

Simulación de Sistemas Continuos y a Tramos

Prof. Dr. François E. Cellier
Institut für Computational Science
ETH Zürich

28 de junio 2007

Introducción

A lo largo del curso, aprendimos que para poder confiar en los resultados de una simulación, es crucial analizar la **estabilidad** y la **precisión** de los métodos numéricos utilizados.

Introducción

A lo largo del curso, aprendimos que para poder confiar en los resultados de una simulación, es crucial analizar la **estabilidad** y la **precisión** de los métodos numéricos utilizados.

Para esto, analizábamos la relación entre los **autovalores** de la matriz **A** del sistema continuo original y los de la matriz **F** de la aproximación numérica resultante.

Introducción

A lo largo del curso, aprendimos que para poder confiar en los resultados de una simulación, es crucial analizar la **estabilidad** y la **precisión** de los métodos numéricos utilizados.

Para esto, analizábamos la relación entre los **autovalores** de la matriz **A** del sistema continuo original y los de la matriz **F** de la aproximación numérica resultante.

Lamentablemente no podemos aplicar esta idea en el caso del método de QSS, ya que el **sistema discreto** resultante no tiene ninguna matriz **F**.

Introducción

A lo largo del curso, aprendimos que para poder confiar en los resultados de una simulación, es crucial analizar la **estabilidad** y la **precisión** de los métodos numéricos utilizados.

Para esto, analizábamos la relación entre los **autovalores** de la matriz **A** del sistema continuo original y los de la matriz **F** de la aproximación numérica resultante.

Lamentablemente no podemos aplicar esta idea en el caso del método de QSS, ya que el **sistema discreto** resultante no tiene ninguna matriz **F**.

Sin embargo, vimos que la aproximación QSS admite, en su definición, una representación como **sistema de tiempo continuo**. Por lo tanto, intentaremos aprovechar esto para estudiar sus propiedades de estabilidad y precisión.

Representación Perturbada de QSS

Dado el **sistema continuo**:

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

su **aproximación QSS** puede escribirse como:

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

donde $\mathbf{x}(t)$ y $\mathbf{q}(t)$ se relacionan **componente a componente** mediante funciones de cuantificación con histéresis.

Representación Perturbada de QSS

Dado el **sistema continuo**:

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

su **aproximación QSS** puede escribirse como:

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

donde $\mathbf{x}(t)$ y $\mathbf{q}(t)$ se relacionan **componente a componente** mediante funciones de cuantificación con histéresis.

Definiendo $\Delta\mathbf{x}(t) = \mathbf{q}(t) - \mathbf{x}(t)$, el QSS de la Ec.(2) puede reescribirse como:

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

Representación Perturbada de QSS

Dado el **sistema continuo**:

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

su **aproximación QSS** puede escribirse como:

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

donde $\mathbf{x}(t)$ y $\mathbf{q}(t)$ se relacionan **componente a componente** mediante funciones de cuantificación con histéresis.

Definiendo $\Delta\mathbf{x}(t) = \mathbf{q}(t) - \mathbf{x}(t)$, el QSS de la Ec.(2) puede reescribirse como:

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

El modelo de simulación (3) es una **versión perturbada** del sistema original (1).

Representación Perturbada de QSS

Dado el **sistema continuo**:

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

su **aproximación QSS** puede escribirse como:

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

donde $\mathbf{x}(t)$ y $\mathbf{q}(t)$ se relacionan **componente a componente** mediante funciones de cuantificación con histéresis.

Definiendo $\Delta\mathbf{x}(t) = \mathbf{q}(t) - \mathbf{x}(t)$, el QSS de la Ec.(2) puede reescribirse como:

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

El modelo de simulación (3) es una **versión perturbada** del sistema original (1).

Además, llamando ΔQ_i al quantum, la perturbación resulta acotada:

$$|\Delta x_i| = |q_i(t) - x_i(t)| \leq \max(\Delta Q_i, \varepsilon_i) \quad (4)$$

Convergencia, Precisión y Estabilidad en QSS

Podemos estudiar la **estabilidad** y la **precisión** analizando los **efectos de perturbaciones acotadas**.

Convergencia, Precisión y Estabilidad en QSS

Podemos estudiar la **estabilidad** y la **precisión** analizando los **efectos de perturbaciones acotadas**.

En base a esto, está demostrada la **convergencia** del método para sistemas no lineales. Es decir, cuando el **quantum** y la **histéresis** tienden a **cero**, las soluciones de QSS tienden a las soluciones del sistema original.

Convergencia, Precisión y Estabilidad en QSS

Podemos estudiar la **estabilidad** y la **precisión** analizando los **efectos de perturbaciones acotadas**.

En base a esto, está demostrada la **convergencia** del método para sistemas no lineales. Es decir, cuando el **quantum** y la **histéresis** tienden a **cero**, las soluciones de QSS tienden a las soluciones del sistema original.

También está demostrada una propiedad de **estabilidad** en la simulación de **sistemas no lineales**. Cuando el sistema original tiene un punto de equilibrio **asintóticamente estable**, puede encontrarse un quantum que garantiza que las soluciones de QSS están **finalmente acotadas** en una región en torno a dicho punto de equilibrio.

Convergencia, Precisión y Estabilidad en QSS

Podemos estudiar la **estabilidad** y la **precisión** analizando los **efectos de perturbaciones acotadas**.

En base a esto, está demostrada la **convergencia** del método para sistemas no lineales. Es decir, cuando el **quantum** y la **histéresis** tienden a **cero**, las soluciones de QSS tienden a las soluciones del sistema original.

También está demostrada una propiedad de **estabilidad** en la simulación de **sistemas no lineales**. Cuando el sistema original tiene un punto de equilibrio **asintóticamente estable**, puede encontrarse un quantum que garantiza que las soluciones de QSS están **finalmente acotadas** en una región en torno a dicho punto de equilibrio.

De todas maneras, estas son propiedades de más importancia **teórica** que práctica, ya que no brindan información **cuantitativa**. Como en los demás métodos, los resultados más interesantes se deducen para los **sistemas lineales y estacionarios**.

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta \mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \boldsymbol{\Delta Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz **A** es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta \mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- ▶ Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.
- ▶ $\mathbf{A} = \mathbf{V} \cdot \boldsymbol{\Lambda} \cdot \mathbf{V}^{-1}$ es la descomposición espectral de **A**.

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta \mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- ▶ Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.
- ▶ $\mathbf{A} = \mathbf{V} \cdot \boldsymbol{\Lambda} \cdot \mathbf{V}^{-1}$ es la descomposición espectral de $\mathbf{A}$.

$$\text{▶ } \Delta \mathbf{Q} = \begin{bmatrix} \max(\Delta Q_1, \varepsilon_1) \\ \vdots \\ \max(\Delta Q_n, \varepsilon_n) \end{bmatrix}.$$

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta \mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- ▶ Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.
- ▶ $\mathbf{A} = \mathbf{V} \cdot \boldsymbol{\Lambda} \cdot \mathbf{V}^{-1}$ es la descomposición espectral de $\mathbf{A}$.

$$\Delta \mathbf{Q} = \begin{bmatrix} \max(\Delta Q_1, \varepsilon_1) \\ \vdots \\ \max(\Delta Q_n, \varepsilon_n) \end{bmatrix}.$$

Notar que:

- ▶ El método de QSS tiene una **cota de error global calculable**.

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta \mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- ▶ Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.
- ▶ $\mathbf{A} = \mathbf{V} \cdot \boldsymbol{\Lambda} \cdot \mathbf{V}^{-1}$ es la descomposición espectral de $\mathbf{A}$.

$$\text{▶ } \Delta \mathbf{Q} = \begin{bmatrix} \max(\Delta Q_1, \varepsilon_1) \\ \vdots \\ \max(\Delta Q_n, \varepsilon_n) \end{bmatrix}.$$

Notar que:

- ▶ El método de QSS tiene una **cota de error global calculable**.
- ▶ La solución numérica, para cualquier **quantum**, es siempre **prácticamente estable**.

QSS en Sistemas Lineales y Estacionarios.

