Index Theory for Hybrid DAE Systems

Albert Benveniste Timothy Bourke
Benoît Caillaud Marc Pouzet

INRIA and ENS, France

November 27, 2014

Compositionality and reuse: from ODE to DAE

from Simulink (ODE):
HS in state space form

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x, u) \end{cases}$$

the state space form
depends on the context

reuse is difficult

$$\longrightarrow$$

to Modelica (DAE):
HS as physical balance equations

$$\begin{cases} 0 = f(\dot{x}, x, u) \\ 0 = g(x, u) \end{cases}$$

Ohm & Kirchhoff laws, bond graphs,
multi-body mechanical systems

reuse is much easier

Compositionality and reuse: from ODE to DAE

- Modeling tools supporting DAE
  - Most modeling tools provide only a library of predefined models ready for assembly (Mathworks/Simscape, LMS/AmeSim)
  - Modelica comes with a full programming language that is a public standard `https://www.modelica.org/` ; also Spice for EDA

- Strange outcomes for the simulations were known to occur with Simulink/Stateflow (ask Tim Bourke and Marc Pouzet for nice ones);
- Exploration of Modelica is only starting…

Compositionality and reuse: from ODE to DAE

- Modeling tools supporting DAE
  - Most modeling tools provide only a library of predefined models ready for assembly (Mathworks/Simscape, LMS/AmeSim)
  - Modelica comes with a full programming language that is a public standard `https://www.modelica.org/` ; also Spice for EDA

- Strange outcomes for the simulations were known to occur with Simulink/Stateflow (ask Tim Bourke and Marc Pouzet for nice ones);
- Exploration of Modelica is only starting…

Compositionality and reuse: from ODE to DAE

- We do not claim that these tools are bad, as there are real difficulties:
  - from ODE solvers to DAE solvers
  - events of mode changes for Hybrid DAE systems
  - and the physics itself:

semiconductor,
circuit breaker,
...

multibody
mechanics

sliding
modes

A key notion in DAE Systems: the index

- Differential Algebraic Equations systems (continuous time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl}\dot{x} & = & f(x,u)\\ 0 & = & g(x)\end{array}\right. \overset{\text{differentiating}}{\Longrightarrow} \left\{\begin{array}{lcl}\dot{x} & = & f(x,u)\\ 0 & = & g(x)\\ 0 & = & g'_x(x)\dot{x}\end{array}\right.$$

$$\overset{\text{substituting}}{\Longrightarrow} \left\{\begin{array}{lcll}\dot{x} & = & f(x,u) & (1)\\ 0 & = & g(x) & (2)\\ 0 & = & g'_x(x)f(x,u) & (3)\end{array}\right.$$

- What is the effect on execution schemes? ($\sim$ constructive semantics)

A key notion in DAE Systems: the index

- difference Algebraic Equations systems (discrete time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl} x^{\bullet} & = & f(x,u) \\ 0 & = & g(x) \end{array}\right. \overset{\text{shifting}}{\Longrightarrow} \left\{\begin{array}{lcl} x^{\bullet} & = & f(x,u) \\ 0 & = & g(x) \\ 0 & = & g(x^{\bullet}) \end{array}\right.$$

$$\overset{\text{substituting}}{\Longrightarrow} \left\{\begin{array}{lcll} x^{\bullet} & = & f(x,u) & (1) \\ 0 & = & g(x) & (2) \\ 0 & = & g(f(x,u)) & (3) \end{array}\right.$$

A key notion in DAE Systems: the index

- difference Algebraic Equations systems (discrete time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl} x^\bullet & = & f(x,u) \\ 0 & = & g(x) \end{array}\right. \quad \overset{\text{shifting}}{\Longrightarrow} \quad \left\{\begin{array}{lcl} x^\bullet & = & f(x,u) \\ 0 & = & g(x) \\ 0 & = & g(x^\bullet) \end{array}\right.$$

$$\overset{\text{substituting}}{\Longrightarrow} \quad \left\{\begin{array}{lcll} x^\bullet & = & f(x,u) & (1) \\ 0 & = & g(x) & (2) \\ 0 & = & g(f(x,u)) & (3) \end{array}\right.$$

- Execution scheme ($\sim$constructive semantics):

  1. Given x such that $g(x) = 0$
  2. Use (3) to evaluate u (constraint solver needed)
  3. Use (1) to evaluate $x^\bullet$, which satisfies $g(x^\bullet) = 0$, and repeat

  Adding (3) essential in deriving the constructive semantics $x \rightarrow x^\bullet$

A key notion in DAE Systems: the index

- difference Algebraic Equations systems (discrete time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl} x^{\bullet} & = & f(x,u) \\ 0 & = & g(x) \end{array}\right. \overset{\text{shifting}}{\Longrightarrow} \left\{\begin{array}{lcl} x^{\bullet} & = & f(x,u) \\ 0 & = & g(x) \\ 0 & = & g(x^{\bullet}) \end{array}\right.$$

$$\overset{\text{shifting}^2}{\Longrightarrow} \left\{\begin{array}{lcll} x^{\bullet} & = & f(x,u) & (1) \\ 0 & = & g(x) & (2) \\ 0 & = & g(x^{\bullet}) & (3) \\ 0 & = & g(x^{\bullet 2}) & (4) \end{array}\right.$$

