QusCos[1]

Quantized State System Simulation in Scicos

Dirk Reusch

Kybernetik Dr. Reusch
Technische Systeme analysieren und optimieren

3rd International Workshop on Simulation at the System Level for Industrial Applications

October 14th to 16th, 2015

ENS Cachan (France)

[1] speak 'Couscous'

Outline

Introduction

- In 2001, Quantized State Systems were introduced
  - accompagnied by a first order integration method (QSS1)
- Today, there are different integration methods (solvers) available:
  - QSS (order 1...4; non-stiff problems)
  - CQSS (order 1 only; marginally stable problems)
  - LIQSS (order 1...4; stiff problems)
- Traditionally, implementations are based on the DEVS formalism
  - we may regard PowerDEVS as a reference implementation
- In general, QSS methods
  - have pleasing convergence, stability and error bound properties
  - update system states asynchronously
  - need no iteration scheme for discontinuity handling
  - require explicit calculations only (for ODE solving w/ method order $\leq$4)
- Focus of this talk:
  - proof-of-concept implementation of QSS1 & QSS2 methods in Scicos
  - illustration of QSS peculiarities (no mathematical subtleties)

Quantized State System (QSS)

A continous time invariant system

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

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is approximated by a QSS

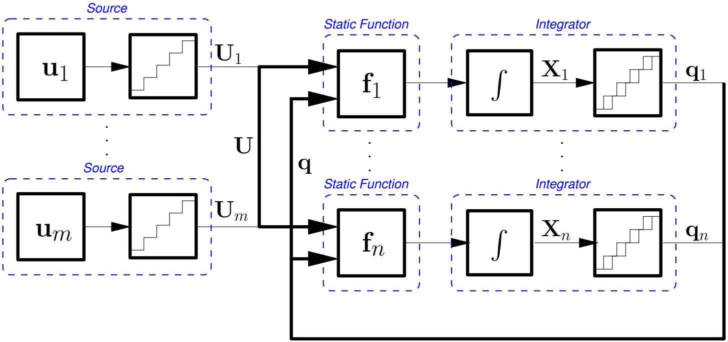
$$\dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{U}(t)) \tag{2}$$

with[2] $\mathbf{X}, \mathbf{q} \in \mathbb{R}^n$ and $\mathbf{U} \in \mathbb{R}^m$, where

- $\mathbf{x}(t)$ is replaced by the quantized state $\mathbf{q}(t)$
- $\mathbf{U}(t)$ is a quantized approximation of $\mathbf{u}(t)$
- $\dot{\mathbf{X}}(t)$ is an approximation of $\dot{\mathbf{x}}(t)$
- $\mathbf{q}(t)$ and $\mathbf{X}(t)$ are related by hysteretic quantization
- changes of quantization levels are events
- eq. (2) specifies a legitimate discrete event system

[2] In literature, usually the same notation is used for $\mathbf{x}$ and $\mathbf{X}$ or $\mathbf{u}$ and $\mathbf{U}$

Block Diagram



- Source, Static Function and Integrator are the building blocks
- For simplicity, focus on blocks with scalar inputs/outputs only
- How does/could an Integrator block work?

First Order Integration Method (QSS1, Scalar)

Choice of hysteretic quantization for $X, q \in \mathbb{R}$ as

$$q(t) = \begin{cases} X(t) & \text{if } |q(t^-) - X(t)| = \Delta Q, \\ q(t^-) & \text{otherwise} \end{cases} \tag{3}$$

with the quantum $\Delta Q > 0$. From eq. (3) follows:

- $q(t)$ is piecewise constant
- $q(t)$ changes only, when it differs from $X(t)$ by ΔQ

Quantized input $U(t) \in \mathbb{R}$ is piecewise constant. From eq. (2)[3] follows:

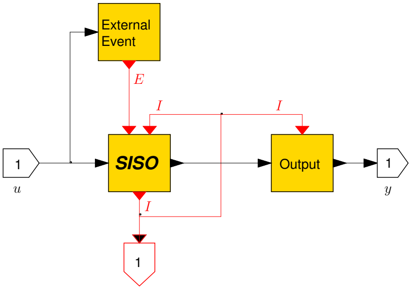
- $\dot{X}(t)$ is piecewise constant,
- thus integration is easy, and $X(t)$ is piecewise linear

Second order QSS2 method works likewise[4].

