

INTEGRACIÓN POR CUANTIFICACIÓN DE SISTEMAS STIFF. PARTE II: IMPLEMENTACIÓN

Gustavo Migoni[†], Ernesto Kofman[†] y François Cellier^{††}

[†]*Laboratorio de Sistemas Dinámicos. FCEIA - UNR -
CONICET. Riobamba 245 bis - (2000) Rosario*
^{††}*Institute of Computational Science - ETH Zürich.
CH-8092 Zürich. Switzerland.*

Abstract: En esta segunda parte, se abordan aspectos prácticos y ejemplos de aplicación del método de integración BQSS presentado. Se desarrolla un algoritmo para la implementación explícita del método y se propone un modelo de eventos discretos que lleva a cabo la integración con esta aproximación. Se discuten también algunos aspectos de la programación del método y se presentan resultados de simulación sobre dos sistemas stiff.

Keywords: Stiff System Simulation, Quantization Based Integration, DEVS

1. INTRODUCCIÓN

Para simular un sistema continuo en una computadora digital algo debe discretizarse, tradicionalmente la variable temporal. Casi todos los métodos de integración de Ecuaciones Diferenciales Ordinarias en la literatura se basan en polinomios de extrapolación (algoritmos explícitos) o de interpolación (algoritmos implícitos) sobre los valores pasados de los estados y sus derivadas. La mayor parte de los métodos son sincrónicos, es decir, utilizan un único reloj y actualizan todas las variables de estado a la vez. Además, la mayor parte de los métodos controlan el paso de integración mediante un esquema heurístico que busca balancear los requerimientos de estabilidad y precisión con los de eficiencia (Cellier and Kofman, 2006).

Sin embargo, la discretización temporal no es la única alternativa. Hay otro enfoque basado en la cuantificación de las variables de estado que mantiene el tiempo continuo. Los métodos de Quantized State Systems (QSS) pertenecen a esta clase de algoritmos. Los integradores QSS funcionan de manera asíncrona, esto es, cada variable

de estado se actualiza independientemente de las otras y el control de paso es una característica intrínseca de los métodos (y por lo tanto no hay necesidad de ningún esquema heurístico de control de paso).

Se demuestra que los métodos de QSS son numéricamente estables y que realizan un control de precisión excelente (de hecho, tienen una cota de error global calculable para sistemas lineales).

Sin embargo, los métodos de QSS anteriores a este trabajo (Kofman and Junco, 2001; Kofman, 2002, 2006) eran explícitos y por lo tanto no eran apropiados para simular sistemas stiff. Como cualquier método explícito, permanecían numéricamente estables al costo de utilizar pasos de integración extremadamente pequeños.

A modo de solución a este problema, en la primera parte de este trabajo (Kofman *et al.*, 2006) se introdujo un nuevo método de QSS basado en principios de integración implícita que permite integrar sistemas stiff de manera eficiente, conservando las propiedades de estabilidad y cota de error de los métodos explícitos de QSS. En esta

segunda parte se tratan los aspectos de implementación de este método, denominado BQSS.

El trabajo está organizado de la siguiente forma. Tras un breve repaso sobre la implementación de QSS a través de modelos DEVS (Sección 2), en la Sección 3 se presenta un algoritmo que permite obtener de manera explícita las variables cuantificadas. En base a este algoritmo y a los principios de simulación de QSS, se propone un modelo DEVS que lleva a cabo la integración con BQSS y luego se discuten algunos aspectos de la programación del método en un entorno de software. Finalmente, en la Sección 4 se presentan resultados de simulación sobre dos sistemas stiff.

2. DEVS Y MÉTODOS DE CUANTIFICACIÓN

En esta sección presentamos brevemente el formalismo DEVS y los principios básicos de los métodos de integración por cuantificación.

2.1 Formalismo DEVS

Un modelo DEVS (Zeigler *et al.*, 2000) procesa una trayectoria de eventos de entrada y , en base a la misma y a sus propias condiciones iniciales, produce una trayectoria de eventos de salida.

