

# Modelado orientado a objetos de laboratorios virtuales con aplicación a la enseñanza del control de procesos químicos

Carla Martin, Alfonso Urquia y Sebastian Dormido

Departamento de Informática y Automática

E.T.S. de Ingeniería Informática, UNED

Juan del Rosal 16, 28040 Madrid

{carla, aurquia, sdormido}@dia.uned.es

## Resumen

El uso combinado de Modelica/Dymola con Ejs y Sysquake permite reducir el esfuerzo de modelado y programación de los laboratorios virtuales. Este enfoque permite sacar partido de la capacidad de Modelica para el modelado físico, de Dymola para la simulación de modelos DAE-híbridos, y de Ejs y Sysquake para la construcción de interfaces interactivas.

Para ponerlo en práctica, se han completado las dos siguientes tareas: (1) se ha propuesto una nueva metodología para el desarrollo de los modelos en Modelica, que permite adecuarlos a su empleo, junto con Ejs, en la construcción de simulaciones interactivas; y (2) se ha programado una interfaz entre Sysquake y el ejecutable generado por Dymola para el modelo escrito en Modelica.

Todo ello se ha aplicado al desarrollo de un conjunto de laboratorios virtuales para la enseñanza del control de procesos químicos. En este trabajo se muestran dos de ellos: un intercambiador de calor y un evaporador.

## 1. Introducción

Un *laboratorio virtual* es un entorno distribuido de herramientas de simulación y animación, cuyo propósito es realizar la simulación interactiva de un modelo matemático. Los laboratorios virtuales proporcionan un método flexible y amigable para definir los experimentos que se llevan a cabo sobre el modelo siendo por ello herramientas útiles para la enseñanza.

Típicamente, la definición de un laboratorio virtual incluye las dos partes siguientes: el *modelo* y la *vista*. La *vista* es la interfaz entre el usuario y el modelo. Su objetivo es proporcionar una representación visual del comportamiento dinámico del modelo y facilitar las acciones interactivas del usuario sobre aquel. Las propiedades gráficas de los elementos de la *vista* se enlazan a las variables del *modelo*, produciendo un flujo bidireccional de información entre la *vista* y el *modelo*.

En este contexto, cabe distinguir los dos tipos de interactividad siguientes:

- *Interactividad continua*. Se permite al usuario modificar el valor de las entradas, de los parámetros y de las variables de estado del modelo en cualquier instante durante la ejecución de la simulación. De este modo, el usuario percibe instantáneamente cómo estos cambios afectan a la dinámica del modelo.
- *Interactividad discontinua*. Se permite al usuario fijar el valor de los parámetros y del estado inicial del modelo previamente a la ejecución de la simulación. Durante la ejecución de la simulación, no se permite al usuario interactuar con el modelo. Una vez finalizada la simulación, sus resultados son mostrados y analizados, permitiéndose entonces al usuario interaccionar de nuevo con el modelo a fin de fijar un nuevo conjunto de valores para los parámetros y las condiciones iniciales.

Actualmente, existen herramientas software

cuyo propósito es facilitar la creación de laboratorios virtuales. Dos ejemplos son Easy Java Simulations [4] (abreviado: Ejs) y Sysquake (<http://www.calerga.com/>). Un punto fuerte de estas herramientas son sus facilidades para la programación de la *vista*. Un punto débil son las limitadas capacidades ofrecidas para la descripción del *modelo*. Éste debe formularse como un ODE explícito y describirse indicando su causalidad computacional, lo cual dificulta la tarea de modelado.

Por otra parte, los lenguajes de modelado orientados a objetos reducen considerablemente el coste de la realización del modelo, ya que facilitan la aplicación del paradigma de modelado físico y la reutilización de los modelos [1]. Los entornos que soportan estos lenguajes permiten la simulación de modelos DAE-híbridos. Sin embargo, estos lenguajes no encuentran aplicación en la programación de laboratorios virtuales puesto que no proporcionan capacidades interactivas.

### 1.1. Contribuciones de este trabajo

En este trabajo se ilustra, mediante dos casos de estudio, el uso combinado de Modelica ([www.modelica.org](http://www.modelica.org)), con el entorno Dymola [3], y de las herramientas Ejs y Sysquake.

Para la composición de los modelos físicos, se ha empleado la librería JARA [5], que ha sido adaptada para su aplicación al desarrollo de simulaciones interactivas. JARA es un conjunto de librerías, escritas en Modelica, que permiten modelar algunos principios físico-químicos fundamentales. La metodología, con carácter general, para la realización de este tipo de adaptación se propone en [6].