Consideremos un sistema **lineal y estacionario** y su aproximación QSS:

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t); \quad \dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t) \quad (5)$$

Si la matriz $\mathbf{A}$ es Hurwitz, llamando $\mathbf{x}_a(t)$ y $\mathbf{x}(t)$ a las soluciones del sistema continuo y del QSS respectivamente, con $\mathbf{x}(t_0) = \mathbf{x}_a(t_0)$, se cumplirá que:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\operatorname{Re}\{\boldsymbol{\Lambda}\}^{-1} \cdot \boldsymbol{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta\mathbf{Q}; \quad \forall t \geq t_0 \quad (6)$$

donde:

- ▶ Los operadores “ $\leq$ ”, “ $|\cdot|$ ” y “ $\operatorname{Re}\{\cdot\}$ ” actúan componente a componente.
- ▶ $\mathbf{A} = \mathbf{V} \cdot \boldsymbol{\Lambda} \cdot \mathbf{V}^{-1}$ es la descomposición espectral de $\mathbf{A}$.

$$\Delta\mathbf{Q} = \begin{bmatrix} \max(\Delta Q_1, \varepsilon_1) \\ \vdots \\ \max(\Delta Q_n, \varepsilon_n) \end{bmatrix}.$$

Notar que:

- ▶ El método de QSS tiene una **cota de error global calculable**.
- ▶ La solución numérica, para cualquier **quantum**, es siempre **prácticamente estable**.
- ▶ Estas características son notables, ya que QSS es un método **explícito**.

La solución brindada por el método de QSS suele tener **oscilaciones** finales, cuya **frecuencia** depende de la **histéresis**. Por ejemplo, la simulación del sistema escalar

con $\Delta Q = 1$ y distintos valores de ε da:

La solución brindada por el método de QSS suele tener **oscilaciones** finales, cuya **frecuencia** depende de la **histéresis**. Por ejemplo, la simulación del sistema escalar

con $\Delta Q = 1$ y distintos valores de ε da:

Soluciones de (7)



La solución brindada por el método de QSS suele tener **oscilaciones** finales, cuya **frecuencia** depende de la **histéresis**. Por ejemplo, la simulación del sistema escalar

con $\Delta Q = 1$ y distintos valores de ε da:

- ▶ Para **disminuir la frecuencia** (y el número de pasos), conviene utilizar ε grande.
- ▶ Como la **cota de error** depende de $\max(\Delta Q, \varepsilon)$, la mejor elección es tomar $\Delta Q = \varepsilon$.

La solución brindada por el método de QSS suele tener **oscilaciones** finales, cuya **frecuencia** depende de la **histéresis**. Por ejemplo, la simulación del sistema escalar

con $\Delta Q = 1$ y distintos valores de ϵ da:

- ▶ Para **disminuir la frecuencia** (y el número de pasos), conviene utilizar ϵ grande.
- ▶ Como la **cota de error** depende de $\max(\Delta Q, \epsilon)$, la mejor elección es tomar $\Delta Q = \epsilon$.
- ▶ La cota de error (6) depende **linealmente** de ΔQ . Luego el **quantum** debe elegirse **proporcional** al error tolerado en cada variable.

La solución brindada por el método de QSS suele tener **oscilaciones** finales, cuya **frecuencia** depende de la **histéresis**. Por ejemplo, la simulación del sistema escalar

con $\Delta Q = 1$ y distintos valores de ε da:

- ▶ Para **disminuir la frecuencia** (y el número de pasos), conviene utilizar ϵ grande.
- ▶ Como la **cota de error** depende de $\max(\Delta Q, \epsilon)$, la mejor elección es tomar $\Delta Q = \epsilon$.
- ▶ La cota de error (6) depende **linealmente** de ΔQ . Luego el **quantum** debe elegirse **proporcional** al error tolerado en cada variable.
- ▶ Una **regla práctica** es tomar el quantum proporcional a la **amplitud** de cada señal (si se conoce el orden de la misma).

Señales de Entrada en el Método de QSS

Para incorporar **señales de entrada $u(t)$** a la simulación con QSS, necesitamos simplemente construir **modelos DEVS** que las generen y acoplar dichos modelos DEVS con los **integradores cuantificados** y **funciones estáticas**.

Señales de Entrada en el Método de QSS

Para incorporar **señales de entrada $u(t)$** a la simulación con QSS, necesitamos simplemente construir **modelos DEVS** que las generen y acoplar dichos modelos DEVS con los **integradores cuantificados** y **funciones estáticas**.

Programar un modelo DEVS que produzca una señal de entrada **seccionalmente constante** es trivial. En particular, las librerías de PowerDEVS tienen varias **fuentes** de este tipo: función escalón, onda cuadrada, función pulso, etc.

Señales de Entrada en el Método de QSS

Para incorporar **señales de entrada $u(t)$** a la simulación con QSS, necesitamos simplemente construir **modelos DEVS** que las generen y acoplar dichos modelos DEVS con los **integradores cuantificados** y **funciones estáticas**.

Programar un modelo DEVS que produzca una señal de entrada **seccionalmente constante** es trivial. En particular, las librerías de PowerDEVS tienen varias **fuentes** de este tipo: función escalón, onda cuadrada, función pulso, etc.

También pueden incorporarse **entradas arbitrarias** de forma muy simple, **aproximándolas** por señales seccionalmente constantes. PowerDEVS tiene fuentes que producen señales senoidales, rampas, ondas triangulares, etc.

Señales de Entrada en el Método de QSS

Para incorporar **señales de entrada** $u(t)$ a la simulación con QSS, necesitamos simplemente construir **modelos DEVS** que las generen y acoplar dichos modelos DEVS con los **integradores cuantificados** y **funciones estáticas**.

Programar un modelo DEVS que produzca una señal de entrada **seccionalmente constante** es trivial. En particular, las librerías de PowerDEVS tienen varias **fuentes** de este tipo: función escalón, onda cuadrada, función pulso, etc.

También pueden incorporarse **entradas arbitrarias** de forma muy simple, **aproximándolas** por señales seccionalmente constantes. PowerDEVS tiene fuentes que producen señales senoidales, rampas, ondas triangulares, etc.

Sin embargo, al aproximar una señal de entrada con una trayectoria seccionalmente constante, se agrega una nueva **fuentes de error**. En este caso, se modifica la expresión de la **cota de error global** (6) con la adición de un nuevo término:

$$|\mathbf{x}(t) - \mathbf{x}_a(t)| \leq |\mathbf{V}| \cdot |\Re\{\mathbf{\Lambda}\}^{-1} \cdot \mathbf{\Lambda}| \cdot |\mathbf{V}^{-1}| \cdot \Delta Q + |\mathbf{V}| \cdot |\Re\{\mathbf{\Lambda}\}^{-1} \cdot \mathbf{V}^{-1} \cdot \mathbf{B}| \cdot \Delta u$$

Costo vs. Precisión en QSS

Aunque el método de QSS ofrece varias cualidades atractivas, su **principal limitación** radica en que realiza sólo una **aproximación de primer orden**.

Costo vs. Precisión en QSS

Aunque el método de QSS ofrece varias cualidades atractivas, su **principal limitación** radica en que realiza sólo una **aproximación de primer orden**.

La Ec.(6) muestra que el **error es proporcional al quantum**. Por otro lado, puede verse fácilmente que el **número de pasos** en una variable (en ausencia de oscilaciones) es la **división entre la amplitud de la señal y el quantum**.

Costo vs. Precisión en QSS

Aunque el método de QSS ofrece varias cualidades atractivas, su **principal limitación** radica en que realiza sólo una **aproximación de primer orden**.

La Ec.(6) muestra que el **error es proporcional al quantum**. Por otro lado, puede verse fácilmente que el **número de pasos** en una variable (en ausencia de oscilaciones) es la **división entre la amplitud de la señal y el quantum**.

Esto quiere decir que el número de pasos es **inversamente proporcional al error**. Por lo tanto, si queremos aumentar **100** veces la **precisión**, deberemos reducir **100** veces el **quantum**, lo que incrementará **100** veces el **número de pasos**.

Costo vs. Precisión en QSS

Aunque el método de QSS ofrece varias cualidades atractivas, su **principal limitación** radica en que realiza sólo una **aproximación de primer orden**.

La Ec.(6) muestra que el **error es proporcional al quantum**. Por otro lado, puede verse fácilmente que el **número de pasos** en una variable (en ausencia de oscilaciones) es la **división entre la amplitud de la señal y el quantum**.

Esto quiere decir que el número de pasos es **inversamente proporcional al error**. Por lo tanto, si queremos aumentar **100** veces la **precisión**, deberemos reducir **100** veces el **quantum**, lo que incrementará **100** veces el **número de pasos**.

Esto no sirve si queremos una buena precisión. Evidentemente, necesitamos **métodos de mayor orden de aproximación**.

Cuantificador de Primer Orden

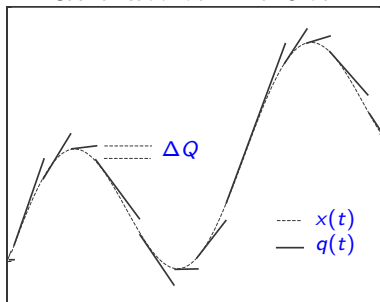
El **cuantificador** utilizado en QSS, con $\Delta Q = \varepsilon$, produce una señal $q(t)$ **seccionalmente constante**, que cambia cada vez que $|q(t) - x(t)| = \Delta Q$.