El comportamiento de un modelo atómico DEVS se define formalmente mediante la siguiente estructura:

$$M = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta) \quad (1)$$

donde

- X es el conjunto de eventos de entrada, es decir, el conjunto de todos los valores que puede adoptar un evento.
- Y es el conjunto de eventos de salida.
- S es el conjunto de valores de estado.
- $\delta_{\text{int}}, \delta_{\text{ext}}, \lambda$ y ta son las funciones que definen la dinámica del sistema.

Cada posible estado s ($s \in S$) tiene asociado un tiempo de avance calculado a través de la función de avance de tiempo $ta(s)$ ($ta(s) \rightarrow \mathbb{R}_0^+$). El tiempo de avance es un número no negativo que indica cuanto tiempo permanece el sistema en un estado determinado en ausencia de eventos de entrada.

Luego, si en el tiempo t_1 el estado s toma el valor s_1 , tras $ta(s_1)$ unidades de tiempo (o sea, en el tiempo $t_1 + ta(s_1)$) el sistema realiza una transición interna yendo a un nuevo estado s_2 . El nuevo estado se calcula como $s_2 = \delta_{\text{int}}(s_1)$. La función δ_{int} ($\delta_{\text{int}} : S \rightarrow S$) es la función de transición interna.

Cuando el estado pasa de s_1 a s_2 se produce un evento de salida con el valor $y_1 = \lambda(s_1)$. La función λ ($\lambda : S \rightarrow Y$) se denomina función de salida. Las funciones ta , δ_{int} y λ definen el comportamiento autónomo de un modelo DEVS.

Cuando llega un evento de entrada, el estado cambia instantáneamente. El nuevo estado no sólo depende del evento de entrada sino también del valor del estado anterior y del tiempo transcurrido desde la última transición. Si el sistema llega al estado s_3 en t_3 y luego llega un evento de entrada en el instante $t_3 + e$ con un valor x_1 , el nuevo estado se calcula según $s_4 = \delta_{\text{ext}}(s_3, e, x_1)$ (notar que $e < ta(s_3)$). En este caso se dice que el sistema realiza una transición externa. La función δ_{ext} ($\delta_{\text{ext}} : S \times \mathbb{R}_0^+ \times X \rightarrow S$) se llama función de transición externa. Durante una transición externa no se produce ningún evento de salida.

Los modelos DEVS pueden acoplarse, y el resultado del acoplamiento define un modelo DEVS atómico equivalente.

2.2 DEVS y QSS

Dada una ecuación de la forma

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2)$$

con $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, el método de QSS (Kofman and Junco, 2001; Kofman *et al.*, 2006) lo aproxima por

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) \quad (3)$$

donde las componentes q_i y x_i están relacionadas a través de una función de cuantificación con histéresis. En consecuencia, las $q_i(t)$ son seccionalmente constante.

Cada componente de (3) puede pensarse como el acoplamiento de dos subsistemas elementales. Uno estático

$$d_i(t) = f_i(q_1, \dots, q_n, u_1, \dots, u_m) \quad (4)$$

y otro dinámico

$$q_i(t) = Q_i(x_i(\cdot)) = Q_i\left(\int d_i(\tau) d\tau\right) \quad (5)$$

donde q_i es la función de cuantificación con histéresis (no es una función del valor instantáneo de x_i sino un funcional sobre la trayectoria $x_i(\cdot)$).

Dado que las componentes $u_i(t)$ y $q_i(t)$ son seccionalmente constantes, la salida del subsistema (4), o sea $d_i(t)$, será seccionalmente constante. De esta manera, ambos subsistemas tienen trayectorias de entrada y salida seccionalmente constantes y estas pueden ser representadas por secuencias de eventos.

Así, los subsistemas (4) y (5) definen una relación entre sus secuencias de eventos de entrada y salida. En consecuencia, existen modelos DEVS

equivalentes para estos subsistemas, denominados *funciones estáticas* e *integradores cuantificados* respectivamente (Kofman and Junco, 2001).

3. IMPLEMENTACIÓN DE BQSS

En esta sección, tras reintroducir la definición de BQSS presentada en la primera parte (Kofman *et al.*, 2006), desarrollaremos un algoritmo y los modelos DEVS correspondientes a la implementación del método.

