

Entorno de simulación en tiempo real para sistemas fisiológicos

L. García, M. J. López, J. Lorenzo, M. Prian, F. M. Verdulla

Departamento Ingeniería de Sistemas y Automática, Tecnología Electrónica y Electrónica

Universidad de Cádiz. España

{luis.garcia, manueljesus.lopez, jose.lorenzo, manuel.prian, francisco.verdulla}@uca.es

Resumen

En este trabajo se presenta un entorno para simulación y control de sistemas en tiempo real, que se puede utilizar para simulación de procesos fisiológicos. Incluye hardware para la generación (salidas) de señales eléctricas, así como para la inyección (entradas) de éstas. El entorno de simulación (EPESC) utiliza software libre (Linux/RTAI), emplea tarjetas de adquisición de datos y ordenadores tipo PC. Como ejemplos ilustrativos se presentan los resultados obtenidos con dos modelos matemáticos relacionados con la dinámica del sistema cardiovascular.

1. Introducción

Es conocido que la simulación por computador de sistemas dinámicos es de gran utilidad para el análisis de procesos complejos y para el diseño de sistemas de control [1], [2]. Los sistemas no lineales, y como caso particular los fisiológicos, pueden presentar una riqueza y complejidad de comportamientos muy diversos [3], [4], [5], [6], [7], [8], [9], [10]. Por ello, además de la propia simulación, es conveniente que se disponga de herramientas software que implementen algoritmos para análisis de señales y series temporales [11], [12].

Como resultado de la colaboración entre el grupo de Ingeniería Biomédica y el grupo de Automática (GAPGIS) de la universidad de Cádiz, se ha planteado la conveniencia de disponer de un entorno de simulación en tiempo real con hardware en el lazo (*“hardware in the loop simulation”*, HILS) para simulación de sistemas fisiológicos; junto con un software para análisis de series temporales e identificación de sistemas. La característica HIL supone que la conexión entre los equipos se hace a través de tarjetas I/O de adquisición de datos [13,14,15], de forma que un equipo tipo PC (PC1) hace de simulador del proceso, y otro equipo tipo PC (PC2) hace de controlador (en su caso), ambos funcionando en tiempo real [16]. Un tercer equipo está conectado a ambos mediante un switch ethernet de 8 puertos, lo que permite aumentar el número de equipos/simuladores conectados y realizar simulación distribuida en varios equipos. Este equipo (PC3) sirve de interfase para el usuario/operador, se utiliza para la monitorización del proceso simulado, y en su caso, para modificar los parámetros del proceso y/o el ajuste del controlador usado (ControlAvH Tune [20] en figura 1).

Para una realización previa y testeo del modelo matemático a usar se emplea el software Matlab/Simulink [15]. Una vez verificado un funcionamiento adecuado del modelo, se pasa a la codificación en lenguaje C/C++ para su uso en el simulador en tiempo real (PC1) y/o en el controlador (PC2) respectivamente [16]. El entorno de simulación es flexible y ampliable. Los datos procedentes de la simulación se pueden procesar a posteriori (*off-line*) a través de una librería de funciones para Matlab que se está elaborando, para lo que se podría utilizar el equipo PC3 u otro PC (PC4) o portátil externo conectado a la red local o vía internet. También se considera, como alternativa complementaria, para comparar y analizar prestaciones, la opción de utilizar como simulador del proceso el sistema DS1104 R&D *Controller Board* de dSPACE [14].

El resto del trabajo se organiza como sigue: en el apartado dos se presenta EPESC, continuando con los apartados tercero y cuarto donde se presentan modelos matemáticos utilizados para caracterizar la variabilidad de la dinámica del sistema cardiovascular. El primero está basado en osciladores del tipo de Van der Pol acoplados que reproducen, al menos cualitativamente, el ECG humano. En el segundo, se utiliza un enfoque basado en el modelo de Windkessel y sistema baroreflex. Los resultados de simulación se presentan en el apartado quinto, y finalmente se resumen las conclusiones.