Cuantificador de Primer Orden

El **cuantificador** utilizado en QSS, con $\Delta Q = \varepsilon$, produce una señal $q(t)$ **seccionalmente constante**, que cambia cada vez que $|q(t) - x(t)| = \Delta Q$.

Una forma de **incrementar el orden** de la aproximación, es usar un cuantificador con salida **seccionalmente lineal**, que cambie cada vez que $|q(t) - x(t)| = \Delta Q$.

Cuantificador de Primer Orden



El **cuantificador** utilizado en QSS, con $\Delta Q = \varepsilon$, produce una señal $q(t)$ **seccionalmente constante**, que cambia cada vez que $|q(t) - x(t)| = \Delta Q$.

Una forma de **incrementar el orden** de la aproximación, es usar un cuantificador con salida **seccionalmente lineal**, que cambie cada vez que $|q(t) - x(t)| = \Delta Q$.

$\cdots x(t)$
 $\text{— } q(t)$

Diremos que $x(t)$ y $q(t)$ se relacionan por una función de cuantificación de primer orden si:

$$q(t) = \begin{cases} x(t), & \text{si } t = t_0 \vee |q(t^-) - x(t^-)| = \Delta Q \\ q(t_j) + m_j \cdot (t - t_j), & \text{en otro caso} \end{cases}$$

La secuencia $t_0, \dots, t_j, \dots$ se define como:

$$t_{j+1} = \min_{t > t_j \wedge |x(t_j) + m_j \cdot (t - t_j) - x(t)| = \Delta Q} (t)$$

y las pendientes son $m_0 = 0$, $m_j = \dot{x}(t_j^-)$,
para $j = 1, \dots, k, \dots$

Método de QSS de Segundo Orden (QSS2)

El *método de QSS de Segundo Orden*, o *método de QSS2*, se define de manera idéntica al método de QSS, sólo que las variables de estado x_i y las variables cuantificadas q_i ahora se relacionan mediante **funciones de cuantificación de primer orden**.

Método de QSS de Segundo Orden (QSS2)

El *método de QSS de Segundo Orden*, o *método de QSS2*, se define de manera idéntica al método de QSS, sólo que las variables de estado x_i y las variables cuantificadas q_i ahora se relacionan mediante **funciones de cuantificación de primer orden**.

Es decir, dado el sistema continuo:

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

su **aproximación QSS2** puede escribirse como:

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

Método de QSS de Segundo Orden (QSS2)

El *método de QSS de Segundo Orden*, o *método de QSS2*, se define de manera idéntica al método de QSS, sólo que las variables de estado x_i y las variables cuantificadas q_i ahora se relacionan mediante **funciones de cuantificación de primer orden**.

Es decir, dado el sistema continuo:

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

su **aproximación QSS2** puede escribirse como:

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

Ahora aceptaremos que las **entradas $u_j(t)$** tengan trayectorias **seccionalmente lineales**.

Método de QSS de Segundo Orden (QSS2)

El *método de QSS de Segundo Orden*, o *método de QSS2*, se define de manera idéntica al método de QSS, sólo que las variables de estado x_i y las variables cuantificadas q_i ahora se relacionan mediante **funciones de cuantificación de primer orden**.

Es decir, dado el sistema continuo:

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

su **aproximación QSS2** puede escribirse como:

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

Ahora aceptaremos que las **entradas $u_j(t)$** tengan trayectorias **seccionalmente lineales**.

También podremos tratar muy fácilmente con **sistemas no estacionarios** de la forma:

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

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.
- ▶ En el caso lineal y estacionario, las derivadas de las variables de estado $\dot{x}_i(t)$ serán seccionalmente lineales.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.
- ▶ En el caso lineal y estacionario, las derivadas de las variables de estado $\dot{x}_i(t)$ serán seccionalmente lineales.
- ▶ En tal caso, las variables de estado $x_i(t)$ serán seccionalmente parabólicas.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.
- ▶ En el caso lineal y estacionario, las derivadas de las variables de estado $\dot{x}_i(t)$ serán seccionalmente lineales.
- ▶ En tal caso, las variables de estado $x_i(t)$ serán seccionalmente parabólicas.
- ▶ En el caso general (no lineal) ni las variables de estado ni sus derivadas seguirán una forma particular de trayectoria.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.
- ▶ En el caso lineal y estacionario, las derivadas de las variables de estado $\dot{x}_i(t)$ serán seccionalmente lineales.
- ▶ En tal caso, las variables de estado $x_i(t)$ serán seccionalmente parabólicas.
- ▶ En el caso general (no lineal) ni las variables de estado ni sus derivadas seguirán una forma particular de trayectoria.
- ▶ En este caso, aproximaremos las derivadas por trayectorias seccionalmente lineales. Por lo tanto, las variables de estado serán seccionalmente parabólicas.

Trayectorias en el Método de QSS2

De acuerdo a la definición de QSS2, las trayectorias en el sistema aproximado tendrán la siguiente forma:

- ▶ Las variables cuantificadas $q_i(t)$ serán seccionalmente lineales.
- ▶ Las entradas $u_j(t)$ serán seccionalmente lineales.
- ▶ En el caso lineal y estacionario, las derivadas de las variables de estado $\dot{x}_i(t)$ serán seccionalmente lineales.
- ▶ En tal caso, las variables de estado $x_i(t)$ serán seccionalmente parabólicas.
- ▶ En el caso general (no lineal) ni las variables de estado ni sus derivadas seguirán una forma particular de trayectoria.
- ▶ En este caso, aproximaremos las derivadas por trayectorias seccionalmente lineales. Por lo tanto, las variables de estado serán seccionalmente parabólicas.

En los casos no lineales los resultados de simulación no coincidirán exactamente con la definición del método de QSS2 que dimos antes.

Representación DEVS de QSS2.

La idea para obtener un modelo DEVS equivalente a un QSS2 es la misma que en QSS, es decir, dividir al sistema en **funciones estáticas** e **integradores cuantificados**. Las diferencias son las siguientes:

Representación DEVS de QSS2.

La idea para obtener un modelo DEVS equivalente a un QSS2 es la misma que en QSS, es decir, dividir al sistema en **funciones estáticas** e **integradores cuantificados**. Las diferencias son las siguientes:

- ▶ Como las trayectorias son **seccionalmente lineales**, cada evento debe llevar dos números: el **valor** y la **pendiente** de cada sección de trayectoria.

Representación DEVS de QSS2.

La idea para obtener un modelo DEVS equivalente a un QSS2 es la misma que en QSS, es decir, dividir al sistema en **funciones estáticas** e **integradores cuantificados**.

Las diferencias son las siguientes:

- ▶ Como las trayectorias son **seccionalmente lineales**, cada evento debe llevar dos números: el **valor** y la **pendiente** de cada sección de trayectoria.
- ▶ Consecuentemente, los modelos DEVS de las **funciones estáticas** deben tener en cuenta los **valores** y **pendientes** de sus entradas y con estos calcular los **valores** y **pendientes** de las derivadas de los estados.

Representación DEVS de QSS2.

La idea para obtener un modelo DEVS equivalente a un QSS2 es la misma que en QSS, es decir, dividir al sistema en **funciones estáticas** e **integradores cuantificados**. Las diferencias son las siguientes:

- ▶ Como las trayectorias son **seccionalmente lineales**, cada evento debe llevar dos números: el **valor** y la **pendiente** de cada sección de trayectoria.
- ▶ Consecuentemente, los modelos DEVS de las **funciones estáticas** deben tener en cuenta los **valores** y **pendientes** de sus entradas y con estos calcular los **valores** y **pendientes** de las derivadas de los estados.
- ▶ Los modelos DEVS de los **integradores cuantificados** deben también tener en cuenta las pendientes. Como las trayectorias de estado son **seccionalmente parabólicas**, para calcular el tiempo en el que $|x_i(t) - q_i(t)| = \Delta Q_i$, deben resolver una ecuación cuadrática en cada paso.

Representación DEVS de QSS2.

La idea para obtener un modelo DEVS equivalente a un QSS2 es la misma que en QSS, es decir, dividir al sistema en **funciones estáticas** e **integradores cuantificados**. Las diferencias son las siguientes:

- ▶ Como las trayectorias son **seccionalmente lineales**, cada evento debe llevar dos números: el **valor** y la **pendiente** de cada sección de trayectoria.
- ▶ Consecuentemente, los modelos DEVS de las **funciones estáticas** deben tener en cuenta los **valores** y **pendientes** de sus entradas y con estos calcular los **valores** y **pendientes** de las derivadas de los estados.
- ▶ Los modelos DEVS de los **integradores cuantificados** deben también tener en cuenta las pendientes. Como las trayectorias de estado son **seccionalmente parabólicas**, para calcular el tiempo en el que $|x_i(t) - q_i(t)| = \Delta Q_i$, deben resolver una ecuación cuadrática en cada paso.