3.1 Definición de BQSS

Dado el sistema (2), el método BQSS lo aproxima por:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) + \Delta \mathbf{f} \quad (6)$$

donde las componentes q_j de \mathbf{q} verifican:

$$q_j(t) \in \{\underline{q}_j(t), \bar{q}_j(t)\} \quad (7)$$

con¹

$$\underline{q}_j(t) = \begin{cases} \underline{q}_j(t^-) - \Delta Q_j & \text{si } x_j(t) - \underline{q}_j(t^-) \leq 0 \\ \underline{q}_j(t^-) + \Delta Q_j & \text{si } x_j(t) - \underline{q}_j(t^-) \geq \varepsilon_j + \Delta Q_j \\ \underline{q}_j(t^-) & \text{en otro caso} \end{cases} \quad (8)$$

$$\bar{q}_j(t) = \begin{cases} \bar{q}_j(t^-) + \Delta Q_j & \text{si } \bar{q}_j(t^-) - x_j(t) \leq 0 \\ \bar{q}_j(t^-) - \Delta Q_j & \text{si } \bar{q}_j(t^-) - x_j(t) \geq \varepsilon_j + \Delta Q_j \\ \bar{q}_j(t^-) & \text{en otro caso} \end{cases} \quad (9)$$

y además:

$$\begin{aligned} f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) &> 0 \\ \bigvee \\ \exists \hat{\mathbf{q}}^{(j)}(t) / f_j(\hat{\mathbf{q}}^{(j)}(t), \mathbf{u}(t)) &= 0 \end{aligned} \quad (10)$$

donde cada componente del vector $\hat{\mathbf{q}}^{(j)}(t)$ cumple:

$$|x_i(t) - \hat{q}_i^{(j)}(t)| < \Delta Q_i + \varepsilon_i \quad (11)$$

Por otro lado

$$\Delta f_j = \begin{cases} 0, & \text{si } f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j - x_j) > 0 \\ -f_j(\mathbf{q}(t), \mathbf{u}(t)), & \text{en otro caso} \end{cases} \quad (12)$$

Como en QSS, a ΔQ_j es llamado quantum y $\varepsilon_j < \Delta Q_j$ es el ancho de histéresis.

3.2 Obtención explícita de \mathbf{q} .

La diferencia principal de BQSS respecto de QSS es la forma de obtención de \mathbf{q} a partir de \mathbf{x} , ya que (10) implica cierta interdependencia entre los valores de las distintas componentes.

En QSS, los cambios en q_j se dan cuando x_j difiere de q_j en ΔQ_j . Similarmente, en BQSS los cambios se dan cuando x_j alcanza a q_j . Además, un cambio en q_j puede provocar cambios en otras variables cuantificadas debido a (10). Asimismo, cambios en alguna componente de \mathbf{u} pueden de la misma forma provocar cambios en algunas variables cuantificadas.

En definitiva, los eventos pueden ser provocados por cambios en las entradas o porque una variable de estado alcanzó la correspondiente variable cuantificada. Tras cualquiera de estos cambios, la principal dificultad aparenta ser encontrar el valor de \mathbf{q} correcto, que cumple con las Ecs.(7)–(11).

Veremos entonces un algoritmo que encuentra, de una forma muy simple y explícita, un valor de \mathbf{q} que satisface las mencionadas ecuaciones.

Definimos para esto \mathcal{D} como el conjunto de subíndices de las funciones f_i que van siendo evaluadas y \mathcal{A} como el conjunto de subíndices de los q_i que cambian de valor (ambos inicialmente vacíos).

Algoritmo 1.