- Shifting^2 is useless since the second shifting introduces

  1. eqn (4) but also
  2. the fresh variable $x^{\bullet 2}$

  Thus, adding (4) does not help getting the value of $x^{\bullet}$

A key notion in DAE Systems: the index

- difference Algebraic Equations systems (discrete time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl} x^{\bullet 2} & = & f(x,u) \\ 0 & = & g(x) \end{array}\right. \overset{\text{shifting}}{\Longrightarrow} \left\{\begin{array}{lcl} x^{\bullet 2} & = & f(x,u) \\ 0 & = & g(x) \\ 0 & = & g(x^{\bullet}) \end{array}\right.$$

$$\overset{\text{shifting}^2}{\Longrightarrow} \left\{\begin{array}{lcll} x^{\bullet 2} & = & f(x,u) & (1) \\ 0 & = & g(x) & (2) \\ 0 & = & g(x^{\bullet}) & (3) \\ 0 & = & g(x^{\bullet 2}) & (4) \end{array}\right.$$

- Shifting^2 is useful since the second shifting introduces
  1. eqn (4) and
  2. the variable $x^{\bullet 2}$, which is not fresch but already there

A key notion in DAE Systems: the index

- Differential Algebraic Equations systems (continuous time) may involve more constraints than specified:

$$\left\{\begin{array}{lcl} \dot{x} & = & f(x,u) \\ 0 & = & g(x) \end{array}\right. \overset{\text{differentiating}}{\Longrightarrow} \left\{\begin{array}{lcl} \dot{x} & = & f(x,u) \\ 0 & = & g(x) \\ 0 & = & g'_x(x)\dot{x} \end{array}\right.$$

$$\overset{\text{substituting}}{\Longrightarrow} \left\{\begin{array}{lcll} \dot{x} & = & f(x,u) & (1) \\ 0 & = & g(x) & (2) \\ 0 & = & g'_x(x)f(x,u) & (3) \end{array}\right.$$

$$\overset{\text{seen as}}{\Longrightarrow} \left\{\begin{array}{lcll} \dot{x} & = & f(x,u) & (1) \\ 0 & = & G(x,u) & (2,3) \end{array}\right.$$

- $\sim$ ODE if we have a constraint solver getting $x \to u$ from (2,3)

A key notion in DAE Systems: the index

$$\left[\begin{array}{l}\text{differentiations}\\ \text{shiftings}\end{array}\right] \text{ make } \left[\begin{array}{c}\text{DAE}\\ \text{dAE}\end{array}\right] \text{ becoming } \left[\begin{array}{r}\text{ODE}\\ \text{TS}\end{array}\right]\text{-like}$$

(TS: transition system)

Define the **index** as being the minimal number of

$$\left[\begin{array}{l}\text{differentiations}\\ \text{shiftings}\end{array}\right] \text{ needed until } \left[\begin{array}{l}\text{no further differentiation}\\ \text{no further shifting}\end{array}\right]$$

can reveal additional latent constraints

The notion of differentiation index emerged in the late 1980's in the applied mathematics community; other notions of index were proposed, see [Campbell & Gear 1995]

Research Agenda

- So-called index reduction is a front processing of models making DAE/dAE looking like known objects;
- The execution requires a constraint solver but no further deep forward exploration of runs is needed (warning: the index may be infinite)

- Unfortunately, no notion of index was mathematically defined for Hybrid DAE systems
  - it is informally claimed that "Hybrid DAE systems possess a mode-dependent index"
  - unfortunately this has no math basis and leads to problems at compilation: what to do at mode changes?
- This talk is about index for Hybrid DAE systems and it turns out that index for dAE systems is also interesting in itself

Research Agenda

- So-called index reduction is a front processing of models making DAE/dAE looking like known objects;
- The execution requires a constraint solver but no further deep forward exploration of runs is needed (warning: the index may be infinite)
- Unfortunately, no notion of index was mathematically defined for Hybrid DAE systems
  - it is informally claimed that
    “Hybrid DAE systems possess a mode-dependent index”
  - unfortunately this has no math basis and leads to problems at compilation: what to do at mode changes?
- This talk is about index for Hybrid DAE systems and it turns out that index for dAE systems is also interesting in itself

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

The index of a Hybrid DAE System is the dAE index of its NS-semantics

Consequences for simulation

Conclusions

Exact Differentiation/Difference Index

$$F(x, \dot{x}) = 0$$
$$F(x, x^{\bullet}) = 0$$

$$\begin{bmatrix} F(x, \dot{x}) \\ \frac{d}{dt} F(x, \dot{x}) \\ \vdots \\ \frac{d^k}{dt^k} F(x, \dot{x}) \end{bmatrix} =_{\text{def}} \overbrace{\begin{bmatrix} F^{(0)}(x, \dot{x}, w) \\ F^{(1)}(x, \dot{x}, w) \\ \vdots \\ F^{(k)}(x, \dot{x}, w) \end{bmatrix}}^{F_k(x,v,w)} , \begin{cases} v =_{\text{def}} \dot{x} \\ w =_{\text{def}} (x^{(2)}, \dots, x^{(k+1)}) \end{cases}$$

$$\begin{bmatrix} F(x, x^{\bullet}) \\ F^{\bullet}(x, x^{\bullet}) \\ \vdots \\ F^{\bullet k}(x, x^{\bullet}) \end{bmatrix} =_{\text{def}} \underbrace{\begin{bmatrix} F^{(0)}(x, x^{\bullet}, w) \\ F^{(1)}(x, x^{\bullet}, w) \\ \vdots \\ F^{(k)}(x, x^{\bullet}, w) \end{bmatrix}}_{F_k(x,v,w)} , \begin{cases} v =_{\text{def}} x^{\bullet} \\ w =_{\text{def}} (x^{\bullet 2}, \dots, x^{\bullet k+1}) \end{cases}$$

$$\boxed{\text{Index } =_{\text{def}} \text{ min } k \text{ s.t. } \quad x \to v : \exists w. F_k(x, v, w) = 0 \quad \text{ is a partial function}}$$

solving $F_k = 0$ while eliminating w uniquely determines v as a partial function of x