2. Entorno de simulación

El entorno de simulación en tiempo real (EPESC) que se ha empleado para realizar las simulaciones se ha desarrollado en lenguaje C sobre el sistema operativo Linux con extensión para tiempo real estricto RTAI. Se ha empleado software libre a fin de ahorrar el coste elevado de otros sistemas operativos para tiempo real. EPESC es un Entorno de Pruebas y Evaluación de Sistemas de Control que hemos desarrollado en el ámbito del grupo GAPGIS de la Universidad de Cádiz, con la finalidad de servir como banco de pruebas de algoritmos de control y simulación de procesos.

Para alcanzar el máximo grado de aproximación y realismo, los algoritmos se implementan en tiempo real estricto y la entrada/salida de señales se canaliza con tarjetas de adquisición de datos, [1], [2], [16].

EPESC está desarrollado usando COTS (*commercial off-the-shelf*) y software gratuito. La configuración básica

está constituida por dos ordenadores tipo PC con Sistema Operativo Linux (Distribución OpenSuse 10.2) y tarjetas de adquisición de datos. Un ordenador incorpora el controlador (PC2) y el otro el proceso simulado (PC1). Puede simular procesos híbridos, compuestos de partes de tiempo continuo (con elementos de retardo), así como procesos de tiempo discreto.

El tiempo real estricto se consigue añadiendo al núcleo del Sistema Operativo el módulo RTAI (*Real Time Application Interface*) desarrollado por el Instituto Politécnico de Milán, que soporta procesos en tiempo real estricto, es código abierto y está ampliamente utilizado por la comunidad científica en centros de investigación europeos [1], [2], [16].

Se usan tarjetas de adquisición de datos (NI PCI-6014) y los drivers correspondientes a la librería NI-DAQ de National Instruments [13], proporcionados por el proyecto COMEDI (*Control and Measurements Device Interface*) [1], [2], [16], que ha sido desarrollada por una comunidad de software libre. Por medio de un conjunto de drivers y una librería de funciones (COMEDILIB), COMEDI permite la programación, calibrado y operación con más de un centenar de tarjetas de adquisición de datos de diferentes fabricantes.

En el desarrollo realizado aquí, se ha considerado que para la simulación y control de procesos en tiempo real se deben satisfacer los siguientes requisitos básicos: 1) ejecución periódica de tareas en tiempo real, 2) proporcionar una interfase con procesos externos y hardware asociado, 3) capacidad de incorporar fácilmente modelos desarrollados por medio de simulación off-line; y 4) aprovechar el máximo número de paquetes de software libre.

El controlador en tiempo real ha sido desarrollado para emular las funcionalidades de un controlador de propósito general; de forma que puede ser fácilmente adaptado para interactuar con simuladores de procesos. Gracias a la utilización de hardware en el lazo de control (tarjetas de adquisición) y los drivers de COMEDI, el controlador puede actuar, sin distinción, sobre un sistema físico real, o un modelo simulado del mismo.

En la Figura 1 se muestra un esquema de bloques con la arquitectura del software de EPESC. Los componentes o módulos se subdividen en dos grupos: los de tiempo real (RT) y los que no lo son (NRT). Estos dos grupos de módulos se comunican mediante colas FIFO (*first input - first output*). Los componentes de tiempo real se ejecutan bajo Linux/RTAI como un módulo de Linux (en el espacio del núcleo), el resto del software se ejecuta bajo Linux como un proceso (en el espacio de usuarios) [1, 2,16].

Tanto en el simulador de la planta o proceso como en el controlador, los módulos de tiempo real se implementan por medio de dos procesos ligeros o RT-threads. El proceso MONITOR (ver Figura 1) se comunica mediante FIFO con el resto del sistema y comanda al RT-thread PLANT (o CONTROLLER) según los mensajes recibidos del operador local y del resto del sistema mediante red

Ethernet. Las señales E/S de tarjetas de adquisición de datos se indican mediante la leyenda “*wired connections*” en la Figura 1.

