Cortes^{1*}, *Eduardo Caicedo*¹, *José Ramírez*²

¹Grupo de Investigación en Percepción y Sistemas Inteligentes, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Cra. 91 N.º 28-23, Cali, Valle, Colombia

²Grupo de Investigación en Control Industrial, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Cra. 91 N.º 28-23, Cali, Valle, Colombia

(Recibido el 11 de mayo de 2010. Aceptado el 10 de marzo de 2011)

Resumen

JMat es una herramienta de cálculo basada en JAVA y EJS (Easy Java Simulations), con un esquema cliente / servidor, soporte multi-usuario y acceso remoto a Matlab. La aplicación está orientada a brindar a los usuarios una interacción con Matlab usando tres interfaces: Consola de Comandos, donde se invocan remotamente comandos de texto compatibles con Matlab. Espacio de Trabajo y Graficación, donde se mantiene un registro automático de las variables de usuario y se grafican individualmente. Funciones de usuario y Transferencia de Archivos, donde el usuario crea sus funciones, envía y recibe datos hacia y desde el servidor. JMat requiere un acceso a Internet, un servidor remoto donde esté instalado Matlab y un cliente (Navegador WEB o aplicación). No se requiere Matlab en el cliente. JMat está siendo usada actualmente en la Universidad del Valle en los cursos de Control Automático de Procesos, Control Inteligente, Redes Neuronales Artificiales, Procesamiento de Señales y Tratamiento Digital de Imágenes como herramienta para el aprendizaje basado en problemas empleando la plataforma de eLearning de la Universidad del Valle.

----- *Palabras clave:* JAVA, EJS, cliente / servidor, Matlab, remoto, multiusuario

* Autor de correspondencia: teléfono: + 57 + 2 + 332 79 27, fax: + 57 + 4 + 339 21 40, correo electrónico: evbacca@univalle.edu.co (B. Bacca)

Abstract

JMat is a software calculation tool developed on JAVA and EJS (Easy Java Simulations), client/server-based, multiuser support and remote access to Matlab. The main goal of the calculation software tool is to offer to remote users an interaction with Matlab using its three interfaces: Command Console, where users executes Matlab text commands; Work Space and Plotting, where automatically the remote user variables are stored and plotted individually; Functions and File Transfer, where users functions can be created, and data can be send or receive to and from a server. JMat requires an Internet access, a PC server with Matlab installed, and a client (WEB browser or stand-alone application). It is not necessary a Matlab on client side. Actually, JMat is being used at Universidad of Valle in courses like Automatic Process Control, Intelligent Control, Artificial Neural Networks, Signal Processing and Digital Image Processing as a tool for problem-based learning through the University of Valle's eLearning platform.

----- *Keywords:* JAVA, EJS, cliente / servidor, Matlab, remoto, multiusuario

Introducción

Durante los últimos años, la educación en ingeniería ha experimentado múltiples cambios debido al acelerado desarrollo de las tecnologías de la información y comunicación, a la globalización y a la necesidad social de ampliar la cobertura. En estos nuevos mercados abiertos se requiere de profesionales aptos para trabajar en equipos multidisciplinarios, con destrezas comunicativas y cooperativas, donde cada uno aporta al grupo su especialidad en la resolución de problemas prácticos. Una metodología particularmente apropiada para formar este nuevo profesional es el aprendizaje basado en problemas o PBL [1], en el cual a partir de problemas reales, el estudiante requiere buscar el conocimiento por sí mismo. La metodología

PBL requiere el acceso a la información, análisis y herramientas de experimentación de una manera más flexible que otras metodologías de aprendizaje tradicionales, por tanto se requieren herramientas tecnológicas apropiadas para soportarla. Estas tecnologías han sido orientadas de diferentes maneras: interfaces WEB para visualizar contenidos teóricos, laboratorios virtuales y experimentación remota. Sin embargo, estas tecnologías fuerzan al estudiante a cambiar frecuentemente de plataforma e interfaces de usuario, afectando el proceso de aprendizaje. Actualmente existen varias Universidades que han adoptado estrategias en el marco de las metodologías de aprendizaje activo [1-2]. La tabla 1 muestra varios tipos de herramientas para experimentación, simulación y aprendizaje no activo.

Tabla 1 Herramientas tecnológicas de aprendizaje

<i>Ref.</i>	<i>Software usado</i>	<i>Área de conocimiento</i>	<i>Objetivo</i>
[3]	PHP – HTML	Procesamiento de señales	Aprendizaje no activo
[4]	Labview, Visual C++	Control	Experimentación
[5-6]	C++	Inteligencia computacional	Simulación
[7]	Visual JAVA, Matlab	Control	Experimentación
[8]	Delphi, Matlab	Control	Experimentación
[9-10]	JAVA, Matlab	Control	Experimentación

Un importante paso en ingeniería es el modelado matemático de sistemas, para lo cual la educación en ingeniería requiere fuertes bases matemáticas y herramientas computacionales. El trabajo que se presenta responde a estas nuevas necesidades del aprendizaje en ingeniería, mostrando una herramienta de cálculo desarrollada en JAVA, usando EJS [11], empleando a Matlab como herramienta de procesamiento, con soporte multi-usuario a un solo servidor donde se ha instalado Matlab y ofreciendo tres interfaces de usuario: Consola de comandos, Espacio de trabajo y graficación, y Funciones de usuario con transferencia de datos; las cuales se utilizan como soporte para el aprendizaje basado en problemas en cursos de Control Automático, Procesamiento de Señales e Imágenes, Control Inteligente y Redes Neuronales Artificiales [12]. Gracias a que está enteramente desarrollada con software de libre distribución es una solución de bajo costo para desarrollos en equipo, incentivando el aprendizaje y desarrollo de conocimiento en espacios colaborativos; además, a diferencia de las herramientas descritas anteriormente está orientada al proceso del modelado matemático de sistemas en lugar del control o acceso remoto a un proceso complejo.