The case of smooth systems (F smooth)

$$\boxed{\text{Index } =_{\text{def}} \text{ min } k \text{ s.t. } \quad x \to v : \exists w. F_k(x, v, w) = 0 \quad \text{is a partial function}}$$

Whence the following questions of interest if F is smooth:

1. does $x \to v : \exists w. F(x, v, w) = 0$ define a partial function?

   $\Leftrightarrow$ (by implicit function theorem)

2. does $x \to v : \exists w. Av + Cw + Ex = 0$ define a partial function?
   where $A = F'_v, C = F'_w, E = F'_x$ are Jacobians at a solution (v_o, w_o, x_o)

The case of smooth systems (F smooth)

$$\boxed{\text{Index } =_{\text{def}} \text{ min } k \text{ s.t. } \quad x \to v : \exists w. F_k(x,v,w) = 0 \quad \text{is a partial function}}$$

Whence the following questions of interest if F is smooth:

1. does $x \to v : \exists w. F(x,v,w) = 0$ define a partial function?

   $\Leftrightarrow$ (by implicit function theorem)

2. does $x \to v : \exists w. Av + Cw + Ex = 0$ define a partial function?
   where $A = F'_v, C = F'_w, E = F'_x$ are Jacobians at a solution (v_o, w_o, x_o)

We are interested in structural properties, i.e., properties that are valid outside exceptional values for the nonzero coefficients of the matrices

The case of smooth systems (F smooth)

does $x \to v \ : \ \exists w . Av + Cw + Ex = 0$ define a partial function, almost everywhere when the nonzero coefficients of A, C, E vary over some neighborhood?

The case of smooth systems (F smooth)

- Subcase $Av + x = 0$ with A a square matrix: A structurally invertible $\Leftrightarrow \exists P$ permutation matrix such that PA has a nonzero diagonal

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}, PA = \begin{bmatrix} a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$$

A is structurally invertible. It may be singular for exceptional values of the nonzero coefficients of A, e.g., if $\det(A) = a_{31}a_{12}a_{23} - a_{32}(a_{11}a_{23} - a_{21}a_{13}) = 0$.

does $x \rightarrow v \ : \ \exists w.Av + Cw + Ex = 0$ define a partial function, almost everywhere when the nonzero coefficients of A, C, E vary over some neighborhood?

The case of smooth systems (F smooth)

- Subcase $Av + x = 0$ with A a square matrix: A structurally invertible $\Leftrightarrow \exists P$ permutation matrix such that PA has a nonzero diagonal

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}, PA = \begin{bmatrix} a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$$

  A is structurally invertible. It may be singular for exceptional values of the nonzero coefficients of A, e.g., if $\det(A) = a_{31}a_{12}a_{23} - a_{32}(a_{11}a_{23} - a_{21}a_{13}) = 0$.

- Finding P amounts to pivoting, which is a graph based algorithm:

  1. reorder equations, and then
  2. use the kth equation to eliminate v_k as a function of x and $\{v_j \mid j>k\}$

does $x \rightarrow v \ : \ \exists w. Av + Cw + Ex = 0$ define a partial function, almost everywhere when the nonzero coefficients of A, C, E vary over some neighborhood?

The case of smooth systems (F smooth)

- Subcase $Av + x = 0$ with A a square matrix: A structurally invertible $\Leftrightarrow \exists P$ permutation matrix such that PA has a nonzero diagonal

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}, PA = \begin{bmatrix} a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$$

  A is structurally invertible. It may be singular for exceptional values of the nonzero coefficients of A, e.g., if $\det(A) = a_{31}a_{12}a_{23} - a_{32}(a_{11}a_{23} - a_{21}a_{13}) = 0$.

- A similar result holds for structural properties of

$$\boxed{x \to v : \exists w . Av + Cw + Ex = 0}$$

  which also leads to a graph based algorithm

does $x \to v : \exists w . Av + Cw + Ex = 0$ define a partial function, almost everywhere when the nonzero coefficients of A, C, E vary over some neighborhood?

