



Actas del I Simposio en Modelado y Simulación de Sistemas Dinámicos SIMOSI'07

Editora
María Jesús de la Fuente Aparicio

Zaragoza, 12 y 13 de Septiembre de 2007





**ACTAS DEL I SIMPOSIO EN MODELADO Y SIMULACIÓN DE SISTEMAS
DINÁMICOS. SIMOSI'07**

No está permitida la reproducción total o parcial de este libro, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier otro medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros medios, sin el permiso previo y por escrito de los titulares del Copyright.

Derechos reservados ©2007 respecto a la primera edición en español, por LOS AUTORES
Derechos reservados ©2007 International Thomson Editores Spain, S.A.

Magallanes, 25; 28015 Madrid, ESPAÑA
Teléfono 91 4463350
Fax: 91 4456218
clientes@paraninfo.es

ISBN: 978-84-9732-619-3

Depósito legal: M-

Maquetación: Los Autores
Coordinación del proyecto: @LIBROTEX
Portada: Estudio Dixi
Impresión y encuadernación: FER Fotocomposición, S. A.

IMPRESO EN ESPAÑA-PRINTED IN SPAIN

Comité de Programa

María Jesús de la Fuente Aparicio. Universidad de Valladolid (**Organizador**)
César de Prada Morga. Universidad de Valladolid
Sebastián Dormido Bencomo. Universidad Nacional de Educación a Distancia
Josep Vehí Casellas. Universidad de Gerona
Jorge Bondía Company. Universidad Politécnica de Valencia
Rosa María Aguilar Chinea. Universidad de la Laguna
Emilio Jiménez Macias. Universidad de La Rioja.
Antonio Álvarez Alonso. IIM-CSIC
Miguel Angel Piera Eroles. Universidad Autónoma de Barcelona
Miguel Angel Ridaó Carlini. Universidad de Sevilla.
Diego Rubén Rodríguez. Universidad de Zaragoza

Presentación

Este Simposio en Modelado y Simulación de Sistemas Dinámicos (SIMOSI07) ha sido organizado por el Grupo de Modelado y Simulación de CEA (Comité Español de Automática).

Este Simposio pretende ser una conferencia de referencia donde investigadores y profesionales de España, y de Iberoamérica en el campo del modelado y la simulación de sistemas dinámicos, puedan intercambiar y debatir ideas, compartir experiencias y divulgar resultados, estableciendo, además un marco propicio de colaboración entre los distintos sectores y grupos de trabajo en estas áreas, haciendo especial hincapié en todos los aspectos del uso de las técnicas de Simulación como herramienta de ingeniería indispensable en la fase de toma de decisiones tan necesaria en la actual sociedad post-industrial.

En este volumen se presentan la colección de los 17 artículos aceptados en esta primera edición del simposio SIMOSI07. Los artículos están organizados en cuatro sesiones, por las siguientes áreas temáticas: Modelado y Simulación de Sistemas Continuos, Redes de Petri, Herramientas de Modelado y Aplicaciones del Modelado y la Simulación a Sistemas Biológicos y/o de Biomedicina. Esta revisión y aceptación de artículos se ha hecho con la ayuda del Comité de Programa del simposio.

El organizador de este congreso SIMOSI quiere expresar su agradecimiento a los miembros del Comité de Programa y aquellas personas que han participado como revisores adicionales, (Pastora Vega Cruz, de la Universidad de Salamanca, Gregorio Sainz Palmero y Felipe Acebes Arconada de la Universidad de Valladolid, Joaquín Armengol de la Universidad de Gerona, Juan José Ramos González de la Universidad Autónoma de Barcelona y Mercedes Pérez de la Parte de la Universidad de la Rioja).

Junio 2007

María Jesús de la Fuente Aparicio
Universidad de Valladolid

Índice General

Modelado y Simulación de Sistemas Continuos

Modelado, simulación y análisis de sistemas térmicos HVAC	3
B. Pérez-Lancho, E. Sanz, P. Vega, <i>Univ. de Salamanca (España)</i>	
Módulos de simulación dinámica de problemas típicos de control de procesos	9
Luis Felipe Acebes, Raul Alves, Alejandro Merino, Luis Palacín, Almudena Rueda, Rogelio Mazaeda, <i>Univ. de Valladolid (España)</i>	
José Acedo, <i>Repsol YPF (España)</i>	
Modelos con Incertidumbre: Identificación de parámetros intervalares y simulación usando cuantificadores	17
Sandra M. Castillo, Esteban R. Gelso, Joaquim Armengol, <i>Univ. de Girona (España)</i>	
Sistema Reconfigurable para la Simulación en Bloque de Experimentos Útiles en Tareas de Soporte para Supervisión y Diagnosis	25
Anibal Bregón, Diego García, Oscar Prieto, Belarmino Pulido, Carlos Alonso, <i>Univ. de Valladolid (España)</i>	
Simulación Centralizada Múltiple Frente a Simulación Distribuida Aplicada a un Robot Móvil Autónomo	33
Norberto Cañas de Paz, <i>Universidad Politécnica de Madrid (España)</i>	
Gracián Triviño Barros, <i>European Centre for Soft Computing, Mieres (España)</i>	

Modelado y Simulación de Sistemas de Eventos Discretos con Redes de Petri

Especificación, mediante Redes de Petri Coloreadas, del comportamiento de un sistema multiagente para abordar problemas logísticos	43
Sònia Busquets, Miquel Àngel Piera, <i>Univ. Autònoma de Barcelona (España)</i>	
Distribución del espacio de estados de una Red de Petri Coloreada	51
Laia Villalta, Miquel Àngel Piera, <i>Univ. Autònoma de Barcelona (España)</i>	

Integración de modelo gráfico virtual y modelo de toma de decisiones para gestión basada en simulación de una bodega.....	59
Mercedes Pérez De la Parte, Emilio Jiménez Macías, Marta Vega Ollora, <i>Universidad de La Rioja (España)</i>	
Un entorno de simulación para el diseño de herramientas de toma de decisiones en logística de transporte.....	67
D. Guimarsans, J. J. Ramos, M. À. Piera, <i>Univ. Autónoma de Barcelona (España)</i>	
A. Guasch, <i>Univ. Politécnica de Catalunya (España)</i>	

Herramientas de Modelado y Simulación

Simulación de Sistemas Dinámicos complejos mediante Autómatas Celulares de comportamiento macroscópico definido	77
Joaquín Cerdá Boluda, Néstor Ferrando Jódar, Rafael Gadea Gironés, Ricardo José Colom Palero, <i>Univ. Politécnica de Valencia (España)</i>	
Modelado y simulación de un Centro de Atención al Usuario con SIGHOS	85
Iván Castilla, Roberto Muñoz, Rosa M. Aguilar, Pedro J. Baquero, Yurena García-Hevia, <i>Univ. de La Laguna (España)</i>	
MEST y CEST: Herramientas para la Predicción de Descarga Electrónica en Guías de Onda	95
Francisco Pérez, Juan de Lara, Manuel Alfonseca, Luis Galán, <i>Univ. Autónoma de Madrid (España)</i>	
Luis Conde, <i>Univ. Politécnica de Madrid (España)</i>	
Métodos de reducción del tiempo de cálculo en la exploración de espacios de estados generados a partir de modelos de redes de Petri Coloreadas	103
Mercedes E. Narciso Farias, Miquel A. Piera i Eroles, <i>Univ. Autònoma de Barcelona (España)</i>	

Aplicación del Modelado y la Simulación a Procesos de Biomedicina y Biológicos

Optimización y diseño de intervenciones adaptativas considerando gestión de riesgos	113
Ascensión Zafra-Cabeza, Miguel A. Ridao, Eduardo F. Camacho, <i>Universidad de Sevilla (España)</i>	
Predicción de glucemia en pacientes de diabetes tipo 1 ante incertidumbre	121
J. Bondia, C. Tarín, <i>Univ. Politécnica de Valencia (España)</i>	
R. Calm, M. García-Jaramillo, J. Vehí, W. García-Gabín, <i>Univ. de Girona (España)</i>	

Sistema de Ayuda a la Toma de Decisiones en la Gestión del Clima Interior de un Invernadero Basado en Modelos de Crecimiento de Cultivos de Tomate	129
Javier Bonilla, Luís José Yebra, <i>PSA-CIEMAT, Almería (España)</i>	
Francisco Rodríguez, <i>Universidad de Almería (España)</i>	
Armando Ramírez, <i>Universidad Autónoma de Chapingo (Méjico)</i>	
Desarrollo de un modelo de red neuronal (Sierra2) para la selección de especies arbustivas del ámbito mediterráneo, en el marco del Sistema de Ayuda a la Decisión MicroLEIS DSS	137
N. Heredia, <i>Centro Internacional de Altos Estudios Agronómicos Mediterráneos (CIHEAM), Instituto Agronómico Mediterráneo de Zaragoza (IAMZ) (España)</i>	
M. Anaya-Romero, F. Mayol, E. Díaz-Pereira, D. de la Rosa, Consejo Superior de Investigaciones Científicas (CSIC), Instituto de Recursos Naturales y Agrobiología de Sevilla (IRNAS) (España)	

MODELADO Y SIMULACIÓN DE SISTEMAS CONTINUOS

Modelado, simulación y análisis de sistemas térmicos HVAC

B. Pérez-Lancho

Dept. Informática y Automática
Univ. de Salamanca
Facultad de Ciencias
Plaza de la Merced, s/n
37008 Salamanca
lancho@usal.es

E. Sanz

Dept. Informática y Automática
Univ. de Salamanca
Facultad de Ciencias
Plaza de la Merced, s/n
37008 Salamanca
esanz@usal.es

P. Vega

Dept. Informática y Automática
Univ. de Salamanca
ETS Ingeniería Industrial
C/ Fernando Ballesteros, s/n
37700 Béjar (Salamanca)
pvega@usal.es

Resumen

El objetivo de este trabajo es presentar una herramienta flexible de modelado y simulación de sistemas térmicos HVAC (Heating, Ventilating and Air Conditioning) en grandes edificios, la cual puede utilizarse para diseñar nuevas instalaciones y probar estrategias de control clásicas y avanzadas. El modelado de los diferentes elementos que componen un sistema HVAC a partir de los principios generales de balance de energía ha permitido crear una estructura de bloques basada en *Matlab/Simulink* con la que se puede simular fácilmente el comportamiento de un sistema térmico. En particular se ha utilizado la librería de bloques para modelar una parte de un edificio universitario concreto donde se han comparado los resultados del modelo con los experimentales. Se ha logrado así validar la herramienta que posteriormente permitirá analizar diferentes estrategias de control.

1. Introducción

Las características y las condiciones ambientales en el interior de edificios están evolucionando continuamente, debido a las mayores exigencias en la calidad del aire y a las demandas crecientes en la confortabilidad de estas instalaciones. También se hace necesario conjugar los argumentos anteriores con la necesidad de un ahorro energético y económico motivado, entre otras razones, por el cumplimiento de los acuerdos establecidos en la convención de Kyoto.

En pocos casos se hace un análisis exhaustivo de los factores determinantes en el consumo energético y en las condiciones medio-

ambientales referidos a las instalaciones térmicas. En los últimos años solamente en [4] podemos ver este estudio pormenorizado referido a edificios en UK.

Grandes y medianos edificios comparten desde hace tiempo el mismo sistema de aclimatación. Básicamente se usan dos unidades de producción, una de agua caliente y otra fría. El agua circula por circuitos separados hasta intercambiadores de calor agua-aire, elementos conocidos como Unidades de Tratamiento de Aire (UTA o AHU). Desde ellas el aire es impulsado por conducciones propias a las salas y recintos donde se impulsa y posteriormente retorna por medio de toberas y difusores. Su finalidad es conseguir mantener ciertos perfiles deseados de temperatura en las distintas zonas del interior del recinto durante su período de actividad. La arquitectura y la ingeniería térmica del conjunto y de sus elementos pueden verse en [3].

Se hace necesario así realizar un control jerárquico cuyo objetivo final está determinado por caudales y temperaturas tanto de agua como de aire en las diversas unidades. Pueden establecerse hasta tres lazos de control anidados en función del diseño de la instalación. Esta jerarquía, junto con la no linealidad intrínseca en muchos elementos así como los retardos y las dinámicas de características variables hace bastante compleja la misión de cumplir los objetivos citados.

Tanto para el diseño térmico como para el control de las variables del proceso es de una gran utilidad disponer de una herramienta de modelado y simulación de este tipo de sistemas, con condiciones de generalidad y adecuación al ámbito específico de la ingeniería térmica. El objetivo final es poder crear modelos fiables y

flexibles que permitan comprobar la dinámica completa de los sistemas y comparar los resultados de simulación con los obtenidos experimentalmente. Algunos trabajos ([5], [6], [7]) aportan soluciones a este último enfoque considerando algún diseño particular. En [2] se parte de una analogía de este tipo de sistemas con sus equivalentes eléctricos. El intento más cercano a una descripción global del sistema podemos encontrarlo en [8].

En este trabajo se presenta una herramienta muy flexible que permite modelar los diferentes elementos de un sistema HVAC así como el sistema global. La herramienta, basada en *Matlab/Simulink*, permite el diseño de sistemas térmicos y el análisis y comparación de estrategias de control. Está constituida por un conjunto de bloques agrupados por sus funcionalidades. En concreto se incluyen elementos de producción, distribución e intercambio energético, elementos de control, bloques asociados a recintos y entornos y señales adicionales. Está integrada como una *toolbox* nueva, dispone de ayudas y demostraciones y es fácilmente ampliable. Se ha utilizado con éxito para la simulación del comportamiento térmico de un edificio universitario y los resultados han permitido el análisis de los problemas que actualmente se presentan y facilitarán en el futuro la implantación de nuevas estrategias de control.

2. Descripción del sistema

La arquitectura de un sistema HVAC consta de tres elementos básicos: la unidad de producción, el intercambiador y el recinto. A ellos habrá que añadir los componentes adicionales tales como los referentes a la conducción de agua y aire (tuberías, toberas, etc.) y a los elementos de control (controladores, actuadores y sensores).

Cada elemento básico ha sido objeto de un minucioso estudio de modelado basado en principios de conservación y transferencia energética y en sus características geométricas y térmicas.

Se han tenido en cuenta asimismo los factores perturbadores externos, tales como el clima, el grado de ocupación del edificio o las fuentes ocasionales de energía.

2.1. Modelos de recinto

Los modelos térmicos de los recintos interiores de un edificio que pueden desarrollarse tienen diferentes niveles de complejidad. Se podría considerar por ejemplo una dinámica de fluidos completa con variaciones temporales y espaciales tridimensionales en las variables térmicas. Una aproximación más simplificada sería tener en cuenta exclusivamente variaciones preferentes en direcciones privilegiadas, o dividir el recinto en volúmenes de control con transferencias energéticas entre ellos. La más sencilla y manejable consiste en establecer exclusivamente modelos homogéneos únicos en el espacio del recinto. En este caso se considera un solo volumen de control asociado al fluido interior. Las transferencias se efectúan con los volúmenes de control asociados a los elementos que componen contornos. En el caso de climatización de salas con sistemas HVAC la capacidad calorífica del recinto es generalmente muy elevada, la dinámica del proceso es muy lenta y la aproximación de homogeneidad de la temperatura es correcta siempre que los difusores estén distribuidos uniformemente a lo largo de la sala.

Por todo ello se ha escogido un modelo simplificado de segundo orden que representa fielmente el comportamiento térmico de una sala regular de unos 5000 m³ y que aproxima suficientemente bien los resultados teóricos a los experimentales. Las ecuaciones que representan el modelo del recinto considerado son:

$$\begin{aligned} C_a \frac{dT}{dt} &= Q - (U_T A)_{mi}(T - T_m) - (U_T A)_v(T - T_{ex}) \\ C_m \frac{dT_m}{dt} &= (U_T A)_{mi}(T - T_m) - (U_T A)_{me}(T_m - T_{ex}) \end{aligned} \quad (1)$$

y el flujo calorífico en la sala se modela como:

$$Q = c_a \rho_a q_a (T_{ai} - T) \quad (2)$$

Las variables T , T_m , T_{ex} y T_{ai} son respectivamente las temperaturas del aire en el recinto, de las paredes (muros), del aire exterior y del aire procedente del intercambiador de calor. Los términos $(U_T A)_{mi}$, $(U_T A)_v$ y $(U_T A)_{me}$ representan las transferencias térmicas

equivalentes y totales de muros interiores, ventanas y muros exteriores. Los valores de U son los inversos de las resistencias térmicas en cada caso. Las capacidades caloríficas C_a y C_m se refieren al aire del interior del recinto y a las paredes o muros exteriores respectivamente, y los parámetros c_a , ρ_a y q_a representan el calor específico, la densidad y el flujo volumétrico del aire inyectado en la sala.

Considerando aisladamente el recinto y la temperatura interior como variable controlada, el control podría considerarse lineal y el sistema asimismo lineal si se considerase la temperatura de impulsión como única variable de control, en cuyo caso debe ser constante el flujo (caudal de aire), que es lo que ocurre frecuentemente en la realidad. En cualquier otra situación, con sistemas de control más elaborados, aparecerán en el modelo términos no lineales cruzados como productos de señales de control y variables de estado.

Una variable muy importante, tanto en el diseño térmico inicial de una instalación como en el control y mantenimiento de la misma, de cara a asegurar simultáneamente la confortabilidad interior y el ahorro energético es la carga térmica del recinto (TL). Se define como la diferencia energética entre el valor deseado o esperado (T_{di}) y el actual (T).

$$TL = c_a \rho_a V_a (T_{di} - T) \quad (3)$$

2.2. Unidad de Tratamiento del Aire (UTA)

El intercambiador de calor agua-aire, más conocido como la unidad de tratamiento de aire, es el elemento nuclear de una instalación HVAC. Su función es realizar el intercambio energético entre el agua (fría o caliente) y el aire, e impulsar este aire al recinto cerrado que se pretende climatizar.

Las ecuaciones dinámicas del proceso con ligaduras entre los dos fluidos dan lugar a un bloque complejo que se ha pretendido visualizar, desde el punto de vista externo, con los esquemas de un subsistema, con la descripción de las entradas y salidas y también con un ícono gráfico representativo de la funcionalidad del elemento. En la figura 1 puede verse el aspecto de este

bloque, y en la figura 2 el cuadro de diálogo a través del cual se particulariza el tipo de intercambiador, asignando a sus parámetros los valores correspondientes.

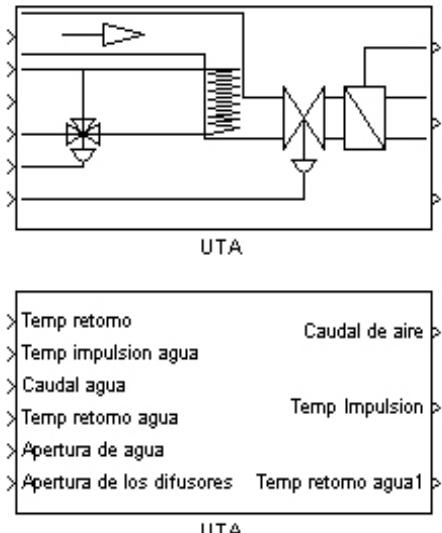


Figura 1. Ícono del módulo UTA y señales de entrada/salida.

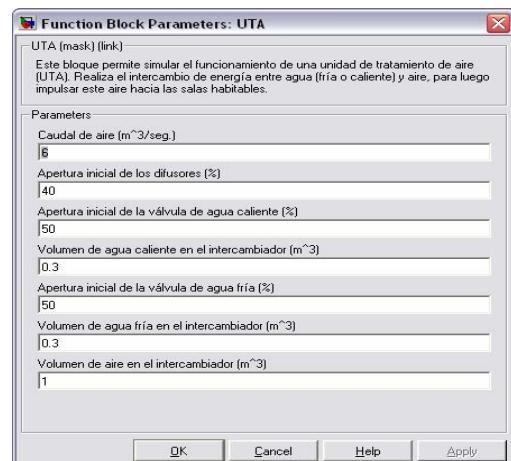


Figura 2. Parámetros del intercambiador del agua.

El elemento esencial de control de la temperatura del aire de salida de la UTA es la válvula de agua de tres vías, cuya operación y

características aparecen en la figura 3. El servomotor que actúa sobre dicha válvula está accionado por un regulador de tipo PI cuyos parámetros pueden ser modificados exclusivamente de forma manual.

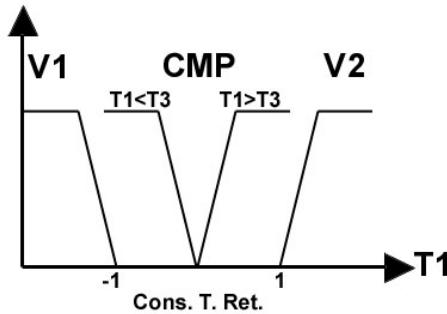


Figura 3. Sistema de actuación en las válvulas de agua de la UTA.

Adicionalmente existen bombas soplantes que recirculan el aire del interior y que, mediante renovaciones periódicas o variables del aire exterior, modifican igualmente el flujo energético del volumen de control. El caudal de aire intercambiado con el exterior podría ser considerado como una perturbación o como una variable de control alternativa.

Los parámetros asociados al bloque, que representan las características geométricas y físicas de la UTA utilizada, pueden ser obtenidos de los datos reales o deducidos experimentalmente.

Las ecuaciones asociadas a este elemento son:

$$\begin{aligned} C_a \frac{dT_{ai}}{dt} &= U_T A(T_{wr} - T_{ai}) - q_a \rho_a c_a T_{ai} + \\ &+ f(q_{ae}, q_{ar}, T_{ex}, T_{ar}) \quad (4) \\ C_w \frac{dT_{wr}}{dt} &= q_w \rho_w c_w (T_{wi} - T_{wr}) - U_T A(T_{wr} - T_{ai}) \end{aligned}$$

La interpretación de las variables es similar a la descrita en la Ec. (1). En este caso los subíndices *a* y *w* hacen referencia al aire y al agua respectivamente, y los subíndices *i* y *r* indican si se trata de una temperatura de impulsión o de retorno a la UTA. La función $f(\cdot)$ representa la combinación de flujos de aire exterior y de retorno con sus temperaturas correspondiente. De acuerdo

con las necesidades y normativas de ASHRE [1] sobre ventilación y calidad de aire interior en un edificio público con características de biblioteca universitaria la renovación de aire tiene como valor mínimo 10 l(seg) por persona.

Las descripciones de las variables de control utilizadas en el intercambiador pueden ser sencillas o elaboradas, pero todas ellas conducen a sistemas de control no lineales. Caudales y temperaturas (de agua y aire) en las vías de entrada al intercambiador, esto es las correspondientes del retorno o ventilación del aire, y en las de impulsión del agua procedente del elemento de producción (caldera o enfriadora) pueden dar lugar a multitud de estrategias de control, complejas o simplificadas, cercanas a la realidad o modeladas por sistemas sobredimensionados en el vector de control.

2.3. Modelo de la caldera y el circuito de agua

La dinámica asociada a este elemento viene regida por las siguientes ecuaciones:

$$\begin{aligned} C_{wb} \frac{dT_{wi}}{dt} &= Q_g + q_{wb} \rho_w c_w (T_{wr} - T_{wi}) - \\ &- (U_T A)_{pe} (T_{wr} - T_{ar}) \quad (5) \\ Q_g &= \eta P u_0 \end{aligned}$$

Aparecen tres términos que determinan la dinámica de la temperatura del agua, el primero es debido a la combustión en el quemador, el segundo a la conducción a través de las tuberías entre caldera y UTA y el tercero es debido al calor disipado por radiación del propio fluido al entorno. Los dos primeros términos desaparecen en el período de apagado del sistema en los horarios establecidos de forma sincrónica entre las unidades del sistema (combustión y bombeo) y el tercero es el responsable del enfriamiento gradual del agua en estos períodos de apagado. De acuerdo con la terminología habitual, q_{wb} es el caudal de agua a través del quemador, C_{wb} la capacidad calorífica del agua contenida en el circuito en contacto con el quemador, P la potencia nominal de la unidad de producción y η su rendimiento. El parámetro $(U_T A)_{pe}$ establece la transferencia térmica equivalente por radiación entre el agua de las tuberías y el entorno.

La variable controlada en este bloque es la temperatura del agua de salida T_{wi} cuyo valor

deseado se intenta alcanzar mediante un controlador no lineal de tipo todo/nada que actúa sobre el flujo porcentual del gas, u_0 .

3. Simulación y resultados

Se ha simulado el comportamiento dinámico individual de cada elemento básico del sistema. Cada uno de ellos corresponde a un bloque, cuyo modelo y características pueden especificarse previamente.

Con la librería de elementos creada es posible simular los comportamientos dinámicos de los modelos parciales o globales del sistema y simultáneamente compararlos con los resultados procedentes del sistema real, que pueden añadirse al modelo como ficheros propios del espacio de trabajo.

Así el esquema de la figura 4 permite la simulación del modelo del recinto y además la representación simultánea y sincrónica de los datos reales tomados a lo largo de varios días consecutivos junto con los simulados. Se ha tenido en cuenta que el flujo calorífico de control en el modelo sea el mismo que el registrado experimentalmente, esto es, el procedente de la UTA real. De igual forma sucede con la temperatura exterior y los eventos de encendido/apagado del sistema.

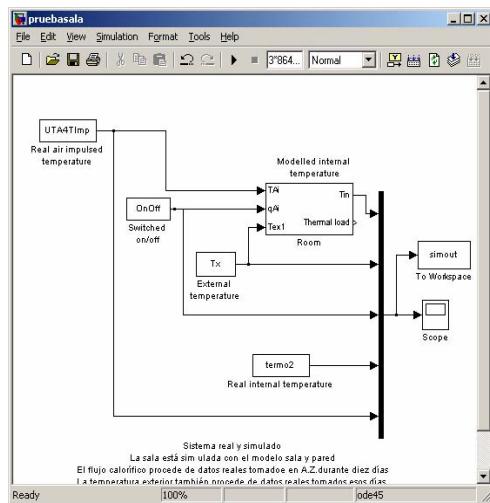


Figura 4. Modelo del recinto y datos reales.

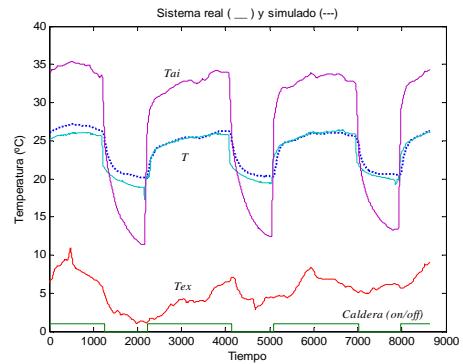


Figura 5. Valores reales y simulados del recinto.

La representación gráfica de los resultados se presenta en la figura 5. En la parte superior de la gráfica aparecen las señales correspondientes a las temperaturas interiores del recinto durante tres días. En línea continua se muestran las temperaturas reales del aire impulsado (T_{ai}) y aire en el recinto (T), mientras que la línea discontinua representa la temperatura simulada por el modelo propuesto. En la parte inferior se pueden observar la temperatura exterior registrada esos días (T_{ex}) y la secuencia de encendidos y apagados del proceso real.

La situación reflejada es muy habitual en el comportamiento térmico del edificio, pero está bastante alejada de los parámetros adecuados de confortabilidad y de ahorro energético. En efecto puede observarse un sobrecalentamiento muy pronunciado en los períodos de ocupación. Este hecho está provocado por un mal diseño de la instalación del edificio así como por una inadecuada estrategia de control que consistía exclusivamente en el control de la variable de salida de la caldera y en una secuencia fija de encendidos y apagados. No obstante ha servido para poder ajustar y validar el modelo del recinto a la realidad, que en este caso no es confortable y tampoco es rentable desde el punto de vista económico ni medioambiental.

Este modelo térmico obtenido se puede utilizar para probar y analizar otras estrategias de control tanto clásicas como avanzadas que pudieran mejorar los resultados ofrecidos en esta simulación. A modo de ejemplo se presenta en este contexto el estudio de una posible alternativa de control predictivo en lazo abierto basado en

modelo (MPC). Si se dispusiera de elementos actuadores que controlaran adecuadamente la temperatura del aire impulsado al recinto se podría implementar dicho control para conseguir unos perfiles de temperatura interiores deseados. Por ejemplo estableciendo como valores umbráles de referencia 22°C y 15°C, se podrían obtener comportamientos similares al que se muestra en la simulación representada en figura 6.

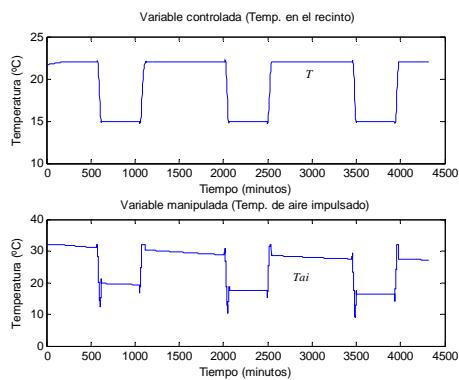


Figura 6. Temperatura del aire interior y de impulsión con un control predictivo del recinto.

4. Conclusiones

En este trabajo se ha desarrollado una herramienta muy flexible para el diseño de sistemas térmicos de edificios. La herramienta permite el análisis de los distintos modelos de elementos del sistema HVAC y la simulación de la dinámica completa de los comportamientos térmicos en grandes edificios. Se ha probado con varios modelos y se ha verificado la concordancia entre los resultados obtenidos por simulación y los experimentales procedentes de algunos sistemas reales.

En particular se ha considerado una sala dentro de una biblioteca universitaria y, tras un estudio detallado de sus elementos y sus características, se ha realizado su modelo térmico completo. La herramienta ha permitido la simulación de dicho modelo y la comparación de su respuesta con el funcionamiento observado en el sistema real durante varios días de operación. Se ha validado así el planteamiento teórico y las hipótesis consideradas, y además se han puesto de manifiesto problemas ya detectados de

sobrecalentamiento debidos a un mal diseño de la instalación y a una inadecuada estrategia de control. El modelo obtenido será de gran ayuda a la hora de intentar solucionar estos problemas, puesto que permite plantear alguna técnica de control más elaborada, como por ejemplo un control predictivo basado en modelos (MPC).

En definitiva, herramienta y modelo puede ser utilizados como banco de simulación y prueba de distintos métodos y estrategias de control de la instalación. El planteamiento es muy útil para la experimentación con técnicas avanzadas de control tales como control predictivo, control neuronal, control fuzzy, etc. que permitan lograr el objetivo deseado del compromiso entre la confortabilidad, calidad de aire interior y ahorro energético.

Referencias

- [1] Andersen K., Madsen H., Hansen L. (2000). Modelling the heat dynamics of a building using stochastic differential equations. In: *Energy and Buildings*, 31 , p. 13-24.
- [2] García-Sanz M. (1997). A reduced model of central heating systems as a realistic scenario for analyzing control strategies. In: *Appl. Math. Modelling*, 21 , p. 535-545.
- [3] Kuehn T.H., Ramsey J.W., Threlkeld J.L. (1998). *Thermal Environmental Engineering*, Prentice Hall Int.
- [4] Liao Z., Swainson M., Dexter A. (2005). On the control of heating system in the UK. In: *Building and Environment*, 40 , p. 343-351.
- [5] Mathews E. H., van Heerden E., and Arndt D.C. (1999). A tool for integrated HVAC, building, energy, and control analysis. In: *Building and Environment*, 34 , p. 429-449.
- [6] Riederer P., Marchio D., Visier J.C., Husaunnde A. and Lahrech R. (2002). Room thermal modelling adapted to the test of HVAC control systems. In: *Building and Environment*, 37 , p. 777-790.
- [7] Stec W.J., van Paassen A.H.C. (2005). Symbiosis of the double skin façade with the HVAC system. In: *Energy and Buildings*, 37 , p. 461-469.
- [8] Tashtoush B., Molhim M., Al-Rousan M. (2005) Dynamic model of an HVAC system for control analysis. In: *Energy*, 30, p. 1729-1745.

Módulos de simulación dinámica de problemas típicos de control de procesos

Luis Felipe Acebes

Dept. de Ingeniería de Sistemas y Automática
EUP. Univ. de Valladolid.
47014 Valladolid
felipe@autom.uva.es

Raul Alves, Alejandro Merino, Luis

Palacín, Almudena Rueda,
Rogelio Mazaeda

Centro de Tecnología Azucarera. Univ. de Valladolid

José Acedo
Repsol YPF

Resumen

El objeto de este artículo es presentar los resultados de un proyecto consistente en el desarrollo de una aplicación informática, para entornos Windows®, que permita la selección de un problema típico de control de procesos, de entre una librería de problemas de control predefinida, y su posterior simulación en un entorno amigable de modo que pueda ser usado como recurso docente para el aprendizaje del problema de control seleccionado.

El proyecto se desarrolla dentro del marco de colaboración del Departamento de Ingeniería de Sistemas y Automática y el Centro de Tecnología Azucarera, ambos pertenecientes a la Universidad de Valladolid, con la sección española de la ISA España (Instrument Society of America [1]) y su Master en Instrumentación y Control de Procesos.

1. Características de los módulos de simulación

Los sistemas simulados son los siguientes:

- ✓ Control antibombeo en compresor centrífugo.
- ✓ Control de reciclo mínimo en bomba centrífuga.
- ✓ Control de reactores químicos.
- ✓ Control de carga y reciclo en compresores alternativos.
- ✓ Control de nivel “on-off” de un depósito
- ✓ Control de un horno de producto.
- ✓ Control de temperatura de salida de un cambiador con compensación estática *feedforward*.

Existe una aplicación, denominada *LanzaSim*, que muestra un menú desplegable desde el cual se puede seleccionar el módulo de simulación deseado (Figura 1).



Figura 1

Para cada módulo de simulación (en la Figura 2 puede verse uno de ellos), la interfaz de usuario consta del esquema del proceso tipo P&I (*Process & Instrumentation*, esquemático del proceso a controlar, instrumentación instalada y estructura de control), y de un conjunto de botones y barras de desplazamiento mediante los cuales el alumno podrá cambiar las condiciones de operación del sistema (condiciones de contorno, Figura 3) y acceder a los parámetros de sintonía de la estructura de regulación. También se dispone de controles gráficos a través de los cuales se visualiza la evolución temporal de las distintas variables de interés, pudiéndose representar gráficas de tendencia actuales e históricos (Figura 5).

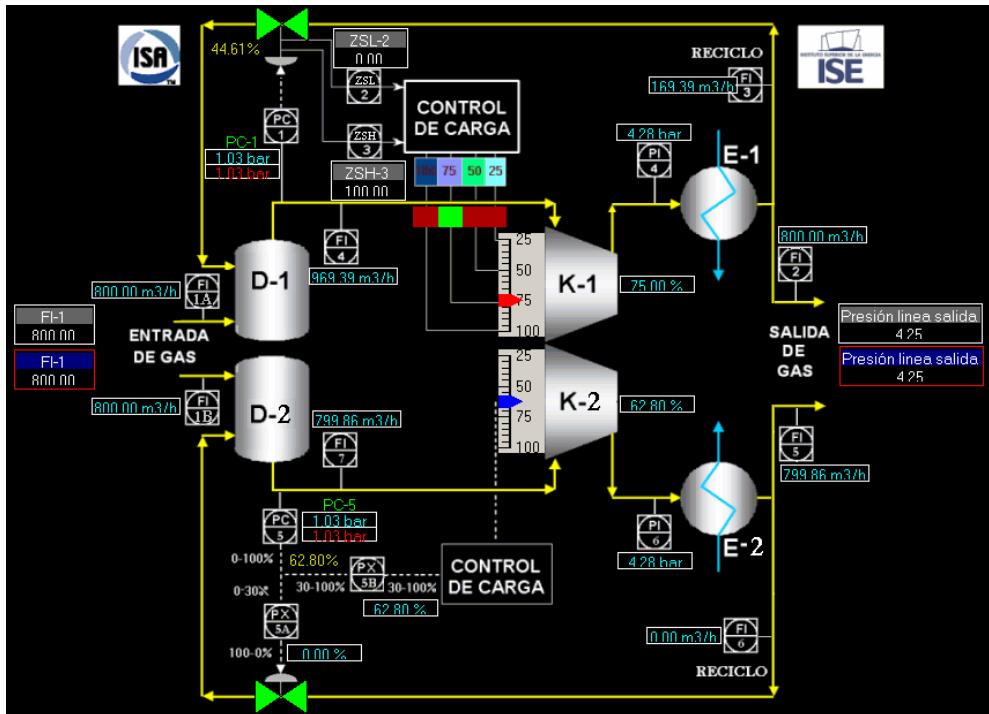


Figura 2. Módulo compresores alternativos



Figura 3

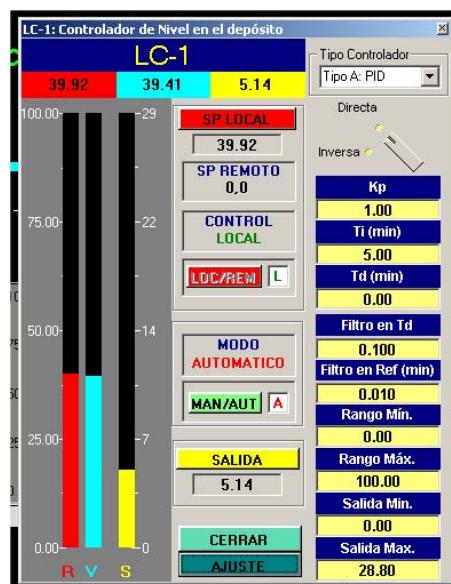


Figura 4

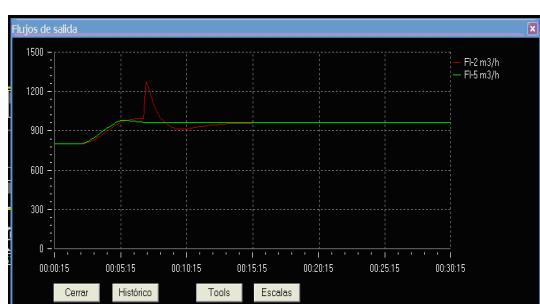


Figura 5

En particular, y con respecto a los controladores, que en general son de tipo PID (Figura 4), aunque también los hay de tipo incremental, el alumno puede seleccionar la ley de control, modificar la referencia del lazo, sintonizar el controlador, comutar de modo automático a manual (modificando si es preciso la señal de control cuando el controlador esté en manual) y viceversa, comutar de modo local a remoto (permitiendo que la referencia le llegue de un bloque externo) y viceversa. También puede modificar los parámetros del regulador (ganancia, tiempos integral y derivativo) y los rangos de medida y actuación.

La simulación por defecto se ejecuta en tiempo real, siendo posible parar y reanudar la simulación, así como acelerar su velocidad de ejecución con respecto al tiempo real.

2. Estructura software y herramientas de desarrollo

Cuando se selecciona un determinado módulo se invoca la ejecución de un sistema SCADA propio, denominado *EDUSCA*, que actúa con interfaz de usuario, dicho SCADA ordena la ejecución de la simulación dinámica asociada a dicho módulo.

La desarrollo de la interfaz de usuario de cada simulador, implica configurar *EDUSCA*, lo cual se realiza mediante la estrategia de *drag&drop* a través de una aplicación propia denominada *SetupEdusca* (Figura 6). Dicha configuración resulta rápida, sencilla e intuitiva. Se escapa del alcance de esta comunicación la descripción tanto de *EDUSCA*, como de su herramienta de configuración, en [2] puede encontrarse información al respecto.

La comunicación entre *EDUSCA* y la simulación se realiza por medio del estándar de *facto* para comunicaciones en aplicaciones de control de procesos bajo entornos *Windows* denominado *OPC* (*OLE for Process control* [3]). *EDUSCA* dispone de capacidades para actuar como cliente *OPC* y la simulación actúa como servidor *OPC*.

Las simulaciones dinámicas se han realizado usando el entorno de modelado de sistemas físicos

orientado a objetos (en base a ecuaciones algebraico-diferenciales y eventos) denominado *EcosimPro* [4]. La programación de los modelos se hace en un lenguaje de denominado *EcosimPro Language*, que resulta muy próximo al considerado estándar de este tipo de lenguajes denominado *Modelica* [5]. El entorno de programación de modelos en *EcosimPro* puede verse en la Figura 7.

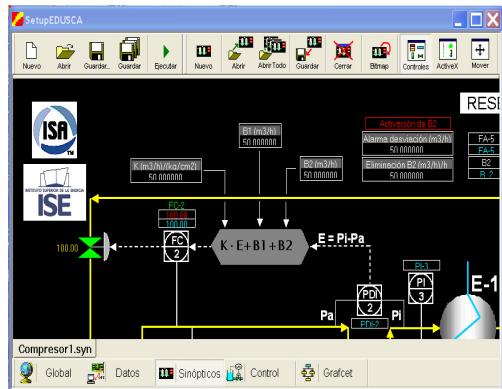


Figura 6

EcosimPro tiene una interfaz para el uso de las simulaciones (*EcoMonitor*), pero dicha interfaz no era de nuestro interés como interfaz gráfica de usuario final (Figura 8). Ya que para acceder a las variables y representarlas gráficamente es necesario conocer su identificador. Consideramos que *Ecomonitor* es una herramienta adecuada para el programador, pero no tanto para un usuario externo del modelo.

Por otro lado, *EcosimPro* a partir del modelo matemático del sistema a simular y del experimento definido por el usuario genera un conjunto de clases *VC++* que usa para generar el ejecutable del modelo de simulación. Este conjunto de clases *VC++* están disponibles para el usuario, y pueden ser incrustadas en el desarrollo de distintas aplicaciones, en particular pueden ser utilizadas para crear una aplicación que simule el sistema y se comunique con otras aplicaciones vía *OPC* [6].

Para facilitar, y automatizar, esta tarea se ha desarrollado una aplicación informática propia

denominada *CreaOPC*, que a partir de un modelo de simulación en forma de clase *VC++* desarrollado con *EcosimPro*, permite generar una aplicación informática cuya ejecución sea la

simulación del proceso en tiempo real (o acelerado) y con capacidad para comunicarse con otras aplicaciones a través del interfaz *OPC*.

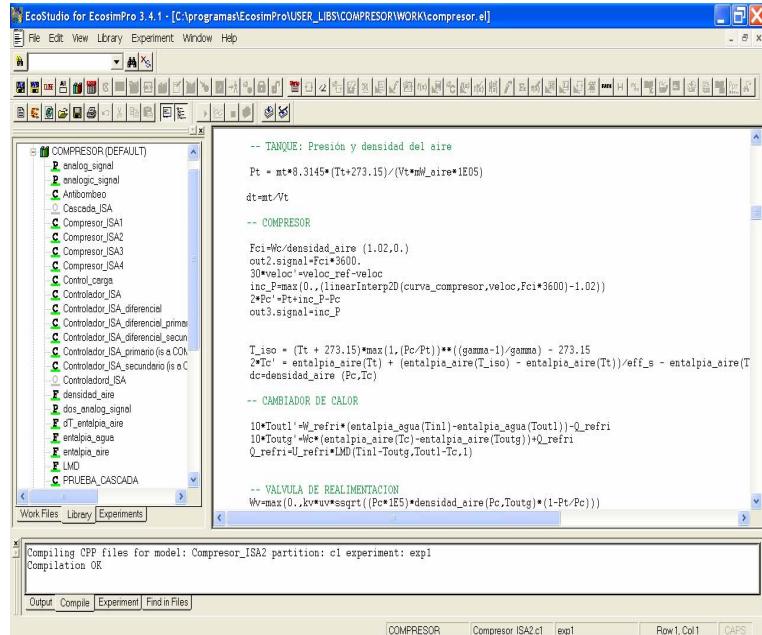


Figura 7

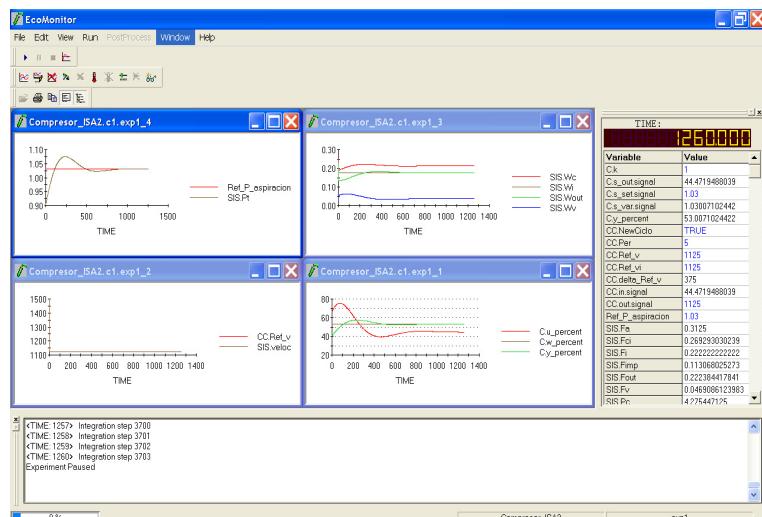


Figura 8

Realmente, el ejecutable generado es una aplicación informática con tres partes. La primera, la simulación dinámica, la segunda, un hilo que controla el avance de la variable tiempo de la simulación y, la tercera, una capa de comunicaciones que le permite actuar como servidor OPC (Figura 9).

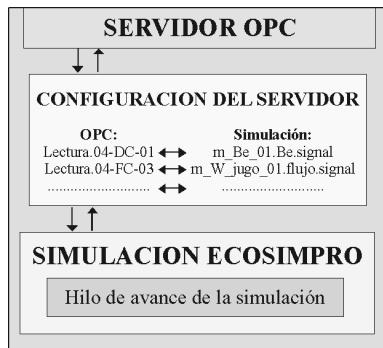


Figura 9

3. Ejemplo: Control anti-bombeo de un compresor centrífugo

3.1. Descripción del proceso

Los compresores son máquinas que tienen por finalidad aportar energía a los fluidos compresibles (gases y vapores) sobre los que operan, para hacerlos fluir aumentando al mismo tiempo su presión. Durante la compresión se engendra calor, y si no se elimina, se eleva la temperatura del fluido a medida que se va comprimiendo.

La unidad de proceso principal que se simula en este módulo consiste en un *compresor centrífugo de velocidad variable* (K-1), en el que se comprime un gas. Además, en el módulo están presentes un depósito (D-1), que recoge el gas de entrada y de la recirculación al compresor, y un cambiador líquido-gas a la salida del compresor (E-1), utilizado para refrigerar el gas de proceso. Como gas de proceso se considerará aire seco, y como líquido refrigerante agua (Figura 10).

El modelo del compresor está basado en balances de materia y energía, considera una

compresión isoentrópica y utiliza parámetros reales de compresores (Figura 11). Se escapa del alcance de esta comunicación la exposición detallada de las ecuaciones que modelan el sistema.

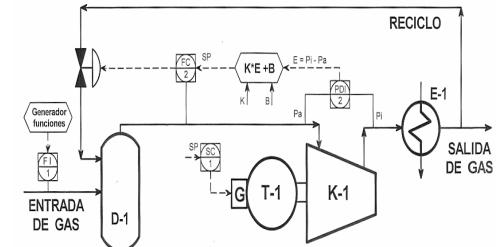


Figura 10

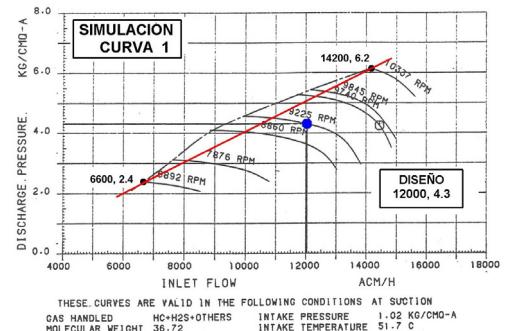


Figura 11. Curva de descarga de un compresor centrífugo real

El interés de este módulo radica en que la operación de un compresor centrífugo puede llegar a ser inestable debido a cambios en alguna condición del proceso, principalmente el caudal de gas que llega al compresor, de manera que se llegue hasta la situación no deseable de surge o bombeo límite. Para evitar esa situación se utiliza una estructura de recirculo de gas en la que se controla el caudal que pasa a través de la recirculación, de modo que trata de garantizarse un caudal mínimo a la entrada de aspiración del compresor, en función del incremento de presión dado por el compresor.

El objetivo es controlar el caudal en la aspiración del compresor, de forma que nunca descienda de un flujo mínimo. Para ello se cuenta con un controlador tipo PID que regulará

el flujo volumétrico de aspiración, en condiciones acm, por medio del recírculo. La referencia de flujo para ese regulador se calcula de modo remoto, de acuerdo al procedimiento que se detalla a continuación, aunque el alumno puede situar el regulador en local seleccionando el valor de la referencia para ese lazo.

En condiciones normales, la válvula de recirculación estará cerrada, y sólo cuando el sistema pueda entrar en situación de bombeo límite deberá abrirse la válvula de recirculación.

El cálculo de la referencia de flujo de aspiración se hace en base a la ecuación de la línea antibombeo: $SP = K \cdot E + B1 + B2$.

Siendo:

- ✓ SP: set point del caudal de gas en la aspiración del compresor, m³/h.
- ✓ K, B1 y B2. Los parámetros a configurar en el sistema de control anti-bombeo.
- ✓ E: incremento de presión proporcionado por el compresor.

3.2. Interfaz gráfica

El objetivo del módulo (Figura 12) es que el alumno sea capaz de parametrizar adecuadamente el sistema de control anti-bombeo del compresor centrífugo de acuerdo con la curva de carga del compresor (Figura 11).

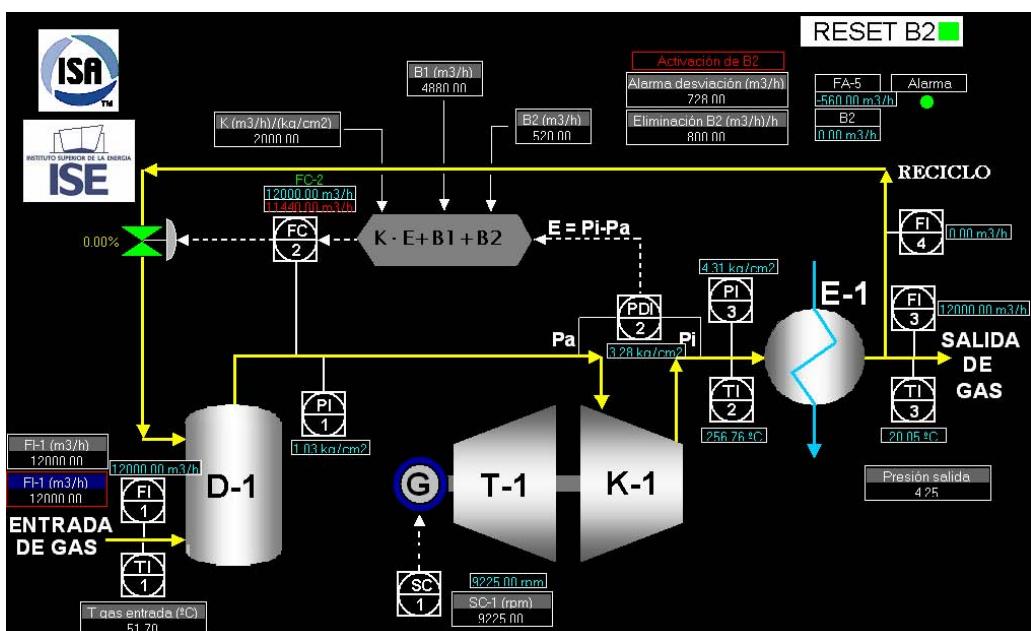


Figura 12

A modo de ejemplo, la política del módulo de control antibombeo es tal que en condiciones normales B2 será igual a 0, pero en el momento que se sobreponga el límite de la alarma del error del lazo de regulación del caudal de aspiración, B2 tomará el valor nominal especificado. También cuando se sobreponga el umbral de la alarma del error del lazo de regulación del

caudal de aspiración se activará una señal de alarma. Cuando esa señal de alarma se desactive el alumno podrá reiniciar el valor de B2 a 0, para ello deberá pulsar el botón de RESET (cuando esté en rojo). B2 se desactivará progresivamente de acuerdo a una recta en el tiempo cuya pendiente podrá elegir el alumno.

En la interfaz gráfica el alumno podrá modificar, además de los parámetros de la estructura de control, las siguientes variables de contorno:

- ✓ Flujo volumétrico, en condiciones normales, y temperatura del gas de entrada.
- ✓ Presión en la línea de salida.
- ✓ Velocidad del compresor.

Y puede visualizar todas las variables de proceso importantes.

Por ejemplo un ejercicio puede ser disminuir en exceso el caudal de gas que llega a D-1 (Figura 13), entonces el caudal de gas que pasa por el compresor disminuye. Para garantizar que por el compresor circula un flujo mínimo, el sistema de control aumenta la referencia del caudal de recirculo (Figura 15), adicionalmente como el caudal de entrada ha descendido en exceso se activa la alarma y el valor de B2 pasa de 0 a su valor nominal (Figura 14).

Cuando el sistema vuelve a una situación normal el alumno pulsa el botón de RESET (en torno al tiempo 7:00) y el valor de B2 vuelve a ser 0 de forma lineal con el tiempo y la referencia del flujo de recirculo desciende progresivamente. El caudal de gas impulsado por el compresor sigue a su referencia.

A continuación el alumno aumenta la entrada de gas al sistema (en torno al tiempo 17:00), volviendo a las condiciones iniciales y el sistema progresivamente vuelve a la situación de partida. Fíjese que en esa situación de partida la referencia de caudal de impulsión es menor al caudal que circula por el compresor, no existiendo problemas de flujo mínimo. El alumno observa que el sistema de control que ha parametrizado funciona adecuadamente y además estudia y entiende esta estructura de control particular.



Figura 13. Evolución del flujo de gas a la entrada



Figura 14. Evolución del parámetro B2.

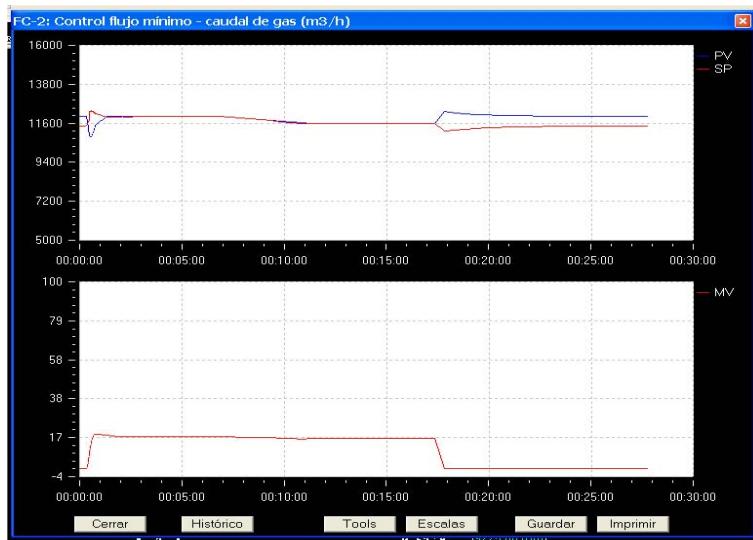


Figura 15. Evolución de la referencia de caudal de aspiración, caudal de aspiración y apertura de la válvula de regulación de flujo de reciclo.

4. Conclusiones

Se ha presentado de forma breve una librería de módulos de simulación de problemas típicos de control, sus características funcionales, estructura software y herramientas de desarrollo.

Esa librería aborda problemas de control específicos que no aparecen en otras herramientas de simulación de este tipo, por ejemplo *Control Station* [7]. Esto se ha ilustrado con el ejemplo de control de un compresor por reciclo mínimo. La particularidad de los casos desarrollados constituye el valor añadido de esta librería, ya que algunos de los módulos son casos de control normales, pero no resultan ser los ejemplos típicos de control que aparecen de forma habitual en otras herramientas.

La aplicación desarrollada se ha usado con éxito en el master de control de procesos que imparte la sección española de ISA. Siendo un desarrollo a medida y contratado por una entidad, este no puede ponerse a libre disposición en el WEB.

Consideramos que es una herramienta de fácil uso, para sistemas operativos Windows, sin necesidad de licencias adicionales, adaptada al

mundo industrial (ha sido validada por técnicos de Repsol YPF), y con pocos requerimientos hardware y software, estando abiertos a su distribución y comercialización si alguien presenta interés.

Por otro lado se dispone de las herramientas y el conocimiento necesario para ampliar la librería con otros módulos adicionales.

Referencias

- [1] ISA. <http://www.isa-spain.org>
- [2] Alves, R., Normey-Rico, J. E., Merino, A., & de Prada, C. (2003). Un SCADA vía OPC aplicado a una planta piloto. 2º Congresso Brasileiro de P&D em Petróleo & Gás. Rio de Janeiro.
- [3] OPC. <http://www.opc-foundation.org>
- [4] EcosimPro. <http://www.ecosimpro.com>
- [5] Modelica. <http://www.modelica.org>
- [6] Zamarreño J.M., Acebes L.F., Alves R., OPC-based real time simulator: architecture and practical example. Proceedings of the 41st SIMS Simulation Conference, SIMS_2000.
- [7] Control Station.
<http://www.controlstation.com>

Modelos con Incertidumbre: Identificación de parámetros intervalares y simulación usando cuantificadores

Sandra M. Castillo, Esteban R. Gelso, Joaquim Armengol

Institut d'Informatica i Aplicacions (IIIA)

Universitat de Girona

Campus Montilivi

E-17071 Girona

{sandra.castillo, esteban.gelso, joaquim.armengol}@udg.edu

Resumen

Uno de los retos de la ingeniería, es trasladar las condiciones de diseño o modelado a la práctica, lo cual es casi imposible debido a diversas fuentes de incertidumbre. Como se muestra en este trabajo, una manera especial de contemplar la incertidumbre e incluirla directamente en los modelos, es representarla por medio de intervalos. Al simular un modelo con intervalos clásicos, el resultado puede ser representado por una envolvente que compacta dentro de una misma curva todos los comportamientos posibles a partir de un estado inicial determinado. Con intervalos modales, se introducen además diferentes interpretaciones físicas, por medio del uso de cuantificadores. En este trabajo, se presenta una metodología basada en el análisis de consistencia con intervalos modales, que introduce la semántica de detección de fallos para realizar la estimación de los parámetros del modelo. La metodología propuesta es usada para la estimación de los parámetros de un modelo de crecimiento demográfico y de un modelo de una planta hidráulica. El comportamiento normal del segundo modelo estudiado, es simulado por medio de una herramienta para la detección robusta de fallos.

1. Introducción

Un *modelo* [17] es una herramienta que usamos para responder preguntas sobre el siste-

ma, sin necesidad de hacer experimentación sobre el mismo, debido a que puede ser muy costoso, destructivo, peligroso o simplemente porque el sistema aún no existe (en el caso de pruebas de diseño). Por modelo matemático se entiende que las relaciones entre variables (distancias, corrientes, flujos, desempleo, etc.) que son observadas en el sistema son descritas como relaciones matemáticas en el modelo.

Se pueden distinguir dos áreas de conocimiento relacionadas con la construcción de modelos. La primera es el *dominio de la experiencia*, relacionada con el entendimiento de la aplicación y el dominio de todos los hechos relevantes para el modelo. La segunda, es el *conocimiento del ingeniero*, quien tiene que poner el conocimiento de los expertos dentro de un modelo práctico, útil y explícito. Si se posee un modelo del sistema, este se puede usar para calcular o predecir cómo habría reaccionado el sistema, solucionando matemáticamente las ecuaciones que describen el sistema y estudiando las respuestas. Este tipo de experimentación generalmente es llamado simulación. Sin embargo, el valor de los resultados de dicha experimentación, depende completamente de la calidad del modelo del sistema.

La dificultad al construir modelos es hacer que sean buenos y confiables. Para que un modelo sea útil, se debe tener confianza en los resultados y predicciones que se pueden inferir de él. Esta *confiabilidad* puede ser obtenida por medio de verificar o validar el modelo,

generalmente, comparando el comportamiento del modelo con el comportamiento del sistema y evaluando la diferencia.

Los modelos matemáticos pueden tener características diferentes dependiendo de las propiedades del sistema al que representan y de las herramientas usadas para construirlos. Algunas clasificaciones que se pueden encontrar en la literatura son: Dinámicos o Estáticos, Discretos o Continuos, Concentrados o Distribuidos, Orientados al cambio o De eventos discretos, Deterministas o Estocásticos.