La información que circula por el sistema se gestiona en el proceso DISPLAY, que almacena la información en estructuras de datos y la habilita para los servicios de red. Existe un servicio de red para la comunicación con cada subsistema, cada servicio está gestionado a nivel EPESC por un *thread*. Tanto el controlador como la planta utilizan dos servicios cada uno para comunicarse con un equipo que se utilice para monitorizar la evolución dinámica del sistema, análisis y diseño de controlador (ControlAvH-Tune en equipo PC3 o portátil), uno para comandos (TCP) y otro para datos o data-logging (UDP). También existen dos servicios para lectura/escritura (UDP) con Matlab/Simulink (HOST COMPUTER en equipo PC4 o portátil).

Como ejemplos ilustrativos para probar el simulador realizado en EPESC se utilizan los modelos matemáticos de la dinámica del corazón y del sistema circulatorio que se indican en los apartados siguientes.

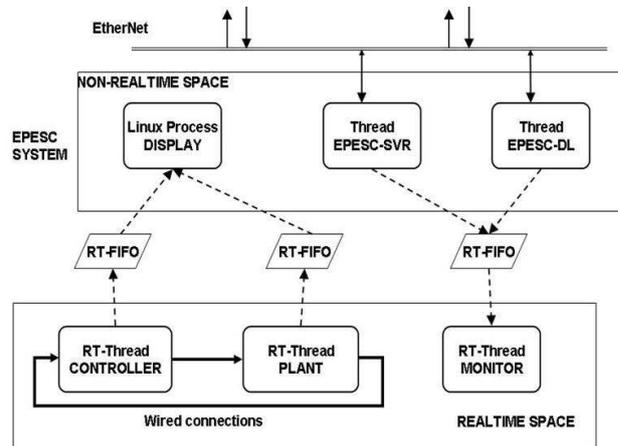


Figura 1. Arquitectura del software de EPESC

3. Modelado matemático basado en la ecuación diferencial de VdP

La ecuación diferencial de Van der Pol (VdP)

$$\ddot{y} + a(1 - by^2)\dot{y} + cy = f(t)$$

se corresponde a un circuito electrónico de relajación, y se ha usado especialmente en modelos teóricos para describir la dinámica del corazón asociada con frecuencia cardiaca. Esto se debe a que las características cualitativas de un oscilador de VdP se caracterizan por tener una gran similitud con el comportamiento observado en los potenciales de actuación del corazón [3,4,5]. Así mismo, la ecuación de VdP también se utiliza habitualmente para el modelado fenomenológico de otros procesos biológicos que se dan en la naturaleza, dado que sirve para modelar comportamientos dinámicos no lineales tales como ciclos límite, bifurcaciones, sincronización y caos [11,12,17,18].

En la práctica, se utilizan versiones modificadas de la ecuación de VdP para simular el comportamiento cardiaco, donde éste se modela mediante dos osciladores acoplados [7,19,21], o mediante tres osciladores acoplados con retardos de tiempo asociados a la transmisión de las señales [6].

Como la idea de la plataforma EPESC es que sirva como herramienta de simulación en tiempo real flexible para diversos sistemas, se ha propuesto la incorporación de un modelo matemático basado en dos osciladores de VdP acoplados [7, 18, 19, 21],

$$\begin{aligned}\dot{x}_1 &= \frac{1}{C_{SA}}x_2 \\ \dot{x}_2 &= -\frac{1}{L_{SA}}[x_1 + f_1(x_2) + R(x_2 + x_4)] \\ \dot{x}_3 &= \frac{1}{C_{AV}}x_4 \\ \dot{x}_4 &= -\frac{1}{L_{AV}}[x_3 + f_2(x_4) + R(x_2 + x_4)]\end{aligned}$$

donde f_1 y f_2 son funciones no lineales.

