

INTEGRACIÓN POR CUANTIFICACIÓN DE SISTEMAS STIFF

Gustavo Migoni* Ernesto Kofman*
François Cellier**

* *Laboratorio de Sistemas Dinámicos. FCEIA - UNR - CONICET. Riobamba 245 bis - (2000) Rosario. Argentina*

** *Institute of Computational Science - ETH Zürich. CH-8092 Zürich. Switzerland.*

Resumen: Este artículo presenta un nuevo método de integración numérica de ecuaciones diferenciales ordinarias basado en la cuantificación de las variables de estado. Utilizando conceptos de integración implícita, el nuevo método denominado BQSS (Backward Quantized State Systems) permite integrar sistemas stiff de manera eficiente. Siendo el primer método de cuantificación para sistemas stiff, BQSS es en sí un método explícito, por lo que la contribución es importante en el contexto general de los métodos de integración numérica. Además de presentar el método, el artículo estudia las principales propiedades teóricas del mismo, aborda aspectos prácticos de la implementación del algoritmo y presenta resultados de simulación sobre dos sistemas stiff.

Copyright ©2007 CEA-IFAC

Palabras Clave: Métodos de Integración Numérica, Sistemas Stiff, Integración por Cuantificación.

1. INTRODUCCIÓN

Muchos sistemas dinámicos de la práctica, tanto en las ciencias como en la ingeniería, son stiff. Esto es, tienen matrices Jacobianas con autovalores muy separados sobre el eje real negativo del plano complejo.

La integración de estos sistemas mediante métodos numéricos tradicionales de discretización temporal requiere de la utilización de algoritmos implícitos, ya que todos los métodos explícitos deben necesariamente restringir excesivamente el paso de integración para garantizar estabilidad numérica.

Sin embargo, los métodos implícitos tienen una mayor carga computacional ya que requieren de algoritmos iterativos en cada paso para despejar el siguiente valor. Esto es además inaceptable en aplicaciones de tiempo real dado que no puede

garantizarse cuanto tiempo demandará la convergencia de las iteraciones (típicamente se utilizan iteraciones de Newton). Por este motivo, la simulación de sistemas stiff en tiempo real es un problema para el cual no se habían encontrado aún soluciones satisfactorias.

Un enfoque esencialmente distinto al de los métodos de integración clásica surge al reemplazar la discretización del tiempo por la cuantificación de las variables de estado. Esta idea dio lugar a los *métodos de integración por cuantificación*, que aproximan las ecuaciones diferenciales ordinarias por *sistemas de eventos discretos* en términos del formalismo DEVS (Zeigler *et al.*, 2000).

El primero de estos métodos es el de QSS1 (Quantized State Systems) (Kofman and Junco, 2001), que realiza una aproximación de primer orden. En base a principios similares, se desarrollaron tam-

bién métodos de segundo orden (QSS2) (Kofman, 2002) y de tercer orden (QSS3) (Kofman, 2006), que permiten obtener una mejor precisión sin incrementar mucho el número de cálculos.

Los métodos de QSS tienen propiedades teóricas muy fuertes (estabilidad y existencia de cota de error global calculable para sistemas lineales) y presentan grandes ventajas al simular sistemas discontinuos (Kofman, 2004). Sin embargo, no son apropiados para la simulación de sistemas stiff, debido a la aparición de oscilaciones de alta frecuencia (Cellier and Kofman, 2006). Lo que ocurre es que, al igual que cualquier método clásico explícito, el paso de integración se reduce excesivamente para mantener la estabilidad numérica.

El presente artículo desarrolla entonces un nuevo método de cuantificación de primer orden, similar a QSS1, pero utilizando principios de integración implícita, realizando la cuantificación sobre valores futuros de los estados. Esta idea ya había sido planteada en (Kofman, 2003) y (Cellier and Kofman, 2006), pero nunca formalizada ni implementada.

Este nuevo método –que denominaremos BQSS por Backward Quantized State Systems– veremos que permite la simulación de sistemas stiff sin necesidad de reducir el paso de integración.

Además de ser el primer algoritmo tipo backward de la familia de QSS, BQSS tiene la particularidad de no requerir ningún tipo de iteraciones para su implementación. Gracias a la cuantificación, un estado cuantificado cuyo valor es q_i puede en el siguiente paso tomar sólo dos valores: $q_i + \Delta q_i$ o $q_i - \Delta q_i$ lo que simplifica muchísimo el problema. Más aún, veremos que la determinación del siguiente estado puede realizarse aún probando sólo uno de los dos valores posibles, lo que conlleva una solución explícita.

Además de definir el método, en este trabajo estudiaremos sus principales propiedades teóricas, que resultarán similares a las de los anteriores métodos de QSS. Veremos también los principales detalles relativos a la implementación del método y analizaremos dos ejemplos de simulación.