La *incertidumbre* es un factor importante a tener en cuenta cuando se habla de la *confiabilidad* del modelo. En la práctica, una descripción matemática del proceso que sea exacta y completa, nunca está disponible. Algunas veces la estructura matemática del sistema no se conoce por completo (o se conoce, pero se simplifica), también puede suceder que los parámetros del sistema no pueden ser conocidos por completo, o solo se los conoce en un rango limitado de operación de la planta. Por lo tanto siempre hay una diferencia entre la realidad y el modelo usado. Otra de las suposiciones es considerar como conocidas las características (media, varianza, etc.) de las perturbaciones y señales de ruido que actúan en el sistema, situación que tampoco es cierta en la práctica. A medida que la complejidad de un sistema aumenta, más difícil es la tarea de modelar el sistema y sus perturbaciones. Se puede hablar de un modelo con incertidumbre, cuando hay incertidumbre en el conocimiento de la estructura del sistema, parámetros y efectos de las perturbaciones.

La incertidumbre en los parámetros del modelo del sistema puede aparecer por una o varias de las siguientes razones [20]:

- Los valores nominales de los parámetros del sistema son el resultado de los pasos de diseño del mismo, pero los valores actuales son a menudo diferentes de los nominales.
- Parámetros cuyos valores fueron estimados mediante intervalos de confianza, en los cuales está implícito determinado error de estimación y nivel de confianza.

- Los valores de los parámetros reales pueden variar en el tiempo a causa de no linearidades, derivas, etc.
- Modelos simplificados o de orden reducido son usados para modelar sistemas complejos.

Los métodos garantistas obtienen una cota exterior de los parámetros del sistema que son consistentes con el error y medidas consideradas, alteradas por un error acotado. Estos métodos han sido aplicados para manejar la incertidumbre en diferentes problemas de ingeniería, uno de ellos es la identificación de sistemas. Dentro de los métodos garantistas, se puede encontrar: elipsoides, poliedros, intervalos clásicos [18], intervalos modales [8], y zonotopes [6], entre otros. Son diversos los problemas de identificación de modelos que se han solucionado empleando métodos garantistas, como antecedentes de este trabajo son de especial interés los que consideran la incertidumbre en los modelos por medio de intervalos (sección 2). Los Modelos Intervalares [15], son modelos en los que se supone que los errores de las entradas, las salidas, el estado y los parámetros son la suma de muchos componentes independientes y pequeños, por lo que el conjunto de posibles valores que toman son intervalos cerrados. Estos modelos ofrecen una manera especial de contemplar la incertidumbre e incluirla directamente en el diseño de modelos, representándola por medio de intervalos.

Cuando se trabaja con intervalos clásicos, un modelo intervalar permite representar un conjunto de modelos. Al simular un modelo intervalar, para cada variable de salida, el resultado puede ser representado por una envolvente que compacta dentro de una misma curva todos los comportamientos posibles a partir de un estado inicial determinado. De esta manera, la envolvente incluye toda una familia de curvas temporales. Cuando se trabaja con intervalos modales, se introducen diferentes interpretaciones físicas, por medio del uso de cuantificadores universales o existenciales para caracterizar los intervalos que representan las entradas, salidas y parámetros del modelo.

En este paper, se presenta una metodología basada en el análisis de consistencia con intervalos modales, que introduce la semántica de detección de fallos para realizar la estimación de los parámetros del modelo (sección 3). La metodología propuesta es usada para la estimación de los parámetros de un modelo de crecimiento demográfico y del modelo de una planta hidráulica compuesta por dos tanques acoplados (sección 4). La simulación del comportamiento normal del segundo modelo estudiado, se realiza por medio de una herramienta para la detección robusta de fallos (sección 5). Finalmente conclusiones y trabajo futuro son presentados en la sección 6.

2. Estimación e intervalos

La idea básica de describir la incertidumbre de las plantas de forma acotada, es tan antigua, como el diseño con tolerancia en ingeniería [18]. Los métodos intervalares son introducidos en el contexto de estimación de parámetros por [12], e independientemente en 1992 por Moore [19] y en 1993 por Jaulin y Walter [13].

Estos métodos cobran importancia debido a que hay varias razones para rechazar la caracterización del ruido como un proceso aleatorio con distribución normal. Una de ellas es que bajo este supuesto se obtienen resultados paradójicos, cuya interpretación sería que nada puede ser estimado con una exactitud establecida arbitrariamente. Pero, si se supone que no se desea asumir nada más que $|e(t)| \leq C_v$, entonces el problema de estimación de parámetros es encontrar aquellos modelos que son consistentes con esta descripción del error. Lo que es equivalente a asumir que $e(t)$ corresponde a una distribución uniforme o rectangular con $|e(t)| \leq C_v$ [16]. A continuación se presentan algunos métodos de estimación de parámetros usando intervalos clásicos e intervalos modales, tomados como referencia para el desarrollo de este trabajo.

2.1. SIVIA: Inversión de conjuntos

El algoritmo SIVIA (Set Inversion Via Interval Analysis) [14] es un algoritmo intervalar de

branch-and-bound dedicado, en su versión original, a la aproximación de conjuntos solución de la forma:

$$\Sigma = \{x \in X \mid f(x) \leq 0\}, \quad (1)$$

donde f es una función real continua de \mathbb{R}^n a \mathbb{R}^m y X la caja a evaluar.

2.2. QSI: Inversión de conjuntos con intervalos modales

En [10, 11] se presenta un algoritmo basado en técnicas de branch-and-bound y el Análisis Intervalar Modal [21], una teoría que extiende la Aritmética Intervalar Clásica. Este algoritmo inspirado en el algoritmo SIVIA (Set Inversion Via Interval Analysis), es llamado QSI (Quantified Set Inversion). QSI obtiene aproximaciones internas y externas de un conjunto solución de la clase de Restricciones Reales Cuantificadas (quantified real constraints -QRCs). Las QRCs, son empleadas para el planteamiento de problemas que se desean resolver empleando MIA. Donde, un intervalo modal es definido como un par formado por un intervalo conjuntista clásico, A' , y al que se le llama *extensión*, y por un cuantificador Q_A , *modalidad*, que puede ser \forall o \exists . La definición mas general del conjunto solución obtenido empleando QSI, es:

$$\Sigma = \{x \in X \mid (\forall u \in U)(\exists v \in V)f(x, u, v) = 0\}, \quad (2)$$

donde f es una función real de \mathbb{R}^n a \mathbb{R} , U hace referencia a las variables cuantificadas de manera universal y V a las cuantificadas como existenciales. QSI se basa en un algoritmo de branch-and-bound y las dos reglas:

- Determinar si X (cualquier caja resultante del proceso de bisección), está contenida en Σ :

$$(\forall x \in X)(\forall u \in U)(\exists v \in V)f(x, u, v) = 0 \Leftrightarrow X \subseteq \Sigma \quad (3)$$

- Determinar si X no se intersecta con Σ :

$$(\forall x \in X)(\exists u \in U)(\forall v \in V)f(x, u, v) \neq 0 \Leftrightarrow X \cap \Sigma = \emptyset \quad (4)$$

La estimación de parámetros, acotando el error por medio de intervalos modales, es una de las aplicaciones que se le puede dar al algoritmo QSI. Este problema se traduce a encontrar una aproximación interna y otra externa del conjunto solución de parámetros para un

modelo dado, de manera que este sea consistente con las observaciones (medidas) de entradas y salidas.

2.3. Estimación de Parámetros vía uso de cuantificadores

Hasta ahora, se han presentado dos métodos para obtener aproximaciones de la región consistente de parámetros, en las cuales el resultado final es una lista de cajas o pavimentos, caracterizados según correspondan a la región interna de la región de parámetros, a la región externa, y aquellas que se consideran indeterminadas, es decir, que no pueden ser caracterizadas exclusivamente como correspondientes a la región interna o externa de la región de parámetros. Sin embargo, esta no es la única manera de obtener una región de parámetros consistente con un conjunto de mediciones, empleando intervalos.

En el ámbito de los intervalos modales [8, 7], se introduce la obtención de aproximaciones internas de la región de parámetros, empleando diversas combinaciones de los cuantificadores aunados a los intervalos que representan entradas, salidas y parámetros del modelo. Donde, cada una de las combinaciones de los cuantificadores, brinda una semántica y por consiguiente una interpretación física diferente a la región obtenida. Esta aproximación es de tipo intervalar y se puede obtener empleado una librería de optimización, como la GRG2, empleada en los trabajos citados anteriormente.

3. Estimación de Parámetros para la detección robusta de fallos

El trabajo propuesto en este documento, presenta una metodología para realizar estimación paramétrica empleando Análisis Intervalar Modal (MIA - Modal interval Analysis), con el fin de obtener modelos intervalares, empleados posteriormente para la detección y el diagnóstico de fallos usando la herramienta robusta de detección, SQualTrack (descrita en [2, 3, 1]).

Generalmente, la caracterización de la incertidumbre de los parámetros del modelo por medio de intervalos, requerida para el uso de la herramienta de detección SQualTrack, se encontraba asociada al nivel de conocimiento del proceso por parte de los expertos o en algunos casos, a la tolerancia respecto al valor nominal de los mismos, cuando los parámetros tienen significado físico. Para mantener la filosofía de evitar falsas alarmas de detección, se debe encontrar una aproximación externa de la región de parámetros. Para ello, se han estudiado las diferentes combinaciones de cuantificadores y se ha encontrado, que el conjunto solución en este caso, está caracterizado por la ecuación 5:

$$\Sigma = \{x \in X \mid (\exists u_0 \in \tilde{U}_0), (\exists y_0 \in \tilde{Y}_0) f(x, u_0) - y_0 = 0 \\ \vdots \\ (\exists u_k \in \tilde{U}_k), (\exists y_k \in \tilde{Y}_k) f(x, u_k) - y_k = 0\} \quad (5)$$

Para obtener esta aproximación externa de la región de parámetros, se propone una nueva metodología basada en la segunda regla del algoritmo QSI presentada en la ecuación 4.

La metodología de estimación está implementada en un algoritmo de branch-and-prune con dos partes, la primera que se encarga de descartar capas externas de la región inicial de parámetros cuando no se verifica su consistencia. La segunda parte del algoritmo que subdivide la región resultante de la rutina anterior y verifica la consistencia de las diferentes cajas. Como resultado se obtiene una lista de cajas según sea su consistencia con el conjunto de entrenamiento: (i) externas o no consistentes (ii) indeterminadas, que no pudieron descartarse como externas o que pueden estar contenidas en su totalidad o en parte en la región consistente de parámetros. La aproximación externa, requerida para la detección robusta de fallos, se obtiene por medio del join de las cajas indeterminadas. Esta metodología propuesta, es también la base para la implementación de un test de detección de fallos, basado en el refinamiento del espacio de parámetros. En la sección 4, se presentan algunos resultados de estimación de parámetros.

4. Aplicaciones

4.1. Modelo de crecimiento demográfico

Para ilustrar la metodología de estimación presentada en la sección 3, se emplea un modelo sencillo de crecimiento demográfico, presentando en [9]. Los datos experimentales considerados, corresponden con el censo de la población de Estados Unidos, durante un periodo de 120 años, comprendido entre 1790 y 1910, con periodo de muestreo correspondiente a 10 años. La ecuación 6, representa la estructura del modelo.

$$x(t) = \frac{kx_o e^{rt}}{x_o(e^{rt} - 1) + k} \quad (6)$$

Los datos experimentales presentados en la tabla 1, fueron escalados dentro del intervalo $[0, 100]$ y afectados por un error de ± 1 .

t	x	t	x
0	3.929	70	31.43
10	5.308	80	39.81
20	7.239	90	50.15
30	9.638	100	62.94
40	12.86	110	75.99
50	17.06	120	91.97
60	23.19		

Tabla 1: Población de EEUU durante 120 años (1790-1910)

El objetivo de este ejemplo es ajustar los parámetros k , r y x_o , encontrando la aproximación externa requerida para la detección de fallos, caracterizada por la ecuación 5. Para esta tarea, se emplearon QSI y la nueva metodología propuesta. Los resultados obtenidos se presentan en las figuras 1 y 2. Estos resultados son comparados en la tabla 2.

Debido a la alta cantidad de subdivisiones que QSI debe realizar para hallar la región consistente de parámetros, la cantidad de cajas indeterminadas encontradas es 3763, mientras que con la nueva metodología es 44. Este es un indicador del costo computacional que implica obtener esta aproximación empleando QSI, que además hace prácticamente inviable usar

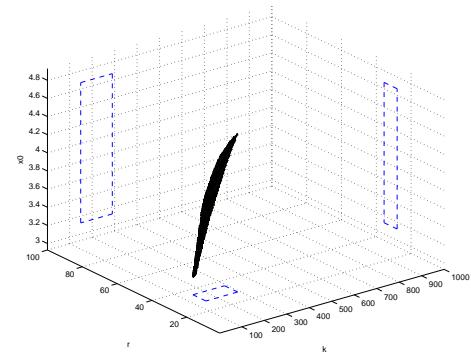


Figura 1: Región consistente de parámetros empleando QSI

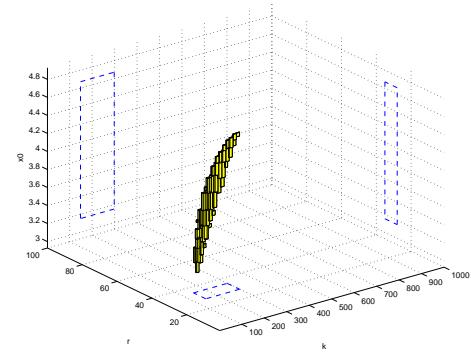


Figura 2: Aproximación externa de la región consistente de parámetros empleando la nueva metodología

	QSI	Nueva Metodología
k	[149.200, 289.800]	[148.139, 298.251]
r	[28.050, 35.550]	[28.025, 35.0716]
x_o	[3.124, 4.671]	[3.152, 4.6709]

Tabla 2: Comparación de resultados obtenidos para el modelo de crecimiento demográfico

este algoritmo para un número mayor de parámetros. Con la nueva metodología, al incluir el algoritmo de branch-and-prune, esta complejidad se reduce, logrando resultados similares, con un costo mucho más bajo.

4.2. Modelo de una planta hidráulica compuesta por dos tanques acoplados

Un ejemplo de modelo dinámico no-lineal, muy conocido, es el de una planta hidráulica compuesta por dos tanques acoplados (ver Figura 3). El sistema está compuesto por dos tanques, T1 y T2, una válvula, V1, y un controlador PI1, el cual recibe como entrada el nivel de agua en T2 y controla la válvula que a su vez, regula el flujo de agua en T1.

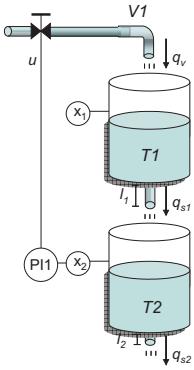


Figura 3: Diagrama de un sistema de dos tanques acoplados.

Para efectos de detección de fallos basada en modelos, se pueden obtener dos ecuaciones de redundancia analítica a partir del modelo del sistema. En este trabajo, se estiman los parámetros de una de ellas, correspondiente a la ecuación 7. Donde las variables medidas son \tilde{x}_1 que representa el nivel de agua contenido en T1 y \tilde{u} que es la señal de salida del controlador. k , k_{s1} , S_1 y l_1 son los parámetros del sistema.

$$\begin{aligned} r_1 = & \tilde{x}_1(k) - \tilde{x}_1(k-1) + \frac{T_s}{S_1} \left(-k \tilde{u}(k-1)^3 + \right. \\ & \left. + k_{s1} \sqrt{\tilde{x}_1(k-1) + l_1} \right) \end{aligned} \quad (7)$$

Los parámetros fueron estimados a partir de 200 muestras en un escenario de funcionamiento normal, sin fallos. La región inicial de parámetros considerada fue: $k = [0.1, 10]$, $k_{s1} = [0.5, 5]$ y $l_1 = [71.0, 71.5]$. Con la nueva metodología, se encontraron los siguientes resultados:

$k = [0.17251, 1.13334]$, $k_{s1} = [0.5, 2.10212]$ y $l_1 = [71.0, 71.5]$.

5. Simulación del comportamiento normal del sistema

SQualTrack (Semi-Qualitative Tracking) es una herramienta de detección de fallos desarrollada por miembros del grupo de investigación Intervalos Modales e Ingeniería de Control (MICE) de la Universitat de Girona. Está basada en el Análisis Intervalar Modal (MIA, Modal Interval Analysis). Como se mencionó anteriormente, mediante los intervalos se tienen en cuenta las incertidumbres asociadas a la medición de las variables conocidas y a los parámetros del modelo. Debido a que genera umbrales adaptativos, podría clasificarse como una herramienta con robustez pasiva. En SQualTrack se usan modelos discretos y la detección está basada en la redundancia analítica, las envolventes con error acotado [4] y en múltiples ventanas temporales deslizantes [5].

Un fallo es detectado cuando $\tilde{Y}(k) \cap \hat{Y}_{ex}(k) = \emptyset$, siendo \tilde{Y} la medición intervalar, \hat{Y}_{ex} la aproximación externa. Asimismo, la aproximación interna permite detener el algoritmo de branch-and-bound encargado de obtener mejores aproximaciones (externa e interna) de la envolvente exacta, cuando $\tilde{Y}(k) \cap \hat{Y}_{in}(k) \neq \emptyset$, siendo \hat{Y}_{in} la aproximación interna. Las tres zonas definidas por las estimaciones de error acotado (envolventes interna y externa) están representadas en la Figura 4.

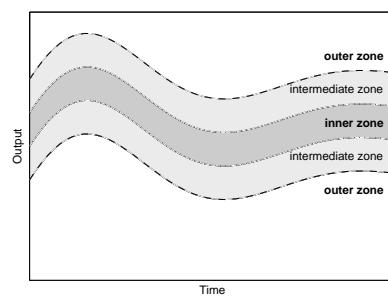


Figura 4: Las tres zonas definidas por las envolventes en SQualTrack

Dado que para la estimación de parámetros realizada en la sección 4 se introdujo la semántica correspondiente a la detección de fallos, se empleará SQualTrack para simular el comportamiento normal (en ausencia de fallos) del sistema de los dos tanques. Al realizar esta prueba, se verifica la consistencia del modelo con la semántica de detección de fallos, debido a que no se generan falsas alarmas al simular este escenario, como se puede observar en la figura 5

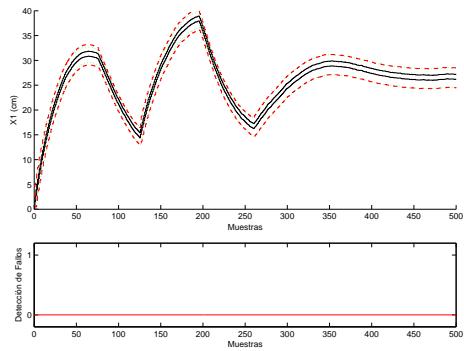


Figura 5: Comportamiento normal del sistema de dos tanques

6. Conclusiones y trabajo futuro

El trabajo presentado es un ejemplo más, de que dependiendo del objetivo que se persiga, se pueden obtener diversos modelos de un mismo sistema. En este caso en particular, se requiere la obtención de modelos intervalares que incluyan la semántica de la detección robusta de fallos.

El uso de intervalos permite el manejo de la incertidumbre, y en el caso de intervalos modales, el uso de cuantificadores permite dar interpretabilidad a los resultados obtenidos. La metodología planteada en la sección 3, permite obtener una aproximación externa de la región de parámetros consistentes con el modelo de funcionamiento normal del sistema, de esta manera se evita en la detección de fallos la generación de falsas alarmas, característica de esta herramienta de detección SQualTrack.

Está propuesta, constituye un eslabón más, para lograr una única herramienta de sopor-

te a la supervisión basada en MIA, que involucra desde la etapa de obtención del modelo del sistema, hasta la diagnosis del mismo. Además, constituye el primer paso para implementar una vía de diagnosis alternativa, mediante el refinamiento on-line del espacio de parámetros.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Fondo Social Europeo de la Unión Europea, por el Gobierno Español, a través de los proyectos coordinados de investigación No. DPI2006-15476-C02-02 y No. DPI2005-08668-C03-02, y la subvención No. 2005SGR00296 y por el Departament d'Innovació, Universitats, i Empresa del Gobierno de Cataluña.

Referencias

- [1] J. Armengol, P. Herrero, and M. Á. Sainz. *El análisis de intervalos en España: desarrollos, herramientas y aplicaciones*, chapter Detección de fallos en procesos reales (Proyecto CHEM), pages 229–238. 2005.
- [2] J. Armengol, J. Vehí, M. Á. Sainz, and P. Herrero. Application of interval models to the detection of faults in industrial processes. In *World Automation Congress*, volume 4, 2004.
- [3] J. Armengol, J. Vehí, M. Á. Sainz, and P. Herrero. Industrial application of a fault detection tool based on interval models. In *International Conference on Integrated Modeling and Analysis in Applied Control and Automation 2004 (IMAACA 2004)*, *International Mediterranean Modeling Multiconference (IMM 2004)*, volume 2, pages 150–157, Bergeggi, Génova, 2004.
- [4] J. Armengol, J. Vehí, L. Travé-Massuyès, and M. Á. Sainz. Application of modal intervals to the generation of error-bounded envelopes. *Reliable Computing*, 7(2):171–185, April 2001.

- [5] J. Armengol, J. Vehí, L. Travé-Massuyès, and M. Á. Sainz. Application of multiple sliding time windows to fault detection based on interval models. In Sh. McIlraith and D. Theseider Dupré, editors, *12th International Workshop on Principles of Diagnosis DX 2001. San Sicario, Italy*, pages 9–16, 2001.
- [6] J.M. Bravo, T. Alamo, and F. Camacho. Bounded error identification of systems with time-varying parameters. *IEEE Transactions on Automatic Control*, 51(7):1144–1150, July 2006.
- [7] R. Calm, M. Sainz, P. Herrero, J. Vehí, and J. Armengol. Parameter identification with quantifiers. In *5th IFAC Symposium on Robust Control Design (ROCOND06)*, 2006.
- [8] Remei Calm. *Análisis Intervalar Modal: Su construcción teórica, implementación y posibilidades de aplicación a la simulación y al control*. Phd thesis, Universitat Politècnica de Catalunya, Barcelona, 2005.
- [9] J. Cruz and P. Barahona. Maintaining global-hull consistency with local search for continuous csp's. In *Selected Papers, LNCS. First International Workshop on Global Constrained Optimization and Constraint Satisfaction (Cocos'02)*, pages 178–193, 2003.
- [10] P. Herrero, L. Jaulin, M. Sainz, and J. Vehí. Quantified set inversion algorithm. In *2nd International Workshop on Interval Mathematics and Constraint Programming IMCP-2004*, pages 272–279, 2004.
- [11] P. Herrero, M. Á. Sainz, J. Vehí, and Luc Jaulin. Quantified set inversion algorithm with applications to control. *Reliable Computing*, 11(5):369–382, 2005.
- [12] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.
- [13] L. Jaulin and E. Walter. Guaranteed nonlinear parameter estimation via interval computations. *Interval Computations*, 3:61–75, 1993.
- [14] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
- [15] V. Krenovich. Why intervals? a simple limit theorem that is similar to limit theorems from statistics. *Reliable Computing*, 1(1):30–40, 1995.
- [16] Lennart Ljung. Issues in system identification. *IEEE Control Systems*, pages 25–29, January 1991.
- [17] Lennart Ljung and Torkel Glad. *Modeling of Dynamic Systems*. Prentice Hall, 1994.
- [18] Mario Milanese, John Norton, Hélène Piet-Lahanier, and Éric Walter. *Bounding approaches to system identification*. Plenum Publishing Corporation, New York, 1996.
- [19] R. E. Moore. Parameter sets for bounded-error data. *Mathematics and Computers in Simulation*, 34(2):113–119, 1992.
- [20] V. Puig, A. Stancu, T. Escobet, F. Ñejjar, J. Quevedo, and R. J. Patton. Passive robust fault detection using interval observers: Application to the damadics benchmark problem. *Control Engineering Practice*, 14(6):621–633, 2006.
- [21] SIGLA/X. *Applications of Interval Analysis to Systems and Control. Proceedings of MISC 1999*, chapter Modal Intervals, pages 157–227. Universitat de Girona, 1999.

Sistema Reconfigurable para la Simulación en Bloque de Experimentos Útiles en Tareas de Soporte para Supervisión y Diagnosis

Anibal Bregón¹, Diego García², Oscar Prieto¹, Belarmino Pulido¹, Carlos Alonso¹

¹ Departamento de Informática

ETS Ingeniería Informática

Universidad de Valladolid

(anibal,belar,calonso,oscapri@infor.uva.es)

² Departamento de Ingeniería de Sistemas y Automática

Universidad de Valladolid

dieggar@cta.uva.es

Resumen

La aparición de fallos en sistemas industriales continuos puede tener consecuencias desastrosas. Con el objetivo de reducir estas consecuencias, los sistemas deben ser capaces de detectar e identificar esos fallos. El desempeño de estas tareas se puede facilitar usando ejemplos cuantitativos del comportamiento de los sistemas en presencia de fallo. Las herramientas de modelado y simulación resultan muy útiles para esta tarea.

En este artículo se describe un sistema para llevar a cabo simulaciones de experimentos en bloque. El sistema usa modelos matemáticos escritos en EcosimPro. Los resultados se almacenan en una base de datos, y el intercambio de información se realiza mediante ficheros XML. Debido al uso de la base de datos y de los ficheros XML el sistema es completamente reconfigurable. El sistema es capaz de trabajar independientemente del modelo.

1. Introducción

Nuestra vida diaria depende enormemente de la tecnología, desde el gran número de procesos tecnológicos complejos hasta los pequeños dispositivos de uso diario. El empleo de sistemas de control en dichos procesos o dispositivos se ha visto incrementado exponencialmente durante las últimas décadas. A pesar de que resulta bastante claro que el uso de estos sistemas de control ha liberado al operador humano de la tediosa tarea del control manual, no ha sido capaz de inmunizar a estos sistemas frente a fallos [1, 7] (consideraremos un fallo como “una desviación de la estructura o de los parámetros del sistema de su situación nominal” [1]).

Cuando aparece un fallo durante el funcionamiento de un sistema, las consecuencias del fallo pueden ser desastrosas (pérdida de calidad en un producto, daños humanos o de maquinaria, etc...) y tienen que ser evitadas tan rápidamente como sea posible. El sistema tiene que detectar el fallo y entonces intentar minimizar su propagación y/o consecuencias. Esta es la razón por que la que se pretende que los

sistemas sean tolerantes a fallos [7].

De acuerdo con Blanke et al. [1] la manera de hacer que un sistema sea tolerante se realiza mediante dos pasos:

- **Diagnóstico de fallo:** La ocurrencia de un fallo tiene que ser detectada y el fallo tiene que ser identificado.
- **Re-diseño del controlador:** El controlador tiene que adaptarse a la situación con fallo para que el sistema global continúe cumpliendo con sus objetivos.

Con el fin de poder llevar a cabo estas tareas, resulta muy interesante conocer cómo falla el sistema y cuáles son los efectos de un fallo desde un punto de vista cuantitativo. El problema aquí radica en que en determinado tipo de sistemas continuos, es imposible, peligroso o realmente difícil obtener estos ejemplos cuantitativos de fallos. Sin embargo, en la mayoría de los casos podemos obtener modelos matemáticos del comportamiento de los sistemas. A partir de estos modelos podemos usar herramientas de simulación y reproducir su comportamiento.

Dentro del contexto del diagnóstico de fallos, resulta muy útil proporcionar módulos de diagnóstico de fallo que utilizan un gran número de simulaciones de funcionamiento en diferentes condiciones de trabajo. Esto también es útil a la hora de realizar Análisis de los Efectos de Modos de Fallo (FMEA) [5]. El problema es que algunas veces resulta complicado, en las herramientas de simulación, realizar y almacenar automáticamente un gran número de simulaciones en bloque. En este artículo se describe un sistema que realiza y almacena automáticamente múltiples simulaciones fuera de línea de fallos que pueden ocurrir en una planta.

El artículo está organizado de la siguiente manera. Primero, describiremos en detalle el sistema y su arquitectura. Segundo, presentaremos la herramienta de simulación usada en este sistema. Después, explicaremos las características distintivas de la herramienta de simulaciones en bloque y como puede usarse para ayudar en la tarea de diagnosis. Finalmente, discutiremos los resultados y veremos las conclusiones finales.

2. Arquitectura del Sistema

La arquitectura del sistema se muestra en la figura 1. Como se ve, el sistema está formado por cuatro módulos bien diferenciados.

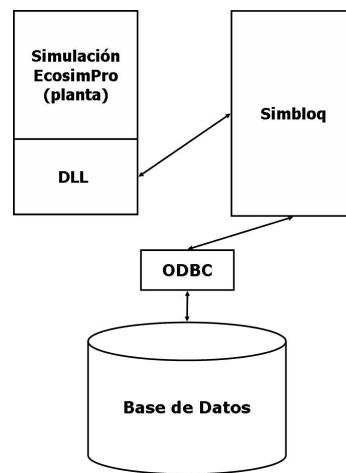


Figura 1: Arquitectura del sistema.

La parte más importante de todas es la llamada **Simbloq** [3]. Esta parte maneja los distintos parámetros y condiciones para cada simulación en bloque. Esta parte también integra las comunicaciones con el resto de las partes del sistema.

La parte llamada **Simulación EcosimPro** abstrae el modelo de una planta, y se encarga de realizar una simulación de acuerdo a los parámetros seleccionados en Simbloq, que se comunican mediante una DLL asociada a un experimento genérico de EcosimPro.

ODBC realiza las tareas de comunicación entre Simbloq y la base de datos.

Finalmente, la **base de datos** almacena toda la información de las simulaciones. Por una parte almacena los datos de salida, pero también almacena los parámetros y las condiciones de cada simulación.

En el resto del artículo describiremos el funcionamiento de cada una de estas partes.

3. Herramienta de Simulación EcosimPro

Como se ha explicado previamente (figura 1), el sistema usa el entorno de simulación EcosimPro, se utilizan experimentos relativos a los modelos matemáticos de la planta que se diseñaron previamente en el lenguaje que la plataforma proporciona para ello.

EcosimPro proporciona un lenguaje de simulación orientado a objeto; este lenguaje permite crear modelos basados en primeros principios y conocimiento, entre otros, mediante la formulación de ecuaciones algebraicas y ecuaciones diferenciales relativa a las leyes de conservación fisicoquímicas.

Una vez que se ha creado un modelo EcosimPro éste puede compilarse y se pueden generar cuantas particiones se desee. Una partición es el modelo matemático de cálculo, y en ella se especifican cuáles son las variables conocidas, así como las variables seleccionadas para romper lazos algebraicos si existieran. Correspondientes a cada partición, pueden generarse distintos experimentos, particularizando los valores de las condiciones iniciales y los datos, y definiendo los parámetros de la simulación. Cuando estos experimentos son compilados se genera una biblioteca dinámica (dll) que encapsula el modelo de la planta y que puede ser usado por herramientas externas, mediante una útil interfaz de funciones, para ejecutar tantas simulaciones como se desee. El sistema que este artículo presenta utiliza estos archivos para llevar a cabo las simulaciones.

A pesar de todo esto, EcosimPro tiene algunas limitaciones, cómo configurar y llevar a cabo automáticamente un gran número de simulaciones en bloque y posteriormente almacenar sus resultados para su uso futuro, por ejemplo en una base de datos. Además, cierta información relativa a los modelos no se representa en el lenguaje de EcosimPro y tiene que ser almacenada externamente.

4. Especificación de la Base de Datos

Hemos optado por una base de datos para solucionar ambos problemas de almacenamiento. Esta resulta esencial por dos razones: almacenar la información no incluida en el modelo, y almacenar los resultados de las simulaciones.

Hay cierto tipo de información que no viene encapsulada en el modelo matemático de la planta. Esta información se corresponde con la situación de la planta (identificación, a alto nivel, de las condiciones de funcionamiento de la planta), el protocolo (criterios para la configuración y el manejo de los procesos) y los modos de fallo (descripción de cómo puede fallar el sistema). La base de datos tiene que almacenar esta información con objeto de ser usada durante las simulaciones.

Además, toda la información generada por el proceso de simulación se almacena en la base de datos. El diagrama entidad-relación de la base de datos se puede ver en la figura 2, y viene descrito en [6]. La gran complejidad en el diseño de la base de datos se corresponde con el hecho de que independiente del modelo de la planta que se quiera utilizar, la estructura de la base de datos se mantiene invariable. Cada planta tiene su propia base de datos para almacenar su estructura y experimentos, pero la estructura global de tablas y claves primarias no se ve modificada.

5. Simbloq

El elemento principal del sistema de simulaciones en bloque es la herramienta llamada *SimBloq*. Esta herramienta no sólo interactúa con resto de partes del sistema sino que también dirige la comunicación con el usuario. Es esta herramienta la que permite configurar y simular los diferentes experimentos y modos de fallo que pueden producirse en la planta. La simulación es ejecutada por los experimentos de EcosimPro y almacenada en la base de datos. SimBloq también permite visualizar gráficamente la evolución temporal de las variables de los experimentos almacenados en dicha ba-

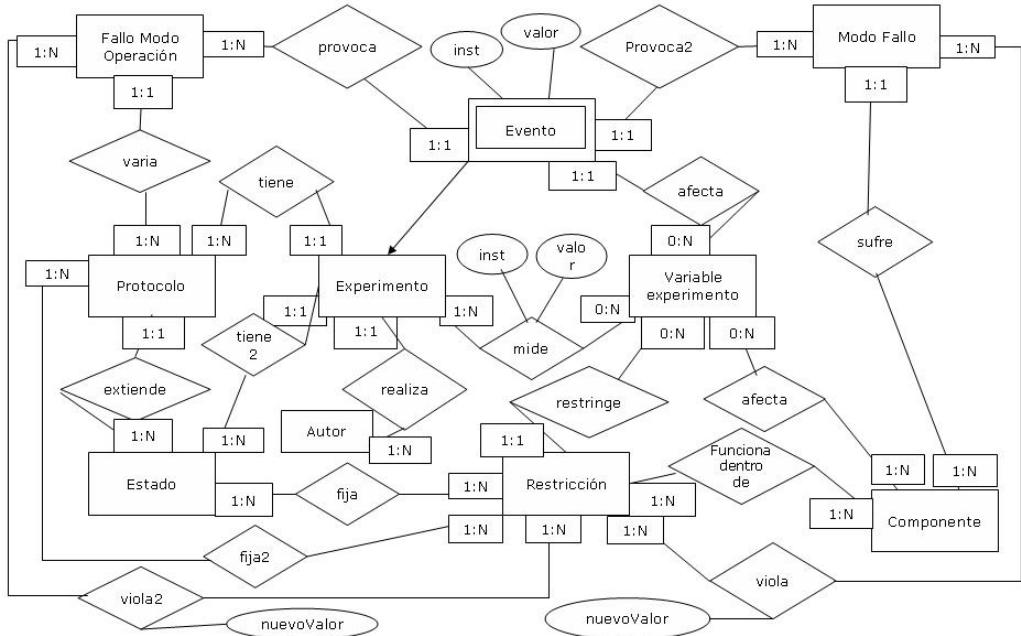


Figura 2: Esquema entidad-relación de la base de datos.

se de datos.

Una de las principales características de SimBloq es que es completamente reconfigurable. SimBloq es independiente respecto al modelo matemático usado. Esta independencia se consigue gracias al uso de la base de datos y de ficheros de configuración XML intermedios. La base de datos almacena la información no incluida en el modelo. El fichero XML almacena los parámetros y condiciones de simulación de acuerdo a una configuración fijada (ver apéndice para consultar el DTD relativo a este XML). Además Simbloq es casi completamente independiente respecto a la plataforma de simulación utilizada. El sistema ha sido diseñado para trabajar con EcosimPro, pero en su diseño se ha encapsulado en módulos aislados la funcionalidad relativa a la simulación y a la comunicación, posibilitando cambiar la herramienta de simulación modificando solamente estos módulos.

El funcionamiento de SimBloq puede dividirse en cuatro pasos: inserción en la base de

datos de la información estática relativa a la planta (realizado off-line), configuración de las simulaciones en bloque, ejecución y almacenamiento de las simulaciones, y visualización de los resultados. Cada uno de estos pasos se explican a continuación.

5.1. Inserción en la base de datos de la información estática

Este paso se desarrolla fuera de línea. Antes de configurar ningún experimento o ejecutar ninguna simulación, el usuario debe insertar información estática relativa a la planta en la base de datos. Dicha información estática complementa al modelo de la planta. Cuando éste es diseñado, el usuario tiene que especificar algunas características que EcosimPro no puede abstraer e insertarlas en la base de datos. Dichas características son las relativas al estado de la planta, al protocolo de operación, estados iniciales, variables de contorno y variables medidas.

5.2. Configuración de las simulaciones en bloque

Antes de que los experimentos puedan ser ejecutados, el usuario debe poder seleccionar una serie parámetros y condiciones. SimBloq permite al usuario configurar estos parámetros de una manera interactiva. Toda esta información configurable puede proceder de la base de datos o del modelo de la planta. Los parámetros que el usuario puede configurar son los siguientes:

- Modelo (dll) de la planta a utilizar (figura 3¹).
- Base de datos donde está la información complementaria de la planta y donde se almacenarán los resultados de los experimentos (figura 3).
- Estado de la planta y protocolo de operación.
- Tiempo de simulación.
- Intervalo de comunicación.
- Número de experimentos a simular y tipo de valores de los parámetros (aleatorios o fijados).
- Variables de salida del sistema (procedentes de los sensores).
- Condiciones iniciales de los parámetros de la planta (figura 4).
- Finalmente, el conjunto de fallos que pueden ocurrir en la planta (figura 5). Simbloq permite introducir fallos en los componentes de la planta. Para cada fallo se puede especificar el instante de tiempo en el que ocurrirá y su cuantía. Esto se consigue mediante programación de eventos discretos en EcosimPro.

¹Los resultados mostrados en esta figura y en las siguientes se han obtenido utilizando el modelo matemático de una planta de laboratorio. Este planta está compuesta de cuatro tanques, diversas bombas, controladores PID,... El lector puede encontrar una descripción detallada de la planta en [2]

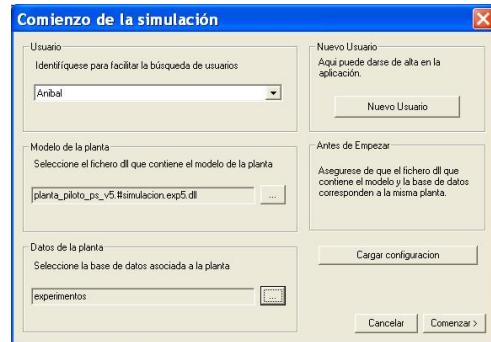


Figura 3: Configuración del modelo de la planta y de la base de datos a utilizar.

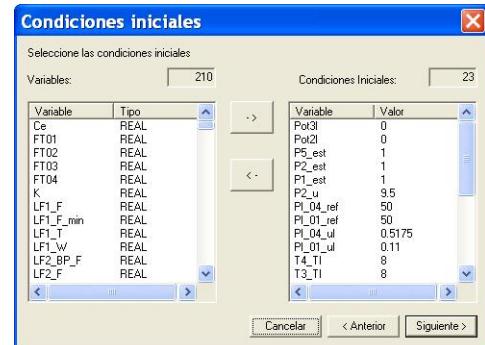


Figura 4: Selección y configuración de las condiciones iniciales de la simulación.

5.3. Ejecución de las simulaciones y almacenamiento

Una vez que se han establecido los parámetros relativos a las simulaciones, SimBloq procede a ejecutar los experimentos de simulación. La Figura 6 muestra el diagrama de flujo del proceso de ejecución. Como se puede ver SimBloq se conecta con la base de datos y obtiene toda la información referente a la planta, realiza una iteración por cada experimento a ejecutar, en cada una de ellas inserta las condiciones iniciales, y realiza la integración a lo largo del período de simulación. Durante este proceso el sistema inserta los eventos correspondientes a los fallos. Finalmente, el resultado de cada uno de los experimentos se almacena en la base de datos.

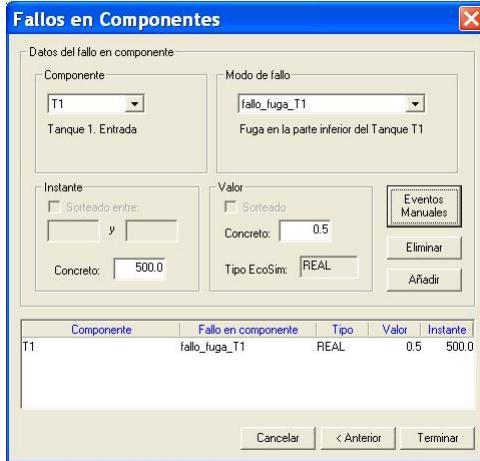


Figura 5: Configuración del conjunto de fallos que se produzcan durante la simulación.

5.4. Visualización de los resultados

El usuario puede acceder a los datos de los experimentos que ha ejecutado. Éste puede seleccionar la base de datos que desee y SimBloq le mostrará la lista de todos los experimentos. Seleccionando uno de estos experimentos, la herramienta muestra una ventana donde el usuario puede escoger las variables que deseé y representar gráficamente su evolución temporal. Un ejemplo de esta ventana puede observarse en la Figura 7. SimBloq también permite realizar búsquedas en función de diferentes parámetros como el nombre del experimento, el autor o la descripción.

6. Discusión y Trabajo Futuro

Se ha desarrollado un sistema para llevar a cabo experimentos de simulación en bloque. El sistema integra diversas herramientas y amplía la funcionalidad de algunas de ellas con objeto de facilitar la generación de las simulaciones en bloque. El sistema se ha utilizado dentro del contexto del diagnóstico de fallos, y ha probado ser útil proporcionando módulos de diagnóstico de fallos con un gran número de simulaciones en muy diferentes condiciones de trabajo.

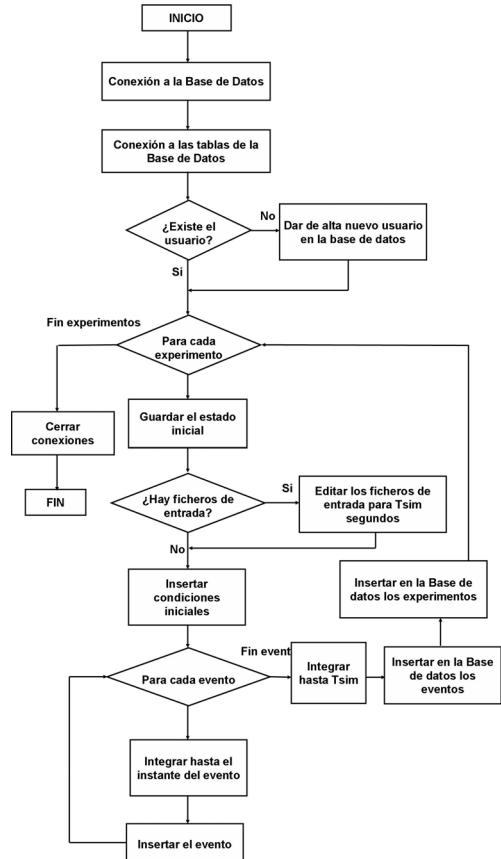


Figura 6: Funcionamiento de Simbloq.

El sistema desarrollado puede trabajar adecuadamente independientemente del modelo de planta utilizado. El sistema solamente necesita conocer la ruta del fichero que encapsule el modelo matemático de la planta, y la ubicación de la base de datos. Además, la información generada durante la configuración de las múltiples simulaciones en bloque se almacena en ficheros XML.

El sistema ha sido probado con el modelo de una planta de laboratorio (ver [2] para más detalles), donde se llevaron a cabo diversos experimentos de 14 modos de fallo diferentes.

Como conclusión principal podemos destacar que la utilización de elementos de almacenamiento independientes de la simulación

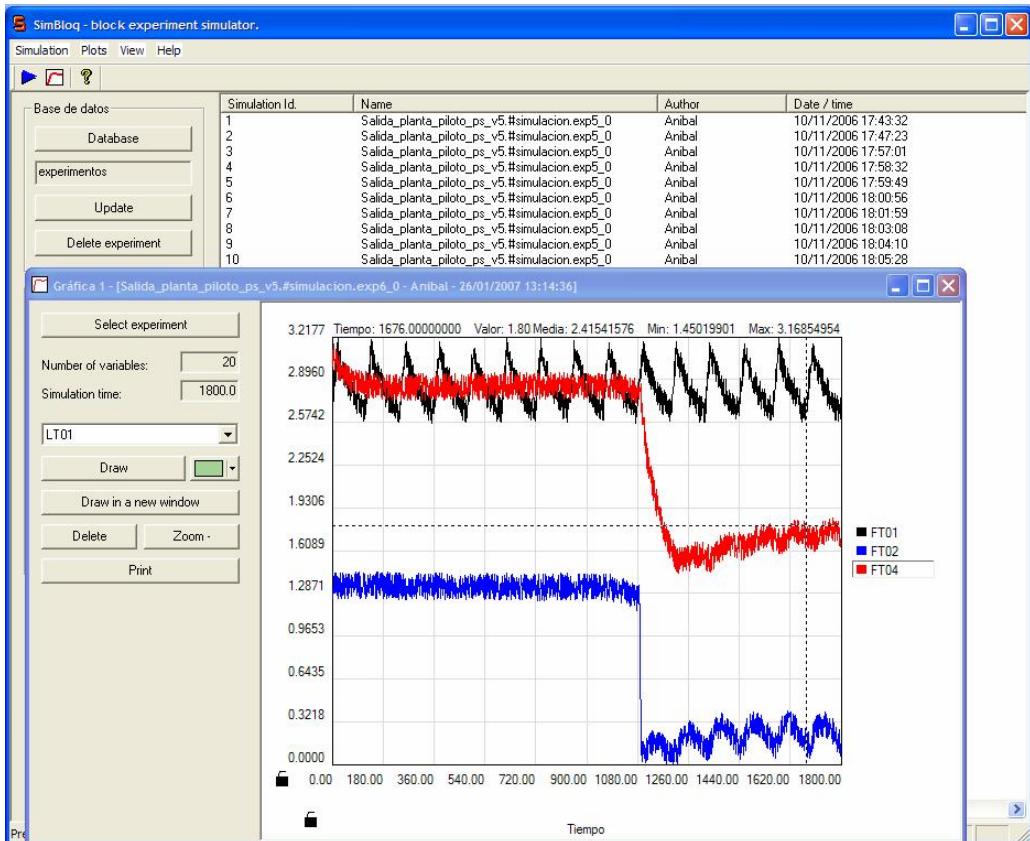


Figura 7: Interfaz gráfica de Simbloq con algunos resultados de un experimento.

(configurar experimentos con XML y una base de datos estándar) permite aislar el proceso de simulaciones en bloque del control del mismo.

Como trabajo futuro, planeamos extender el fichero XML, añadiendo cierta información que actualmente solo está incluida en la base de datos (como por ejemplo el modo de operación de la planta).

Referencias

- [1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault Tolerant Control*. Springer, 2003.
- [2] A. Bregón, M.A. Simón, J.J. Rodríguez, C. Alonso, B. Pulido, and I. Moro. Early

Fault Classification in Dynamic Systems Using Case-Based Reasoning. In *Current Topics in Artificial Intelligence. CAE-PIA 2005 Revised Selected Papers.*, volume 4177 of *LNAI*, pages 211–220. Springer Verlag, Berlin, Germany, 2006.

- [3] D. García and B. Pulido. *Herramienta Configurable para la Gestión de Experimentos de Simulación de Modelos Ecosim-Pro*. E.T.S.I. Informática. Universidad de Valladolid., 2006.
- [4] J.J. Gertler. *Fault detection and diagnosis in Engineering Systems*. Marcel Dekker, Inc., Basel, 1998.

- [5] C. Price. *Computer-based diagnostic systems*. Springer Verlag, New York, 1999.
- [6] O. Prieto, E. Gelso, B. Pulido, J. Rodríguez, and J. Maestro. Specification for a Versatile Data Acquisition Module for a Supervision Application. *3rd International Workshop on Practical Applications of Agents and Multiagents systems (IW-PAAAMS 2004)*, pages 361–364, 2004.
- [7] V. Puig, J. Quevedo, T. Escobet, B. Morigeo, and C. Ocampo. Control Tolerante a Fallos (parte I): Fundamentos y Diagnóstico de Fallos. *Revista Iberoamericana de Automática Industrial*, pages 39–59, 2004.

A. Apéndice

```
<!ELEMENT DATOSIMULACION (USUARIO, DATOSFICHERO, ESTADO,
PROTOCOLO,VARIABLES, CONDICIONESSIMULACION, FICHEROENTRADA*, SALIDAS,CONDICIONES_INICIALES, FALLOSCOMPONENTE*)>
<!ELEMENT USUARIO (#PCDATA)>
<!ELEMENT DATOSFICHERO (RUTAFICHERO, NOMBREFICHERO,
NOMBREBASE, NOMBREBBDD, DATOSDLL)>
<!ELEMENT RUTAFICHERO (#PCDATA)>
<!ELEMENT NOMBREFICHERO (#PCDATA)>
<!ELEMENT NOMBREBASE (#PCDATA)>
<!ELEMENT NOMBREBBDD (#PCDATA)>
<!ATTLIST DATOSDLL
experimento      CDATA  #REQUIRED
fecha_experimento CDATA #REQUIRED
biblioteca       CDATA  #REQUIRED
componente        CDATA  #REQUIRED
fecha_componente  CDATA  #REQUIRED
fecha_particion   CDATA  #REQUIRED>
<!ELEMENT ESTADO
id      CDATA #REQUIRED
nombre  CDATA #REQUIRED>
<!ELEMENT PROTOCOLO>
<!ATTLIST PROTOCOLO
id      CDATA #REQUIRED
nombre  CDATA #REQUIRED>
```

```
<!ELEMENT VARIABLES (VAR+)>
<!ELEMENT VAR>
<!ATTLIST VAR
id_variable      CDATA  #REQUIRED
nombre          CDATA  #REQUIRED
tipo    (INICIO|MEDIDA|FALLO) CDATA #REQUIRED
nombre_ecosim    CDATA  #REQUIRED
valor_sup        CDATA  #REQUIRED
valor_exac       CDATA  #REQUIRED
valor_inf        CDATA  #REQUIRED>
<!ELEMENT CONDICIONESSIMULACION (TSIM, CINT)>
<!ELEMENT TSIM (#PCDATA)>
<!ELEMENT CINT (#PCDATA)>
<!ELEMENT FICHEROENTRADA (COLUMNA+)>
<!ATTLIST FICHEROENTRADA
variable      CDATA  #REQUIRED
direccion    CDATA  #REQUIRED>
<!ELEMENT COLUMNA>
<!ATTLIST COLUMNA
valor          CDATA  #REQUIRED
ruido    (Si|No) CDATA  #REQUIRED
valor_ruido    CDATA  #REQUIRED
sierra          CDATA  #REQUIRED
sorteo    (Si|No) CDATA  #REQUIRED
sorteo_inf      CDATA  #REQUIRED
sorteo_sup      CDATA  #REQUIRED
orden          CDATA  #REQUIRED>
<!ELEMENT SALIDAS (VARIABLE_MEDIDA+)>
<!ELEMENT VARIABLE_MEDIDA (#PCDATA)>
<!ELEMENT CONDICIONES_INICIALES (CI*)>
<!ELEMENT CI
<!ATTLIST CI
nombre      CDATA  #REQUIRED
tipo       CDATA  #REQUIRED
valor      CDATA  #REQUIRED
valor_inf   CDATA  #REQUIRED
valor_sup   CDATA  #REQUIRED>
<!ELEMENT FALLOSCOMPONENTE (FC*)>
<!ELEMENT FC
<!ATTLIST FC
componente   CDATA  #REQUIRED
id_componente CDATA  #REQUIRED
modo_fallo   CDATA  #REQUIRED
id_modo_fallo CDATA  #REQUIRED
id_variable   CDATA  #REQUIRED
tipo         CDATA  #REQUIRED
valor        CDATA  #REQUIRED
instante     CDATA  #REQUIRED
tmin         CDATA  #REQUIRED
tmax         CDATA  #REQUIRED>
```

SIMULACIÓN CENTRALIZADA MÚLTIPLE FRENTE A SIMULACIÓN DISTRIBUIDA APLICADA A UN ROBOT MÓVIL AUTÓNOMO

Norberto Cañas de Paz

Departamento de Informática Aplicada
Universidad Politécnica de Madrid
28031 Madrid
norberto@eui.upm.es

Gracián Triviño Barros

European Centre for Soft Computing
Mieres
33600 Asturias
gracian.trivino@softcomputing.es

Resumen

En el ámbito de los sistemas embarcados tiene interés realizar pruebas del software simulando el resto del sistema anfitrión y el entorno. Ello permite evaluar un número muy elevado de situaciones con ventajas significativas (detección temprana de defectos, ajuste de reguladores, obtención temprana de medidas de fiabilidad del software, etc.)

En este trabajo mostramos que, para el caso de tener que efectuar muchos ensayos con un simulador parametrizable en un sistema multicomputador, en general, es más eficiente utilizar una configuración centralizada de dicho simulador, para cada nodo de la red, que utilizar una configuración distribuida del mismo repetida el número de veces que permita el tamaño de la red. Acompañamos la propuesta con un ejemplo práctico de aplicación, relacionado con el software de control embarcado de un robot móvil.

Palabras clave: Simulación, incremento de velocidad “speedup”, sistemas móviles autónomos.

1. Introducción

Una de las ventajas de la simulación de un sistema es la posibilidad de realizar miles de pruebas que

tendrían una duración real inaceptable. El interés por realizar simulaciones en las que el sistema simulado evolucione más rápidamente que el equivalente sistema real no es reciente. Desde finales de los años 70 y principios de los 80 [7] se desarrollaron diferentes alternativas para afrontar el problema, destacando rápidamente dos opciones para conseguirlo: simulación dirigida por tiempo y simulación dirigida por eventos. En relación con las mismas aparecieron variantes distribuidas, dado el beneficio que se obtiene al parallelizar la ejecución de componentes que en la realidad evolucionan de forma concurrente. Sin embargo, pronto surgieron dificultades como consecuencia de, por ejemplo, la necesidad de un reloj centralizado o una cola de eventos compartida, situaciones que limitaban la escalabilidad del simulador. Algunos de dichos impedimentos fueron superados de forma muy elegante, hasta el punto de asentar precedentes muy referenciados en la actualidad. Tal es el caso de la propuesta de relojes locales, en lugar de uno global, planteada por Chandy y Misra en [4] o de la alternativa especulativa para simulación dirigida por eventos propuesta por Jefferson [8]. Como efecto negativo es conveniente destacar que, en esta y otras propuestas con similares objetivos, se introduce una complejidad importante en los simuladores, en relación con sus versiones centralizadas.

A fecha de hoy parece aceptarse que el incremento de computadores en una simulación que se puede distribuir (cada nodo de la red ejecuta un componente del simulador), generalmente produce un incremento en la velocidad en que esta se realiza. Sin embargo, muy diferentes experiencias han mostrado la

existencia de un techo, en dicho aumento, que difícilmente se puede superar [12]. De hecho, las curvas de incremento de velocidad frecuentemente parecen adaptarse a la expresada en la ecuación (1), donde “n” es el número de computadores disponible y los parámetros “a” y “b” son específicos del tipo de sistema a simular, red de comunicaciones utilizada, etc.

$$v = a(1 - e^{-bn}) \quad (1)$$

La gran mayoría de los trabajos, que abordan el tema de incremento de velocidad en las simulaciones, centran su esfuerzo en resolver el problema asociado al incremento de velocidad de una única simulación. En el caso que nosotros planteamos existe la necesidad de ejecutar un simulador parametrizable un número muy elevado de veces.

Al disponer de una red de computadores para realizar simulaciones, dado que se obtienen mejores velocidades en configuraciones distribuidas (para una única simulación) podemos tener la tentación de intentar ejecutar nuestro número elevado de ensayos siguiendo esta alternativa. Lo cierto es que consideramos que hay argumentos (nosotros planteamos uno) para no actuar de esta manera. Si lo que perseguimos es ejecutar el mayor número de simulaciones por unidad de tiempo, otra alternativa consiste en utilizar cada nodo de la red para soportar una sesión de simulación independiente. Dicha alternativa, además de ser más sencilla, por no tener que abordar la problemática de la sincronización distribuida, ofrece desde un punto de vista teórico un mayor número de simulaciones por unidad de tiempo que la alternativa distribuida, como posteriormente analizaremos.

El resto de esta comunicación está organizado de la siguiente manera. En primer lugar argumentamos el beneficio de una configuración centralizada frente a otra distribuida, al tener que hacer un número elevado de simulaciones en una red de computadores. Seguidamente presentamos un prototipo de simulador de un robot móvil construido bajo las directrices que se deducen del apartado anterior. Por último presentamos algunos resultados obtenidos con dicho simulador.

2. Propuesta de simulación múltiple

La propuesta que planteamos ha surgido de necesidades concretas en trabajos que debemos realizar en el ámbito de los robots móviles autónomos, no obstante, consideramos que es extrapolable a otros ámbitos de desarrollo con necesidades similares.

La propuesta tiene interés para las situaciones de ensayo que seguidamente planteamos:

1. Disponemos de un sistema multicamputador para realizar simulaciones.
2. Contamos con un simulador parametrizable que nos permite realizar, con los mismos componentes internos, distintos ensayos.
3. El número de ensayos a realizar es elevado.

En este caso concreto, defendemos la utilización de una configuración centralizada para cada una de las simulaciones, por obtenerse en una red de computadores un mayor número de simulaciones por unidad de tiempo que en configuración distribuida.

2.1. Análisis de la propuesta

Supongamos que disponemos de un conjunto finito de componentes de software que configuran un simulador $C = \{a, b, c, d, \dots\}$, siendo “q” la cardinalidad del conjunto C.

Consideremos que el incremento de velocidad que se obtiene, como consecuencia de ejecutar cada componente de la simulación en un nodo distinto (configuración distribuida), se calcula, según se define tradicionalmente (por ejemplo en [17]), como la relación que existe entre el tiempo que tarda en ejecutarse la simulación en un único computador y el tiempo que tarda en ejecutarse en un sistema multicamputador. En nuestro caso $S = T(1)/T(q)$.

El número de simulaciones, en configuración distribuida que podemos ejecutar simultáneamente en nuestra red, de “n” computadores, es n/q (supongamos que “n” es múltiplo exacto de “q” y que queremos respetar el incremento de velocidad calculado). En su configuración centralizada, el número de simulaciones que podemos ejecutar simultáneamente es igual al número de computadores “n”.

El tiempo que tardarían en ejecutarse las (n/q) simulaciones, en configuración distribuida, que caben simultáneamente en la red es $T(1)/S$. El

tiempo que tardarían en ejecutarse las (n) simulaciones centralizadas, que caben en la red, es $T(1)$.

El total de simulaciones terminadas por unidad de tiempo en la configuración distribuida es el expresado en (2), mientras que en la configuración centralizada es el de (3).

$$\left(\frac{n}{q}\right) / \left(\frac{T(1)}{S}\right) \quad (2)$$

$$\frac{n}{T(1)} \quad (3)$$

Para que sea más ventajosa la opción distribuida es necesario que se cumpla la desigualdad (4), de la que se deduce (5) y finalmente (6).

$$\left(\frac{n}{q}\right) / \left(\frac{T(1)}{S}\right) > \frac{n}{T(1)} \quad (4)$$

$$(1/q)/(1/S) > 1 \quad (5)$$

$$S/q > 1 \quad (6)$$

No es frecuente que el incremento de velocidad que se obtiene al descomponer una simulación sea superior al número de componentes (de hecho, tradicionalmente se ha considerado como ideal alcanzar un incremento de velocidad igual al número de componentes en que se divide la simulación [17]). Es por ello que, en las condiciones planteadas, resulta más adecuado optar por una configuración centralizada para cada nodo de la red, frente a la alternativa distribuida.

En el análisis no hemos tenido en cuenta el retardo por incremento de comunicaciones cuando “ q ” y/o “ n ” son grandes. Tampoco hemos considerado que los elementos que constituyen una simulación puedan consumir distintos tiempos de procesador así como arrancar y acabar en distintos instantes. En cualquier caso, si consideramos estos factores, no hacen sino empeorar la opción distribuida.