La *vista* de los laboratorios virtuales que soportan interactividad continua ha sido implementada usando Ejs, y la de los laboratorios con interactividad discontinua usando Sysquake. En las Secciones 2 y 3 se describe cómo se realiza la conexión entre estas herramientas, y en las Secciones 4 y 5 se muestran dos casos de estudio: un intercambiador de calor y de un evaporador. En cada caso se realizan dos laboratorios virtuales, uno con interactividad continua y otro con interactividad discontinua.

## 2. Interactividad continua

Ejs (<http://fem.um.es/Ejs/>) es una herramienta software gratuita, de código abierto, para la programación de laboratorios virtuales. Ejs incorpora una interfaz con Matlab/Simulink, la cual permite describir partes del modelo usando bloques de Simulink. Ejs sincroniza la simulación de estas partes, que realiza Simulink, con las demás partes del modelo, que se realiza usando los métodos numéricos proporcionados por Ejs.

Por otra parte, Dymola tiene la capacidad de encapsular dentro de un bloque Simulink el código C que genera para el modelo escrito en Modelica [3].

Las interfaces Ejs-Simulink y Dymola-Simulink permiten el uso combinado de Ejs y Modelica/Dymola. Los detalles sobre esta conexión pueden encontrarse en [6].

Para encapsular el modelo escrito en Modelica dentro de un bloque DymolaBlock, es necesario especificar la causalidad computacional de la interfaz del modelo de Modelica [3]. Asimismo, el bloque de Modelica debe ser escrito para permitir la comunicación con la vista del laboratorio virtual. A tal fin, el modelo necesita soportar cambios discontinuos en los valores de sus variables de estado, parámetros y variables de entrada que son el resultado de la acción del usuario. Todos los detalles pueden encontrarse en [6].

## 3. Interactividad discontinua

Se ha programado un conjunto de funciones para realizar las tareas de comunicación entre Sysquake y el fichero ejecutable generado por Dymola para el modelo en Modelica. Estas tareas incluyen: la inicialización del modelo, descripción del experimento, control de la simulación y escritura de resultados en el espacio de trabajo de Sysquake.

Empleando las funciones anteriores se puede controlar la simulación de un modelo escrito en Modelica y traducido con Dymola desde una aplicación de Sysquake. Dichas funciones han sido programadas en LME, que es el lenguaje propio de Sysquake. En [6] puede encontrarse

información adicional acerca del uso de estas funciones.

#### 4. Intercambiador de calor

Se ha realizado el modelo de un intercambiador de calor compuesto por dos tubos concéntricos. Por el tubo central circula una mezcla gaseosa de  $CO_2$  y  $SO_2$ , que es refrigerada mediante el agua que circula por el tubo externo. El intercambiador puede funcionar con ambos fluidos en contracorriente o bien circulando en el mismo sentido. Las transferencias de calor convectivas entre la mezcla gaseosa y la pared del tubo central, y entre ésta y el líquido, se calculan a partir de la correlación Dittus-Boelter [2]. Se considera la conducción térmica a lo largo de la pared del tubo central, mientras que se supone que la tubería externa está muy bien aislada del entorno.

El modelo físico del intercambiador de calor se ha compuesto usando JARA [5]. La metodología propuesta en [6] fue aplicada para hacer JARA útil para la simulación interactiva continua.

##### 4.1. Laboratorio virtual con Ejs

La vista del laboratorio virtual se muestra en la Figura 1a. La ventana principal (lado izquierdo) contiene: (1) un diagrama del intercambiador de calor; (2) botones para controlar la ejecución de la simulación (pause, reset y play); (3) deslizadores y campos numéricos para modificar las variables de entrada (flujo de líquido y gas, temperatura de líquido y gas, y fracción de  $CO_2$  y  $SO_2$  en el gas de la mezcla); y (4) selectores para mostrar y ocultar tres ventanas secundarias. El resto de la ventana contiene elementos que permiten cambiar interactivamente las longitudes y diámetros de las tuberías, cambiar la temperatura del medio dentro de cada volumen de control y mostrar en gráficos la evolución de las variables del modelo.