- 1.a. Si cambia una entrada ($u_j(t) \neq u_j(t^-)$):
 - > Se agregan a \mathcal{D} los subíndices de las f_i que dependen explícitamente de u_j .
 - > Para cada $i \in \mathcal{D}$:
 - Definimos $\tilde{\mathbf{q}}^{(i)} \triangleq \mathbf{q}(t^-)$.
 - Si $f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) \cdot (\tilde{q}_i^{(i)} - x_i) < 0$
 - . Se agrega i a \mathcal{A}
 - . Definimos
- $$q_i(t) \triangleq \begin{cases} \bar{q}_i(t) & \text{si } f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) > 0 \\ \underline{q}_i(t) & \text{si } f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) < 0 \end{cases} \quad (13)$$
- Sino, $q_i(t) \triangleq q_i(t^-)$
- 1.b. Si x_i alcanza a q_i :
 - > Se agrega i a \mathcal{A} y \mathcal{D} .
 - > Se define $\tilde{\mathbf{q}}^{(i)} \triangleq \mathbf{q}(t^-)$.
 - > Se calcula $q_i(t)$ según la Ec.(13)
 2. Mientras $\mathcal{A} \neq \emptyset$
 - > Sea j el menor elemento de \mathcal{A} .
 - > Definimos \mathcal{B} como el conjunto de subíndices $i \notin \mathcal{D}$ y tales que f_i depende explícitamente de q_j .
 - > Para cada $i \in \mathcal{B}$:
 - Definimos $\tilde{\mathbf{q}}^{(i)}$ con componentes
- $$\tilde{q}_k^{(i)} = \begin{cases} q_k(t) & \text{si } k \in \mathcal{D} \\ q_k(t^-) & \text{si } k \notin \mathcal{D} \end{cases} \quad (14)$$
- Si $f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) \cdot (\tilde{q}_i^{(i)} - x_i) < 0$
 - . Se agrega i a \mathcal{A} .
 - . Calculamos $q_i(t)$ según la Ec.(13).
 - Sino, $q_i(t) \triangleq q_i(t^-)$.
- > Se agregan los elementos de \mathcal{B} al conjunto \mathcal{D} y se saca j de \mathcal{A} .
3. Para los $i \notin \mathcal{D}$, se deja $q_i(t) = q_i(t^-)$.

Como puede verse, este algoritmo siempre encuentra un valor de \mathbf{q} (luego demostraremos que cumple con la definición dada en la sección 3.1). Notar que ningún subíndice puede ser agregado más de una vez al conjunto \mathcal{D} , por lo que cada componente de la función \mathbf{f} es evaluada a lo sumo una vez.

¹ Llamamos $q_j(t^-)$ al límite por izquierda de $q_j(t)$

Teorema 1. Supongamos que en la aproximación BQSS (6) conocemos $\mathbf{x}(t)$ y $\mathbf{q}(t^-)$ que satisface las Ecs.(7)–(11). Asumamos además que $u_i(t) \neq u_i(t^-)$ o bien que $x_i(t) = q_i(t^-)$ para un subíndice i . Luego, el Algoritmo 1 siempre encuentra un valor $\mathbf{q}(t)$ que cumple con las Ecs.(7)–(11).

Demostración: Para los $i \notin \mathcal{D}$, definimos $\tilde{\mathbf{q}}^{(i)} \triangleq \mathbf{q}(t^-)$.

Demostraremos entonces que una componente $q_j(t)$ arbitraria cumple las Ecs.(7)–(11).

Observar que los únicos valores que pueden ser asignados a $q_j(t)$ son $q_j(t^-)$, $\underline{q}_j(t)$ o $\bar{q}_j(t)$. En los últimos dos casos, la Ec.(7) se cumple automáticamente.

En el primer caso, al asignar $q_j(t) = q_j(t^-)$ puede verse que se cumple $\underline{q}_j(t) = \underline{q}_j(t^-)$ y $\bar{q}_j(t) = \bar{q}_j(t^-)$. De otra forma, $x_j(t)$ hubiera alcanzado a su variable cuantificada $q_j(t^-)$ y no hubiéramos asignado $q_j(t) = q_j(t^-)$ (punto 1.b.). Con esto, se garantiza que siempre se cumple la Ec.(7).

Para probar el cumplimiento de (10) y (11) se debe demostrar que si $f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) \leq 0$ entonces $\exists \hat{\mathbf{q}}^{(j)}$ tal que $f_j(\hat{\mathbf{q}}^{(j)}, \mathbf{u}(t)) = 0$ y las componentes de $\hat{\mathbf{q}}$ satisfacen (11).

Notar que si $q_j(t) \neq q_j(t^-)$, el nuevo valor se calcula según la Ec. (13). Por lo tanto, teniendo en cuenta (8)–(9) se cumplirá que

$$f_j(\tilde{\mathbf{q}}^{(j)}, \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) \geq 0 \quad (15)$$

Por otro lado, si se asignó $q_j(t) = q_j(t^-)$, era porque la ecuación (15) se cumplía. En definitiva, la ecuación (15) se cumple siempre.