Más allá de estas diferencias, el modelo DEVS resultante tendrá exactamente la misma estructura con QSS y con QSS2.

Simulación con QSS2 en PowerDEVS

PowerDEVS tiene incorporados en sus librerías el modelo DEVS del **integrador cuantificado** y muchas **funciones estáticas** y **fuentes** del método de QSS2.

Simulación con QSS2 en PowerDEVS

PowerDEVS tiene incorporados en sus librerías el modelo DEVS del **integrador cuantificado** y muchas **funciones estáticas** y **fuentes** del método de QSS2.

Más precisamente, hay un bloque de **integrador cuantificado** que permite elegir el método (QSS, QSS2 o QSS3) como un parámetro más.

Simulación con QSS2 en PowerDEVS

PowerDEVS tiene incorporados en sus librerías el modelo DEVS del **integrador cuantificado** y muchas **funciones estáticas** y **fuentes** del método de QSS2.

Más precisamente, hay un bloque de **integrador cuantificado** que permite elegir el método (QSS, QSS2 o QSS3) como un parámetro más.

Los modelos DEVS de las **fuentes** también permiten elegir el orden de método.

Simulación con QSS2 en PowerDEVS

PowerDEVS tiene incorporados en sus librerías el modelo DEVS del **integrador cuantificado** y muchas **funciones estáticas** y **fuentes** del método de QSS2.

Más precisamente, hay un bloque de **integrador cuantificado** que permite elegir el método (QSS, QSS2 o QSS3) como un parámetro más.

Los modelos DEVS de las **fuentes** también permiten elegir el orden de método.

En el caso de las **funciones estáticas**, los bloques detectan el método en uso y calculan sus salidas de manera acorde.

Simulación con QSS2 en PowerDEVS

PowerDEVS tiene incorporados en sus librerías el modelo DEVS del **integrador cuantificado** y muchas **funciones estáticas** y **fuentes** del método de QSS2.

Más precisamente, hay un bloque de **integrador cuantificado** que permite elegir el método (QSS, QSS2 o QSS3) como un parámetro más.

Los modelos DEVS de las **fuentes** también permiten elegir el orden de método.

En el caso de las **funciones estáticas**, los bloques detectan el método en uso y calculan sus salidas de manera acorde.

Por lo tanto, los bloques son los mismos para todos los métodos. Simplemente puede armarse el **Diagrama de Bloques** y elegir el método cambiando los parámetros correspondientes.

Estabilidad y Precisión de QSS2

La aproximación QSS2 de un sistema **lineal y estacionario**

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

se escribe igual que la aproximación QSS:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

Estabilidad y Precisión de QSS2

La aproximación QSS2 de un sistema **lineal y estacionario**

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

se escribe igual que la aproximación QSS:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

En QSS2 es también cierto que

$$|q_i(t) - x_i(t)| \leq \Delta Q_i \quad \forall t \geq t_0$$

Estabilidad y Precisión de QSS2

La aproximación QSS2 de un sistema **lineal y estacionario**

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

se escribe igual que la aproximación QSS:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

En QSS2 es también cierto que

$$|q_i(t) - x_i(t)| \leq \Delta Q_i \quad \forall t \geq t_0$$

Por lo tanto, QSS y QSS2 tienen la misma **representación perturbada** (en sistemas lineales y estacionarios).

Estabilidad y Precisión de QSS2

La aproximación QSS2 de un sistema **lineal y estacionario**

$$\dot{\mathbf{x}}_a(t) = \mathbf{A} \cdot \mathbf{x}_a(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

se escribe igual que la aproximación QSS:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{q}(t) + \mathbf{B} \cdot \mathbf{u}(t)$$

En QSS2 es también cierto que

$$|q_i(t) - x_i(t)| \leq \Delta Q_i \quad \forall t \geq t_0$$

Por lo tanto, QSS y QSS2 tienen la misma **representación perturbada** (en sistemas lineales y estacionarios).

Como la fórmula de la **cota de error global** (6) de QSS se deduce de esta representación, resulta que la aproximación QSS2 también cumple con dicha cota. En definitiva, QSS2 y QSS tienen **las mismas propiedades** de error y estabilidad en **sistemas lineales y estacionarios**.

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. ¿Cuál es entonces la ventaja de utilizar QSS2?

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. ¿Cuál es entonces la ventaja de utilizar **QSS2**?

La respuesta es simple. En QSS, habíamos visto que al reducir **100** veces el **quantum**, el **número de pasos** aumentaba **100** veces.

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. ¿Cuál es entonces la ventaja de utilizar QSS2?

La respuesta es simple. En QSS, habíamos visto que al reducir **100** veces el **quantum**, el **número de pasos** aumentaba **100** veces.

En QSS2, puede verse fácilmente que reducir **100** veces el **quantum** implica que el **número de pasos** aumente sólo **10** veces.

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. ¿Cuál es entonces la ventaja de utilizar QSS2?

La respuesta es simple. En QSS, habíamos visto que al reducir **100** veces el **quantum**, el **número de pasos** aumentaba **100** veces.

En QSS2, puede verse fácilmente que reducir **100** veces el **quantum** implica que el **número de pasos** aumente sólo **10** veces.

Más precisamente, en QSS2 el costo es **inversamente proporcional** a la **raíz cuadrada** del error tolerado.

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. **¿Cuál es entonces la ventaja de utilizar QSS2?**

La respuesta es simple. En QSS, habíamos visto que al reducir **100** veces el **quantum**, el **número de pasos** aumentaba **100** veces.

En QSS2, puede verse fácilmente que reducir **100** veces el **quantum** implica que el **número de pasos** aumente sólo **10** veces.

Más precisamente, en QSS2 el costo es **inversamente proporcional** a la **raíz cuadrada** del error tolerado.

Por ejemplo, si queremos mejorar la precisión de una simulación en un factor de **10000**, en QSS el costo aumentará **10000** veces mientras que en QSS2 sólo aumentará **100** veces.

Costo vs. Precisión en QSS2

Del análisis anterior resulta que al utilizar QSS y QSS2 con **el mismo quantum**, obtendremos resultados con **similar precisión**. ¿Cuál es entonces la ventaja de utilizar QSS2?

La respuesta es simple. En QSS, habíamos visto que al reducir **100** veces el **quantum**, el **número de pasos** aumentaba **100** veces.

En QSS2, puede verse fácilmente que reducir **100** veces el **quantum** implica que el **número de pasos** aumente sólo **10** veces.

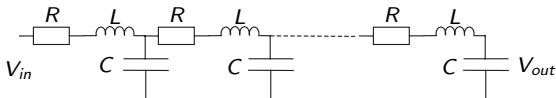
Más precisamente, en QSS2 el costo es **inversamente proporcional** a la **raíz cuadrada** del error tolerado.

Por ejemplo, si queremos mejorar la precisión de una simulación en un factor de **10000**, en QSS el costo aumentará **10000** veces mientras que en QSS2 sólo aumentará **100** veces.

El **costo adicional** de tener en cuenta las **pendientes** se paga muy rápidamente al aumentar un poco la **precisión requerida**.

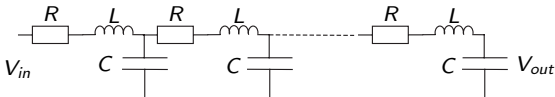
Un Ejemplo Ilustrativo sobre QSS2

El circuito de la figura muestra un **modelo concentrado** de una línea de transmisión. En nuestro caso, el mismo representa una sección de una pista de un **circuito integrado** que transmite datos a muy **alta frecuencia**.



Un Ejemplo Ilustrativo sobre QSS2

El circuito de la figura muestra un **modelo concentrado** de una línea de transmisión. En nuestro caso, el mismo representa una sección de una pista de un **circuito integrado** que transmite datos a muy **alta frecuencia**.