##### 4.2. Laboratorio virtual con Sysquake

En este laboratorio se pretende identificar y controlar el intercambiador de calor. El objeti-

vo es controlar la temperatura del gas, variando para ello el flujo de agua de la bomba de entrada. Para la identificación de la respuesta del sistema a una entrada escalón, se ha creado una aplicación cuya vista se muestra en la Figura 2a. El sistema se identifica mediante una función de transferencia (F.T.), cuya salida es la temperatura de salida del gas y cuya entrada es el flujo de agua. Esta aplicación permite al usuario:

1. Cambiar los parámetros y variables de entrada del intercambiador, el intervalo de comunicación y el tiempo de simulación.
2. Seleccionar el método de identificación, empleando para ello una F.T. de primer orden con retardo, de segundo orden con retardo o identificación no paramétrica.
3. Analizar la F.T. que se ha obtenido. Para ello se facilita la visualización del diagrama de Bode, de polos y ceros en el plano  $s$  y los márgenes de robustez.
4. Exportar la F.T. obtenida a otra aplicación en Sysquake.

Para la síntesis y análisis de controladores se ha creado otra aplicación en Sysquake, cuya vista se muestra en la Figura 2b. Mediante esta aplicación el usuario puede:

1. Importar la F.T. previamente identificada.
2. Seleccionar un controlador PID, una red de adelanto o de retardo de fase.
3. Modificar los parámetros del controlador PID. Especificar el error y margen de fase del sistema controlado con las redes de adelanto o de retardo de fase.
4. Visualizar la salida del controlador y del sistema controlado.
5. Analizar las características de la F.T. en lazo abierto mediante diagramas de Nyquist, Nichols y Bode.

#### 5. Evaporador

Se ha empleado el modelo matemático de un evaporador industrial descrito en [7]. La entrada de agua se encuentra en la parte baja