The case of smooth systems (F smooth)

$$\left\{\begin{array}{lcl} 0 & = & \dot{x} - f(x,u) \\ 0 & = & g(x) \end{array}\right. \quad \text{set} \quad S(x,v) =_{\text{def}} \left[\begin{array}{l} \dot{x} - f(x,u) \\ g(x) \end{array}\right]$$

where x is the state and $v =_{\text{def}} (\dot{x}, u)$; consider the Jacobian

$$\mathcal{J} =_{\text{def}} dS/dv = \left[\begin{array}{cc} 1 & -f'_u(x,u) \\ 0 & 0 \end{array}\right] \text{ is structurally singular}$$

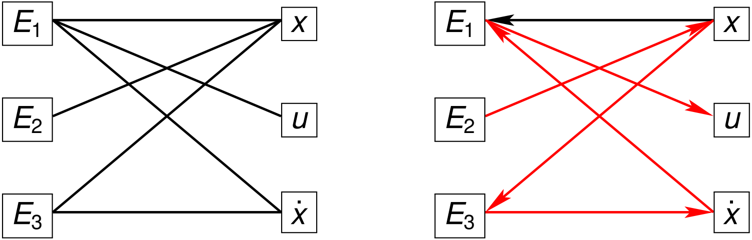
v cannot be determined and S has index > 0. Set $w = (\ddot{x}, \dot{u})$ and consider

$$S_1(x,v,w) =_{\text{def}} \left[\begin{array}{c} S(x,v) \\ \frac{d}{dt}S(x,v,w) \end{array}\right] = \left[\begin{array}{l} \dot{x} - f(x,u) \\ g(x) \\ g'(x)\dot{x} \\ \ddot{x} - \frac{d}{dt}f(x,u) \end{array}\right] \Rightarrow \left[\begin{array}{l} \dot{x} - f(x,u) \\ g(x) \\ g'(x)\dot{x} \end{array}\right]$$

$$\mathcal{J}_1 =_{\text{def}} dS_1/dv = \left[\begin{array}{cc} 1 & -f'_u(x,u) \\ 0 & 0 \\ 1 & 0 \end{array}\right] \Rightarrow \left[\begin{array}{cc} 1 & -f'_u(x,u) \\ 1 & 0 \end{array}\right] \text{ invertible}$$

The case of smooth systems (F smooth)

$$\left\{\begin{array}{lcl} E_1: & 0 & = & \dot{x} - f(x,u) \\ E_2: & 0 & = & g(x) \\ E_3: & 0 & = & g'(x)\dot{x} \end{array}\right.$$

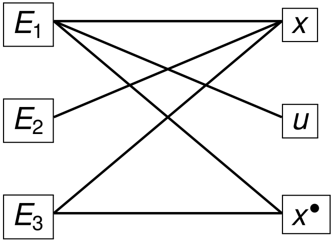


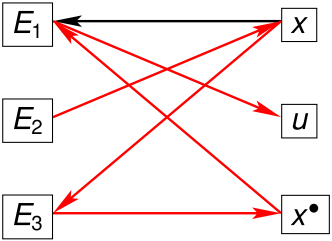
structural pivoting (Pantelides algorithm) ⇔ searching for a consistent orientation of the incidence graph

Pantelides algorithm ⇔ causality analysis for constraint system

The case of smooth systems (F smooth)

$$\left\{\begin{array}{lrcl} E_1 : & 0 & = & x^\bullet - f(x,u) \\ E_2 : & 0 & = & g(x) \\ E_3 : & 0 & = & g(x^\bullet) \end{array}\right.$$





structural pivoting (Pantelides algorithm) $\Leftrightarrow$ searching for a consistent orientation of the incidence graph

Pantelides algorithm $\Leftrightarrow$ causality analysis for constraint system

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

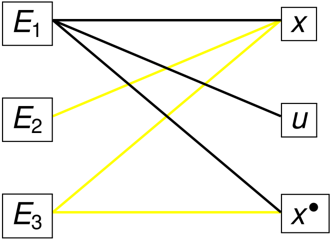
The index of a Hybrid DAE System is the dAE index of its NS-semantics

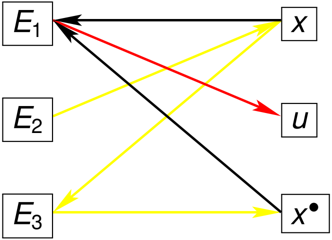
Consequences for simulation

Conclusions

The need for guards

$$\left\{\begin{array}{lcl} E_1: & 0 & = & x^\bullet - f(x,u) \\ E_2: & 0 & = & g(x) \\ E_3: & 0 & = & g(x^\bullet) \end{array}\right.$$



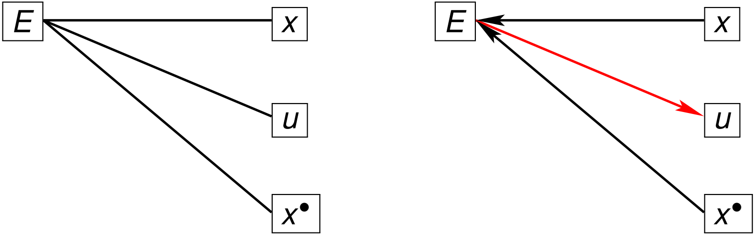


Abstraction Principle: in $F(x,u,v)=0$, any of x, u, v can be turned to an output

This is a legitimate consequence of implicit function theorem if F is smooth
What if F is not smooth? e.g., it involves “if-then-else” or guards

The need for guards

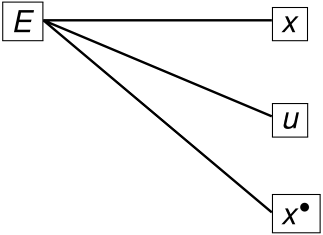
$$E(x,u,v) \quad : \quad b = [x{>}0] \ \bigwedge \ \textbf{if } b \textbf{ then } v{=}f(u) \textbf{ else } u{=}g(v)$$