Este artículo se estructura así: primero se realiza una descripción de la herramienta software y de sus partes; en la sección 3, el uso práctico de la herramienta; en la sección 4, una descripción de las experiencias adquiridas y finalmente las conclusiones.

Metodología

Aspectos generales de la herramienta

JMat es una herramienta de cálculo para brindar una interfaz de análisis matemático en el contexto del proyecto de investigación descrito en [13], el cual se enmarca en la metodología PBL. La figura 1 muestra la estructura general de la herramienta, cada cliente posee una interfaz de usuario con un módulo de comunicación, el cual a través de Internet se conecta a un PC servidor donde Matlab está instalado y ejecutando un

servidor de peticiones multi-hilos. El esquema de comunicaciones se basa en JAVA [14] y la interfaz con Matlab en [15].

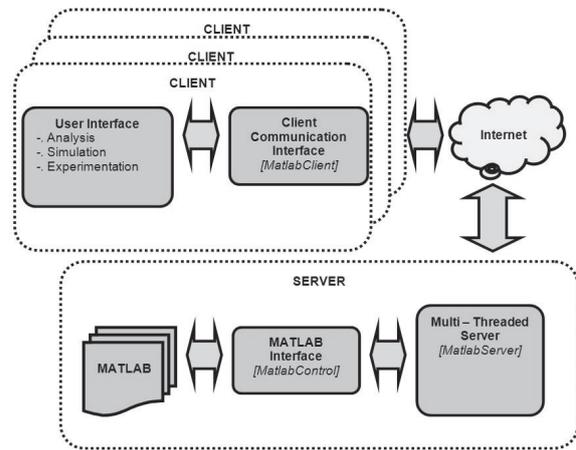


Figura 1 Esquema general de la herramienta

Consola de comandos

La figura 2 muestra la interfaz de usuario para los comandos de texto compatibles con Matlab y visualización de sus resultados; en la izquierda se listan los últimos comandos invocados. Adicionalmente se entregan herramientas para: guardar y limpiar la lista de comandos, el contenido de la consola, cambiar el nombre o número IP del servidor Matlab y la ayuda. Todo comando, variable o archivo cargado es privado, ningún otro usuario puede tener acceso; toda función de usuario o archivo de datos cargado en el servidor desde la interfaz descrita en la sección 2.4 puede ser usado en esta interfaz. Sin embargo, existen restricciones: ningún comando de Matlab con interfaz gráfica tiene efecto, no se permiten comandos de Matlab relacionados con el *Workspace* o sistema de archivos. Todo nombre de variable que se digite en la consola de comandos debe tener el prefijo '\$', lo cual es usado para identificar a cada usuario y todas sus variables, sino ningún cambio es revelado en el espacio de trabajo.

Espacio de trabajo y graficación

La figura 3 muestra la interfaz para el Espacio de Trabajo y Graficación. La parte superior

contiene herramientas para: guardar, actualizar y restaurar una sesión de trabajo con todas las variables de usuario; debajo, se encuentra la herramienta de graficación de datos que toma el nombre de la variable existente en el espacio de trabajo, la grafica y si es posible modifica las propiedades del gráfico hasta en sus tres ejes; la mayor área está dedicada a mostrar las variables, su nombre, valor y tipo. A medida que el usuario invoca comandos en la Consola, las variables afectadas se actualizan en esta interfaz. El espacio de trabajo soporta los siguientes tipos de datos: cadenas de caracteres, enteros cortos y largos, flotantes y bytes de hasta 3 dimensiones. Ya que la herramienta está diseñada en JAVA, existen inconvenientes de compatibilidad con tipos nativos de Matlab como: celdas, estructuras y manejadores de funciones, éstos deben ser tratados internamente en funciones de usuario.

