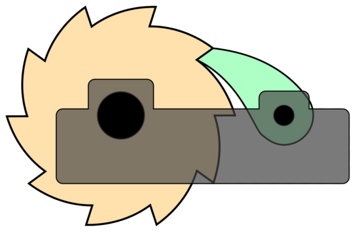
Mathematical Foundations of Physical Systems Modeling Languages

Albert Benveniste Benoît Caillaud
Khalil Ghorbal Mathias Malandain
(Inria, Rennes)

November 25, 2019

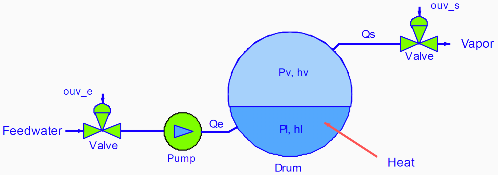
Multimode (aka. hybrid) systems

- Natural models for physical phenomena
  - mechanics (engagement/release of links)



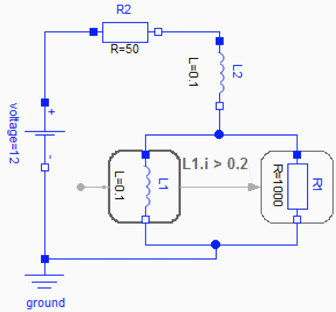
Multimode (aka. hybrid) systems

- Natural models for physical phenomena
  - mechanics (engagement/release of links)
  - thermodynamics (phase (dis)appearance)
  - hydraulics (opening/closing of a valve)
  - electronics (switching diode/transistor)



Multimode (aka. hybrid) systems

- Natural models for physical phenomena
  - mechanics (engagement/release of links)
  - thermodynamics (phase (dis)appearance)
  - hydraulics (opening/closing of a valve)
  - electronics (switching diode/transistor)
- Fault modeling (component break)

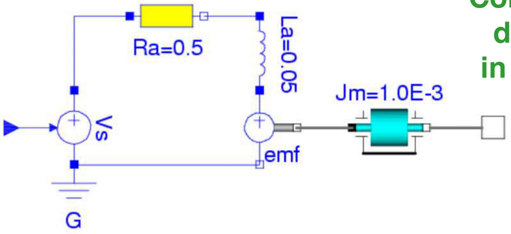


Multimode (aka. hybrid) systems

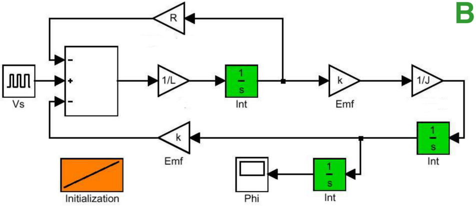
- Natural models for physical phenomena
  - mechanics (engagement/release of links)
  - thermodynamics (phase (dis)appearance)
  - hydraulics (opening/closing of a valve)
  - electronics (switching diode/transistor)
- Fault modeling (component break)
- Reconfigurable systems ((dis)appearance of components)

Compositionality and reuse: Simulink → Modelica

From Block Diagram to Component Diagram



Component diagram in Dymola



Block diagram in Simulink

Component diagrams generalize Block diagrams
=> **The next generation of simulation tools**

Compositionality and reuse: ODE $\rightarrow$ DAE

from Simulink (ODE):
HS in state space form

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

the state space form
depends on the context

reuse is difficult

$\longrightarrow$

to Modelica (DAE):
HS as physical balance equations

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

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

reuse is much easier

Compositionality and reuse: ODE $\rightarrow$ DAE

- Modeling tools supporting DAE
  - Most modeling tools provide a library of predefined models ready for assembly (Mathworks/Simscape, Siemens-LMS/AmeSim, Mathematica/NDSolve)
  - Modelica comes with a full programming language that is a public standard `https://www.modelica.org/` ;
  - Simscape and NDSolve use Matlab extended with "$==$"
  - Also Spice dedicated to EDA

A sketch of Modelica and its semantics [Fritzson]

- Modelica = DAE + Objects
- Class = container for equations

```
model SimpleDrive
   ..Rotational.Inertia   Inertia1  (J=0.002);
   ..Rotational.IdealGear IdealGear1(ratio=100)
   ..Basic.Resistor       Resistor1 (R=0.2)
      ...
equation
   connect(Inertia1.flange_b, IdealGear1.flange_a);
   connect(Resistor1.n, Inductor1.p);
      ...
end SimpleDrive;
```

```
model Resistor
   package SIunits = Modelica.SIunits;
   parameter SIunits.Resistance R = 1;
   SIunits.Voltage v;
   ..Interfaces.PositivePin p;
   ..Interfaces.NegativePin n;
equation
   0 = p.i + n.i;
   v = p.v - n.v;
   v = R*p.i;
end Resistor;
```

```
type Voltage =
     Real(quantity="Voltage",
          unit    ="V");
```

```
connector PositivePin
   package SIunits = Modelica.SIunits;
   SIunits.Voltage      v;
   flow SIunits.Current i;
end PositivePin;
```