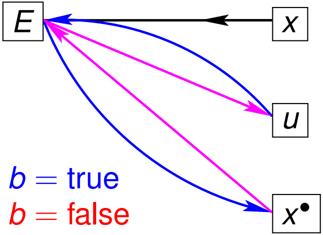


Brute force application of the Abstraction Principle yields an incorrect abstraction

The need for guards

$$E(x,u,v) \quad : \quad b = [x{>}0] \;\bigwedge\; \textbf{if } b \textbf{ then } v{=}f(u) \textbf{ else } u{=}g(v)$$





$$\mathcal{P}_E(x,u,v) \quad : \quad x{-\!\!-}b \;\bigwedge\; \textbf{if } b \textbf{ then } u{-\!\!-}v \textbf{ else } u{-\!\!-}v$$

$$\vec{\mathcal{P}}_E(x,u,v) \quad : \quad x{\rightarrow}b \;\bigwedge\; \textbf{if } b \textbf{ then } u{\rightarrow}v \textbf{ else } v{\rightarrow}u$$

The need for guards

Refined formalism: guarded equations (we consider flat guards only)

$$
\begin{aligned}
S &= \left(\bigwedge_i A_i \,,\ \bigwedge_j E_j\right) \text{ where} \\
A_i &= \text{predicate over the set of guards } b_j \\
E_j &= \textbf{if } b_j \textbf{ then } F_j(x, u, x^\bullet) \quad \text{and Abstraction Principle applies to } F
\end{aligned}
$$

$$\begin{tikzcd} & x \\ \mathcal{P}_{E_j} = b_j \to E_j \ \bigwedge \ \textbf{if } b_j \textbf{ then } F_j \arrow[ur, dash] \arrow[r, dash] \arrow[dr, dash] & u \\ & x^\bullet \end{tikzcd}$$

Compute a directed covering $\vec{\mathcal{P}}_{E_j}$ of $\mathcal{P}_{E_j}$ ensuring

- single assignment modulo assertions on guards
- circuitfreeness modulo assertions on guards

An extension of Signal clock-and-causality calculus; yields constructive semantics

Guards are not enough

$$\left.\begin{array}{r}\text{Unilateral Constraints}\\ \text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq g(x)$$

$$\left.\begin{array}{r}\text{Complementarity Conditions}\\ \text{(circuits with perfect diodes)}\\ \text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq U(x) \perp V(y) \geq 0 \equiv \left\{\begin{array}{r} U(x) \geq 0 \\ V(y) \geq 0 \\ U(x)V(y) = 0 \end{array}\right.$$

$$\left\{\begin{array}{ccc} x^\bullet & = & f(x,u) \\ 0 & \leq & g(x) \end{array}\right. \quad \overset{\text{expands as}}{\Rightarrow} \quad \left\{\begin{array}{lcl} E_1 & : & x^\bullet = f(x,u) \\ E_{21} & : & b = [0 \geq g(x)] \\ E_{22} & : & \textbf{if } b \textbf{ then } g(x)=0 \end{array}\right.$$

- Problem: $\{E_{21}, E_{22}\}$ is a fixpoint equation in (b, x)
- Approach:
  1. See $E_2 = \{E_{21}, E_{22}\}$ as an atom handled like a single equation
  2. Assign to E_2 a set of candidate causality constraints

Guards are not enough

$$\left.\begin{array}{r}\text{Unilateral Constraints}\\\text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq g(x)$$

$$\left.\begin{array}{r}\text{Complementarity Conditions}\\\text{(circuits with perfect diodes)}\\\text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq U(x) \perp V(y) \geq 0 \equiv \left\{\begin{array}{r} U(x) \geq 0 \\ V(y) \geq 0 \\ U(x)V(y) = 0 \end{array}\right.$$

$$\left\{\begin{array}{ccc} x^{\bullet} & = & f(x,u) \\ 0 & \leq & g(x) \end{array}\right. \quad \overset{\text{expands as}}{\Rightarrow} \quad \left\{\begin{array}{lcl} E_1 & : & x^{\bullet} = f(x,u) \\ E_{21} & : & b = [0 \geq g(x)] \\ E_{22} & : & \textbf{if } b \textbf{ then } g(x)=0 \end{array}\right.$$

- Problem: $\{E_{21}, E_{22}\}$ is a fixpoint equation in (b, x)
- Approach:
  1. See $E_2 = \{E_{21}, E_{22}\}$ as an atom handled like a single equation
  2. Assign to E_2 a set of candidate causality constraints

Guards are not enough

$$\left.\begin{array}{r}\text{Unilateral Constraints}\\ \text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq g(x)$$

$$\left.\begin{array}{r}\text{Complementarity Conditions}\\ \text{(circuits with perfect diodes)}\\ \text{(multi-body mechanics)}\end{array}\right\} \quad : \quad 0 \leq U(x) \perp V(y) \geq 0 \equiv \left\{\begin{array}{r} U(x) \geq 0\\ V(y) \geq 0\\ U(x)V(y) = 0\end{array}\right.$$

$$\left\{\begin{array}{lcl} x^\bullet & = & f(x,u)\\ 0 & \leq & g(x)\end{array}\right. \quad \overset{\text{expands as}}{\Rightarrow} \quad \left\{\begin{array}{lcl} E_1 & : & x^\bullet = f(x,u)\\ E_{21} & : & b = [0 \geq g(x)]\\ E_{22} & : & \textbf{if } b \textbf{ then } g(x) = 0\end{array}\right.$$