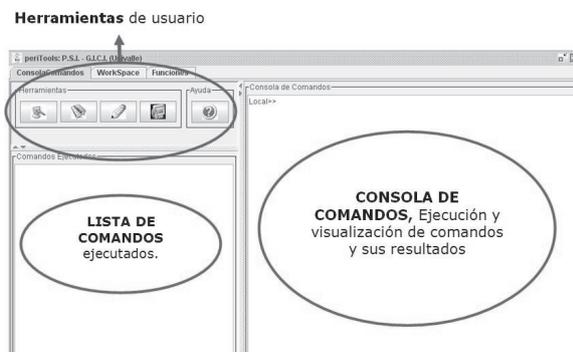


Figura 2 Interfaz de comandos por consola

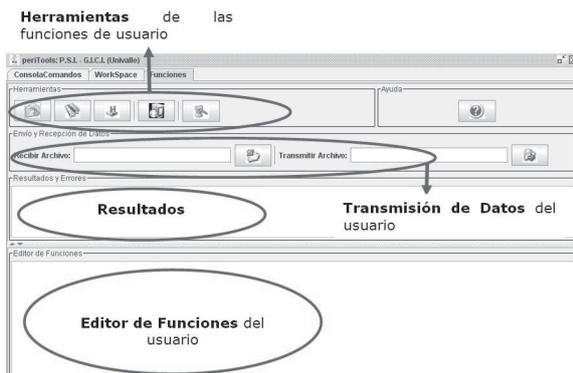


Figura 3 Interfaz del espacio de trabajo y graficación

Funciones de usuario y transferencia de datos

La figura 4 muestra la interfaz para la creación y envío de funciones de usuario hacia el servidor, posee un editor sencillo de texto y los comandos para guardar, abrir y enviar la función. Normalmente, en los procesos de modelado de sistemas o análisis matemático es necesario el trabajo con datos de campo, datos de inicialización, de entrenamiento o validación, para este fin el usuario puede enviar datos al servidor, o seleccionándolo de su sistema de archivos; también es posible recibir archivos; debajo se encuentra el editor, sobre el cual una función a la vez puede digitarse, almacenarse y enviarse al servidor. El prefijo '\$' no es necesario para funciones que se escriban en este editor.

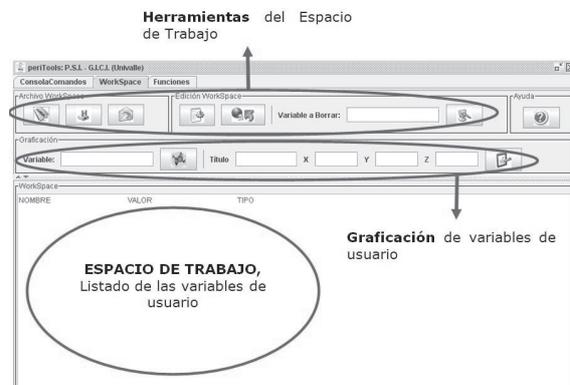


Figura 4 Interfaz de funciones y transferencia de datos

Empleo de la herramienta

La herramienta puede ser usada empleando una interfaz WEB como lo estipula el proyecto [13], o como una aplicación independiente sobre el ordenador del usuario, en cualquiera de ellas la apariencia de la interfaz es la misma. Es necesario que el servidor de Matlab remoto esté en línea, y una vez ejecutada la aplicación cliente el usuario encuentra una interfaz como la mostrada en la figura 2. Ya que la mejor forma de aprender es haciendo, la sección 3.1 muestra el uso de la herramienta a través de un problema clásico de control aplicado a la robótica [16]. Luego, la sección 3.2 muestra su

uso en las prácticas del curso de tratamiento digital de imágenes, y en la sección 3.3 para el curso de inteligencia computacional.

Control automático y robótica

La figura 5 muestra el dirigible empleado en [16], el cual lleva una cámara inalámbrica como único sensor de realimentación y cuyos objetivos de control son: la distancia en pixeles al objetivo o trayectoria que sigue, y la orientación del objetivo o trayectoria. Según [16-17] y [18] el sistema puede modelarse como lo indican las ecuaciones 1 a 4.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -M^{-1}D(t) \\ J(t) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} M^{-1}B \\ 0 \end{bmatrix} u \quad (1)$$

$$x_1 = v(t) = [v_x(t) \ v_y(t) \ v_z(t) \ w_x(t) \ w_y(t) \ w_z(t)] \quad (2)$$

$$x_2 = n(t) = [x(t) \ y(t) \ z(t) \ \phi(t) \ \theta(t) \ \psi(t)] \quad (3)$$

$$M = \begin{bmatrix} 0,49003 & 0 & 0 \\ 0 & 0,59995 & 0 \\ 0 & 0 & 0,10813 \end{bmatrix}$$

$$D = \begin{bmatrix} 0,052355 & 0 & 0 \\ 0 & 0,15888 & 0 \\ 0 & 0 & 0,075 \end{bmatrix} N/ms^{-1} \quad (4)$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0,09 & 0 \end{bmatrix}$$

Donde, x_1 es el vector de las velocidades lineales y angulares en cada eje, x_2 es el vector de las posiciones en el espacio y sus inclinaciones, M es la matriz de masas, D es la matriz de amortiguamiento, J es el Jacobiano del sistema y B define la posición de los motores.

Con un sistema del tipo $\dot{x} = Ax + Bu$ como se muestra en la ecuación 1, se usa la interfaz para digitar una función de usuario llamada *getBlimpModel()* y obtener el modelo del dirigible. La figura 6a muestra la función digitada y almacenada en disco. Una vez transferida

al servidor ésta se usa como un comando de Matlab. La figura 6b muestra los comandos y sus respuestas del servidor permitiendo: la obtención de la función de transferencia para el control de distancia, la definición del controlador, la obtención de los sistemas realimentados (con y sin control) y la respuesta al escalón unitario.