Teniendo en cuenta esta ecuación, si $f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) \leq 0$, entonces

$$f_j(\tilde{\mathbf{q}}^{(j)}(t), \mathbf{u}(t)) \cdot f_j(\mathbf{q}(t), \mathbf{u}(t)) \leq 0$$

y por el teorema del valor medio, existe un $\hat{\mathbf{q}}^{(j)}$ entre $\tilde{\mathbf{q}}^{(j)}(t)$ y $\mathbf{q}(t)$ tal que $f_j(\hat{\mathbf{q}}^{(j)}, \mathbf{u}) = 0$.

Las Ecs.(8) y (9) aseguran que

$$|\bar{q}_i - x_i| \leq \Delta Q_i + \varepsilon_i; \quad |\underline{q}_i - x_i| \leq \Delta Q_i + \varepsilon_i \quad (16)$$

Por otro lado, puede verse fácilmente que las componentes $\tilde{q}_i^{(j)}$ pueden adoptar sólo los valores $\bar{q}_i(t)$ ó $\underline{q}_i(t)$, o directamente $x_i(t)$ (en el punto 1.b). Entonces

$$\begin{aligned} |\tilde{q}_i^{(j)} - x_i(t)| &\leq \Delta Q_i + \varepsilon_i \\ |q_i^{(j)} - x_i(t)| &\leq \Delta Q_i + \varepsilon_i \end{aligned} \quad (17)$$

De esta ecuación y por estar $\hat{q}_i^{(j)}$ entre $q_i(t)$ y $\tilde{q}_i^{(j)}(t)$ resulta

$$|\hat{q}_i^{(j)} - x_i(t)| \leq \Delta Q_i + \varepsilon_i \quad (18)$$

lo que concluye la demostración. \square

3.3 Modelo DEVS del integrador BQSS

A continuación se describirá una forma de implementar un modelo DEVS equivalente a la aproximación BQSS de un sistema. Esta implementación tendrá una estructura igual a la de QSS, es decir, consistirá en el acoplamiento de funciones estáticas e integradores cuantificados como se muestra en la Fig.1.

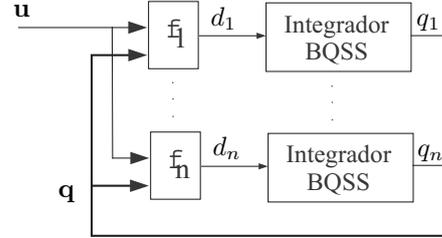


Fig. 1. Diagrama de Bloques de un BQSS

En el mismo, dado que las trayectorias de $u_i(t)$ y $q_i(t)$ son seccionalmente constantes (esto último demostrado en la primera parte del trabajo), las funciones estáticas son idénticas a las de QSS.

La diferencia entre BQSS y QSS radicaré en la forma de calcular q_i en los integradores cuantificados y eventualmente, en la forma de evaluar la derivada de x_i cuando $\Delta f_i \neq 0$ en (12).

Para calcular q_i utilizaremos el Algoritmo 1, pero implementado de manera local por los integradores cuantificados. Esto nos lleva a lo siguiente:

Supongamos que el integrador que calcula q_i recibe en el tiempo t un evento de entrada con valor d_i y supongamos que no había recibido ni provocado otro evento en dicho instante. Entonces, debe calcular

$$q_i(t) = \begin{cases} \bar{q}_i & \text{si } d_i > 0 \\ \underline{q}_i & \text{si } d_i < 0 \\ q_i(t^-) & \text{c.o.c.} \end{cases} \quad (19)$$

Si el nuevo valor de q_i coincide con $q_i(t^-)$, no se produce ningún evento de salida manteniéndose $q_i(t) = q_i(t^-)$. Por otro lado, si el nuevo valor de q_i no coincide con $q_i(t^-)$, instantáneamente produce un evento de salida con el nuevo valor de $q_i(t)$. Luego, cualquier nuevo evento de entrada d_i en el tiempo t no cambia el valor fijado en q_i .