- Problem: $\{E_{21}, E_{22}\}$ is a fixpoint equation in (b, x)
- Approach:
  1. See $E_2 = \{E_{21}, E_{22}\}$ as an atom handled like a single equation
  2. Assign to E_2 a set of candidate causality constraints

Summary on dAE causality analysis

Causality analysis (from which the index follows):

- Guarded equations with assertions on guards
- Guarded causality analysis
- Atoms
  - Warning: “atom” indicates that it must be evaluated at once
  - Atoms may not be small (minimal circuits in the causality graph)

Constructive semantics and execution schemes

- Execution mode of synchronous languages, albeit
- Evaluating atoms requires dedicated constraint solvers

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

The index of a Hybrid DAE System is the dAE index of its NS-semantics

Consequences for simulation

Conclusions

Through NonStandard semantics DAE become dAE

$$\begin{aligned} \mathbb{T} &=_{\text{def}} \{t_n = n\partial \mid n \in {}^\star\mathbb{Z}\} \;\text{ where } \partial \text{ is an infinitesimal} \\ \forall t \in \mathbb{T} \;:\; {}^\bullet t &=_{\text{def}} \max\{s \mid s \in \mathbb{T}, s < t\} = t - \partial \\ t^\bullet &=_{\text{def}} \min\{s \mid s \in \mathbb{T}, s > t\} = t + \partial \end{aligned}$$

$$\dot{x}_t =_{\text{def}} \frac{x_{t^\bullet} - x_t}{\partial} \text{ (explicit scheme) } \left(\frac{x_t - x_{{}^\bullet t}}{\partial} \text{ (implicit scheme)} \right)$$

$$\begin{bmatrix} x \\ \frac{d}{dt}x \\ \frac{d^2}{dt^2}x \\ \vdots \end{bmatrix} = \mathcal{L} \begin{bmatrix} x \\ x^\bullet \\ x^{\bullet 2} \\ \vdots \end{bmatrix}, \quad \text{where } \mathcal{L} \text{ is an invertible lower triangular matrix}$$

Theorem: structural index of a DAE = structural index of its NS-semantics

Through NonStandard semantics DAE become dAE

$$\begin{array}{rrcl} & \mathbb{T} & =_{\text{def}} & \{t_n = n\partial \mid n \in {}^\star\mathbb{Z}\} \text{ where } \partial \text{ is an infinitesimal} \\ \forall t \in \mathbb{T} : & {}^\bullet t & =_{\text{def}} & \max\{s \mid s \in \mathbb{T}, s < t\} = t - \partial \\ & t^\bullet & =_{\text{def}} & \min\{s \mid s \in \mathbb{T}, s > t\} = t + \partial \end{array}$$

$$\dot{x}_t =_{\text{def}} \frac{x_{t^\bullet} - x_t}{\partial} \text{ (explicit scheme) } \left(\frac{x_t - x_{{}^\bullet t}}{\partial} \text{ (implicit scheme)} \right)$$

$$\begin{bmatrix} x \\ \frac{d}{dt}x \\ \frac{d^2}{dt^2}x \\ \vdots \end{bmatrix} = \mathcal{L} \begin{bmatrix} x \\ x^\bullet \\ x^{\bullet 2} \\ \vdots \end{bmatrix}, \quad \text{where } \mathcal{L} \text{ is an invertible lower triangular matrix}$$

Theorem: structural index of a DAE = structural index of its NS-semantics

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

The index of a Hybrid DAE System is the dAE index of its NS-semantics

Consequences for simulation

Conclusions

A conservative extension of the index

$\boxed{\text{index of Hybrid DAE System} \quad =_{\text{def}} \quad \text{dAE index of its NS-semantics}}$

- By the previous Theorem this yields a conservative extension of
  - the index of a DAE system
  - the index of a dAE system
- Warning: the above result requires considering the structural index (not the exact one)

- The computation of the index is a byproduct of the causality analysis
- The index is a global notion (the index may be finite or infinite)
- The causality analysis is guarded, i.e., mode dependent

A conservative extension of the index

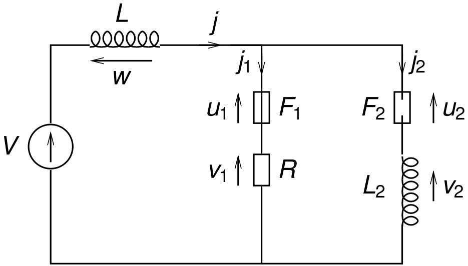
$\boxed{\text{index of Hybrid DAE System} \quad =_{\text{def}} \quad \text{dAE index of its NS-semantics}}$

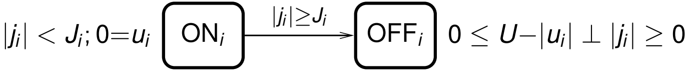
- By the previous Theorem this yields a conservative extension of
  - the index of a DAE system
  - the index of a dAE system
- Warning: the above result requires considering the structural index (not the exact one)

- The computation of the index is a byproduct of the causality analysis
- The index is a global notion (the index may be finite or infinite)
- The causality analysis is guarded, i.e., mode dependent

A conservative extension of the index

We can perform causality analysis for the following kind of example:





$|j_i| < J_i ; 0{=}u_i$ $\boxed{\text{ON}_i} \xrightarrow{|j_i| \geq J_i} \boxed{\text{OFF}_i}$ $0 \leq U{-}|u_i| \perp |j_i| \geq 0$

A simple circuit breaker. Top: the circuit. Bottom: the mode automaton for each fuse $i = 1, 2$. For the ON mode, the current must stay below a threshold J_i, while in the OFF mode, the complementarity condition shown holds.