2.2. Otros aspectos de interés

Perfil de diseño de procesos

El software asociado a actividades periódicas suele respetar un diseño como el indicado en el algoritmo 1.

```

loop
  Lectura del estado actual
  Cálculo de la respuesta
  Generación de la respuesta
  Espera siguiente actuación
end loop

```

Algoritmo 1.

En el caso de actividades no periódicas y esporádicas la diferencia más notable es que no tiene porqué respetarse en ellas un patrón de activación de frecuencia constante.

En cualquier caso, para una simulación rápida, tiene interés controlar el estado de bloqueo de los procesos ya que si se produce simultáneamente en todos ellos es posible adelantar el reloj del sistema hasta el instante en que deba desbloquearse al primero.

Restricciones de tiempo real

Muchos sistemas embarcados deben respetar restricciones de tiempo real, que de no hacerse pueden generar consecuencias inaceptables (como por ejemplo inestabilidad por no respetarse el periodo de los controladores discretos). Actualmente existen sistemas operativos de tiempo real (VxWorks, QNX, etc.) y lenguajes de programación como ADA (95 y 2005) que soportan políticas de planificación de tiempo real avanzadas y suficientemente maduras, como por ejemplo techo de prioridad y sus variantes [2,10]. Una simulación rápida debe garantizar el respeto de la política de planificación seleccionada, para así poder extraer conclusiones con respecto al sistema real equivalente.

Coordinación cliente servidor

Para utilizar la propuesta de simulación múltiple planteada al principio del presente apartado, parece tener interés utilizar algún instrumento que facilite la ejecución distribuida. Esto es así porque, si bien cada simulación va a tener un diseño centralizado, debe ahorrarse tiempo también en su activación, recogida de resultados y, a ser posible, en su evaluación. Un esquema que parece adecuarse a nuestras necesidades es el de profesor-alumnos (teacher-pupils) [5], en el que un servidor distribuye pruebas a diferentes estaciones de trabajo y recoge los resultados obtenidos en el ejercicio realizado.

3. Prototipo de simulador de robot móvil

Como ejemplo de aplicación de lo expuesto en el apartado (2) hemos desarrollado el simulador para un robot móvil. Dicho simulador va a ser utilizado en un experimento relativamente sencillo en el que se pretende optimizar, con un algoritmo genético, los parámetros de un controlador de velocidad PID multivariable. Cada nodo de la red tendrá un simulador (alumno) y esperará a que un servidor (profesor) le suministre los parámetros de la prueba a realizar, devolviendo los resultados al mismo al terminar. El sistema completo ha sido programado en Ada 95 con glade (anexo de sistemas distribuidos de Ada para el compilador Gnat [15])

Los componentes más destacados del simulador son los siguientes:

- Manejador de la simulación. Aplicando un algoritmo genético prepara distintas simulaciones con diferentes parámetros para el controlador del vehículo.
- Simulador. Existe uno por nodo en la red de ordenadores. Coordina la simulación solicitada por el Manejador de la simulación y devuelve el resultado de la misma.
- Componentes del sistema software embarcado
 - Navegador. Ejecuta las actividades de navegación del robot, estableciendo consignas para el controlador de velocidad según un plan establecido.
 - Controlador. Ejecuta el algoritmo de control de velocidad.
- Componentes del sistema anfitrión
 - Inclinómetro. Dispositivo que suministra los ángulos de alabeo (Φ), cabeceo (θ) y guiñada (ψ) del vehículo (marco de referencia global).
 - Sensor de velocidad. Dispositivo que suministra la velocidad lineal del vehículo respecto a su eje longitudinal así como las tres velocidades angulares asociadas a los tres ejes de referencia local (velocidad de alabeo, cabeceo y guiñada).
 - Motores. Simulan motores de corriente continua sobre los que el controlador establece la tensión de alimentación.
 - Dinámica del vehículo. Encapsula la simulación del comportamiento dinámico del vehículo atendiendo, entre otras cosas,

al estímulo de los motores así como a la inclinación del suelo.

- *Entorno*

- Suelo. Simula la superficie en la que evoluciona el robot.

- *Componentes auxiliares*

- Reloj. Gestiona la evolución virtual del tiempo, así como da soporte al bloqueo temporizado de procesos que permite sustituir el servicio *delay until* de ADA95.
- Gestor de tiempo. Componente que se encarga de adelantar el tiempo cuando se detecta que todos los procesos están bloqueados.
- Sonda. Elemento que supervisa la evolución de la simulación. Suministra la evaluación final de la misma o la detiene en caso de ser defectuosa.

3.1. Modelo del robot

Un modelo es una estructura que permite entender el comportamiento de un sistema [3]. El grado de fidelidad del modelo dependerá del tipo de simulación que se pretenda efectuar con él.

Existen muchos y diferentes modelos de robots que en cierta medida cubren parte de nuestras inmediatas expectativas, como por ejemplo algunos de los planteados en las referencias [9,13], pero finalmente hemos optado por desarrollar uno ligeramente distinto a los consultados, por motivos que no tiene interés detallar en el presente escrito.

El robot que modelamos es una plataforma con configuración de triciclo en sus ruedas (dos motrices en la parte delantera y una rueda de apoyo central trasera tipo castor). La figura 1 presenta una vista cenital del mismo.

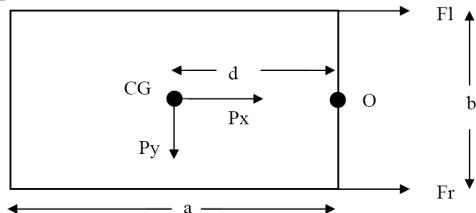


Figura 1.

Consideraremos que el cuerpo del vehículo lo constituye un paralelepípedo de longitud “a”, ancho “b” y altura “c”. El origen del marco de

referencia local (O) se encuentra en el centro del eje motriz, teniendo sus ejes alineados con los ejes principales del cuerpo del robot (eje "x" paralelo al eje longitudinal, eje "y" paralelo al eje transversal y eje "z" paralelo al eje vertical)

En nuestro caso, para obtener el modelo, planteamos un conjunto de simplificaciones de las cuales destacamos las siguientes¹:

1. En el marco de referencia local del vehículo consideramos la velocidad lineal "u" (en el eje x) y las velocidades angulares "p", "q" y "r" (alrededor de los ejes x,y,z).
2. En el marco de referencia local solo consideramos explícitamente las fuerza actuante "X" (longitudinal) y el momento "N" (rotación respecto al eje vertical).
3. Las velocidades angulares "p" y "q" se aproximan a partir del cambio de inclinación del vehículo.
4. Algunas fuerzas de resistencia se han considerado infinitas o nulas, según el caso, mientras otras, de valor muy reducido, se han programado pero no las consideramos en la siguiente exposición.

En la figura 1, F_r y F_l son las fuerzas ejercidas por el motor derecho y el izquierdo respectivamente y P_x y P_y las fuerzas en que se descompone el peso del vehículo cuando tiene algún tipo de inclinación. La fuerza "X" y el momento "N" se calculan según las siguientes ecuaciones.

$$X = F_r + F_l + P_x \quad N = \frac{b}{2}(F_r - F_l) + P_y \frac{a}{2} \quad (7)$$

Teniendo en cuenta la posición del centro de coordenadas local (O), el centro de gravedad se encuentra en la posición indicada en (8), lo que nos permite obtener el tensor de inercia, referido a los ejes coordinados indicados (ecuación 9 y 10), siendo "M" la masa del paralelepípedo (uniformemente distribuida).

$$C_N = \left[-\frac{a}{2}, 0, \frac{c}{2} \right]^T \quad (8)$$

¹ En las ecuaciones que seguidamente planteamos utilizamos la notación SNAME y las mismas responden a un análisis mecánico Newtoniano.

$$I_N = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \quad (9)$$

$$I_N = \begin{bmatrix} M\left(\frac{b^2+c^2}{12} + \frac{a^2}{3}\right) & 0 & \frac{1}{4}Mac \\ 0 & \frac{1}{3}M(a^2+c^2) & 0 \\ \frac{1}{4}Mac & 0 & M\left(\frac{a^2+b^2}{3} + \frac{c^2}{12}\right) \end{bmatrix} \quad (10)$$

Ello nos permite obtener la matriz de inercia M_{RB} y una posible configuración de la matriz de componentes centrífugos y de Coriolis C_{RB} .

$$M_{RB} = \begin{bmatrix} m & 0 & mzG & 0 \\ 0 & I_x & 0 & -I_{xz} \\ mzG & 0 & I_y & 0 \\ 0 & -I_{xz} & 0 & I_z \end{bmatrix} \quad (11)$$

$$C_{RB} = \begin{bmatrix} 0 & mzG^r & -mxG^q & -mxG^r \\ -mzG^r & 0 & -I_{xz}p + I_zr & -I_yq \\ mxG^q & I_{xz}p - I_zr & 0 & -I_{xz}r + I_xp \\ mxG^r & I_yq & I_{xz}r - I_xp & 0 \end{bmatrix} \quad (12)$$

Por último el vector de entrada y el de estado son los expresados en las ecuaciones (13 y 14).

$$\tau = [X \quad N]^T \quad (13)$$

$$v = [u, p, q, r]^T \quad (14)$$

Todo ello nos permite obtener la ecuación de estado del sistema continuo.

$$M_{RB} \frac{dV}{dt} + C_{RB}(V)V = \tau \quad (15)$$

$$v' = -M_{RB}^{-1}C_{RB}(v)v + M_{RB}^{-1}\tau \quad (16)$$

Por tanto, la ecuación de estado discreto pasa a ser la siguiente (según las directrices establecidas en el capítulo 9 de la referencia [1]).

$$V(t_{k+1}) = \Phi(t_k)V(t_k) + \Gamma(t_k)\tau(t_k) \quad (17)$$

Calculándose los componentes de dicha ecuación según las siguientes expresiones.

$$\Phi(t_k) = e^{AT} \quad T = t_{k+1} - t_k \quad (18)$$

$$A = A(t_k) = -M_{RB}^{-1}C_{RB}(V(t_k)) \quad (19)$$

$$\Gamma(t_k) = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} d\tau B(t_k) \quad (20)$$

$$B(t_k) = M_{RB}^{-1} \quad (21)$$

Existen interesantes métodos aproximados para obtener las matrices Φ y Γ , algunos de los cuales pueden consultarse en la referencia [6].

Para determinar las componentes, en el marco de referencia local, del peso del vehículo (P_x y P_y), podemos efectuar la rotación del vector peso atendiendo a la actitud del vehículo (Φ, θ, ψ) según marcan las siguientes ecuaciones.

$$P_2 = J_1(\eta_2) P_1 \quad (22)$$

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & s\psi c\theta & s\theta \\ c\psi s\theta s\phi - s\psi c\phi & s\psi s\theta s\phi + c\psi c\phi & -c\theta s\phi \\ -s\psi s\phi - c\psi s\theta c\phi & c\psi s\phi - s\psi s\theta c\phi & c\theta c\phi \end{bmatrix} \quad (23)$$

Con todas estas consideraciones, el cálculo del vector "X" y "N", de forma matricial, se puede hacer de la siguiente manera.

$$\begin{bmatrix} X \\ N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ b & -b & 0 & a \\ 2 & -2 & 0 & 2 \end{bmatrix} \begin{bmatrix} F_r \\ F_l \\ P_x \\ P_y \end{bmatrix} \quad (24)$$

Para el cálculo de la actitud del vehículo efectuamos un proceso cíclico consultando al simulador del suelo la inclinación del terreno, hasta conseguir que la posición de las ruedas respete las dimensiones del vehículo.

El cálculo de la velocidad en el marco de referencia global puede hacerse según la siguiente ecuación.

$$\begin{bmatrix} x' \\ y' \\ \psi' \end{bmatrix} = \begin{bmatrix} c\psi c\theta & 0 \\ c\psi s\theta s\phi - s\psi c\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix} \quad (25)$$

Para simular el comportamiento de los motores nos hemos basado en el análisis que de los mismos puede encontrarse en el apartado 2.6 de la referencia [14] (referencia a la que remitimos al lector interesado).

3.2. Validación preliminar del modelo

Hemos efectuado una evaluación preliminar del modelo, a expensas de hacer una más completa y contrastada con un robot real. Las pruebas se han efectuado en lazo abierto (sin regulación) imponiendo tensiones de alimentación concreta a los motores. En las pruebas realizadas se han observado los efectos esperados debido a la actuación de los motores así como a la acción de la gravedad cuando el suelo estaba inclinado.

3.3. Control de velocidad

Como primer ensayo, hemos dotado al modelo de un controlador PID multivariable. Para ello hemos considerado que el vector error de velocidad, para el instante siguiente, se calcula como:

$$e(t_{k+1}) = -K_p e(t_k) - K_d \frac{e(t_k) - e(t_{k-1})}{T} - K_i T \sum_{n=1}^k e(t_n) \quad (26)$$

Siendo K_p , K_i , K_d la matriz proporcional, integral y derivativa respectivamente.

El error de velocidad puede expresarse como la diferencia entre la velocidad deseada y la real.

$$e(t_{k+1}) = V_D(t_{k+1}) - V_R(t_{k+1}) \quad (27)$$

Lo que nos lleva a:

$$V_R(t_{k+1}) = V_D(t_{k+1}) - e(t_{k+1}) \quad (28)$$

Despejando en la ecuación (17) obtenemos:

$$\tau(t_k) = \Gamma^{-1}(t_k) [V_R(t_{k+1}) - \Phi(t_k) V_R(t_k)] \quad (29)$$

Sustituyendo en la ecuación anterior $V_R(t_{k+1})$ por el valor indicado en la ecuación (28) y posteriormente $e(t_{k+1})$ por el indicado en (26) obtenemos el controlador completo.

Teniendo en cuenta que el vector de pares de fuerzas así calculado es el total, es necesario restar la contribución de la gravedad si hay inclinación en el suelo.

$$\begin{aligned} \tau(t_k) &= \tau_{motores} + \tau_{gravedad} \\ \tau_{motores} &= \tau(t_k) - \tau_{gravedad} \end{aligned} \quad (30)$$

Y de esta última ecuación puede despejarse el par a establecer como consigna para cada uno de los motores, de forma que se respete la velocidad perseguida.

4. Resultados

Planteamos a continuación los resultados más destacados que permiten valorar una hipotética descomposición para una configuración distribuida así como un pequeño resumen de características relevantes de un ejercicio realizado con el simulador desarrollado.

4.1. Incremento de velocidad

En el simulador centralizado que hemos construido, las tareas que más tiempo consumen

son la tarea controladora y la tarea que simula el comportamiento dinámico del vehículo.

Al repetir una misma simulación 10 veces hemos obtenido los siguientes tiempos de ejecución (utilizando el reloj del sistema).

Elemento	Tiempo consumido
Simulador completo	69.636s
Dinámica del vehículo	32.3219s
Tarea controladora	35.7706s
Otras actividades	1.5435s

Tabla 1.

Con estos datos, el incremento teórico de velocidad que se puede obtener, con una descomposición para dos computadores (dinámica del vehículo + otras actividades –un componente-, frente a tarea controladora –otro componente-), es 1.95. Creemos difícil obtener un incremento mayor con otras descomposiciones. Si retomamos la desigualdad (6), para la descomposición plantada tendríamos $S=1.95$ y $q=2$. Claramente no se cumple la desigualdad, luego es más ventajoso realizar simulaciones centralizadas frente a simulaciones distribuidas. De hecho, con estos datos, si disponemos de dos computadores, hacer dos simulaciones centralizadas (una en cada ordenador) consumiría 6,9636s, mientras que hacer dos simulaciones distribuidas (un componente en cada ordenador y dos simulaciones consecutivas) consumiría 7.15412s ($3.57706 * 2$).

En detrimento de la opción distribuida habría que añadir el perjuicio para el incremento de velocidad que supone la utilización de elementos centralizados (reloj, lista de eventos, etc.) o, en su defecto, la necesidad de utilizar algoritmos netamente más complejos que los necesarios para las simulaciones centralizadas, lo que incrementa la dificultad de obtener simuladores fiables.

4.2. Estimación de parámetros de un controlador

Los algoritmos genéticos han sido y son utilizados con frecuencia en la optimización de controladores [11,16,18]. Ello es debido, entre otros motivos, a la posibilidad de realizar búsquedas relativamente eficientes, en espacios de soluciones muy amplios, con el beneficio añadido de poder sortear mínimos locales que podrían desembocar en soluciones poco adecuadas.

Hemos querido hacer un primer ensayo relativamente sencillo de nuestra propuesta de simulación múltiple centralizada. Para ello hemos preparado un sistema que, por medio de un algoritmo genético básico (elitismo + combinación + mutación, con población constante de 750 individuos), trata de optimizar los parámetros del controlador de velocidad de nuestro vehículo simulado.

El simulador se ha utilizado para evaluar las propuestas generadas por el algoritmo genético (para todos los casos se exigía efectuar un recorrido con velocidad lineal de 0.1m/s durante 60 segundos, a la que posteriormente se debía añadir una velocidad de rotación de 0.05 rad/s durante 375 segundos, repitiéndose el proceso en cinco superficies distintas).

Presentamos seguidamente, de forma gráfica, el recorrido realizado por el vehículo con un controlador obtenido en la segunda generación de un ensayo con el algoritmo genético (figura 2 para suelo plano y figura 4 para suelo ondulado). Puede observarse una mejora significativa con un controlador obtenido en la décima generación del mismo ensayo (figura 3 para suelo plano y figura 5 para suelo ondulado).

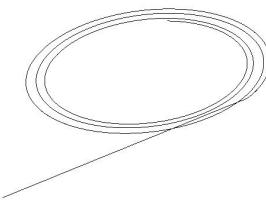


Figura 2.

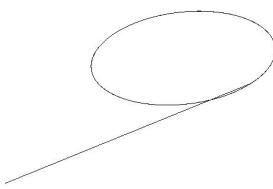


Figura 3.

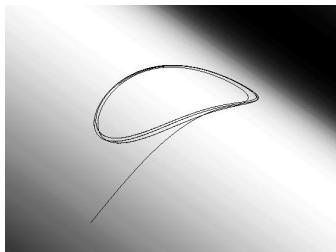


Figura 4.

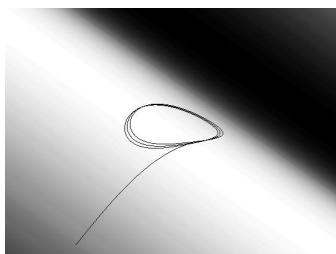


Figura 5.

5. Conclusiones

Existe una creciente demanda de sistemas de simulación que permitan realizar ajustes y aumentar la fiabilidad de los sistemas embarcados. Este trabajo presenta una contribución práctica y sencilla que debe tenerse en cuenta a la hora de configurar un simulador basado en un sistema multicamputador, en los numerosos casos en los que se dan las circunstancias descritas en esta comunicación.

El resultado de nuestro análisis recomienda un esquema más sencillo de simulación que además proporcionará los resultados en un tiempo más reducido.

Referencias

- [1] Brogan, W.L. Modern Control Theory. (pp. 319-322). Prentice Hall. 1991.
- [2] Burns, A.; Wellings, A. Real-Time Systems and Programming Languages. (pp. 418-422) Addison Wesley 1988.
- [3] Caughlin, D. An Integrated Approach to Verification, Validation and Accreditation of models and Simulations. Winter Simulation Conference. 2000.
- [4] Chandy, K.M.; Misra, J. Asynchronous Distributed Simulation via a Sequence or Parallel Computing. ACM Communications. Vol 24, Nº 11. 1981.
- [5] Fewster, M.; Graham D. Software Test Automation. (pp. 540) Addison Wesley. 1999.
- [6] Franklin, G.F.; Powell, J.D.; Workman, M.L. Digital Control of Dynamic Systems. (pp. 46-56) Addison-Wesley. 1990.
- [7] Fujimoto, R.M. Distributed Simulations Systems. Winter Simulation Conference. 2003.
- [8] Jefferson, D.R. Virtual Time. ACM Transactions of Programming Languages and Systems. Vol 7, Nº 3. 1985.
- [9] LaValle, S.M. Planning Algorithms. (pp. 718-776) Cambridge University Press. 2006.
- [10] Liu, J. Real-Time Systems. (pp. 290-326) Prentice Hall 2000.
- [11] Martínez, M.A.; Sanchis, J.; Blasco, X. Algoritmos Genéticos Aplicados al Diseño de Controladores Robustos. Revista Iberoamericana de Automática e Informática Industrial. Vol 3 Nº1. 2006.
- [12] Muttoni, L.; Casale, G.; Granata, F.; Zanero, S. Optimal Number for Computation in Grid Environments. Euromicro Conference on Parallel, Distributed and Networks-Based Processing. IEEE. 2004.
- [13] Nwokah, O.; Hurmuzlu, Y. The Mechanical System Design Handbook. (cap 26) CRC Press. 2002.
- [14] Ogata, K. Ingeniería de Control Moderna. (pp. 126-130) Prentice Hall. 1993.
- [15] Pautet, L.; Tardieu, S. Glade User's Guide. Glade, GNAT Library for Ada Distributed Environment. Free Software Foundation. 2005.
- [16] Pedersen, G.; Yang, Z. Multi-Objective PID-Controller tuning for a Magnetic Levitation System using NSGA-II. Genetic and Evolutionary Computation Conference. ACM. 2006.
- [17] Righter, R.; Walrand, J. Distributed Simulation of discrete Event Systems. Proceedings of the IEEE. Vol 77 Nº1. 1989.
- [18] Soltoggio, A. An Enhanced GA to Improve the Search Process Reliability in Tuning of Control Systems. Genetic and Evolutionary Computation Conference. ACM. 2005.

**MODELADO Y SIMULACIÓN DE SISTEMAS DE EVENTOS
DISCRETOS CON REDES DE PETRI**

Especificación, mediante Redes de Petri Coloreadas, del comportamiento de un sistema multiagente para abordar problemas logísticos

Sònia Busquets, Miquel Àngel Piera

Dept. de Telecomunicacions i d'Enginyeria de Sistemes

Universitat Autònoma de Barcelona

Campus UAB - Edifici Q, 08193 Bellaterra

Sonia.Busquets@uab.cat, MiquelAngel.Piera@uab.cat

Resumen

Si bien diferentes metodologías y herramientas han sido desarrolladas en el campo de sistemas de soporte a la decisión para mejorar el rendimiento en los problemas logísticos, la investigación de este campo está aún abierta debido a que la mayor parte de los actuales enfoques comerciales, no son lo suficientemente robustos cuando los problemas a tratar son considerados en su totalidad, como es el caso de sistemas de producción, transporte o logísticos. Se propone un enfoque basado en el desarrollo de un Sistema Multiagente junto con herramientas de Simulación para dividir problemas complejos de decisión en problemas individuales con objetivos locales. Cada problema local es manejado por un agente y un agente administrador decide cual es la mejor vía para conseguir el objetivo global. Este enfoque permite reducir considerablemente el crecimiento exponencial de los posibles escenarios a evaluar. Para modelar el problema y el comportamiento de los agentes, se ha utilizado el formalismo de Redes de Petri Coloreadas. Para explorar el espacio de búsqueda y poder así encontrar la solución óptima o una de factible, se ha aplicado la técnica de análisis del árbol de cobertura.

1. Introducción

En una economía cada vez más globalizada y competitiva, la logística, junto con la mejora

de la calidad de los productos y servicios, se convierte en uno de los aspectos clave para poder alcanzar unos resultados competitivos en la productividad de las empresas.

La logística se puede ver como la planificación, la implementación y el control eficaz del flujo y almacenaje de las materias primas, productos en el proceso, producto final y de toda la información relacionada, desde el punto de origen hasta el punto de consumo, para poder satisfacer los requerimientos del cliente minimizando el coste [14].

Los problemas de decisión en Sistemas Logísticos se consideran problemas NP-completos debido al crecimiento exponencial del espacio de búsqueda cuando crece el número de recursos a coordinar. Estos problemas han sido estudiados desde diferentes perspectivas como Investigación Operativa, Inteligencia Artificial y Simulación, pero aún son áreas de investigación abiertas. La industria requiere nuevos métodos que permitan tratar la complejidad inherente a problemas logísticos obteniendo soluciones óptimas o factibles.

La simulación permite, mediante un modelo del sistema, determinar los cambios que se provocarían en el sistema real al realizar una modificación sobre las configuraciones de sus parámetros. Aunque la utilización de técnicas de simulación como herramienta de soporte a la decisión comporta ventajas, el número de posibles estados a explorar para obtener una solución óptima puede ser intratable cuando

estos modelos se aplican a la optimización de problemas logísticos.

Los métodos heurísticos, aplicados adecuadamente, permiten conseguir soluciones factibles (incluso la óptima) utilizando menos recursos informáticos que la simulación. Sin embargo, las aproximaciones basadas en heurísticas son poco aceptadas en algunos contextos industriales debido a la falta de confianza en el comportamiento de los resultados obtenidos delante cambios en el contexto del sistema. De alguna manera, el uso de modelos con un conocimiento no-estructurado para representar el comportamiento del sistema, junto con la falta de formalización en el comportamiento heurístico, provocan la necesidad de diseñar heurísticas particulares para resolver problemas particulares, en vez de intentar integrar especificaciones heurísticas en el modelo del sistema para evaluar el comportamiento de forma integrada, independientemente de la reutilización del contexto.

En este artículo se presenta un nuevo enfoque para la obtención de una solución factible a un problema completo. Éste se basa en el principio *Dividir y Conquistar*, donde se propone dividir el problema en subproblemas más tratables. Para buscar una solución local para cada uno de los problemas locales, se utiliza un modelo del sistema formalizado mediante Redes de Petri Coloreadas (RdPC) y se analizan los posibles escenarios mediante el árbol de cobertura utilizando una herramienta de simulación de RdPC orientada a objetos [8], [10]. Finalmente, para encontrar una solución global, se propone la utilización de un sistema multiagente (SMA), desarrollado sobre la plataforma FIPA-OS, que permite coordinar y combinar las soluciones locales encontradas. Para evitar el problema de la no-estructuración del conocimiento en los SMA, se propone formalizar su comportamiento mediante RdPC. La utilización del mismo formalismo, para modelar el sistema logístico y describir el comportamiento de los agentes, permite desarrollar un sistema integrado.

El Problema de Enrutamiento de Vehículo (VRP) [15] se utiliza como caso práctico para ilustrar la metodología propuesta. El proble-

ma se divide en subproblemas, uno para cada camión, de manera que cada camión decide cuáles son las peticiones que puede servir. Un agente administrador coordinará las decisiones locales tomadas por cada camión para decidir como se realizará la asignación. El uso de un SMA para coordinar la distribución del VRP en subproblemas diferentes, reduce la cantidad de escenarios a explorar y por consiguiente es posible encontrar una solución factible en un tiempo computacional reducido.

En la sección 2 se introducen los aspectos principales del formalismo de modelado de Redes de Petri Coloreadas usado para describir tanto el comportamiento de sistema como el comportamiento de los agentes, junto con el análisis del árbol de cobertura utilizado para examinar el espacio de búsqueda. En la sección 3, se introducen los sistemas multiagentes como responsables para gestionar eficazmente la interacción entre los subproblemas locales. La sección 4 presenta un nuevo enfoque, basado en una especificación en RdPC del comportamiento del sistema juntamente con la formalización del comportamiento del SMA, usado para evaluar los posibles escenarios del problema logístico. Para ilustrarlo, se detalla la especificación realizada para abordar el VRP juntamente con los resultados obtenidos aplicando esta metodología a un caso particular.

2. Formalismo

Para especificar de manera estructurada el comportamiento de un sistema, un formalismo que puede ser utilizado son las Redes de Petri (RdP). Éste se basa en el uso de un lenguaje gráfico que permite diseñar, especificar, simular y verificar los sistemas [7].

Normalmente, los modelos de procesos logísticos realizados en RdP, son complejos debido a la incapacidad del formalismo para describir los cambios que se producen en la información de cada entidad. En general, hay muchas circunstancias inherentes a los sistemas logísticos en las cuales se hace necesaria la distinción de las entidades (tokens) que fluyen por el sistema. Como consecuencia, una desventaja de este formalismo es la falta de herramientas que

permitan representar eficazmente la información del sistema, no admite la descripción de las actividades en las que su comportamiento depende del estado de las entidades [9]. Para poder introducir esta información al modelo, es habitual el uso de las Redes de Petri Coloreadas (RdPC) como formalismo.

Los modelos formalizados mediante RdPC son una representación compacta del sistema estudiado. Este formalismo permite describir tanto la estructura completa del sistema como su comportamiento, y la información del estado del sistema mediante una representación compacta que facilita el mantenimiento del modelo [11].

Las RdPC ofrecen las herramientas necesarias para representar los atributos de las entidades (características que fluyen por el sistema) y las propiedades que determinan cuando un evento del sistema puede ejecutarse (cuándo una transición puede ser disparada). Además, el proceso de realizar la descripción y el análisis del sistema, usualmente permite al modelador mejorar la comprensión del sistema modelado [6]. Debido a una falta de herramientas de análisis cualitativas, los modelos realizados con RdPC normalmente son analizados mediante técnicas de simulación (análisis cuantitativo).

Las componentes principales de las RdPC [16] que habilitan el uso de este formalismo como una Representación de Conocimiento (RC) en el campo logístico son:

- Vector de Estado: La información más pequeña necesaria para poder predecir los eventos que pueden ocurrir. El vector de estado representa la cantidad de entidades que hay en cada lugar, y los colores de cada una de ellas.
- Expresiones de Arco y Guardas: Son utilizados para indicar qué tipo de entidades puede ser utilizado para disparar una transición.
- Conjuntos de Colores: Determinan los tipos, operaciones y funciones que pueden ser utilizados por los elementos del modelo.

- Lugares: Son muy útiles para especificar tanto las colas como condiciones lógicas.
- Transiciones: Son utilizadas para especificar los eventos que determinan la dinámica de sistema.

2.1. Árbol de cobertura

El análisis del árbol de cobertura es una técnica muy útil para poder examinar el espacio de búsqueda completo de un modelo basado en RdPC. La raíz del árbol consiste en la marcación inicial (M_0 -estado inicial-). Para cada nodo en el árbol, se reconocen todos los eventos permitidos (identificación de alternativas), para cada evento se calcula el nuevo estado (evaluación de alternativas) y su marcación se añade al árbol juntamente con un arco que une los dos nodos que especifican la transición disparada. En caso de que se haya definido un estado final, se pueden obtener los caminos que conducen de la marcación inicial al estado final [12].

Para realizar esta exploración, se puede usar una herramienta de Simulación de RdPC orientada a objetos. En la utilizada en este artículo [8] es necesario describir el modelo en RdPC de forma sintáctica, en un archivo ASCII, juntamente con el objetivo, especificado mediante el estado final de la RdPC, y las condiciones iniciales del problema. Una función de costes puede ser añadida para podar el espacio de búsqueda. La ejecución de esta herramienta da como resultado los posibles caminos diferentes que conducen del estado inicial al final [10].

3. Sistemas multiagente

Los sistemas multiagente (SMA) son especialmente adecuados en situaciones en las que son necesarios múltiples métodos de resolución de problemas y/o existen muchas entidades para la resolución de éstos. Una de sus ventajas es que a parte de tener las propiedades de los sistemas concurrentes, pueden utilizar patrones de interacción más complejos como pueden ser la cooperación, la coordinación y la negociación.

Un agente inteligente es un sistema informático que está situado en un entorno, y que es capaz de ejecutar acciones de forma autónoma en el entorno para conseguir los objetivos definidos, donde las acciones que realiza son flexibles, es decir, son reactivas, pro-activas y sociales [5]. Un sistema es multiagente cuando está diseñado e implementado como un conjunto de agentes autónomos que son capaces de interactuar y trabajar conjuntamente para conseguir un objetivo.

Para poder conseguir los objetivos globales de un SMA es necesaria una coordinación entre los diferentes agentes autónomos que lo conforman. Esta coordinación tiene que gestionar los diferentes tipos de dependencias que se producen entre los agentes como puede ser que los objetivos estén entrelazados, los agentes comparten un entorno común o bien se utilicen recursos compartidos. Los diferentes mecanismos de coordinación se pueden dividir en dos grandes grupos que son cooperación y negociación [4]. Además, es esencial que los agentes puedan comunicarse y entenderse para poder utilizar los mecanismos introducidos, por lo que se hace necesario el uso de un lenguaje de comunicación entre agentes (ACL) que les permita intercambiar información y conocimiento, sincronizar acciones o resolver conflictos. El estándar FIPA¹ contiene las especificaciones necesarias para construir SMA escalables, interoperables y comunicativos donde los agentes que puedan localizarse y comunicarse.

La plataforma FIPA-OS fue diseñada para operar en un entorno heterogéneo soportando múltiples tipos de transporte y múltiples codificaciones para el contenido basándose en las especificaciones del estándar FIPA. Además de las componentes obligatorias del estándar FIPA (Directory Facilitator (DF), Agent Management System (AMS), Agent Communication Channel (ACC) e Internal Platform Message Transport (IPMT)), incluye diferentes tipos de *AgentShells* para producir agentes los cuales se pueden comunicar con otros utilizando las facilidades de FIPA-OS, soporte multi-capas para la comunicación entre agentes, gestión de los mensajes y la conversación, configuración

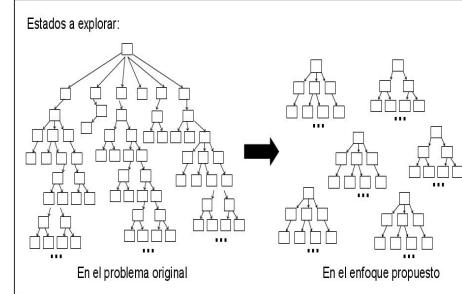


Figura 1: Representación de los escenarios a explorar en el problema original y en el nuevo enfoque propuesto.

dinámica de la plataforma para soportar múltiples IPTM's, múltiples tipos de persistencia y múltiples codificaciones, interfaces abstractas y patrones de diseño de software y herramientas de diagnosis y visualización [1].

4. Metodología

El método que se propone a continuación ha sido definido para poder abordar el crecimiento exponencial de los posibles estados a evaluar a la hora de resolver problemas logísticos.

La idea principal radica en el hecho que puede ser útil dividir el problema en subproblemas locales reduciendo la cantidad de escenarios a explorar en cada uno y por lo tanto poder llegar a soluciones locales óptimas. La figura 1 representa esta idea ilustrando cómo se ve reducido el espacio de búsqueda al dividir el problema. De esta manera, se obtiene que un incremento de los recursos no provoque un aumento exponencial del espacio de búsqueda. Solamente se incrementará la cantidad de nuevos problemas locales, que tendrán un tamaño de espacio de búsqueda similar a los ya existentes.

Sin embargo, distribuyendo el problema de esta manera es muy difícil obtener una solución global para el problema original. Por consiguiente, en este enfoque se ha definido un SMA que permita una interacción eficaz entre las diferentes soluciones locales y poder decidir cuál es la mejor combinación para obtener

¹www.fipa.org

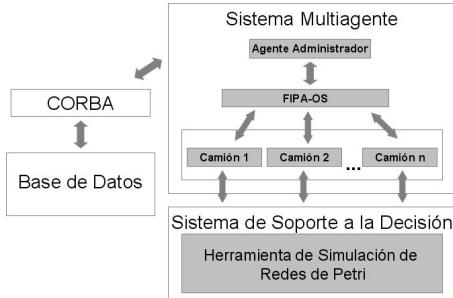


Figura 2: Estructura de la herramienta.

un óptimo o una solución cercana a la óptima para el problema global. Para evitar el problema de la no-estructuración del conocimiento en los SMA, se propone formalizar su comportamiento usando RdPC.

Para ilustrar la metodología propuesta junto con la herramienta de optimización desarrollada (figura 2), se ha utilizado una aproximación al problema de enrutamiento de vehículos donde el modelo del sistema ha sido formalizado mediante RdPC [3]. El SMA está formado por dos tipos de agentes diferentes, Administrador y Camiones, donde la coordinación entre ellos está basada en el protocolo de subasta, donde el comportamiento del subastador es realizado por el administrador y los agentes camión son los compradores. Este concepto de cooperación tiene como ventajas la relación centralizada que permite al subastador tener una visión global del sistema permitiendo evitar conflictos potenciales, el proceso de planificación se simplifica porque no existe negociación entre los compradores y como consecuencia, el coste de comunicación entre los agentes es bajo [13].

La información de sistema, que será el entorno del sistema multiagente, se introduce en una base de datos que se actualiza cada vez que ocurre un cambio en el entorno. Esta información es visible para todos los agentes activos en el sistema pero algunas tablas sólo pueden ser modificadas por el agente administrador. La comunicación entre la base de datos y los agentes ha sido implementada mediante CORBA.

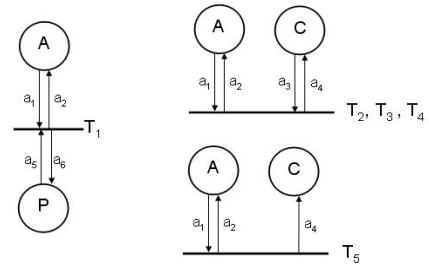


Figura 3: Especificación del comportamiento global del SMA mediante RdPC.

Cada agente camión utiliza la herramienta de simulación de RdPC orientada a objetos para poder explorar los posibles escenarios y poder así lograr el objetivo local.

5. Especificación del SMA mediante RdPC

El comportamiento global de los agentes y sus iteraciones pueden ser modelados mediante las RdPC ilustradas en la figura 3. Los lugares A, C y P representan el agente Administrador, los agentes Camión y las Peticiones respectivamente, y las transiciones T_1 , T_2 , T_3 , T_4 y T_5 simbolizan los diferentes eventos que se pueden activar según el estado de las entidades (control de las peticiones, inicio de un proceso, control de las ofertas realizadas, finalización de un proceso y activación de un nuevo camión respectivamente).

El objetivo global es la asignación de todas las peticiones utilizando la mínima cantidad de camiones posible. Con tal fin, inicialmente el administrador y un único camión están activos. Cuando en la base de datos hay peticiones pendientes, se realiza un primer proceso en el cual el camión que está activo decide las peticiones que puede atender y en el caso que queden algunas pendientes, se van activando nuevos camiones hasta que todas estén asignadas. En este primer proceso, lo más seguro es que se hayan activado más camiones de los necesarios y es aquí donde empieza la subasta de las peticiones.

Cada camión determina su interés por las

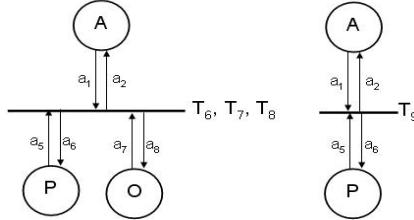


Figura 4: Especificación del comportamiento local del Administrador.

peticiones demandadas y este es el indicador usado por el administrador para decidir cuales son las peticiones que entrarán en un reproceso en el cual participaran todos los camiones activos. De esta forma, el número de camiones que tienen peticiones asignadas se irá reduciendo hasta obtener el mínimo.

En esta definición del comportamiento global del SMA se pueden distinguir dos conductas locales, una para el agente Administrador y otra para los agentes Camión. El papel del agente Administrador es controlar la asignación de las peticiones; por lo que, tiene que controlar el estado de peticiones y las ofertas hechas por los camiones y decidir qué asignaciones se tienen que hacer para poder satisfacer su objetivo local que es asignar la máxima cantidad de peticiones a la mínima cantidad de camiones. La RdPC que describe dicho comportamiento se ilustra en la figura 4 donde T_6 permite decidir el camión al que se le asigna una petición que ha recibido oferta, T_7 se utiliza para controlar las ofertas aceptadas y las que tiene que ser liberadas, T_8 ejecuta la liberación y finalmente T_9 comprueba si queda alguna petición sin atender y decide el siguiente paso a realizar (la especificación de los arcos y los guardas de las transiciones se pueden consultar en [3]).

El objetivo local para cada uno de los agentes Camión es encontrar una buena combinación de peticiones a servir y enviar las ofertas a la base de datos para que el administrador pueda decidir si las acepta. Cuando el camión recibe el mensaje enviado por el Administrador que informa de un nuevo proceso, el camión actualiza su entorno, genera el ar-

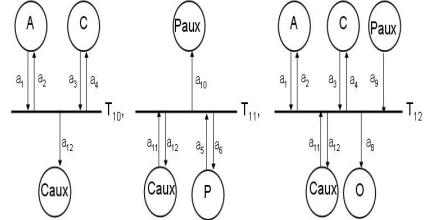


Figura 5: Especificación del comportamiento local del Camión.

chivo ASCII con a especificación de la RdPC que contiene las peticiones a evaluar y ejecuta la herramienta de simulación para obtener las posibles soluciones locales y poder así decidir cual es la mejor. La RdPC que describe el comportamiento del camión es mostrada en la figura 5 donde la transición T_{10} se utiliza para crear y destruir una entidad auxiliar a la del Camión, T_{11} hace lo mismo para las peticiones y T_{12} decide las ofertas que tiene que realizar y su interés [3].

6. Resultados

Para ilustrar la herramienta presentada en este artículo, se presenta un caso particular del VRP basado en un sistema de distribución con un almacén central (Vic) de donde parten los camiones y se deben atender las 19 peticiones descritas en la figura 6. El objetivo, que en este problema se ha alcanzado usando cuatro camiones, es asignar todas las peticiones utilizando el mínimo número de camiones, teniendo en cuenta que todas las peticiones tienen que ser servidas dentro de los intervalos de tiempo fijados por los almacenes.

La dinámica de este sistema se describe mediante la RdPC mostrada en la figura 7. Está formada por tres nodos lugar (C : Camiones, P : Peticiones y R : Rutas) y tres transiciones (T_1 : Acción de Carga/Descarga, T_2 : Desplazamiento y T_3 : Espera).

La primera transición, T_1 , se habilita cuando el camión está en la misma posición que la petición a atender y el tiempo en el que se encuentra el camión está dentro de la ventana de tiempo del almacén. Si es una acción de carga,

Id	Origen	Destino	Peso	Intervalo Carga	Intervalo Descarga	Tiempo C/D
1	Barcelona1	Badalona	50	70 - 90	100 - 120	3/10
2	Terrassa	Manresa	100	280 - 310	350 - 370	3/5
3	Terrassa	Vic	50	280 - 310	420 - 440	3/10
4	Barcelona2	Piera	100	80 - 100	180 - 200	3/10
5	Gavà	Vallirana	90	120 - 140	235 - 255	5/10
6	Molins	Sabadell	50	265 - 285	310 - 330	5/10
7	Molins	Vic	50	265 - 285	390 - 410	4/10
8	Sabadell	Vic	50	310 - 330	390 - 410	4/10
9	Barcelona1	Mataró	50	70 - 90	145 - 165	4/5
10	Barcelona1	Sant Celoni	50	70 - 90	220 - 240	5/10
11	Barcelona1	Vic	40	70 - 90	325 - 345	5/10
12	Mataró	Arenys	20	145 - 165	180 - 200	4/10
13	Mataró	Granollers	20	145 - 165	260 - 280	10/10
14	Barcelona3	Ripoll	50	80 - 100	115 - 135	5/5
15	Barcelona3	Sallent	50	80 - 100	205 - 225	5/10
16	Barcelona3	Terrassa	50	80 - 100	280 - 310	5/10
17	Barcelona4	Martorell	100	80 - 100	130 - 150	10/10
18	Terrassa	Sabadell	100	175 - 195	205 - 225	10/10
19	Muntanyola	Vic	100	280 - 300	315 - 325	10/10

Figura 6: Descripción de las peticiones. Diferentes identificadores de Barcelona corresponden a diferentes códigos postales. Los intervalos de carga y descarga son fijados por el almacén. El peso se muestra en kilogramos.

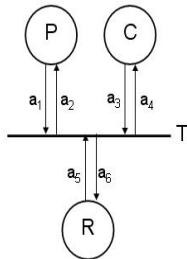


Figura 7: Modelo en RdPC del problema descrito.

un requisito adicional para el camión es tener suficiente espacio libre. Para activar un evento de movimiento (es decir T_2) la única condición previa es la existencia de una ruta que conecte la posición del camión con la de la petición. Finalmente, la transición de espera, T_3 , se habilita cuando el camión llega al almacén antes de su apertura.

El modelo describe todas las posibilidades factibles y como consecuencia el espacio de búsqueda contiene 10^{95} nodos que provocan que no sea factible encontrar la solución en un espacio de tiempo razonable. La adición a la RdPC de conocimiento del sistema evitando la exploración de ramas estériles permite una re-

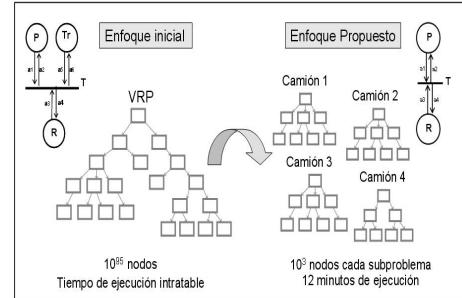


Figura 8: Resultados obtenidos en este artículo.

ducción considerable del espacio de búsqueda y del tiempo computacional siendo aún considerables [2].

El SMA descrito en este artículo divide el problema en subproblemas locales, uno para cada camión. Inicialmente se ejecuta un preproceso que permite asignar todas las peticiones usando 6 camiones. A partir de este momento el agente administrador decide cuales son las peticiones que pueden ser redistribuidas y los agentes camión empiezan a competir por ellas. Esta distribución del sistema original permite obtener la solución óptima al problema reduciendo el espacio de búsqueda a 10^3 nodos para cada subproblema local (los resultados se ilustran en la figura 8). Por consiguiente, debido a la reducción del espacio de búsqueda y al hecho de que todos los agentes camión pueden actuar de forma paralela, el tiempo computacional se ve reducido a 12 minutos.

7. Conclusión

Se ha utilizado un enfoque basado en sistemas multiagente para abordar problemas NP-completos, realizando una distribución del problema que evita que un incremento de recursos en el problema implique un crecimiento exponencial del espacio de búsqueda. Con este enfoque se reduce considerablemente el tiempo computacional.

Las Redes de Petri Coloreadas han sido utilizadas como formalismo permitiendo integrar el sistema multiagente con herramientas de si-

mulación modelando la dinámica del sistema junto con el comportamiento de los agentes.

Para ilustrar la metodología propuesta en este artículo, se ha introducido un enfoque para el problema de enrutamiento de vehículos.

8. Agradecimientos

Este trabajo ha sido realizado con el apoyo del Departamento de Universidades, Investigación y Sociedad de la Información del Gobierno de Cataluña.

También ha sido posible gracias a la Dirección General de Investigación, Subdirección General de Proyectos, a través del proyecto DPI2004-080056-C03-01.

Referencias

- [1] Buckle, P., Moore, T., Robertshaw, S., Treadway, A., Tarkom, S. and Polsad, S., *Scalability in Multi-agent Systems: The FIPA-OS Perspective*, Springer-Verlag Berlin Heidelberg 2002.
- [2] Busquets, S., Vilalta, L., Guasch, T., Piera, M.A. and Narciso, M., *Specification of Metaheuristics in Colored Petri Net Models to Tackle the Vehicle Routing Problem*, European Modeling and Simulation Symposium, EMSS'05, 2005.
- [3] Busquets, S., *Descripció del Comportament del agents en el disseny d'un Sistema Multi-Agent pel VRP*, Internal Report, Dept. TESA, UAB, 2006.
- [4] Huhns, M.N. and Stephens, L.M., *Multiagent Systems and Societies of Agents*, G. Weiss ed. Multiagent systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, pp. 79–120, 1999.
- [5] Jennings, N.R., Sycara, K. and Wooldridge, M., *A Roadmap of Agent Research and Development*, Autonomous Agents and Multi-Agent Systems, Vol. 1, pp. 275–306, 1998.
- [6] Jensen, K. *Coloured Petri Nets: Basics Concepts, Analysis Methods and Practical Use*, vol. 1,2,3, Springer-Verlag, 1997.
- [7] Murata, T., *Petri Nets: Properties, Analysis and Applications*, Proceedings IEEE, 1989.
- [8] Narciso, M.E. and Piera, M.A., *Coloured Petri Net Simulator: A Generic Tool for Production Planning*, ETFA'2001 - 8th IEEE International Conference on Emerging Technologies and Factory Automation, Antibes, 2001.
- [9] Narciso, M.E., Piera, M.A. and Guasch, A., *A Knowledge Representation Approach Suitable to Support Heuristic Search Methods*, Frontiers in Artificial Intelligence and Applications: Artificial Intelligence Research and Development, 100:397-408, IOS Press, 2003.
- [10] Narciso, M.E., Piera, M.A. and Guasch, A., *A Decision Support System to Tackle the Vehicle Routing Problem*, Catalonian Conference on AI, 2004.
- [11] Piera, M.A. and Guasch, A., *Coloured Petri Net for Simulation FMS Model Maintenance*, EUROSIM'01, Delft, 2001.
- [12] Piera, M.A., Narciso, M.E., Guasch, A. and Canovas, J., *A Methodology to Improve Simulation Optimization Approach for Process Scheduling through Petri Net Formalism*, Summer Computer Simulation Conference (SCSC'03), 2003.
- [13] Roggero, D., Patrone, F. and Mascardi, V., *Designing and Implementing Electronic Auctions in a Multiagent System Environment*, WOA'05, 2005.
- [14] Timben-Lembeke, R.S. and Rogers, D.S., *The Council of Logistics Management*, RLEC: Reverse Logistics Executives Council, pp. 8, 1998.
- [15] <http://neo.lcc.uma.es/radi-aeb/WebVRP>, *The VRP Web*, (2007).
- [16] Yang, B., Geunes, J. and O'Brien, W.J., *A heuristic approach for minimizing weighted tardiness and overtime costs in single resource scheduling*, Computers and Operations Research, Article in Press, 2003.

Distribución del espacio de estados de una Red de Petri Coloreada

Laia Vilalta , Miquel Angel Piera

Dept. de Telecomunicació i d'Enginyeria de Sistemes

Universitat Autònoma de Barcelona

Campus UAB - Edifici Q, 08193 Bellaterra

Laia.Vilalta@uab.cat , MiquelAngel.Piera@uab.cat

Resumen

La logística se ha convertido en un aspecto muy importante en cualquier industria que tenga como objetivo reducir stocks y servir las demandas a tiempo. La gran cantidad de variables que aparecen en un problema logístico y de relaciones entre ellas hace que sea necesaria una herramienta de toma de decisiones. Es imprescindible que esta herramienta sea capaz de dar respuesta dentro del intervalo de tiempo requerido. Es común en las herramientas comerciales considerar una simplificación del problema ya que el tiempo de cómputo puede ser muy grande. Para poder dar respuesta a los problemas logísticos sin tener que recurrir a una simplificación que no pueda garantizar la optimalidad, se ha desarrollado un simulador de RdPC. En este artículo se exponen las ventajas que conforma el diseño de este simulador respecto otras aproximaciones en relación al tiempo de cómputo y al uso de memoria.

1. Introducción

Desde hace años la industria está experimentando cambios, que afectan de forma importante al mundo del transporte y la distribución pero también a la logística en general. Actualmente la demanda cambia de forma constante, tanto en la cantidad de producto de algún tipo, como en la variedad de éste. Para poder ser competitivo en el mercado es vital tener una capacidad de respuesta muy rápida. Para ello se dispone, entre otros, de hardware flexible, bases de datos, comunicaciones. Aun así,

no existen herramientas de apoyo a la decisión que ofrezcan una solución óptima considerando objetivos operacionales, estratégicos y tácticos [4].

Tradicionalmente se han utilizado diferentes métodos para dar respuesta a problemas logísticos. El modelado y la simulación han demostrado ser muy útiles en el diseño estratégico y táctico. Aun así, presentan algunas limitaciones. En un problema con complejidad NP, la simulación no puede evaluar más que una fracción del inmenso rango de opciones posibles [10]. También es usual la utilización de métodos de optimización. Éstos se basan en formalizar el problema en términos de una función objetivo, o función a maximizar o minimizar, y un conjunto de restricciones sobre las variables que aparecen en dicha función. Cuando se trata con un número elevado de variables y relaciones entre ellas, es común, simplificar el problema. En problemas logísticos la solución óptima para una simplificación no tiene porqué ser un óptimo para el original. Es más, puede ser que esta solución no cumpla los requerimientos del problema inicial [8].

En este contexto, se ha decidido utilizar una metodología que permita una integración de optimización, modelado y simulación al mismo tiempo que permite la inclusión de heurísticas. En este artículo, se presenta tal metodología basada en el análisis de Redes de Petri Coloreadas a partir del árbol de alcance. Esta metodología aunque presenta ciertas ventajas respecto otras, también tiene algunas limitaciones. Éstas se refieren básicamente a cuestiones de tiempo y de memoria. En las seccio-

nes 3 y 4 se presentan las propuestas de este trabajo para hacer frente a tales limitaciones.

2. Modelos para la toma de decisiones

Las Redes de Petri (RdP) se consideran una herramienta de gran ayuda para el modelado de sistemas, especialmente en el caso de sistemas secuenciales y concurrentes [7].

Una *Red de Petri* es un grafo dirigido, bipartito y ponderado con dos tipos de nodos: lugares (representados gráficamente por un círculo) y transiciones (representados por un rectángulo); y arcos (flechas) que unen dos nodos de tipos diferentes [6]. Al modelar un sistema, cada nodo lugar representa un estado de un elemento del sistema. Por ejemplo, puede simbolizar un stock o el estado libre de una máquina. Cada nodo transición representa un evento que puede ocurrir en el sistema o al sistema. Por ejemplo, puede simbolizar la llegada de piezas o el inicio de una operación. Los arcos representan las precondiciones y postcondiciones de cada evento.

Por su condición de modelo gráfico, las Redes de Petri son una herramienta que permiten una rápida interpretación de las causas y las consecuencias de cada decisión tomada sobre el sistema. En otras palabras, es fácil analizar los factores que intervienen sobre cada evento. Por ello, son un instrumento útil para la toma de decisiones. En contrapartida, las RdP clásicas presentan algunas limitaciones cuando modelan sistemas complejos donde intervienen muchas componentes o variables. Uno de los problemas más importantes que presentan el mantenimiento un modelo cuando se añaden nuevas variables de interés. Por este motivo una RdP para un problema logístico no sería tratable. En consecuencia, se han considerado las Redes de Petri Coloreadas.

Las Redes de Petri Coloreadas (RdPC) son un formalismo que parte de la base de las RdP clásicas y añade tipos de datos y modularidad. Así, las RdPC constituyen una buena herramienta en los sistemas logísticos por su condición gráfica pero también por la posibilidad de evaluación el modelo utilizando técnicas heu-

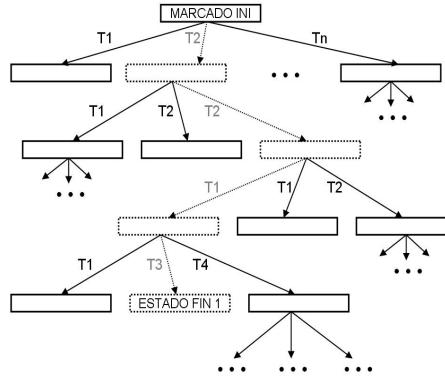


Figura 1: Estructura general de un árbol de alcance

rísticas [12]. Las RdP son modelos fáciles de mantener cuando se modelan actividades paralelas de un sistema. Las RdP han sido usadas generalmente en decisiones tácticas y operacionales, pero se ha podido comprobar que también pueden dar buenos resultados en el campo estratégico [5].

Igual que en las RdP, dado un cierto estado del sistema es posible analizar los eventos que se encuentran activos y los estados que pueden ser alcanzados al disparar cada uno de estos eventos. Para analizar los diferentes estados o marcas que se pueden alcanzar dado un estado inicial, se utiliza el árbol de alcance. El hecho de poder tener gráficamente todos los estados que se pueden alcanzar a partir de un estado inicial permite identificar fácilmente todos los caminos que llevan desde tal estado hasta el estado final deseado. Se puede constatar si el estado final es alcanzable y si se corresponde con más de un marcado.

En la figura 1 se muestra un ejemplo de un árbol de alcance. Cada uno de los rectángulos se corresponde a un estado del sistema o marcado. Las flechas relacionan cada estado con el estado que se genera al disparar la transición correspondiente a la etiqueta de las mismas. El camino marcado de con punteado hace referencia al conjunto de decisiones que se deberían tomar para llegar a un estado final. Una vez abierto todo el árbol, podríamos decidir si este estado final se corresponde con el óptimo.

Uno de los mayores inconvenientes de esta herramienta de análisis es su tamaño. En un problema de complejidad NP el árbol es demasiado extenso para poder abrirlo completamente. Al aplicar esta aproximación a la solución de un problema logístico, esto se traduce en no poder explorar el árbol completamente por lo que no se puede asegurar que, si se encuentra una solución, ésta sea la óptima.

3. Entorno distribuido

La exploración y generación del espacio de estado para hallar una solución a un problema NP puede ser intratable. Por un lado hay un gasto computacional importante a la hora de evaluar, para cada estado, que transición o transiciones pueden dispararse. Esto es debido a la gran cantidad de variables que intervienen en cada una de las decisiones o transiciones. Una forma de reducir el tiempo de cómputo es distribuir el sistema. La distribución de la simulación de eventos discretos permite obtener una ejecución del proceso más rápida [9].

La distribución que se ha adoptado se corresponde con un modelo híbrido master/worker. Un modelo master/worker se constituye de un solo hilo de control (master) que dirige y envía la información que cada uno del resto de hilos (workers) debe procesar. La única tarea del master es esperar a que todos los workers hayan respondido. El modelo híbrido se basa en el master/worker permitiendo un mayor paralelismo. Con este fin el master determina la carga de trabajo de los workers y cada uno de éstos trabaja con la carga de trabajo independientemente del resto de workers i del master [13]. En la figura 2 se muestra un esquema de funcionamiento del modelo híbrido utilizado.

Hay diferentes maneras de distribuir una red de Petri para que la simulación del sistema modelado resulte más rápida. Una posible forma de distribución hubiera sido dividir la red en subredes. Entonces se simularía cada una de las subredes en computadoras diferentes. El problema de este enfoque es que implica una sincronización entre las redes ya que tendrían, en todos los casos, nodos lugar compartidos.

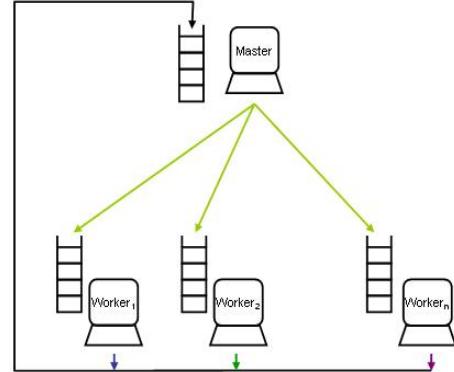


Figura 2: Esquema de funcionamiento de un modelo híbrido master/worker

La forma de distribución escogida es la separación de evaluación de transiciones de la red de Petri. Aparentemente no hay diferencia entre este método y la separación en subredes. No obstante, la diferencia reside en la formulación de esta distribución. Así como en la separación en subredes la distribución hace referencia a la red y distribuye el árbol de alcance, en la separación por transiciones, la distribución se refiere a la evaluación de éstas y por lo tanto el árbol está centralizado. Esta elección supone ventajas importantes. Una de ellas es la no necesidad de sincronización [12].

En la generación del árbol de alcance, dado un estado la evaluación de una de las transiciones es independiente a la evaluación de otra. Además, también es independiente de la evaluación de una transición respecto otro estado diferente. Esto permite, por un lado, disponer de un alto grado de paralelismo y, por otro, la flexibilidad de no haber de verificar la causalidad durante la ejecución del simulador.

4. Gestión de la memoria

Una de las mayores desventajas del análisis del árbol de alcance de una RdPC correspondiente a un sistema logístico es el hecho que el árbol puede crecer de forma exponencial en relación al número de variables que intervienen en el sistema. Si se está tratando un problema lo-

gístico es más que probable de que sea un problema con complejidad NP, lo que implicaría un intratable número de nodos a explorar.

Dos de los objetivos de este trabajo son asegurar optimalidad y poder conocer las decisiones que deben tomarse desde el estado inicial indicado hasta el estado final que proporciona tal optimalidad. Es por ello que no se han considerado métodos de almacenamiento como el sweep-line method [2] o el condensed state space [3].