El artículo está organizado de la siguiente forma: en la Sección 2 se presenta el método de QSS1 y sus problemas frente a la simulación de sistemas stiff. Luego, en la Sección 3 se introduce el nuevo método BQSS y se describe su implementación. En la Sección 4 se estudian las principales propiedades del nuevo método y en la Sección 5 se analizan dos ejemplos de aplicación.

2. INTEGRACIÓN POR CUANTIFICACIÓN

Presentaremos en esta sección el método de QSS1, la idea básica de su implementación mediante el formalismo DEVS y los problemas que aparecen en este método al integrar sistemas stiff.

2.1 Método de QSS1

Dado el sistema

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (1)$$

el método de QSS1 lo aproxima por

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

En este último sistema, \mathbf{q} es el vector de variables cuantificadas, cuyas componentes siguen trayectorias seccionalmente constantes relacionadas con las del vector de estado \mathbf{x} mediante la siguiente función de cuantificación con histéresis¹

$$q_j(t) = \begin{cases} q_j(t^-) + \Delta Q_j & \text{si } x_j(t) - q_j(t^-) \geq \Delta Q_j \\ q_j(t^-) - \Delta Q_j & \text{si } q_j(t^-) - x_j(t) \geq \varepsilon_j \\ q_j(t^-) & \text{en otro caso} \end{cases}$$

Aquí ΔQ_j se denomina *quantum* y ε_j se denomina *ancho de histéresis*.

El ancho de histéresis en general se toma igual al quantum (para reducir oscilaciones sin aumentar el error). En tal caso, puede verse fácilmente que $q_j(t)$ sigue una trayectoria seccionalmente constante, que sólo cambia cuando la diferencia con $x_j(t)$ se hace igual al quantum, y tras cada cambio queda $q_j(t) = x_j(t)$.

2.2 Ejemplo Introductorio

Para ilustrar el funcionamiento del método de QSS1, consideremos el sistema

$$\begin{aligned} \dot{x}_1(t) &= 0.01 x_2(t) \\ \dot{x}_2(t) &= -100 x_1(t) - 100 x_2(t) + 2020 \end{aligned} \quad (3)$$

El método de QSS1 aproximará este sistema según

$$\begin{aligned} \dot{q}_1(t) &= 0.01 q_2(t) \\ \dot{q}_2(t) &= -100 q_1(t) - 100 q_2(t) + 2020 \end{aligned} \quad (4)$$

Considerando condiciones iniciales $x_1(0) = 0$, $x_2(0) = 20$, y cuantificación $\Delta Q_1 = \Delta Q_2 = 1$, la integración por QSS1 hará los siguientes pasos:

En $t = 0$ tenemos $q_1(0) = 0$ y $q_2(0) = 20$. Por lo tanto, de acuerdo a la Ec.(4), tendremos $\dot{q}_1(0) = 0.2$ y $\dot{q}_2(0) = 20$. Esta situación se mantiene hasta que $|q_i - x_i| = \Delta Q_i = 1$.

El próximo cambio en q_1 es entonces agendado para $t = 1/0.2 = 5$ mientras que el cambio en q_2 se agenda para $t = 1/20 = 0.05$.

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

Por lo tanto en $t = 0.05$ hay un nuevo paso, tras el cual, $q_1(0.05) = 0$, $q_2(0.05) = 21$, $x_1(0.05) = 0.01$, $x_2(0.05) = 21$. Las derivadas quedan $\dot{x}_1(0.05) = 0.21$ y $\dot{x}_2(0.05) = -80$.

El próximo cambio en q_1 se reagenda para $t = 0.05 + (1 - 0.01)/0.21 = 4.764$ mientras el siguiente cambio en q_2 se agenda para $0.05 + 1/80 = 0.0625$. Por lo tanto, el siguiente paso se hace en $t = 0.0625$.

Luego retomaremos este ejemplo, pero puede verse que los cálculos son muy sencillos.

Si bien el algoritmo descripto se puede implementar en forma directa en cualquier lenguaje de programación, una manera simple y general de hacerlo es mediante el formalismo DEVS.

2.3 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:

$$M = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta) \quad (5)$$

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.
- δ_{int} , δ_{ext} , λ 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.

2.4 DEVS y QSS

Cada componente de la Ec.(2) puede pensarse como el acoplamiento de dos subsistemas elementales. Uno estático

$$d_i(t) = f_i(q_1, \dots, q_n, u_1, \dots, u_m) \quad (6)$$

y otro dinámico

$$q_i(t) = Q_i(x_i(\cdot)) = Q_i\left(\int d_i(\tau) d\tau\right) \quad (7)$$

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 (6), 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 (6) y (7) 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).

Luego, el método de QSS1 puede implementarse sencillamente acoplando los modelos DEVS correspondientes a los integradores cuantificados y funciones estáticas.

Un programa que permite implementar de manera directa el método de QSS1 (y también los de orden superior) es PowerDEVS (Pagliero and Lapadula, 2002). El mismo cuenta con un entorno gráfico en el cual se puede armar directamente el diagrama de bloques que representa la ecuación diferencial a simular. En el mismo, cada bloque es en realidad un modelo DEVS elemental de una función estática y/o integrador cuantificado. En definitiva, el entorno de modelado es casi igual al de Simulink.