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

The index of a Hybrid DAE System is the dAE index of its NS-semantics

Consequences for simulation

Conclusions

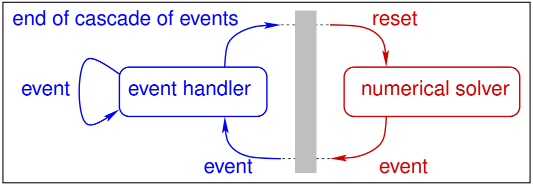
Simulating dAE systems (discrete time)

Not much to be said:

- Guarded causality analysis yields the constructive semantics
- Not very different from synchronous languages, albeit. . .
- Solvers are needed to evaluate every atom (local call to solver)

Simulating Hybrid DAE systems (continuous time)

synchronous
mode of exec



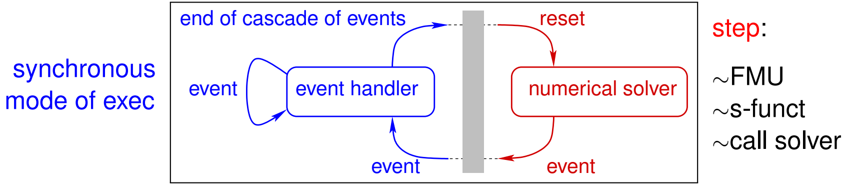
step:

∼FMU
∼s-funct
∼call solver

This is the technique of slicing used in Zelus tool [Pouzet & Bourke]

- synchronous mode of execution: may require the use of solvers for evaluating atoms
- step: a step may have “long” duration, e.g., the solver may be stopped only at the next zero-crossing

Simulating Hybrid DAE systems (continuous time)



This is the technique of slicing used in Zelus tool [Pouzet & Bourke]

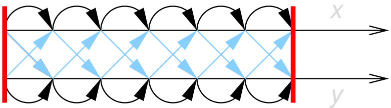
- synchronous mode of execution: may require the use of solvers for evaluating atoms
- step: a step may have “long” duration, e.g., the solver may be stopped only at the next zero-crossing

Simulating Hybrid DAE systems (continuous time)

Difficulties with this technique of slicing :

- Can the evaluation of step consist of local calls to a solver for each block? (this holds true for for dAE)
  - No if the different blocks must interact while performing step

$$\left\{\begin{array}{rcl}\dot{x} & = & f(x,y)\\ \dot{y} & = & g(x,y)\end{array}\right.$$



  - Yes otherwise
  - Ok if “slow interactions”: $f(x,y) \approx f(x,y_0)$ or $g(x,y) \approx g(x_0,y)$

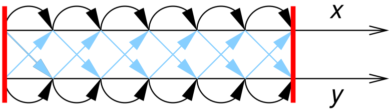
- What should be considered as an event?
  - Some but not all discontinuities (next slides)

Simulating Hybrid DAE systems (continuous time)

Difficulties with this technique of slicing :

- Can the evaluation of step consist of local calls to a solver for each block? (this holds true for for dAE)
  - No if the different blocks must interact while performing step

$$\begin{cases} \dot{x} = f(x,y) \\ \dot{y} = g(x,y) \end{cases}$$



  - Yes otherwise
  - Ok if “slow interactions”: $f(x,y) \approx f(x,y_0)$ or $g(x,y) \approx g(x_0,y)$

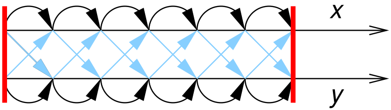
- What should be considered as an event?
  - Some but not all discontinuities (next slides)

Simulating Hybrid DAE systems (continuous time)

Difficulties with this technique of slicing :

- Can the evaluation of step consist of local calls to a solver for each block? (this holds true for for dAE)
  - No if the different blocks must interact while performing step

$$\begin{cases} \dot{x} & = & f(x, y) \\ \dot{y} & = & g(x, y) \end{cases}$$

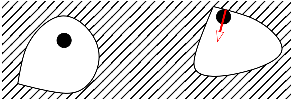


  - Yes otherwise
  - Ok if "slow interactions": $f(x, y) \approx f(x, y_0)$ or $g(x, y) \approx g(x_0, y)$

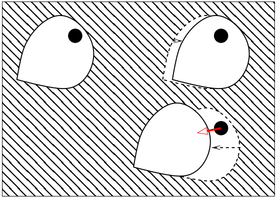
- What should be considered as an event?
  - Some but not all discontinuities (next slides)

What should be considered as an event?

Some numerical solvers ignore discontinuities (B. Caillaud at Synchron'13):



Moreau sweeping process:
A cavity moves
and pushes the ball



Its numerical scheme:
Fixed step size; no event handler
Only convex projections

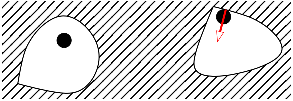
This applies to unilateral constraints and complementarity conditions

Since such solvers are not bothered by discontinuities, it is subtle to decide

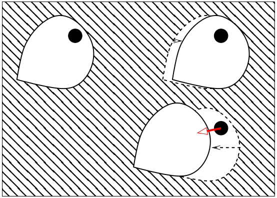
- which discontinuities to detect for handling as an event vs.
- which discontinuities to ignore and delegate to the solver

What should be considered as an event?