Como es bien conocido, el disparo de una transición implica el cambio en el marcado de la red. En otras palabras, la ocurrencia de un evento implica algún cambio en el estado del sistema. Tradicionalmente en el almacenaje del árbol de alcance se guarda cada uno de los estados de forma completa. Muchas veces, una transición no afecta a todos los lugares de la red, y por lo tanto, hay algunos lugares que tienen el mismo marcado antes y después del disparo. La propuesta a la que se remite este trabajo va dirigida hacia el intento de no almacenar completamente dos estados parcialmente iguales.

Considérese una RdPC que conste de dos nodos lugar y de una sola transición. Supóngase además que, en un momento dado de la generación del árbol de alcance, se obtiene un marcado que es igual a uno anterior en lo que refiere al marcado de un nodo lugar. Con tal de utilizar menos memoria, se puede intuir que se podría almacenar solamente aquella parte del marcado que constituye una diferencia. Es decir, solamente el marcado correspondiente al nodo lugar donde se ha constituido la diferencia. De la misma manera se puede apreciar fácilmente que esta solución conlleva el peligro de no poder recuperar alguno de los dos marcados. Por este motivo se ha desarrollado un sistema de identificación a la estructura de datos de cada nodo.

Dado un marcado, correspondiente a un estado, la metodología seguida para guardarlo en memoria es la siguiente. Primero se separa en submarcados por nodos lugar. Así, si la RdPC consta de n nodos lugar, se obtienen n marcados. Para cada uno de ellos se procede de la misma manera. En la especificación de

la RdPC además del detalle propio de la red se pide puntualizar que colores de cada nodo lugar son importantes. Estos colores se corresponden con los colores que influyen de manera directa o indirecta en las decisiones que se pueden tomar a partir de la red, i.e. en el disparo de las transiciones. Un ejemplo de un color no importante podría ser el respectivo al identificador de un camión en una RdPC correspondiente a un problema VRP (Vehicle Routing Problem). El identificador del camión, o el kilometraje que lleva, puede ser importante en algún caso: si se quiere controlar el recorrido de un camión concreto, o establecer algún tipo de prioridad sobre los camiones menos rodados. El hecho de poder distinguir qué información es importante y cual no, permite mantener un mismo modelo evaluar para diferentes decisiones. Al guardar cada submercado, se separa el conjunto de colores que son importantes del conjunto que no lo son. De esta manera, si en otro marcado el conjunto de colores importantes se repite, se podrá considerar que es un marcado ya evaluado puesto que producirá marcados con colores importantes iguales a los producidos por el primero.

Con el fin de poder ilustrar de una forma más clara el diseño de tal estructura, considérese el problema VRP siguiente [11]. Considérese un sistema con un almacén central desde donde todos los vehículos parten. Para simplificar se supone que existen tan solo tres peticiones y que se dispone de dos camiones. El objetivo de este problema consiste en ver que asignación se debe hacer a cada camión para que las peticiones sean servidas lo antes posible. El modelo formulado a través de una RdPC consta de tres nodos lugar: camiones, peticiones y rutas; y de tres transiciones: carga/descarga, movimiento y espera.

La información (o colores) necesaria para cada nodo lugar es la siguiente. Para el nodo C (camiones) se definen los atributos: identificador, capacidad libre, posición y tiempo. Los colores correspondientes al nodo lugar P (peticiones) son: identificador, peso, almacén de origen, almacén destino, horario de apertura y cierre de cada uno de ellos, tiempos de carga y descarga e identificador del camión a la que ha

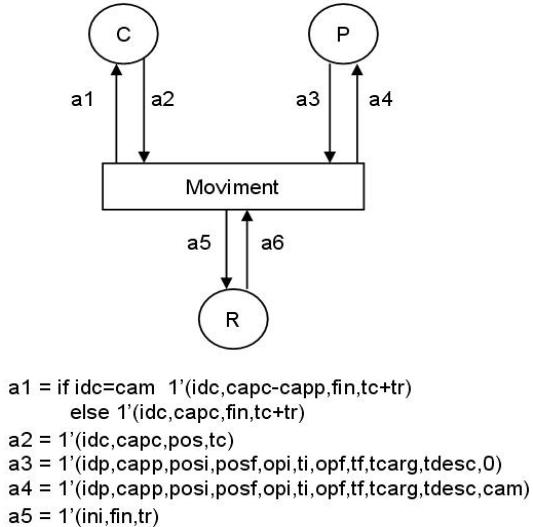


Figura 3: Transición de movimiento para un VRP

sido asignada. Finalmente las marcas del nodo lugar R (rutas) contienen información sobre la posición inicial y final del itinerario así como el tiempo requerido para realizarlo.

Ya que el objetivo de este ejemplo es mostrar la metodología de almacenamiento de los marcados y no dar una solución al VRP propuesto, este ejemplo se centrará en el marcado del nodo lugar C y solamente en la transición de movimiento. Tal como ya se ha comentado, el primer paso de dicha metodología es separar el marcado por nodos lugar. Por ello, la exposición del almacenamiento correspondiente a un solo lugar es generalizable. Se supone que en el nodo lugar R hay las rutas correspondientes a todas las combinaciones posibles. En la figura 3 se muestra la parte de la RdPC correspondiente.

En este ejemplo se tiene que el estado inicial para el nodo C es: $1'(1, 200, 0, 0)+1'(2,200,0,0)$. Supóngase que una de las peticiones tiene la siguiente información $1'(1,50,2,5,0,30,20,50,5,5,0)$ y supóngase que la ruta que une los puntos 0 y 2 tiene la siguiente información $1'(0,2,70,5)$. Entonces, disparar la transición con el fin de que el primer camión se mueva para atender a tal

peticIÓN y con el fin de que lo haga el segundo, llevará a los marcados siguientes para el lugar C: $1'(1,150,2,70,5)+1'(2,200,0,0)$ y $1'(1,200,0,0)+1'(2,150,2,70,5)$ respectivamente.

Estos dos estados pueden considerarse iguales ya que se corresponden a que un camión se haya desplazado del punto de origen al almacén de carga de una petición. Si se considera que el identificador del camión no influye en la toma de decisiones, entonces se puede suponer que los dos estados son simétricos. No obstante, al construir el árbol de alcance de forma tradicional, los dos estados se considerarían diferentes. Para seguir la metodología que se presenta en este trabajo, se deben separar primero los colores importantes de los que no lo son. Se considera que, en este ejemplo, los colores importantes (para el nodo lugar C) son todos excepto el primero. Una vez separados, si se pueden agrupar tokens por tener los mismos colores importantes, se procede. A partir de este punto con la palabra marcado se hará referencia al submarcado del nodo lugar C.

Así pues el marcado inicial queda guardado como: $2'(200,0,0)$ y $1'(1)+1'(2)$. De estas dos estructuras, la primera hace referencia al hecho que tenemos 2 camiones con capacidad 200 en la posición 0 y a tiempo 0. La segunda estructura nos dice que uno de los camiones tiene identificador 1 y el otro 2. Estos dos números se corresponden a los valores del color no importantes. Aunque no sean importantes deben almacenarse ya que forman parte del modelo.

El marcado $1'(1,150,2,70,5)+1'(2,200,0,0)$ quedaría guardado como: $1'(150,2,70,5)+1'(200,0,0)$ y $1'(1)+1'(2)$. Y el marcado $1'(1,200,0,0)+1'(2,150,2,70,5)$ como: $1'(150,2,70,5)+1'(200,0,0)$ y $1'(2)+1'(1)$. La figura 4 muestra un esquema de como se guarda tal estructura para consumir menos memoria y la compara con el almacenaje tradicional. En esta figura también se puede apreciar que el segundo marcado está marcado como old. Esto implica que no deberá abrirse el árbol de alcance por debajo de él.

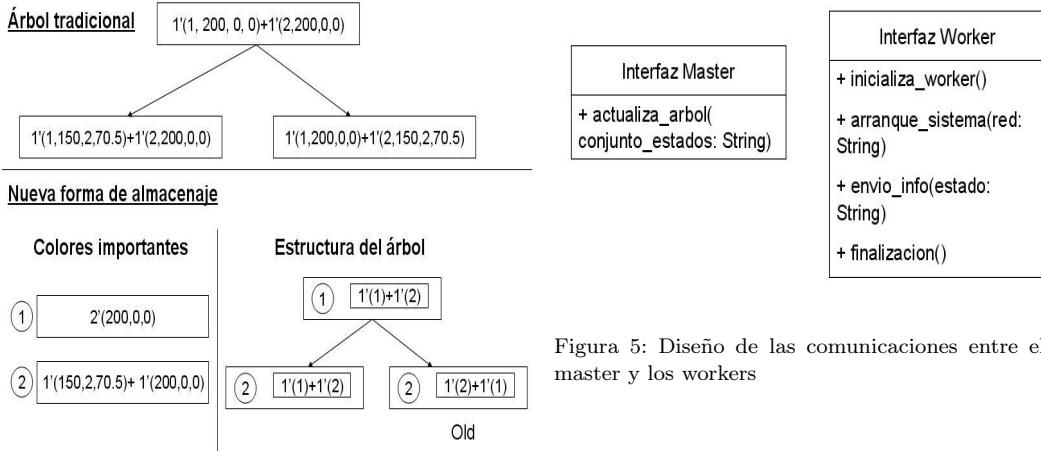


Figura 4: Esquema de la nueva estructura de almacenamiento

5. Aspectos funcionales

Inicialmente para generar el árbol de alcance se parte de un estado inicial, juntamente con la especificación de la RdPC. El master recibe esta información, inicializa la estructura de datos con el estado inicial y envía la información correspondiente a cada transición y el estado inicial a cada uno de los workers que dirige. Una vez finalizada esta etapa de inicialización, empieza la etapa de funcionamiento habitual. Durante esta etapa, cada worker evalúa la transición a la que se corresponde con los estados que el master le ha enviado. Una vez hallados los estados que genera un estado concreto al disparar la transición, el worker los envía al master. En caso que la transición no se pueda disparar, lo indica al master con un mensaje similar. El funcionamiento del master se basa en la recepción de marcados y su correspondiente almacenaje.

El protocolo de comunicación ha sido definido a través de dos interfaces. El primero se corresponde al envío de información por parte del master a cada uno de los workers. Las funciones son: inicialización, arranque, envío de información y finalización. El segundo se corresponde al envío de información por parte de los workers al master. La única función

Figura 5: Diseño de las comunicaciones entre el master y los workers

que contiene es la de transmisión de datos. Es importante que en el remite de un estado del master a un worker se adjunte el código identificador de tal estado. De esta manera cuando el worker responda con los correspondientes estados generados, pueda añadir tal información y el master pueda interpretar como debe guardar los estados en el árbol de alcance. La figura 5 muestra los interfaces de comunicación.

El diseño de la estructura que almacena el árbol de alcance, se muestra gráficamente en la figura 6. Para poder entender más fácilmente a qué se refiere cada de las clases, se toma como referencia el ejemplo de la sección anterior. El marcado de un nodo lugar se almacena en dos estructuras, una conteniendo los colores denominados importantes y la otra, el resto. La primera estructura se traduce en la clase TokenArbre, la segunda, en Resta. El conjunto de las dos estructuras, mostrado en la figura 1 como un elemento del árbol, se traduce en la clase Node. El conjunto formado por un estado y todos los estados que se pueden generar a partir de éste disparando alguna transición, conforman un objeto de la clase RelArbre. Placa contiene el conjunto de objetos TokenArbre que se han obtenido en algún momento en un nodo lugar determinado. Cada EstatFinal representa la especificación del estado final para uno de los lugares. Finalmente, Arbre es la clase que representa el árbol de alcance.

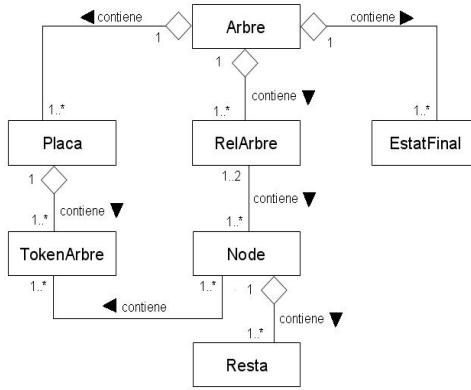


Figura 6: Diseño de la estructura del árbol de alcance

6. Resultados

Con el fin de poder ilustrar de una forma más clara el diseño de tal estructura considérese el ejemplo del Job-Shop 6x6. En líneas generales, este problema consiste en programar 6 tareas a 6 máquinas donde cada tarea tiene un orden concreto. Para una información más detallada de este problema se puede consultar el artículo [8]. Con tal de poder evaluar la distribución del sistema en transiciones, se ha dividido la transición en 6 transiciones. En el problema original la transición simboliza la ejecución de una tarea. En este ejemplo, cada una de las transiciones representa la ejecución de una tarea en cada una de las máquinas. Por ejemplo, evaluar la primera transición equivale a comprobar si la primera máquina puede realizar algún trabajo. La especificación de cada uno de los arcos, guardas y estado inicial, se encuentra en el artículo previo [12].

Para construir el árbol de alcance de dicho problema, se ha utilizado la estructura de distribución master/worker híbrido el funcionamiento del cual ha sido antes especificado. Cada uno de los workers ha evaluado una de las 6 transiciones. Así, el master ha podido construir el árbol de alcance a partir de los marcados que cada uno de los worker ha ido obteniendo. Éstos, a su vez, han generado los estados que se obtienen disparando la transición

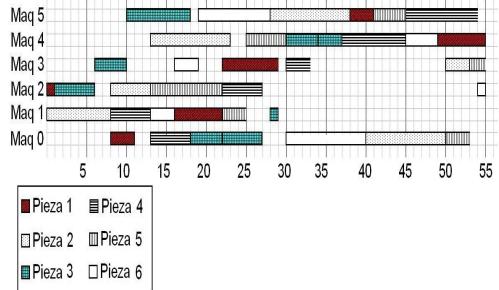


Figura 7: Diagrama de Gantt de la solución encontrada para el JobShop 6x6

correspondiente y partiendo de cada uno de los estados enviados por el master.

El problema Job-Shop 6x6 en un entorno no distribuido consume 75 minutos para poder explorar el árbol de alcance completamente. Esto asegura que, al haber encontrado todas las soluciones posibles, se pueda distinguir la solución óptima. El mismo problema formulado tal como se presenta anteriormente, con una RdPC formado por 6 transiciones, ha necesitado aproximadamente 15 minutos para poder llegar a los mismos resultados. En la figura 7 se muestran los resultados obtenidos a través de un diagrama de Gantt.

7. Conclusiones

La estructura para almacenar la información referente al árbol de alcance ha sido diseñada con tal de utilizar menos memoria. Dependiendo de la cantidad de colores importantes en cada nodo lugar y de la variabilidad de sus valores en relación con el resto de nodos lugar dependerá el decremento de uso de memoria. Por otro lado también contribuye en ese sentido el posible aumento de nodos old. Esto implica una reducción notable del árbol de alcance.

La distribución física de la evaluación por transiciones permite un mayor rendimiento y una obtención más rápida de resultados.

8. Agradecimientos

Este trabajo ha sido realizado con el apoyo del Departamento de Universidades, Investigación y Sociedad de la Información del Gobierno de Cataluña.

También ha sido posible gracias a la Dirección General de Investigación, Subdirección General de Proyectos a través del proyecto DPI2004-080056-C03-01.

Referencias

- [1] Aguilar, R.M., *Diseño de una herramienta para la Simulación Paralela de Sistemas de Eventos Discretos en la Red*, XXII Jornadas de Automática, 2001.
- [2] Christensen, S., Kristensen L. M., Mailund T., *A Sweep-Line Method for State Space Exploration*, Lecture Notes in Computer Science, 2001.
- [3] Christensen S., Kristensen L. M., Mailund T., *Condensed State Spaces for Timed Petri Nets*, Conference on Application and Theory of Petri Nets, 2001.
- [4] Guasch, A., Ramos, J.J., *Sección Especial: Metodologías de Modelado y Simulación para la Mejora de Sistemas Logísticos y de Producción*, Revista Iberoamericana de Automática e Informática Industrial, 2005.
- [5] Jiménez, E., Pérez, M., Sanz, F., *Modelado y Simulación de Sistemas Logísticos y de Producción Mediante Redes de Petri*, Revista Iberoamericana de Automática e Informática Industrial, 2005.
- [6] Guasch, A., Piera, M.A., Casanovas, J., Figueras, J., *Modelado y Simulación. Aplicación a Procesos Logísticos de Fabricación y Servicios*, Edicions UPC, 2003.
- [7] Jiménez, E., *Redes de Petri de Supervisión y Simulación de Procesos Industriales Automatizados*, XXI Jornadas de Automática, 2000.
- [8] Narciso, M.E., Piera M.A., Figueras J., *Optimización de Sistemas Logísticos Mediante Simulación: Una Metodología Basada en Redes de Petri Coloreadas*, Revista Iberoamericana de Automática e Informática Industrial, 2005.
- [9] Park, J., Moraga, R., Rabelo, L., Dawson, J., Marin, M., Sepulveda, J., *Addressing Complexity using Distributed Simulation: A Case Study in Spaceport Modeling*, Winter Simulation Conference'05, 2005.
- [10] Piera, M.A., Guasch, T., Vilalta, L., Mújica, M., *Several Hints for a Faster Search using Coloured Petri Net Simulators*, Summer Computer Simulation Conference, 2006.
- [11] Shaw, P., *Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems*, Lecture Notes in Computer Science, 1998.
- [12] Vilalta, L., Piera, M.A., Narciso, M., Guasch, T., *Distribution of the Search Space for a Colored Petri Net Simulator*, Proceedings on the EMMS'06, 2006.
- [13] Vajracharya S., Beckman P., Karmesin S., *A Programming Model for Clusters of SMPs*, International Conference on Parallel and Distributed Systems, 1999.

Integración de modelo gráfico virtual y modelo de toma de decisiones para gestión basada en simulación de una bodega

Mercedes Pérez De la Parte Departamento de Ingeniería Mecánica, Universidad de La Rioja mercedes.perez@unirioja.es	Emilio Jiménez Macías Departamento de Ingeniería Eléctrica, Universidad de La Rioja Emilio.jimenez@unirioja.es	Marta Vega Ollora Dept. de Ingeniería Eléctrica, Universidad de La Rioja
---	---	--

Resumen

En este artículo se ilustra el desarrollo de una aplicación de gestión de la producción, que busca optimizar el sistema productivo con la ayuda de la simulación, basada en la integración del modelo del estado del sistema para la toma de decisiones con un modelo gráfico de realidad virtual, y a su vez integrados dentro del sistema de aplicaciones de gestión de la empresa (una bodega vitivinícola de denominación de origen Rioja).

Ello ha permitido no sólo optimizar el sistema productivo sino además una mejor sinergia entre los diversos departamentos involucrados (ingeniería, producción, gestión, almacenaje,...), ya que es totalmente intuitivo visualizar y analizar las consecuencias de cada decisión en el entorno de producción virtual desarrollado.

Para su desarrollo se han empleado modelos discretos e híbridos de redes de Petri, diseños tridimensionales realizados con Solid Edge, entornos virtuales desarrollados en VRML (virtual reality language modeling), sistemas de gestión llevados a cabo con PHP con MySQL, así como técnicas de inteligencia artificial para la toma de decisiones en los modelos de producción virtuales.

1. Introducción

Los sistemas productivos actuales, con altos grados de automatización, pero a su vez con necesidad de adaptación constante a condiciones variables (pedidos, lotes, suministros, etc.), deben ser gestionados de la manera más eficiente posible para intentar optimizar la producción en base a los recursos disponibles (mano de obra, materias

primas, maquinaria, almacén, etc.) y a los resultados deseados.

Para la búsqueda de esa optimización, a veces conseguida y otras veces sólo aproximada, la simulación puede ser una herramienta muy importante, basada en modelos del sistema productivo en los que se estiman las consecuencias de las decisiones que se pueden tomar, para elegir la más apropiada y llevarla a cabo más adelante.

Esos modelos, construidos para la simulación de la toma de decisiones, normalmente son manejados por personal especializado, ya que se basan en formalismos de modelados complejos para alguien no experto (sistemas de ecuaciones lineales, no lineales, o diferenciales, diagramas de Forrester, etc., para sistemas continuos, o redes de Petri, diagramas de estado, etc. para sistemas discretos). Esto ocasiona muchas veces que el personal encargado de la toma de decisiones sea reacio a la utilización de los modelos del sistema para la optimización mediante simulación.

La forma en la que se ha solucionado dicho problema en el último trabajo desarrollado desde nuestro grupo de investigación para la optimización del sistema productivo de una empresa, en concreto una importante bodega vitivinícola de la denominación de origen de Rioja, ha sido la integración de dicho modelo con un modelo gráfico tridimensional que representa, con todo lujo de detalles, las decisiones tomadas en la simulación.

Dicho modelo gráfico tridimensional está basado en herramientas de diseño, simulación y realidad virtual, en concreto en el lenguaje de realidad virtual VRML (virtual reality modeling

language) y el paquete de diseño tridimensional Solid Edge.

La utilización de dichas herramientas ha permitido además modelar en su entorno ciertas características continuas (llenado de depósitos, avance de cintas transportadoras, etc.) que por tanto han podido ser descargadas del modelo para la toma de decisiones, con lo que dicho modelo se ha transformado de híbrido (continuo y discreto) en discreto, recibiendo del modelo gráfico los eventos correspondientes a las características modeladas de forma continua (el depósito se ha llenado, una barrica ha llegado al final de la cinta transportadora, etc.).

A dicho modelo se le ha implementado una herramienta de gestión, basada en PHP con MySQL, de forma que la simulación puede realizarse en base a las órdenes de producción ya lanzadas en días anteriores o el mismo día, y una vez tomada la decisión de manera firme se generan las siguientes órdenes de trabajo así como la reestructuración de los trabajadores en las zonas y los puestos. Es decir, la toma de decisiones puede hacerse previamente, o al instante (on-line), y una vez realizada se convierte en órdenes de trabajo.

La utilización de esta herramienta de gestión para la optimización de la producción por simulación, basada en la integración del modelo del estado del sistema para la toma de decisiones junto a un modelo gráfico de realidad virtual conectado con las aplicaciones de gestión de la empresa, ha permitido no sólo optimizar el sistema productivo sino además una mejor sinergia entre los diversos departamentos involucrados (ingeniería, producción, gestión,

almacenaje,...), ya que es totalmente intuitivo ver las consecuencias de cada decisión en el entorno de producción virtual desarrollado.

En este artículo se presenta tanto la parte más relacionada con la informática en proyecto, como es la realización del entorno virtual, su adaptación a internet, y su integración en el sistema de gestión mediante PHP y MySQL, como la parte más relacionada con el modelado para la simulación, concretamente las características de los modelos del sistema que se han desarrollado, para los que se ha empleado el formalismo de las redes de Petri, concretamente redes de Petri híbridas coloreadas, con diferentes colores tanto en la parte discreta como en la continua.



Figura 1. Vista inicial del entorno de la bodega.

2. El entorno virtual

Dado que la Realidad Virtual [1][2] se centra en simular por ordenador entornos tridimensionales, en los que el usuario es capaz de visualizar y



Figura 2. Vista general del entorno virtual y detalles.



Figura 3. Ejemplos de interacciones en el mundo virtual.

actuar sobre los objetos de la escena, se ha creado un *mundo* que además de acogerse a la definición anterior, cumple con todas las especificaciones definidas en la idea inicial. Para ello, se seleccionó VRML, que se trata de un lenguaje de programación orientado a Internet dirigido a la creación de entornos tridimensionales de realidad virtual. Sus características básicas son:

- Dispone de independencia de plataformas.
- Se apoya en una serie de recursos gratuitos (visualizadores, constructores...).

- Ofrece facilidad de aprendizaje.
- Es un gran recurso para la publicidad comercial apoyada en páginas web.
- Constituye un escalón hacia el manejo de información en 4D.

Con el fin de lograr un entorno lo más realista posible, se aplicaron mapeados con las texturas originales (obtenidas mediante fotografías), analizando y respetando los sombreados existentes, así como las dimensiones establecidas en todos los elementos del entorno

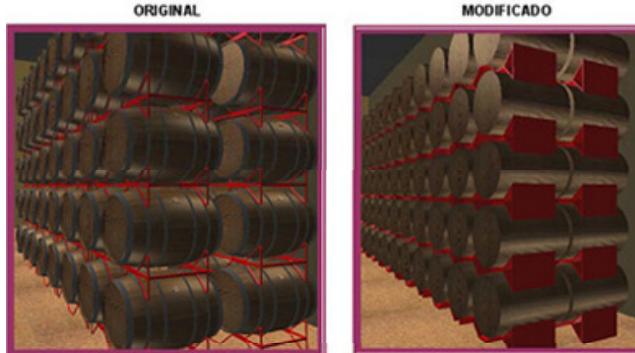


Figura 4. Comparación entre el modelo original y la versión esquemática.

mantienen su fidelidad con la bodega real (vea la Figura 2).

No hay que olvidar la inclusión de interacciones [3][4] en el mundo, de modo que la visita del usuario, además de disponer de total movilidad por el entorno a su voluntad, ofrece un carácter interactivo.

Como ejemplos de interacción se pueden citar dos tipos claramente diferenciados, como se ilustran en la Figura 3:

- Elementos estructurales, cuya interacción sirve para la navegación por el entorno virtual con totalrealismo (puertas, accesos, etc.)
- Elementos de control de procesos (bombas de llenado o de trasiego, accionamientos de temperaturas, etc.) a los cuales se accede desde el entorno virtual a través de la pantalla de ordenador de la sala barricas, la cualenlaza con la gestión informatizada de las barricas.

El empleo del programa Solid Edge [5] (Figura 5), dedicado al diseño profesional de conjuntos mecánicos y piezas directamente en 3D, permitió crear la máquina de tratado de barricas de manera virtual, incorporando opciones de detección de colisiones entre piezas, movimientos en tiempo real, y otras muchas opciones que siguen rigurosamente las Normas ISO. Debido a la gran cantidad de detalle aplicado, se requería de un equipo con elevadas prestaciones. Por lo cual se compuso una nueva versión más esquemática para que la aplicación pudiera ser visualizada en el mayor número de computadores.

En la Figura 4 se muestran las dos opciones disponibles.

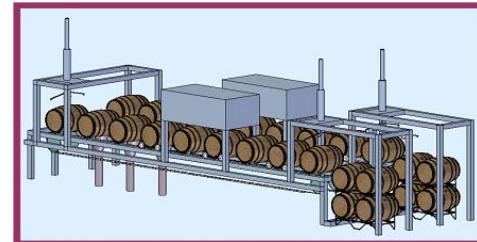


Figura 5. Diseño virtual de la máquina de tratado de barricas realizado en Solid Edge.

3. Base de datos.

Centrando ahora la atención en la informatización de la gestión de las barricas, se pensó en el lenguaje PHP y en la base de datos MySQL [6][7] como la mejor herramienta para lograr el objetivo perseguido. Algunas características que apoyaron la elección de PHP fueron:

- Es un lenguaje multiplataforma.
- Se ejecuta en el servidor.
- Tiene licencia de software libre.
- Tiene una sintaxis cómoda, es similar a C.
- Excelente soporte de acceso a bases de datos.

La aplicación del Control de Gestión de Barricas está dotada de acceso restringido, es decir, se requiere la entrada por teclado de una clave contraseña predefinida anteriormente en el sistema, para que la aplicación ponga a

disposición del usuario el cuadro de opciones (ver Figura 6).



Figura 6. Apariencia inicial de la interfaz del sistema de Gestión de barricas

Una vez validado el código introducido, se muestra un abanico de opciones relacionadas con el control de las barricas, así como información perteneciente a usuarios del sistema.



Figura 7. Opciones disponibles.

Las opciones que pertenecen a la gestión de las barricas y que se muestran en la Figura 7 son:

- Añadir una nueva barrica.
- Visualizar un listado de las barricas registradas con posibilidad de borrar o editar alguna de ellas.
- Obtener un gráfico de barras que ilustra de manera porcentual y gráfica el número de barricas que contienen cada tipo de vino.

Así como las opciones relacionadas con los usuarios son:

- Añadir un nuevo operario a la Base de Datos.
- Obtener un listado con los operarios existentes en la Base de Datos, con

la posibilidad de borrar o editar dichos registros.

Las opciones citadas anteriormente están limitadas, de manera que solamente los usuarios que estén calificados con la categoría de encargado dispondrán de todas ellas sin limitación, mientras que aquellas que pudieran poner en peligro información privada de los usuarios serán ocultadas a los demás operarios. Las pantallas correspondientes a tres de las opciones se ilustran en la Figura 8.

4. Modelado del sistema

El modelo del sistema es el núcleo de toda la aplicación. Por un lado se modela el sistema físico (tanto barriles, vino, tuberías y conductos, etc. como los demás recursos disponibles: personal, maquinaria, y horas disponibles), por otro lado se modela el funcionamiento del mismo (la automatización implementada en los sistemas de control), y en tercer lugar se incluyen en el modelo las características de gestión (políticas de producción, etc.).

Se trata de un modelo innovador por varias cuestiones:

- Las redes de Petri constituyen un gran abanico de formalismos integrados bajo dicha denominación común, y de ellas cada vez son empleadas en mayor medida para aplicaciones industriales los formalismos más abstractos, que poseen mayor capacidad descriptiva (aunque normalmente menos capacidad de análisis), en concreto modelos coloreados o modelos híbridos (continuos y discretos). En este caso se ha empleado la combinación de ambos, es decir, modelos continuos e híbridos.

- Se ha desarrollado un modelo que integra la simulación del proceso con la gestión del mismo, de tal manera que la simulación y la optimización de los procesos generan las órdenes de trabajo, e igualmente pero en sentido opuesto, las órdenes de trabajo generadas por parte de las aplicaciones de gestión de la empresa constituyen una entrada para el modelo que proporciona una simulación de cómo será previsiblemente la sesión de trabajo.

Concretando los aspectos anteriores, aunque sin llegar al detalle por evidente falta de espacio, al ser un modelo híbrido se han empleado dos tipos de elementos, continuos y discretos.

Como variables continuas se han modelado los diferentes caldos del sistema, es decir, los mostos, los vinos en sus diferentes modalidades, aditivos líquidos, etc. y también pasta y material de desecho que no es propiamente líquido, como son los orujos (residuos de triturar la uva y extraer el jugo) o las ramas de los racimos que se obtienen al despalillar. El hecho de colorear el modelo, para condensar su estructura (no se aumenta capacidad descriptiva pero si capacidad de condensación) no se debe sólo a los diferentes tipos de vinos (claro/tinto o de-año/crianza/reserva), como podría pensarse inicialmente, sino también a vinos del mismo tiempo que hayan sufrido un proceso diferente (distinto tipo de barricas, más tiempo, etc.).

Como variables discretas se ha empleado tanto las correspondientes a barricas y depósitos, como a los recursos que se emplean para el trasiego entre las mismas, como mano de obra, maquinaria (bombas, filtros, etc.) y por supuesto las botellas de salida, cada una de las cuales está totalmente identificada.

En cuanto a la simulación del funcionamiento del modelo para su optimización, se realiza en un doble sentido. En primer lugar se puede realizar un simulado paramétrico con distintos rangos de utilización de los recursos disponibles, dependiendo del precio en un instante dado y la disponibilidad máxima de los mismos, y con él se obtiene el óptimo productivo. Una vez conseguida la política que se va a emplear en busca de la producción óptima, se introduce en la aplicación de gestión de la empresa, que

genera las órdenes de trabajo. Y finalmente esas órdenes de trabajo sirven al simulador para realizar una simulación de la producción de cada día, que permite tanto detectar posibles incongruencias (debidas a elementos mal identificados) como desviaciones de la producción esperada, lo que permite estudiar las causas para mejorar el proceso constantemente.

5. Integración de la automatización con los modelos de control

En el apartado anterior se han explicado las características más peculiares e innovadoras del modelo del sistema. Pero también se han empleado otros aspectos más comúnmente empleados con modelos de redes de Petri, como por ejemplo la integración del modelo de simulación con el de automatización y supervisión del sistema.

La utilización de sistemas de automatización basados en modelos de redes de Petri ha permitido incorporar dichos modelos en los modelos discretos para la simulación de los elementos. De esta manera se ha conseguido un diseño integral, abarcando automatización y control, gestión y toma de decisiones, visualización y monitorización.

A modo de ejemplo se puede representar en la sala de barricas empleada para mostrar la aplicación, una de las máquinas existentes que está totalmente automatizada. Para la realización de la automatización, se dividió el proceso global en bloques concurrentes. Seguidamente se muestra la planta de la máquina en la Figura 9 con una breve descripción de la tarea que se realiza en cada uno de los bloques:



Figura 8. Tres pantallas correspondientes a las opciones “Registro de Nueva Barrica”, Estadística y Listado

- Bloque 1: traslado de las barricas a la cinta trasportadora.
- Bloque 2: posicionamiento de la barrica aproximadamente en la vertical.
- Bloque 3: extracción del tapón de la barrica.
- Bloque 4: posicionamiento exacto de la barrica en la vertical.
- Bloque 5: extracción del vino contenido en la barrica.
- Bloque 6: giro de la barrica 180° para vertido de residuos.
- Bloque 7: aplicación de vapor de agua en el interior de la barrica.
- Bloque 8: aplicación de agua caliente en el interior de la barrica.
- Bloque 9: traslado de la barrica de la Cinta-1 a la Cinta-2.
- Bloque 10: posicionamiento exacto de la barrica en la vertical.
- Bloque 11: proceso de llenado de vino de la barrica.
- Bloque 12: colocación del tapón de la barrica.
- Bloque 13: posicionamiento de la barrica aproximadamente 45° respecto a la vertical.
- Bloque 14: traslado de las barricas a los soportes.
- Bloque 15: traslado de soportes desde la Zona de Carga hasta la Zona de Descarga.

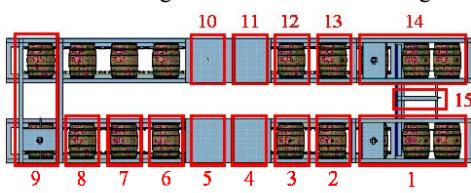


Figura 9. Planta de la máquina de tratado de barricas, y numeración de los bloques.

Para cada uno de los Bloques descritos anteriormente, se desarrollaron los modelos de estado que los representan, para llevar a cabo la automatización (y facilitar las tareas de reparación en caso de avería), y a su vez se emplearon para el modelo de toma de decisiones. De esta manera en cuanto se cambia la red de Petri de la automatización, ese cambio puede incluirse al instante en el modelo de simulación y toma de decisiones, el cual se adapta automáticamente y todo sigue funcionando correctamente.

6. Conclusiones

La aplicación para optimización de la producción mediante toma de decisiones basada en simulación del modelo y representación tridimensional que se ha presentado, ha permitido no sólo mejorar el sistema productivo sino además una mejor interrelación entre los diversos departamentos involucrados (ingeniería, producción, gestión, almacenaje,...), cuyos miembros emplean la herramienta independientemente de su conocimiento o desconocimiento de los modelos.

Referencias

- [1] Asdas Fisher, S. Virtual Interface Environments. In Multimedia: From Wagner to Virtual Reality, edited by Randall Packer and Ken Jordan. New Cork: Norton, 2001.

- [2] Sherman, W.R. Understanding virtual reality : interface, application, and design. San Francisco, CA : Morgan Kaufmann, 2003.
- [3] Bobadilla, J., Alcoce, A., Alonso, S. y Gutiérrez, A. HTML Dinámico, ASP y JavaScript a través de ejemplos. Ra-Ma®, 1999.
- [4] De la Cruz, D. y Zumbado, C. Flash, PHP y MySQL. Contenidos dinámicos. Anaya Multimedia, 2006.
- [5] Meneses, J., Álvarez, C., Rodríguez, S. Introducción al Solid Edge. Ed. Thomson, 2006.
- [6] Converse, T., Plark, J y Morgan, C. PHP 5 and MySQL® Bible. Wiley, 2004.
- [7] Gilmore, W. J. Beginning PHP 5 and MySQL: From Novice to Professional. Apress®, 2007.
- [8] García, E. Automatización de procesos industriales. Coedición: Alfaomega-Universidad Politécnica de Valencia. 2005.

Un entorno de simulación para el diseño de herramientas de toma de decisiones en logística de transporte

D. Guimaraans, J. J. Ramos, M. À. Piera

Dept. de Telecomunicaciones e
Ingeniería de Sistemas - LogiSim
Universitat Autònoma de Barcelona
08193 Bellaterra, Barcelona
Daniel.Guimaraans, JuanJose.Ramos,
MiquelAngel.Piera@uab.cat

A. Guasch

Dept. de Ingeniería de Sistemas y
Control Automático - LogiSim
Universitat Politècnica de Catalunya
08028 Barcelona
Toni.Guasch@upc.cat

Resumen

Durante los últimos años, la industria del transporte se ha visto obligada a adaptarse a las nuevas necesidades de los sistemas logísticos y de producción. El suministro *Just-In-Time* y una respuesta rápida y eficiente a las demandas de los clientes son algunas de las razones por las que la industria del transporte ha reconocido la necesidad de incorporar algunas de las nuevas tecnologías de la información y la comunicación, como los sistemas de posicionamiento global (GPS), el intercambio electrónico de información (EDI) o Internet. Estas tecnologías pueden mejorar la capacidad de optimizar las operaciones de transporte, ya que proporcionan la información necesaria requerida para la toma de decisiones en tiempo real. Este artículo presenta la plataforma que se está desarrollando con el objetivo de ayudar en el diseño de herramientas de toma de decisiones en tiempo real para la industria del transporte.

1. Introducción

En un contexto cada vez más competitivo y globalizado de la producción industrial, basado en una economía orientada a los servicios, la posibilidad de disponer de sistemas de transporte capaces de responder eficientemente a la demanda será un factor primordial para la

competitividad de la industria y de las compañías de servicios. La logística del transporte es un ejemplo típico de entorno industrial altamente complejo donde la gestión del tiempo es crítica: se debe conseguir una asignación óptima de los recursos disponibles mientras se trabaja con peticiones, condiciones y restricciones que cambian dinámicamente y de forma continua. Usualmente, este tipo de problemas logísticos se conoce como *Problema de enruteamiento de vehículos (Vehicle Routing Problem - VRP)*, donde el objetivo es la distribución eficiente de una flota de vehículos cubriendo todas las operaciones de transporte demandadas, a la vez que se reducen los costes totales, expresados como una combinación de distancia, tiempo y coste económico.

El transporte de mercancías en las áreas metropolitanas merece especial atención, debido al gran número de deficiencias que presenta hoy en día y su correspondiente contribución a la creciente congestión del tráfico en zonas urbanas. Dicha congestión se ha convertido en un preocupación creciente en los últimos años [14]: no sólo implica una mayor cantidad de tiempo invertido durante los desplazamientos en hora punta, sino que también contribuye a la degradación de la calidad del aire. La optimización del transporte en áreas metropolitanas permite reducir los correspondientes costes asociados [11], contribuyendo a su vez a la reducción de la congestión del tráfico y a un

incremento de la fiabilidad del servicio.

Así, la elevada complejidad del VRP supone un importante objetivo para los métodos de optimización [1]. Las herramientas de ayuda a la toma de decisiones en tiempo real deben combinar la información proporcionada por los sistemas GPS, las tecnologías GIS (*Geographic Information System*), el intercambio de información on-line y algoritmos eficientes de optimización para poder afrontar las necesidades particulares de cada problema. Estas nuevas tecnologías abren la puerta al diseño de sistemas de planificación capaces de manejar correctamente toda la información disponible. Este artículo presenta una plataforma tecnológica que integra dichas tecnologías como soporte para el diseño de herramientas de toma de decisiones en tiempo real en logística de transporte. Esta plataforma tiene una arquitectura de computación distribuida orientada a poder incorporar las características propias de los diferentes problemas incluidos en la categoría VRP. Las herramientas de decisión deben reaccionar en tiempo real, especialmente en áreas metropolitanas, donde la duración de las rutas es relativamente corta y cada nueva petición de servicio debe ser atendida en un período de tiempo muy restrictivo. Asimismo, el artículo presenta una aproximación en dos fases para afrontar posibles aplicaciones reales: una fase de preprocessamiento donde la información geográfica es tratada y reordenada según las necesidades del VRP a resolver; una fase de optimización donde se asignan los recursos disponibles y se definen las correspondientes rutas. Probar y mejorar un sistema complejo de estas características en un entorno real sería inefficiente y extremadamente caro. La plataforma presentada integra un sistema de simulación de flotas capaz de representar los problemas de transporte a diferentes niveles de complejidad. La siguiente sección introduce la arquitectura de la plataforma, donde el entorno de simulación proporciona la información de la flota de transporte. Las secciones siguientes describen el modelo del sistema y las herramientas que procesan la información geográfica requerida para la toma de decisiones, describiendo las características básicas de

los algoritmos de optimización propuestos para abordar los problemas VRP. Finalmente, se apuntan algunas conclusiones y líneas de trabajo futuras.

2. Arquitectura del entorno

El proyecto tiene una aproximación multidisciplinar, ya que la arquitectura integra diferentes tecnologías en cuanto a software, lenguajes de programación y sistemas operativos. La figura 1 ilustra la arquitectura del sistema. Este incluye cinco módulos principales: el *Sistema de Información Geográfica (GIS)*, el simulador de flotas, las herramientas de toma de decisiones, las aplicaciones de visualización y el sistema de localización de vehículos. El entorno ha sido diseñado como una plataforma distribuida basada en CORBA [10] para simplificar la coordinación y la comunicación entre los módulos de la aplicación. La completa modularidad de esta arquitectura también permite reemplazar módulos para responder a las necesidades de simulación o del caso de estudio de gestión de flotas: por ejemplo, el transporte de mercancías en áreas metropolitanas [2] o la gestión de los servicios de emergencia en accidentes de tráfico [5].

El Sistema de Información Geográfica gestiona toda la información estática y dinámica presente en la aplicación. La información estática está compuesta por los datos geográficos, proporcionados por una base de datos cartográfica, y la información geográfica filtrada, extraída de la anterior para generar una representación de la red de carreteras en función de las necesidades del módulo de toma de decisiones. La información dinámica está compuesta por los datos relativos a la estructura de la flota (características de los vehículos, base a la que pertenecen, estado de servicio, etc.), la posición geográfica de cada vehículo y la localización de las operaciones de transporte, así como sus características, que dependen del VRP que está siendo tratado. Por ejemplo, las restricciones de capacidad, dimensiones y de las ventanas temporales de servicio en el caso del transporte de mercancías o el número de vehículos implicados o de víctimas en la gestión

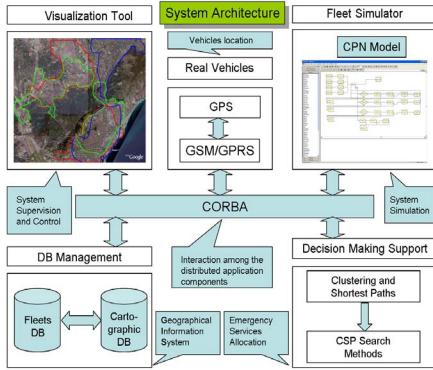


Figura 1: Arquitectura del entorno de simulación distribuido

de los servicios de emergencia.

El módulo de visualización está constituido por una aplicación gráfica donde se representan sobre un mapa los movimientos de los vehículos y las rutas configuradas. Las posiciones de los vehículos son generadas y registradas en la base de datos por el simulador de flotas durante las fases de diseño y desarrollo, mientras que su localización será proporcionada por los sistemas de posicionamiento global incorporados en los vehículos durante las operaciones en situaciones reales.

El objetivo del módulo de soporte a la toma de decisiones es la determinación de una solución factible para el problema tratado, que en todos los casos corresponde a una instancia particular del VRP. Incluidas en las técnicas de resolución se encuentran los procedimientos heurísticos y metaheurísticos, ya que ningún algoritmo exacto puede garantizar que se encuentre una solución óptima en un tiempo de cálculo razonable, debido al crecimiento exponencial de estados que presentan este tipo de problemas.

3. Modelo de la flota de transporte

El modelo de la flota de transporte se utiliza en la fase de desarrollo para probar la eficiencia de las rutinas de optimización diseñadas. Dicho modelo de simulación tiene dos funcio-

nes principales: la generación de las cargas de trabajo en forma de peticiones de transporte para el sistema y la simulación de las rutas asignadas por el módulo de optimización.

El modelo del sistema de transporte debe ser fácilmente adaptable a las diferentes variantes del problema de enrute de vehículos, siendo claros ejemplos el transporte de mercancías y la gestión de los servicios de emergencia. Así, el desarrollo del modelo no ha sido guiado por una configuración particular de un problema de estas características. En su lugar, ha sido diseñado de manera que la estructura del modelo permita una parametrización adaptable en función del problema VRP que se desee tratar y sus características: recursos para el transporte, capacidad de los vehículos, número de bases existentes, particularidades de la red de transporte, etc.. De esta manera, se hace necesaria la utilización de una metodología de modelado que permita representar conjuntamente y de forma clara la estructura del sistema y su comportamiento dinámico. El formalismo de las Redes de Petri Coloreadas (*Coloured Petri Nets - CPN*) [6] ha demostrado ser una técnica de modelado muy útil para representar las características de cualquier tipo de sistema orientado a eventos discretos, ya que presenta algunas ventajas como la capacidad de concentrar y representar simultáneamente la estructura estática y el comportamiento dinámico del sistema [13], así como la disponibilidad de técnicas de análisis matemático para la validación de modelos [15]. Por otro lado, su naturaleza gráfica proporciona una representación intuitiva y fácilmente interpretable de la estructura del sistema y su comportamiento dinámico.

El modelo CPN del sistema se muestra en la figura 2. Las peticiones de transporte se generan en el instante inicial siguiendo un modelo estocástico, aunque pueden generarse nuevas peticiones durante la simulación. En la dinámica del sistema representado por este modelo se pueden distinguir claramente dos partes: la generación de las peticiones de transporte y la simulación de los movimientos de los vehículos de la flota en la red de carreteras.

El procesado de una petición de transporte

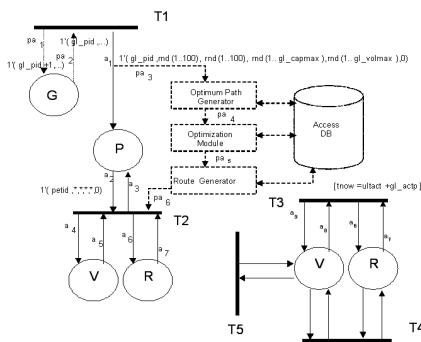


Figura 2: Modelo CPN del sistema de transporte donde se muestra su interacción con el módulo de información geográfica.

consiste, básicamente, en decidir que vehículo es asignado y, en consecuencia, en que ruta debe ser incluida la petición. Estas decisiones son tomadas por el módulo de la aplicación descrito en la siguiente sección. Los movimientos de los vehículos de la flota corresponden a la transición T3. Cuando esta transición se dispara, el estado de la flota cambia y las peticiones de transporte son atendidas cuando los vehículos llegan a los puntos de carga/descarga (transición T4). Asimismo, el modelo permite la simulación de diferentes tipos de incidencias (averías de los vehículos, pérdidas de la señal de localización, etc.), representadas mediante el disparo de la transición T5. Toda la información generada por el simulador se almacena en el módulo GIS, ya que todos los datos del sistema son compartidos por los módulos de la aplicación mediante el acceso a las bases de datos correspondientes. Para la implementación del modelo de simulación se ha utilizado la aplicación comercial Arena [7].

4. Módulo de toma de decisiones

El objetivo del VRP es servir a un conjunto de clientes con demandas conocidas utilizando aquellas rutas que provocan un coste mínimo, frecuentemente iniciadas y finalizadas en la misma base. Las instancias del VRP están

englobadas en la categoría de los problemas NP completos, con un crecimiento exponencial de estados a explorar en función del número de variables. Para su resolución, se sugiere una aproximación en dos fases en función de los criterios de optimización. En una primera fase, la información GIS se procesa con el objetivo de reducir significativamente la carga computacional en el cálculo de los caminos mínimos, necesario para la determinación de las rutas óptimas. Los datos proporcionados por el sistema GIS se clasifican en un conjunto de *clusters* dependientes del tipo de problema VRP que se desea resolver (una sola base o multi-depot, flotas homogéneas o heterogéneas, rutas con una única visita o múltiples visitas, etc.), mientras se mantiene de forma exacta toda la información referente a tiempos de viaje y costes entre los puntos de carga y descarga. En la segunda fase se resuelve el problema VRP con la información geográfica obtenida durante la primera fase.

4.1. Preproceso de la información GIS

La información geográfica contenida en las bases de datos cartográficas es necesaria para calcular costes y tiempos de ruta realistas. Sin embargo, la ingente cantidad de información aportada por este tipo de bases de datos convertiría en intratable cualquier instancia de un problema VRP en una situación real debido a la elevada cantidad de tiempo de cómputo requerida. Por ejemplo, el cálculo de una ruta mínima entre dos puntos situados en el área metropolitana de Barcelona (la ciudad de Barcelona y alrededores cuenta con $3 \cdot 10^5$ nodos geográficos y sus correspondientes arcos de unión) necesita un segundo en un ordenador personal estándar usando una aplicación comercial específica. Una carga de trabajo equivalente a 100 peticiones de servicio de recogida y entrega requiere el cálculo de 10.000 rutas mínimas entre los puntos de origen y destino.

La aplicación de generación de caminos mínimos, incluida en el módulo de soporte a la toma de decisiones, se ha desarrollado con la finalidad de preparar la información geográfica en función del problema tratado. Esta aplicación proporciona una simplificación de la

información contenida en las bases de datos GIS, obteniendo una representación de los caminos de mínimo coste entre los nodos geográficos relevantes, determinados por las peticiones de transporte, las posiciones instantáneas de los vehículos y puntos relevantes de la red de carreteras.

La aplicación de generación de caminos mínimos genera dos tipos de representación de la información GIS: un grafo de caminos mínimos, utilizado por el módulo de optimización basado en satisfacción de restricciones, y la matriz de distancias mínimas, usada en el módulo de optimización basado en modelos CPN y la exploración del correspondiente árbol de cobertura. La representación escogida depende de la instancia del problema VRP que debe abordarse.

La gestión de los servicios de emergencia [5] corresponde un problema VRP considerado como multi-depot con una flota heterogénea y ventanas temporales de servicio muy restrictivas. Con el propósito de reducir el tiempo de cómputo en la fase de optimización, la información GIS es procesada clasificando los nodos en función de la distancia y el tiempo de desplazamiento en regiones localizadas alrededor de las diferentes bases, como se muestra en la figura 3. Así, un nodo sólo es accesible por los vehículos propios de aquellas bases que minimizan el tiempo y la distancia empleados para desplazarse entre los dos puntos, creando una estructura de *clusters*. El objetivo de esta reordenación de los datos geográficos es reducir un problema VRP multi-depot a un conjunto de problemas VRP con una única base, de manera que cuando se produce un accidente sólo sea necesario resolver una de estas instancias para cada servicio de emergencia. De esta forma, los dominios de las variables de decisión quedan reducidos, mejorando el tiempo de respuesta de los algoritmos de búsqueda empleados para hallar la solución óptima. Además, junto con la reordenación de la información geográfica se lleva a cabo una tarea de cálculo de caminos mínimos dentro de cada *cluster*. Así, cuando se recibe la notificación de un accidente, se conoce una aproximación del camino mínimo entre la base y el nodo donde

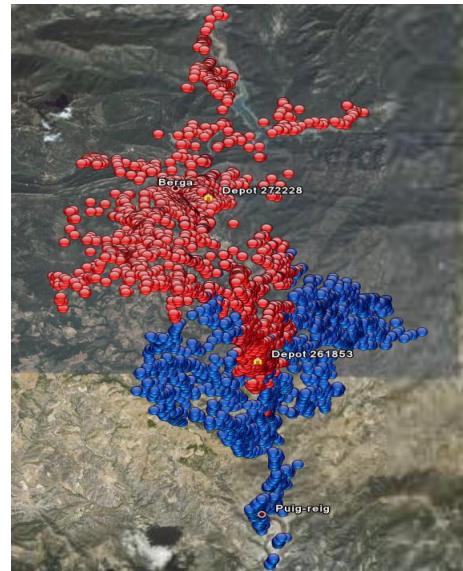


Figura 3: Ejemplo de dos clusters en una zona rural. Se puede apreciar la distribución de los nodos seleccionados alrededor de las respectivas bases.

se ha producido el siniestro, consiguiendo una importante reducción del tiempo de cómputo.

Para el proceso de los datos geográficos y su clasificación en dicha estructura de *clusters*, se emplea un procedimiento de búsqueda común A* basado en el empleo de funciones heurísticas que mejoran su eficiencia. En este caso, se emplea una función heurística de coste uniforme, con valor nulo, ya que un accidente se puede producir en cualquier nodo presente en la red y, por tanto, no puede ponderarse ninguna dirección de búsqueda concreta durante la clasificación de los nodos. Cada camino originado en la base es explorado hasta que las restricciones temporales del servicio no se cumplen, quedando así delimitada la frontera del *cluster*. Una característica de la función heurística de coste uniforme radica en el hecho que no sobreestima nunca el coste real del camino entre dos nodos. Igualmente, se trata de una función decreciente y, por tanto, los costes acumulados a lo largo de la exploración siempre aumentarán a medida que avance la búsqueda. Estas dos propiedades garantizan que todos los ca-

minos incluidos en la estructura de un *cluster* son óptimos y que ningún camino óptimo será descartado durante el proceso [12]. Asimismo, el algoritmo siempre explora aquellos nodos con un valor menor del coste acumulado en cada iteración, hecho que permite garantizar que el camino entre la base y cualquier nodo del correspondiente *cluster* es mínimo. Debido a que cualquier camino subóptimo a un nodo concreto tiene un coste mayor que la ruta óptima, se puede demostrar que cualquier camino subóptimo no se explorará nunca y que la solución devuelta será óptima.

Para el problema de transporte de mercancías en áreas metropolitanas es difícil encontrar una clasificación adecuada de la información GIS tan eficiente como en el problema anterior. En este caso, la localización de las peticiones de recogida y entrega no se conocen previamente, así que la información geográfica disponible debe aportar los rudimentos de cálculo del camino mínimo entre dos visitas cualesquiera. Como solución, se plantea una estructura de capas formada por grafos definidos a partir del hecho que la mayoría de rutas entre dos puntos se realiza por vías de comunicación principales. El número de capas queda determinado en función de la configuración de la red de carreteras del área tratada. En primera instancia, se detectan los núcleos principales de población y se determinan las principales vías de unión entre ellos. Seguidamente, los nodos y arcos correspondientes a dichos núcleos se clasifican recursivamente en estructuras de menor tamaño (distritos, barrios, etc.) siguiendo las conexiones principales entre ellas. De esta manera, el número de posibles caminos mínimos que deben ser calculados ante la llegada de una petición se ve reducida de forma significativa. Por ejemplo, únicamente se deberán calcular 200 caminos mínimos en un caso con 100 peticiones de recogida en una ciudad y entrega en otra ciudad, en lugar de las 10.000 rutas que deberían calcularse utilizando la información GIS que no ha sido procesada. Este hecho se debe a que sólo se tienen en cuenta las 100 rutas entre el punto de entrada o salida de las ciudades y los nodos donde se localizan las diferentes visitas. Igualmente, el tiempo em-

pleado en el cómputo de los caminos mínimos se reduce sustancialmente, ya que el número de nodos y arcos presentes en el grafo es menor. Para obtener la representación de los datos GIS en forma de estructura de capas se utiliza una aproximación basada en una metaheurística de colonia de hormigas (*Ant Colony - ACO*) [4]. Una aplicación interesante de esta estructuración de la información corresponde a las estrategias *cluster-first/route-second* para solucionar los problemas VRP en la fase de optimización [3].

4.2. Asignación de recursos

La formulación del problema de optimización difiere dependiendo del tipo de VRP que se aborde. En el transporte de mercancías, la función objetivo incluirá el número de camiones que se necesitan para satisfacer todas las peticiones existentes, mientras que las restricciones reflejarán los horarios de trabajo, de descanso y las ventanas temporales de entrega, entre otros. Se han adoptado dos aproximaciones diferentes para resolver el problema de optimización discreta presentado, donde el objetivo es reducir los costes de transporte. La primera estrategia está basada en métodos de programación de restricciones [8], en la que se combinan dos elementos: el modelo, donde se definen las variables de decisión, las restricciones del problema y la función objetivo; el algoritmo de búsqueda utilizado para su resolución, que selecciona un valor en el dominio de cada variable restringida, de manera que se cumplan todas las restricciones. La segunda aproximación consiste en una combinación de técnicas de simulación y de búsqueda. El modelo de transporte se representa mediante el formalismo CPN y se utiliza el árbol de cobertura que de él se deriva para evaluar los estados factibles que se obtienen a partir de un estado inicial dado. Sin embargo, el árbol de cobertura de un sistema logístico puede crecer exponencialmente en función del número de eventos que se pueden disparar en paralelo. Por esta razón, se aplican técnicas basadas en heurísticas para reducir el espacio de búsqueda y conducir al sistema hacia el estado final deseado [9].

En la gestión de servicios de emergencia, el objetivo consiste en asignar de forma adecuada los recursos disponibles a los accidentes existentes, de acuerdo con las restricciones implícitas del problema, englobado en la categoría de problemas de satisfacción de restricciones (*Constraints Satisfaction Problem - CSP*). Estas restricciones determinan que el tiempo máximo de respuesta no debe ser superado por la solución dada e impiden que queden regiones desprovistas de recursos. En consecuencia, no se pueden asignar todos los recursos disponibles en una base a un único accidente, ya que no se cumpliría la segunda condición. Para resolver un problema de estas características se utilizan técnicas CSP, donde se utiliza la información obtenida durante el procesado de los datos geográficos para reducir el dominio de las variables de decisión. Por otro lado, un accidente puede ser directamente asignado a una base determinada con un coste computacional mínimo, ya que el nodo donde se ha producido pertenece a un *cluster* concreto o, en todo caso, se puede calcular el camino mínimo entre el accidente y la frontera del *cluster* más cercano. Así, la búsqueda de la solución se reduce a explorar los *clusters* más cercanos al punto donde se ha producido el accidente, reduciendo el espacio de búsqueda y el coste computacional. Sin embargo, eventualmente no es posible hallar una solución que satisfaga las condiciones temporales impuestas (accidentes ocurridos en zonas rurales, tráfico colapsado, etc.). En este caso, las restricciones temporales pueden ser relajadas o ponderadas, mientras se mantiene como objetivo la minimización del tiempo de respuesta. De esta forma, el objetivo consiste en asignar los recursos de manera que se satisfagan todas las condiciones impuestas excepto la ventana temporal de respuesta, que queda asimilada en la minimización de la función objetivo.

En ambos casos de estudio, la información disponible durante las operaciones de los sistemas de transporte cambia dinámicamente. Asimismo, la información disponible llega de forma continua desde diferentes fuentes, conteniendo un importante factor de incertidumbre. Por otro lado, la situación del tráfico puede

cambiar durante el transcurso de las operaciones de transporte, desde el momento en que la ruta es asignada hasta que finaliza el servicio, de manera que las rutas calculadas pueden perder su condición de optimalidad. Este tipo de situaciones podrían ser detectadas utilizando la información recibida desde los dispositivos GPS de los vehículos, ya que permiten determinar su posición y los movimientos realizados. Así, las correspondientes rutas deben ser nuevamente calculadas y, en algunas ocasiones, los vehículos tendrán que ser reasignados. Todas estas situaciones pueden ser representadas mediante el módulo de simulación, de forma que las herramientas de soporte a la toma de decisiones pueden ser diseñadas de forma apropiada y validadas sin los costes asociados a la experimentación en un sistema de transporte real.

5. Conclusiones

Este artículo presenta una plataforma distribuida basada en CORBA desarrollada para el diseño, implementación y validación de herramientas de soporte a la toma de decisiones en problemas de logística y transporte. Debido a su modularidad y su estructura distribuida, esta plataforma puede ser fácilmente adaptada para afrontar diferentes instancias de problemas VRP, mediante la introducción o sustitución de los diferentes módulos que la componen. En el proceso de validación se utiliza un simulador de flotas para determinar la eficiencia de las herramientas diseñadas. Dicho simulador está basado en un modelo que puede ser parametrizado en función del problema VRP que se desea abordar.