[3] $\dot{X}(t) = f(q(t), U(t))$

[4] with piecewise linear $q(t)$, $U(t)$, $\dot{X}(t)$, and piecewise parabolic $X(t)$

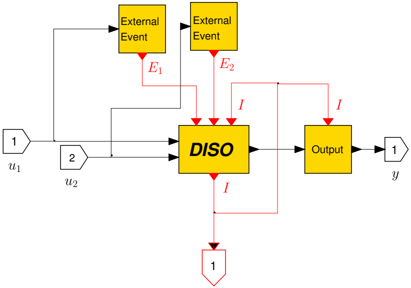
Anatomy of Blocks (SISO)



Single Input / Single Output:

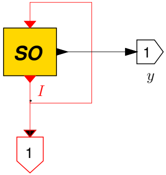
- e.g. Integrator, Gain, . . .
- **External Event** is generic
- **Output** is generic
  - no state
  - direct feedthrough
  - purpose: event filtering
- **SISO** is operation-specific
  - discrete state
  - no direct feedthrough
  - ***E*** or ***I*** triggers state update
  - only ***I*** triggers output update
- ***E*** and ***I*** are asynchronous

Anatomy of Blocks (DISO, SO)



Dual Input / Single Output:

- e.g. Sum, Product, . . .
- ***E*$_1$** and ***E*$_2$** are not synchronized with ***I***
- ***E*$_1$** and ***E*$_2$** might be synchronous
- scheme allows up to 31 inputs



Single Output (Source):

- e.g. Step, Pulse, . . .
- **Output** block is not needed

Representation of Quantities

QSS method of order $k = 1, 2, \ldots$:

- input/output signal **v** of a block

$$\mathbf{v} \in \mathbb{R}^k$$

- approximated state **X** used inside an integrator block

$$\mathbf{X} \in \mathbb{R}^{k+1}$$

- components of a quantity $\mathbf{a} = (a_1, a_2, a_3, \ldots)$
    - a_1 : value
    - a_2 : slope (1st derivative)
    - a_3 : curvature (2nd derivative)
    - $\ldots$

Static Function (Product, QSS2)

Inputs $\mathbf{u_1}, \mathbf{u_2} \in \mathbb{R}^2$, Output $\mathbf{y} \in \mathbb{R}^2$
State $(\mathbf{v_1}, \mathbf{v_2}, t_l, \sigma) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R}$

Time of Last State Update t_l, Lifespan of State σ

Initialization

$$\mathbf{y} \leftarrow \mathbf{0}$$
$$\mathbf{v_1} \leftarrow \mathbf{u_1}$$
$$\mathbf{v_2} \leftarrow \mathbf{u_2}$$
$$t_l \leftarrow 0$$
$$\sigma \leftarrow 0$$

Output Update

$$\boldsymbol{I}: \quad y_1 \leftarrow v_{11} \cdot v_{21}$$
$$y_2 \leftarrow v_{11} \cdot v_{22} + v_{12} \cdot v_{21}$$

State Update & Schedule of ***I***

$$e \leftarrow t - t_l$$
$$t_l \leftarrow t$$
$$\boldsymbol{E_1} \wedge \overline{\boldsymbol{E_2}}: \quad \mathbf{v_1} \leftarrow \mathbf{u_1}$$
$$v_{21} \leftarrow v_{21} + v_{22} \cdot e$$
$$\overline{\boldsymbol{E_1}} \wedge \boldsymbol{E_2}: \quad \mathbf{v_2} \leftarrow \mathbf{u_2}$$
$$v_{11} \leftarrow v_{11} + v_{12} \cdot e$$
$$\boldsymbol{E_1} \wedge \boldsymbol{E_2}: \quad \mathbf{v_1} \leftarrow \mathbf{u_1}$$
$$\mathbf{v_2} \leftarrow \mathbf{u_2}$$
$$\boldsymbol{E_1} \vee \boldsymbol{E_2}: \quad \sigma \leftarrow 0$$
$$\boldsymbol{I}: \quad \sigma \leftarrow \infty$$
$$\text{next } \boldsymbol{I} \text{ at}: \quad t + \sigma$$