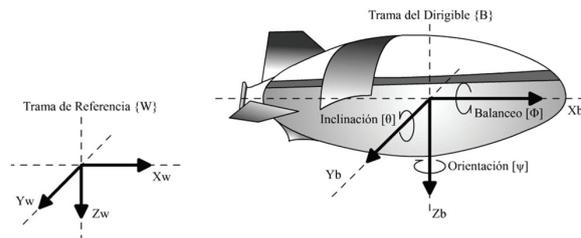


Figura 5 Dirigible y sistemas coordinados

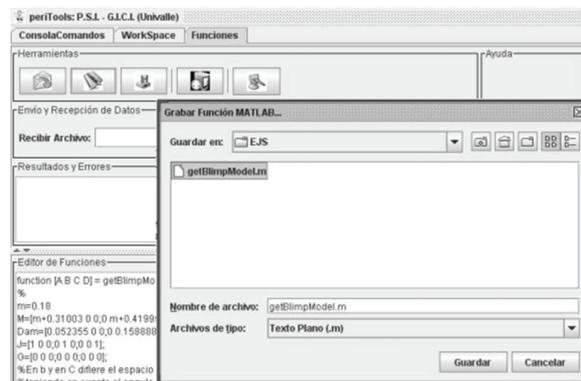


Figura 6 Herramienta y funciones de usuario

Usando las respuestas del servidor es posible una corrección de sintaxis. Todo comando de la consola de texto cambia el Espacio de Trabajo del usuario, ver figura 7.

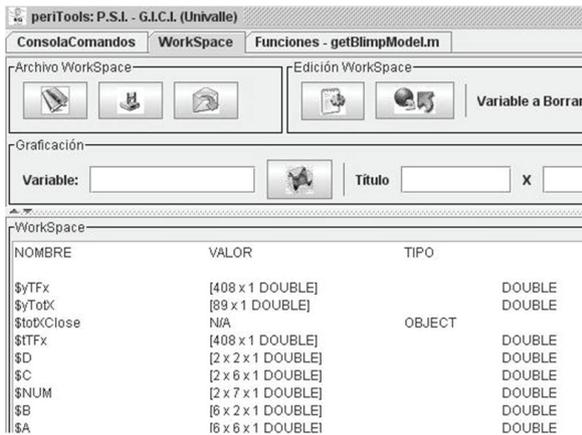


Figura 7 Espacio de trabajo

La figura 8 muestra la respuesta al escalón del sistema con y sin control, los cuales se basan en los datos calculados por Matlab, transmitidos a la aplicación y graficados localmente.

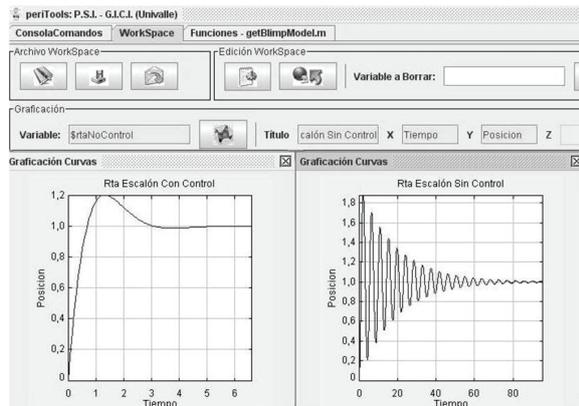


Figura 8 Sistema con y sin controlador

Inteligencia computacional

El curso de Redes Neuronales Artificiales de la Universidad del valle cubre temáticas como: el perceptrón, redes MLP, mapas auto-organizados, redes RBF y Bayesianas, con aplicaciones en

control y reconocimiento de patrones. Usando JMat es posible realizar una sesión remota para implementar sus prácticas. A continuación se muestra cómo entrenar una red MLP con algoritmo de aprendizaje Backpropagation y una red RBF para identificación de procesos.

Los datos de entrenamiento y validación son cargados usando la interfaz de transferencia de datos. Es importante recalcar que el comando *train* de MatLab debe ser usado así: $[red\ tr\ Y\ E\ pF\ aF]=train(red,x,y)$; donde, *red* es la nueva red neuronal entrenada con la visualización de la curva de error desactivada, y *tr* son los datos de la evolución del error de entrenamiento. La función *learnFunctionMLP()* implementa una red MLP con algoritmo Backpropagation cuyo objetivo es aprender una función determinada, ésta recibe el método de aprendizaje, número de iteraciones, número de datos de la función (*seno*, en este caso), número de neuronas de la capa oculta y error final; y entrega los datos de la función objetivo, la función aprendida y la evolución del error de aprendizaje. Al cargar esta función en el servidor, el usuario usa la consola de comandos (ver figura 6b) digitando:

$$[\$xyInit\ \$xyFin\ \$epochsPerf] = learnFunctionMLP(2, 1500, 10, 10, 0,001);$$

Donde, *\$xyInit* almacena la función objetivo, *\$xyFin* almacena la función aprendida y en *\$epochsPerf* la evolución del error. Sus gráficas se muestran en la figura 9.