Considerando 5 secciones, resulta un sistema **lineal y estacionario** de orden 10, con una **matriz dispersa** de sistema:

$$A = \begin{pmatrix} -R/L & -1/L & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/C & 0 & -1/C & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/L & -R/L & -1/L & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/C & 0 & -1/C & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/C & 0 \end{pmatrix}$$

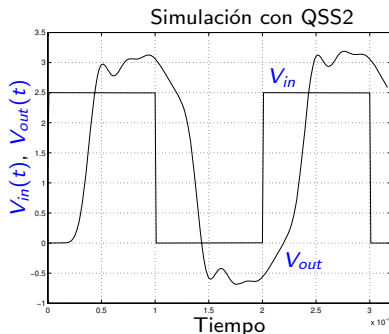
Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

El resultado fue:

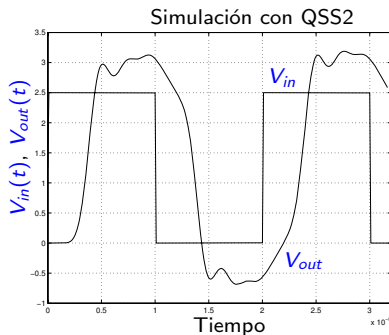


Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

El resultado fue:

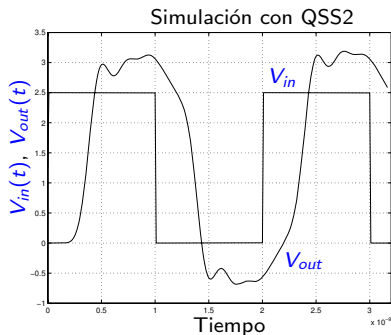
- La simulación tomó 2536 pasos (entre 198 y 319 pasos en **cada integrador**).



Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

El resultado fue:

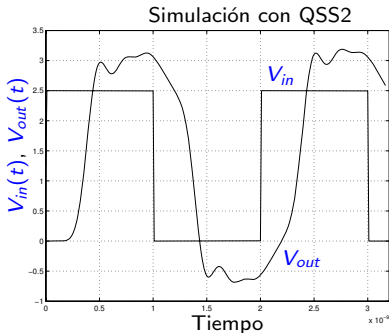


- ▶ La simulación tomó 2536 pasos (entre 198 y 319 pasos en **cada integrador**).
- ▶ Aunque el número de pasos parece grande, cada paso sólo produce cálculos **escalares** en dos o tres integradores. Esto se debe a la **dispersión** de la matriz **A**.

Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

El resultado fue:

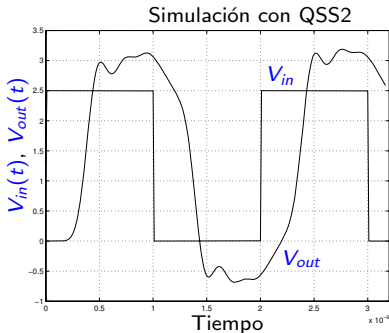


- ▶ La simulación tomó **2536** pasos (entre **198** y **319** pasos en **cada integrador**).
- ▶ Aunque el número de pasos parece grande, cada paso sólo produce cálculos **escalares** en dos o tres integradores. Esto se debe a la **dispersión** de la matriz **A**.
- ▶ Como la entrada es **seccionalmente lineal**, la Ec.(6) nos da una cota teórica del **error global** en toda la simulación. Para la variable V_{out} la cota de error es **250mV** (es algo **conservadora**).

Un Ejemplo Ilustrativo sobre QSS2 II

Utilizando PowerDEVS, construimos el **Diagrama de Bloques** y simulamos con QSS2 la respuesta de dicho sistema a una entrada **trapezoidal** (forma típica de las señales en los circuitos integrados). Utilizamos un quantum de $4mV$ en las variables de tensión y $10\mu A$ en las de corriente.

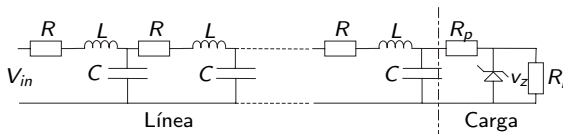
El resultado fue:



- ▶ La simulación tomó **2536** pasos (entre **198** y **319** pasos en **cada integrador**).
- ▶ Aunque el número de pasos parece grande, cada paso sólo produce cálculos **escalares** en dos o tres integradores. Esto se debe a la **dispersión** de la matriz **A**.
- ▶ Como la entrada es **seccionalmente lineal**, la Ec.(6) nos da una cota teórica del **error global** en toda la simulación. Para la variable V_{out} la cota de error es **$250mV$** (es algo **conservadora**).
- ▶ Disminuyendo **100** veces el quantum, corroboramos que el **número de pasos** aumentó aproximadamente **10** veces. La nueva cota teórica de error en V_{out} es de **$2.5mV$** .

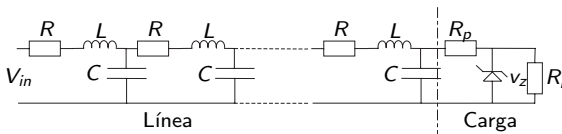
Simulación de DAEs con los Métodos de QSS

La figura muestra el circuito del ejemplo anterior con una **protección** contra **sobretensiones**, formada por una resistencia y un **diodo Zener**.



Simulación de DAEs con los Métodos de QSS

La figura muestra el circuito del ejemplo anterior con una **protección** contra **sobretensiones**, formada por una resistencia y un **diodo Zener**.



El nuevo sistema de ecuaciones resulta en una **DAE**:

$$\frac{di_1}{dt} = \frac{1}{L} \cdot v_{in} - \frac{R}{L} \cdot i_1 - \frac{1}{L} \cdot u_1$$

$$\frac{du_1}{dt} = \frac{1}{C} \cdot i_1 - \frac{1}{C} \cdot i_2$$

$$\vdots$$

$$\frac{di_5}{dt} = \frac{1}{L} \cdot u_4 - \frac{R}{L} \cdot i_5 - \frac{1}{L} \cdot u_5$$

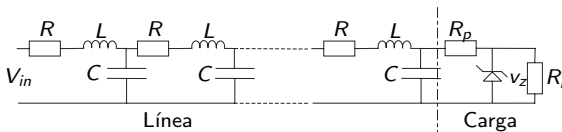
$$\frac{du_5}{dt} = \frac{1}{C} \cdot i_5 - \frac{1}{R_p C} \cdot (u_5 - v_z)$$

$$\frac{1}{R_p} \cdot u_5 - \left(\frac{1}{R_p} + \frac{1}{R_l} \right) \cdot v_z - \frac{l_0}{1 - (v_z/v_{br})^m} = 0$$

donde v_z es una **variable algebraica** que cumple:

Simulación de DAEs con los Métodos de QSS

La figura muestra el circuito del ejemplo anterior con una **protección** contra **sobretensiones**, formada por una resistencia y un **diodo Zener**.



El nuevo sistema de ecuaciones resulta en una **DAE**:

$$\frac{di_1}{dt} = \frac{1}{L} \cdot v_{in} - \frac{R}{L} \cdot i_1 - \frac{1}{L} \cdot u_1$$

$$\frac{du_1}{dt} = \frac{1}{C} \cdot i_1 - \frac{1}{C} \cdot i_2$$

$$\vdots$$

$$\frac{di_5}{dt} = \frac{1}{L} \cdot u_4 - \frac{R}{L} \cdot i_5 - \frac{1}{L} \cdot u_5$$

$$\frac{du_5}{dt} = \frac{1}{C} \cdot i_5 - \frac{1}{R_p C} \cdot (u_5 - v_z)$$

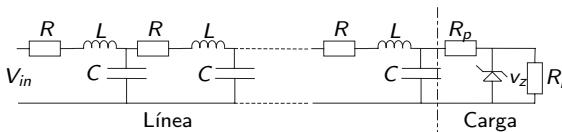
$$\frac{1}{R_p} \cdot u_5 - \left(\frac{1}{R_p} + \frac{1}{R_l} \right) \cdot v_z - \frac{l_0}{1 - (v_z/v_{br})^m} = 0$$

Para usar QSS o QSS2 simplemente cambiamos i_j y u_j por q_{ij} y q_{uj} . El problema es el cálculo de v_z .

donde v_z es una **variable algebraica** que cumple:

Simulación de DAEs con los Métodos de QSS

La figura muestra el circuito del ejemplo anterior con una **protección** contra **sobretensiones**, formada por una resistencia y un **diodo Zener**.