Integrator (QSS1)

Inputs $u \in \mathbb{R}$, Output $y \in \mathbb{R}$
Parameters $(x_0, \Delta Q) \in \mathbb{R} \times \mathbb{R}_+$
State $(\mathbf{X}, q, t_l, \sigma) \in \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$

Time of Last State Update t_l, Lifespan of State σ

Initialization

$$\begin{aligned} y &\leftarrow 0 \\ X_1 &\leftarrow x_0 \\ X_2 &\leftarrow 0 \\ q &\leftarrow x_0 \\ t_l &\leftarrow 0 \\ \sigma &\leftarrow 0 \end{aligned}$$

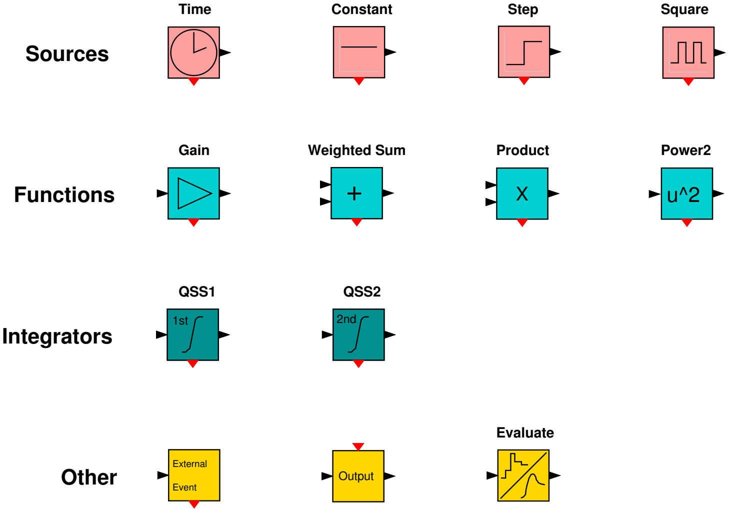
Output Update

$$\boldsymbol{I}: \quad y \leftarrow X_1 + X_2 \cdot \sigma$$

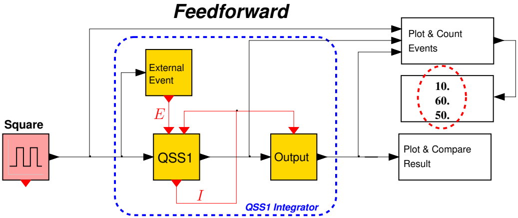
*State Update & Schedule of **I***

$$\begin{aligned} & e \leftarrow t - t_l \\ & t_l \leftarrow t \\ \boldsymbol{I}: \quad & X_1 \leftarrow X_1 + X_2 \cdot \sigma \\ & q \leftarrow X_1 \\ & \sigma \leftarrow \begin{cases} \left|\frac{\Delta Q}{X_2}\right| & \text{if } X_2 \neq 0, \\ \infty & \text{otherwise} \end{cases} \\ \boldsymbol{E}: \quad & X_1 \leftarrow X_1 + X_2 \cdot e \\ & X_2 \leftarrow u \\ & \sigma \leftarrow \\ & \quad \min_{\sigma' \geq 0} \{\sigma' | \Delta Q = |q - X_1 - X_2 \cdot \sigma'|\} \\ \text{next } \boldsymbol{I} \text{ at}: \quad & t + \sigma \end{aligned}$$

Implemented Blocks (QSS1 & QSS2)

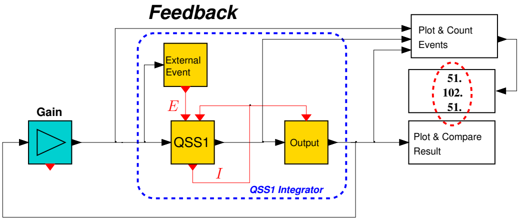


Event Propagation & Filtering (Feedforward)