Esto, implementado en todos los integradores, asegura que tras un número máximo de 1 evento de salida por cada integrador quede fijado en cada uno de estos el valor correcto de q_i de acuerdo al Algoritmo 1.

Una vez calculado q_i es simple agendar el próximo evento y fijar el valor de Δf_i . Si al recibir un evento de entrada con valor d_i resulta $(q_i - x_i) \cdot d_i > 0$, se calcula el tiempo en el que se producirá

el próximo evento como $\sigma = (q_i - x_i)/d_i$ y se mantiene $\Delta f_i = 0$, sino, se fija $\sigma = \infty$ y $\Delta f_i = -d_i$.

Este comportamiento puede traducirse fácilmente en el siguiente modelo DEVS.

$$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \quad (20)$$

donde

$$X = Y = \mathbb{R}; S = \mathbb{R}^5 \times \mathbb{R}^+ \\ \delta_{int}(s) = \hat{s}; \delta_{ext}(s, e, u) = \tilde{s}; ta(s) = \sigma$$

siendo

$$s = [x, d_x, q, q_i, q_s, \sigma] \\ \hat{s} = [\hat{x}, \hat{d}_x, \hat{q}, \hat{q}_i, \hat{q}_s, \hat{\sigma}] \\ \hat{x} = x + \sigma \cdot d_x \\ \hat{q} = \begin{cases} \hat{q}_s & \text{si } d_x > 0 \wedge \sigma > 0 \\ q_s & \text{si } d_x > 0 \wedge \sigma = 0 \\ \hat{q}_i & \text{si } d_x \leq 0 \wedge \sigma > 0 \\ q_i & \text{si } d_x \leq 0 \wedge \sigma = 0 \end{cases} \\ \hat{q}_i = \begin{cases} \hat{q}_s - 2 \cdot \Delta Q & \text{si } d_x > 0 \wedge \sigma > 0 \\ q_i - \Delta Q & \text{si } d_x \leq 0 \wedge \sigma > 0 \\ q_i & \text{c.o.c} \end{cases} \\ \hat{q}_s = \begin{cases} q_s + \Delta Q & \text{si } d_x > 0 \wedge \sigma > 0 \\ \hat{q}_i + 2 \cdot \Delta Q & \text{si } d_x \leq 0 \wedge \sigma > 0 \\ q_s & \text{c.o.c} \end{cases} \\ \hat{\sigma} = \begin{cases} (\hat{q} - \hat{x})/d_x & \text{si } d_x \neq 0 \\ \infty & \text{c.o.c} \end{cases} \\ \tilde{s} = [x + e \cdot d_x, \tilde{d}_x, q, \tilde{q}_i, \tilde{q}_s, \tilde{\sigma}] \\ \tilde{d}_x = \begin{cases} 0 & \text{si } u \cdot (q - \tilde{x}) < 0 \wedge e = 0 \\ u & \text{c.o.c.} \end{cases} \\ \tilde{q}_i = \begin{cases} q_i + \Delta Q & \text{si } \tilde{x} - q_i \geq \Delta Q + \epsilon \\ q_i & \text{c.o.c} \end{cases} \\ \tilde{q}_s = \begin{cases} q_s - \Delta Q & \text{si } q_s - \tilde{x} \geq \Delta Q + \epsilon \\ q_s & \text{c.o.c} \end{cases} \\ \tilde{\sigma} = \begin{cases} (q - \tilde{x})/u & \text{si } u \neq 0 \wedge u \cdot (q - \tilde{x}) \geq 0 \\ 0 & \text{si } u \cdot (q - \tilde{x}) < 0 \wedge e > 0 \\ \infty & \text{c.o.c.} \end{cases}$$

y la Función de Salida resulta

$$\lambda(s) = \begin{cases} q_s & \text{si } \sigma = 0 \wedge u > 0 \\ q_i & \text{si } \sigma = 0 \wedge u \leq 0 \\ q + \Delta Q & \text{si } \sigma \neq 0 \wedge u > 0 \\ q - \Delta Q & \text{c.o.c} \end{cases}$$

3.4 Implementación en PowerDEVS

PowerDEVS (Pagliero and Lapadula, 2002) es un paquete de programas para la simulación de modelos DEVS. El mismo, cuenta con un editor gráfico de diagrama de bloques que permite construir sistemas DEVS acoplados de modelos DEVS elementales (atómicos).