El nuevo sistema de ecuaciones resulta en una **DAE**:

$$\frac{di_1}{dt} = \frac{1}{L} \cdot v_{in} - \frac{R}{L} \cdot i_1 - \frac{1}{L} \cdot u_1$$

$$\frac{du_1}{dt} = \frac{1}{C} \cdot i_1 - \frac{1}{C} \cdot i_2$$

$$\vdots$$

$$\frac{di_5}{dt} = \frac{1}{L} \cdot u_4 - \frac{R}{L} \cdot i_5 - \frac{1}{L} \cdot u_5$$

$$\frac{du_5}{dt} = \frac{1}{C} \cdot i_5 - \frac{1}{R_p C} \cdot (u_5 - v_z)$$

$$\frac{1}{R_p} \cdot u_5 - \left(\frac{1}{R_p} + \frac{1}{R_l} \right) \cdot v_z - \frac{l_0}{1 - (v_z/v_{br})^m} = 0$$

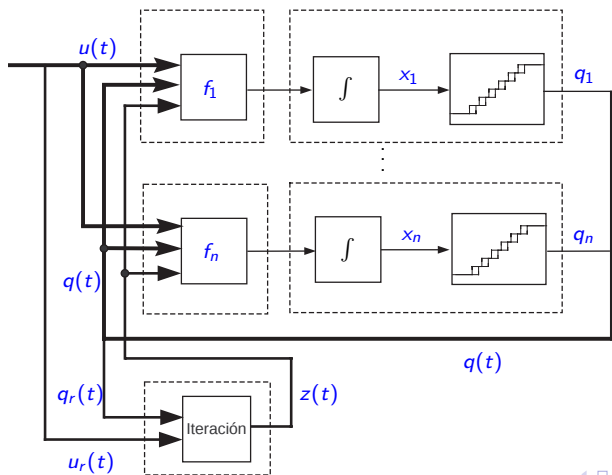
Para usar QSS o QSS2 simplemente cambiamos i_j y u_j por q_{ij} y q_{uj} . El problema es el cálculo de v_z .

Podemos utilizar un nuevo modelo DEVS que calcule v_z a partir de q_{u5} , **iterando** sobre la restricción algebraica cada vez que cambie esta **variable cuantificada**.

donde v_z es una **variable algebraica** que cumple:

Simulación de DAEs con los Métodos de QSS II

El **esquema general** para simular DAEs con los métodos de QSS es el siguiente:



$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{q}, \mathbf{u}, \mathbf{z}) \\ 0 &= \mathbf{g}(\mathbf{q}_r, \mathbf{u}_r, \mathbf{z})\end{aligned}$$

Simulación de DAEs con los Métodos de QSS III

En general, al utilizar los métodos de QSS en **DAEs de índice 1**, sólo es necesario **iterar** en los pasos que involucran a las variables que intervienen en los lazos algebraicos.

Simulación de DAEs con los Métodos de QSS III

En general, al utilizar los métodos de QSS en **DAEs de índice 1**, sólo es necesario **iterar** en los pasos que involucran a las variables que intervienen en los lazos algebraicos.

Esto puede ser una gran ventaja. En el ejemplo de la **línea de transmisión**, la simulación bajo las mismas condiciones que antes llevó **2640** pasos (entre **200** y **316** por cada integrador). Sin embargo, hubo sólo **200** pasos (los de q_{u5}) que provocaron **iteraciones**. Esto agregó un costo computacional casi **despreciable** respecto del ejemplo anterior.

Simulación de DAEs con los Métodos de QSS III

En general, al utilizar los métodos de QSS en **DAEs de índice 1**, sólo es necesario **iterar** en los pasos que involucran a las variables que intervienen en los lazos algebraicos.

Esto puede ser una gran ventaja. En el ejemplo de la **línea de transmisión**, la simulación bajo las mismas condiciones que antes llevó **2640** pasos (entre **200** y **316** por cada integrador). Sin embargo, hubo sólo **200** pasos (los de q_{u5}) que provocaron **iteraciones**. Esto agregó un costo computacional casi **despreciable** respecto del ejemplo anterior.

En cuanto a la implementación, **PowerDEVS** tiene en sus librerías un bloque que resuelve una **restricción algebraica** genérica $g(\mathbf{q}_r, \mathbf{u}_r, z) = 0$ (con $g()$ y z escalares). Para la iteración, utiliza el **método de las secantes**.

Tratamiento de Discontinuidades

Vimos que para simular de sistemas discontinuos era necesario dar un paso en el momento exacto de la discontinuidad. Para esto,

- ▶ en el caso de **eventos temporales** debíamos ajustar el paso de integración para que coincida con los mismos, y,

Tratamiento de Discontinuidades

Vimos que para simular de sistemas discontinuos era necesario dar un paso en el momento exacto de la discontinuidad. Para esto,

- ▶ en el caso de **eventos temporales** debíamos ajustar el paso de integración para que coincida con los mismos, y,
- ▶ en el caso de los **eventos de estado** además teníamos que **iterar** para detectar **cuando** se producían los eventos.

Tratamiento de Discontinuidades

Vimos que para simular de sistemas discontinuos era necesario dar un paso en el momento exacto de la discontinuidad. Para esto,

- ▶ en el caso de **eventos temporales** debíamos ajustar el paso de integración para que coincida con los mismos, y,
- ▶ en el caso de los **eventos de estado** además teníamos que **iterar** para detectar **cuando** se producían los eventos.

Con los métodos de QSS estos problemas desaparecerán, porque:

- ▶ Frente a **eventos temporales**, todos los bloques están preparados para recibir eventos de manera **asíncrona**. De hecho, los métodos funcionan así aún con los sistemas sin discontinuidades.

Tratamiento de Discontinuidades

Vimos que para simular de sistemas discontinuos era necesario dar un paso en el momento exacto de la discontinuidad. Para esto,

- ▶ en el caso de **eventos temporales** debíamos ajustar el paso de integración para que coincida con los mismos, y,
- ▶ en el caso de los **eventos de estado** además teníamos que **iterar** para detectar **cuando** se producían los eventos.

Con los métodos de QSS estos problemas desaparecerán, porque:

- ▶ Frente a **eventos temporales**, todos los bloques están preparados para recibir eventos de manera **asíncrona**. De hecho, los métodos funcionan así aún con los sistemas sin discontinuidades.
- ▶ En el caso de los **eventos de estado**, la detección de los mismos es trivial, ya que las formas de las trayectorias son **seccionalmente** constantes (QSS), lineales (QSS2) o parabólicas (QSS3). Por lo tanto, no hace falta **iterar** para detectar los **cruces por cero**.

Tratamiento de Discontinuidades

Vimos que para simular de sistemas discontinuos era necesario dar un paso en el momento exacto de la discontinuidad. Para esto,

- ▶ en el caso de **eventos temporales** debíamos ajustar el paso de integración para que coincida con los mismos, y,
- ▶ en el caso de los **eventos de estado** además teníamos que **iterar** para detectar **cuando** se producían los eventos.

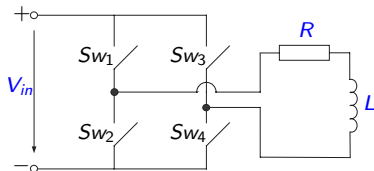
Con los métodos de QSS estos problemas desaparecerán, porque:

- ▶ Frente a **eventos temporales**, todos los bloques están preparados para recibir eventos de manera **asíncrona**. De hecho, los métodos funcionan así aún con los sistemas sin discontinuidades.
- ▶ En el caso de los **eventos de estado**, la detección de los mismos es trivial, ya que las formas de las trayectorias son **seccionalmente** constantes (QSS), lineales (QSS2) o parabólicas (QSS3). Por lo tanto, no hace falta **iterar** para detectar los **cruces por cero**.

Por esto, la **principal ventaja** de los métodos de QSS será en la **simulación de sistemas con discontinuidades**.

Eventos Temporales – Ejemplo Introductorio

El siguiente es un **circuito inversor**:



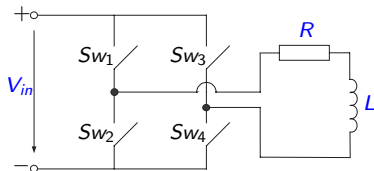
que puede modelarse con la ODE:

$$\frac{di_L}{dt} = -\frac{R}{L} \cdot i_L + \frac{V_{in}}{L} \cdot s_w(t)$$

donde $s_w(t)$ vale ± 1 según la posición de los conmutadores.

Eventos Temporales – Ejemplo Introductorio

El siguiente es un **circuito inversor**:

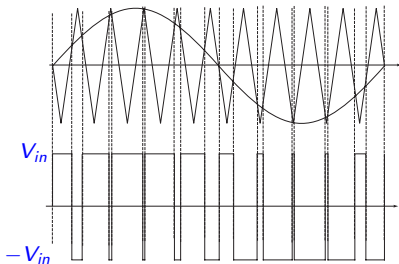


que puede modelarse con la ODE:

$$\frac{di_L}{dt} = -\frac{R}{L} \cdot i_L + \frac{V_{in}}{L} \cdot s_w(t)$$

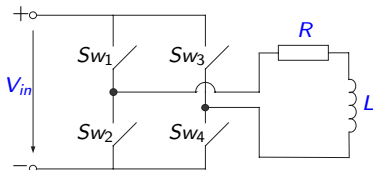
donde $s_w(t)$ vale ± 1 según la posición de los conmutadores.