Some numerical solvers ignore discontinuities (B. Caillaud at Synchron’13):



Moreau sweeping process:
A cavity moves
and pushes the ball



Its numerical scheme:
Fixed step size; no event handler
Only convex projections

This applies to unilateral constraints and complementarity conditions

Since such solvers are not bothered by discontinuities, it is subtle to decide

- which discontinuities to detect for handling as an event vs.
- which discontinuities to ignore and delegate to the solver

Current status for Hybrid DAE System modeling tools

General scope tools for engineering (e.g., Modelica):

- Spurious behaviors may be encountered when handling events (cascades of)
- Discontinuities are handled as events
  - unless the engineer manually enforces the use of certain event-agnostic discretization schemes
- Causality analysis is mode-dependent
  - Still, separately compiled blocks (e.g., targeting FMU) must have an interface with static (mode-independent) causality
- No local solvers
  - unless manually enforced (e.g., for slow/fast dynamics)

Current status for Hybrid DAE System modeling tools

Solvers dedicated to nonsmooth systems (e.g., Siconos library [Acary & Brogliato])

- eventless and event based processing both supported
- global solvers
- complementarity conditions with linear coupling
- no index reduction; replaced by the evaluation of the “relative degree”

Exact and Structural DAE index (linear algebra reasoning)

Structural dAE index and causality analysis

Through NonStandard semantics DAE become dAE

The index of a Hybrid DAE System is the dAE index of its NS-semantics

Consequences for simulation

Conclusions

We have formally defined the index for Hybrid DAE

- Index of Hybrid DAE $=_{\text{def}}$ index of its NonStandard semantics
  (Yet another evidence that NonStandard semantics helps. . .)

- Requires guarded causality analysis alike in synchronous languages (particularly Signal)

- Allows giving a mathematical semantics to more Hybrid DAE systems (of little help if the index is infinite)

dAE Systems are interesting

dAE Systems with general data types (e.g. Bool $=$ numerics):

- Extend synchronous programming to Transition Systems where transition relations (constraints) are specified via systems of equations
- Guarded equations with atoms form an expressive syntax
- Index analysis for dAE is new (though an easy translation from DAE):
  - relies on guarded causality analysis
  - when the index is finite, index reduction identifies the look-ahead horizon that is sufficient to avoid future blocking
- Requires constraint solvers (no all purpose solver exists. . .)

What are dAE Systems useful for?

- model-guided testing?
- planification?
- . . . ?

dAE Systems are interesting

dAE Systems with general data types (e.g. Bool $=$ numerics):

- Extend synchronous programming to Transition Systems where transition relations (constraints) are specified via systems of equations
- Guarded equations with atoms form an expressive syntax
- Index analysis for dAE is new (though an easy translation from DAE):
    - relies on guarded causality analysis
    - when the index is finite, index reduction identifies the look-ahead horizon that is sufficient to avoid future blocking
- Requires constraint solvers (no all purpose solver exists. . .)

What are dAE Systems useful for?

- model-guided testing?
- planification?
- . . . ?

Hybrid DAE Systems are more difficult than dAE Systems

- Index-and-causality analysis is a symbolic pre-processing
  It does not address numerical difficulties
- Making a step is more difficult than calling an S-function
- Global vs. Distributed solvers
  - Global solver is normally used for simulation
  - Distributed solvers are used:
    - in code coupling (e.g. multi-physics)
    - in slow/fast dynamics
    - in FMI based simulation with several FMU
- What to do with events?
  - Handling every discontinuity as an event is not good
  - Handling no discontinuity as an event is not good either

Hybrid DAE Systems are more difficult than dAE Systems

- Index-and-causality analysis is a symbolic pre-processing
  It does not address numerical difficulties
- Making a step is more difficult than calling an S-function
- Global vs. Distributed solvers
  - Global solver is normally used for simulation
  - Distributed solvers are used:
    - in code coupling (e.g. multi-physics)
    - in slow/fast dynamics
    - in FMI based simulation with several FMU
- What to do with events?
  - Handling every discontinuity as an event is not good
  - Handling no discontinuity as an event is not good either

Hybrid DAE Systems are more difficult than dAE Systems

- Index-and-causality analysis is a symbolic pre-processing
  It does not address numerical difficulties

- Making a step is more difficult than calling an S-function

- Global vs. Distributed solvers
  - Global solver is normally used for simulation
  - Distributed solvers are used:
    - in code coupling (e.g. multi-physics)
    - in slow/fast dynamics
    - in FMI based simulation with several FMU

- What to do with events?
  - Handling every discontinuity as an event is not good
  - Handling no discontinuity as an event is not good either

Hybrid DAE Systems are more difficult than dAE Systems

- Index-and-causality analysis is a symbolic pre-processing
  It does not address numerical difficulties
- Making a step is more difficult than calling an S-function
- Global vs. Distributed solvers
  - Global solver is normally used for simulation
  - Distributed solvers are used:
    - in code coupling (e.g. multi-physics)
    - in slow/fast dynamics
    - in FMI based simulation with several FMU
- What to do with events?
  - Handling every discontinuity as an event is not good
  - Handling no discontinuity as an event is not good either

an exciting but difficult subject