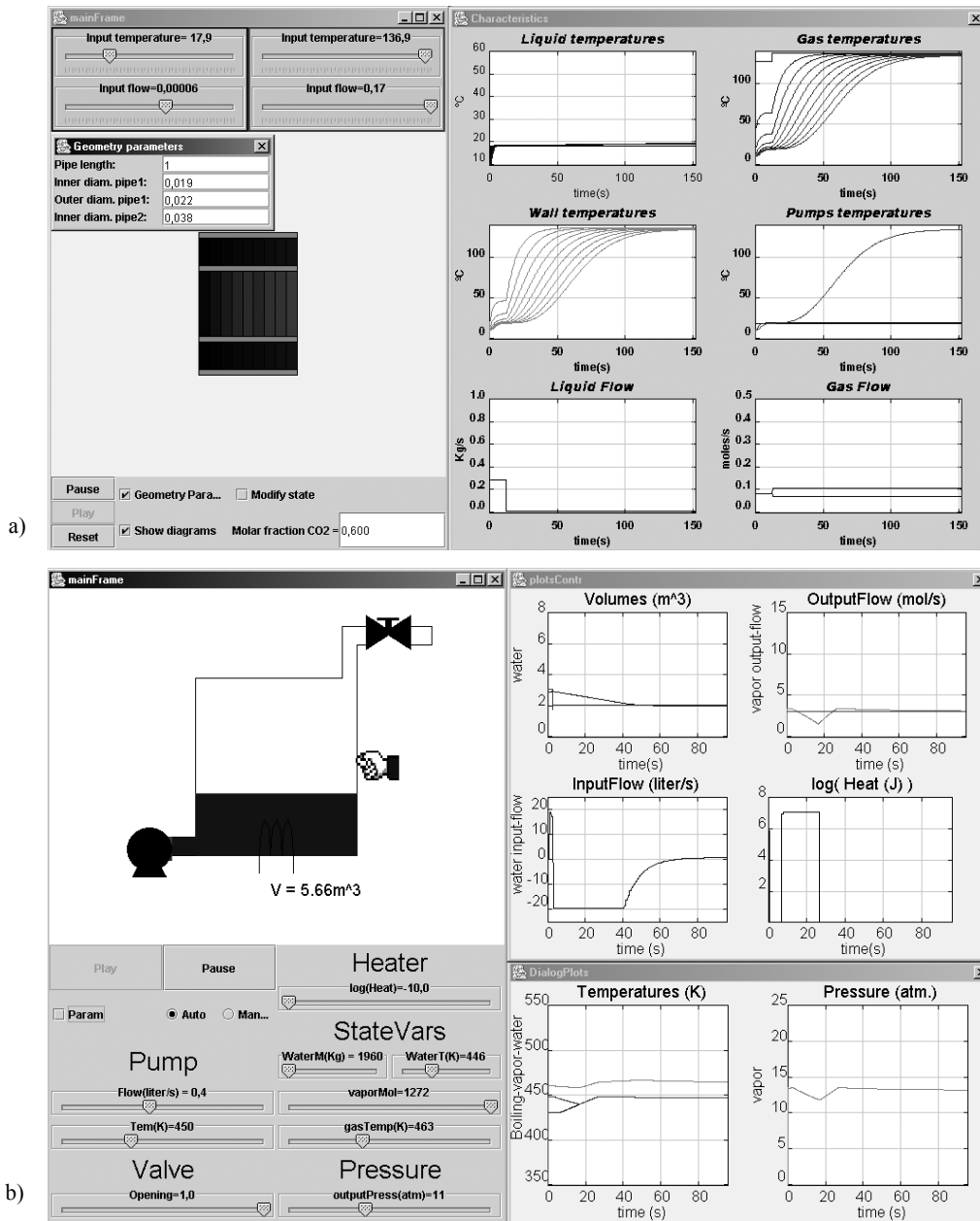


Figura 1: Vista en Ejs: a) Intercambiador de calor; b) Evaporador

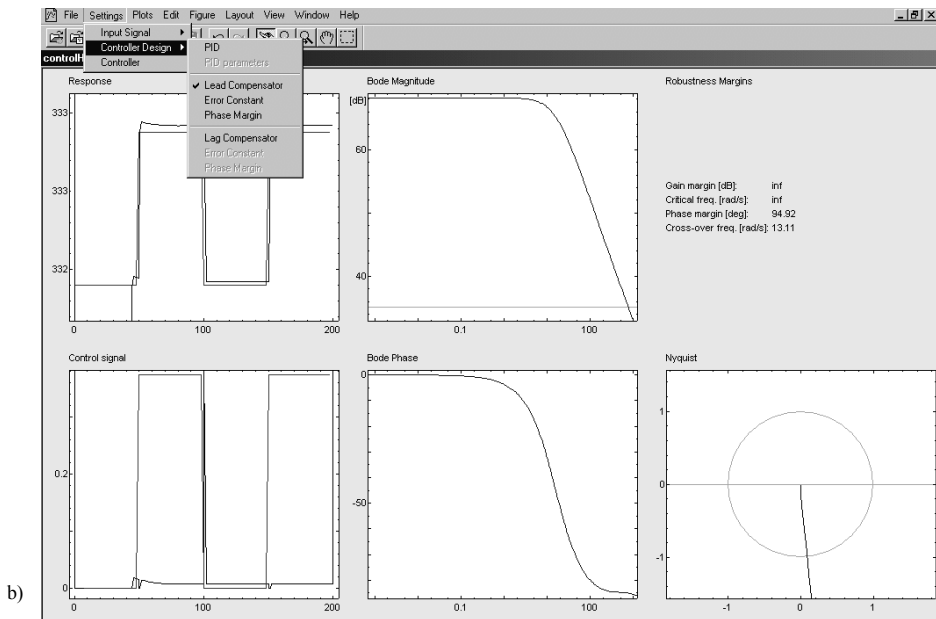
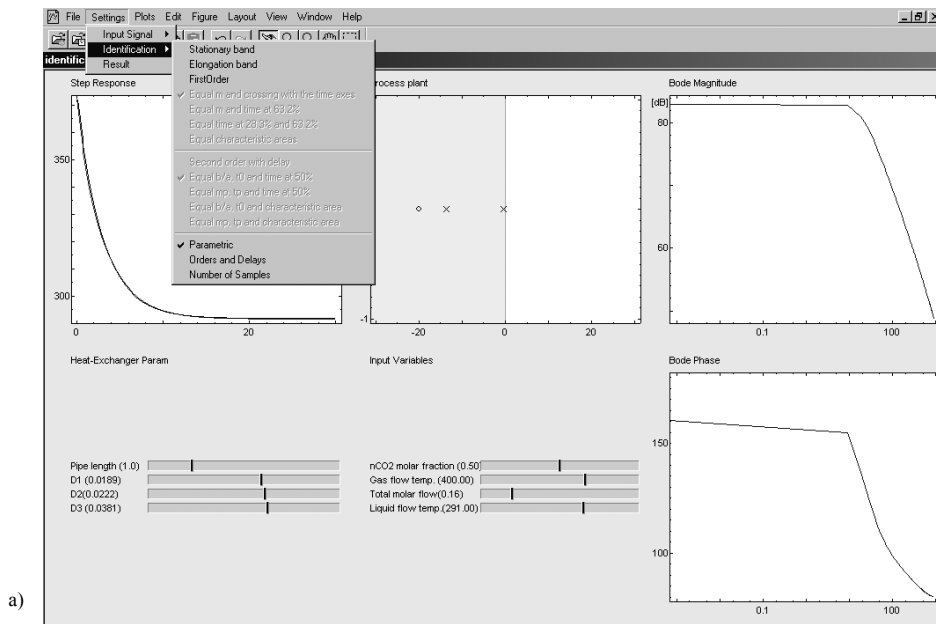