Una forma típica de **controlar** los conmutadores para obtener una corriente similar a una **senoidal** es utilizar **modulación de ancho de pulso (PWM)**:



Eventos Temporales – Ejemplo Introductorio

El siguiente es un **circuito inversor**:

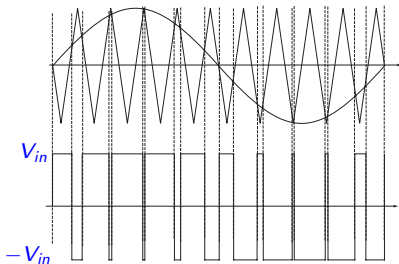


que puede modelarse con la ODE:

$$\frac{di_L}{dt} = -\frac{R}{L} \cdot i_L + \frac{V_{in}}{L} \cdot s_w(t)$$

donde $s_w(t)$ vale ± 1 según la posición de los conmutadores.

Una forma típica de **controlar** los conmutadores para obtener una corriente similar a una **senoidal** es utilizar **modulación de ancho de pulso (PWM)**:

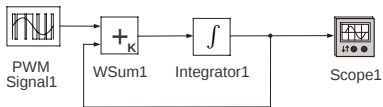


Dado que podemos calcular de antemano los instantes de conmutación, estamos frente a **eventos temporales**.

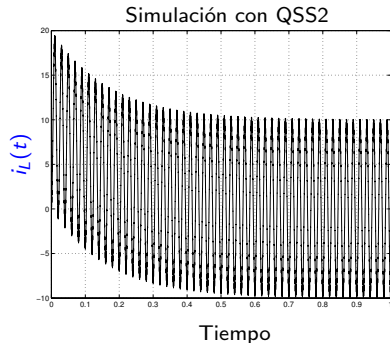
Eventos Temporales – Métodos de QSS

Para simular el sistema anterior con QSS o QSS2, necesitamos un modelo DEVS que **genere la secuencia de eventos** correspondiente a $s_w(t)$. Obtener dicho modelo DEVS es muy sencillo.

El modelo en **PowerDEVS** tendrá la siguiente forma:



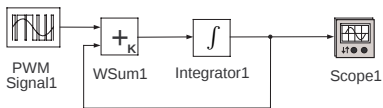
y los resultados de la simulación son:



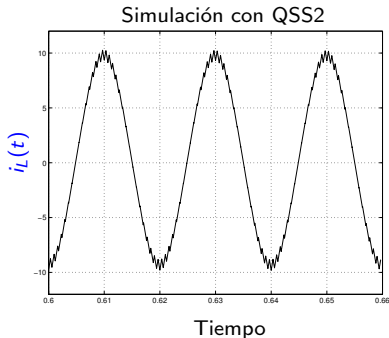
Eventos Temporales – Métodos de QSS

Para simular el sistema anterior con QSS o QSS2, necesitamos un modelo DEVS que **genere la secuencia de eventos** correspondiente a $s_w(t)$. Obtener dicho modelo DEVS es muy sencillo.

El modelo en **PowerDEVS** tendrá la siguiente forma:



y los resultados de la simulación son:



Los **eventos temporales** se tratan como una **entrada** cualquiera. En este caso vale la **cota de error global** (aquí el error está acotado por el quantum utilizado, $\Delta i_L = 0.01$ A).

Eventos de Estado – Ejemplo Introductorio

Una manera de proteger el circuito anterior de sobrecorrientes durante fallos o transitorios es agregando un dispositivo que **sense** la corriente y cierre los conmutadores forzando $s_w = 0$ cuando se sobrepasa cierto límite.

Eventos de Estado – Ejemplo Introductorio

Una manera de proteger el circuito anterior de sobrecorrientes durante fallos o transitorios es agregando un dispositivo que **sense** la corriente y cierre los conmutadores forzando $s_w = 0$ cuando se sobrepasa cierto límite.

Agregando este dispositivo, las ecuaciones del sistema quedan:

$$\frac{di_L}{dt} = -\frac{R}{L} \cdot i_L + V_{in} \cdot \tilde{s}_w(t)$$

con:

$$\tilde{s}_w(t) = \begin{cases} s_w(t) & \text{si } i_L(t) < i_M \\ 0 & \text{en otro caso} \end{cases} \quad (8)$$

donde i_M es la **máxima** corriente.

Eventos de Estado – Ejemplo Introductorio

Una manera de proteger el circuito anterior de sobrecorrientes durante fallos o transitorios es agregando un dispositivo que **sense** la corriente y cierre los conmutadores forzando $s_w = 0$ cuando se sobrepasa cierto límite.

Agregando este dispositivo, las ecuaciones del sistema quedan:

$$\frac{di_L}{dt} = -\frac{R}{L} \cdot i_L + V_{in} \cdot \tilde{s}_w(t)$$

con:

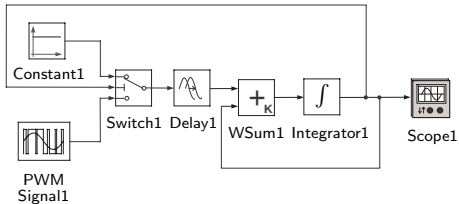
$$\tilde{s}_w(t) = \begin{cases} s_w(t) & \text{si } i_L(t) < i_M \\ 0 & \text{en otro caso} \end{cases} \quad (8)$$

donde i_M es la **máxima** corriente.

Ahora, evidentemente tenemos **eventos de estado**.

Eventos de Estado – Métodos de QSS

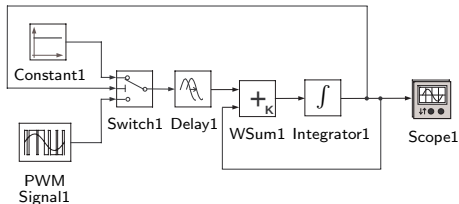
El Modelo en PowerDEVS es el siguiente:



El bloque **Delay** modela un retraso en la conmutación. Esto evita la **ilegitimidad** del modelo.

Eventos de Estado – Métodos de QSS

El Modelo en PowerDEVS es el siguiente:



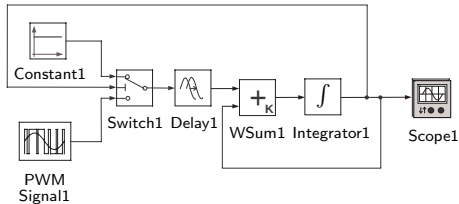
El bloque **Delay** modela un retraso en la conmutación. Esto evita la **ilegitimidad** del modelo.

El bloque **Switch** modela la Ec.(8). Funciona así:

- ▶ cuando recibe un **evento** por el segundo puerto, calcula el tiempo que falta para que dicha señal cruce un determinado nivel. Transcurrido ese tiempo **conmuta** la posición del conmutador.

Eventos de Estado – Métodos de QSS

El Modelo en PowerDEVS es el siguiente:



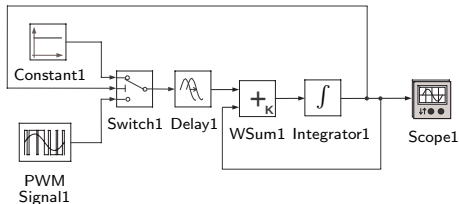
El bloque **Delay** modela un retraso en la conmutación. Esto evita la **ilegitimidad** del modelo.

El bloque **Switch** modela la Ec.(8). Funciona así:

- ▶ cuando recibe un **evento** por el segundo puerto, calcula el tiempo que falta para que dicha señal cruce un determinado nivel. Transcurrido ese tiempo **conmuta** la posición del conmutador.
- ▶ cuando recibe eventos en los otros dos puertos, los retransmite según la posición del conmutador.

Eventos de Estado – Métodos de QSS

El Modelo en PowerDEVS es el siguiente:



El bloque **Delay** modela un retraso en la conmutación. Esto evita la **ilegitimidad** del modelo.