Las herramientas de soporte a la toma de decisiones están basadas en una aproximación que consta de dos fases. En la primera de ellas, los datos geográficos son procesados, mediante su análisis y clasificación en una estructura acorde con las necesidades del problema VRP objeto de estudio. A su vez, con este proceso se reduce el tiempo que será necesario para calcular los caminos mínimos entre las localizaciones correspondientes a las diferentes peticiones de transporte. Igualmente, esta reorde-

nación de la información geográfica puede ser utilizada para reducir el dominio de las variables de decisión, delimitando así el espacio de búsqueda en la resolución del problema. En la segunda fase, los recursos son asignados y se calculan las correspondientes rutas, considerando las estructuras de caminos mínimos obtenidas durante el preproceso. Actualmente, se está trabajando en dos aproximaciones diferentes para modelar y resolver las diferentes instancias del problema VRP: programación con restricciones y la combinación de técnicas de simulación y búsqueda.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo del *Departament d'Universitats, Recerca i Societat de la Informació* de la Generalitat de Catalunya, del Ministerio de Fomento y del Ministerio de Ciencia y Tecnología (proyectos DPI2004-08056-C03-01 y DPI2004-08056-C03-02).

Referencias

- [1] Bektas, T., *The multiple traveling salesman problem: an overview of formulations and solutions procedures*, The International Journal of Management Science, 34, 2006.
- [2] Busquets, S., Vilalta, L., Piera, M.A., Guasch, A. and Narciso, M., *Specification of metaheuristics in Colored Petri Nets models to tackle the Vehicle Routing Problem*, International Mediterranean Modeling Multiconference, Marsella (Francia), 2006.
- [3] Dondo, R. and Cerdá, J., *A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows*, European Journal of Operational Research, 176, 2007.
- [4] Dorigo, M. and Stützle, T., *The Ant Colony optimization metaheuristics: algorithms, application and advances*, Ed. Glover, F. and Kochenberger, G.A., *Handbook of Metaheuristics*, Kluwer Academic Publishers, Boston, 2003.
- [5] Guimaraens, D., Ramos, J.J., Piera, M.A. and Guasch, A., *A simulation based decision tool to coordinate emergency services in a road accident*, Summer Computer Simulation Conference, Calgary (Canadá), 2006.
- [6] Jensen, K., *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, vol. 1,2,3, Springer-Verlag, Berlín, 1997.
- [7] Kelton, W.S., Sadowski, R.P. and Sadowski, D.A., *Simulation with Arena*, McGraw-Hill, 1998.
- [8] Marriott, K. and Stuckey, P., *Programming with constraints: an introduction*, MIT Press, 1998.
- [9] Narciso, M.E., Piera, M.A. and Guasch, A., *A knowledge representation approach suitable to support heuristic search methods frontiers in Artificial Intelligence and applications*, Artificial Intelligence Research and Development, 100, 2003.
- [10] OMG, Inc., *Common Object Request Broker Architecture: Core Specification*, 2004.
- [11] Papows, J., *Route optimization as energy policy*, Transport Topics, June 2001.
- [12] Russell, S. and Norvig, P., *Artificial Intelligence: a modern approach*, Prentice-Hall, 2003.
- [13] Silva, M. and Valette, R., *Petri Nets and flexible manufacturing*, Lecture Notes in Computer Science, vol. 424, Advances in Petri Nets, 1989.
- [14] Wachs, M., *Fighting traffic congestion with information technology*, Issues in Science and Technology, 2002.
- [15] Zimmerman, A., Dalkowski, K. and Hommel, G., *A case study in modeling and performance evaluation of manufacturing systems using Colored Petri Nets*, European Simulation Symposium, Génova (Italia), 1996.

HERRAMIENTAS DE MODELADO Y SIMULACIÓN

Simulación de Sistemas Dinámicos complejos mediante Autómatas Celulares de comportamiento macroscópico definido

Joaquín Cerdá
Boluda
Dept. de Ingeniería
Electrónica
Universidad Politécnica de
Valencia
46022 Valencia
joacerbo@eln.upv.es

Néstor Ferrando
Jódar
Dept. de Ingeniería
Electrónica
Universidad Politécnica de
Valencia
46022 Valencia
nesferjo@etsii.upv.es

Rafael Gadea
Gironés
Dept. de Ingeniería
Electrónica
Universidad Politécnica de
Valencia
46022 Valencia
rgadea@eln.upv.es

Ricardo José
Colom Palero
Dept. de Ingeniería
Electrónica
Universidad Politécnica de
Valencia
46022 Valencia
rcolom@eln.upv.es

Resumen

Los Autómatas Celulares (ACs) modelan fenómenos físicos y estructuras de computación masivamente paralelos. Los ACs consisten, básicamente, en una disposición regular de nodos idénticos denominados *células*. Cada célula almacena un valor perteneciente a un conjunto discreto. El valor de cada célula evoluciona en el tiempo en pasos discretos, actualizándose mediante la aplicación reiterada de una regla determinista, la cual toma como entrada los valores de las células vecinas; un proceso paralelo, síncrono, local y uniforme.

En este artículo discutiremos acerca de las propiedades para la simulación y modelización de sistemas que ofrecen los Autómatas Celulares. El caso concreto que vamos a trabajar es la posibilidad de obtener un modelo celular a partir de un fenómeno conocido a priori y del que se tiene una ecuación macroscópica que describe su comportamiento. Eso nos llevará a plantear el Método BE-PE, cuyo fin es, precisamente, el indicado. Como ejemplo de los resultados que ofrece este método nos centraremos en el caso del campo electrostático, lo que nos permitirá introducir el Autómata RNR que, en el límite macroscópico, describe el potencial en una determinada región.

1. Motivación

Toda simulación numérica surgida a partir de una ecuación matemática lleva inherente una serie de problemas relacionados con el proceso de truncamiento inevitable a la hora de realizar cálculos y de traducir una ecuación matemática abstracta a una serie de operaciones numéricas.

Cuando estamos trabajando con fenómenos lineales, los cuales presentan un comportamiento regular y una baja sensibilidad a fluctuaciones en los datos de entrada, estos redondeos y truncamientos no ofrecen demasiados problemas, pero cuando nos encontramos trabajando con ecuaciones no lineales con elevada sensibilidad a los datos de entrada, esto ya no es así.

En oposición a los métodos clásicos de simulación basados en el planteamiento de ecuaciones matemáticas y en su posterior aproximación numérica, recientemente han tenido bastante auge los métodos de simulación basados en Autómatas Celulares [1].

En un Autómata Celular no existen problemas de redondeo o de truncamiento, pues los modelos propuestos son exactamente computables, al estar basados en sistemas discretos, tanto en el tiempo como en el espacio. Las variables vienen representadas por valores de un determinado alfabeto, no números reales, lo que elimina la incertidumbre en el cálculo.

A diferencia de lo que se podría pensar en un primer momento, el hecho de utilizar Autómatas Celulares para el modelado de fenómenos físicos y sistemas dinámicos no acorta el rango de fenómenos aplicables ni los simplifica de manera que el tipo de comportamientos que pueden ser sintetizados queda reducido a un pequeño grupo de no demasiada importancia. Antes bien, Los métodos de modelización basada en Autómatas Celulares de han demostrado como posibilidades muy firmes y candidatos muy adecuados para una modelización de casi cualquier tipo de fenómenos, habiéndose utilizado en áreas en principio tan diversas como puede ser la física fundamental, la biología, la química o el modelado social, por citar sólo algunos campos [2].

Existe un principio general que tomamos como base para nuestro trabajo: el comportamiento macroscópico de un sistema compuesto de muchos constituyentes interactuantes depende en poca medida de los detalles microscópicos de las interacciones. Como ejemplo podemos citar la ecuación de Navier-Stokes, que gobierna el movimiento de un fluido y que es, fundamentalmente, una expresión de la conservación del momento lineal durante las interacciones elementales entre las moléculas del fluido. Los detalles de estas interacciones se ven reflejados en los coeficientes de la ecuación (por ejemplo, la viscosidad), pero no en su estructura algebraica [3].

Es, por lo tanto, uno de los puntos más importantes que subyacen al concepto de modelado físico mediante autómatas celulares. Existen numerosos niveles de realidad cuando se abarca el estudio de un sistema físico. Polarizando la cuestión, podemos distinguir dos niveles extremos de observación: microscópico y macroscópico. Un mismo sistema puede parecer muy distinto cuando se observa en estos niveles. A nivel microscópico, las interacciones pueden venir gobernadas por complicadas funciones de potencial y, a veces, requieren el uso de mecánica cuántica para ser descritos propiamente. Desde un nivel macroscópico, las propiedades del sistema se ven dominadas por el efecto de agregación de todas las interacciones microscópicas. El comportamiento resultante, visto a gran escala de observación, está más relacionado con las características genéricas de las interacciones. Así, la complejidad del mundo macroscópico está aparentemente desconectada de la del mundo microscópico, pese a que el primero es consecuencia directa del segundo. Los detalle microscópicos se pierden cuando el sistema se mira a través de un filtro macroscópico, y nuevas propiedades colectivas aparecen.

La aproximación basada en Autómatas Celulares es útil cuando la simplificación de las leyes microscópicas que rigen un determinado fenómeno no es relevante a una escala macroscópica de observación. De la Mecánica Estadística sabemos que este hecho es correcto a menudo para sistemas cuya complejidad deriva más del comportamiento colectivo que de algunos aspectos distintivos de las interacciones microscópicas.

A partir de la observación de que el comportamiento microscópico de muchos sistemas tiene poca importancia sobre la naturaleza macroscópica, resulta particularmente útil inventar sistemas microscópicos sencillos que lleven a los mismos resultados macroscópicos. Estos sistemas pueden ser diseñados de forma que se adapten más a los recursos específicos de nuestra investigación numérica. El punto importante es capturar los ingredientes esenciales y relevantes de un fenómeno real y considerarlos como las leyes físicas fundamentales de un nuevo sistema imaginario.

Modelar un sistema a nivel microscópico tiene ventajas importantes. La interpretación de la dinámica de un Autómata Celular en términos de reglas microscópicas sencillas ofrece una aproximación intuitiva y poderosa para modelar fenómenos que son difíciles de abordar desde aproximaciones más tradicionales.

El diseño de modelos correctos basados en Autómatas Celulares implica que los aspectos esenciales de un fenómeno complejo han sido identificados y reducidos a una forma sencilla. Este proceso de reducir un comportamiento complejo a la suma de mecanismos simples es un paso esencial en cualquier investigación científica.

Así pues disponemos de dos niveles fundamentales de descripción de los fenómenos físicos: el nivel microscópico y el macroscópico. Eso nos suscita dos posibles interrogantes que obedecen a la necesidad de relacionar entre sí ambos niveles:

En primer lugar, dado un Autómata Celular con una regla local dada, ¿cuál es su comportamiento macroscópico? ¿Qué tipo de distribuciones sigue? ¿Cuáles son sus soluciones? ¿Qué leyes de conservación se cumplen?

En segundo lugar planteamos la pregunta inversa. Si partimos de un modelo macroscópico que puede ser modelizado correctamente por una serie de ecuaciones, ¿cómo diseñar una regla de evolución local que, en el límite macroscópico, me conduzca a resultados numéricamente compatibles?

Ha habido numerosos intentos por sistematizar de algún modo la respuesta a la primera pregunta. Quizá el método más famoso para obtener las ecuaciones que rigen el comportamiento macroscópico en el límite continuo de un Autómata Celular es la Expansión de Chapman-Enskog. Este Método fue el que

utilizó Wolfram en [4] para probar que el modelo FHP de fluido bidimensional de Frisch, Hasslacher y Pomeau [5] cumplía, en el límite macroscópico, la Ecuación de Navier-Stokes. Para una descripción detallada del Método es interesante consultar [6].

Existen toda una serie de procedimientos derivados de la Ecuación de Boltzmann y de la Expansión de Chapman-Enskog para obtener el límite macroscópico de una determinada regla. Sin embargo, es más raro encontrar la respuesta a la pregunta inversa, esto es, dado un modelo macroscópico, encontrar una regla de evolución que sea capaz de dar soporte a ese tipo de fenómenos. Esa será la dirección que tomará nuestra investigación y que se presenta en el presente artículo.

2. El Método BE-PE

Tras las consideraciones introducidas en el epígrafe anterior, vamos a introducirnos de lleno en el objetivo final de nuestro estudio: plantear un Autómata Celular que, observado a escala macroscópica, se comporte de una forma definida.

Para tal fin introduciremos el Método BE-PE (*Binary evaluation – Probabilistic Evolution*), que se basa en el enfoque probabilístico que hemos discutido en el apartado anterior. En esencia, el Método consiste en seguir los siguientes pasos:

- Reemplazar tiempo y espacio continuo por un retículo discreto.
- Reemplazar variables continuas por valores binarios.
- Reemplazar Ecuaciones en Derivadas Parciales por sus equivalentes en términos de Diferencias Finitas, introduciendo un enfoque probabilístico para los valores resultantes.
- Aplicar las ecuaciones discretas a una célula y todas sus posibles configuraciones de vecindad para obtener todos los resultados posibles.
- Encontrar una regla de evolución sencilla que permita obtener esos mismos resultados.

(f) Efectuar la evolución del espacio celular completo sometido a unas determinadas condiciones de contorno para comprobar en qué medida los resultados obtenidos se adaptan al comportamiento macroscópico deseado.

3. El problema de las magnitudes continuas

Ya en [3] Toffoli postula cómo pueden emergir cantidades escalares a partir de un retículo binario si no se considera el valor en cada célula, sino el valor medio promediado en toda una región. A medida que el tamaño de la región promediada crece, también crece el conjunto de posibles valores de su valor medio.

Como ejemplo nos referiremos a la figura 1. Supongamos que estamos interesados en el valor de la célula central. En este caso concreto, su valor es ‘0’, aunque en general, esta célula puede presentar dos posibles valores: ‘0’ ó ‘1’.

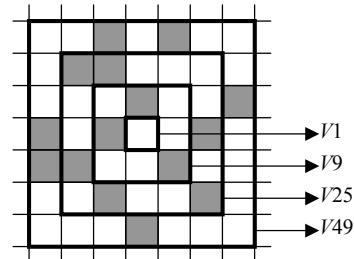


Figura 1. Región con valores binarios

Pero supongamos ahora que no estamos interesados en el valor de una célula concreta sino en el valor medio en un determinado volumen, V . De este modo podemos definir la magnitud α_V como:

$$\alpha_V = \frac{\sum_{i \in V} c(i)}{N} \quad (1)$$

siendo N el número de células que contiene V . En la expresión se ha utilizado la suma y la división

en el cuerpo de los números reales según la notación acostumbrada. Así pues, si nos referimos a la región V_9 de la figura 1, tenemos que $\alpha_{V_9}=0,333$ aunque, en general, esta magnitud puede tomar 10 valores distintos $\{0/9, 1/9, \dots, 9/9\}$. Aumentemos la región de interés a V_{25} . En nuestro caso $\alpha_{V_{25}}=9/25$ y, en general, es posible obtener 26 valores distintos, $\{0/25, 1/25, \dots, 25/25\}$. Así, si V está formado por N células, α_V puede tomar $N+1$ valores distintos en el intervalo $[0,1]$. A medida que el tamaño de V se va incrementando, el rango de α_V se aproxima más y más a un número real, con lo que puede ser considerado una magnitud continua dentro de los límites adecuados. Una descripción más detallada de estos conceptos puede consultarse en [3].

A lo largo de nuestro trabajo nos planteamos el adscribirnos al punto de vista expuesto, al representar una magnitud escalar como el promediado espacial de valores binarios. No obstante, hemos querido introducir una nueva variante considerando no sólo el promediado espacial sino también el promediado temporal de los valores de las células. Así, nuestro estudio se basa, fundamentalmente, en el cálculo de $\alpha_{V,T}$ definido por:

$$\alpha_{V,T} = \frac{1}{T} \sum_{j=0}^{T-1} \left(\frac{\sum_{i \in V} \tau^j \mathbf{c}(i)}{N} \right) \quad (2)$$

Físicamente, el hecho de introducir magnitudes físicas promediadas es la expresión matemática de una restricción operacional fundamental común a todos los campos científicos: la imposibilidad de medir una cantidad en un punto concreto del espacio. Por muy refinados que sean nuestros métodos de medida, siempre estaremos restringidos a efectuar tomas de medidas dentro de volúmenes finitos. Piénsese, por ejemplo, en todos los métodos empleados en medida de temperaturas, que consisten, básicamente, en poner en contacto térmico el sistema bajo medida con un sistema indicador llamado genéricamente termómetro. Dicho termómetro tiene una magnitud característica, llamada Magnitud Termométrica, que varía en función de la temperatura de una determinada Zona de Contacto. En función del Termómetro particular que se utilice dicha zona de contacto será mayor o menor, pero desde luego nunca llega

a ser puntual, motivo por el cual es adecuada la consideración de promediados espaciales.

Del mismo modo puede razonarse para los promediados temporales. Físicamente es imposible tomar valores instantáneos de una variable, sino que dichos valores deben tomarse a lo largo de un cierto tiempo. Volvamos sobre el caso de la medida de temperaturas. Cuando ponemos un termómetro en contacto térmico con el sistema a medir se establece un flujo de calor entre ambos sistemas que no cesa hasta que se llega a la condición de Equilibrio Térmico. Es, por tanto, necesario que la medida se lleve a cabo durante un intervalo determinado de tiempo, suficiente para mitigar el efecto de las fluctuaciones hacia el equilibrio. Lo expuesto en este caso puede hacerse extensible a cualquier magnitud física que deseemos medir.

4. Aplicación del Método BE-PE para la simulación del campo electrostático

Como aplicación específica del método expuesto anteriormente, con todas las consideraciones previas, vamos a centrarnos en el problema de la descripción de fenómenos electrostáticos.

Es de sobra conocido que el campo electrostático en una región libre de cargas puede modelizarse mediante la famosa Ecuación de Laplace:

$$\nabla^2 \Phi = 0 \quad (3)$$

Teniendo en cuenta la expresión del operador Laplaciano en cartesianas, para una región bidimensional la ecuación toma la forma:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (4)$$

Dicha ecuación tiene solución analítica sólo para unas condiciones de contorno muy específicas. Y, normalmente, dicha solución no tiene una expresión sencilla. Por ejemplo, para una región cuadrada de lado L sometida a las siguientes condiciones de contorno:

$$\begin{aligned}\Phi(0, y) &= \Phi(L_x, y) = \Phi(x, 0) = 0 & (5) \\ \Phi(x, L_y) &= \Phi_0\end{aligned}$$

La solución por medios analíticos lleva a una serie infinita:

$$\Phi(x, y) = \Phi_0 \cdot \sum_{n=1}^{\infty} \frac{4}{n\pi} \sin\left(\frac{n\pi x}{L_x}\right) \frac{\sinh(n\pi y/L_x)}{\sinh(n\pi L_y/L_x)} \quad (6)$$

Dado que la solución consiste en una suma infinita el operar con la misma es tedioso y poco práctico. De hecho, si representamos gráficamente esta solución encontraremos que son necesarios muchos términos para obtener resultados decentes, especialmente en los extremos, donde la existencia de una discontinuidad provoca la aparición de unas oscilaciones especialmente molestas. Estas oscilaciones son características en aquellos problemas cuya solución está basada en el empleo de series de Fourier.

Hay un problema más añadido al hecho de operar en base a una solución basada en una serie infinita. Teniendo en cuenta su naturaleza, todo intento de cuantificación necesariamente pasa por el truncamiento de la serie a un número finito de términos, despreciando los infinitos términos restantes. Es evidente que puede elegirse el número de términos suficientemente elevado para asegurar que el máximo error cometido es menor de una determinada precisión, pero aún así vemos que todo intento de numerizar la expresión no es sino una aproximación a la verdadera solución.

La clásica alternativa numérica al problema consiste en discretizar la función y las ecuaciones, llegando a:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0 \quad (7)$$

Lo único que restaría hacer ahora es resolver las ecuaciones, por ejemplo utilizando el Método de eliminación de Gauss, que nos daría como resultado los valores del potencial en cada nodo del retículo. En la figura 2 Puede verse el resultado de aplicar este método en un retículo de 20x20.

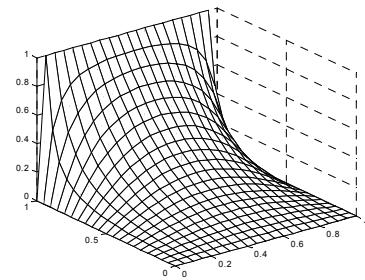


Figura 2. Potencial en un retículo

El método expuesto adolece de serios problemas en lo referente a esfuerzo computacional y necesidad de memoria. Resulta evidente que cuando la separación entre nodos se hace pequeña el número de ecuaciones crece siguiendo una ley cuadrática, lo que supone una gran cantidad de memoria para almacenar los valores. Asimismo, la resolución por Reducción de Gauss de un sistema de grandes dimensiones puede ser extremadamente lenta.

Se plantea, pues, el recurrir a métodos iterativos de modo que se pueda acelerar el proceso en la medida de lo posible.

El Método de Liebmann surge como una primera aproximación al procesado iterativo. Basta con reordenar la ecuación en diferencias para obtener:

$$u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}}{4} \quad (8)$$

Esta fórmula se aplica iterativamente sobre todos los nodos del dominio hasta que dos configuraciones sucesivas difieren menos de un determinado valor de precisión δ . Los resultados obtenidos por este método no difieren significativamente de los que se muestran en la figura 2.

En este proceso seguido, vemos como poco a poco hemos introducido mayor grado de discretización en el sistema. Vamos a llevar hasta las últimas consecuencias esta aproximación y discretizaremos los valores de la función en cada punto hasta un valor binario, lo que constituirá el último estadio de la aplicación del método BE-PE al problema electrostático.

En nuestro estudio partiremos de la ecuación (8), pero la reformularemos para obtener, ya no valores estrictos de una determinada célula, sino el valor medio que una célula debe tener si promediáramos en un intervalo finito, lo que, por definición estadística, coincide con la probabilidad de que esa misma célula esté a 1 en un instante dado. Utilizando este enfoque probabilístico podemos escribir, como ecuación fundamental:

$$p_{i,j}(1) = \frac{c(i+1,j) + c(i-1,j) + c(i,j+1) + c(i,j-1)}{4} \quad (9)$$

donde $p_{i,j}(1)$ es la probabilidad de que la célula (i,j) se encuentre en estado '1' en la siguiente evolución.

Los pasos (a) y (b) del Método BE-PE ya han sido dados implícitamente mediante la introducción de Autómatas Celulares. El paso (c) implica el uso de la ecuación (9), la cual nos da las probabilidades para los valores siguientes en la evolución del Autómata Celular en función de los estados presentes de los vecinos. Para la aplicación de esta ecuación, paso (d), utilizaremos la Vecindad de von Neumann con radio unidad. Evaluaremos el valor siguiente para cada posible configuración de vecinos. Tras ello deberemos invocar un proceso aleatorio que nos permita obtener, tras el promediado temporal, los valores medios deseados.

Estudiemos las posibilidades. Particularizaremos los casos más sencillos para luego interpretar el caso general:

CASO A: Los cuatro vecinos a '0': $p_{i,j}(1)=0$. Sin ninguna componente probabilística el estado siguiente es '0'.

CASO B: Los cuatro vecinos a '1': $p_{i,j}(1)=1$. Sin ninguna componente probabilística el estado siguiente es '1'.

CASO C: Dos vecinos a '1' y dos vecinos a '0': $p_{i,j}(1)=0,5$. Aquí es donde entra en juego nuestro enfoque probabilístico. Necesitamos una variable aleatoria que, al ser promediada temporalmente, su valor tienda a 0,5. Dado que dicha variable sólo puede dar valores binarios, lo que realmente necesitamos es un generador aleatorio que tenga la misma probabilidad de tomar el valor '1' que de tomar el valor '0'.

CASO D: Un vecino a '1' y tres vecinos a '0': $p_{i,j}(1)=0,25$. De nuevo necesitamos una variable aleatoria pero, en esta ocasión, su promedio

Herramientas de Modelado y Simulación

temporal debe tender a 0,25. Por lo tanto, necesitamos un generador aleatorio que tenga una probabilidad de tomar el valor '1' de 0,25 y, consecuentemente, una probabilidad de tomar el valor '0' de 0,75.

CASO E: Un vecino a '0' y tres vecinos a '1': $p_{i,j}(1)=0,75$. El promedio temporal de nuestra variable aleatoria debe tender a 0,75. Necesitamos un generador aleatorio que tenga una probabilidad de tomar el valor '1' de 0,75 y, consecuentemente, una probabilidad de tomar el valor '0' de 0,25.

Los resultados para todos los casos aparecen resumidos en la figura 3 donde hemos denotado las células en estado '1' como sombreadas en negro y las células en estado '0' como en blanco. A la derecha se indica los posibles estados siguientes de la célula central y la probabilidad con que deben aparecer.

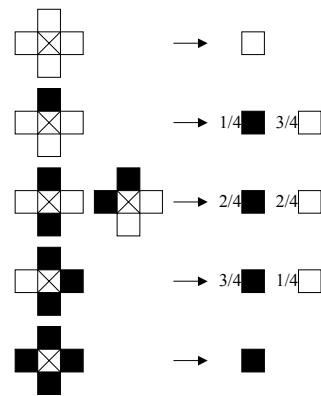


Figura 3. Resumen de casos

5. El Autómata Celular RNR

¿Podemos encontrar una regla que se adapte a los resultados anteriores? Comprobaremos la siguiente regla:

```
para todas las células
  elegir un vecino aleatorio;
  copiar su valor;
fin para
```

Algoritmo 1. Regla RNR

Si evaluamos todas las configuraciones de vecindad posibles vemos que este autómata se adapta perfectamente a nuestras necesidades, cumpliendo las probabilidades que se indican en la figura 3. Esto cumple el objetivo (e) de nuestro estudio preliminar. Este Autómata Celular, al que nos referiremos en adelante como RNR (por la denominación descriptiva en inglés *Random Neighbor Rule*), ha sido introducido por nosotros y algunas propiedades relativas al mismo han sido presentadas en [7].

Para comprobar que nuestro autómata se adapta correctamente al comportamiento predicho, lo primero que haremos serán unas simulaciones que nos permitan observar cualitativamente que el sistema evoluciona hacia un estado más o menos adaptado a la situación requerida. Dispondremos de una matriz bidimensional de células interconectadas, cada una de esas células implementando las reglas propuesta. Inicializaremos el sistema y veremos cómo evoluciona a través del tiempo.

En la figura 4 presentamos algunas generaciones de una simulación típica. Para este caso concreto utilizamos una matriz de 50x50 células sometidas a las condiciones de contorno (5), en donde se asumirá que $\Phi_0=1$. La simulación se inició a partir de una distribución aleatoria de los valores de las células en la que $p_{i,j}(1)=1/4$.

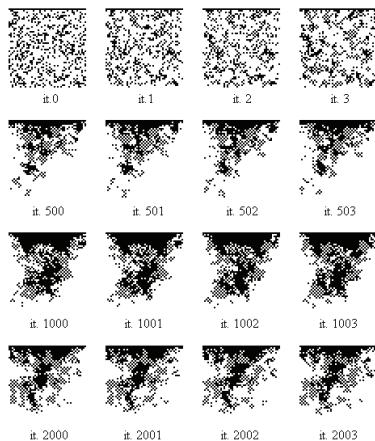


Figura 4. Simulación del Autómata RNR

Las imágenes que se dan en la figura 4 muestran que, tras algún período transitorio, el

sistema evoluciona hacia una situación estacionaria en la que las células cercanas al borde superior tienen mayor probabilidad de estar a '1' que las que se encuentran en el límite inferior. Si ahora efectuamos el promediado temporal a lo largo de un determinado intervalo, el resultado que se obtiene va tendiendo al que se muestra en la figura 2. Una secuencia obtenida mediante el promediado sucesivo puede verse representada en la figura 5.

Existen dos efectos principales que deben ser comprobados en relación con la exactitud del método propuesto. El primero de ellos es que el error en la estimación del potencial debe verse disminuido a medida que el número de iteraciones promediadas crece. Los resultados de simulaciones en un retículo de 20 por 20 células se muestran en la figura 6, donde se representan tanto el error máximo como el error medio entre la solución obtenida mediante el autómata RNR y la solución analítica. El error se ve decrementado a medida que el número de evoluciones promediadas crece. La introducción de un promediado espacial también debería incrementar la exactitud de nuestros resultados. Para comprobar esto, hemos observado la evolución de los errores a medida que la región promediada crece, lo que puede verse en la figura 7, donde se da el error en dos puntos distintos del retículo como función del tamaño de la región de observación. Cada punto de las gráficas representa el conjunto de 20 simulaciones tras un tiempo fijo de 1000 iteraciones en un retículo de 100 por 100 células. Se comprueba que el error decrece a medida que la región de observación aumenta de tamaño [7].

6. Conclusión

En este artículo hemos planteado la introducción del Método BE-PE para obtener un modelo celular a partir de un fenómeno conocido a priori y del que se tiene una ecuación macroscópica que describe su comportamiento. Como ejemplo de los resultados que ofrece este método hemos descrito el caso del campo electrostático, lo que nos ha llevado a introducir el Autómata RNR que, en el límite macroscópico, describe el potencial en una determinada región.

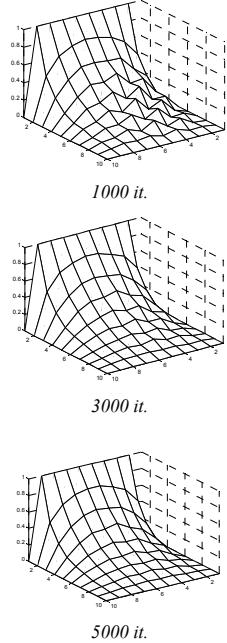


Figura 5. Promedio de iteraciones del Autómata RNR

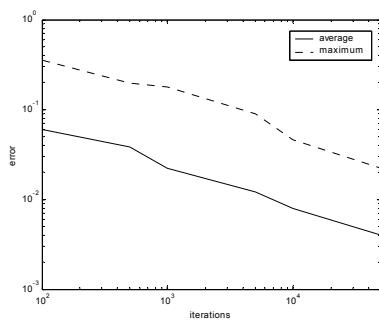


Figura 6. Dependencia del error con el número de iteraciones promediadas

El Método BE-PE es un método general y puede aplicarse a una gran cantidad de situaciones específicas con el resultado de Autómatas Celulares que nos permitan estudiar el fenómeno desde aproximaciones computacionalmente eficientes.

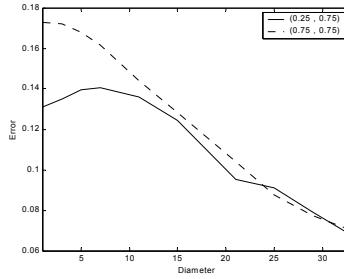


Figura 7. Dependencia del error con el diámetro de la región de observación

7. Agradecimientos

Este trabajo ha sido financiado por la Universidad Politécnica de Valencia, Vicerrectorado de Investigación, Desarrollo e Innovación, proyecto PPI-06-05.

Referencias

- [1] S. Wolfram. *A New Kind of Science*. Wolfram Media, Inc. 2001
- [2] J. Cerdá. *Arquitecturas VLSI de Autómatas Celulares para Modelado Físico*. Ed. Universidad Politécnica de Valencia, 2004.
- [3] T. Toffoli. Cellular Automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D* 10:117-127, 1984.
- [4] S. Wolfram. Cellular automaton fluid: basic theory. *J. Stat. Phys.*, 45:471, 1986.
- [5] U. Frisch, B. Hasslacher, Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Phys. Rev. Lett.* 56:1505, 1986.
- [6] [Chop98] B. Chopard, M. Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press. 1998.
- [7] J. Cerdá, O. Amoraga, R. Torres, R. Gadea, A. Sebastià. FPGA Implementations of the RNR Cellular Automata to Model Electrostatic Field. *High Performance Computing for Computational Science. Lecture Notes in Computer Sciences Vol*, pp 382-395, 2005

Modelado y simulación de un Centro de Atención al Usuario con SIGHOS

Iván Castilla, Roberto Muñoz,
Rosa M. Aguilar

Dept. de Ingeniería de Sistemas y Automática,
y Arquitectura y Tecnología de Computadores
Univ. de La Laguna
38217 Tenerife
{ivan, rmglez, rosi}@isaatc.ull.es

Pedro J. Baquero

Dpto. de Física Fundamental y
Experimental, Electrónica y
Sistemas
Univ. de La Laguna
38217 Tenerife
pbaquero@ull.es

Yurena García-Hevia

Escuela Técnica Superior de
Ingeniería Informática
Univ. de La Laguna
38217 Tenerife

Resumen

SIGHOS es una librería Java para la simulación de sistemas de eventos discretos. Aunque en principio es válida para cualquier sistema de este tipo, su orientación al proceso la hace más adecuada para modelar organizaciones, especialmente aquéllas en las que la dinámica viene expresada por un flujo de trabajo. En este artículo se describe el uso de la librería para modelar y simular un centro de atención al usuario. Se propone un caso que tenga utilidad como prototipo base para el análisis de credibilidad con el experto.

1. Introducción

Los centros de atención al usuario (CAU) son parte fundamental de las organizaciones en las que hay un uso intensivo de las tecnologías de la información por parte de usuarios que no son especialistas en este campo. En este contexto, un CAU es un servicio que responde las preguntas de los usuarios, resuelve sus dificultades técnicas, y se encarga, en general, del mantenimiento de la infraestructura TIC de la organización. La gerencia de la organización necesita asegurarse de que el CAU preste su servicio de acuerdo a unos ciertos criterios de calidad, especialmente cuando se da servicio a una o varias organizaciones externas, que probablemente exijan el cumplimiento de unos mínimos, incluso contractualmente y sujetos a penalizaciones en la facturación.

La consecución de los objetivos marcados lleva al gerente a utilizar su experiencia y la

observación de otras organizaciones similares (benchmarking) como herramienta básica para el correcto dimensionamiento y estructuración de los recursos. En este campo, la simulación aparece como una alternativa económica y flexible para ayudar al gerente en su toma de decisiones.

Sin embargo, ganar la credibilidad de los gestores de la organización en la simulación es una tarea ardua, especialmente cuando éstos no han tenido ningún contacto previo con este tipo de metodologías. La presentación de un ejemplo sencillo que modele un caso conocido por los gestores puede servir para incrementar su confianza en la herramienta de simulación empleada. Si además se involucra activamente a los responsables de la empresa en la validación de los modelos utilizados, las posibilidades de éxito de la simulación aumentan notoriamente.

En este artículo se describe brevemente un CAU dependiente del Gobierno de Canarias (sección 2) y se plantea un pequeño modelo que simula el flujo de incidencias típico en ese centro (sección 3). En la sección 4 se explica cómo se ha implementado este modelo con la librería SIGHOS [7], y en la sección 5 se comenta la validación de dicho modelo con un experto. Finalmente se presentan algunas conclusiones y líneas abiertas en la investigación.

2. El Centro de Atención al Usuario de MEDUSA

MEDUSA es el nombre de un proyecto de la Consejería de Educación, Cultura y Deportes del Gobierno de Canarias, cuyo principal objetivo es la integración de las Tecnologías de la

Información y de la Comunicación (TIC) en los centros educativos de Canarias. Dentro de este objetivo, “se procede a la dotación a la comunidad educativa de un centro de atención avanzado dotándolo de las herramientas y personas para que la comunidad educativa pueda disponer de unos niveles de servicio y calidad adecuados” [3]. Este CAU recibe el nombre de Centro de Apoyo a los Sistemas de Información y Usuarios Educativos (en adelante, CASE), y trabaja conjuntamente con el Centro de Gestión del Gobierno de Canarias (Cibercentro) para resolver las incidencias que se producen.

Como cualquier CAU, el trabajo del CASE es la resolución de las incidencias que se producen en los centros bajo su responsabilidad. La forma de estructurar los recursos disponibles influye sobremanera en la eficiencia del servicio. En este sentido, la metodología ITIL (Information Technology Infrastructure Library) [6] se presenta como un marco de referencia ampliamente reconocido y adoptado en el sector de las TIC; y que además soporta y refuerza estándares de calidad como el PD0005 (British Standards Institution’s Code of Practice for IT Service Management) y el ISO9000.

La estructura que emplea el CASE es muy similar a la propuesta en ITIL, utilizando varios niveles que disponen de recursos diferenciados:

- Nivel 1: Atención directa (teléfono, fax, email, ...) de incidencias básicas. Compartido entre el Cibercentro y el CASE.
- Nivel 2: Atención directa (teléfono, fax, email, ...) de incidencias específicas o de mayor complejidad.
- Nivel 3: Atención “in situ” para incidencias no resueltas por los niveles anteriores.

3. Modelo Conceptual

En la Figura 1 puede verse el flujo que sigue una incidencia en el CASE. En dicha figura se han omitido por simplicidad las tareas correspondientes a los recursos de nivel 3. La gestión de las incidencias que llegan al CASE comienza con la llamada de un usuario. El primer operador de nivel 1 que esté disponible atenderá la llamada, tomará los datos del usuario y hará una descripción preliminar de la incidencia. Mediante

esta descripción puede determinarse si se trata de un tipo conocido de incidencia, con un mecanismo de resolución procedimentado; o si se trata de una incidencia que responde a una situación nueva o no procedimentada. Concretamente, se puede hablar de tres tipos de incidencia según su mecanismo de resolución:

- Conocida simple: Incidencias muy frecuentes y cuya resolución está perfectamente procedimentada. Son resueltas por operadores de nivel 1.
- Conocida compleja: Incidencias con una resolución procedimentada, pero que requieren de una mayor especialización para resolverlas. Por tanto, se necesita un operador de nivel 2.
- Desconocida: Requieren hacer un estudio más en profundidad de la situación. Primero se debe consultar la configuración software y hardware contexto de la incidencia; después se define su prioridad en función del número de usuarios y centros afectados, y la importancia de estos usuarios; finalmente se evalúa la incidencia. La evaluación de la incidencia permite descubrir si su clasificación como “desconocida” fue un error, en cuyo caso se devuelve al nivel 1 como “conocida simple” o se asigna a algún operador del nivel 2 como “conocida compleja”. Si se determina que la incidencia es realmente desconocida se inicia un procedimiento de investigación y análisis en aras de conseguir una resolución. La resolución de la incidencia lleva aparejada además la instrumentación de un nuevo procedimiento para resolver todos los casos similares que se presenten.

Sea cual sea el tipo de incidencia, el último paso es siempre su codificación y cerrado.

4. Modelo computacional

El modelo conceptual explicado en la sección anterior se ha implementado utilizando SIGHOS, una librería java para la simulación de sistemas de eventos discretos, especialmente indicada para la simulación de organizaciones.

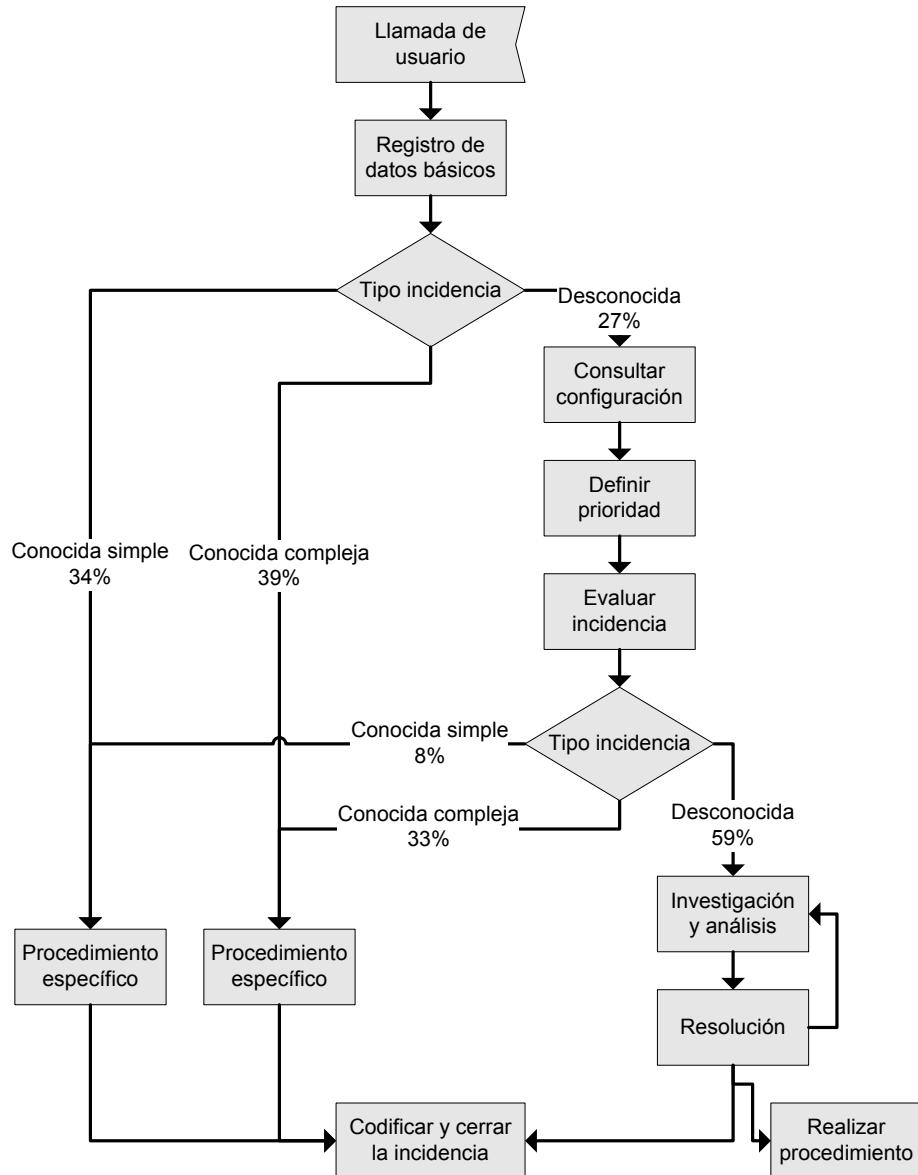


Figura 1. Modelo conceptual de la resolución de incidencias

4.1. SIGHOS

SIGHOS sigue una orientación al proceso [5], lo que permite modelar de forma más directa los

flujos de trabajo (workflows) que maneja una empresa. SIGHOS permite realizar la descripción de estos flujos de trabajo, así como la definición de actividades, categorías o tipos de recursos, y los recursos en sí mismos utilizando Java, o a través de documentos en formato XML [2].

Algunas de las características más destacables de SIGHOS son:

- Implementa los patrones 1 (sequence), 2, 3, 4, 5 y 20 de flujo de trabajo (workflow patterns) definidos en [9].
- Permite la especificación de ciclos complejos para los horarios de los recursos, generación de elementos (incidencias, pacientes...)...
- Permite la utilización de distribuciones de probabilidad para caracterizar duraciones de actividades, disponibilidad de los recursos...
- La mayoría de entidades del sistema pueden definirse con un nivel de prioridad.
- La disponibilidad de los recursos se describe mediante entradas horarias en las que se define el tiempo durante el cual el recurso está disponible y para qué rol o “tipo de recurso” se puede utilizar esa disponibilidad. Un recurso puede estar disponible para varios tipos de recurso en el mismo horario, dotando de mayor flexibilidad a los modelos.
- Una misma actividad puede realizarse con distintos grupos de trabajo (conjuntos de recursos) con diferente duración.
- Las actividades pueden caracterizarse como “interrumpibles” o “no interrumpibles”. Una actividad “no interrumpible”, una vez adquiridos los recursos necesarios para llevarla a cabo, no los libera hasta que no ha terminado, incluso si alguno de los recursos terminó su horario de disponibilidad. Las actividades “interrumpibles” se paran cuando alguno de sus recursos termina su horario, reiniciándose cuando vuelve a tener suficientes recursos disponibles.
- Permite definir actividades que no requieren la presencia física del elemento (de tal manera que éste puede estar haciendo otras actividades simultáneamente). Un ejemplo de esto son los pacientes de un hospital cuando van a hacerse un análisis de sangre: la extracción de la sangre requiere la presencia del paciente; sin embargo, mientras la muestra se estudia o procesa, el paciente puede estar haciendo otras actividades.
- Utiliza el patrón “Listener” [4] de java para “escuchar” lo que ocurre en el simulador y extraer todo tipo de estadísticas. Este mecanismo también puede emplearse para interactuar con el simulador (por ejemplo

generando elementos) respondiendo a cambios del estado del sistema.

El simulador está actualmente en desarrollo bajo licencia CDDL y puede descargarse desde [7]. Al ser código abierto, las clases están sujetas a ser modificadas o ampliadas según las necesidades del usuario.

4.2. Análisis del rendimiento del CASE

Para poder implementar el modelo conceptual del CASE hacen falta los valores de la duración de las actividades, número y disponibilidad de los recursos. Una estimación de estos parámetros, tal como aparece en la Tabla 1, fue facilitada por Pedro J. Baquero, Director Técnico del Proyecto MEDUSA durante 2 años.

Actividad	Duración (seg.)	Recursos necesarios
Registro de datos básicos	25	1 N1
Procedimiento específico nivel 1	30	1 N1
Procedimiento específico nivel 2	425	1 N2
Consultar configuración	25	1 N2
Definir prioridad	15	1 N2
Evaluuar incidencia	20	1 N2
Investigación y análisis	600	1 N2
Resolución	500	1 N2
Codificar y cerrar la incidencia nivel 1	15	1 N1
Codificar y cerrar la incidencia nivel 2	15	1 N2

Tabla 1. Duración de las actividades

A través de datos estadísticos de MEDUSA se puede saber qué porcentaje de incidencias de cada tipo se ha atendido. Con este porcentaje puede calcularse analíticamente el máximo de incidencias que podrían atenderse en el CASE con un número determinado de recursos. Para simplificar el desarrollo matemático se divide el flujo de trabajo de la Figura 1 para obtener todos los posibles caminos simples que puede seguir una incidencia. El resultado se muestra en la Figura 2. La única diferencia significativa con el modelo conceptual es que se ha eliminado la actividad de

“Realizar procedimiento” para simplificar el flujo y porque su efecto influye en la proporción de incidencias desconocidas, que para este ejemplo se toma como fija.

Se considera que se dispone de infinitos recursos de nivel 1, ya que este nivel no está controlado por la gerencia del CASE, sino que pertenece al Cibercentro. Para el nivel 2 se suponen, por ejemplo, 10 operarios que trabajan 8 horas diarias. En cada uno de los caminos de la Figura 2 se han marcado en un color más oscuro las actividades que corresponden a los recursos de nivel 2. Además se han creado dos grupos de flujos, los que requieren recursos de nivel 2 para ser resueltos, y los que pueden resolverse sólo con recursos de nivel 1.

Como se observa en la Tabla 1, cada actividad se resuelve bien con recursos de nivel 1, bien con

recursos de nivel 2. Por tanto, en el flujo de cada tipo de incidencia puede distinguirse perfectamente entre el tiempo requerido de recursos de nivel 1 y nivel 2. Como ya se dijo que se dispone de infinitos recursos de nivel 1, el número máximo de incidencias que se puede resolver está condicionado únicamente por el número de recursos de nivel 2 de que se disponga. Sea t_i al tiempo total de recursos de nivel 2 necesarios para resolver una incidencia de tipo i ; y p_i la proporción de incidencias diarias de ese tipo. Si se considera T_R como el tiempo total de disponibilidad de los recursos por día (en el ejemplo propuesto 80 horas), entonces n_i , tal como se calcula con la ecuación 1, representa el número de incidencias de tipo i que puede resolverse en un día con los recursos de nivel 2 disponibles.

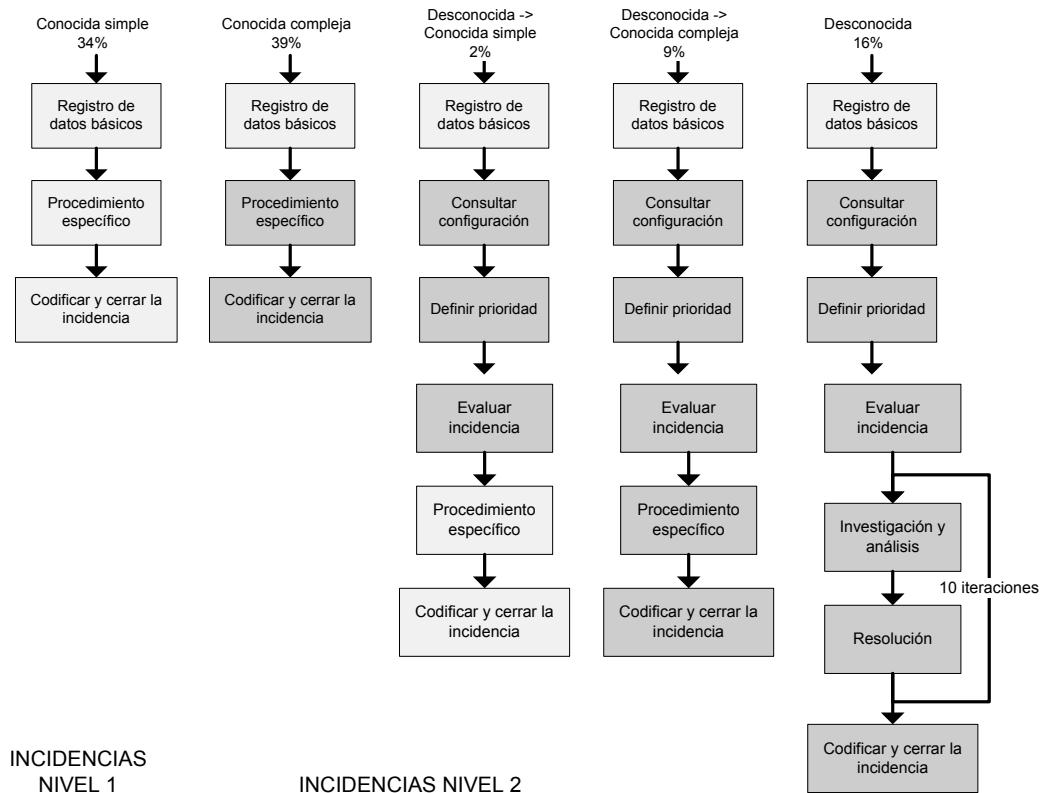


Figura 2. Descomposición en flujos de trabajo simples por tipo de incidencia

$$n_i = \frac{T_R \cdot p_i}{\sum_{\forall i} p_i \cdot t_i} \quad (1)$$

La Tabla 2 muestra el tiempo que dedican los recursos de nivel 2 a resolver cada tipo de incidencia, así como el resultado final de aplicar la ecuación 1. Los recursos de nivel 2 no intervienen en las incidencias “conocidas simples”; por lo tanto, el número de incidencias de este tipo que hay que generar con respecto al total se calcula a partir de las proporciones iniciales con una simple regla de tres. El resultado final es que hay que generar unas 144~145 incidencias diarias, de las cuales 95,53 corresponden a los recursos de nivel 2.

Tipo de incidencia	%	Tiempo N2 (segs.)	Incidencias / día
Conocida compleja	39	440	56,45
Desc. Con. Simple	2	60	2,89
Desc. Con. Compleja	9	500	13,03
Desconocida	16	11075	23,16
Subtotal N2			95,53
Conocida simple	34	0	49,21
Total			144,74

Tabla 2. Número de incidencias / día que pueden resolverse

4.3. Implementación

Para desarrollar un modelo computacional del CAU con una metodología orientada al proceso, se considera cada incidencia como una entidad (o “elemento” en la nomenclatura de SIGHOS) que interacciona con el sistema mediante un flujo de trabajo. Este flujo de trabajo incluye las diferentes etapas del procesado de la incidencia. Cada una de estas etapas puede considerarse una actividad para la que se requiere un número y tipo determinado de recursos, y que utiliza estos recursos durante un periodo de tiempo.

La Figura 3 muestra un extracto del modelo computacional, tal como se implementa en el formato XML de SIGHOS.

5. Verificación y Validación

Tal como se vio en la Tabla 2, si se dispone de 10 recursos de nivel 2 y con las proporciones preestablecidas, el CASE debería ser capaz de resolver como máximo 95,53 incidencias diarias de nivel 2. Para un número menor de incidencias los recursos estarían ociosos; y para un número mayor, ya estarían saturados.

Para poder comparar los resultados del simulador con este valor teórico se estableció como constante la duración de las actividades, el número de iteraciones (donde las hubiera) y el total de incidencias a generar. La única aleatoriedad venía dada por el manejo de la proporción de incidencias de cada tipo: no se puede generar un número no entero de incidencias, por lo que iba a resultar imposible cuadricular el comportamiento de la simulación con el valor analítico. Se planteó hacer una serie de experimentos con dos objetivos:

- Que la proporción media de incidencias generadas coincidiese con la planteada inicialmente.
- Que la media de incidencias de nivel 2 resueltas por día se aproximase lo más posible al valor calculado analíticamente.

La comprobación de la proporción de incidencias se podía reducir a un problema de verificación de las herramientas de generación de números de aleatorios del simulador. SIGHOS emplea el paquete de generación de números aleatorios de SimKit [8], por lo que sólo se procedió a una verificación empírica de su buen comportamiento.

Para el segundo objetivo era fundamental escoger correctamente la manera de generar las incidencias. Aquí surgían dos dificultades: por un lado, no se disponía de datos exactos de la distribución de la llegada de incidencias; por otro lado, la fórmula analítica parte de un aprovechamiento total de la disponibilidad de los recursos. El segundo de los problemas supone que, ante una llegada escalonada de las incidencias, los recursos disponibles pueden pasar de momentos totalmente ociosos a períodos en los que se vieran desbordados por las incidencias. Como solución de compromiso, se optó por generar todas las incidencias al principio del día, maximizando por tanto el aprovechamiento de los recursos.

```

<Model id="1">
    <description>Modelo del CASE de MEDUSA </description>
    <baseTimeUnit>SECOND</baseTimeUnit>
    <!-- Recursos de nivel 2 -->
    <resourceType id="2">
        <description>Level 2</description>
    </resourceType>
    <!-- Grupo de trabajo de nivel 1 -->
    <workGroup id="1">
        <description>Level 1</description>
        <role>
            <rt_ref model_id="1" id="1"/>
            <units>0</units><!-- Para simular infinitos recursos -->
        </role>
    </workGroup>
    <!-- 1 recurso de nivel 2 -->
    <resource id="2">
        <description>Level 2 Resource</description>
        <units>10</units>
        <timeTable>
            <rt_ref model_id="1" id="2"/>
            <cycle type="PERIODIC">
                <timeUnit>HOUR</timeUnit>
                <startTs>8</startTs>
                <iterations>0</iterations>
                <period>
                    <random>
                        <function>ConstantVariate</function>
                        <param>24</param>
                    </random>
                </period>
            </cycle>
            <dur timeUnit="HOUR">8</dur>
        </timeTable>
    </resource>
    <!-- Actividad 1: Registro de datos básicos -->
    <activity id="1" presential="true" interruptible="false">
        <description>Data Registration</description>
        <priority>0</priority>
        <workGroup id="1">
            <workGroup model_id="1" id="1"/>
            <priority>0</priority>
            <duration>
                <random>
                    <function>ConstantVariate</function>
                    <param>25</param>
                </random>
            </duration>
        </workGroup>
    </activity>
    <!-- Actividad 3: Procedimiento específico de nivel 2 -->
    <activity id="3" presential="false" interruptible="true">
        <description>Specific procedure Complex Known</description>
        <priority>20</priority>
        <workGroup id="1">
            <workGroup model_id="1" id="2"/>
            <priority>0</priority>
            <duration>
                <random>
                    <function>ConstantVariate</function>
                    <param>425</param>
                </random>
            </duration>
        </workGroup>
    </activity>

```

```

<!-- Tipo de incidencia "Conocida simple" -->
<elementType id="1">
  <description>Simple Known</description>
  <priority>1</priority>
</elementType>
<!-- Workflow de las incidencias "Conocidas simples" -->
<rootFlow id="2">
  <description>Simple Known</description>
  <flow>
    <sequence>
      <single>
        <act_ref model_id="1" id="1"/>
      </single>
      <single>
        <act_ref model_id="1" id="2"/>
      </single>
      <single>
        <act_ref model_id="1" id="9"/>
      </single>
    </sequence>
  </flow>
</rootFlow>
</Model>

```

Figura 3. Extracto del modelo en XML del CASE de MEDUSA

Aunque el número de incidencias resueltas por día parecía el resultado más adecuado para ser comparado con el valor analítico, se descubrió que este valor era en general mucho menor que el esperado. El problema lo originaba un comportamiento cuyos efectos eran despreciados en el modelo analítico, pero de vital importancia en el modelo computacional: el orden de resolución de las incidencias. Por ejemplo, para una variación a mayor en la proporción de incidencias de tipo “desconocido” (que como se ve en la Tabla 2, tienen un tiempo de resolución mucho mayor que el resto) se producía una drástica reducción en el número de incidencias resueltas. El problema de medir el total de incidencias resueltas es que se trata de un valor entero, lo que implica que las incidencias en proceso de resolución no se tienen en cuenta. Tras discutirlo con el experto, se decidió tomar el tiempo total dedicado por los recursos en resolver cada tipo de incidencia ya que, al dividir ese tiempo por el tiempo requerido para resolver una incidencia de ese tipo, se puede obtener una mejor estimación del número de incidencias terminadas. La razón de esta mejora es que el nuevo valor incluye las porciones de incidencias no terminadas.

Para comparar los resultados con el valor analítico se tomaron la media y desviación estándar de 2000 simulaciones independientes de un día, con valor respectivo de 93,4 y 5,69. Para

validar el uso de esa media se hizo una prueba de la bondad del ajuste a una normal mediante el test de Chi cuadrado, como se observa en la Figura 4. El test de Chi cuadrado para ese ajuste dio un valor p de 0,54 y un valor del estadístico de 11,9 para 13 grados de libertad, lo que permite, para un nivel de significancia del 10%, aceptar la hipótesis nula.

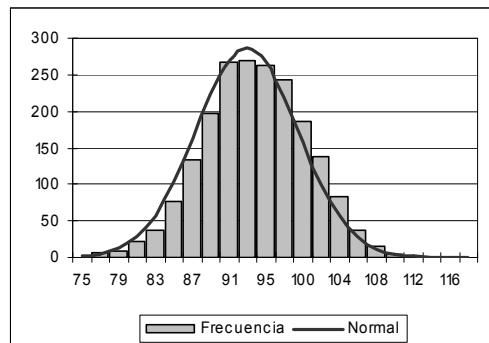


Figura 4. Ajuste del resultado a una normal

Con estos resultados se obtiene un error relativo respecto al valor analítico propuesto del 2,45%. La Tabla 3 muestra la proporción de incidencias de cada tipo creadas por el simulador, resueltas completamente, y las resueltas

contabilizando las porciones de incidencias no terminadas.

Tipo de incidencia	% Creadas	% Resueltas	% Resueltas (corregido)
Conocida compleja	39	43	39,5
Desc. Con. Simple	2	2	2
Desc. Con. Compleja	9	10	9
Desconocida	16	8	15
Conocida simple	34	37	34,5

Tabla 3. Resultados de la simulación: proporción de incidencias

Como última fase de la validación se discutieron estos resultados con el experto, quien corroboró su validez tras explicarle los motivos de las pequeñas diferencias entre el esperado y lo obtenido.

6. Conclusiones y líneas abiertas

En este artículo se ha presentado un modelo básico del CAU del proyecto MEDUSA. El modelo se ha implementado con la librería de simulación SIGHOS y los resultados han sido validados con un experto.

Este modelo debe servir como base para futuras simulaciones del CAU, bien para la redistribución y correcto dimensionamiento de recursos de cada nivel [1]; bien para la optimización de los flujos de las incidencias.

En este sentido se está trabajando con modelos que incorporan acciones de control del gerente mediante lógica difusa para asegurar una cierta calidad de servicio (QoS).

Agradecimientos

Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por MEC y fondos FEDER con referencia DPI2006-01803, titulado “Aplicación de la simulación paralela de sistemas de eventos discretos y los sistemas multiagente en la

asignación de recursos de un centro de atención de usuarios”.