También se propone para EPESC la incorporación del modelo basado en tres osciladores de VdP acoplados [6],

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a_{SA}x_2(x_1 - w_{SA1})(x_1 - w_{SA2}) \\ &\quad -x_1(x_1 + d_{SA})(x_1 + e_{SA}) + \rho_{SA} \sin(\omega_{SA}t) \\ &\quad + K_{SA-AV}(x_1 - x_3^{T_{dSA-AV}}) \\ &\quad + K_{SA-HP}(x_1 - x_5^{T_{dSA-HP}}) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -a_{AV}x_4(x_3 - w_{AV1})(x_3 - w_{AV2}) \\ &\quad -x_3(x_3 + d_{AV})(x_3 + e_{AV}) + \rho_{AV} \sin(\omega_{AV}t) \\ &\quad + K_{AV-SA}(x_3 - x_1^{T_{dAV-SA}}) \\ &\quad + K_{AV-HP}(x_3 - x_5^{T_{dAV-HP}}) \\ \dot{x}_5 &= x_6 \\ \dot{x}_6 &= -a_{HP}x_6(x_5 - w_{HP1})(x_5 - w_{HP2}) \\ &\quad -x_5(x_5 + d_{HP})(x_5 + e_{HP}) + \rho_{HP} \sin(\omega_{HP}t) \\ &\quad + K_{HP-SA}(x_5 - x_1^{T_{dHP-SA}}) \\ &\quad + K_{HP-SA}(x_5 - x_3^{T_{dHP-AV}})\end{aligned}$$

En este caso, se emplea la notación $x_i^{Td} = x_i(t-Td)$ para indicar un retardo Td . Los valores de los parámetros de los modelos matemáticos (1) y (2) se han tomado de sus respectivas referencias [6,7,19,21]. La señal de un electrocardiograma (ECG) se construye a partir de

$$ECG = \alpha_0 + \alpha_1 x_1 + \alpha_3 x_3 + \alpha_5 x_5$$

4. Simulación basada en modelo de Windkessel

Para simular la dinámica del sistema cardiovascular se utiliza de forma habitual un modelo de Windkessel de tres elementos, complementado con el sistema de control baroreflex [3, 4, 8, 9], al que se le incorporan retardos de tiempo para transmisión de las señales. El modelo de Windkessel está dado por la siguiente ecuación diferencial, que relaciona la presión de la sangre arterial P (mm Hg) con el caudal de salida cardiaco Q (ml/s)

$$R_p C_a \dot{P} + P = R_c R_a C_a \dot{Q} + (R_a + R_c)Q$$

Las relaciones en régimen estacionario del sistema baroreflex para el periodo cardiaco T y el volumen V en función de la presión P están dadas respectivamente por

$$\begin{aligned}T(P) &= T_{min} + \frac{T_{max} - T_{min}}{1 + \gamma e^{-\alpha P/P_e}} \\ V(P) &= \frac{V_{max}}{1 + \beta((P/P_v) - 1)^{-k}}\end{aligned}$$

Como valores para los parámetros se toman los dados en las publicaciones correspondientes [3,4,8,9]. Para modelar la dinámica transitoria del sistema baroreflex se utiliza un filtro pasa-bajas de primer o segundo orden con un retardo de tiempo efectivo para tener en cuenta la dinámica de alta frecuencia no modelada y los propios retardos de transporte en el sistema.

5. Resultados de simulación

En este apartado se muestran, a modo de ejemplos ilustrativos, los resultados de simulación obtenidos utilizando los modelos matemáticos de los apartados anteriores.