2.5 QSS1 en Sistemas Stiff

Volviendo al ejemplo introductorio, puede notarse que en realidad, el sistema de la Ec.(3) es stiff, ya que tiene autovalores $\lambda_1 \approx -0.01$ y $\lambda_2 \approx -99.99$.

Continuaremos entonces la simulación *a mano* del mismo con el método de QSS1 para analizar lo que ocurre.

En $t = 0.0625$ tenemos $q_1(0.0625) = 0$, $q_2(0.0625) = x_2(0.0625) = 20$, $x_1(0.0625) \approx 0.0126$ y las derivadas son las mismas que en $t = 0$.

Esto se repite cíclicamente hasta que efectivamente se produce el cambio en q_1 (esto se da en $t \approx 4.95$, tras 158 cambios en el valor de q_2 , oscilando entre 20 y 21).

La simulación continua de la misma manera. En las Figs.1–2 puede observarse la evolución de $q_1(t)$ y $q_2(t)$ hasta $t = 500$ seg.

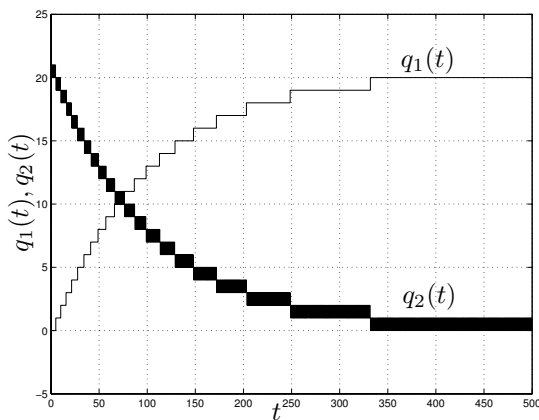


Figura 1. Simulación con QSS1.

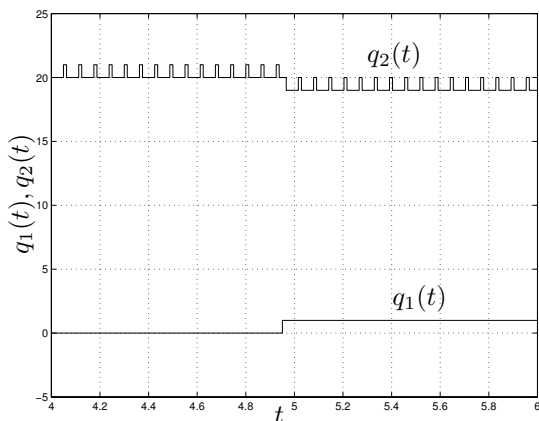


Figura 2. Simulación con QSS1 (detalle).

Como puede verse, hay oscilaciones rápidas en q_2 que provocan un total de 15995 transiciones en dicha variable, mientras que q_1 realiza sólo 21 cambios. En definitiva hay más de 16000 pasos para completar la simulación (es del orden de los 25000 pasos que realizaría como mínimo el método de Euler para obtener un resultado estable).

Evidentemente, el método de QSS1 no es capaz de integrar eficientemente el sistema (3). En realidad, algo muy parecido ocurre con los métodos de orden superior (QSS2 y QSS3)

3. BACKWARD QSS

En esta sección introduciremos el nuevo método de BQSS y analizaremos los principales detalles de su implementación.

3.1 Idea Básica

La integración eficiente de sistemas stiff requiere del uso de métodos implícitos, que evalúen la derivada en instantes futuros de tiempo.

Esta idea, aplicada a QSS1 implicaría que las componentes de $\mathbf{q}(t)$ en (2) sean versiones cuantificadas de valores futuros de $\mathbf{x}(t)$. En otras palabras, dado $x_i(t)$, $q_i(t)$ será un valor cuantificado cercano a $x_i(t)$ tal que $x_i(t)$ evolucione hacia $q_i(t)$.

Para el ejemplo introductorio de la Ec.(3), con las condiciones iniciales y cuantificación utilizada anteriormente, esta idea nos llevaría a la siguiente simulación:

En $t = 0$, podemos elegir $q_2(0) = 19$ o $q_2(0) = 21$ según $\dot{x}_2(0)$ sea positivo o negativo. En ambos casos resultará $\dot{x}_1(0) > 0$ por lo que el valor cuantificado *futuro* de x_1 será $q_1(0) = 1$.

Si elegimos $q_2(0) = 21$ resulta $\dot{x}_2(0) = -180 < 0$, por lo que x_2 no evoluciona hacia q_2 , sino en sentido contrario.

Por otro lado, si elegimos $q_2(0) = 19$ resulta $\dot{x}_2(0) = 20 > 0$. Luego, no es posible elegir q_2 tal que x_2 evolucione hacia q_2 .