Referencias

- [1] Aguilar, R. M., Castilla, I., Muñoz, V., Estévez, J. I., Martín, C. A., Moreno, L. 2005. Predictive Simulation for Multiagent Resource Distribution: An Application in Hospital Management. International Mediterranean Modeling Multiconference. Marseille-France. 2005.
- [2] Aguilar, R. M., Muñoz, R., Castilla, I., Martín, C. A., Muñoz, V. An interface to integrate workflow diagrams into discrete event simulation. International Mediterranean Modeling Multiconference. Barcelona-España. 2006.
- [3] Consejería de Educación, Cultura y Deportes del Gobierno de Canarias. Pliego de Prescripciones Técnicas del Proyecto Medusa Primario. 2001.
- [4] Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA. 1995
- [5] Pidd M. Computer Simulation in Management Science. 5th Edition. John Wiley & Sons, Chichester, UK. 2004.
- [6] Rudd, C., Hodgkiss, G. The IT Infrastructure Library: An Introductory Overview of ITIL. Ed. Alison Cartlidge. itSMF Ltd. 2004.
- [7] SIGHOS webpage. <http://sourceforge.net/projects/sighos>. Accedido 3/5/2007.
- [8] Simkit webpage. <http://diana.cs.nps.navy.mil/simkit>. Accedido 3/5/2007.
- [9] van der Aalst, W. M. P., ter Hofstede, A., Kiepuszewski, B. and Barros, A.P. “Workflow Patterns”. 2002.

MEST y CEST: Herramientas para la Predicción de Descarga Electrónica en Guias de Onda

Francisco Pérez, Juan de Lara, Manuel Alfonseca
Dept. de Ingeniería Informática
Escuela Politécnica Superior
Univ. Autónoma de Madrid
28049 Madrid
[\[francisco.perez_jdlara.manuel.alfonseca\]@uam.es](mailto:[francisco.perez_jdlara.manuel.alfonseca]@uam.es)

Luis Conde
Dept. Física
E.T.S. Ingenieros Aeronáuticos
Univ. Politécnica de Madrid
28040 Madrid
lconde@faia.upm.es

Luis Galán
Dept. Física Aplicada
Univ. Autónoma de Madrid
28049 Madrid
luis.galan@uam.es

Resumen

En este artículo se presentan las herramientas desarrolladas dentro de un proyecto financiado por la Agencia Espacial Europea para la simulación y predicción de la descarga electrónica que se produce dentro de dispositivos de radiofrecuencia (RF) en distintas condiciones atmosféricas.

1. Introducción

Los efectos Multipactor [1,2] y Corona [3] son descargas eléctricas espúreas que se producen dentro de guías de onda de componentes aeroespaciales, causando la destrucción o la inutilización de los dispositivos de RF. Estas descargas son causadas por un número inicialmente pequeño de cargas eléctricas, en su mayoría electrones, asociados a los rayos cósmicos o la radioactividad normal del ambiente en el espacio, que se aceleran por el campo electromagnético y colisionan con las paredes de la guía de onda, pudiendo arrancar nuevos electrones; o con los átomos neutros del gas en el interior del dispositivo, pudiendo ionizarlos. Estos procesos multiplican la población de electrones libres hasta producir una descarga por avalancha eléctrica. En el caso de Multipactor (MP), la descarga tiene lugar en el vacío. En el caso de la descarga de Corona (CR) la descarga se produce a bajas presiones, con la presencia de un gas neutro en la guía de ondas.

Hemos desarrollado dos simuladores (MEST [4] y CEST) con el objetivo de estudiar las condiciones de descarga, así como el análisis de buenos recubrimientos para las guías de onda que eviten en lo posible su aparición. Los simuladores

implementan dos modelos físicos novedosos cuya característica principal es que los electrones se modelan y tratan individualmente. En el caso de MEST para el efecto MP, se lleva a cabo una simulación discreta, tomando como eventos las colisiones de los electrones contra las paredes. En el caso de CEST para CR, la simulación es híbrida, añadiendo incertidumbre mediante ecuaciones estocásticas de movimiento, ya que las colisiones con los átomos del gas no son predecibles, como son las colisiones con las paredes.

En ambos casos se han obtenido caracterizaciones de las descargas que coinciden con datos experimentales. De esta forma, su uso evita realizar experimentos reales, más costosos en tiempo y recursos. Consideramos que son una buena ayuda en la toma de decisiones para el diseño (recubrimiento, geometría, voltaje operativo, etc.) de componentes aeroespaciales.

El resto del artículo se organiza de la siguiente manera: en la sección 2 describimos los fenómenos MP y CR. En la sección 3 se describen los modelos utilizados, en la 4 las técnicas de simulación utilizadas. El punto 5 muestra las herramientas, el 6 los resultados y se finaliza con las conclusiones y trabajo futuro.

2. Descripción de MP y CR

En el caso de MP [1,2], la descarga se produce en vacío, por lo que la producción de nuevos electrones se debe exclusivamente a la colisión de los electrones existentes con las paredes. Los electrones son desplazados por el campo electromagnético presente en la guía de onda. Si las colisiones se sincronizan con la señal, entonces es cuando puede producirse la descarga MP.

Por tanto en MP, es importante conocer cómo se arrancan los electrones en función del tipo de colisión, de las propiedades del material en superficie (SEY, *secondary emission yield*), del ángulo de incidencia del electrón en la colisión y de la rugosidad de la superficie. Cuando un electrón impacta con la pared, éste puede ser absorbido, reflejado (*backscattered*) de manera elástica o inelástica, o bien se pueden generar una serie de electrones *verdaderos* por emisión secundaria. Cada material se caracteriza por una curva de SEY [5], que da el número de electrones producidos (distinguiendo cada tipo) en función de la energía del electrón en el momento del impacto, para un ángulo de incidencia del electrón normal a la superficie.

En el caso del efecto CR, la descarga se produce a bajas presiones, por lo que a las colisiones con las paredes, hay que añadirle las colisiones con los átomos del gas existente en la guía de onda. Estas producen dos efectos relevantes: una distorsión de la trayectoria del electrón, y la producción de nuevos electrones por la ionización de los átomos neutros del gas. Por tanto, además de caracterizar el material de las paredes, se hace necesaria una descripción del gas. En este fenómeno, dependiendo de la presión, se pueden producir descargas debido puramente a los electrones de las paredes (a muy bajas presiones), a ionización de los átomos del gas (a presiones más altas) o a una mezcla de ambas (zona intermedia).

3. Modelos

En los dos modelos hemos asumido una serie de simplificaciones para posibilitar su computación y mejorar la eficiencia de su simulación, suprimiendo otros procesos físicos existentes no relevantes para la descarga. Así, la geometría de la guía de onda se simplifica a dos placas paralelas, tanto finitas como infinitas, en el plano xy , una en $z = 0$, y la otra en $z = d$, siendo d la distancia entre placas, sobre la que actúa un campo de radiofrecuencia uniforme y perpendicular a la disposición de las placas. Se descarta además el campo eléctrico interno generado por las cargas de los electrones e iones del gas por ser despreciable en relación al campo RF externo. Los efectos de borde de la geometría no son considerados, excepto en el caso de placas finitas, cuando un

electrón sale de la geometría, en cuyo caso es eliminado. La influencia del campo viene definida como:

$$\begin{aligned} E_z(t) &= E_0 \cdot \sin(\omega t), \quad E_x = E_y = 0; \\ E_0 &= -V_0/d, \quad \omega = 2\pi f \end{aligned} \quad (1)$$

De manera adicional, en ambos modelos consideramos que la trayectoria de los electrones no se ve afectada por otros electrones.

Las dos secciones siguientes describen con más detalle cada uno de los modelos.

3.1. Modelo para Multipactor

El aspecto principal del modelo de MP es la colisión de un electrón con la pared, que puede dar lugar a la generación de cero (absorción) o más electrones (reflejados elásticos, inelásticos o secundarios verdaderos). La energía cinética de los electrones generados ha de ser menor o igual que la del electrón que impacta. El SEY se modela como una curva que depende de la energía del electrón primario que impacta (E_p) y su ángulo (ϕ), y en la que cada tipo de electrón generado hace su contribución. Su valor nos da un valor estadístico de cuántos electrones se generan en cada impacto. La Figura 1 muestra una curva típica, donde E_1 y E_2 son los valores de la energía con el primer y segundo corte del SEY con el valor 1. E_{max} es la energía en la que el SEY alcanza su máximo valor (σ_{max}). En nuestro modelo [4], hemos considerado para cada tipo de colisión, primero la contribución debido a la energía (E_p), y luego un término multiplicativo adicional que da la dependencia con el ángulo de llegada.

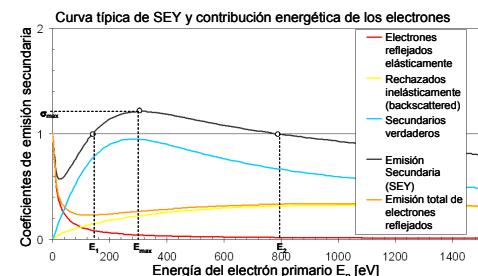


Figura 1. Curva típica de SEY.

Utilizando los valores del SEY, podemos calcular la *probabilidad* de cada tipo de colisión. Las

colisiones elásticas e inelásticas producen siempre un electrón. En el primer caso, el electrón primario es reflejado; en el segundo el electrón penetra en el material, queda retenido, pero arranca un electrón de algún átomo, que sale reflejado sin pérdida de energía [5]. En el caso de los secundarios verdaderos la interacción con el material es más compleja. Su número se calcula con una probabilidad con distribución de Poisson, y su energía iterativamente mediante una función de probabilidad inversa acumulativa [4]. Finalmente el ángulo de salida se calcula con la distribución de la ley del coseno.

3.2. Modelo para Corona

El modelo utilizado en CEST amplia el modelo previo, soportando la interacción de los electrones libres con los átomos de un gas dentro de la guía de onda. Este modelo, al igual que el previo, modela los electrones individualmente, sin embargo, es computacionalmente inviable modelar los átomos del gas de la misma manera, dada su densidad ($n_a \approx 10^{17} \text{ cm}^{-3}$). Por ello, se hizo necesario utilizar un modelo macroscópico para las colisiones entre los electrones y los átomos, en las que los primeros ven afectadas sus trayectorias por dos fuerzas, una de fricción (2), que simula la resistencia que encuentra un electrón al traspasar una nube de átomos, y otra de difusión (3), aleatoria, que simula la desviación de la trayectoria del electrón debida a las reiteradas colisiones con los átomos del gas.

$$F_r = -m_e v_{ea} u_e \quad (2)$$

$$F_d = \sqrt{C} \cdot \Gamma(t)$$

$$C = \frac{2K_B T_e}{m_e} v_{ea} \quad (3)$$

$$\langle \Gamma(t) \rangle = 0$$

$$\langle \Gamma_i(t) \Gamma_j(t') \rangle = \delta_{ij} \delta(t - t')$$

v_{ea} es uno de los parámetros que caracterizan al gas, correspondiendo a la *frecuencia de colisión* electrón-átomo. $\Gamma(t)$ representa el ruido blanco estocástico que modela la incertidumbre en el sentido de la trayectoria resultante del electrón tras la colisión.

Estas dos fuerzas descritas, más la influencia determinista del campo de radiofrecuencia en el eje z , integradas, determinan las ecuaciones de

movimiento estocástico de los electrones para cada una de las coordenadas, mediante un modelo de ecuaciones de Langevin [6].

$$\begin{aligned} \frac{\partial r_e}{\partial t} &= u_e \\ \frac{\partial u_e}{\partial t} &= -\frac{e}{m_e} E(t) - v_m u_e + \sqrt{C} \cdot \Gamma(t) \end{aligned} \quad (4)$$

Adicionalmente a las consecuencias de las colisiones electrón-átomo sobre las trayectorias de los electrones, estas colisiones pueden suponer la ionización de los átomos que reciben el impacto. Desgraciadamente, no se dispone de información empírica a nivel atómico de este proceso, por lo que es necesario utilizar un modelo estadístico, basado en las dos premisas siguientes: aparecen electrones nuevos por ionización, si el electrón que colisiona con el átomo tiene una energía mayor a la *energía de ionización* del gas, y si dicho electrón ha superado el *camino libre medio* del gas, esto es, la distancia media estadística recorrida por un electrón entre dos ionizaciones consecutivas.

4. Tipos de Simulación Utilizada

En ambos simuladores, el diseño se ha guiado por el paradigma de la orientación a objetos. Para la simulación del modelo MP, hemos utilizado simulación discreta [7]. Así, los eventos relevantes (que hacen que avance del tiempo de simulación se produce) son las colisiones de los electrones con las paredes. Esto permite una simulación eficiente, ya que el instante de colisión puede calcularse de manera analítica. El esquema de la simulación se refleja en la Figura 2.

Por el contrario, la simulación del modelo de CR es continua, donde se integran las ecuaciones estocásticas del modelo y además suceden eventos de tipo contingente. Para la integración estocástica se ha utilizado el método de Milstein [8]. Como los eventos son contingentes (e.d. no se programan en el futuro) no se hace necesaria una cola de eventos futuros. La Figura 3 muestra el esquema de la simulación de CEST.

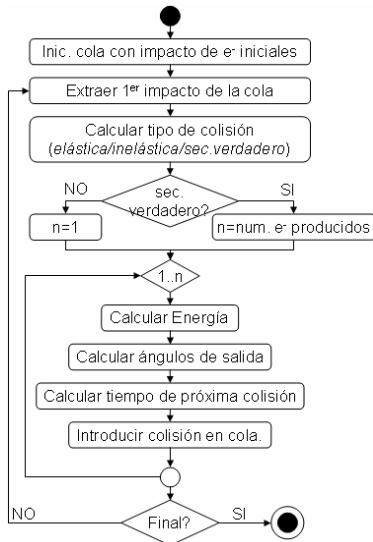


Figura 2. Esquema de simulación MEST

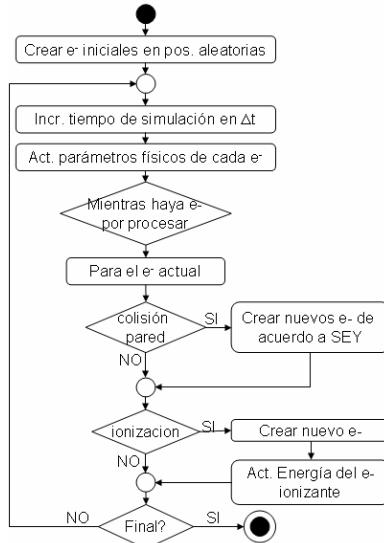


Figura 3. Esquema de simulación de CEST

En el proceso de ionización de un átomo del gas, un nuevo electrón con energía 0 es producido, reduciendo la energía del electrón que produce la ionización en la *energía de ionización* del gas.

En ambos tipos de simulación, un aspecto clave son las condiciones de descarga. En MEST, se considera que hay descarga cuando el número de electrones crece por encima del número inicial

multiplicado por una variable, que ha de ser mayor que el valor del SEY máximo; y se considera que no hay descarga cuando el número decrece por debajo del número inicial dividido entre este valor. En CEST la condición es más compleja ya que depende de la zona de descarga (presión baja, media o alta). A bajas presiones las condiciones son similares a MEST (ya que las colisiones en paredes predominan). A altas presiones, las descargas se dan por ionización y suelen ser mucho más rápidas (crecimiento sobre-exponencial). A presiones intermedias se tiene una superposición de ambos efectos.

5. Herramientas

Con el objetivo puesto en la usabilidad de las herramientas por usuarios no expertos en simulación, se ha creado una interfaz gráfica amigable para ambos simuladores, con una pantalla principal en la que poder introducir los datos del experimento, como se ve en la Figura 4, más otros diálogos complementarios en los que especificar las propiedades del material, y/o del gas para el cual se va a realizar las simulaciones.

En estas herramientas, un experimento consta de un grupo de simulaciones individuales para un rango de valores, para los cuales se quiere calcular la región en la que se produce la descarga. Esta región se muestra como salida gráfica principal, a la que hay que añadir otras específicas para el experimento tipo del fenómeno simulado (e.g. modos), y otras en las que se muestra de forma detallada la evolución temporal de un valor en la simulación actual (e.g. número de electrones, o energía media).

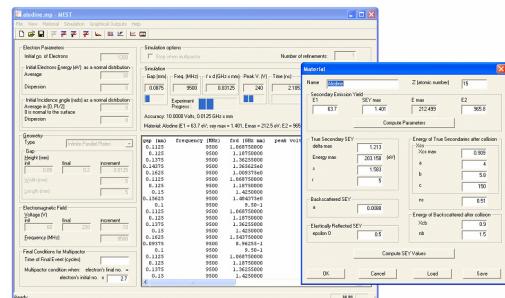


Figura 4. Pantalla principal de MEST, con el diálogo de las propiedades de los materiales

5.1. MEST

La Figura 4 muestra la pantalla principal de MEST, donde a la izquierda se ven los parámetros de simulación, que son, de arriba abajo: condiciones iniciales de simulación, compuestas por el número inicial de electrones, energía y ángulo de incidencia inicial; la geometría, placas paralelas finitas o infinitas, sus dimensiones y un rango de valores para la distancia entre placas; el campo electromagnético, frecuencia y un rango de valores para el voltaje; y finalmente las condiciones de finalización, las cuales determinan la precisión con la que la simulación dará resultados correctos. Las propiedades del material simulado se recogen en el diálogo mostrado en la misma figura, en él se pueden ver los dos posibles conjuntos de datos para introducir las propiedades del material, en la mitad superior el modelo extendido explicado previamente, y en la mitad inferior, el modelo clásico basado en E_1 , E_2 y σ_{max} , que se puede ver en la Figura 1. Dado un conjunto de valores, las aplicaciones calculan el equivalente para el otro modelo.

La salida principal para la simulación MP es el mapa de voltajes contra el producto de la frecuencia con la distancia entre placas ($V\text{-}f\text{-}d$), (Figura 5). En ella se muestra, para la combinación de todos los valores posibles del experimento, un punto de un determinado color, indicando si el resultado de la simulación (de descarga o no) con un grado de seguridad. Los colores van de rojo (alta probabilidad de descarga) al azul (alta probabilidad de no descarga).

También es posible obtener los *modos* de funcionamiento de la descarga (ver p.ej. la Figura 10). Los modos, tal como fueron definidos por Hatch & Williams en su teoría del electrón resonante [9], son el número de semiperíodos de la onda electromagnética que tarda un electrón de llegar de una placa a la contraria. Así, el modo 1 es un semiperíodo, el modo 2 tres semiperíodos, y así sucesivamente.

MEST permite otras salidas gráficas asociadas a la evolución de las simulaciones individuales. La Figura 6, muestra la evolución temporal de la población de electrones. Se pueden observar los picos de producción de nuevos electrones en las colisiones con las paredes, indicando el *modo* en el que trabaja el experimento simulado. La Figura 7 muestra la posición de los electrones entre las

placas paralelas a cada instante, indicando con el color de cada electrón el número de colisiones que ha experimentado desde el inicio de la simulación.

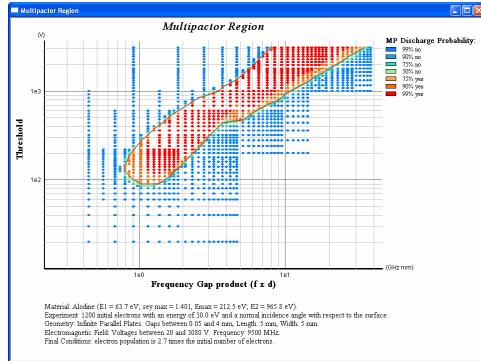


Figura 5. Mapa V-f-d para el material Alodine

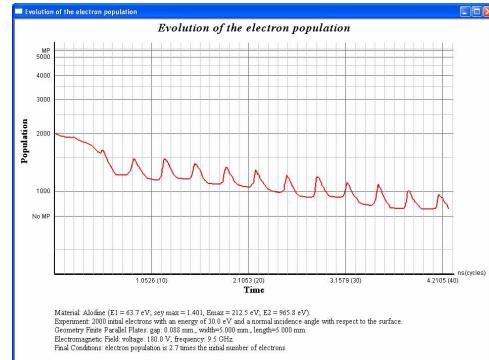


Figura 6. Evolución del número de electrones en una simulación de Alodine con Mest.

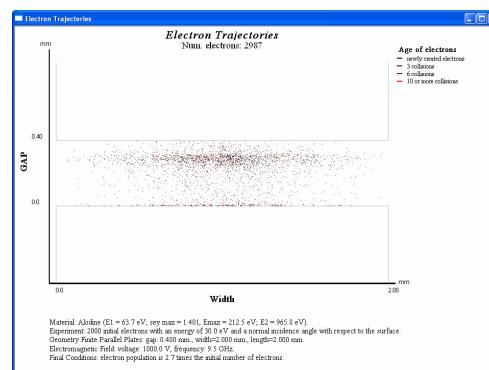


Figura 7. Animación de la nube de electrones en una simulación de Alodine con Mest.

5.2. CEST

La aplicación para la simulación del efecto Corona, CEST, tiene la misma estructura básica que MEST, con la complicación adicional de que, debido al tipo de simulación utilizada, son necesarios parámetros adicionales, como el intervalo de tiempo para el paso de simulación; y el número de pasos estocásticos necesarios dentro de cada paso de simulación.

Adicionalmente, la simulación del efecto CR requiere de un rango de presiones sobre el que realizar experimentos, por lo que ahora tenemos dos tipos posibles de experimentos, los que buscan obtener como resultado un mapa del tipo obtenido en MEST, mostrando la región $V-f-d$ para una presión dada (ver Figura 5); o un mapa en el que se muestra, para una frecuencia fija, el resultado de descarga o no enfrentando voltaje contra presión ($V-p$), también llamado curva de Paschen, como se muestra en la Figura 8.

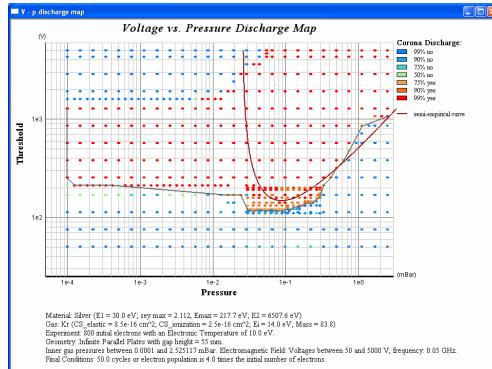


Figura 8. Mapa $V-p$ para Plata y Kripton.

Al igual que en MEST, CEST también posee las dos salidas anteriormente expuestas en tiempo de ejecución, con la salvedad de que en la salida que muestra la evolución de la población de electrones, también se muestra la discriminación por origen de los electrones existentes, esto es, muestra adicionalmente la evolución de los electrones que se han arrancado de las paredes por emisión secundaria y la evolución de los que se han producido por ionización. Esto supone un valor añadido a la hora de delimitar las regiones, ya que permite clasificar las simulaciones entre las pertenecientes a MP, en las que los nuevos electrones provienen principalmente de la emisión

secundaria, localizada a la izquierda de la Figura 8; de las que pertenecen a la zona de CR, situadas a la derecha de la figura, en la zona delimitada por la curva semi-empírica en rojo, en los que el mecanismo de emisión es por ionización de los átomos neutros; de la zona intermedia, en la que cooperan ambos mecanismos de producción de electrones, región conocida como MP Plasma.

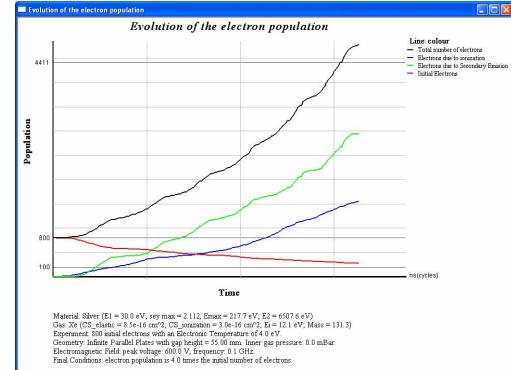


Figura 9. Evolución de la población de electrones en la región Multipactor Plasma.

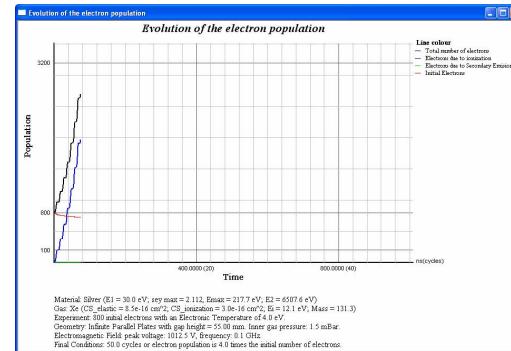


Figura 10. Evolución de la población de electrones en la región de Corona.

La Figura 9 muestra una curva con la evolución de los electrones en la región de MP Plasma. La línea azul muestra los electrones producidos por ionización, mientras la línea verde muestra los generados por emisión secundaria. La figura corresponde a un experimento en la zona intermedia, donde los electrones de la descarga se producen tanto en las paredes como en el volumen. La Figura 10 muestra un experimento a más altas presiones, donde todos los electrones se

producen por ionización. Nótese que la descarga es mucho más rápida en esta zona.

CEST también proporciona una salida gráfica en la que se muestra la temperatura electrónica de la nube de electrones en el tiempo, en la que se puede estudiar la correlación entre la ocurrencia de descargas y energía media de los electrones libres.

6. Resultados

Se ha realizado una extensa validación de los componentes y modelos que forman parte de los dos simuladores. Se realizó primero una validación estructural, en la que se comprobó la corrección de la implementación del modelo; posteriormente se llevó a cabo una validación individual de los resultados proporcionados por cada módulo, entre los que se encuentra la distribución de los ángulos de salida en la emisión de electrones secundarios en las paredes de la guía, la comprobación del espectro de energía emitido por los electrones secundarios ($\partial\sigma/\partial E$), y la comprobación de la corrección de los efectos de fricción y difusión sobre la trayectoria de los electrones capaces de colisionar con átomos del gas. Finalmente, se realizó una validación de los resultados finales, comparándolos con las regiones de descarga MP calculadas con el modelo del electrón resonante de Hatch & Williams [9], así como de la adecuación de los modos de funcionamiento con dicho modelo, como muestran las Figuras 11 y 12; la correlación de los resultados de CEST con MEST a bajas temperaturas; la adecuación, en los rangos de valores válidos de simulación, al modelo semiempírico de descarga CR; y la validación de la región de descarga con experimentos controlados reales, para un rango limitado de presiones [10] (Figuras 13 y 14).

Si bien con CEST a bajas presiones se reproducen los resultados de MEST en el vacío, en este caso, es recomendable el uso de MEST, ya que la simulación es más eficiente y proporciona información más detallada (por ejemplo los modos).

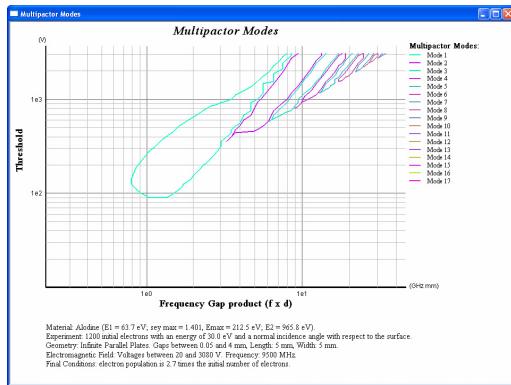


Figura 11. Modos del mapa de V-f-d para Alodine.

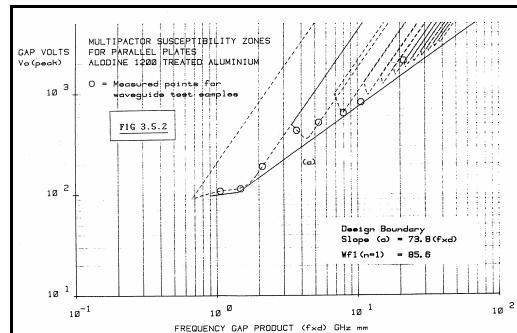


Figura 12. Datos experimentales de ESTEC para Alodine, ajustados con el modelo de Hatch & Williams [9]

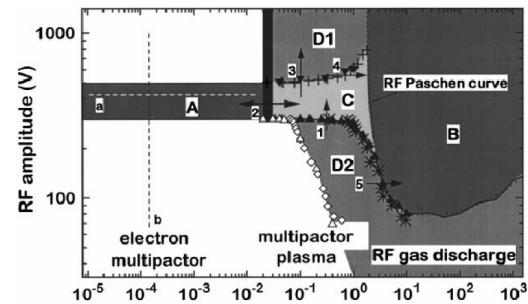


Figura 13. Resultados de [10]. Gas: Ar, Material en placas: Acero inoxidable, 50 MHz (eje x en Pa).

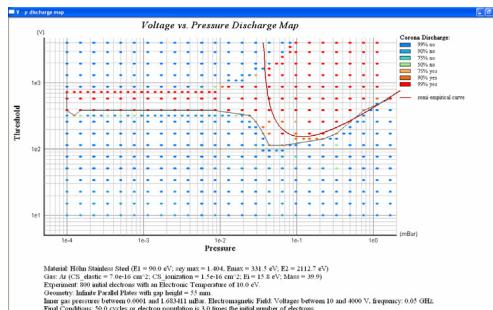


Figura 14. Reproducción del experimento de Höhn (eje x en mBar).

Como se ha podido comprobar, gracias a la validación realizada estamos en disposición de poder decir que la herramienta cumple con las expectativas iniciales de servir como referencia de toma de decisiones para la fabricación de guías de onda capaces de evitar los efectos MP y CR. No obstante ha de tenerse en cuenta que los resultados vienen limitados por los rangos de confianza determinados para su funcionamiento.

7. Conclusiones y Trabajo Futuro

En este artículo hemos presentado MEST y CEST, dos herramientas para la simulación de descargas eléctricas en guías de onda bajo diferentes condiciones de presión. Ambos simuladores se basan en modelos novedosos, en los que los electrones se modelan individualmente. Los resultados de los simuladores se han validado experimentalmente, obteniéndose resultados satisfactorios. Y de manera adicional con CEST hemos validado resultados de MEST.

En el futuro planeamos el uso de un enfoque de familia de productos y tecnologías generativas basadas en lenguajes específicos de dominio para la personalización de la interfaz, salidas gráficas e incluso los modelos de simulación.

Referencias

- [1] P. T. Farnsworth, "Television by electron image scanning" J. Franklin Inst., vol. 218, pp. 411–444, 1934.
- [2] J. R. M. Vaughan, "Multipactor," IEEE Trans. Electron Devices, vol. 35, no. 7, pp. 1172–1180, Jul. 1988.
- [3] Y.P. Raizer. Gas discharge physics. Springer Verlag, Berlin, 1991.
- [4] J. de Lara, F. Pérez, M. Alfonseca, L. Galán, I. Montero, E. Román, D. Raboso."Multipactor Prediction for On-Board Spacecraft RF Equipment with the MEST Software Tool". IEEE Transactions on Plasma Science, Vol 34(2), April 2006, pp: 476- 484.
- [5] Furman, M. A., Pivi, M. T. "Probabilistic Model for the Simulation of Secondary Electron Emission". Physical Review Special Topics – Accelerators and Beams, Vol 5, 124404 (2002).
- [6] J. Qiang and S. Habib, "A second order stochastic leap-frog algorithm for Langevin simulation", Proceedings of the XX Linac Conference, pp. 89-91, Monterrey, California USA. August 21-25, (2000).
- [7] Fishman, G. S. Discrete Event Simulation. Modeling, Programming and Analysis. Springer Series in Operations Research. 2001.
- [8] Higham, D. J. "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations" . SIAM Review, Vol 43(3), pp.: 525-546. 2001.
- [9] Hatch, A. J., Williams, H. B. 1954. "The Secondary Electron Resonance of Low-Pressure High-Frequency Gas Breakdown". Journal of Applied Physics, Vol 25, Num 4, April 1954.
- [10] F. Höhn, W. Jacob, R. Beckmann and R. Whilhelm. "The transition of a multipactor to a low pressure gas discharge". Phys. Plasmas, 4, (10), pp. 940-4111, (1997).

Métodos de reducción del tiempo de cálculo en la exploración de espacios de estados generados a partir de modelos de redes de Petri Coloreadas

Mercedes E. Narciso Farias

Dept. de Telecomunicación e
Ingeniería de Sistemas
Universitat Autònoma de Barcelona
UAB
08193 Bellaterra (España)
Mercedes.Narciso@uab.es

Miquel A. Piera i Eroles

Dept. de Telecomunicación e
Ingeniería de Sistemas
Universitat Autònoma de Barcelona
UAB
08193 Bellaterra (España)
MiquelAngel.Piera@uab.es

Resumen

En los últimos años han aparecido en el mercado una gran diversidad de paquetes informáticos de ayuda a la toma de decisiones en el campo de los sistemas logísticos, sin embargo, los modelos con los que trabajan no contemplan la dificultad en el modelado de los recursos ni tampoco las interacciones entre ellos. Para tratar la complejidad que presenta la optimización de problemas logísticos caracterizados por la diversidad de relaciones causa-efecto y un número elevado de variables de decisión altamente acopladas se ha propuesto una metodología que consiste en automatizar la toma de decisiones mediante la resolución de problemas de optimización, cuya idea central es la generación del espacio de estados de sistemas modelados mediante Redes de Petri Coloreadas, y explorar ese espacio de estados a fin de encontrar una solución óptima o quasi óptima a problemas logísticos. Para mejorar la eficiencia en la generación del espacio de estados, en este artículo se presentan dos métodos para reducir el tiempo de cálculo durante la exploración del espacio de estados, los cuales se han denominado Método de los Caminos Prohibidos y Método de Evaluación Jerárquica de Guardas.

1. Introducción

Aunque existen diferentes definiciones en la literatura, en este artículo se considera la logística como la planificación y control del flujo de materiales y la información asociada en cualquier tipo de organización. En general, los sistemas logísticos se componen de un conjunto de recursos que interactúan entre ellos mediante unidades de transporte [2,9].

Un aspecto importante es la toma de decisiones sobre como, cuando y donde la materia prima, producto intermedio y producto final deben ser adquiridos, transportados y almacenados. A título de ejemplo, algunos aspectos que hay que considerar en cada recurso compartido es la decisión de la operación a realizar en función no tan solo de la información local al recurso (por ejemplo, piezas en cola de espera, última operación realizada en el recurso, etc.) sino también como afectará al resto de recursos compartidos con los cual interactúa directa o indirectamente.

La elevada competitividad industrial, así como las exigencias en precios y calidad del producto, obliga a que las empresas deban reconfigurar continuamente su cartera de productos, sus métodos operativos, sus enfoques de mercado, sus procesos de acopio de materiales y distribución de los productos sometido a un conjunto de restricciones económicas que obligan a tomar decisiones bajo conceptos de producción ajustada (*lean manufacturing*). En este contexto, la logística pasa a considerarse como fuente de ventaja competitiva y aparece la necesidad de instrumentar herramientas informatizadas para la planificación de la cadena de suministro.

En los últimos años han aparecido en el mercado una gran diversidad de paquetes informáticos de ayuda a la toma de decisiones en la planificación y gestión de materiales, como son por ejemplo: MRP (*Material Requirements Planning*), ERP (*Enterprise Resource Planning*), DRP (*Distribution Requirements Planning*), SCM (*Supply Chain Management*). Todos estos paquetes intentan facilitar la toma de decisiones para dar respuesta a la demanda y procesamiento del producto final a lo largo del tiempo. Sin embargo, los modelos con los que trabajan no

contemplan la dificultad en el modelado de los recursos ni tampoco las interacciones entre ellos: características técnicas (orden de las operaciones), dependencia (un operador puede supervisar varias unidades pero solo puede atender a una), capacidad limitada (en función del tipo de operación). Por otro lado, las metodologías y entornos informáticos de ayuda a la toma de decisiones sobre la gestión y planificación de recursos (*Just-in-Time, Theory of Constraints, Schedulers y Planners*), suelen asumir en algunos casos que el material es un problema secundario, y en otros se limitan a mejorar el rendimiento de los recursos críticos (cuellos de botella), así como también, algunas de estas aproximaciones (*Schedulers y Planners*) asumen que la disponibilidad de los recursos responde a actividades deterministas.

Desafortunadamente, en los sistemas reales de producción, hay un elevado grado de dependencia entre las decisiones sobre recursos y sobre materiales. Este tipo de conflictos tan solo pueden ser tratados eficientemente si el modelo puede contemplar de modo simultáneo tanto los aspectos relacionados con el material, como los aspectos relacionados con los recursos a gestionar. Esta falta de herramientas y metodologías justifica que algunos autores definan el concepto de logística en el contexto industrial como un arte y una ciencia: "el arte y la ciencia de obtener, producir y distribuir material y productos, en el lugar, tiempo y cantidades adecuadas" [9]. La dificultad en el uso de sistemas informáticos que ayuden a la toma de decisiones estriba en la gran cantidad de variables que intervienen, las interacciones existentes entre ellas y los intereses conflictivos e incluso contrapuestos que muchas veces presentan.

La falta de herramientas analíticas que ayuden y faciliten la toma de decisiones es uno de los principales problemas con los que se encuentran los ingenieros y directivos que tienen la necesidad de mejorar el rendimiento de aquellos sistemas cuyo comportamiento depende de un número elevado de variables. Determinar cambios en los procedimientos, en los flujos de información, así como en la organización y en las políticas de funcionamiento tradicionales de la empresa son algunos ejemplos de tareas que deben ser realizadas para reaccionar ante perturbaciones y adaptarse a los constantes cambios del mercado o

del entorno.

Con el objetivo de abordar la complejidad que presenta la optimización de problemas logísticos caracterizados por la diversidad de relaciones causa-efecto y un número elevado de variables de decisión altamente acopladas, en [6] se presenta una propuesta metodológica para la resolución de problemas de optimización de sistemas logísticos, basada en la configuración automática de escenarios para analizar y evaluar mediante simulación como afectarían ciertos cambios al rendimiento del sistema, tanto en su organización como en las políticas de gestión de recursos compartidos, antes de que dichos cambios sean aplicados en el sistema real. Sin embargo, debido a que la evaluación automática puede llegar a consumir una cantidad incalculable de tiempo, en este artículo se presentan dos métodos para la reducción del tiempo de cálculo.

En la sección 2 de este artículo se introduce conceptualmente el formalismo de Redes de Petri Coloreadas como herramienta para el modelado de sistemas logísticos. En las secciones 3 y 4 se describen dos métodos que han sido desarrollados para la reducción del tiempo de cálculo en la exploración del espacio de estados generados a partir de modelos de Redes de Petri Coloreadas: Método de los Caminos Prohibidos y Método de Evaluación Jerárquica de Guardas. Finalmente, en la sección 5 se presentan las conclusiones de este artículo.

2. Redes de Petri Coloreadas y Espacio de Estados

El objetivo principal de la metodología propuesta en [6] consiste en automatizar la toma de decisiones mediante la resolución de problemas de optimización, para ello la metodología permite combinar la información almacenada en el modelo de simulación (identificación de alternativas) junto con los resultados obtenidos mediante diferentes estrategias de evaluación de las alternativas, permitiendo de esta forma cerrar un ciclo completo de un sistema de ayuda a la toma de decisiones: definición del problema, identificación de alternativas, análisis, y evaluación de alternativas, seguido de la prescripción de la mejor alternativa.

La idea central de la metodología es la generación del espacio de estados de sistemas

modelados mediante Redes de Petri Coloreadas, y explorar ese espacio de estados empleando métodos heurísticos que permitan hacer la búsqueda más eficiente a fin de encontrar una solución óptima o cuasi óptima a problemas logísticos, *scheduling*, *routing*, planificación, o cualquier problema complejo que pueda ser considerado NP-Completo.

Las Redes de Petri Coloreadas ó *Coloured Petri Nets*, en adelante RPC, han demostrado ser herramientas acertadas para el modelado de sistemas logísticos debido a las diversas ventajas tales como la capacidad de contener tanto la estructura estática como la dinámica del sistema, arquitectura del sistema de producción, y su naturaleza gráfica [3,10,11]. Además, las RPC son adecuadas para modelar y visualizar patrones de comportamiento que muestran concurrencia, sincronización y recursos compartidos, los cuales son factores claves cuando se trata de mejorar el rendimiento de sistemas logísticos o de manufactura.

Las RPC soportan un cierto nivel de abstracción en la etapa de modelado mediante el uso de colores, los cuales representan los atributos de las entidades que son soportadas por la mayor parte de paquetes de software de simulación comercial. Tanto las Redes de Petri (*Petri Nets*) [8] como las RPC [11] han sido usadas extensivamente para la evaluación del rendimiento de sistemas de producción, ya que el modelo contiene todos los eventos y sus interacciones, así como el tiempo consumido por cada actividad.

Las RPC proporcionan una herramienta de análisis que consiste en construir el espacio de estados, llamado también grafo de ocurrencia o grafo/árbol de alcance, de un sistema descrito mediante el formalismo de RPC. El primero de estos nombres refleja el hecho de que un espacio de estados contiene todas las secuencias posibles de acciones (disparos de transiciones) que pueden ocurrir en el sistema, mientras que los dos últimos nombres reflejan que el espacio de estados contiene todos los marcados alcanzables [3,4].

La idea básica subyacente de un árbol de alcance es la construcción de un grafo que contenga un nodo por cada estado (marcado) alcanzable en el sistema modelado, y un arco por cada disparo de una transición, lo cual permite representar las propiedades dinámicas de la RPC (comportamiento del sistema). El espacio de

estados permite describir formalmente un problema como un conjunto de transformaciones, desde una situación o estado inicial dado hasta unas situaciones o estados deseados (estados finales), a través de un conjunto de operaciones permitidas.

La metodología propuesta permite resolver problemas de planificación, *scheduling* o *routing* de sistemas de eventos discretos en particular, y a problemas de optimización en general, modelados mediante el formalismo de RPC [6]. El objetivo de la metodología consiste en la obtención de soluciones óptimas mediante la búsqueda en el espacio de estados generado a partir de modelos RPC temporales. El algoritmo para la exploración del espacio de estados está basado en el tipo de búsqueda exhaustiva conocida como primero en profundidad. En este método de búsqueda el árbol se examina hasta el nivel más profundo de una rama, donde un nodo no tenga ningún sucesor (nodo hoja), antes de que otro camino sea considerado. Cuando se alcanza un nodo hoja, la búsqueda retrocede al nivel anterior y explora cualquiera de las alternativas restantes en este nivel, y así sucesivamente hasta haber explorado todo el árbol. Este procedimiento sistemático de *backtracking* o retroceso garantiza que se examinarán exhaustivamente todas las posibilidades.

La utilización de esta estrategia de búsqueda es a su vez la base de un método de reducción del espacio de estados que ha sido desarrollado y que se ha denominado *método de los nodos repetidos* [6,7], el cual está basado en el tratamiento de aquellos nodos que se consideran repetidos, cuando difieren solamente en la información temporal (sellos de tiempo y reloj global) [3], durante la exploración del espacio de estados, contribuyendo a minimizar el problema de explosión de estados que surge al generar el espacio de estados de sistemas concurrentes. Este método de reducción presenta ventajas muy importantes en lo que respecta a la exploración del espacio de estados cuando se pretende optimizar sistemas logísticos, los cuales se caracterizan principalmente por un número muy elevado de actividades que se ejecutan en paralelo, presentan concurrencia y comparten recursos, propiciando que ciertos estados se repitan con frecuencia.

Sin embargo, aunque el método para el tratamiento de nodos repetidos alivia

considerablemente el problema de explosión de estados por medio de la reducción del número de nodos del árbol de alcance que representa el espacio de estados, el tiempo requerido para la exploración y análisis del espacio de estados suele ser prohibitivo cuando se trata de problemas logísticos o industriales reales. En las siguientes secciones se describen dos métodos que han sido desarrollados e implementados a fin de abordar este problema.

3. Método de los Caminos Prohibidos

El primero de los métodos para la reducción del tiempo de exploración del espacio de estados que se presenta en este trabajo está orientado más específicamente a la resolución de problemas de optimización. El método consiste en almacenar, en la estructura de datos que contiene cada nodo del espacio de estados, la información mínima necesaria para discriminar aquellos caminos que han sido explorados y que finalizan en un nodo hoja que no es un estado final, de tal manera que cuando se genera un nodo repetido que coincide (en el número de marcas del marcado y los valores de color de cada una de ellas) con un nodo previamente generado que se encuentra en cualquier nivel de este camino señalado, será rechazado inmediatamente y no será analizado innecesariamente, obviando la aplicación del *método de los nodos repetidos* descrito en [6,7]. La Figura 1 ilustra gráficamente el método.

Si se señalan los nodos hoja usando el carácter *h*, los nodos final utilizando el carácter *f*, y los nodos que no son nodos hoja ni nodos final con el carácter *n*, los nodos que no conducen a un nodo final, y por lo tanto no forman parte de una rama o camino que pueda considerarse solución factible del problema, se podrían señalar con el carácter *p*. De esta manera, un nodo repetido que coincide con un nodo señalado como *tipo p*, será un nodo prohibido y sus hijos no serán analizados.

Es de hacer notar que la exploración del espacio de estados que se ha implementado se realiza mediante la estrategia de búsqueda exhaustiva primero en profundidad, de esta manera, durante el proceso de expansión los nodos se señalan como *tipo n*, a menos que se trate de un nodo hoja, en cuyo caso se señala como *tipo h*, o de un nodo final (estado final), el cual se señala como *tipo f*. Al alcanzar un nodo

hoja o un nodo final se realiza un *backtracking* o retroceso al padre y se señala éste como un nodo prohibido, *tipo p*, siempre y cuando el retroceso se realice a partir de un nodo hoja.

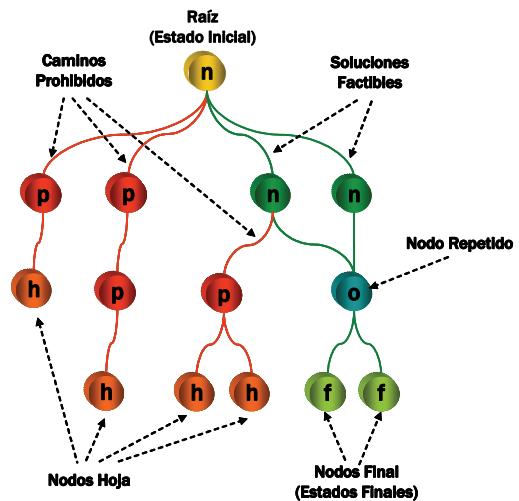


Figura 1. Método de los caminos prohibidos.

Tomando en cuenta que una vez que un nodo ha sido señalado como nodo prohibido no se analizará de nuevo durante el resto de la exploración del espacio de estados, es posible eliminar toda la información de tiempo asociada al nodo (reloj global y sellos de tiempo de las marcas), reduciendo así el uso de memoria sin que esto afecte el resultado de la búsqueda de soluciones.

La utilización éste método conjuntamente con el *método de los nodos repetidos* permite integrar los beneficios de cada uno de los tres tipos de métodos de reducción del espacio de estados: representación del espacio de estados en una forma condensada o compacta, representación de solamente un subconjunto del espacio de estados, y eliminación de estados o información del estado durante la exploración del espacio de estados.

4. Evaluación Jerárquica de Guardas

Teniendo en cuenta que la velocidad de respuesta de un simulador de RPC depende directamente de la eficiencia del algoritmo que evalúa las

transiciones [1,5], el segundo de los métodos que ha sido desarrollado para la reducción del tiempo de cálculo durante la generación del espacio de estados, está basado en la utilización de guardas en los modelos de RPC. El espacio de estados de un modelo RPC puede ser restringido por medio de la utilización de guardas que especifican las condiciones lógicas que se deben cumplir para que una transición pueda ser disparada, esto se debe al hecho de que el uso eficiente de guardas en el modelo puede disminuir considerablemente el número de transiciones habilitadas en los diferentes estados del sistema.

Sin embargo, el uso de una gran cantidad de guardas en el modelo puede ser contraproducente en lo que respecta al tiempo de cálculo requerido para la evaluación de las expresiones lógicas, teniendo en cuenta que todas las marcas de un marcado deben ser inspeccionados durante el proceso de evaluación, a fin de determinar la combinación de marcas de los diferentes nodos lugar que habilitan una determinada transición.

Un gran número de comparaciones podría ser evitado si cada guarda asociado a una transición se descompone en varios guardas, a fin de que puedan ser evaluados siguiendo un cierto orden jerárquico, de tal manera que solo aquellas marcas que satisfagan un guarda en un nivel serían utilizados para evaluar el guarda en el siguiente nivel de la jerarquía. Por ejemplo considérese el siguiente guarda genérico:

```
[expresión lógica 1] ^
[expresión lógica 2] ^
[expresión lógica 3] ^
[expresión lógica 4] ^ ... ^
[expresión lógica n]
```

el cual puede descomponerse en varios guardas organizados en diferentes niveles jerárquicos:

```
Nivel 1: [expresión lógica 1]
Nivel 2: [expresión lógica 2] ^
[expresión lógica 3]
Nivel 3: [expresión lógica 4]
.
.
.
Nivel m: [expresión lógica n]
```

De esta manera, sólo las marcas que satisfacen la expresión lógica del nivel 1 serán evaluadas en el segundo nivel, y de esas marcas, solamente aquellas que satisfacen la expresión lógica en el nivel 2 podrán ser evaluadas en el siguiente nivel, y así sucesivamente hasta evaluar todas las expresiones hasta el último nivel. Nótese que al descomponer el guarda en niveles se puede independizar la combinación de marcas de los diferentes nodos lugar. Sin embargo, este método, a diferencia de los anteriores, requiere que el modelador jerarquice los guardas de una forma favorable que permita reducir al máximo el número de evaluaciones y en consecuencia el tiempo de cálculo.

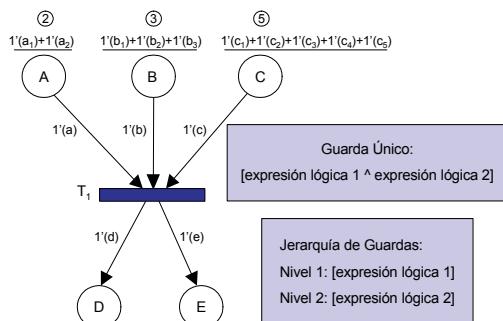


Figura 2. Guardas Jerárquicos.

Para ilustrar lo anterior considérese la RPC de la Figura 2, en la cual un único guarda requiere para su evaluación una marca de cada uno de los nodos lugar que tienen arcos de entrada a la transición T_1 , por lo tanto en número de combinaciones de marcas dependerá de la cantidad de marcas en cada uno de los nodos lugar A, B y C .

Al descomponer el guarda en varios guardas jerárquicos, sólo será necesario combinar las marcas de aquellos nodos lugar que se requieren para evaluar las expresiones lógicas en cada nivel. Así, si se descompone el guarda de la Figura 2 en dos guardas, de tal forma que en el primer nivel se evalúa la expresión lógica 1, la cual especifica, por ejemplo, una restricción sobre los nodos lugar A y C pero no sobre B , y en el segundo nivel se evalúa la expresión lógica 2, nótese que el número de veces a evaluar la expresión lógica 2 dependerá

del número de combinaciones de marcas de A y B para las cuales la expresión lógica 1 sea cierta, esto es, sólo para aquellas combinaciones de marcas que satisfacen la expresión lógica 1 se evaluará la expresión lógica 2, lo cual puede requerir marcas tanto del nodo lugar B como de cualquiera de los otros nodos lugares A y/o C, pero en cualquier caso el número de combinaciones de marcas posibles será menor que para el caso de un guarda único.

Obsérvese que para evaluar el guarda sin descomponer es necesario evaluar las expresiones lógicas 1 y 2 para todas las posibles combinaciones de marcas en un marcado, pero al descomponer el guarda en una jerarquía, solo en el caso de que todas las combinaciones de marcas satisfagan las expresiones lógicas en los diferentes niveles, se evaluarán todas las combinaciones posibles de marcas, y esto solo ocurriría en el peor de los casos. Por ejemplo, supóngase que en la RPC de la Figura 2 la cantidad de marcas en los nodos lugar A, B y C es la siguiente:

Nodo Lugar	Marcado	Cantidad de Marcas
A	$1'(a_1)+1'(a_2)$ 2	marcas
B	$1'(b_1)+1'(b_2)+1'(b_3)$ 3	marcas
C	$1'(c_1)+1'(c_2)+1'(c_3)+1'(c_4)+1'(c_5)$ 5 marcas	

Tabla 1. Cantidad de marcas en los nodos lugar A, B y C de la RPC de Figura 2.

Si se supone cada nodo lugar de entrada a la transición T_1 como un conjunto que contiene un determinado número de elementos (marcas), las posibles combinaciones compuestas por un elemento de cada nodo lugar de entrada a la transición se pueden expresar como el producto cartesiano de todos los pares ordenados donde primer elemento pertenece al nodo lugar A, el segundo elemento pertenece al nodo lugar B, y el tercer elemento pertenece al nodo lugar C, lo cual se puede representar matemáticamente como:

$$A \times B \times C = \{(a, b, c), a \in A, b \in B, c \in C\}$$

El cardinal o número de elementos del

producto cartesiano sería entonces el producto de los cardinales de cada uno de los conjuntos, esto es:

$$|A \times B \times C| = |A| \cdot |B| \cdot |C|$$

De la Tabla 1 se tiene que 2 marcas del nodo lugar A, por 3 marcas del nodo lugar B, por 5 marcas del nodo lugar C dan un total de 30 combinaciones posibles:

$$|A \times B \times C| = 2 \cdot 3 \cdot 5 = 30$$

Gráficamente se pueden representar las 30 combinaciones de marcas mediante un diagrama de árbol, tal como el que se muestra en la Figura 3. El conjunto de pares ordenados que representan todas esas combinaciones sería:

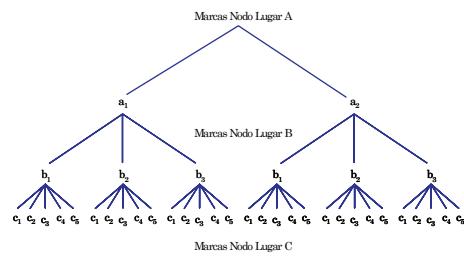


Figura 3. Diagrama de árbol de combinaciones de marcas de los nodos lugar A, B y C.

$$A \times B \times C = \{(a_1, b_1, c_1), (a_1, b_1, c_2), (a_1, b_1, c_3), (a_1, b_1, c_4), (a_1, b_1, c_5), (a_1, b_2, c_1), (a_1, b_2, c_2), (a_1, b_2, c_3), (a_1, b_2, c_4), (a_1, b_2, c_5), (a_1, b_3, c_1), (a_1, b_3, c_2), (a_1, b_3, c_3), (a_1, b_3, c_4), (a_1, b_3, c_5), (a_2, b_1, c_1), (a_2, b_1, c_2), (a_2, b_1, c_3), (a_2, b_1, c_4), (a_2, b_1, c_5), (a_2, b_2, c_1), (a_2, b_2, c_2), (a_2, b_2, c_3), (a_2, b_2, c_4), (a_2, b_2, c_5), (a_2, b_3, c_1), (a_2, b_3, c_2), (a_2, b_3, c_3), (a_2, b_3, c_4), (a_2, b_3, c_5)\}$$

Así, para cada marcado diferente que presente la RPC, el número de veces que debe ser evaluado el guarda único será igual al número de combinaciones de marcas posibles, tal que cada combinación contenga una marca por cada nodo lugar de entrada a la transición (A, B y C). En este ejemplo el guarda único deberá ser evaluado 30 veces, cada vez que cambie el marcado de la RPC.

De manera general, para n nodos lugar de

entrada a una transición, el número de combinaciones de marcas viene dado por el producto cartesiano:

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n), \\ a_1 \in A_1, \dots, a_n \in A_n\}$$

El cardinal o número de elementos del producto cartesiano es el producto de los cardinales de los conjuntos:

$$|A_1 \times A_2 \times \dots \times A_n| = |A_1| \cdot |A_2| \dots |A_n|$$

A partir de esta generalización se puede deducir que a mayor número de nodos lugares de entrada a una transición, mucho mayor será el número de combinaciones de marcas que deberán ser evaluadas por el guarda asociado a dicha transición. Y si además la RPC tiene varias transiciones, cada una con sus respectivos guardas asociados, el número de evaluaciones aumentará considerablemente.

Ahora supóngase que para el ejemplo de la Figura 2 el guarda del primer nivel que evalúa la expresión lógica 1 requiere marcas de los nodos lugar A y C , por lo que el conjunto de combinaciones a evaluar vendría dado por el producto cartesiano:

$$A \times C = \{(a, c), a \in A, c \in C\}$$

El cardinal o número de elementos del producto cartesiano sería:

$$|A \times C| = |A| \cdot |C|$$

De la Tabla 1 se tiene que 2 marcas del nodo lugar A , por 5 marcas del nodo lugar C dan un total de 10 combinaciones posibles:

$$|A \times C| = 2 \cdot 5 = 10$$

La Figura 4 representa gráficamente las 10 combinaciones de marcas mediante un diagrama de árbol. El conjunto de pares ordenados que representan esas combinaciones sería:

$$A \times B \times C = \{ (a_1, c1), (a_1, c2), (a_1, c3), (a_1, c4), \\ (a_1, c5), (a_2, c1), (a_2, c2), (a_2, c3), \\ (a_2, c4), (a_2, c5) \}$$

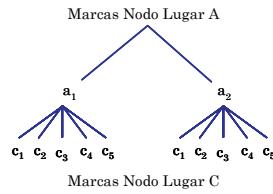


Figura 4. Diagrama de árbol de combinaciones de marcas de los nodos lugar A y C para evaluar el guarda del primer nivel de la jerarquía.

Si de las 10 combinaciones posibles de marcas sólo una, por ejemplo $(a_1, c4)$, satisface el guarda del primer nivel (expresión lógica 1), y suponiendo que el guarda del segundo nivel (expresión lógica 2) evalúa marcas de los nodos lugar A , B y C , entonces el número total de evaluaciones para los dos niveles de la jerarquía se reduce a 13, esto es, 10 combinaciones para evaluar el guarda del primer nivel y 3 combinaciones para evaluar el guarda del segundo nivel, tal como se muestra en el diagrama de la Figura 5.

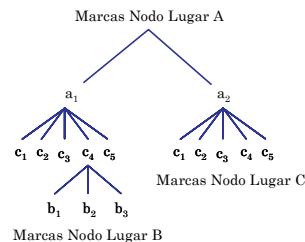


Figura 5. Diagrama de árbol de combinaciones de marcas de los nodos lugar A , B y C para evaluar los guardas del primer y segundo nivel de la jerarquía.

Así, de las 30 combinaciones de marcas que se requerirían para evaluar un guarda único, en este caso, si se evalúan los guardas como una jerarquía, solo sería necesario evaluar 13 combinaciones de marcas. La división jerárquica en niveles dependerá del diseñador de la RPC, y siempre tendrá como objetivo reducir lo más posible el número total de evaluaciones de todos los guardas asociados a las transiciones del modelo.

Este método para la evaluación jerárquica de

guardas ha sido implementado con éxito y ha permitido probar que el tiempo de exploración del espacio de estados construido a partir de un modelo RPC, con un número considerable de nodos lugar, marcas y nodos transición, se reduce notablemente.

En [6] se documenta detalladamente la aplicación de la metodología propuesta a varios casos de estudio, en los que se describen los beneficios en tiempo de cálculo que aportan los métodos aquí presentados.

5. Conclusión

En este artículo se han presentado dos métodos para la reducción del tiempo de cálculo en la exploración del espacio de estados: Método de los Caminos Prohibidos y Evaluación jerárquica de los guardas de las RPC. El primer método puede ser utilizado para señalar aquellos caminos del espacio de estados que no conducen a ninguna solución factible de un problema de optimización, evitando así examinarlos de nuevo cuando aparece un nodo repetido que coincide con el marcado de alguno de los nodos que se encuentran en el camino señalado. El segundo método es muy útil para evitar la evaluación innecesaria de combinaciones de marcas de los nodos lugar que no satisfacen las condiciones lógicas que los guardas representan, ya que permite minimizar el número de evaluaciones necesarias para descartar aquellas combinaciones de marcas de un marcado que no habilitan una transición.

Agradecimientos

Este trabajo ha sido realizado con el apoyo del Departamento de Universidades, Investigación y Sociedad de la Información del Gobierno de Cataluña. También ha sido posible gracias a la Dirección General de Investigación, Subdirección General de Proyectos a través del proyecto DPI2004-080056-C03-01.