Las redes neuronales son usadas para la identificación de sistemas complejos, aquí se mostrará el procedimiento sobre JMat para un sistema de primer orden, ecuación 7. Pero esto no significa que se pueda realizar con sistemas de órdenes superiores. La figura 10 muestra los datos entrada / salida del sistema que se desea identificar.

$$F(s) = \frac{1}{s + 0,5} \quad (7)$$

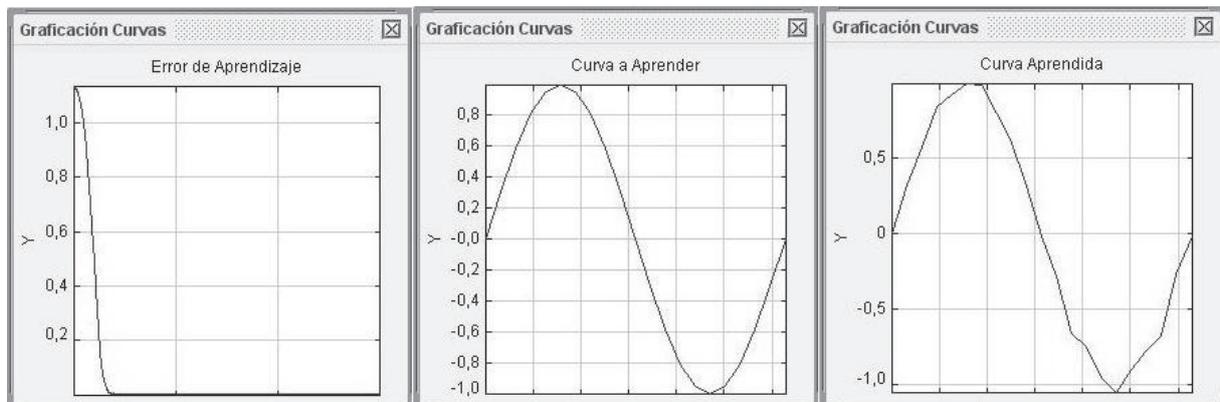


Figura 9 Curva objetivo, curva aprendida y error de aprendizaje

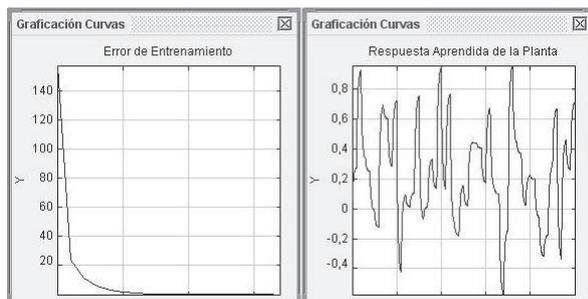


Figura 10 Referencia y salida del sistema a identificar

Para entrenar una red RBF se implementaron dos funciones, la primera carga los datos desde un archivo transmitido por el usuario y la segunda realiza el entrenamiento. La función de entrenamiento de la red RBF es *learnRBF()*, ésta recibe los datos de entrada y salida de la planta, y entrega: los datos de salida originales, los aprendidos y la evolución del error de entrenamiento. Luego, usando la consola de comandos:

```
[ $procIn $procOut ] = loadDatos();
[ $dInit $dFin $dError ] = learnRBF($procIn,
    $procOut, 0,01);
```

Donde, *\$dInit* y *\$dFin* almacenan los datos originales y aprendidos, y *\$dError* la evolución del error. Sus gráficas se muestran en la figura 11.

Procesamiento digital de imágenes

El curso de Tratamiento Digital de Imágenes posee temáticas como: la manipulación básica

de imágenes, mejoramiento de imágenes, filtros, y procesamiento en color. Usando JMat también es posible realizar una sesión remota para implementar sus prácticas. Por ejemplo, a continuación se describe el proceso para implementar filtros *high boost*, filtros en frecuencia (transformada de Fourier) y procesamiento en color en el espacio HSI.

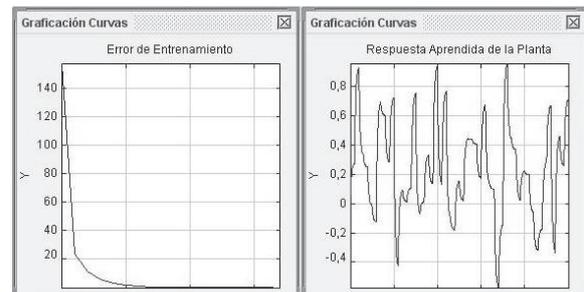


Figura 11 Salida del modelo neuronal RBF y error de entrenamiento

Usando JMat se cargan las funciones a emplear: *getImgFiltHighBoost()*, aplica un filtrado *high boost* a una imagen en nivel de grises; *getImgFrecFiltered()*, aplica un filtro en frecuencia previamente calculado sobre la transformada de Fourier de la imagen; y *getImgHSIProcess()*, realiza la equalización del histograma, aplica un filtro mediana o detecta bordes usando *Sobel* a una imagen en el espacio de color HSI. La función *getImgFiltHighBoost()* se invoca en la consola de comandos de la siguiente manera:

```
$imgIn = imread('moon.tif');
```

```
$imgHB = getImgFiltHighBoost('moon.tif', 5, 1,3);
```

Donde, se carga la imagen, luego se aplica el filtro con una ventana de 5x5 pixeles y una constante de filtrado de 1.3, los resultados se muestran en la figura 12. En esta figura se muestra el empleo de la función *getImgFrecFiltered()*, la cual se puede invocar así:

```
$imgIn = imread('pout.tif');
```

```
[$imR $imC] = size($imgIn);
```

```
$filt = getImgFiltro($imR, $imC, 'PB', 15, 1, 0);
```

```
[$imFrec $fftIn $fftOut] =  
getImgFrecFiltered($imgIn, $filt);
```

Donde, se lee la imagen, se captura su tamaño, se calcula un filtro paso bajo ideal de tamaño 15 pixeles y finalmente se realiza el filtrado en frecuencia, obteniendo la imagen filtrada y las transformadas de Fourier de la imagen de entrada y la imagen filtrada.