El modelo DEVS descrito se agregó al bloque integrador que implementaba los métodos de QSS1, QSS2 y QSS3. Este bloque permite elegir entre los métodos mencionados y ahora también

BQSS. Además, permite incorporar el quantum y la condición inicial como parámetros.

En la implementación, se tomó un ancho de histéresis $\epsilon = \Delta Q/100$ para evitar que aumente la cota de error, manteniendo la legitimidad.

4. EJEMPLOS

Presentamos a continuación resultados de simulación sobre dos sistemas stiff.

Todas las simulaciones presentadas en esta sección se realizaron en AMD Sempron 2400 de 1.67 GHz con 248MB de RAM y fueron realizadas en los simuladores PowerDEVS y Simulink 6.3.

4.1 Sistema Lineal de Segundo Orden

En la primera parte de este artículo (Kofman *et al.*, 2006) se presentó el sistema

$$\dot{x}_1(t) = 0.01 x_2(t) \\ \dot{x}_2(t) = -100 x_1(t) - 100 x_2(t) + 2020 \quad (21)$$

con condiciones iniciales $x_1(0) = 0$ y $x_2(0) = 20$.

Se realizó primero una simulación con BQSS usando quantum $\Delta Q_1 = \Delta Q_2 = 1$. Luego se repitió el experimento disminuyendo en 10 y 100 veces la cuantificación. En el primer caso, la simulación se completó con 20 y 22 pasos en x_1 y x_2 respectivamente. Para $\Delta Q_i = 0.1$ se realizaron 201 pasos en cada variable y para $\Delta Q_i = 0.01$, 2006 y 2024 pasos en cada variable. En todos los casos el tiempo final se seleccionó $t_f = 1000$, pero la simulación alcanzó una situación estable (sin realizar más pasos) antes de $t = 500$.

En la Fig.2 se observan las respuestas obtenidas con $\Delta Q_i = 1$ y 0.1 (no se incluyó la correspondiente a 0.01 por no ser apreciable a la vista la diferencia con la de $\Delta Q_i = 0.1$).

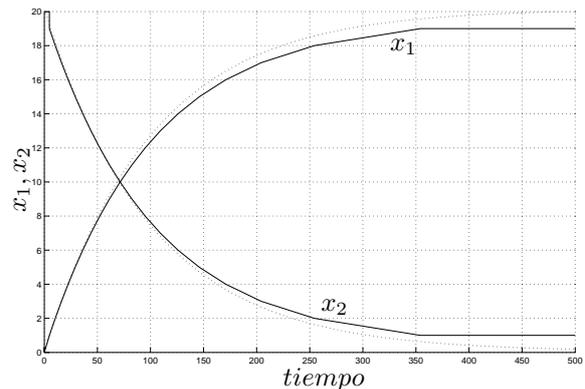


Fig. 2. Respuesta del sistema lineal (21)

La cota de error global teórica, según lo calculado en el Teorema 2 de (Kofman *et al.*, 2006), para $\Delta Q_i = 1$ es de

$$e_1 \leq 3.004; e_2 = 5.001 \quad (22)$$

mientras que para $\Delta Q_i = 0.1$ es 10 veces menor y para $\Delta Q_i = 0.01$ es 100 veces menor. Sin embargo, puede verse en la gráfica que la diferencia entre la primera trayectoria y la última es siempre menor que 1 para ambas variables, por lo que el error en la primera simulación no puede ser mayor que 1.03 en x_1 y 1.05 en x_2 , lo que muestra que la cota de error es en realidad bastante conservadora.

4.2 Sistema No lineal Stiff

El siguiente problema no lineal es un problema stiff estándar de prueba (Enright and Pryce, 1987).

$$\begin{aligned} \dot{x}_1 &= -0.013x_1 - 1000x_1x_3 \\ \dot{x}_2 &= -2500x_2x_3 \\ \dot{x}_3 &= -0.013x_1 - 1000x_1x_3 - 2500x_2x_3 \end{aligned} \quad (23)$$

con condiciones iniciales $x_1(0) = 1$ $x_2(0) = 1$ $x_3(0) = 0$.