Sin embargo, dado que el signo de \dot{x}_2 cambia según elijamos $q_2 = 19$ o $q_2 = 21$, necesariamente hay un punto intermedio \hat{q}_2 entre estos valores tal que $\dot{x}_2 = 0$.

En este caso, dejaremos $q_2 = 21$ pero fijaremos $\dot{x}_2 = 0$ (como si q_2 hubiera adoptado el valor de la raíz \hat{q}_2).

El próximo cambio en q_1 quedará entonces agendado para $t = 1/0.21 \approx 4.762$ mientras que el próximo cambio en q_2 se agendará para $t = \infty$.

El siguiente paso es entonces en $t = 4.762$. Aquí resulta $x_1 = 1$ y $x_2 = 20$. Luego tendremos $q_1(4.762) = 2$ (porque $\dot{x}_1 > 0$). Aquí evaluamos nuevamente \dot{x}_2 para $q_2 = 19$ y $q_2 = 21$, resultando negativa en ambos casos, por lo tanto el valor correcto es $q_2(4.762) = 19$ ya que de esta manera x_2 evoluciona hacia q_2 .

Con estos valores de q_1 y q_2 se tiene $\dot{x}_1 = 0.19$ y $\dot{x}_2 = -80$. El próximo cambio en q_1 se agenda para $t = 4.762 + 1/0.19 = 10.025$, mientras que el de q_2 se agenda para $t = 4.762 + 1/80 = 4.774$. Por lo tanto, el siguiente paso se da en $t = 4.774$, cuando x_2 alcanza a q_2 .

Los cálculos continúan de la misma manera. El algoritmo es similar a QSS, sólo que siempre intentamos utilizar el valor q_i que esté en el sentido de la evolución de x_i . Cuando esto no es posible, es porque existe un punto intermedio en el cual $\dot{x}_i = 0$ y entonces forzamos esta condición, pero conservando el valor de q_i .

La Figura 3 muestra el resultado de esta simulación, que llevó 21 cambios en q_1 y 22 en q_2 . En realidad, luego de $t = 354.24$ seg no se realizan más cálculos ya que los cambios en ambas variables se agendan en $t = \infty$ (se encuentra una situación estable).

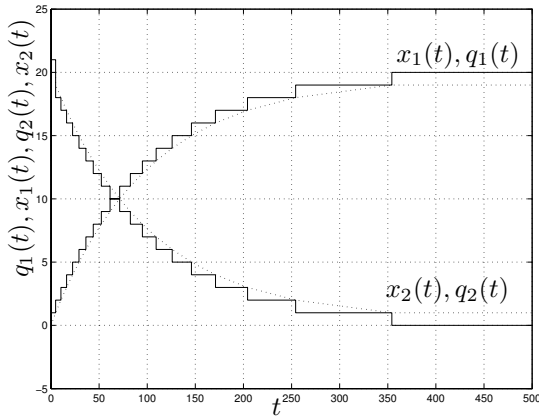


Figura 3. Simulación con BQSS

Esta es la idea básica del método de BQSS: Para cada variable de estado usamos dos funciones de cuantificación, una por encima y otra por debajo del valor del estado x_i . Luego, el valor de q_i adopta una u otra función de cuantificación según el sentido en que quede la derivada de x_i .

Esta idea, en el caso analizado, funcionó muy bien. El algoritmo pudo resolver el sistema stiff (3) con sólo 43 pasos, lo que iguala la performance de cualquier método implícito (si bien el error es aquí algo grande ya que es una aproximación de primer orden).

Sin embargo, como sabemos del caso de QSS, el uso de cuantificación sin histéresis puede provocar oscilaciones infinitamente rápidas. Por ejemplo, usando la misma idea que antes en el sistema

$$\begin{aligned}\dot{x}_1(t) &= 0.5x_1 + x_2(t) \\ \dot{x}_2(t) &= -x_1(t) + 0.5x_2(t)\end{aligned}\quad (8)$$

con $\Delta Q_i = 1$ y condiciones iniciales $x_1(0) = 0.1$, $x_2(0) = -0.01$ se obtiene una solución donde los

cambios en q_1 y q_2 ocurren cada vez más rápido, con una frecuencia que tiende a infinito.

La solución a este problema, al igual que en QSS, es el agregado de histéresis en las funciones de cuantificación superior e inferior.