Figura 12 Filtrado tipo high boost y filtrado en frecuencia

El procesamiento en color también puede realizarse con JMat (en este artículo solo se observa en grises), la figura 13 muestra la aplicación del operador *Sobel* en la componente I de la imagen y usando *getImgHSIProcess()*. En la consola de comandos se digita:

```
$imgIn = imread('onion.png');
```

```
$imgOut = getImgHSIProcess  
( 'onion.png', 'Sobel');
```

La función *getImgHSIProcess()* convierte la imagen RGB a HSI, aplica el operador *Sobel* sobre la Intensidad y luego la convierte a RGB para su visualización en JMat.



Figura 13 Operador Sobel aplicado a una imagen en color

Resultados y discusión

JMat fue concebida en el marco del proyecto de investigación aprobado por COLCIENCIAS [13], cuyo objetivo fue la concepción de una plataforma de procesamiento distribuido con acceso remoto multiusuario y emulación de sistemas dinámicos para investigación y educación en ingeniería. El alcance incluía las herramientas de aprendizaje, la elaboración de toda una metodología basada en el aprendizaje basado en proyectos, y su respectiva validación usando encuestas a los estudiantes y profesores. La evaluación de herramientas como la presentada en este trabajo no solo incluye aspectos técnicos, sino, diversos aspectos relevantes con el aprendizaje basado en proyectos y competencias. Este tipo de evaluación es acogida actualmente por diversos autores en Ingeniería, tal es el caso de Restivo et. al. [19] donde se ha implementado un laboratorio virtual basado en LABVIEW para la enseñanza de mecánica de materiales, cuya evaluación incluye aspectos técnicos, de aprendizaje y comentarios abiertos. En [20] un laboratorio virtual para la simulación de un proceso químico de deposición de vapor es implementado, cuya evaluación incluye además de aspectos técnicos, aspectos cognitivos usando un modelo que determina que es lo que los estudiantes intentan aprender, para lo cual los autores usan encuestas que relacionan temas como el trabajo en equipo, la capacidad de aprender autónomamente y la medición de la efectividad de los medios de aprendizaje. En el contexto del proyecto [13] se han definido una serie de aspectos relevantes como: trabajo en equipo, aprendizaje autónomo, metodología de solución de problemas, expresión oral y escrita, y recursos físicos y tecnológicos; sin embargo no

todos están relacionados con las herramientas de aprendizaje diseñadas, los cuales son:

- Factor trabajo en equipo – 1. El curso me ha ayudado a adquirir capacidad para trabajar en grupo. 2. El curso me permitió desarrollar habilidades para mi desempeño profesional.
- Factor aprendizaje autónomo– 3. El curso me ha ayudado a adquirir capacidad para aprender de forma autónoma. 4. La metodología diseñada para el desarrollo del curso considera el empleo de tecnologías de información y comunicación.
- Factor Metodología en solución de problemas– 5. Capacidad para aprender de forma autónoma. 6. Capacidad para aplicar el conocimiento en la práctica. 7. Capacidad para resolver problemas de ingeniería. 8. Habilidad y actitud investigativa
- Factor Recursos físicos y tecnológicos– 9. Los medios y las instalaciones han sido adecuados. 10. Aprovecho eficazmente los recursos físicos y tecnológicos de los que dispone la Universidad

Las herramientas de aprendizaje diseñadas cubren las áreas del conocimiento especificadas en la sección 3, todas ellas tienen como base a JMat y durante 4 semestres académicos consecutivos se recogieron evaluaciones de los estudiantes en los cursos: Procesamiento Digital de Señales, Control Inteligente, Sistemas de Control Automático y Sistemas Lineales. El resultado de estas encuestas se muestra en las figuras 14, 15, 16 y 17. Donde la calificación promedio otorgada está en un rango entre 0 y 5, y en el eje de las abscisas se encuentra la pregunta relevante a las herramientas de aprendizaje. La confiabilidad de estos datos fue evaluada con el coeficiente *Alpha Cronbach* [23], el cual es calculado usando la ecuación 8.

$$\alpha = \left(\frac{k}{k-1} \right) \left(1 - \frac{\sum_i \sigma_i^2}{\sigma^2} \right) \quad (8)$$

Donde, k es el número de ítems, σ_i^2 es la varianza de cada ítem, y σ^2 es la varianza total de la prueba. Para este trabajo, los coeficientes a lo largo de los

cuatro semestres fueron: 0.94, 0.91, 0.95 y 0.91 correspondientemente.