Referencias

- [1] Gaeta, R. Efficient Discrete-Event Simulation of Colored Petri Nets. En: *IEEE Transactions on Software Engineering*, Vol. 22, Num. 9, 629-639, 1996.
- [2] Ghiani, G., G. Laporte y R. Musmanno. Introduction to Logistics Systems Planning and Control. Ed. JohnWiley & Sons, Ltd., 2004.
- [3] Jensen, K. Coloured Petri Nets: Basics Concepts, Analysis Methods and Practical Use, Vol. 1,2,3. Ed. Springer-Verlag, 1997.
- [4] Jensen, K. An Introduction to the Practical Use of Coloured Petri Nets. En: *Lectures on Petri Nets II: Applications, Lecture Notes in Computer Science*, Vol. 1492, 237-292, 1998.
- [5] Mortensen, K.H. Efficient Data-Structures and Algorithms for a Coloured Petri Nets Simulator. *Third Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, CPN'01*, 2001.
- [6] Narciso, M.E. Metodología para la Resolución de Problemas de Optimización Mediante la Exploración del Espacio de Estados Generado a partir de Modelos de Redes de Petri Coloreadas. Tesis Doctoral. Departamento de Telecomunicación e Ingeniería de Sistemas, Universidad Autónoma de Barcelona, 2007.
- [7] Narciso Farias, M.E., M.A. Piera i Eroles y J. Figueras. Optimización de Sistemas Logísticos Mediante Simulación: Una Metodología Basada en Redes de Petri Coloreadas. En: *Revista Iberoamericana de Automática e Informática Industrial, RIAI*, Vol. 2, Num. 4, 54-65, 2005.
- [8] Piera, M.A., M. Narciso, A. Guasch y D. Riera. Optimization of Logistic and Manufacturing Systems through Simulation: A Colored Petri Net-Based Methodology. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, Vol. 80, Num. 3, 121-130, 2004.
- [9] Piera, M.A., T. Guasch, J. Casanovas, J. J. Ramos. Como mejorar la logística de su empresa mediante la simulación. Ed. Diaz de Santos, 2006.
- [10] Silva, M. y R. Valette. Petri Nets and Flexible Manufacturing. En: *Lecture Notes in Computer Science, Advances in Petri Nets*, Vol. 424, 374-417, 1989.
- [11] Zimmermann, A., K. Dalkowski y G. Hommel. A Case Study In Modelling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets. En: Proceedings of the 8th European Simulation Symposium, ESS '96, 282-286, 1996.

APLICACIÓN DEL MODELADO Y LA SIMULACIÓN A PROCESOS DE BIOMEDICINA Y BIOLÓGICOS

Optimización y diseño de intervenciones adaptativas considerando gestión de riesgos

Ascensión Zafra-Cabeza, Miguel A. Rida, Eduardo F. Camacho

Dept. de Ingeniería de Sistemas y Automática

Universidad de Sevilla

41092 Sevilla

{asun,ridao,eduardo}@esi.us.es

Resumen

Este trabajo muestra como técnicas de ingeniería del control se pueden aplicar al área de la ciencia de la conducta o comportamiento humano a través de las intervenciones adaptativas. Las intervenciones adaptativas variantes en el tiempo (IAVT) constituyen un campo con muchas expectativas en la prevención y tratamiento de enfermedades crónicas y reincidentes, tales como el consumo de alcohol, drogas o depresión. Las aportaciones de este artículo se centran en el desarrollo de un algoritmo de control predictivo basado en modelo (CPBM) considerando riesgos para el diseño de IAVT. Se ha trabajado de forma explícita con los proyectos reales de prevención de conductas no deseables en adolescentes *Fast Track*, auspiciado por el *National Institute for Mental Health* de Estados Unidos y por otro lado, el programa *Communities that Care*, orientado a la identificación de riesgos para tratar el problema de la drogodependencia. La variable controlada es el nivel familiar (NF) de adolescentes con problemas. El algoritmo propuesto basado en CPBM decide la frecuencia apropiada de las visitas de un asesor familiar, tutorías u otras acciones apropiadas para mitigar los posibles riesgos que puedan variar el estado de los pacientes de forma no esperada. El CPBM es una estrategia de control óptima apta para esta aplicación por el tratamiento de las restricciones y problemas multivariados. Se muestran varias simulaciones bajo distintos escenarios de perturbaciones

para mostrar los beneficios del método.

1. Introducción

Las intervenciones adaptativas variantes en el tiempo (IAVT), también llamadas tratamientos dinámicos [7, 4], constituyen un campo con muchas expectativas en la prevención y tratamiento de enfermedades crónicas y reincidentes, tales como el consumo de alcohol, drogas o depresión. En las intervenciones adaptativas se proporcionan diferentes tratamientos, cuya selección y dosificación dependerá de la evolución y diagnóstico de cada paciente. Se dice que son variantes en el tiempo ya que en cada etapa de revisión se puede hacer variar la medicación o tratamiento, dependiendo del estado del sujeto en estudio.

Tras estudios realizados por psicólogos y trabajadores sociales en el área de tratamientos y diagnósticos a problemas como la adicción a drogas, problemas infantiles y otras enfermedades en las que el entorno social es bastante influyente, se ha concluido en la necesidad de considerar los factores externos que rodean al paciente para obtener resultados eficientes. Adicionalmente, el entorno conceptual establecido en [2] justifica ideas de control para aplicar en el diseño de intervenciones adaptativas. Han sido éstas las bases que se han considerado para emprender este trabajo, introduciendo un sistema de gestión de riesgos en el proceso de diseño y análisis de las intervenciones. Esta herramienta considera información adicio-

nal sobre otros factores (riesgos) que pueden influir en la evolución de los pacientes y pudieran darse durante el periodo de tratamiento.

Tras la identificación de riesgos, se necesitará un plan que gestione qué hacer en caso de que ocurran los riesgos que han sido previamente identificados. De esta forma, es necesario añadir más variables de decisión o control al problema de optimización para representar las correcciones o modificaciones al diagnóstico inicial. Estas modificaciones van a poder cambiar la frecuencia de un tratamiento e incluso cambiar o añadir otra intervención para satisfacer el nuevo escenario en el que los riesgos están involucrados y a los cuales los pacientes están expuestos. Así pues, se pretende obtener un sistema de diseño de intervenciones adaptativas robusto ante riesgos actualizado en cada revisión del paciente.

El marco de trabajo en el que se incluyen las aportaciones de este trabajo se engloban en [2, 8, 9], incorporando nuevos escenarios de pacientes. De forma específica se ha trabajado con el proyecto multidisciplinar de prevención *Fast Track* auspiciado por el *National Institute for Mental Health* de Estados Unidos y desarrollado por el *Conduct Problems Prevention Research Group (CPPRG)* [5]. El objetivo del programa *Fast Track* es la prevención de conductas anómalas y trastornos en el comportamiento de chicos con dificultades. Los componentes adaptativos que se han considerado para la prevención han sido las visitas de un asesor social a las familias que lo necesiten, así como tutorías para aquellos que mostrasen dificultades académicas. Por otro lado, se pueden encontrar trabajos dedicados a la identificación de factores riesgos en el comportamiento humano [6]. Entre ellos, se puede citar el *Social Development Research Group (SDRG)* de la Universidad de Washington [3] que ha trabajado en el programa *Communities that Care* para prevenir problemas de comportamiento en los adolescentes. En este artículo se usarán los riesgos identificados por el grupo SDRG para el caso de estudio.

La técnica de control seleccionada para llevar a cabo la prevención basada en riesgos, ha sido el control predictivo basado en modelo. Su

aplicación en este sistema ha resultado de gran relevancia al disponer del modelo del proceso y permitir la incorporación de restricciones.

A continuación se va a describir una IAVT, particularizada para el programa *Fast Track*, donde se incluye el modelo dinámico de este proceso. En la sección 3, se muestra la formulación del sistema de gestión de riesgos para las intervenciones adaptativas. El problema de optimización basado en CPBM se describe en la sección 4. Finalmente, se exponen los resultados que se han obtenido aplicando esta propuesta y basado en datos reales.

2. Descripción del sistema

En la sección que ahora se abarca, se describe el modelo dinámico considerado de las intervenciones adaptativas dentro del programa *Fast Track*. El objetivo fue diseñar intervenciones adaptativas para evitar comportamientos no deseados, tomando como acción preventiva el asesoramiento periódico a padres por parte de trabajadores sociales, así como de tutorías a los alumnos.

La intensidad de las intervenciones se determina por los resultados obtenidos de los cuestionarios llenados por los padres en cada visita del trabajador social, así como las percepciones del personal clínico. Basado en estas medidas, la estimación del nivel familiar (NF) se puede establecer como muy bajo, bajo, no suficiente y adecuado; la frecuencia de la intervención será semanal si el NF es muy bajo, bisemanal si es bajo, mensual si es no suficiente y en el caso de poseer un NF adecuado no habrá intervención alguna hasta la próxima revisión. La frecuencia de las revisiones es de cada 3 meses y el periodo completo de tratamiento es de 3 años, por lo que se darán 12 revisiones a cada uno de las familias por parte del personal clínico. El objetivo de interés final será medir el nivel de trastorno en los niños, un año después de que el tratamiento completo haya terminado.

Este sistema ha sido previamente estudiado, resaltando los esfuerzos aplicados sobre este proceso en materias de control. Nótese que el sistema que se está estudiando puede mode-

larse como un sistema de control básico en el que se quiere obtener un resultado específico (estado favorable del paciente) bajo la determinación de una frecuencia de visitas de trabajadores sociales (señal de control) para el asesoramiento y seguimiento del estado del paciente.

2.1. Modelo dinámico de la IAVT

En la referencia [8] se expone una descripción del sistema de control básico con realimentación para las intervenciones adaptativas. Este sistema puede ser modelado como un sistema dinámico. El modelo inicial de las IAVT para el programa *Fast Track* ha sido el siguiente:

$$\begin{aligned} NF(t+1) &= NF(t) + K_I I(t-\theta) + D(t) \\ D(t) &= \sum_{i=1}^m D_i(t) \end{aligned} \quad (1)$$

donde

$I(t)$ es la variable manipulada determinando la frecuencia de las visitas de asesoramiento a la familia en el instante de tiempo t .

$NF(t)$ es la variable que se desea controlar y representa el nivel familiar en el instante de tiempo t .

K_I es la ganancia de la intervención $I(t)$.

θ es el tiempo de retraso desde que se aplica la intervención hasta que tiene efecto en la salida.

$D(t)$ es el conjunto de perturbaciones en el instante t , considerada como un conjunto de i elementos que pueden afectar a la salida y modeladas como incertidumbres aditivas.

Si el sistema fuese ideal, el término $D(t)$ sería nulo. Sin embargo, en esta clase de sistemas los factores externos pueden influir en gran medida en el sistema. La siguiente sección describe como se introduce la gestión de riesgos en el modelo, adelantando que será este término $D(t)$, el que contenga toda la información sobre riesgos en cada instante de tiempo.

3. Modelado de riesgos para las IAVT

El diseño de las IAVT es un proceso que tiene asociado un alto grado de subjetividad debido al procedimiento de medida de las variables y hechos determinantes como que un mismo tratamiento aplicado a dos individuos distintos, obtiene diferentes resultados.

Los factores de riesgo externo e interno se pueden considerar como perturbaciones del sistema, pudiendo hacer variar la respuesta esperada de la intervención. Estas perturbaciones necesitan ser gestionadas de forma eficiente para reducir los efectos de los eventos inesperados. Las perturbaciones se pueden considerar como beneficiosas para el paciente (logro de un nuevo trabajo, periodo de vacaciones,...) así como desfavorables (pérdida de trabajo, enfermedad, defunción de un familiar, relación de amigos,...).

En la propuesta que aquí se detalla, cada una de las perturbaciones va a ser considerada como un posible riesgo que puede alterar la salida esperada de forma negativa o positiva. De esta forma, se propone un sistema de gestión de riesgos a integrar en el proceso de diseño de las intervenciones adaptativas para obtener resultados más robustos en los pacientes.

En este trabajo, el término riesgo se define como un evento que puede darse en un paciente causando impactos. De esta forma, sea $R = \{R_1, \dots, R_m\}$, el conjunto de riesgos identificados en un paciente. Cada riesgo se caracteriza por tener estimada una probabilidad que va a depender del tiempo, $P_i(t)$, así como un conjunto de impactos (II_i) que van a poder modificar la eficiencia de la intervención planificada y consecuentemente el estado final del paciente. Los impactos iniciales se pueden expresar en términos de un decremento/incremento relativo a la eficiencia de la intervención o señal de control I , o también como un cantidad absoluta afectando al NF , si fuese conocida.

Una vez que se han identificado los riesgos que deberían considerarse para diseñar la intervención adaptativa, se deberá diseñar un plan estratégico desde el punto de vista clínico

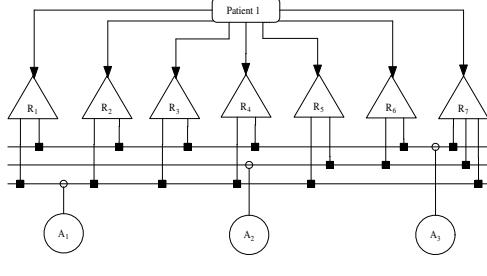


Figura 1: ADR para las IAVT.

para mitigar los riesgos atendiendo a sus probabilidades e impactos. La mitigación de los riesgos se solventa a través del análisis de las acciones de mitigación. La figura 1 ilustra un ejemplo de dependencia entre riesgos y acciones que puede presentar un paciente.

Como puede observarse en el árbol de datos de riesgos (ADR) de un paciente, varios riesgos pueden afectar a un paciente y cada uno de ellos puede mitigarse por varias acciones, si así fuese necesario.

Supóngase conocido el conjunto de acciones de mitigación $A = \{A_1, \dots, A_p\}$ que constituyen el ADR, siendo p el número de acciones. Formalmente, cada acción de mitigación A_i se describe por los elementos:

$$A_i = \{u_i, f_i, g_i\} \quad i = 1 \dots p$$

La variable de decisión para la acción A_i se denota por u_i ; $f_i : \mathbb{R} \rightarrow \mathbb{R}$ es la función que determina la reducción del impacto del riesgo en función de u en cada unidad de tiempo cuando la acción A_i se ejecuta.

Las intervenciones que se diagnostican sobre los pacientes pueden tener asociadas un coste económico de ejecución a cambio de reducir el efecto de los riesgos. Esta propiedad queda modelada por las funciones $g_i(u_i) : \mathbb{R} \rightarrow \mathbb{R}$.

Sea $u = [I \ u']$ el vector de las variables de decisión incluyendo por una parte, la variable de intervención inicial I , cuyo valor inicial puede verse alterado al considerar riesgos y por lo que se englobará dentro del conjunto de las acciones de mitigación. $u' = [u'_1, \dots, u'_{p-1}]$ es el vector de acciones adicionales para llevar a

cabo la mitigación.

El término $D(t)$ que representa las perturbaciones en el modelo del nivel familiar presentado en la ecuación (1), se puede formalizar como sigue:

$$D(u, t) = \sum_{i=1}^m D_i(u, t) = \sum_{i=1}^m PR(a, i) RE_i(u, t) \quad (2)$$

La matriz PR representa la relación entre los pacientes y los riesgos que presentan. De esta forma, el elemento $PR(a, i)$ posee un valor 1 si el paciente a puede estar sujeto a los efectos del riesgo R_i . Los términos $RE_i(u, t)$ modelan el efecto del riesgo R_i en el instante de tiempo t y se denominan exposición del riesgo. La exposición del riesgo en las intervenciones adaptativas toma la forma:

$$RE_i(u, t) = P_i(t)(II_i(u) - \sum_{j=1}^p f_j(u_j)) \quad (3)$$

siendo $P_i(t)$ la probabilidad del riesgo R_i en el instante t e II_i su impacto sobre la variable de salida. El coste total de las acciones de mitigación viene determinado por la expresión:

$$\text{Coste} = \sum_{j=1}^p g_j(u_j)$$

4. Formulación del problema de optimización usando control predictivo basado en modelo

El CPBM es una estrategia de control óptima basada en el uso explícito de un modelo dinámico para predecir la salidas del proceso en tiempos futuros [1]. Al intervalo de tiempo para el cual se calcula la predicción de las salidas del proceso, se le denomina horizonte de predicción. El conjunto de señales de control futuras se calcula en base a una función objetivo, usualmente cuadrático. Las salidas futuras que se obtienen usando el modelo de predicción dependen de los valores de las salidas y entradas pasadas hasta el instante t y de las

señales de control futuras. Solo se usa la señal de control calculada para el instante t , siendo el resto descartada. Algunas de las ventajas que presenta esta técnica frente a otras, es el manejo de las restricciones, la extensión al caso multivariable y su implementación.

En este artículo, la función objetivo del CPBM se trata de un funcional ponderado por tres variables β_1 , β_2 y β_3 , para los tres términos que se diferencian en la expresión siguiente:

$$\begin{aligned} J_N = & \beta_1 \sum_{j=1}^N \delta(j)[\hat{NF}(t+j|t) - NF^r(t+j)]^2 \\ & + \beta_2 \sum_{j=1}^N \lambda(j)[\Delta u(t+j-1)]^2 \\ & + \beta_3 \sum_{j=1}^p g_j(u) \end{aligned} \quad (4)$$

siendo N el horizonte de control y predicción, \hat{NF} la estimación de la salida en el instante $t+j$ realizado en el instante t , $NF^r(t+j)$ la referencia a seguir en el instante $t+j$ y $\Delta u(t+j-1)$ el incremento de la señal de control. El esfuerzo de control es importante en este problema, ya que se puede intentar aumentar o disminuir la frecuencia de las intervenciones de forma paulatina.

Las ecuaciones (1), (2) y (3) vuelven a ser consideradas para obtener el modelo de entrada/salida del sistema, tratándose de un modelo de 1 salida y p entradas.

$$A(z^{-1})NF(t) = B(z^{-1})u(t-1)$$

donde $A(z^{-1}) = 1 - (z^{-1})$ y $B(z^{-1})$ es un vector de dimensión $p \times 1$. $B(z^{-1})$ es un vector variante en el tiempo y va a ser actualizado en cada instante de tiempo t . El modelo del proceso cambia por tanto en cada unidad de tiempo como consecuencia del cambio de estado del paciente, y los riesgos que para ese periodo de tiempo hayan tenido lugar, de acuerdo a sus probabilidades.

El periodo considerado en el programa *Fast Track* ha sido de 36 meses y la frecuencia de

Cuadro 1: Descripción de riesgos identificados en el programa *Fast Track*.

R_i	Descripción	P_i	II_i
R_1	Accesibilidad fácil a drogas en zona residencial o escuelas.	0.5	8
R_2	Problemas de drogodependencia en la familia.	0.01	3
R_3	Amigos con problemas de drogodependencia.	0.2	2.7
R_4	Comunidad de escasos recursos en la que habita.	0	3
R_5	Dedicación de tiempo escaso de padres a los hijos.	0.9	4.7
R_6	Resultados académicos insatisfactorios.	0.2	4.3
R_7	Comportamientos violentos.	0.17	2.8

revisión por parte del personal clínico será de 3 meses, por lo que se esperan 12 intervenciones a lo largo de todo el periodo de estudio. El periodo de muestreo se ha establecido en 1 mes. Las acciones se realizan al principio de cada periodo de revisión.

5. Un ejemplo ilustrativo

El ejemplo que se ha adoptado para mostrar el método propuesto, está basado en el programa *Fast Track*. Los datos considerados provienen de los estudios realizados en el programa *Communities that Care*, donde se identificaron un conjunto de riesgos que podían presentarse en los pacientes, así como un conjunto de tareas preventivas que fueron consideradas como eficientes.

El conjunto de riesgos que ha sido identificado en un determinado paciente es $R = \{R_1, \dots, R_7\}$. La tabla 1 describe cada uno de estos riesgos, su probabilidad, así como los impactos expresados según el formato $II_i = K_c$.

El siguiente paso es determinar las acciones que pueden mitigar cada uno de los riesgos identificados en la tabla 1. La tabla 2 muestra el conjunto de las acciones de mitigación considerado $A = \{A_1, A_2, A_3\}$. El árbol de da-

Cuadro 2: Descripción de las acciones mitigadoras del programa *Fast Track*.

A_i	Descripción	f_i
A_1	Visitas de asesoramiento	$f_1(I) = 4,75I$
A_2	Sesiones de tutorías	$f_2(u'_2) = 3,2u'_2$
A_3	Tareas extraescolares (cursos de formación, deportes, excursiones,...).	$f_3(u'_3) = 4,1u'_3$

tos de riesgos para el ejemplo considerado se muestra en la figura 1 donde se relacionan los riesgos y acciones identificados para la familia en estudio.

La acción A_1 que actúa sobre la variable de decisión I , se ha considerado como una acción de mitigación, ya que su valor puede verse modificado en el caso de que fuese necesario si los riesgos se produjeseen. Existen tres variables de control, $u = [I, u'_2, u'_3]$, de las cuales las variables I y u'_2 son naturales y u'_3 es booleana. Para las dos primeras acciones se ha de determinar la frecuencia de las visitas o tutorías. Para la tercera acción, será una decisión de realizar o no la acción. De esta forma $I, u'_2 \in \{0, 1, 2, 3\}$ y $u'_3 \in \{0, 1\}$ son las restricciones que han de añadirse al problema de optimización mixto. El valor 0 significa que no hay intervención y los valores 1,2,3 determinan que las revisiones sean mensuales, bisemanales y semanales, respectivamente.

El modelo del nivel familiar ha sido descrito anteriormente en la ecuación (1). En esta expresión, la variable que representan el nivel familiar, $NF(t)$, se considera normalizada con valores entre 0 y 100. Para establecer el nivel familiar de cada paciente, se establecen los rangos $NF^{Muy\ bajo} = [0, 16,7]$, $NF^{Bajo} = (16,7, 33]$, $NF^{Nosuficiente} = (33, 50]$ y $NF^{Adecuado} = (50, 100]$.

El nivel familiar que se quiere conseguir es $NF^r = 50$ e inicialmente se supone que la familia posee un valor 0. La ganancia de la intervención es $K_I = 1,665$ y el retraso de la intervención I se ha adoptado como $\theta = 0$.

El sistema considerado es un sistema lineal

de primer orden sin retraso y por consiguiente $B(z^{-1}) = B_0(t)$. Note que $B_0(t)$ es una matriz (3×1) y $d(t)$ es un vector 1×1 . A continuación se muestran las expresiones de estas matrices.

$$B_0(t)(1, 1) = (P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) + P_7(t))f_1$$

$$B_0(t)(2, 1) = (P_5(t) + P_6(t) + P_7(t))f_2$$

$$B_0(t)(3, 1) = (P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_6(t) + P_7(t))f_3$$

$$d(t) = \sum_{i=1}^7 P_i(t)II_i$$

La función objetivo se toma de (4) con $\beta_1 = 1$, $\beta_2 = 1$ y $\beta_3 = 0$ debido a que las variables δ y λ ya son usadas para ponderar el error y el esfuerzo de control, respectivamente; el coste de las acciones de mitigación, no se usa.

5.1. Resultados obtenidos

El primer experimento abordado se muestra en la figura 2. La gráfica superior representa la evolución de la salida del sistema (estado del paciente) en función del tiempo (línea gruesa magenta) y la referencia establecida (línea continua roja). La línea discontinua azul representa la evolución del paciente en el caso de no considerar riesgos (o las probabilidades 0). La línea punteada verde representa el caso de considerar riesgos, pero no llevar a cabo ninguna acción de mitigación, poniéndose de manifiesto el resultado negativo que se obtendría. Las gráficas inferiores representan las señales de control o acciones de mitigación en el caso de aplicar o no mitigación, con línea gruesa magenta y azul discontinua, respectivamente. Nótese que las acciones A_2 y A_3 no se ejecutan en este último caso. Los valores de sintonización iniciales que se han tomado del controlador son $N = 1$, $\delta = 0,01$ y $\lambda = 0,3$.

Llevando a cabo la sintonización del controlador predictivo con un horizonte mayor ($N = 5$) y una ponderación mayor en la salida ($\delta = 0,5$), se obtienen resultados significativamente mejores. Las acciones de control o

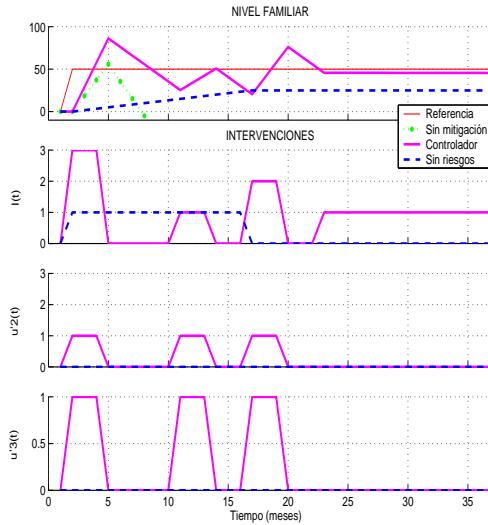


Figura 2: Resultados obtenidos en el nivel familiar y acciones de mitigación.

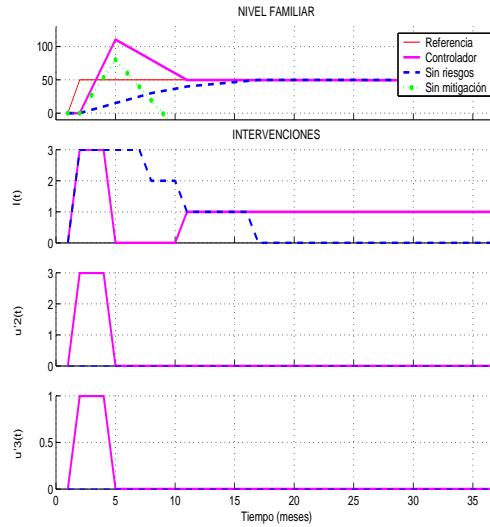


Figura 3: Resultados obtenidos sintonizando el controlador con un horizonte mayor ($N = 5$).

intervenciones se observan más constantes que en el experimento anterior, pudiendo resultar más coherente para su aplicación. Estos resultados se pueden observar en la figura 3.

El próximo módulo experimental considera un valor de retraso en las señales de control o intervenciones distinto de cero ($\theta \neq 0$). En una primera aproximación, se ha cambiado de $\theta = 0$, a $\theta = 1$, siendo ahora el retraso de un periodo de revisión (tres meses). Los resultados se ilustran en la figura 4.

La última simulación que se ha acometido es la obtención de resultados para el caso de tener un retraso de un periodo de revisión en la intervención I , pero considerando que el periodo de revisión por parte del personal clínico es de 1 mes; esto es, las señales de control no se mantienen durante 3 meses, sino que en cada mes se puede variar el diagnóstico. La figura 5 describe la salida obtenida y las acciones a realizar para este caso. El seguimiento de la referencia es mejor que en el anterior, debido a que las señales de control, pueden variar con mayor frecuencia.

6. Conclusiones

Este trabajo describe como técnicas de control avanzado se aplican al diseño de procesos clínicos como son las intervenciones adaptativas. Más allá de la introducción de nociones de control en este sistema, se ha integrado un sistema de gestión de riesgos en el proceso de diseño, considerando las necesidades identificadas por los psicólogos y el personal médico.

Los datos que se han considerados sobre riesgos fueron extraídos de programas consolidados por diferentes entidades, en las que detectaron la relación existente entre los factores de riesgo y el comportamiento humano.

Los resultados obtenidos ponen de manifiesto el papel de los parámetros de diseño involucrados en el control.

Referencias

- [1] E. F. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Springer-Verlag, 2004.

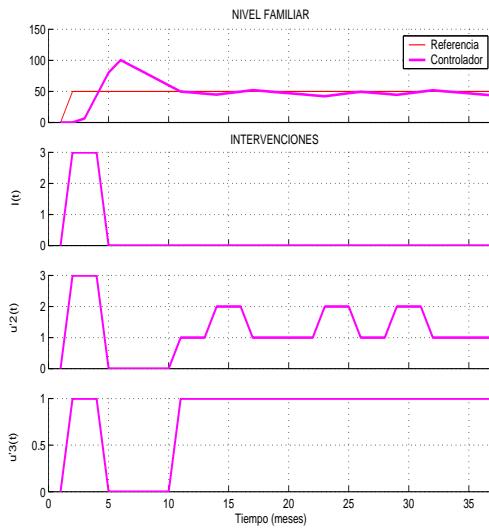


Figura 4: Resultados obtenidos considerando retrasos de un periodo de revisión en la intervención de la asesoría familiar.

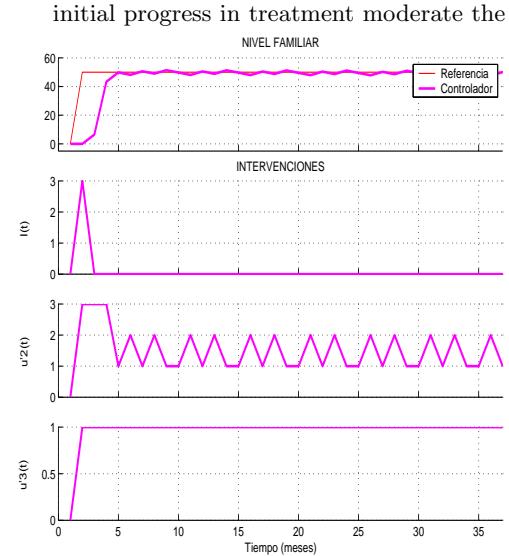


Figura 5: Resultados obtenidos con retraso de tres meses en la intervención y periodo de revisión clínica de un mes.

- [2] L. Collins, S. Murphy, and K. Bierman, “A conceptual framework for adaptive preventive interventions,” *Prevention Science*, vol. 5, no. 3, pp. 185–196, 2004.
- [3] J. Hawkins, “Preventing crime and violence through communities that care,” *European Journal on Criminal Policy and Research*, vol. 7, pp. 443–458, 1999.
- [4] P. Lavori and R. Dawson, “Dynamic treatment regimes: practical design considerations,” *Clinical Trials*, vol. 1, pp. 9–20, 2003.
- [5] Conducts Problem Prevention Research Group, “Initial impact of the fast track prevention trial for conduct problems: I. the high-risk sample,” *Journal of Consulting and Clinical Psychology*, vol. 67, pp. 631–647, 1999.
- [6] J. R. McKay, K. G. Lynch, D. S. Shepard, J. Morgenstern, R. F. Forman, and H. M. Pettinati, “Do patient characteristics and initial progress in treatment moderate the effectiveness of telephone-based continuing care for substance use disorders?” *Addiction*, vol. 100, no. 2, pp. 216–226, 2005.
- [7] S. Murphy and J. McKay, “Adaptive treatment strategies: An emerging approach for improving treatment effectiveness,” *Clinical Science*, 2004, newsletter of the American Psychological Association Division 12, section III: The society for the science of clinical psychology.
- [8] D. Rivera, M. Pew, L. Collins, and S. Murphy, “Engineering control approaches for the design and analysis of adaptive, time-varying interventions,” The Methodology Center Technical Report 05-73, Pennsylvania State University, 2005.
- [9] A. Zafra-Cabeza, D. E. Rivera, L. M. Collins, M. Rida, and E. F. Camacho, “A risk-based model predictive control approach to adaptive interventions in behavioral health,” in *Proceedings of the International Conference on Decision and Control. San Diego.*, 2006.

Predicción de glucemia en pacientes con diabetes tipo 1 ante incertidumbre*

J. Bondia

Instituto de Automática e Informática Industrial
Universidad Politécnica de Valencia
jbondia@isa.upv.es

R. Calm, M. García-Jaramillo, J. Vehí

Institut d'Informàtica i Aplicacions
Universidad de Girona
remei.calm@udg.es,magarcia@eia.udg.es,josep.veh@udg.es

C. Tarín

Inst. de Telecomunicaciones y Apl. Multimedia
Universidad Politécnica de Valencia
critasa@iteam.upv.es

W. García-Gabín

Institut d'Informàtica i Aplicacions
Universidad de Girona
wagarcia@eia.udg.es

13 de junio de 2007

Resumen

En este trabajo se aborda la predicción del nivel de glucosa (glucemia) en pacientes con diabetes tipo 1 ante incertidumbre en los parámetros, entrada y condiciones iniciales. Mediante técnicas de sistemas cooperativos y aritmética intervalar modal se obtiene una envoltura de las posibles evoluciones del nivel de glucosa que puede sufrir el paciente tras un itinerario de inyecciones de insulina y comidas. Ello permitirá, tras una integración en un sistema de ayuda a la dosificación, predecir posibles episodios de hiperglucemia e hipoglucemia y su riesgo, consiguiendo un mejor control metabólico en estos pacientes.

1. Introducción

La diabetes mellitus tipo 1 es una enfermedad crónica que se caracteriza por la destrucción de las células en el páncreas responsables de la secreción de insulina (β -células). La insulina es una hormona clave en el metabolismo de

la glucosa, favoreciendo su paso a través de la membrana celular y su almacenamiento en el hígado y los músculos principalmente. La deficiencia de la producción de insulina por parte del páncreas se traduce en una elevación anormal de la concentración de glucosa en sangre, característica de la diabetes [9]. Esto provoca importantes complicaciones a largo plazo como ceguera, insuficiencia renal, infarto agudo de miocardio, etc., así como una disminución muy significativa de la esperanza de vida.

El objetivo de los tratamientos de la diabetes tipo 1 es imitar cuanto sea posible el funcionamiento de un páncreas sano. Hoy en día se practica la terapia intensiva, basada en la inyección de dos tipos de insulina: insulina basal, una vez al día, para mantener un nivel constante de insulina en sangre; e insulina bolus, en cada comida, para compensar los efectos de ésta. El grave inconveniente que presenta este tratamiento es la dificultad que supone decidir la dosis de insulina a inyectar en función de la medición del nivel actual glucosa, la comida y el ejercicio físico.

Se considera que el desarrollo de sistemas autónomos de ayuda a la dosificación de insulina permitirá una mejora significativa de la calidad de vida de los pacientes con diabetes

*Este trabajo ha sido apoyado por el gobierno español a través de los proyectos DPI-2004-07167-C02-01 y DPI-2004-07167-C02-02, y por la unión europea a través de los fondos FEDER.

tipo 1. En este sentido, este trabajo aborda la predicción de la concentración de glucosa en sangre ante diversos itinerarios de insulina, comidas y niveles iniciales de glucosa. A partir de una predicción adecuada de la glucemia será posible alertar al paciente sobre posibles episodios de hiperglucemias o hipoglucemias, proporcionando la información necesaria para que el paciente tome decisiones acerca de la terapia. En dicha predicción será necesario tener en cuenta la variabilidad intra-paciente, la dificultad de determinar la cantidad de comida ingerida y su composición nutricional, y la incertidumbre en el estado inicial de la predicción, ya que sólo se dispondrá de información acerca del nivel actual de glucosa en sangre, pero no de otras variables de estado como concentración de insulina, concentración de glucosa en espacio intercelular, etc. Se considerará por tanto incertidumbre en los parámetros, entrada y condiciones iniciales. La predicción ante incertidumbre se tratará mediante la adecuada combinación de técnicas de sistemas cooperativos y análisis intervalar para una simulación eficiente. Se empleará el análisis intervalar modal, frente al análisis intervalar clásico, ya que permite evitar el problema de sobreestimación inherente a este último.

2. Modelado del sistema glucorregulatorio en diabetes tipo 1

La predicción de glucemia ante una terapia de insulina y secuencia de ingestas debe contemplar la absorción subcutánea de insulina, su cinética en sangre y sus efectos, la absorción y digestión de carbohidratos, y el metabolismo de la glucosa. La relación entre estos procesos se muestra en la figura 1.

Tras una inyección subcutánea, la insulina inyectada se difundirá a través del tejido adiposo y, tras sufrir una serie de transformaciones moleculares, se absorberá hacia el sistema circulatorio (flujo de insulina exógena). Una vez en sangre, alcanzará su lugar de acción, influyendo en la glucemia, y será degradada y eliminada (farmacocinética y farmacodinámica). Por otra parte, tras una ingesta, los carbohidratos ingeridos se convertirán principal-

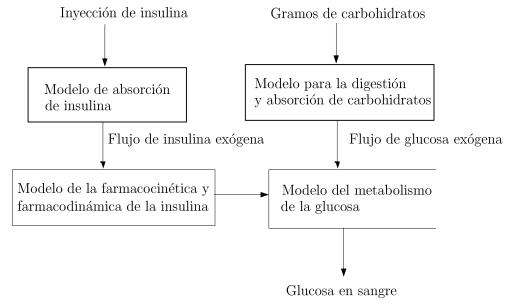


Figura 1: Descripción del modelo glucorregulatorio.

mente en glucosa en el proceso de digestión. Ésta se absorberá en el intestino hacia el sistema circulatorio (flujo de glucosa exógena). La subida de concentración de glucosa en sangre que esto produce será “compensada” por la insulina, estimulando el paso al interior celular y su almacenamiento en el hígado y músculos principalmente. A continuación se describe el modelo considerado para cada uno de estos subsistemas.

2.1. Modelo de absorción subcutánea de insulina

En este artículo se ha utilizado el modelo presentado en [7] para describir la absorción subcutánea de insulina. La elección de este modelo se debe a que cubre el abanico de insulinas más comunes, tanto basales (Protaphane[®], Lantus[®]), como bolus (Actrapid[®], Novorapid[®], Humalog[®]). El modelo considera que la insulina puede encontrarse en tres estados moleculares: “cristalizada”, forma hexámera y forma dímera (agrupaciones de seis y dos moléculas, respectivamente). Debido al tamaño molecular sólo se considera absorción de esta última. Además, se modela el fenómeno de difusión. El modelo corresponde por tanto a un sistema de ecuaciones en derivadas parciales:

$$\frac{\partial c_d(t, r)}{\partial t} = P(c_h(t, r) - Qc_d(t, r)^3) - B_d c_d(t, r) + D \nabla^2 c_d(t, r) \quad (1)$$

$$\begin{aligned}\frac{\partial c_h(t, r)}{\partial t} &= -P(c_h(t, r) - Qc_d(t, r)^3) \\ &\quad + k_{cb}(t, r)(c_{h,max} - c_h(t, r)) \\ &\quad + D\nabla^2 c_h(t, r)\end{aligned}\quad (2)$$

$$\begin{aligned}\frac{\partial c_b(t, r)}{\partial t} &= -k_{cb}(t, r)(c_{h,max} - c_h(t, r)) \\ &\quad + d_b D \nabla^2 c_b(t, r)\end{aligned}\quad (3)$$

$$I_{ex}(t) = B_d \int_{V_{sc}} c_d(t, r) dV. \quad (4)$$

donde c_b representa la concentración de insulina “cristalizada”, c_h la de insulina hexámera, c_d la dímera, r es la distancia al lugar de punción, V_{sc} el volumen de difusión y ∇^2 es el operador Laplaciano. La adecuada parametrización del modelo permite tratar los diferentes tipos de insulina.

El flujo de insulina exógena total, $I_{ex}(t)$, será la suma de los flujos bolus y basal.

2.2. Modelo de absorción y digestión de carbohidratos

Este modelo describe la descomposición de los carbohidratos en monosacáridos (principalmente glucosa) que tiene lugar durante la digestión, así como la absorción intestinal. El modelo considerado en este artículo se describe en [3], donde la tasa de absorción de glucosa $U_G(t)$ viene representada por

$$U_G(t) = \frac{D_G A_G t \exp(-t/t_{\max,G})}{t_{\max,G}^2} \quad (5)$$

siendo D_G la cantidad de carbohidratos ingeridos, A_G la biodisponibilidad (porcentaje que llega a sangre) y $t_{\max,G}$ el tiempo máximo de permanencia de la glucosa en sangre. Se puede observar que la ecuación (5) corresponde a la respuesta impulsional de un sistema lineal de segundo orden con dos polos reales múltiples en $-1/t_{\max,G}$ y ganancia A_G .

2.3. Modelo de la farmacocinética y la farmacodinámica de la insulina

La concentración de insulina en sangre, $I(t)$, se describe como

$$\frac{dI(t)}{dt} = \frac{I_{ex}(t)}{V_I} - k_e I(t) \quad (6)$$

donde $I_{ex}(t)$ es la absorción de insulina exógena descrita anteriormente (suma de la basal y la bolus), k_e es la tasa de eliminación de insulina en sangre y V_I es el volumen de distribución de insulina.

La concentración de la insulina en sangre tiene influencia en el transporte de la glucosa desde la sangre a los tejidos, en la producción de glucosa por parte del hígado (producción endógena) y en el consumo de glucosa por parte de los músculos y grasa principalmente (consumo periférico) [3]. Estas tres acciones se modelan mediante ecuaciones de primer orden:

$$\begin{aligned}\frac{dx_1}{dt} &= -k_{a1}x_1(t) + k_{b1}I(t) \\ \frac{dx_2}{dt} &= -k_{a2}x_2(t) + k_{b2}I(t) \\ \frac{dx_3}{dt} &= -k_{a3}x_3(t) + k_{b3}I(t)\end{aligned}\quad (7)$$

donde x_1 representa los efectos de la insulina en la distribución/trasporte de glucosa, x_2 representa el efecto sobre el consumo de glucosa y x_3 el efecto sobre la producción de glucosa endógena; k_{ai} , $i = 1, \dots, 3$ representan las tasas de desactivación y k_{bi} , $i = 1, \dots, 3$ las tasas de activación. Puede considerarse una parametrización alternativa tomando $\frac{k_{b1}}{k_{a1}} = S_{IT}$, $\frac{k_{b2}}{k_{a2}} = S_{ID}$ y $\frac{k_{b3}}{k_{a3}} = S_{IE}$, donde S_{IT} , S_{ID} y S_{IE} representan las “sensibilidades insulínicas” correspondientes a cada uno de los efectos indicados.

2.4. Modelo del metabolismo de la glucosa

Se distingue entre glucosa en sangre, Q_1 (medible) y glucosa en el resto del organismo, Q_2 (no medible) [3]:

$$\begin{aligned}\frac{dQ_1(t)}{dt} &= -F_{01}^c(t) - x_1(t)Q_1(t) + k_{12}Q_2(t) - \\ &\quad - F_R(t) + U_G(t) + EGP_0(1 - x_3(t))\end{aligned}\quad (8)$$

$$\frac{dQ_2(t)}{dt} = x_1(t)Q_1(t) - (k_{12} + x_2(t))Q_2(t) \quad (9)$$

$$G(t) = \frac{Q_1(t)}{V_G} \quad (10)$$

donde k_{12} representa la constante de transferencia desde sangre a los tejidos, V_G representa el volumen de distribución del compartimento medible, G es la concentración de glucosa en sangre y EGP_0 la producción de glucosa endógena en ausencia de insulina.

La función F_{01}^c representa el consumo de glucosa independiente de la insulina (sistema nervioso principalmente) y F_R representa la eliminación renal de glucosa. Para estas funciones se ha considerado una expresión distinta de la considerada en [3], permitiendo evitar una definición a tramos, aunque dan lugar a la misma función:

$$\begin{aligned} F_{01}^c(t) &= \frac{f_{01}}{9} (G(t) - 4,5 \\ &\quad - \sqrt{(G(t) - 4,5)^2}) + f_{01} \end{aligned} \quad (11)$$

$$\begin{aligned} F_R(t) &= \frac{0,003V_G}{2} (G(t) - 9 \\ &\quad + \sqrt{(G(t) - 9)^2}) \end{aligned} \quad (12)$$

En el Cuadro 1 figuran los distintos valores de los parámetros que se han utilizado [3, 4].

Cuadro 1: Parámetros del modelo glucorregulatorio

Símbolo	Valor	Unidades
A_G	0.8	sin unidades
$t_{\max,G}$	40	min
V_I	0.12	L/Kg
k_e	0.138	1/min
k_{a1}	0.0157	1/min
k_{a2}	0.0231	1/min
k_{a3}	0.0143	1/min
S_{IT}	$18,7 \times 10^{-4}$	min^{-1} per mUL^{-1}
S_{ID}	$6,1 \times 10^{-4}$	min^{-1} per mUL^{-1}
S_{IE}	379×10^{-4}	mUL^{-1}
k_{12}	0.0871	1/min
f_{01}	0.0075	$\text{mmol}/(\text{kg min})$
V_G	0.13	L/kg
EGP_0	0.0143	$\text{mmol}/(\text{kg min})$

3. Simulación de sistemas dinámicos con incertidumbre

Como se ha comentado anteriormente, el uso de un modelo glucorregulatorio como predictor de glucemia requiere inevitablemente la consideración de incertidumbre en los parámetros, entrada y condiciones iniciales.

Dicha incertidumbre puede representarse sustituyendo valores reales por intervalos, definidos a partir de estimaciones de los valores mínimos y máximos que el parámetro en cuestión puede tomar. Ello da lugar a modelos intervalares de la forma

$$\dot{x} = f(x, p, t) \quad (13)$$

donde $p \in P = [\underline{P}, \bar{P}]$, siendo P un vector de intervalos, \underline{P} su extremo inferior y \bar{P} su extremo superior. De la misma forma puede representarse incertidumbre en las condiciones iniciales y en la entrada al sistema.

A diferencia de los modelos cuantitativos, el comportamiento de los modelos intervalares estará representado por una *envolvente* del conjunto de posibles trayectorias (familia de curvas temporales). Se dice que dicha envolvente es *completa* si incluye todas las trayectorias posibles [6] y es *correcta* si cada punto dentro de la envolvente pertenece a una trayectoria del conjunto. Una envolvente correcta pero no completa es una *envolvente interna*. Dicha envolvente infraacotará el conjunto de trayectorias. Una envolvente completa pero no correcta incluye puntos que no corresponden a ninguna trayectoria, luego es una sobreacotación del conjunto de trayectorias o *envolvente externa*. La envolvente exacta será completa y correcta. Estas propiedades se resumen gráficamente en la figura 2.

En este trabajo se plantea como objetivo la obtención de *envolturas externas con sobreacotamiento mínimo o nulo* de la glucemia. Para ello se combinarán técnicas de sistemas cooperativos y análisis intervalar modal.

Sistemas cooperativos. Un sistema es cooperativo si un incremento en cualquier estado induce un incremento en los estados con los que está relacionado. Esto es especialmente así en los sistemas biológicos donde los cambios en

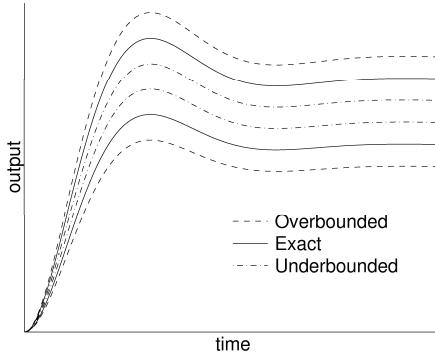


Figura 2: Envoltore interno, externa y exacta

los estados dependen positivamente (cooperatividad) o negativamente (competitividad) del resto (son “activados” o “inhibidos”). Matemáticamente, la propiedad de cooperatividad de un sistema se caracteriza mediante su Jacobiano, que debe ser una matriz Metzler (todos los elementos fuera de la diagonal principal deben ser no negativos) [8, 5].

En el caso de que el sistema sea cooperativo será posible obtener la envoltura exacta, que corresponderá a dos trayectorias específicas de la familia de curvas. Esto es así porque el espacio de trayectorias hereda la relación de orden definida sobre el espacio de estados, es decir, que $x(0) \leq x'(0) \Rightarrow x(t) \leq x'(t)$, $t > 0$.

En particular, dado el sistema incierto $f(x, p, t)$ con $p \in P$, siendo $P = [\underline{P}, \bar{P}]$ un vector intervalar, se cumple que si existe un par de sistemas cooperativos

$$\dot{x} = f(x, \underline{P}, \bar{P}, t) \quad \text{y} \quad \dot{\bar{x}} = \bar{f}(\bar{x}, \underline{P}, \bar{P}, t)$$

que satisfacen

$$f(x, \underline{P}, \bar{P}, t) \leq f(x, p, t) \leq \bar{f}(x, \underline{P}, \bar{P}, t)$$

para todo $p \in [\underline{P}, \bar{P}]$, $t \geq 0$ y $x \in D$, y si

$$x_0(\underline{P}, \bar{P}) \leq x_0(p) \leq \bar{x}_0(\underline{P}, \bar{P})$$

entonces

$$x(t) \leq x(t) \leq \bar{x}(t)$$

para todo $t \geq 0$. Es decir, basta con simular los sistemas cooperativos puntuales resultantes de

acotar superior e inferiormente a f (sistemas limitantes).

Análisis intervalar modal. En el caso de que el sistema no cumpla las condiciones anteriores, será preciso recurrir a métodos numéricos para la evaluación de la envoltura. Para ello se discretizará el sistema continuo y se evaluará en cada iteración el rango de x_{k+1} en función de los intervalos que describen la familia de estados anteriores, x_k , y entradas.

Dados los problemas de sobreestimación de la aritmética intervalar clásica para el cálculo de rangos de funciones cuando hay parámetros multiincidentes, en este trabajo se abordará el problema mediante el análisis intervalar modal (MIA) [1], que proporciona herramientas para el cálculo de envolventes óptimas.

Utilizando MIA, cada función intervalar se analiza y se expresa, si es posible, en su forma óptima (aplicando los denominados teoremas de optimalidad [1]). En este caso la evaluación directa mediante aritmética intervalar modal (librería IvalDB) proporcionará el rango exacto para la función expresada en su forma óptima. Si la función no verifica las condiciones necesarias para poder expresarla en su forma óptima, se emplearán métodos de *branch-and-bound* (algoritmo de cálculo de f^* [2]) para obtener una aproximación de la evaluación óptima del rango de la función.

4. Aplicación a la predicción de glucemia en diabetes tipo 1

A continuación se analizarán los modelos de los diferentes subsistemas del sistema glucoregulatorio para la predicción de glucemia ante incertidumbre en los parámetros, entrada y condiciones iniciales. Se considerará aquí que el subsistema de absorción subcutánea de insulina no presenta incertidumbre (sistema de ecuaciones en derivadas parciales). Dicha suposición es factible dado que, comparativamente, la sensibilidad al resto de parámetros es mucho más significativa.

Respecto al resto de subsistemas, en conjunto no es cooperativo ya que dQ_1/dx_3 y dQ_2/dx_2 son negativos, es decir, la insulina *inhibe* la producción de glucosa por parte del

hígado y potencia el consumo de glucosa en los músculos y grasa.

Sin embargo, el subsistema de farmacocinética y farmacodinámica de insulina (sección 2.3) sí que lo es, ya que a mayor flujo exógeno de insulina, mayor concentración de insulina en sangre y mayores serán sus efectos. Como es lineal, el Jacobiano es la matriz de estados. Si expresamos el modelo en representación interna:

$$\dot{\mathbf{z}} = \begin{pmatrix} -k_e & 0 & 0 & 0 \\ k_{b1} & -k_{a1} & 0 & 0 \\ k_{b2} & 0 & -k_{a2} & 0 \\ k_{b3} & 0 & 0 & -k_{a3} \end{pmatrix} \mathbf{z} + \begin{pmatrix} 1/V_i \\ 0 \\ 0 \\ 0 \end{pmatrix} I_{\text{ex}}(t)$$

donde $\mathbf{z} := (I, x_1, x_2, x_3)^T$. Dada que todas las constantes del modelo son positivas, la matriz de estados es Metzler y el subsistema es cooperativo. El sistema limitante superior corresponderá a

$$\dot{\mathbf{z}} = \begin{pmatrix} -k_e & 0 & 0 & 0 \\ \frac{k_{b1}}{k_{b1}} & -k_{a1} & 0 & 0 \\ \frac{k_{b2}}{k_{b2}} & 0 & -k_{a2} & 0 \\ \frac{k_{b3}}{k_{b3}} & 0 & 0 & -k_{a3} \end{pmatrix} \mathbf{z} + \begin{pmatrix} 1/V_i \\ 0 \\ 0 \\ 0 \end{pmatrix} \overline{I_{\text{ex}}(t)}$$

mientras que el inferior se obtiene intercambiando en el sistema anterior los límites inferiores de los intervalos por los superiores y viceversa.

El resto de modelos será necesario tratarlos mediante métodos intervalares.

Modelo intervalar de absorción y digestión de carbohidratos. El modelo de absorción y digestión de carbohidratos presenta incertidumbres debido a que el paciente generalmente no conoce la composición de la comida que ingiere ni la cantidad exacta de carbohidratos que contiene, por lo que deben ser estimados. Además, el tiempo de absorción dependerá de la composición de la comida, del peso del paciente, etc y por lo tanto también será una aproximación.

Consideraremos por tanto los parámetros como intervalos: $D_G = [\underline{D}_G, \bar{D}_G]$, $A_G = [\underline{A}_G, \bar{A}_G]$ y $t_{\max,G} = [\underline{t}_{\max,G}, \bar{t}_{\max,G}]$, dando lugar al modelo intervalar

$$\begin{aligned} & |\underline{U}_G(t), \bar{U}_G(t)| = \\ & = \frac{t * [\underline{D}_G, \bar{D}_G] * [\underline{A}_G, \bar{A}_G] * \exp(-t / [\underline{t}_{\max,G}, \bar{t}_{\max,G}])}{[\underline{t}_{\max,G}, \bar{t}_{\max,G}]^2} \end{aligned}$$

Puesto que la variable $t_{\max,G}$ es multiincidente en (5) y no se cumplen las condiciones de monotonía necesarias para obtener un cálculo óptimo mediante aplicación directa de los operadores intervalares, el cálculo del rango anterior debe realizarse utilizando el algoritmo f^* .

Modelo intervalar del metabolismo de la glucosa. En este caso se consideran inciertos los parámetros EGP_0 , f_{01} y k_{12} . Notar qué incluye en el caso en que no se considerara incertidumbre en los parámetros de este subsistema, habría que tratarlo de forma intervalar, ya que tanto la salida del subsistema de digestión y absorción de carbohidratos, U_G , así como los estados X_i , $i = 1, \dots, 3$ del subsistema de la insulina, son argumentos de entrada intervalares para este modelo.

Dada la discretización

$$Q_1(t+1) = -F_{01}^c(t) * t + Q_1(t) * (1 - t * X_1(t)) + +t * (k_{12} * Q_2(t) - F_R(t) + U_G(t) +$$

$$+ \text{EGP}_0 * (1 - X_3(t)))$$

$$Q_2(t+1) = Q_2(t) * (1 - t * (k_{12} + X_2(t))) + +X_1(t) * Q_1(t) t$$

$$G(t) = \frac{Q_1(t)}{V_G}$$

es posible obtener cálculos óptimos de la variable Q_2 mediante cálculos racionales usando la librería de intervalos modales IvalDB, mientras que para la variable Q_1 es necesario recurrir al algoritmo de cálculo de f^* .

5. Resultados

Para demostrar la fiabilidad de la aproximación intervalar, se ha considerado un paciente virtual con los parámetros nominales descritos

en el Cuadro 1. Para este paciente, se han simulado tres días con el itinerario de comidas e inyecciones de insulina bolus y basal indicadas en el Cuadro 2.

Cuadro 2: Itinerarios de comida e inyecciones de insulina.

Tiempo (h)	CHO (g)	Insulina basal Pro-taphane (IU)	Insulina bolus Actrapid (IU)
8:20		8	5
8:40	75		
13:15			9
13:20	120		
17:50	20		
21:10		7	
21:20			6
21:30	100		

Se han considerado dos escenarios distintos

- Escenario 1: 10 % de variación en los parámetros del modelo de digestión y absorción de carbohidratos (A_G , $t_{\max,G}$).
- Escenario 2: 10 % y 20 % de variación en los parámetros de sensibilidad insulínica del modelo farmacodinámico de la insulina (S_{ID} y S_{IE})

Las Figuras 3 y 4 muestran la simulación intercalar de la concentración de glucosa en sangre en un periodo de 24 horas correspondiente al tercer día de simulación.

Respecto a la figura 3, se hace patente la sensibilidad del nivel de glucemia respecto a la dinámica de la ingesta, produciendo variaciones de hasta 150 mg/dl en los picos que aparecen tras una ingesta. Aunque en estados de hiperglucemia esto puede no ser crítico, la cuantificación de la variación inducida por la incertidumbre sí que será crítica para la prevención de hipoglucemias. Así por ejemplo, a las 18:00 horas se observa una predicción de glucosa de [30,90] mg/dl, es decir, el estado del paciente puede variar desde la normoglucemia hasta la hipoglucemia grave. La inyección bolus de las 13:15 no es por tanto adecuada y presenta un gran riesgo.

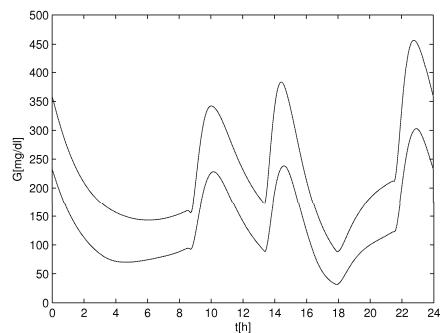


Figura 3: Respuesta obtenida con un 10 % de variación en A_G y $t_{\max,G}$

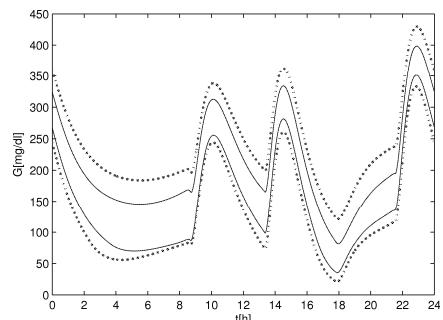


Figura 4: Respuesta para un 10 % y 20 % de variación en S_{ID} y S_{IE}

Resultados similares pueden observarse en la figura 4, respecto a la influencia de las sensibilidades insulínicas, produciéndose riesgo de hipoglucemia hacia las 18:00 horas. En este caso, se observa además que una variación del 20 % en las sensibilidades puede producir un nivel de glucosa en ayunas (8:00 horas) de [80,200] mg/dl, abarcando estados de normoglucemia e hiperglucemia.

La consideración de la incertidumbre en la predicción es por tanto primordial si de ella se quieren tomar decisiones terapéuticas más seguras. El uso de modelos cuantitativos es insuficiente, ya que originarán en muchos casos falsos-negativos (situaciones de riesgo no detectadas).

6. Conclusiones y trabajo futuro

El Análisis Intervalar Modal y la teoría de sistemas cooperativos permite abordar la predicción de las excursiones de glucosa en pacientes con diabetes tipo 1 teniendo en cuenta información incierta. Se ha considerado incertidumbre en los parámetros tales como las sensibilidades hepáticas y periféricas de la insulina, así como en los gramos de carbohidratos de una comida, con el fin de predecir los efectos que produce una terapia dada de insulina sobre la glucemia. Se obtiene como resultado, una banda que contiene todas las posibles excursiones de glucosa sufridas por el paciente teniendo en cuenta la incertidumbre considerada.

Este planteamiento es mucho más realista que la consideración de valores puntuales, permitiendo obtener una predicción más segura de los posibles episodios de hiperglucemia y de hipoglucemia inducidos por la terapia, consiguiendo a su vez una reducción del número de falsos-negativos.

Los métodos aquí desarrollados se están integrando en un sistema de ayuda a la decisión que permita reducir el riesgo de hiperglucemias e hipoglucemias.

Referencias

- [1] E. Gardeñes, M.A. Sainz, L. Jorba, R. Calm, R. Estela, H. Mielgo, and A. Trepaut. Modal intervals. *Reliable Computing*, 7(2):77–111, 2001.
- [2] P. Herrero, M.A. Sainz, J. Vehi, and L. Jaulin. Quantified set inversion algorithm with applications to control. *Reliable Computing*, 11(5):369–382, 2005.
- [3] R. Hovorka, V. Canonico, L.J. Chassin, U. Haueter, M. Massi-Benedetti, M. Orsini Federici, T.R. Pieber, H.C. Schaller, L. Schaupp, T. Vering, and M.E. Wilinska. Nonlinear model predictive control of glucose concentration in subjects with type 1 diabetes. *Physiological Measurement*, 24(4):905–920, 2004.
- [4] R. Hovorka, F. Shojaee-Moradie, P.V. Carroll, L.J. Chassin, I.J. Gowrie, N.C. Jackson, R.S. Tudor, A.M. Umpleby, and R.H. Jones. Partitioning glucose distribution/transport, disposal, and endogenous production during IVGTT. *American Journal of Physiology, Endocrinology and Metabolism*, 282(5):992–1007, 2002.
- [5] H.L. Smith. Monotone dynamical systems: An introduction to the theory of competitive and cooperative systems. *American Mathematical Society*, 41, 1995.
- [6] P. Struss. *Readings in qualitative reasoning about physical systems*, chapter Problems of interval-based qualitative reasoning, pages 288–305. Morgan Kaufmann Publishers, Inc., 1990.
- [7] C. Tarín, E. Teufel, J. Picó, J. Bondia, and H.J. Pfleiderer. Comprehensive pharmacokinetic model of insulin glargine and other insulin formulations. *IEEE Transactions on Biomedical Engineering*, 52(12):1994–2005, 2005.
- [8] W. Walter. *Differential and Integral inequalities*. Springer-Verlag, 1970.
- [9] G. Williams and J.C. Pickup. *Handbook of Diabetes*. Blackwell Science Ltd., 1999.