Figura 2: Vista en Sysquake del intercambiador de calor

del evaporador, y la salida de vapor en su parte superior. En el interior del evaporador se encuentra un calentador. Este modelo se ha compuesto usando JARA [5].

### 5.1. Laboratorio virtual con Ejs

La vista en Ejs del laboratorio virtual se muestra en la Figura 1b. El usuario puede seleccionar interactivamente entre dos estrategias de control: manual o PID descentralizado. Se usa un PID para controlar el nivel de agua y otro PID para controlar el flujo de vapor. Las variables manipuladas son, respectivamente, el flujo de agua de la bomba y el flujo de calor en el evaporador. Se pueden cambiar interactivamente los parámetros de los PID, los valores de las variables de estado del modelo (masa y temperatura del agua y del vapor), parámetros (volumen del recipiente), y variables de entrada (temperatura del agua de entrada, apertura de la válvula y presión de salida).

### 5.2. Laboratorio virtual con Sysquake

En este laboratorio se pretende identificar y controlar un evaporador industrial. Se controla el volumen del agua en el evaporador y el flujo de salida de vapor. Para ello se usan como variables manipuladas, respectivamente, el flujo de agua de la bomba de entrada y el flujo de calor del sistema calefactor. Se va a aproximar el sistema por dos F.T.: una F.T. para cada par de variable controlada y manipulada. Para la obtención de cada F.T. se aproxima la respuesta del sistema a un escalón en la entrada correspondiente.

Para la síntesis y análisis de controladores se ha creado otra aplicación en Sysquake. En esta aplicación el volumen de agua en el evaporador se controla con un PID. Para realizar el control de flujo de salida de vapor se puede seleccionar un controlador PID, una red de adelanto o de retardo de fase.

## 6. Conclusión

Se han demostrado las ventajas de combinar herramientas que facilitan la creación de interfaces interactivas (Ejs y Sysquake) con el

uso de un lenguaje y un entorno de modelado orientado a objetos (Modelica/Dymola). Se han propuesto y desarrollado métodos para combinar estas herramientas. Finalmente, se han aplicado estos métodos al diseño y programación de laboratorios virtuales para el aprendizaje de control de procesos químicos.

## Agradecimientos

Este trabajo ha sido financiado con fondos de los Proyectos CICYT DPI2001-1012 y DPI2004-1804.

## Referencias

- [1] Astrom K.J, Elmqvist H, Mattsson S.E. Evolution of Continuous-Time Modeling and Simulation. In: Proc. of the 12<sup>th</sup> European Simulation Multiconference, pp. 9-18, Manchester, UK, 1998.
- [2] Cutlip M.B, Shacham M. Problem Solving in Chemical Engineering with Numerical Methods. Prentice-Hall, 1999.
- [3] Dynasim AB. Dymola. User's Manual. Version 5.0a. Dynasim AB. Lund, Sweden.
- [4] Esquembre F. Easy Java Simulation: a Software Tool to Create Scientific Simulation in Java. In: Computer Physics Communications, vol. 156, pp. 199-204, 2004.
- [5] Martin C, Urquia A, Dormido S. JARA 2i - A Modelica Library for Interactive Simulation of Physical-Chemical Processes. In: Proc. European Simulation and Modelling Conference, pp. 128-132, 2004.
- [6] Martin C, Urquia A, Dormido S. Object-Oriented Modeling of Interactive Virtual Laboratories with Modelica. 4<sup>th</sup> International Modelica Congress, Hamburgo, Alemania, pp.159-168, Marzo 2005.
- [7] Ramirez, W. Computational Methods for Process Simulation, Butterworths Publishers, Boston, 1989.