3.2 Definición de BQSS

Dado el sistema (1), el método BQSS lo aproxima por:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) + \Delta \mathbf{f} \quad (9)$$

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

$$q_j(t) \in \{\underline{q}_j(t), \bar{q}_j(t)\} \quad (10)$$

con:

$$\underline{q}_j(t) = \begin{cases} \underline{q}_j(t^-) - \Delta Q_j & \text{si } x_j(t) - \underline{q}_j(t^-) \leq 0 \\ \underline{q}_j(t^-) + \Delta Q_j & \text{si } x_j(t) - \underline{q}_j(t^-) \geq \varepsilon_j + \Delta Q_j \\ \underline{q}_j(t^-) & \text{en otro caso} \end{cases} \quad (11)$$

$$\bar{q}_j(t) = \begin{cases} \bar{q}_j(t^-) + \Delta Q_j & \text{si } \bar{q}_j(t^-) - x_j(t) \leq 0 \\ \bar{q}_j(t^-) - \Delta Q_j & \text{si } \bar{q}_j(t^-) - x_j(t) \geq \varepsilon_j + \Delta Q_j \\ \bar{q}_j(t^-) & \text{en otro caso} \end{cases} \quad (12)$$

y además debe verificarse:

$$\begin{aligned}f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) &> 0 \\ \vee \\ \exists \hat{\mathbf{q}}^{(j)}(t) / f_j(\hat{\mathbf{q}}^{(j)}(t), \mathbf{u}(t)) &= 0\end{aligned}\quad (13)$$

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

$$|x_i(t) - \hat{q}_i^{(j)}(t)| < \Delta Q_i + \varepsilon_i \quad (14)$$

Por otro lado

$$\Delta f_j = \begin{cases} 0, & \text{si } f_j(\mathbf{q}(t), \mathbf{u}(t)) \cdot (q_j - x_j) > 0 \\ -f_j(\mathbf{q}(t), \mathbf{u}(t)), & \text{en otro caso} \end{cases} \quad (15)$$

Como en QSS1, a ΔQ_j lo llamaremos quantum y a $\varepsilon_j < \Delta Q_j$ lo denominaremos ancho de histéresis.

Como vemos, la definición del valor de q_j conocido x_j , es implícita. Incluso, puede conllevar más de una solución. Sin embargo, sabemos que $q_j(t)$ puede tomar sólo 2 valores ($\underline{q}_j(t)$ o $\bar{q}_j(t)$).

En principio, podría pensarse que es necesario evaluar todas las combinaciones de valores posibles de los q_j hasta encontrar un vector \mathbf{q} correcto. Sin embargo, como veremos a continuación, esto no es necesario sino que \mathbf{q} puede obtenerse de manera explícita.

3.3 Obtención explícita de \mathbf{q} .

La diferencia principal de BQSS respecto de QSS1 es la forma de obtención de \mathbf{q} a partir de \mathbf{x} , ya

que (13) implica cierta interdependencia entre los valores de las distintas componentes.

En QSS1, 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 (13). 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.(10)–(14).

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

$$q_i(t) \triangleq \begin{cases} \tilde{q}_i(t) & \text{si } f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) > 0 \\ \underline{q}_i(t) & \text{si } f_i(\tilde{\mathbf{q}}^{(i)}, \mathbf{u}(t)) < 0 \end{cases} \quad (16)$$

- 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.(16)
 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

$$\tilde{q}_k^{(i)} = \begin{cases} q_k(t) & \text{si } k \in \mathcal{D} \\ q_k(t^-) & \text{si } k \notin \mathcal{D} \end{cases} \quad (17)$$

- 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.(16).
 - 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 3.2). 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.

Teorema 1. Supongamos que en la aproximación BQSS (9) conocemos $\mathbf{x}(t)$ y $\mathbf{q}(t^-)$ que satisface las Ecs.(10)–(14). 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.(10)–(14).

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.(10)–(14).

Observar que los únicos valores que pueden ser asignados a $q_j(t)$ son $q_j(t^-)$, $\underline{q}_j(t)$ o $\tilde{q}_j(t)$. En los últimos dos casos, la Ec.(10) 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 $\tilde{q}_j(t) = \tilde{q}_j(t^-)$. 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.(10).

Para probar el cumplimiento de (13) y (14) 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 (14).

Notar que si $q_j(t) \neq q_j(t^-)$, el nuevo valor se calcula según la Ec. (16). Por lo tanto, teniendo en cuenta (11)–(12) se cumplirá que

$$f_j(\tilde{\mathbf{q}}^{(j)}, \mathbf{u}(t)) \cdot (q_j(t) - x_j(t)) \geq 0 \quad (18)$$

Por otro lado, si se asignó $q_j(t) = q_j(t^-)$, era porque la ecuación (18) se cumplía. En definitiva, la ecuación (18) 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.(11) y (12) aseguran que

$$|\tilde{q}_i - x_i| \leq \Delta Q_i + \varepsilon_i; \quad |\underline{q}_i - x_i| \leq \Delta Q_i + \varepsilon_i \quad (19)$$

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

$$\begin{aligned} |\tilde{q}_i^{(j)} - x_i(t)| &\leq \Delta Q_i + \varepsilon_i \\ |\underline{q}_i^{(j)} - x_i(t)| &\leq \Delta Q_i + \varepsilon_i \end{aligned} \quad (20)$$

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

$$|\hat{q}_i^{(j)} - x_i(t)| \leq \Delta Q_i + \varepsilon_i \quad (21)$$

lo que concluye la demostración. \square

3.4 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.4.