Mediante el simulador (PC1) de EPESC se simula el proceso, y se genera en tiempo real un comportamiento dinámico, correspondiente a una situación estacionaria constante (punto fijo o de equilibrio), un comportamiento periódico (ciclo límite) o un comportamiento caótico, para unos determinados valores de las condiciones iniciales y parámetros del modelo. El controlador (PC2) de EPESC se utiliza, en su caso, para aplicar un algoritmo de control y conseguir estabilizar al sistema en un punto de equilibrio, o en un comportamiento periódico, o en su caso conseguir la caotificación del sistema si así se requiere. En este trabajo sólo se aborda el problema de la simulación, dejando para una fase posterior el desarrollo y pruebas en tiempo real del algoritmo de control.

Una situación peculiar corresponde a un comportamiento de aspecto caótico. El interés consiste en la detección y aplicación de una técnica de control que saque del caos al sistema y lo estabilice en una órbita periódica adecuada desde el punto de vista clínico.

En la figura 2 se muestra la señal del ECG (escalada para su representación) obtenida utilizando las ecuaciones del modelo basado en tres osciladores acoplados. También se muestran la evolución de las variables de estado indicadas (x_1, x_3, x_5). Modificando los parámetros se pueden conseguir diferentes tipos de comportamiento patológicos: flutter ventricular, bradicardia sinusal y taquicardia ventricular [6,7,8,19,21]. Por razones de espacio no se muestran las gráficas correspondientes a los casos indicados.

Como ejemplo de resultado obtenido con el modelo del apartado tres (basado en modelo de Windkessel y baroreflex), en la figura 3 se muestra la evolución de la presión arterial y la variabilidad de la frecuencia cardiaca.

Dependiendo del retardo considerado en la transmisión de señales en el sistema, se producen diferentes situaciones,

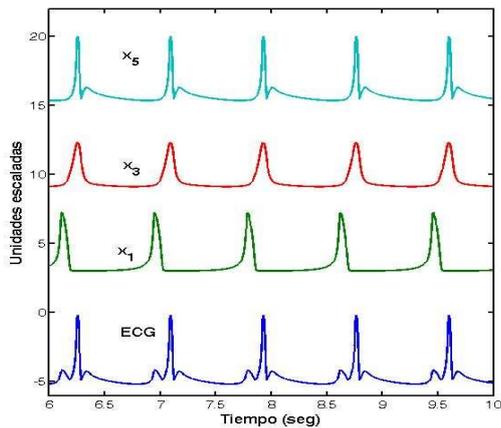


Figura 2. Simulación de tres osciladores acoplados.

que van desde la situación normal donde se tiene un estado estacionario fijo, hasta la situación en la que la evolución de la señal tiene un aspecto caótico (figura 3).

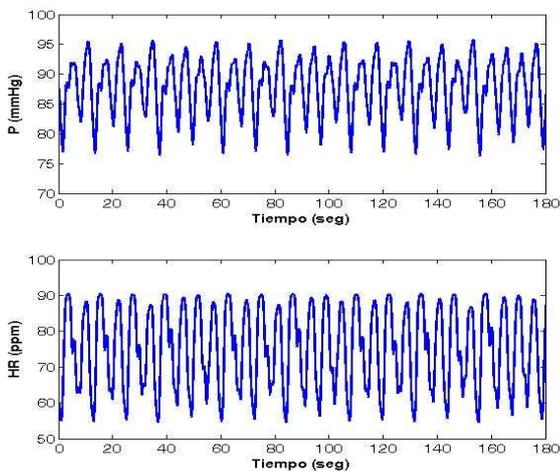


Figura 3. Presión (P) y Frecuencia Cardiaca (HR)

Como se muestra en la figura 3, para una condición inicial dada y unos valores de los parámetros utilizados [3,4,8,9] se obtienen unas señales cuya evolución no parece ser periódica. Otro aspecto a considerar trata de analizar si es mera apariencia debido al ruido de media, o si al contrario es que son señales de naturaleza caótica. Para ello, se está en la actualidad elaborando una toolbox realizada en Matlab para análisis de series temporales [11,12].