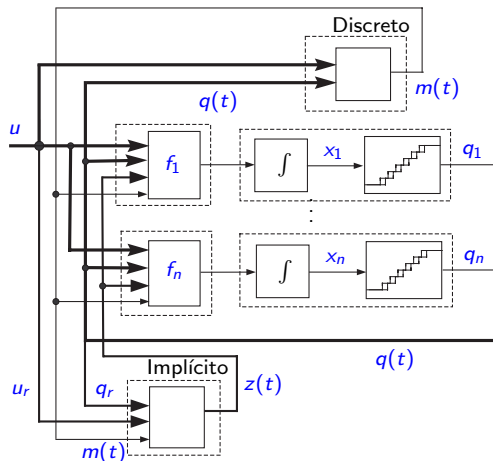
El bloque **Switch** modela la Ec.(8). Funciona así:

- ▶ cuando recibe un **evento** por el segundo puerto, calcula el tiempo que falta para que dicha señal cruce un determinado nivel. Transcurrido ese tiempo **conmuta** la posición del conmutador.
- ▶ cuando recibe eventos en los otros dos puertos, los retransmite según la posición del conmutador.

El cálculo del **tiempo de la conmutación** es trivial ya que la señal es seccionalmente lineal (QSS2) o parabólica (QSS3).

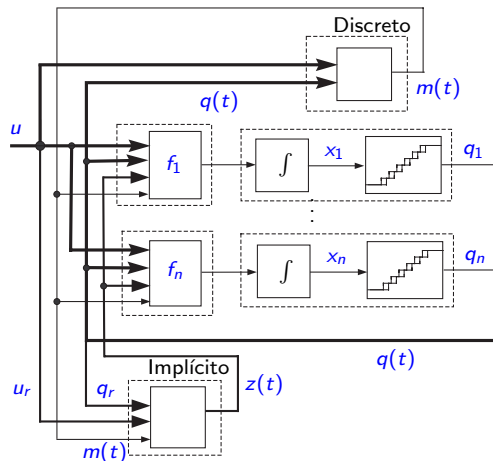
Sistemas Discontinuos – Esquema General

En general, el esquema será el siguiente:



Sistemas Discontinuos – Esquema General

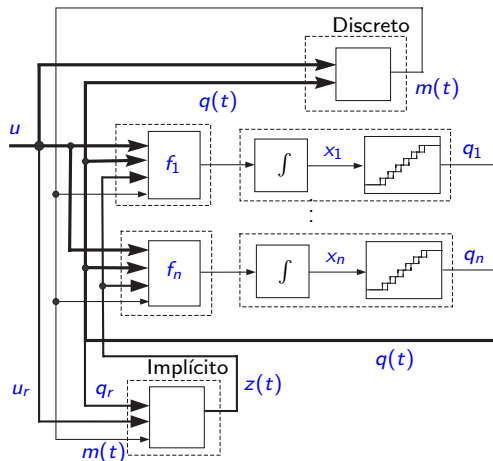
En general, el esquema será el siguiente:



- El subsistema **discreto** es el encargado de **detectar** y **manejar** las discontinuidades, calculando la señal **seccionalmente constante** $m(t)$.

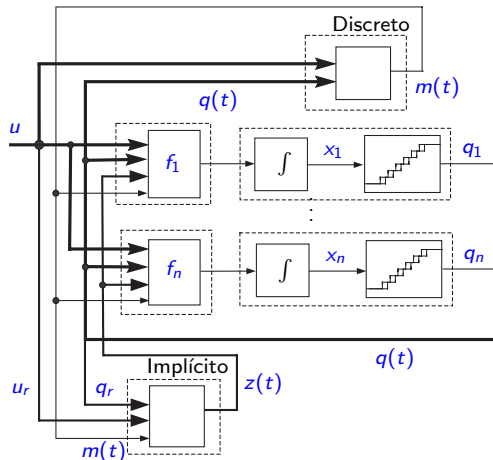
Sistemas Discontinuos – Esquema General

En general, el esquema será el siguiente:



- ▶ El subsistema **discreto** es el encargado de **detectar** y **manejar** las discontinuidades, calculando la señal **seccionalmente constante** $m(t)$.
- ▶ El resto funciona **ignorando** la presencia de las **discontinuidades**. La señal $m(t)$ es vista como una **entrada** o una **variable de estado** cualquiera.

En general, el esquema será el siguiente:



- ▶ El subsistema **discreto** es el encargado de **detectar** y **manejar** las discontinuidades, calculando la señal **seccionalmente constante** $m(t)$.
- ▶ El resto funciona **ignorando** la presencia de las **discontinuidades**. La señal $m(t)$ es vista como una **entrada** o una **variable de estado** cualquiera.
- ▶ Sólo hay que preocuparse en utilizar bloques que generen correctamente la señal $m(t)$. **PowerDEVS** tiene una librería de bloques de este tipo, que incluye **conmutadores**, **cuantificadores**, **comparadores**, etc.

Algunos problemas abiertos

- ▶ Los métodos de QSS vistos no funcionan con **sistemas rígidos**, debido a la aparición de oscilaciones rápidas. Sin embargo, hay un método nuevo llamado BQSS que sí funciona, pero que aún está en desarrollo.

Algunos problemas abiertos

- ▶ Los métodos de QSS vistos no funcionan con **sistemas rígidos**, debido a la aparición de oscilaciones rápidas. Sin embargo, hay un método nuevo llamado BQSS que sí funciona, pero que aún está en desarrollo.
- ▶ En los sistemas **marginalmente estables** se producen soluciones divergentes. Aparentemente, esto también se solucionaría a partir de la idea de BQSS.

Algunos problemas abiertos

- ▶ Los métodos de QSS vistos no funcionan con **sistemas rígidos**, debido a la aparición de oscilaciones rápidas. Sin embargo, hay un método nuevo llamado BQSS que sí funciona, pero que aún está en desarrollo.
- ▶ En los sistemas **marginalmente estables** se producen soluciones divergentes. Aparentemente, esto también se solucionaría a partir de la idea de BQSS.
- ▶ El **quantum** debe elegirse para cada variable de estado. Sería interesante implementar algún tipo de **cuantificación adaptativa**.

Algunos problemas abiertos

- ▶ Los métodos de QSS vistos no funcionan con **sistemas rígidos**, debido a la aparición de oscilaciones rápidas. Sin embargo, hay un método nuevo llamado BQSS que sí funciona, pero que aún está en desarrollo.
- ▶ En los sistemas **marginalmente estables** se producen soluciones divergentes. Aparentemente, esto también se solucionaría a partir de la idea de BQSS.
- ▶ El **quantum** debe elegirse para cada variable de estado. Sería interesante implementar algún tipo de **cuantificación adaptativa**.
- ▶ Otro problema, relacionado al anterior, es que los métodos controlan intrínsecamente el **error absoluto global**, cuando sería deseable que controlen el **error relativo global**. Una idea para solucionar esto es utilizar **cuantificación logarítmica**.

Conclusiones

Conclusiones

- ▶ Continuamos con la discusión de métodos de integración basados en la cuantificación de estados. Introdujimos métodos de segundo y tercer orden.

Conclusiones

- ▶ Continuamos con la discusión de métodos de integración basados en la cuantificación de estados. Introdujimos métodos de segundo y tercer orden.
- ▶ Enseñamos que los problemas de la estabilidad y de la precisión pueden analizarse usando el cálculo de perturbaciones.

Conclusiones

- ▶ Continuamos con la discusión de métodos de integración basados en la cuantificación de estados. Introdujimos métodos de segundo y tercer orden.
- ▶ Enseñamos que los problemas de la estabilidad y de la precisión pueden analizarse usando el cálculo de perturbaciones.
- ▶ Se mostró que los métodos QSS*i* se quedan prácticamente estables y que puede encontrarse una estimación del error global de la simulación. Este último punto es de gran importancia.

Conclusiones

- ▶ Continuamos con la discusión de métodos de integración basados en la cuantificación de estados. Introdujimos métodos de segundo y tercer orden.
- ▶ Enseñamos que los problemas de la estabilidad y de la precisión pueden analizarse usando el cálculo de perturbaciones.
- ▶ Se mostró que los métodos QSS*i* se quedan prácticamente estables y que puede encontrarse una estimación del error global de la simulación. Este último punto es de gran importancia.
- ▶ Los métodos QSS*i* son naturalmente asíncronos y pueden implementarse entonces fácilmente sobre arquitecturas paralelas usando múltiples computadoras.

Conclusiones

- ▶ Continuamos con la discusión de métodos de integración basados en la cuantificación de estados. Introdujimos métodos de segundo y tercer orden.
- ▶ Enseñamos que los problemas de la estabilidad y de la precisión pueden analizarse usando el cálculo de perturbaciones.
- ▶ Se mostró que los métodos QSSi se quedan prácticamente estables y que puede encontrarse una estimación del error global de la simulación. Este último punto es de gran importancia.
- ▶ Los métodos QSSi son naturalmente asíncronos y pueden implementarse entonces fácilmente sobre arquitecturas paralelas usando múltiples computadoras.
- ▶ Los métodos QSSi pueden usarse particularmente bien para la simulación a tramos de modelos híbridos.