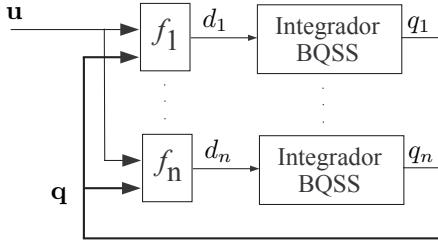


Figura 4. Diagrama de Bloques de un BQSS

En el mismo, dado que las trayectorias de $u_i(t)$ y $q_i(t)$ son seccionalmente constantes (esto último lo demostraremos en la siguiente sección), las funciones estáticas son idénticas a las de QSS.

La diferencia entre BQSS y QSS1 radicará 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 (15).

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

$$q_i(t) = \begin{cases} \bar{q}_i & \text{si } d_i > 0 \\ \underline{q}_i & \text{si } d_i < 0 \\ q_i(t^-) & \text{c.o.c.} \end{cases} \quad (22)$$

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 en el siguiente modelo DEVS.

$$M = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \quad (23)$$

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

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

y la Función de Salida resulta

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

4. PROPIEDADES DE BQSS

Trataremos aquí las propiedades fundamentales del método de BQSS. Primero veremos que en BQSS se realizan un número finito de pasos para cada intervalo finito de tiempo (lo que garantiza que las simulaciones no pueden *trabarse*). Luego, analizaremos las propiedades de estabilidad y cota de error global.

4.1 Legitimidad de BQSS

Teorema 2. Supongamos que la función \mathbf{f} del lado derecho de (1) es acotada en cualquier dominio acotado y que la función $\mathbf{u}(t)$ es seccionalmente constante. Luego,

1. Cualquier solución $\mathbf{x}(t)$ de (9) es continua mientras permanece acotada,
2. La trayectoria de $\mathbf{q}(t)$ es seccionalmente constante mientras permanece acotada.

Demostración: La prueba del punto (1) es trivial, ya que al estar acotada la derivada de \mathbf{x} en (9), la trayectoria resulta continua.

Para el punto (2) en tanto, es claro que cada componente q_j puede sólo tomar valores de la forma $k \cdot \Delta Q_j$. Sin embargo, para probar que \mathbf{q} es seccionalmente constante es necesario asegurar que sólo cambia un número finito de veces en cualquier intervalo finito de tiempo.

Sea entonces un intervalo arbitrario (t_1, t_2) , en el cual una solución $\mathbf{x}(t)$ de (9) permanece acotada. Probaremos que en este intervalo $\mathbf{q}(t)$ realizará sólo un número finito de cambios.

Dado que cada componente q_j difiere de x_j en a lo sumo $\Delta Q_j + \epsilon_j$, resulta que $\mathbf{q}(t)$ está también acotado en (t_1, t_2) . Dado que $\mathbf{u}(t)$ es seccionalmente constante (y por ende acotado en este intervalo) resulta que $\mathbf{f}(\mathbf{q}, \mathbf{u})$ es también acotado. Teniendo en cuenta (9) y (15), existen constantes m_j tales que en el intervalo (t_1, t_2)

$$|\dot{x}_j(t)| \leq m_j; \text{ para } j = 1, \dots, n.$$

Sea $t_c \in (t_1, t_2)$ y supongamos que $\bar{q}_j(t_c^-) \neq \bar{q}_j(t_c^+)$. De acuerdo a (12), esta situación no puede repetirse hasta que $|x_j(t) - x_j(t_c)| \geq \epsilon_j$. Por lo tanto, el intervalo de tiempo mínimo entre dos discontinuidades en $\bar{q}_j(t)$ es

$$t_j = \frac{\epsilon_j}{m_j}$$

Luego, llamando \bar{n}_j al número de cambios que realiza $\bar{q}_j(t)$ en el intervalo (t_1, t_2) , resulta

$$\bar{n}_j \leq (t_2 - t_1) \frac{m_j}{\epsilon_j}$$

Puede verse fácilmente que \underline{q}_j realizará también un máximo número de cambios acotado por la ecuación anterior.

Siendo $\mathbf{u}(t)$ seccionalmente constante, realizará un número de cambios finito n_u en el intervalo (t_1, t_2) .

De acuerdo a la definición de $q_j(t)$, ésta puede tomar sólo los valores $\bar{q}_j(t)$ o $\underline{q}_j(t)$. Además, podrá cambiar sólo si cambian éstas, o si cambia alguna otra variable $q_i(t)$ o $u_i(t)$ para que se siga cumpliendo la restricción (13). En definitiva, los

cambios de $q_j(t)$ estarán ligados a cambios en alguna $\bar{q}_i(t)$, $\underline{q}_i(t)$ o $u_i(t)$. Por lo tanto, el número total de cambios será menor o igual a la suma de cambios en dichas variables, es decir,

$$n_j \leq n_u + 2(t_2 - t_1) \sum_{i=1}^n \frac{m_i}{\epsilon_i}$$

lo que es evidentemente un número finito. \square

4.2 Representación Perturbada

Las propiedades teóricas de los métodos de QSS se basan en el hecho de que la aproximación (2) puede verse como una versión perturbada del sistema original (1), donde las perturbaciones en la primera están acotadas por la cuantificación utilizada. Veremos entonces que en BQSS ocurre algo similar.