Para la simulación con BQSS se utilizó la cuantificación $\Delta Q_1 = 0.01$ $\Delta Q_2 = 0.01$ $\Delta Q_3 = 10^{-7}$ y un tiempo final de 1000.

Los resultados de la simulación pueden verse en la Fig.3. La misma se realizó con un total de 456 pasos (100 en x_1 , 105 en x_2 y 251 en x_3), llegando a una situación estable en $t=419.66$. El tiempo total de cálculo fue de aproximadamente 6 milisegundos.

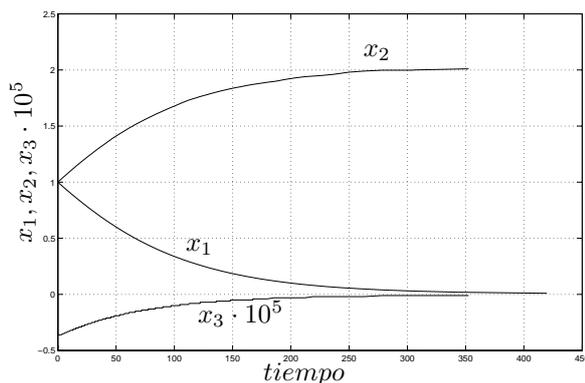


Fig. 3. Respuesta del sistema no lineal (23)

Al simular el mismo sistema con los métodos implícitos de Matlab ode15s, ode23s y ode23tb (utilizando una tolerancia de 10^{-3}) se redujo el número de pasos a 44, 27 y 27 respectivamente, pero el tiempo de cálculo se incrementó a aproximadamente 17 milisegundos (en modo compilado). La comparación de estos resultados con los de BQSS muestran que el error es del orden del quantum utilizado.

El incremento del tiempo de simulación en Matlab es atribuible al costo computacional de cada paso, en el cual debe resolverse una ecuación implícita, mientras que en BQSS cada paso sólo involucra cálculos escalares elementales. Si bien tanto PowerDEVS como Matlab ejecutan código C++ compilado, la forma en que lo hacen no es la misma, lo que relativiza esta comparación.

5. CONCLUSIONES

Este trabajo presentó la implementación del algoritmo de BQSS, un método de integración por cuantificación de primer orden para la simulación de sistemas stiff. BQSS es el primer método implícito de cuantificación.

Siendo un método de definición implícita, la presencia de cuantificación permite su implementación explícita. Desde el punto de vista práctico, BQSS permite integrar sistemas stiff sin necesidad de iteraciones.

Los dos ejemplos estudiados mostraron la eficiencia y eficacia del método y de su implementación en PowerDEVS.

Las direcciones futuras de este trabajo son discutidas en la primer parte del artículo (Kofman *et al.*, 2006).

REFERENCIAS

- Cellier, F. and E. Kofman (2006), *Continuous System Simulation*, Springer, New York.
- Enright, W. H. and J. D. Pryce (1987), "Two (fortran) packages for assessing initial value methods," *(ACM) Transactions on Mathematical Software*, **13**, 1-27.
- Kofman, E. (2002), "A Second Order Approximation for DEVS Simulation of Continuous Systems," *Simulation*, **78**, 76-89.
- Kofman, E. (2006), "A Third Order Discrete Event Simulation Method for Continuous System Simulation," *Latin American Applied Research*, **36**, 101-108.
- Kofman, E. and S. Junco (2001), "Quantized State Systems. A DEVS Approach for Continuous System Simulation," *Transactions of SCS*, **18**, 123-132.
- Kofman, E., G. Migoni, and F. Cellier (2006), "Integración por Cuantificación de Sistemas Stiff. Parte I: Teoría," Technical Report LSD0601, LSD-UNR, Submitted to AADECA 2006.
- Pagliari, E. and M. Lapadula (2002), "Herramienta Integrada de Modelado y Simulación de Sistemas de Eventos Discretos," Diploma Work. FCEIA, UNR, Argentina.
- Zeigler, B., T. Kim, and H. Praehofer (2000), *Theory of Modeling and Simulation. Second edition*, Academic Press, New York.