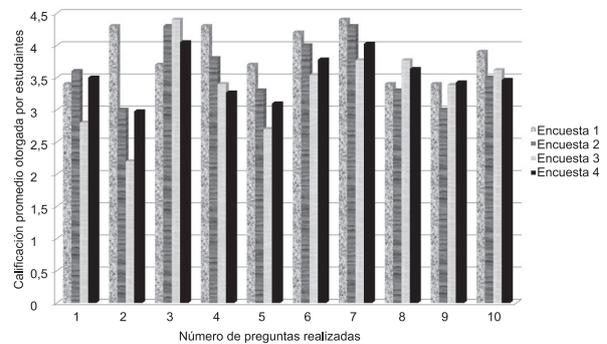


Figura 14 Resultados de las encuestas del curso de procesamiento digital de señales

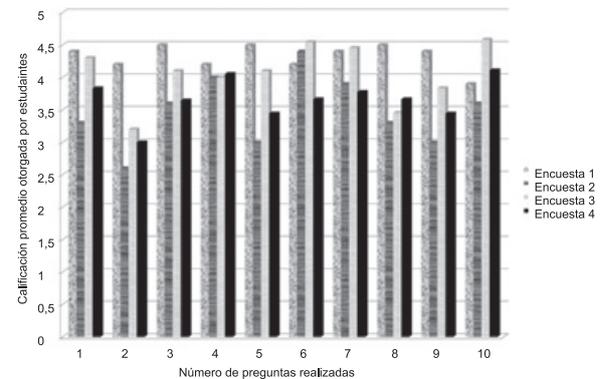


Figura 15 Resultados de las encuestas del curso de control inteligente

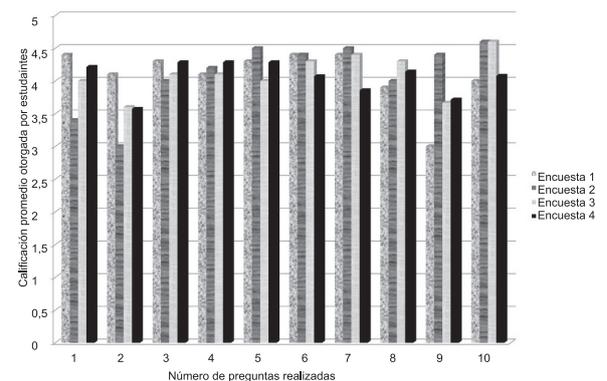


Figura 16 Resultados de las encuestas del curso de sistemas automáticos de control

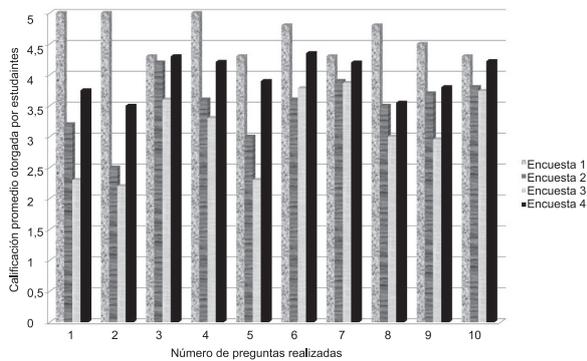


Figura 17 Resultados de las encuestas del curso de sistemas lineales

Los resultados demuestran la buena acogida de la metodología y de las herramientas de aprendizaje. Algunos resultados en ciertos semestres no son del todo satisfactorios, por lo cual se recopilaban también las sugerencias y posibles mejoras a las herramientas:

1. *Un Espacio de Trabajo remoto compatible con estructuras, celdas y objetos de Matlab.* La solución es pre-cargar una serie de funciones y que el usuario pueda emplear.
2. *Graficación múltiple.* Se está trabajando en una visualización del Espacio de Trabajo del usuario más amigable, permitiendo la múltiple selección de variables a graficar.
3. *Edición múltiple de funciones y transferencia de archivos.* Se está trabajando en una interfaz que permita editar, enviar y/o recibir *amigablemente* varias funciones a la vez.
4. *Una mejor consola de comandos.* Se están complementando las funcionalidades del teclado para facilitar la digitación de comandos y la corrección de los nombres de variables.

El funcionamiento de JMat durante estos últimos 4 semestres ha sido interno al campus de la Universidad [21], debido a las políticas de seguridad de la Universidad del Valle. Pero, gracias a la red RENATA [22] se espera que otras Universidades tengan a esta herramienta.

Conclusiones

JMat es una herramienta de cálculo basada en JAVA, desarrollada en Easy Java Simulations, con soporte multi-usuario y acceso remoto a Matlab, la cual permite: la ejecución remota de comandos compatibles con Matlab usando una consola de comandos, la manipulación de las variables de usuario gracias a un Espacio de Trabajo, su graficación, la edición / carga de funciones de usuario en el servidor remoto y la transferencia bidireccional de archivos de datos, características que convierten a JMat en una excelente alternativa después de la discontinuación del Matlab WEB Server. Gracias al acceso remoto de JMat y teniendo en cuenta el actual auge de los laboratorios virtuales, JMat ofrece la oportunidad de desarrollar y trabajar en la solución de problemas multidisciplinarios con diversas entidades, sin que éstas se encuentren geográficamente en el mismo sitio. Lo cual tiene como gran ventaja un abaratamiento de los costos de desplazamiento, logísticos y de desarrollo. Desde el punto de vista técnico, una gran ventaja de JMat es la posibilidad de aprovechar la potencia de cálculo de un servidor remoto con el fin de ejecutar simulaciones y emulaciones de gran complejidad, que en un PC convencional tomaría mucho tiempo en completarse, de tal manera que el costo de los prototipos para solucionar problemas complejos en ingeniería se reduce. JMat es una aplicación que encaja bastante bien en las tendencias actuales de la enseñanza, investigación y desarrollo de proyectos en ingeniería, ya que ofrece herramientas de análisis y experimentación para un trabajo cooperativo, especializado y multidisciplinario. El trabajo futuro se concentrará en atender las sugerencias antes descritas y en el desarrollo de un módulo adicional que permita una interfaz tipo *Simulink*.