6. Conclusiones

Mediante el sistema EPESC desarrollado por el grupo GAPSIS de la UCA, se dispone de un entorno para simulación de procesos fisiológicos en tiempo real, que incorpora hardware para generar señales eléctricas E/S desde y hacia el proceso. El sistema es flexible y escalable, por lo que se puede utilizar para simulación de procesos de naturaleza muy variada.

Agradecimientos

Este trabajo ha sido soportado parcialmente por el proyecto de investigación de la UCA PR/2008-073.

Referencias

- [1] Gazi V., M. L. Moore, K. M. Passino, W. P. Schackelford, F. M. Proctor, J. S. Albus (2001). The RCS Handbook. John Wiley & Sons, Inc.
- [2] Samad T, G. Balas. Software-Enabled Control. IEEE Press (2003).
- [3] Ottesen J. T., M.S. Olufsen, J. K. Larsen. Applied Mathematical Models in Human Physiology. SIAM (2004).
- [4] Khoo M. C. K. Physiological Control Systems. Analysis, Simulation and Estimation (2000).
- [5] Batzel J. J., F. Kappel, D. Schneditz, H. T. Tran. Cardiovascular and Respiratory Systems. SIAM (2007).
- [6] S. R. F. S. M. Gois, M. A. Savi (2009). An analysis of heart rhythm dynamics using a three-coupled oscillator model. Chaos, Sol. & Fractals, Vol. 41, Issue 5, pp 2553-2565.
- [7] Brandt M. E., G. Wang, H-T Shih (2003). Feedback control of a nonlinear dual-oscillator heartbeat model. Bifurcation Control, pp 265-273. Springer.
- [8] Cavalcanti S., E. Belardinelli (1996). Modeling of cardiovascular variability using a differential delay equation. IEEE Transactions on Biomedical Engineering, Vol. 43, No. 10, pp 982-989.
- [9] Kubota T. J. Alexander, R. Itaya, K. Todaza, M. Sugimachi, K. Sunagawa, Y. Nose, A. Takeshita. Dynamic effects of carotid sinus baroflex on ventriculoarterial coupling studied in anesthetized dogs. Circres Res (1992).
- [10] Wessel N., H. Malberg, R. Bauernshmitt, J. Kurths (2007). Nonlinear methods of cardiovascular physics and their clinical applicability. International Journal of Bifurcation and Chaos. Vol. 17 No. 10 pp 3325-3371.
- [11] Small M. (2005). Applied nonlinear time series analysis. Applications in physics, physiology and finance. World Scientific.
- [12] Kantz H., T. Schreiber (2004). Nonlinear time series analysis. Cambridge University Press.
- [13] National Instruments, www.ni.com
- [14] dSPACE, www.dspace.com
- [15] Matlab www.mathworks.com
- [16] García L., M. J. López, J. Lorenzo (2006). Hard Real Time Based on Linux/RTAI for Plant Simulation and Control Systems Evaluation. WSEAS Transactions on Systems and Control, Issue 2, Vol. 1, Dec. 2006, ISSN 1991-8763.
- [17] Sprott J. C. (2006). Chaos and Time-Series Analysis. Oxford University Press. ISBN 0 19 850840 9.
- [18] W. Perruquetti, J-P Barbot. (2006). Chaos in Automatic Control. CRC Taylor & Francis. ISBN: 0-8247-2653-7.
- [19] Grudzinski K., J. J. Zebrowski (2004). Modeling cardiac pacemakers with relaxation oscillators. Physica A 2004;336:153-162.
- [20] Lorenzo J., M.J. Lopez (2006). ControlAvH Tune: Software environment for robust H controller tuning and real time control. IEEE MELECON, pp 417-420, Malaga (Spain).
- [21] Santos AM, Lopes SR, Viana RL (2004). Rhythm synchronization and chaotic modulation of coupled Van der Pol oscillators for heartbeat. Physica A, 338, 335-355.