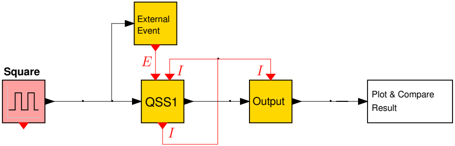
- **Square**
  - generates ***E*** (10)
- **QSS1**
  - generates ***I*** (50)
  - propagates ***E*** and ***I*** (60)
- **Output**
  - filters out ***E*** (10)
  - propagates only ***I*** (50)

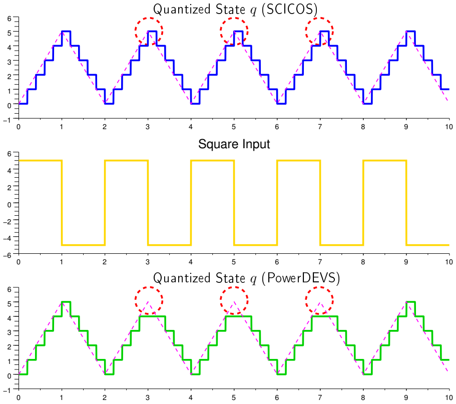
Event Propagation & Filtering (Feedback)



- **Gain**
  - receives ***I*** (51)
  - generates wo/ delay ***E*** (51)
  - propagates only ***E*** (51)
- **QSS1**
  - generates ***I*** (51)
  - propagates ***E*** and ***I*** (102)
- **Output**
  - filters out ***E*** (51)
  - propagates only ***I*** (51)

Indeterminism Effects



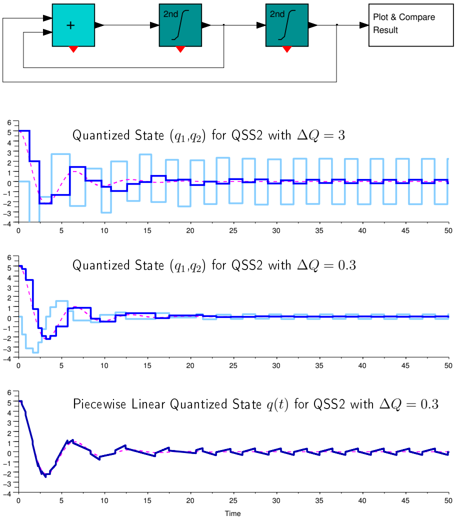


- Example (QSS1 with $\Delta Q = 1$):

  $$\dot{x}(t) = 5\,\mathrm{sign}(\sin(\pi t)),\ x(0) = 0$$

- SCICOS vs. POWERDEVS
  - results are differing
- Why?
- ***E*** and ***I*** at $t = 1, 2, 3, \ldots$
  - simultaneous but asynchronous
  - indetermined processing order
- both results are valid
- different implementations may yield not the very same results

Stability and Ultimately Bounded Oscillations



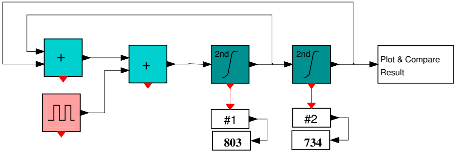
- Example: Stable LTI system

$$\ddot{x}(t)+\frac{1}{2}\dot{x}(t)+x(t) = 0, x(0) = 5, \dot{x}(0) = 0$$

- QSS methods conserve (asymptotic) stability of original system for
  - arbitrary $\Delta Q > 0$ (LTI)
  - sufficient small $\Delta Q > 0$ (nonlinear)
- generated trajectories are
  - practically stable, i.e.
  - ultimately bounded oscillations

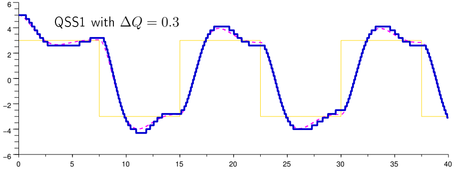
  usually remain at equilibrium points
- calculation of global error bound is
  - easy for LTI systems (explicit formula)
  - possible for nonlinear systems (perturbation theory)