Agradecimientos

Los autores agradecen de manera especial a la Ingeniera Liliana Fernández M.Sc., estudiante del programa de Doctorado en Ingeniería de la Universidad del Valle por su valioso aporte en la concepción de las encuestas dentro del proyecto de investigación.

Referencias

1. L. Uden, C. Beaumont. "Technology and Problem-based Learning". *Information Science Publishing*. Vol. 2006. pp. 1-207.
2. E. Graaff, A. Kolmos. "Management of Change: Implementation of Problem-Based and Project-Based Learning in Engineering". *Sense Publishers*. Vol. 2007. pp. 113-215.
3. S. L. Toral, F. J. Barrero. "Implementation of a web-based educational tool for digital signal processing teaching using the technological acceptance model." *IEEE Transactions on Education*. Vol. 48. 2005. pp. 632-641.
4. H. Hassan, C. Dominguez, J. Martinez, A. Perles, J. Albaladejo. "Remote laboratory architecture for the validation of industrial control applications." *IEEE Transactions on Industrial Electronics*. Vol. 54. 2007. pp. 3094-3102.
5. Y. Liao, C. Sun. "An educational genetic algorithms learning tool." *IEEE transactions on Education*. Vol. 44. 2001. pp. 20
6. J. Jantzen. "Internet learning in control engineering: A fuzzy control course." *6th IFAC Symposium on Advances in Control Education. ACE 2003*. Oulu (Finland). pp. 27-35.
7. J. Sanchez, S. Dormido, R. Pastor, F. Morilla. "A Java/Matlab-based environment for remote control system laboratories: illustrated with an inverted pendulum." *IEEE Transactions on Education*. Vol. 47. 2004. pp. 321-329.
8. B. Doğan, H. Erdal. *System Control Through the Internet and a Remote Access Laboratory Implementation*. Ed. Springer. Berlin (Germany). 2007. pp. 532-541.
9. P. Bistak. "Matlab and Java based virtual and remote laboratories for control engineering". *17th Mediterranean Conf. on Control and Auto*. June 24-26. 2009. Thessaloniki (Greece). Vol. 1. 2009. pp. 1439-1444.
10. Y. Sheng, W. Wang, J. Wang, J. Chen. "A Virtual Laboratory Platform Based on Integration of Java and Matlab." *In Proc. of the 7th int. conf. on Advances in Web Based Learning*. Jinhua (China). Vol. 5145. 2008. pp. 285-295.
11. <http://www.um.es/fem/EjsWiki/>. Consultada el 12 de febrero de 2010.
12. J. Ramírez, E. Caicedo, H. Loaiza, E. Bacca, C. Pinedo. "Platform for Signals and Systems Internet-Based Education, International Technology." *Education and Development Conference*. Valencia (España). Vol. 1. 2008. pp. 558-564.
13. J. Ramírez, E. Caicedo, H. Loaiza, E. Bacca. *Plataforma de Procesamiento Distribuido y Emulación de Sistemas Dinámicos con Acceso Remoto Multiusuario para Educación e Investigación en Ingeniería*. COLCIENCIAS. Contrato N.º 234. 2005. pp. 9
14. <http://java.sun.com/javase/>. Consultada el 14 de febrero de 2010.
15. <http://www.cs.virginia.edu/~whitehouse/>. Consultada el 14 de febrero de 2010.
16. J. Sterling, G. Catuche, E. Bacca. "Seguimiento de Trayectorias y Objetivos en Tierra usando un Dirigible y Visión Artificial." *Rev. Fac. Ing. Univ. Antioquia*. Vol. 50. 2009. pp. 122-134
17. T. Fossen. *Guidance and Control of Ocean Vehicles*. Ed. John Wiley & Sons. Chichester (England) 2003. pp. 5-162.
18. S. Van der Zwaan. *Vision Based Station Keeping and Docking for Floating Robots*. Msc. Thesis, Lisboa-Portugal, Instituto Superior Técnico. Vol. 1. 2001. pp. 47-61.
19. M. Restivo, J. Mendes, A. Lopes, C. Silva, F. Chouzal. "A Remote Laboratory in Engineering Measurement," *Ind.Elec. IEEE Trans. On*. Vol. 56. 2009. pp. 4836-4843.
20. M. Koretsky, D. Amatore, C. Barnes, S. Kimura. "Enhancement of Student Learning in Experimental Design Using a Virtual Laboratory," *Edu. IEEE Trans*. Vol.51. 2008. pp. 76-85
21. <http://campusvirtual.univalle.edu.co/>. Consultada el 14 de febrero de 2010.
22. <http://www.renata.edu.co/>. Consultada el 14 de febrero de 2010.
23. R. Ledesma. "AlphaCI: un programa de cálculo de intervalos de confianza para el coeficiente alfa de Cronbach," *Psico-USF*. Vol. 9. 2004. pp. 31-37.