SISTEMA DE AYUDA A LA TOMA DE DECISIONES EN LA GESTIÓN DEL CLIMA INTERIOR DE UN INVERNADERO BASADO EN MODELOS DE CRECIMIENTO DE CULTIVOS DE TOMATE

Javier Bonilla
Dpto. Automática e
Informática Industrial
PSA - CIEMAT
Tabernas
04200 Almería
España
javier.bonilla@psa.es

Francisco Rodríguez
Dpto. Lenguajes y
Computación
Escuela Politécnica Superior
Universidad de Almería
La Cañada de San Urbano
04120 Almería
España
frodrig@ual.es

Luís José Yebra
Dpto. Automática e
Informática Industrial
PSA - CIEMAT
Tabernas
04200 Almería
España
luis.yebra@psa.es

Armando Ramírez
Dpto. de Horticultura
Protegida
Universidad Autónoma de
Chapingo
Chapingo
56230 Estado de México
México
ariasar_mx@yahoo.com.mx

Resumen

En este artículo se presenta una herramienta software que simula el crecimiento de un cultivo de tomate utilizando varios modelos basados en principios fisiológicos vegetales (*Tomgro* 1991, *Tomgro* 1999 y *Tomsim*). La herramienta proporciona resultados mediante los cuales se puede determinar la producción, en función de las variaciones en las condiciones climáticas de su entorno. De esta forma se puede utilizar para controlar el crecimiento del cultivo de tomate en distintos escenarios, realizando un control óptimo del mismo.

1. Introducción

El crecimiento del cultivo está principalmente influenciado por las variables climáticas del entorno y por la cantidad de agua y fertilizantes que son proporcionados por la fertirrigación. Por lo tanto, el correcto control sobre estas variables permitirá el control del crecimiento de un cultivo. Esta es una de las principales razones por la que los invernaderos son los entornos ideales puesto que la fertirrigación y las variables climáticas pueden ser controladas de manera óptima.

Empíricamente, los requisitos de agua y nutrientes de los diferentes tipos de cultivo son conocidos y pueden ser automatizados. Por lo tanto el problema de gestión del cultivo en un

invernadero se puede simplificar suponiendo que las plantas reciben la cantidad adecuada de agua y fertilizantes en cada momento debiendo controlar las condiciones climáticas del entorno.

El modelado del crecimiento de cultivo en función de las variables climáticas es una buena herramienta que se puede utilizar para la toma de decisiones en la gestión de la producción. El desarrollo de estos modelos ha sido motivado por la necesidad de disponer de información cuantitativa para mejorar la toma de decisiones en la gestión de los cultivos y se ha impulsado por la utilización masiva del computador como herramienta de trabajo con altas prestaciones y bajo coste [3]. Estos modelos son unas herramientas que sirven para probar hipótesis, sintetizar el conocimiento, describir y comprender sistemas complejos y comparar distintos escenarios. Los modelos de crecimiento pueden ser utilizados para la toma de decisiones en la gestión de la producción a tres niveles [9]:

- **Estratégico.** Los modelos proporcionan una estimación de la producción, beneficios, gastos y riesgos asociados con la inversión de capital en sistemas como los de control climático o de fertirrigación con una escala de tiempo de años.
- **Táctico.** Los modelos se utilizan junto a los climáticos para determinar cuando debe plantarse, recolectar o para determinar una estrategia general del clima del invernadero

para toda la campaña con una escala de tiempo en meses.

- *Operación.* Se utilizan para ajustar hora a hora o día a día las desviaciones que se producen en las condiciones del entorno modeladas en el nivel táctico, por lo que tienen una escala de tiempo de horas o días.

El principal objetivo de este trabajo ha sido el desarrollo de una herramienta software que simule el crecimiento del cultivo de tomate en función de las variables climáticas, utilizando para ello varios modelos basados en principios fisiológicos. Esta herramienta proporciona resultados para realizar un control óptimo del crecimiento del tomate modificando las condiciones climáticas de su entorno. Por otra parte, para alcanzar el objetivo principal ha sido necesario satisfacer una serie de objetivos secundarios como el estudio e implementación de modelos de cultivo, el estudio de librerías de resolución de sistemas de ecuaciones diferenciales ordinarias y el diseño e implementación de la interfaz gráfica con el usuario que sea amigable y fácilmente utilizable por los actores de este campo como son los técnicos agrícolas y los propios agricultores.

2. Modelado del crecimiento del cultivo de tomate bajo invernadero

Los cultivos son organismos biológicos que se ven afectados por una gran cantidad de factores físicos, químicos y biológicos de su entorno. Se han desarrollado modelos que usualmente se centran en la respuesta de los cultivos a alguno de estos factores, asumiendo que el resto se encuentran a niveles óptimos. En los modelos que se han desarrollado, se describe el efecto del clima del invernadero (temperatura, radiación PAR¹ y concentración de CO₂) sobre el crecimiento del cultivo. Se asume que el cultivo tiene un adecuado plan de riego y fertilización, que no presentará problemas de enfermedades y no sufrirá ningún tipo de plagas. Es una simplificación necesaria, aunque limita la aplicabilidad de los modelos en el caso de que algunas de esas condiciones no se cumplan. Afortunadamente, en cultivos bajo invernadero el aporte de agua y nutrientes se

controla fácilmente y los sistemas utilizados actualmente funcionan adecuadamente.

Existen dos tipos de modelos de crecimiento de cultivo.

- Los modelos *empíricos* se basan en datos de entrada y salida que ajustan curvas predefinidas o estructuras matemáticas más o menos complejas. Los modelos empíricos se basan en pocas variables de estado, lo que supone un coste computacional bajo que permite ser utilizado por sistemas de control climático de invernadero [13]. Aunque estos modelos presentan buenos resultados, no ocurre lo mismo al extrapolarlos a otras especies o localizaciones. Otra cuestión importante es que al añadir nuevos factores de entrada es necesario rediseñar el modelo completamente.
- Los modelos *basados en principios fisiológicos* describen cuantitativamente los mecanismos y procesos que causan el crecimiento de los cultivos. Suelen ser modelos jerárquicos, en los que el nivel superior describe la producción de materia seca, que da lugar al crecimiento del cultivo y en un nivel inferior la fotosíntesis y la expansión del índice de área foliar.

Para este trabajo se ha optado por utilizar este último tipo de modelos ya que se conocen los principios en los que se basan y se encuentran muy documentados, además de ayudar a comprender mejor las respuestas del cultivo a las entradas y perturbaciones. Por otra parte, son más fáciles de ampliar si, en un futuro, se desean añadir los efectos del aporte de agua, la fertilización, técnicas culturales (podas, etc.) o las consecuencias de plagas y enfermedades.

Las plantas se componen de agua y materia seca (materia que resulta al desecar la planta, compuesta principalmente por las paredes de las células, sus componentes y los azúcares que contienen). Como se muestra en la figura 1, por fotosíntesis se produce un incremento en la materia seca ya que la planta ha creado azúcares, que se utilizan para crear nuevas células (crecer) y como energía para el resto de sus procesos mediante la respiración [2].

¹ Photosynthetically Active Radiation (440-770 nm)

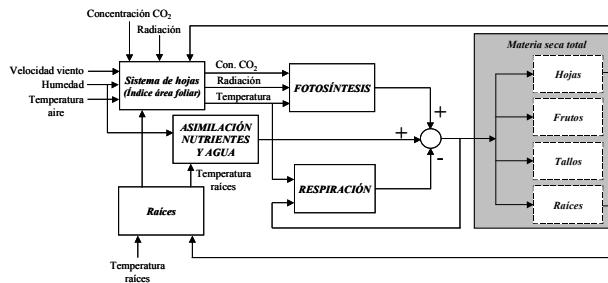


Figura 1. Crecimiento de cultivos

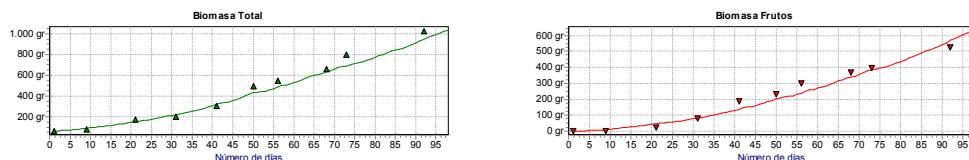
El crecimiento de una planta se refiere al incremento de peso, tanto de materia seca como de agua, por lo que un incremento de materia seca (dividida en raíces, tallos, hojas y frutos) proporciona un criterio de medida del crecimiento de un cultivo, aunque no sea del todo coincidente. Por tanto, al estar todos los modelos de este tipo basados en la fotosíntesis, es necesario calcular previamente la radiación interceptada por el cultivo a través del área foliar, a continuación, hay que modelar la respiración para poder calcular la conversión en materia seca total y su división entre los distintos órganos de la planta, para finalmente estimar el peso real, compuesto por materia seca más agua de cada órgano de la planta.

En la bibliografía existen dos importantes líneas de trabajo en el modelado del crecimiento de un cultivo de tomate: *Tomgro* (Universidad de Florida, USA) [7][8][9] y *Tomsim* [6] (Universidad de Wageningen, Holanda). Ambos se basan en la estructura descrita anteriormente (fotosíntesis y respiración) pero en el modelado de los principios fisiológicos presentan unos enfoques diferentes.

Para analizar cual se adapta mejor a las condiciones climáticas mediterráneas, a las estructuras de los invernaderos del Sudeste de España y a las variedades de tomate que se cultivan en estas zonas, se han adaptado ambos

modelos. Para una descripción completa del proceso, se ha optado por *Tomgro* 1999 con cinco variables de estado, ya que permite simular la dinámica sin perder su carácter explicativo, aunque la distribución de fotoasimilados es realizada de manera simple; y *Tomsim*, con más variables de estado, permitiendo evaluar la dinámica de crecimiento y desarrollo con más detalle, de cada rabillete de tomate. *Tomsim* realiza una distribución de fotoasimilados basada en experimentación agronómica, aunque implica un mayor coste computacional. Por otra parte, hay que indicar que para algunas aplicaciones como el control en tiempo real o la optimización de consignas en sistemas de control jerárquico, se ha utilizado un modelo *Tomgro* reducido [7] que predice el comportamiento del cultivo de forma no completa, proporcionando un primer nivel en la descripción cuantitativa de las necesidades biológicas de un cultivo de tomate.

Las características principales de estos tres modelos se muestran en la tabla 1. La formulación para adaptarlos a estas condiciones y la implementación de todos ellos se puede consultar en [11] y [12]. Para la calibración y validación de los modelos, se han utilizado un conjunto de datos tomados directamente sobre el cultivo y se han comparado con los resultados obtenidos, con cada uno de los modelos de crecimiento.

Figura 2. Validación del modelo *Tomsim* (a) Biomasa total (b) Biomasa frutos

Modelo	<i>Tomgro</i> 1991	<i>Tomgro</i> 1999	<i>Tomsim</i>
Autor	Jones, JW.	Jones, JW.	Heuvelink, E.
Año	1991	1999	1996
Referencias	[7], [8], [12]	[7], [9], [11]	[5], [6], [11]
Experiencias	1994, Gainesville (Florida USA) 1993, Avignon (Francia) 1999, Almería (España)	1988-1989 Lake city (Florida USA), 1994, Gainesville (Florida USA) y Avignon (Francia) 2002, Almería (España)	Holanda, invernaderos tipo venlo de cristal, probado en diferentes cultivares de tomate 2002, Almería (España)
Variables que modela	Número de nodos Biomasa total Índice de Área Foliar (LAI)	Número de nodos Biomasa total Índice de Área Foliar (LAI) Biomasa frutos Biomasa frutos maduros	Número de nodos Biomasa total Índice de Área Foliar (LAI) Biomasa frutos Biomasa frutos maduros Biomasa hojas Biomasa tallo Biomasa raíz Biomasa ramillete Biomasa unidad vegetativa
Escala de tiempo	Minutos	Minutos	Minutos y días
Modelado de los frutos	No	Biomasa frutos y frutos maduros	Biomasa frutos, frutos maduros, por ramillete

Tabla 1. Características principales de los modelos de crecimiento de cultivo de tomate *Tomgro* 91, 99 y *Tomsim*

Estos datos corresponden a cinco campañas agrícolas desde febrero de 1999 a junio de 2003 con distintas densidades de plantación y diferentes variedades comerciales de tomate. Como ejemplo, en la figura 2 se muestra una comparación entre los datos reales medidos sobre el cultivo en la campaña de primavera de 2000 (triángulos) con el resultado del modelo *Tomsim* respecto a la biomasa total de las plantas y la biomasa de frutos (línea continua), obteniéndose errores no mayores de un 8%.

3. Desarrollo de una herramienta de modelado y crecimiento de cultivo de tomate bajo invernadero

Para el desarrollo de la herramienta se han llevado a cabo las siguientes tareas de ingeniería del software:

- Diagramas de casos de uso.
- Diagrama de clases.
- Definición y diseño de la interfaz de usuario.

- Implementación de la herramienta (modelos de crecimiento y utilización de resolvedores numéricos).

Como parte del análisis se han realizado los diagramas de casos de uso y como parte del diseño los diagrama de clases, siguiendo la metodología UML.

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso. Los principios que se han utilizado para el desarrollo de la herramienta son de utilidad para creación de interfaces funcionales y de fácil operación [14], cuestión importante en esta aplicación ya que los usuarios finales no disponen de una buena formación en informática.

Para la implementación de los distintos modelos fisiológicos de crecimiento de cultivo de tomate se ha utilizado el paradigma orientado a objetos que proporciona una interfaz simple para el programador ya que enlaza los modelos de crecimiento con la interfaz de usuario, además de

todos los beneficios de utilizar dicho paradigma como herencia, polimorfismo, encapsulación lo que facilita el mantenimiento y la reutilización del código fuente. Cada uno de los modelos de crecimiento de cultivo de tomate, está descrito por un sistema de ecuaciones diferenciales ordinarias (EDO) de dicho modelo de crecimiento, ecuaciones algebraicas, parámetros y constantes del modelo, variables climáticas para el modelo, variables auxiliares calculadas a partir de parámetros y las constantes del modelo.

La aplicación del resolvelor también se ha realizado empleando el paradigma orientado a objetos, se ha optado por utilizar el resolvelor de libre distribución *DASSL* [10] implementado en Fortan-77, diseñado para la resolución de sistemas implícitos de ecuaciones algebraicas y/o diferenciales, utilizando un método multipaso de orden variable basado en las formulas de integración por atraso (DBF). El método de Newton es empleado para resolver el sistema no lineal resultado de cada paso de integración.

Con el fin de realizar una comparación de los resultados que proporciona el método DASSL, con otros métodos utilizados en programas actuales, se ha realizado una comparación con MATLAB, utilizando el método ODE23S que se basa en la fórmula modificada de Rosenbrock de segundo orden, que resuelve correctamente la integración numérica de sistemas rígidos, en los que aparecen dinámicas con diferentes escalas temporales. Para ambos resolvelores se han utilizado los mismos errores de tolerancia relativa y absoluta. Los residuos obtenidos con DASSL son mucho menores que los obtenidos en MATLAB, tal y como se muestra en las figuras 3 y 4.

La herramienta software desarrollada, simula el crecimiento del cultivo de tomate bajo invernadero de una forma eficiente y mediante una interfaz de usuario intuitiva y consistente, proporcionando buenos resultados para realizar la gestión del clima de un invernadero para alcanzar distintos objetivos en el crecimiento de un cultivo de tomate.

La forma de trabajo de esta herramienta comienza con la *definición del experimento*, que proporciona información general sobre el ensayo, localización, fechas, características del cultivo, tipo de invernadero, actuadores y comentarios sobre los objetivos a conseguir (Figura 5).

Los *datos climáticos* (radiación PAR, temperatura y concentración de CO₂) que se van a estudiar con el experimento, se pueden indicar con distintas opciones (Figura 6):

- Datos climáticos en base a un año patrón de comportamiento climático.
- Datos climáticos reales tomados en invernaderos reales.
- Datos climáticos definidos por el usuario.

La herramienta calcula de forma automática la radiación PAR (mediante un modelo teórico de día claro), el usuario debe de proporcionar los datos relativos a la temperatura y concentración de CO₂.

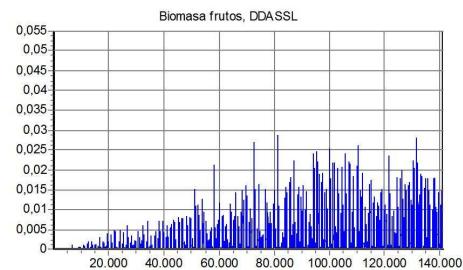


Figura 3. Residuos biomasa *Tomgro* 99 DASSL

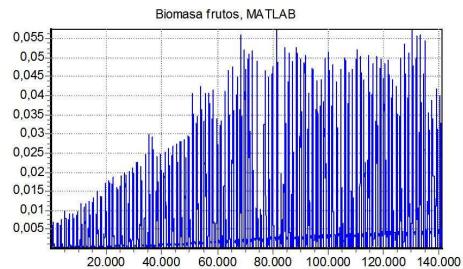


Figura 4. Residuos biomasa *Tomgro* 99 MATLAB

Opcionalmente se pueden especificar *datos de crecimiento*, si se dispone de datos reales de un cultivo. Puede ser muy útil indicarlos, ya que posteriormente se podrán comparar los resultados obtenidos en la simulación con los datos de crecimiento especificados, o pueden emplearse para validar parámetros de los modelos de crecimiento.

Para llevar a cabo la simulación es necesario especificar uno de los *modelos de crecimiento* disponibles en la aplicación (Figura 7). Además,

permite modificar los parámetros de cada uno de estos modelos ya que se ha demostrado en [11] que dependen de la variedad y del tipo de campaña (otoño o primavera).

Los distintos tipos de resultados se pueden consultar mientras se realiza el proceso de simulación o al finalizar la misma. Por ejemplo, el modelo de crecimiento *Tomsim* proporciona la siguiente información:

- Información Global (Figura 9). Representación gráfica de la producción de simulación, datos de producción de simulación y datos sobre el ensayo.
- Número de nodos, índice de área foliar, número de ramaletes, unidades vegetativas.
- Biomasa total, frutos y frutos maduros (Figura 8).
- Peso fresco total, frutos y frutos maduros.
- Biomasa ramaletes y unidades vegetativas.
- Peso fresco ramaletes y unidades vegetativas
- Biomasa órganos (hojas, tallo, raíz y frutos).
- Peso fresco órganos (hojas, tallo, raíz y frutos)
- Variables Fisiológicas: fotosíntesis y respiración.
- Fotoasimilados disponibles y demanda.
- Variables Climáticas (temperatura, radiación PAR, y concentración de CO₂).

4. Resultados

Los resultados obtenidos tras el desarrollo de la herramienta son los siguientes:

- Adaptación a las características del Sudeste de España de los distintos modelos de crecimiento considerados.
- Herramienta validada que simula el crecimiento del cultivo de tomate bajo invernadero y ayuda a la toma de decisiones en la gestión climática.

Con respecto al primer resultado, para llevar a cabo una comparación de los distintos modelos de crecimiento que incorpora la herramienta, se ha realizado una simulación con los mismos datos de entrada y se han analizado los resultados de cada uno de los modelos. En la tabla 2 se puede observar como los resultados para la biomasa y peso fresco son muy similares en los modelos

Tomgro 1999 y *Tomsim*. Hay que señalar que el modelo *Tomgro 1991* sólo estima el peso seco total, calculándose la biomasa y peso fresco de frutos mediante relaciones empíricas basadas en la experiencia. Como se puede observar, los resultados de este último modelo son muy similares a los de los modelos completos.



Figura 5. Definición del experimento



Figura 6. Selección de datos climáticos

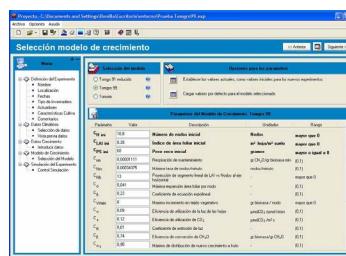


Figura 7. Modelo de crecimiento y parámetros

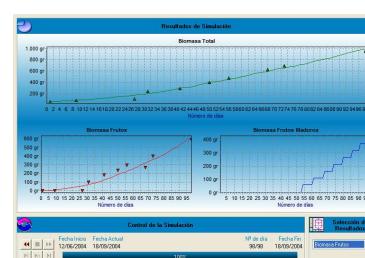


Figura 8. Resultados de biomasa



Figura 9. Panel resumen de información global

Con relación al segundo tipo de resultados, para el estudio de distintas estrategias climáticas para el cultivo de tomate se han probado distintas consignas de temperatura, radiación PAR y concentración de CO₂. Como ejemplo, se muestra la comparación de las siguientes tres estrategias climáticas de temperatura, con los mismos parámetros de experimento para realizar la simulación, considerando el modelo de día claro para la radiación PAR y una concentración constante de CO₂:

- *Estrategia 1.* Temperatura diurna constante a 24°C y temperatura nocturna constante a 16°C durante toda la campaña.
- *Estrategia 2.* Los primeros 44 días se utilizan consignas de temperatura elevadas (diurna a 22°C y nocturna a 18°C), 10 días de transición hasta que los últimos 44 días se utilizan consignas bajas de temperatura (diurna a 16°C y nocturna a 13°C).
- *Estrategia 3.* Los primeros 44 días se utilizan consignas de temperatura bajas (diurna a 16°C y nocturna a 13°C), 10 días de transición hasta

que los últimos 44 días se utilizan consignas elevadas de temperatura (diurna a 22°C y nocturna 18°C).

5. Conclusiones

Tras todo el proceso de desarrollo se pueden extraer las siguientes conclusiones:

- La herramienta proporciona resultados mediante los cuales se puede determinar la producción, en función de las variaciones en las condiciones climáticas de su entorno que ayuden a un control óptimo del crecimiento del cultivo de tomate.
- La herramienta ha sido desarrollada utilizando una metodología orientada a objetos, lo que facilita la reutilización, modificación o ampliación de cada uno de sus componentes.
- Los modelos de crecimiento de cultivo de tomate implementados, pueden ser adaptados a distintas regiones y variedades.
- Las condiciones climáticas en el interior del invernadero también son adaptables, lo que hace que la herramienta sea muy flexible.

Hay que destacar que se sigue trabajando en esta herramienta incorporando nuevos modelos de crecimiento de cultivo como el *Tompsonuse* [1], el de Koning [4] y modelos empíricos. Por otra parte, se prevé la incorporación de un modelo climático de invernadero en base a las condiciones externas por lo que se podrá evaluar el efecto de las diferentes estructuras de invernadero sobre el crecimiento de un cultivo, un sistema de análisis económico de costes e ingresos, para obtener una estimación de los beneficios. Además, se está analizando la posibilidad de ampliar la herramienta con otras variedades hortofrutícolas como pepino o pimiento.

Agradecimientos

Este trabajo está siendo financiado en el marco del proyecto CICYT DPI2004-07444-C04-04, la Fundación Cajamar y la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía.

	<i>Tomgro 1991</i>	<i>Tomgro 1999</i>	<i>Tomsim</i>
Biomasa total (Kgr/m ²)	0,931	0,975	0,997
Peso fresco total(Kgr/m ²)	15,521	16,244	16,613
Biomasa frutos(Kgr/m ²)	0,605	0,602	0,613
Peso fresco frutos (Kgr/m ²)	10,088	10,029	10,212
Biomasa frutos maduros (Kgr/m ²)	0,423	0,408	0,428
Peso fresco frutos maduros (Kgr/m ²)	7,062	6,792	7,137

Tabla 2. Comparación resultados modelos de crecimiento

	Estrategia 1	Estrategia 2	Estrategia 3
Número de nodos	47,41	43,97	43,68
Índice de área foliar	4,41	3,98	3,95
Biomasa total	1,621 kg/m ²	1,689 kg/m ²	1,538 kg/m ²
Biomasa frutos	0,967 kg/m ²	0,783 kg/m ²	0,836 kg/m ²
Biomasa frutos maduros	0,559 kg/m ²	0,443 kg/m ²	0,261 kg/m ²

Tabla 3. Estrategias climáticas de temperatura

Referencias

- [1] Abreu, P. Meneses, J.F. Gary, C. Tompousse, a model of yield prediction for tomato crops: calibration study for unheated plastic greenhouse, *Acta Horticulturae*, Nº 519, 2000.
- [2] Curry, R.B. Dynamic simulation of plant growth. Part I. Development of a model; Winter meeting of the ASAEE, Chicago, Estados Unidos, 1969.
- [3] Gary, C. Modelling greenhouse crops: state of arts and perspectives, World Conference on Horticultural Research, Roma, Italia, 1998.
- [4] Koning, A.N.M. de, Development and dry matter distribution in glasshouse tomato: a quantitative approach, PhD Thesis Dissertation, Agricultural University of Wageningen; Wageningen, The Netherlands, 1994.
- [5] Heuvelink, E. Dry matter partitioning in tomato: validation of a dynamic model, *Annals of Botany*, Nº 77, 1996.
- [6] Heuvelink, E. Evaluation of a dynamic simulation model for tomato crop growth and development, *Annals of Botany*, Nº 83, 1999.
- [7] Jones, J.W. A dynamic-tomato growth and yield model (*Tomgro*), *Transactions of the ASAEE*, Vol. 34(2), 1991. Jones, J.W. Crop growth, development and production modeling, *Symposium on Automated Agriculture for the 21st Century*, 1991.
- [8] Jones, J.W. Kening A. Vallejos, C.E. Reduced state-variable tomato growth model, *Transactions of the ASAEE*, Vol. 42 (1), 1999.
- [9] Petzold, L. R. A Description of DASSL: A Differential-Algebraic System Solver, Sandia National Laboratory Report Ed. IMACS Trans. on Scientific Computing, vol. 1. 1982.
- [10] Ramírez, A. Control jerárquico multiobjetivo de crecimiento de cultivos bajo invernadero, PhD. Thesis, Universidad de Almería, España, 2005.
- [11] Rodríguez, F. Modelado y control jerárquico de crecimiento de cultivos bajo invernadero, PhD. Thesis, Universidad de Almería, España, 2002.
- [12] Rodríguez, F. Yebra, L.J. Berenguel, M. Dormido, S. Modeling and simulation of greenhouse climate using Dymola, 15th IFAC World Congress, Barcelona, España, 2002.
- [13] Shneiderman, B. Designing the user interface: strategies for effective human-computer interaction, Ed. Addison Wesley, 1998.

Desarrollo de un modelo de red neuronal (*Sierra2*) para la selección de especies arbustivas del ámbito mediterráneo, en el marco del Sistema de Ayuda a la Decisión MicroLEIS DSS

Heredia, N.¹, Anaya-Romero, M.², Mayol, F.², Díaz-Pereira, E² & de la Rosa, D²

¹Centro Internacional de Altos Estudios Agronómicos Mediterráneos (CIHEAM), Instituto Agronómico Mediterráneo de Zaragoza (IAMZ), Avda. Montañana 1005, 50059 Zaragoza, España.

²Consejo Superior de Investigaciones Científicas (CSIC), Instituto de Recursos Naturales y Agrobiología de Sevilla (IRNAS), Avda. Reina Mercedes 10, 41012 Sevilla, España.

Resumen

A fin de obtener paisajes con la mayor diversidad, atendiendo tanto al tipo de especies autóctonas como a los estratos verticales de vegetación y dentro de un entorno mediterráneo, se ha desarrollado una herramienta informática (modelo *Sierra2*), como parte del sistema de ayuda a la decisión MicroLEIS DSS, que facilite la selección y distribución de especies de matorral a partir de información del medio físico andaluz y de especies de matorral presentes en Andalucía. Se utiliza como procedimiento de modelización las redes neuronales artificiales (RNAs), en base a su gran capacidad de predicción de los resultados e inmunidad al ruido. En una primera fase, se seleccionaron 39 especies representativas del matorral mediterráneo atendiendo a la distribución bioclimática propuesta por diversos autores. En una segunda fase, se establecieron los requerimientos geo-edafoclímáticos de cada una de las especies arbustivas seleccionadas y de esta información se fijó un máximo común divisor para que todas las especies estuvieran caracterizadas por el mismo número de 14 variables (5 de lugar, 3 de clima y 6 de suelo). A su vez, la revisión bibliográfica se complementó mediante información experta, a través de encuestas a especialistas en Botánica, Edafología y relaciones específicas suelo-planta. En una fase final, se calibró y validó la RNA *Sierra2* mediante un algoritmo de "correlación en cascada-quickpropagation" a partir del sistema arbustivo previamente formulado. Esta RNA se subdividió a su vez en dos subredes, la

primera incluyendo las tres variables más determinantes en la predicción de la aptitud arbustiva (piso bioclimático, precipitación media anual y pH), y la segunda considerando las restantes once variables (fisiografía, pendiente, proximidad a cursos de agua, influencia del hábito marino, exposición, resistencia a heladas, material original, textura, rocosidad, humedad del suelo y acumulación de yesos/sales); aportando esta última RNA una mayor precisión en el pronóstico de especies idóneas para cada lugar de aplicación. La primera subred se entrenó con todos los patrones de los que se disponía (40) y en la segunda con un 75 % (5585) se hizo el entrenamiento, y con el 25% restante (1861) se llevó a cabo la validación; siendo el error cuadrático medio de la subred primera de $8,33 \cdot 10^{-5}$ y el de la segunda de 0,083. A la vista de los primeros resultados, es de resaltar la alta fiabilidad del modelo *Sierra2* en la predicción de la aptitud arbustiva, a fin de seleccionar las especies más idóneas en cada escenario de aplicación.

1. Introducción

La gestión sostenible de los recursos naturales requiere un conocimiento exhaustivo de los componentes del medio físico y especialmente de las interacciones que se producen entre estos y las comunidades vegetales. Así las intervenciones humanas que se realizan en el medio natural, especialmente en los espacios protegidos, deben ser cuidadosamente planificadas para asegurar la

conservación de estos espacios y permitir el desarrollo socioeconómico de las poblaciones insertas en ellos [1], [6], [10] y [13]. Así pues, la gestión sostenible de los bosques (árboles, arbustos, matorral, etc.) se debe considerar como un aspecto primordial de la protección del suelo, basándose en principios agro-ecológicos [15].

En cuanto a las comunidades vegetales se define la vegetación potencial como la comunidad estable que existiría en un área dada como consecuencia de la sucesión geobotánica progresiva sin influencia antrópica sobre los ecosistemas vegetales [19]. Las dinámicas de sucesión y las especies más representativas en éstas, deben influir en la lista de especies arbustivas a considerar.

La importancia del matorral mediterráneo reside esencialmente en su capacidad de adaptación a las condiciones ecológicas adversas que caracterizan los ambientes mediterráneos, especialmente aquellas adaptaciones adoptadas ante el xerofitismo, que hacen que la utilización del matorral en la restauración vegetal asegure la supervivencia de la repoblación, incluso en situaciones de sequía prolongada.

En cuanto a los requerimientos ecológicos o niveles preferentes de desarrollo de especies vegetales, la regeneración natural es el resultado de una cadena de procesos conjuntos que ocurren durante períodos largos, desde la floración hasta la supervivencia o la mortalidad de semillas. El éxito de cada uno de estos procesos depende de muchos factores ambientales, como son los: i) factores climáticos, tales como temperatura, humedad y régimen de radiación; ii) factores edáficos, tales como humedad del suelo, profundidad del horizonte orgánico y textura; y iii) topografía, que puede condicionar los regímenes de temperatura y del agua. Estas consideraciones deben determinar la selección de las variables que mejor caractericen las especies a estudiar [22].

Con relación a los métodos matemáticos empleados en la evaluación de suelos y tierras, existen multitud de autores que han contribuido desde distintos puntos de vista a la variedad de modelos que establecen criterios de evaluación para predecir la capacidad o adaptabilidad del suelo. En el caso de las redes neuronales artificiales (RNAs) han sido aplicadas con éxito en la modelización de sistemas ecológicos y la

predicción del comportamiento y degradación de los sistemas naturales [5], [7], [11] y [12]. El interés por la aplicación de los modelos de RNA en los procesos biofísicos ha crecido rápidamente en los últimos años. Esta técnica basada en la inteligencia artificial ha demostrado su capacidad a la hora de analizar sistemas no lineales de múltiples variables, de discriminar de forma bastante acertada entre la información real y el ruido, y de generalizar, es decir, de procesar patrones de entrada que no se hubieran presentado antes, de una forma muy similar a cómo lo hace la inteligencia humana.

Los sistemas de ayuda a la decisión (DSS) son sistemas informáticos que aúnan información de distintas fuentes, ayudan en la organización, análisis de la información y facilitan la evaluación de hipótesis subyacentes al uso de modelos específicos [9], [20]. Dentro de este marco conceptual se desarrolla el sistema agro-ecológico de ayuda a la decisión MicroLEIS DSS [8]. Actualmente, MicroLEIS DSS constituye un amplio conjunto de herramientas informáticas de utilidad en la toma de decisiones para la sostenibilidad de múltiples esquemas agro-ecológicos. Su diseño sigue un esquema de caja de herramientas integradas de diferentes tipos de software: bases de datos, aplicaciones estadísticas, sistemas expertos, redes neuronales, desarrollos Web, aplicaciones GIS y otras tecnologías de la información.

El Sistema MicroLEIS DSS está integrado por tres módulos; el primero incluye las bases de datos de suelos, clima y de manejo; el segundo está orientado a la evaluación de la calidad de las tierras, y el tercero está dedicado a la evaluación de la vulnerabilidad de las tierras.

En términos de aptitud relativa forestal, originalmente el modelo Sierra, integrado en el segundo módulo de MicroLEIS DSS, se desarrolló para predecir la presencia/ausencia de 22 especies forestales representativas del estrato arbóreo de la vegetación mediterránea. Dicho modelo responde a los criterios propios de un sistema cualitativo, a partir de unos requerimientos edafoclimáticos establecidos para un conjunto de especies forestales, y sigue el procedimiento de limitación máxima.

El objetivo principal del presente trabajo de investigación metodológica es desarrollar una nueva herramienta informática (modelo *Sierra2*), como nuevo componente del sistema

de ayuda a la decisión MicroLEIS DSS. El modelo *Sierra2* se ha generado para evaluar la presencia/ausencia de un nuevo estrato de vegetación, el matorral. *Sierra2* facilita la selección de especies del ecosistema arbustivo mediterráneo, a partir de la información geodafoclimática de la zona de estudio. Para ello, utiliza como método de evaluación la red neuronal artificial.

2. Material y métodos

En la presente investigación se seleccionaron especies representativas del matorral mediterráneo andaluz. La lista inicial de especies, se obtuvo atendiendo a las distribuciones bioclimáticas dadas por Rivas-Martínez (1987), [2], [4], [16] y [17].

A continuación se seleccionaron y acotaron, según la bibliografía disponible, los parámetros ambientales con influencia en la distribución de las mismas. Las variables seleccionadas se clasificaron en tres categorías temáticas: lugar, clima y suelo [14].

Con toda la información bibliográfica ordenada en variables y niveles de generalización para cada especie, se realizaron consultas mediante encuesta a especialistas en Botánica, Edafología y en relaciones específicas suelo-clima-planta. Se obtuvo información adicional sobre los niveles de generalización que determinaban la presencia de las especies seleccionadas. De igual forma, se estableció la lista definitiva de especies, así como la lista de variables [14].

Para la creación de la red neuronal artificial (RNA) se utilizó el software Stuttgart Neural Network Simulator [21]. Dicha red neuronal se constituyó atendiendo a las características que se describen a continuación.

La función de entrada o patrón.

La entrada neta de activación de una neurona. En el presente estudio se utilizó la función de activación sigmoidal logística, así todas las neuronas transfieren a las siguientes un valor entre 0 y 1 que se multiplica por el valor del peso de la conexión.

Cuando la transferencia de la activación llega a las neuronas de la capa de salida, el valor resultante de la función de activación tendrá valores comprendidos entre (0,1).

En la etapa de aprendizaje, se ha utilizado el método backpropagation, ya que es de tipo supervisado.

Por último y con la finalidad de desarrollar la RNA que tiene la mejor ejecución con casos nuevos, es decir que sea capaz de generalizar, se suele subdividir los patrones de entrada en dos grupos: entrenamiento y validación [3], [14] y [18].

3. Resultados y discusión

Se seleccionaron 39 especies representativas del matorral mediterráneo (Tabla 1). Sobre esta lista de especies, y teniendo en cuenta los parámetros ambientales seleccionados con influencia en la distribución de las mismas, se realizó un máximo común divisor para que todas las especies estuvieran caracterizadas por el mismo número de 14 variables (5 de lugar, 3 de clima y 6 de suelo).

La agrupación de lugar está formado por las siguientes variables, con sus correspondientes niveles de generalización: piso bioclimático (termo, meso y supramediterráneo) pendiente ($<15^\circ$, $15-45^\circ$ y $>45^\circ$), fisiografía (llanura, ladera, colinas, barrancos y acantilados), proximidad a cursos de agua (sí/no), e influencia del hábito marino (sí/no). La agrupación de clima por: precipitación media anual (árido, semiárido, seco, subhúmedo y húmedo), exposición (norte/sur), y resistencia a heladas (elevada, moderada y nula). Y por último, la agrupación de suelo por: material original (rocas sedimentarias duras, blandas y rocas ígneas y metamórficas), reacción a caliza (nula, ligera y fuerte), textura (gruesa, media y fina), rocosidad (>40 y $<40\%$), régimen de humedad del suelo (seco y húmedo) y acumulación de yesos/sales (sí/no). La sección de control o sección vertical del suelo en donde medir las variables edáficas se establece entre 0 y 25 cm. de profundidad.

Con la información recopilada sobre las especies arbustivas se establecieron los patrones de presencia, los cuales constituirán todo el universo de posibilidades. Dichos patrones consisten en todas las combinaciones posibles de los niveles de generalización (todas las

Nombre vulgar	Nombre científico	Nombre vulgar	Nombre científico
Adelfa	<i>Nerium oleander</i> L.	Hiniesta	<i>Genista cinerea</i> subsp. <i>speciosa</i> Rivas Mart.
Aladierno	<i>Rhamnus alaternus</i> L.	Jara blanca	<i>Cistus albidus</i> L.
Albaida	<i>Anthyllis cytisoides</i> L.	Jara negra	<i>Cistus salvifolius</i> L.
Albardín	<i>Lygeum spartum</i> Loefl. ex L.	Jara rizada	<i>Cistus crispus</i> L.
Aliaga	<i>Genista scorpius</i> (L.)DC.	Labiérnago	<i>Phillyrea angustifolia</i> L.
Aulaga	<i>Ulex parviflorus</i> Pourret	Lentisco	<i>Pistacia lentiscus</i> L.
Boj	<i>Buxus balearica</i> Lam.	Madroño	<i>Arbutus unedo</i> L.
Bolina	<i>Genista umbellata</i> (L'Her.)Poiret	Majuelo	<i>Crataegus monogyna</i> Jacq.
Brezo	<i>Erica scoparia</i> L.	Mirto	<i>Myrtus communis</i> L.
Brezo de invierno	<i>Erica multiflora</i> L.	Palmito	<i>Chamaerops humilis</i> L.
Bufalaga	<i>Thymelaea hirsuta</i> (L.) Endl.	Retama	<i>Retama sphaerocarpa</i> (L.)Boiss.
Cantueso	<i>Lavandula stoechas</i> L.	Romero	<i>Rosmarinus officinalis</i> L.
Cantueso rizado	<i>Lavandula dentata</i> L.	Romero macho	<i>Cistus clusii</i> Dunal.
Coscoja	<i>Quercus coccifera</i> L.	Sabina	<i>Juniperus sabina</i> L.
Durillo andaluz	<i>Cotoneaster granatensis</i> Boiss.	Sabina negra	<i>Juniperus phoenicea</i> L.
Endrino	<i>Prunus spinosa</i> L.	Salado	<i>Atriplex halimus</i> L.
Enebro	<i>Juniperus oxycedrus</i> L.	Taray	<i>Tamarix canariensis</i> Willd.
Esparto	<i>Stipa tenacissima</i> L.	Tino	<i>Viburnum tinus</i> L.
Espino negro	<i>Rhamnus lycioides</i> L.	Tomillo	<i>Thymus vulgaris</i> L.
Espliego	<i>Lavandula latifolia</i> Medicus		

Tabla 1. Relación de las 39 especies arbustivas seleccionadas para el modelo *Sierra2*

categorías) de las 14 variables independientes, los que constituirían las variables de entrada (lugar, clima y suelo), en relación a los valores de idoneidad de las variables dependientes, que constituirían las variables de salida (39 especies arbustivas). En la Tabla 2, se muestran todas las variables de entrada que se han considerado en el modelo, clasificadas en una serie de niveles de generalización y estos a su vez codificados siguiendo una clasificación dummy de unos y ceros.

La arquitectura de la red deriva del entrenamiento mediante un algoritmo en cascada backpropagation, que tiene el objetivo de generar una red neuronal donde cada unidad aprende una tarea específica lo más rápido

posible, evitando el movimiento aleatorio de las unidades intermedias en el espacio del error.

Del entrenamiento de la red con las 14 variables de entrada y las 39 variables de salida, surgió una RNA-Sierra2 inicial, cuyos resultados eran satisfactorios en el entrenamiento, y también en la fase de validación con un error aceptable. No obstante, presentó el inconveniente de que la red no asimilaba la importancia de unas variables frente a otras. Esto podía originar que las variables que tienen más influencia en el resultado de la red, su dominio, se viese menguado por el conjunto de restantes variables.

Se puso de manifiesto que las variables que tenían más influencia en la presencia/ausencia o en la aptitud de una especie para su uso en

Variables	Niveles de generalización	Codificación	Número de neuronas
Piso bioclimático	Termomediterráneo	(01)	2
	Mesomediterráneo	(10)	
	Supramediterráneo	(11)	
Precipitación	Árido	(001)	3
	Semiárido	(010)	
	Seco	(011)	
	Subhúmedo	(100)	
	Húmedo	(101)	
Reacción a la caliza	Nula	(01)	2
	Ligera	(10)	
	Fuerte	(11)	
Fisiografía	Llanura	(001)	3
	Ladera	(010)	
	Colina	(011)	
	Barranco	(100)	
	Acantilado	(101)	
Pendiente	Ligera	(01)	2
	Moderada	(10)	
	Elevada	(11)	
Proximidad agua	Si	(01)	2
	No	(10)	
Hálito marino	Si	(01)	2
	No	(10)	
Exposición	Norte	(10)	2
	Sur	(01)	
Heladas	Elevada	(01)	2
	Moderada	(10)	
	Nula	(11)	
Material original	Sedimentarias duras	(01)	2
	Sedimentarias blandas	(10)	
	Ígneas y metamórficas	(11)	
Textura	Ligera	(01)	2
	Media	(10)	
	Pesada	(11)	
Rocosidad	Si	(01)	2
	No	(10)	
Humedad suelo	Seco	(01)	2
	Húmedo	(10)	
Yesos/Sales	Si	(01)	2
	No	(10)	

Tabla 2. Correspondencia entre las variables de entrada, niveles de generalización y neuronas del modelo RNA-*Sierra2*

reforestación eran el piso bioclimático, la precipitación y la reacción a la caliza del suelo. Debido a esta serie de consideraciones se subdividió las entradas a la red neuronal inicial en dos partes: la subred1, formada por las variables que se considerarán de mayor peso como el piso bioclimático, precipitación y pH del suelo, y que suponía el 75 % del valor final de aptitud arbustiva de cada una de las especies

y la subred2, estaría formada por el resto de once variables y que influiría en el 25 % final del valor de aptitud. Esto se hizo, para que el conjunto de las tres variables de la subred1, influyan en el resultado de igual manera que el conjunto de las 11 variables de la subred2, aunque individualmente el peso de una variable de la subred1, es mayor que el peso de una variable de la subred2.

De esta forma, otra ventaja añadida de subdividir la RNA (*Sierra2*) en dos subredes ha sido la de facilitar la aplicación y espacialización del modelo, de acuerdo con la cantidad y variabilidad de datos de entrada disponibles, así como con la escala de trabajo empleada.

Una vez establecidas las entradas y salidas que formarían las subredes se llevaron a cabo los entrenamientos de ambas. De la subred1 se disponía de 40 patrones de un total de 60 (debido a todas las combinaciones posibles entre los niveles de generalización de las variables de entrada). Al tratarse de subred pequeña se realizó el entrenamiento con el total de patrones disponibles, ya que el error cometido dedicando 10 patrones a validación era excesivo. Los patrones empleados en el entrenamiento de esta segunda subred fueron 5.585, que corresponden al 75% del total disponible, quedando 1861 patrones (25%) para la validación.

El entrenamiento de ambas subredes, se realizó con el mismo algoritmo de entrenamiento backpropagation de correlación en cascada. Con este algoritmo la red crece en saltos, empieza sin unidades intermedias y se van agregando una por una todas las capas de neuronas ocultas durante el entrenamiento, hasta que el error total sea el que se pretende conseguir, que en el caso de la RNASierra2 fue de $R^2=0,95$.

De esta forma los nodos ocultos que se necesitaron en la subred1 fueron 12 y en la subred2 41, cada uno de los nodos forma a su vez una capa oculta, como puede observarse en la arquitectura final de la red (Figura 1). Este enfoque elimina la necesidad de adivinar el tamaño de la red neuronal de antemano y provee una tipología multi-nivel adecuada al problema a resolver.

El error final de entrenamiento de la subred1 fue de $8,33 \times 10^{-5}$; y para la subred2 que fue validada con el 25% de patrones (1.861) se consiguió un error final de validación de 0,083. Esto indica la gran capacidad de predicción de esta RNA-*Sierra2*, ya que para valores nuevos que no había visto nunca (fase de validación) daba un resultado casi idéntico al que se disponía de partida, sólo cometiendo errores no en la lista de especies seleccionadas sino en los rangos del índice de aptitud de ellas.

Este índice de aptitud viene influenciado por dos factores: la presencia y los óptimos en el patrón de entrada, siendo mayor cuanto mayor sea el número de niveles de generalización (del patrón de entrada) que la especie coincida o si en alguno/s de los niveles de generalización la especie presenta uno o varios óptimos. Aquella especie que esté presente en todos los niveles de generalización descritos y que mayor número de óptimos tenga para ese patrón, obtendrá el índice mayor, es decir igual a 1.

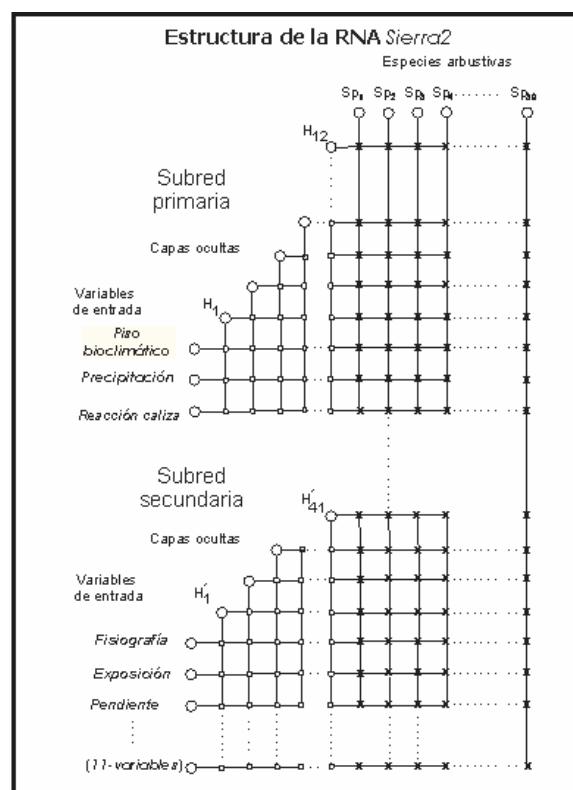


Figura 1. Estructura de la red neuronal artificial Sierra 2, donde SP₁,..., SP₃₉ indican las salidas de especies arbustivas; unidades de entrada, que corresponden a los niveles de generalización de las variables ambientales seleccionadas; H₁,..., H₁₂, las capas ocultas de la subred primaria; H'₁,..., H'₄₁, las capas ocultas de la subred secundaria; X, las conexiones entrenadas; y □, las conexiones congeladas.

Dado que la función de activación aplicada a las neuronas de la RNA-Sierra2 ha sido la sigmoidal logística que toma valores entre 0 y 1. De las neuronas de salida se obtendrá un valor comprendido en este rango e indica la aptitud arbustiva para la reforestación de las distintas especies (39) con ese patrón de entrada.

El incremento en el valor final de idoneidad por la presencia de óptimo/s es distinto dependiendo de si se dan en la subred1 o en la subred2 por los mismos motivos comentados en el apartado de los pesos de las variables de la red neuronal. Un óptimo de una especie en algunos de los niveles de generalización de la subred1 se premia con +0.1 puntos de idoneidad frente a las demás especies que no presentan ese óptimo, y en la subred2 cada óptimo se premia con +0.03.

Por lo tanto la RNA-Sierra2 estaría compuesta por dos subredes, cuyos parámetros pueden verse en la Tabla 3.

Resumen de los Prámetros	Subred neuronal 1	Subred neuronal 2
Número de variables entrada/nodos	3/7	11/23
Número de nodos salidas	39	39
Número de capas	12	41
Número de neuronas en cada capa	1	1
Número de patrones de entrenamiento	40	5.585
Número de patrones de validación	-	1.861
Error cuadrático medio	$8,33 \cdot 10^{-5}$	0,083

Tabla 3. Parámetros de las subredes neuronales del modelo *Sierra2*

4. Conclusiones

Se ha desarrollado un modelo de red neuronal (*Sierra2*) para la selección de especies arbustivas, a partir de la información correspondiente a 39 especies arbustivas presente en la región mediterránea de Andalucía, y a la información del medio físico andaluz.

La RNA "quickpropagation en cascada" *Sierra2* es capaz de aprender de los patrones de distribución de las especies, estableciendo relaciones entre las variables de entrada y las aptitudes relativas de las especies, que en algún caso pueden ser difícilmente apreciables.

El modelo de evaluación desarrollado *Sierra2* es de una elevada sensibilidad. Ya que cambios en algunos de los niveles de generalización de entrada, puede producir cambios en la lista de salida de especies o en el menor de los casos en el porcentaje de idoneidad.

La división de los patrones en datos de entrenamiento y datos de test ha permitido realizar una validación externa del modelo, en una muestra intrínsecamente diferente de la inicial. La gran capacidad de predicción *Sierra2* se justifica por el bajo error que comete la red en la validación de patrones. Esto indica, que para escenarios para la cual no ha sido entrenada la red, se conseguirá un listado de especies aptas con una alta fiabilidad.

La elevada generalización del modelo *Sierra2* permite su aplicación en cualquier lugar de la cuenca mediterránea. Por lo tanto, se puede considerar que tiene un amplio espectro de aplicación. Este modelo *Sierra2* puede ser aplicado para hacer predicciones futuras de cambio climático, tanto de cambios de temperatura como de precipitación, mostrando así facilidad para formular escenarios hipotéticos.

Los resultados de aptitud arbustiva, obtenidos mediante la aplicación del modelo *Sierra2*, pueden ser fácilmente espacializados haciendo uso de un sistema de información geográfica.

Agradecimientos

La presente investigación se ha llevado a cabo gracias a la financiación recibida por el Instituto Agronómico Mediterráneo de Zaragoza (IAMZ), como beca de Master en Ciencias para Norberto Heredia (primer autor del presente trabajo).

Referencias

- [1]Anaya-Romero, M., 2004. Modelo de distribución potencial de usos forestales

- basado en parámetros edáficos, geomorfológicos, climáticos y topográficos. Tesis Doctoral. Universidad de Sevilla, España.
- [2]Benavente Pérez, L. (2004). Árboles y Arboledas Singulares de Andalucía, Málaga. Consejería de medio ambiente. Junta de Andalucía.
- [3]Bishop, C.M. (1995). Neural networks for pattern recognition. New York: Oxford University Press.
- [4]Blanca, G. (2001). Flora amenazada y endémica de Sierra Nevada. Universidad de Granada. Consejería de Medio Ambiente. Junta de Andalucía. Editorial Universidad de Granada.
- [5]Blank, T., & Brown, S. D. (1993). Nonlinear multivariate mapping of chemical data using feed-forward neural networks. *Analytical Chemistry* 65:3081-3089.
- [6]Bullock, P., Jones, R.J.A., & Montanarella, L. (ed.). (1999) Soil Resources of Europe. The European Soil Bureau, Joint Research Centre I-21020 ISPRA. Italy.
- [7]De la Rosa, D., Mayol, F., Moreno, J. A., Bonsón, T., & Lozano, S. (1999). An expert system/neural network model (ImpelERO) for evaluating agricultural soil erosion in Andalucia region, southern Spain. *Agriculture, Ecosystems and Environment*, 73:211-226.
- [8]De la Rosa, D., F. Mayol, E. Diaz-Pereira, M. Fernandez & D. de la Rosa, Jr. (2004). A land evaluation decision support system (MicroLEIS DSS) for agricultural soil protection: With special reference to the Mediterranean region. *Environmental Modelling & Software*, 19:929-942.
- [9]Eom, S. B., Le, S. M., Kim, E. B. y Somarajan, C. (1998) A Survey of Decision Support System Applications. *Journal of Operational Research* 49:109-120.
- [10]Escalera, J. (1993). Espacios Naturales-Espacios Sociales: Por un tratamiento integral del patrimonio ecológico-cultural de Andalucía. El caso del Parque Natural de la Sierra de Aracena y Picos de Aroche (Huelva). En: González, I. (coord.). Parques naturales andaluces. Conservación y cultura. Agencia de Medio Ambiente (Junta de Andalucía). Sevilla.
- [11]Foody, G. M. (1996). Land cover classification by an artificial neural network with ancillary information. *International Journal of Geographical Information Systems*, 9:527-542.
- [12]Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron) – A review of applications in the atmospheric sciences. *Atmospheric Environment*, 32:2627-2636
- [13]Halpern, C. B., & Spies, T. A. (1995). Plant species diversity in natural and managed forests of the Pacific Northwest. *Ecological Applications*, 5:913–934.
- [14]Heredia, N. (2007). Desarrollo de un modelo de evaluación de tierras en red neuronal (*Sierra2*) para la selección de especies arbustivas en la reforestación de zonas mediterráneas. Un nuevo componente del sistema MicroLEIS. Thesis. Master of Science. Centro Internacional de Altos Estudios Agronómicos Mediterráneos (CIHEAM). Zaragoza, España.
- [15]Kimmings, J.P. (1987). Forest Ecology. Macmillan. New York. EEUU.
- [16]Lorite, J., & Valle, F. (1999). Bases para la elaboración de modelos de gestión de la vegetación en la Sierra Nevada Almeriense. Encuentro medioambiental Almeriense: en busca de soluciones.
- [17]Pérez Latorre, A. V., Cabezudo, B., Galán de Mera, A., & Carrión, J. (2001). El papel de las gimnospermas en la vegetación forestal de Andalucía. *Medioambiente*, 38:30-33.
- [18]Ripley, B.D. (1996). Pattern recognition and neural networks. Cambridge: Cambridge University Press.
- [19]Rivas-Martínez, S. (1987). Memoria del mapa de series de vegetación de España. ICONA, Serie Técnica. Servicio de Publicaciones del Ministerio de Agricultura Pesca y Alimentación. Madrid.
- [20]Sauter, V. L. (1997). Decision support systems: an applied managerial approach. New York, John Wiley.
- [21]SNNS, 2004. Software Stuttgart Neural Network Simulator, Versión 4.2. Germany.
- [22]Tegelmark, D. O. (1998). Site factors as multivariate predictors of natural regeneration in Scots pine forests. *Forest Ecology and Management*, 109:231-239.

Índice de autores

- Luis Felipe Acebes, 9
José Acedo, 9
Rosa M. Aguilar, 85
Manuel Alfonseca, 95
Carlos Alonso, 25
Raúl Alves, 9
M. Anaya-Romero, 137
Joaquim Armengol, 17
Pedro J. Baquero, 85
J. Bondia, 121
Javier Bonilla, 129
Anibal Bregón, 25
Sònia Busquets, 43
R. Calm, 121
Eduardo F. Camacho, 113
Norberto Cañas de Paz, 33
Iván Castilla, 85
Sandra M. Castillo, 17
Joaquín Cerdá Boluda, 77
Ricardo José Colom Palero, 77
Luis Conde, 95
E. Díaz-Pereira, 137
Rafael Gadea Gironés, 77
Luis Galán, 95
Diego García, 25
W. García-Gabín, 121
Yurena García-Hevia, 85
M. García-Jaramillo, 121
Esteban R. Gelso, 17
A. Guasch, 67
D. Guimaran, 67
N. Heredia, 137
Emilio Jiménez Macías, 59
Néstor Ferrando Jódar, 77
Juan de Lara, 95
F. Mayol, 137
Rogelio Mazaeda, 9
Alejandro Merino, 9
Roberto Muñoz, 85
Mercedes E. Narciso Farias, 103
Luis Palacín, 9
Francisco Pérez, 95
Mercedes Pérez de la Parte, 59
B. Pérez-Lancho, 3
Miquel À. Piera, i Eroles, 43, 51, 67, 103
Óscar Prieto, 25
Belarmino Pulido, 25
Armando Ramírez, 109
J. J. Ramos, 67
Miguel A. Rida, 103
Francisco Rodríguez, 129
D. de la Rosa, 137
Almudena Rueda, 9
E. Sanz, 3
C. Tarín, 121
Gracián Triviño Barros, 33
P. Vega, 3
Marta Vega Ollora, 59
J. Vehí, 121
Laia Villalta, 51
Luís José Yebra, 129
Ascensión Zafra-Cabeza, 113



SERVICIOS EDITORIALES

- ❑ Especialistas en maquetación y tratamiento de textos desde los archivos originales de **L^AT_EX** y **TEX**.
- ❑ Elaboración de figuras e ilustraciones con programas de diseño gráfico compatibles con **L^AT_EX** y **TEX**.
- ❑ Lectura y revisión de estilo de libros, revistas, publicaciones y artículos científicos, tesis doctorales, páginas web...
- ❑ Realizamos la edición completa de todo tipo de publicaciones científicas.
- ❑ Especialistas en publicación de Actas de Congresos.
- ❑ Si tiene algún proyecto científico, podemos gestionar su publicación en las mejores editoriales.

Tel.: 618 96 45 50

informacion@librotex.com

concepcion.fernandez@librotex.com

<http://www.librotex.com>

Le invitamos a colaborar con



¿Tiene algún proyecto...

- ... editorial que se adapte a los planes actuales de estudio universitarios?
- ... editorial para desarrollar un libro de texto universitario enfocado a los nuevos planes de estudio?
- ... para desarrollar contenidos educativos de *e-learning* para la universidad?
- ... educativo dentro de su área de conocimiento?

¿Quiere ser uno de nuestros colaboradores en la evaluación de libros en inglés, proyectos originales o contenidos electrónicos?

Sí es así, le invitamos a colaborar con el grupo editorial THOMSON para entre todos conseguir publicar los proyectos editoriales mejor adaptados a las necesidades educativas de profesores y estudiantes universitarios.

¿Qué puede ofrecerle THOMSON?

- Evaluar cualquier proyecto editorial en un plazo breve de tiempo.
- Colaborar con una de las editoriales más importantes del mundo a nivel universitario.
- Nuestra amplia experiencia editorial en la publicación de libros científicos y técnicos.
- Nuestros amplios equipos de promoción y marketing al servicio de los libros de Thomson.
- La distribución de todos nuestros libros, tanto a nivel nacional, como en todos los países de habla hispana.
- Pertenecer al club de autores y colaboradores de Thomson.
- Posibilidad de traducir sus libros a otros idiomas como el portugués.

THOMSON PARANINFO

Magallanes, 25
28015 Madrid

TEL. 91-446-33-50, Fax. 91-445-62-18

E-mail: andres.otero@paraninfo.es

Direcciones Web: <http://www.paraninfo.es> y <http://www.thomsonlearning.com>