Cada componente de la Ec.(9) puede escribirse como

$$\dot{x}_i(t) = f_i(\mathbf{q}(t), \mathbf{u}(t)) + \Delta f_i \quad (24)$$

Definiendo

$$\mathbf{q}^{*(i)}(t) = \begin{cases} \mathbf{q}(t) & \text{si } \Delta f_i = 0 \\ \hat{\mathbf{q}}^{(i)}(t) & \text{en otro caso} \end{cases}$$

y utilizando las Ecs. (13)–(15), la Ec.(24) puede reescribirse como

$$\dot{x}_i = f_i(\mathbf{q}^{*(i)}(t), \mathbf{u}(t)) \quad (25)$$

Definiendo $\Delta \mathbf{x}^{(i)}(t) \triangleq \mathbf{q}^{*(i)} - \mathbf{x}(t)$ y reemplazando en la Ec.(25) resulta

$$\dot{x}_i(t) = f_i(\mathbf{x}(t) + \Delta \mathbf{x}^{(i)}(t), \mathbf{u}(t)) \quad (26)$$

donde

$$|\Delta x_j^{(i)}(t)| \leq \Delta Q_j + \epsilon_j \quad (27)$$

ya que, de las Ecs.(10), (11) y (12) resulta

$$|q_j(t) - x_j(t)| \leq \Delta Q_j + \epsilon_j$$

y de la Ec.(14) se tiene que

$$|\hat{q}_j^{(i)}(t) - x_j(t)| \leq \Delta Q_j + \epsilon_j$$

4.3 Estabilidad y Cota de Error Global

El caso Lineal y Estacionario de la Ec.(1) toma la forma

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t) \quad (28)$$

La aproximación BQSS, en tanto, será

$$\dot{\hat{\mathbf{x}}}(t) = A \mathbf{q}(t) + B \mathbf{u}(t) + \Delta \mathbf{f}(t) \quad (29)$$

Para este caso, el siguiente teorema demuestra la existencia de una cota de error global:

Teorema 3. Supongamos que la matriz A es Hurwitz. Sea $\phi(t)$ la solución de (28) con condición inicial $\phi(0)$ y sea $\hat{\phi}(t)$ una solución de (29) desde

la misma condición inicial. Sea $\mathbf{e}(t) \triangleq \phi(t) - \tilde{\phi}(t)$. Luego, para todo $t \geq 0$ resulta²

$$|\mathbf{e}(t)| \leq |V| \cdot |\operatorname{Re}(\Lambda)^{-1} V^{-1}| |A| (\Delta \mathbf{Q} + \varepsilon) \quad (30)$$

donde $\Lambda = V^{-1} A V$ es la descomposición modal de A y $\Delta \mathbf{Q}$ y ε son los vectores de las cuantificaciones y anchos de histéresis en (29).

Demostración:

De acuerdo a la Ec.(26), la i -ésima componente de (29) puede escribirse como

$$\dot{x}_i(t) = A_i(\mathbf{x}(t) + \Delta \mathbf{x}^{(i)}(t)) + B_i \mathbf{u}(t)$$

Definiendo $d_i(t) \triangleq A_i \Delta \mathbf{x}^{(i)}(t)$, podemos reescribir

$$\dot{x}_i(t) = A_i(\mathbf{x}(t)) + d_i + B_i \mathbf{u}(t) \quad (31)$$

Teniendo en cuenta (27) y la definición de d_i resulta

$$|d_i(t)| \leq |A_i| \cdot (\Delta \mathbf{Q} + \varepsilon) \quad (32)$$

Volviendo a la notación vectorial, (31) resulta:

$$\dot{\mathbf{x}}(t) = A \mathbf{x}(t) + B \mathbf{u}(t) + \mathbf{d}(t) \quad (33)$$

con

$$|\mathbf{d}(t)| \leq |A| \cdot (\Delta \mathbf{Q} + \varepsilon) \quad (34)$$

Reemplazando (28) con $\phi(t)$ y (33) con $\tilde{\phi}(t)$ y restando, resulta el sistema

$$\dot{\mathbf{e}}(t) = A \mathbf{e}(t) + \mathbf{d}(t) \quad (35)$$

con $\mathbf{e}(0) = 0$.

Luego, el Teorema 3.3 de (Kofman *et al.*, 2007) establece directamente la validez de (30) a partir de (35) y (34). \square

Un corolario de este Teorema es que si la matriz A es estable, entonces la aproximación numérica da una solución finalmente acotada (no se puede garantizar estabilidad asintótica, pero siempre se garantiza estabilidad *práctica*).

Otro corolario es que la cota de error depende linealmente de la cuantificación.