Forced 2nd Order System



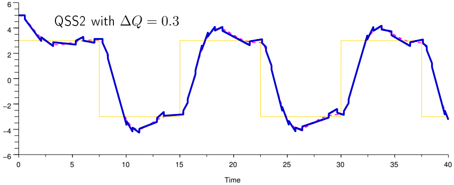
$$\ddot{x}(t) + \dot{x}(t) + x(t) = \operatorname{sign}\left(\sin\left(\frac{2\pi}{15}t\right)\right),$$

$$x(0) = 5,\ \dot{x}(0) = 0$$



State Updates with QSS1 (PowerDEVS)

ΔQ	Integrator #1	Integrator #2
0.1	392 (+1)	412
0.01	3770 (+1)	4066
0.001	37577 (+1)	40609

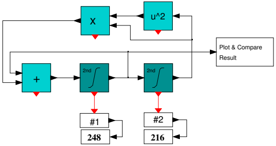


State Updates with QSS2 (PowerDEVS)

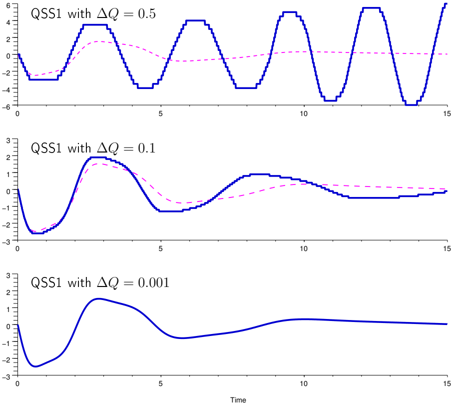
ΔQ	Integrator #1	Integrator #2
0.1	79 (+1)	68
0.01	252 (+1)	229
0.001	803 (+1)	734

(Note: difference (+1) seems to be due to a counting bug)

Nonlinear 2nd Order System (DUFFING Oscillator)



$$\ddot{x}(t) + \frac{1}{3}\dot{x}(t) + x^3(t) = 0,$$
$$x(0) = 2,\ \dot{x}(0) = 0$$



State Updates with QSS1 (PowerDEVS)

ΔQ	Integrator #1	Integrator #2
0.1	143 (+1)	129
0.01	1056 (+1)	965
0.001	10242 (+1)	9431

State Updates with QSS2 (PowerDEVS)

ΔQ	Integrator #1	Integrator #2
0.1	20 (+1)	12
0.01	81 (+1)	64
0.001	248 (+1)	216

(Note: difference (+1) seems to be due to a counting bug)

Conclusions and Outlook

Conclusions:

- implementing QSS methods in SCICOS is feasible
- simulation results are in conformance with POWERDEVS
- starting point for full-featured implementation of QSS methods

Outlook:

- clean-up, optimization, benchmarking with respect to runtime
- possible tweaks of SCICOS for more simulation efficiency
- implementation of
  - illegitimacy detection
  - logarithmic quantization (absolute & relative error control)
  - methods for stiff and marginally stable systems
  - higher order methods
  - switches, hysteresis, saturation, . . .
  - generic static function block
  - vectorial extension of the approach
  - . . .

Merci de votre attention! Questions?

References

Ernesto Kofman.
A second order approximation for DEVS simulation of continuous systems.
Simulation, 78(2):76–89, 2002.

Ernesto Kofman and Sergio Junco.
Quantized state systems. A DEVS approach for continuous system simulation.
Transactions of SCS, 18(3):123–132, 2001.

Frederico Bergero and Ernesto Kofman.
PowerDEVS: a tool for hybrid system modeling and realtime simulation.
Simulation, 87(1–2):113–132, 28 May 2010.

Gustavo Migoni, Ernesto Kofman, and François E. Cellier.
Quantization-based new integration methods for stiff ordinary differential equations.
Simulation, 88(4):387–407, 2012.

Gustavo Migoni, Mario Bortolotto, Ernesto Kofman, and François E. Cellier.
Linearly implicit quantization-based integration methods for stiff ordinary differential equations.
Simulation Modelling Practice and Theory, 35:118–136, 2013.