5. EJEMPLOS E IMPLEMENTACIÓN

El método de BQSS fue implementado en PowerDEVS, agregándose para esto el algoritmo correspondiente 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.

Presentamos a continuación resultados de simulación sobre dos sistemas stiff. Todas las simulaciones se realizaron en una PC con procesador AMD Sempron 2400 de 1.67 GHz con 248MB de RAM y fueron realizadas en los simuladores PowerDEVS y Simulink 6.3.

5.1 Sistema Lineal de Segundo Orden

Consideremos nuevamente el sistema lineal stiff de la Ec.(3) 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.5 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 $\Delta Q_i = 0.1$).

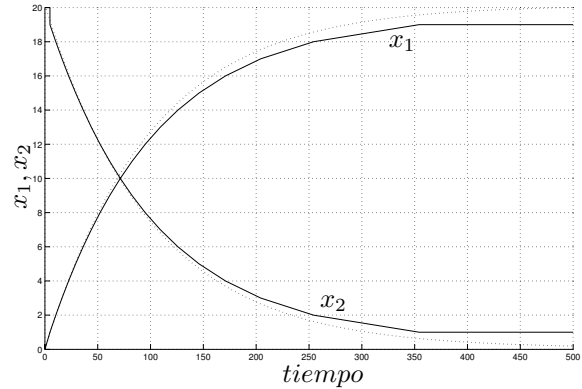


Figura 5. Respuesta del sistema lineal (3)

La cota de error global teórica, según lo calculado en el Teorema 3 para $\Delta Q_i = 1$ es de

$$e_1 \leq 3.004; e_2 = 5.001 \quad (36)$$

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.

² El símbolo " \leq " representa una desigualdad vectorial componente a componente. De manera similar, " $|\cdot|$ " es el módulo por componentes de una matriz o vector.

5.2 Sistema No lineal Stiff

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

$$\begin{aligned} \dot{x}_1 &= -0.013x_1 - 1000x_1x_3 \\ \dot{x}_2 &= -2500x_2x_3 \\ \dot{x}_3 &= -0.013x_1 - 1000x_1x_3 - 2500x_2x_3 \end{aligned} \quad (37)$$

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.6. 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.

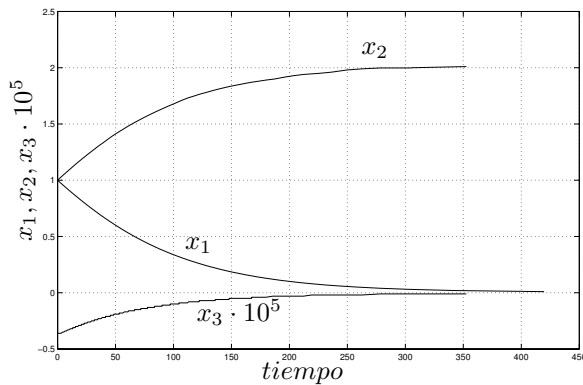


Figura 6. Respuesta del sistema no lineal (37)

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.

6. CONCLUSIONES

Se presentó el método de BQSS, el primer algoritmo de QSS implícito que permite simular sistemas

stiff de manera eficiente. Se demostró también que el método comparte las propiedades de precisión de los métodos de QSS anteriores

Una característica notable de BQSS es que su implementación es en realidad explícita, por lo que se trata de un método explícito que no necesita reducir excesivamente el paso de integración para conservar la estabilidad numérica.

El trabajo futuro deberá orientarse a desarrollar métodos similares de orden mayor, ya que BQSS está limitado en precisión al realizar una aproximación de primer orden. Para esto, debería tomarse como base la idea de las aproximaciones de QSS2 y QSS3.

Además un problema en el que BQSS puede resultar muy importante y que por lo tanto debe estudiarse es la simulación en tiempo real de sistemas stiff, ya que este método no necesita realizar las iteraciones que constituyen el principal inconveniente de los métodos implícitos.

REFERENCIAS

- Cellier, F.E. 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), 1–27.
- Kofman, E. (2002). A Second Order Approximation for DEVS Simulation of Continuous Systems. *Simulation* **78**(2), 76–89.
- Kofman, E. (2003). Simulación y Control de Sistemas Continuos por Eventos Discretos. PhD thesis. Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario.
- Kofman, E. (2004). Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing* **25**(5), 1771–1797.
- Kofman, E. (2006). A Third Order Discrete Event Simulation Method for Continuous System Simulation. *Latin American Applied Research* **36**(2), 101–108.
- Kofman, E. and S. Junco (2001). Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS* **18**(3), 123–132.
- Kofman, E., H. Haimovich and M. Seron (2007). A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control* **80**(2), 167–178.
- Pagliero, E. and M. Lapadula (2002). Herramienta Integrada de Modelado y Simulación de Sistemas de Eventos Discretos. Diploma Work. FCEIA, UNR, Argentina.
- Zeigler, B., T.G. Kim and H. Praehofer (2000). *Theory of Modeling and Simulation. Second edition*. Academic Press. New York.