A sketch of Modelica and its semantics [Fritzson]

- Modelica Reference v3.3:

  > *"The semantics of the Modelica language is specified by means of a set of rules for translating any class described in the Modelica language to a flat Modelica structure"*

- **the good:**
  - Semantics of continuous-time 1-mode Modelica models: Cauchy problem on the DAE resulting from the inlining of all components
  - Modelica supports multi-mode systems

```
x*x + y*y = 1;
der(x) + x + y = 0;
when x <= 0 do reinit(x,1); end;
when y <= 0 do reinit(y,x); end;
```

- **the bad:** What about the semantics of multi-mode systems?
- **and . . . :** Questionable simulations of many physically meaningful models

Examples of multi-mode systems

Cup-and-Ball game (a two-mode extension of the pendulum)

⇒ A Clutch

A Circuit Breaker

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{lllll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) & \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) & \\ \texttt{when}\ \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) & \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) & \cdots \\ \texttt{when not}\ \gamma & \texttt{do} & \tau_1 = 0 & (e_5) & \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) & \cdots \end{array}\right.$$

Note that $\tau_1 + \tau_2 = 0$ holds in all modes, reflecting that the angular momentum is preserved everywhere (we assume that the system is closed)

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{lllll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) & \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) & \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) & \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) & \cdots \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) & \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) & \cdots \end{array}\right.$$

Mode $\gamma = \mathrm{F}$: it is just an ODE system, nothing fancy

$$\left\{\begin{array}{ll} \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \tau_1 = 0 & (e_5) \\ \tau_2 = 0 & (e_6) \end{array}\right.$$

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{llll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \quad \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \quad \cdots \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \quad \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) \quad \cdots \end{array}\right.$$

Mode $\gamma = \text{T}$: it is now a DAE system

$$\left\{\begin{array}{ll} \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \omega_1 - \omega_2 = 0 & (e_3) \\ & \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right.$$

Looking for an execution scheme? Try a 1^{st}-order Euler scheme

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{llll} & & \omega'_1 = f_1(\omega_1, \tau_1) & (e_1) \\ & & \omega'_2 = f_2(\omega_2, \tau_2) & (e_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \quad \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \quad \cdots \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \quad \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) \quad \cdots \end{array}\right.$$

Mode $\gamma = \mathrm{T}$: it is now a dAE system

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \delta.f_1(\omega_1, \tau_1) & (e_1^\delta) \\ \omega_2^\bullet = \omega_2 + \delta.f_2(\omega_2, \tau_2) & (e_2^\delta) \\ \omega_1 - \omega_2 = 0 & (e_3) \\ & \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right. \qquad (1)$$

Regard (1) as a transition system: for a given (ω_1, ω_2) satisfying (e_3),
find $(\omega_1^\bullet, \omega_2^\bullet, \tau_1, \tau_2)$ using eqns $(e_1^\delta, e_2^\delta, e_4)$.
We have 4 unknowns but only 3 eqns: structurally singular

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{llll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \quad \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \quad \cdots \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \quad \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) \quad \cdots \end{array}\right.$$

Mode $\gamma = \mathrm{T}$: it is now a dAE system

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \delta . f_1(\omega_1, \tau_1) & (e_1^\delta) \\ \omega_2^\bullet = \omega_2 + \delta . f_2(\omega_2, \tau_2) & (e_2^\delta) \\ \omega_1 - \omega_2 = 0 & (e_3) \\ \omega_1^\bullet = \omega_2^\bullet & (e_3^\bullet) \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right. \qquad (2)$$

Regard (2) as an algebraic system of eqns: for a given (ω_1, ω_2) satisfying (e_3), find $(\omega_1^\bullet, \omega_2^\bullet, \tau_1, \tau_2)$ using eqns $(e_1^\delta, e_2^\delta, e_3^\bullet, e_4)$: structurally nonsingular.

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{llll} & & \omega_1' = f_1(\omega_1,\tau_1) & (e_1) \\ & & \omega_2' = f_2(\omega_2,\tau_2) & (e_2) \\ \texttt{when}\ \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \quad \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \quad \cdots \\ \texttt{when not}\ \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \quad \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) \quad \cdots \end{array}\right.$$

Mode $\gamma = \mathrm{T}$: it is now a DAE system

$$\left\{\begin{array}{ll} \omega_1' = f_1(\omega_1,\tau_1) & (e_1) \\ \omega_2' = f_2(\omega_2,\tau_2) & (e_2) \\ \omega_1 - \omega_2 = 0 & (e_3) \\ \omega_1' = \omega_2' & (e_3') \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right. \tag{3}$$

Regard (3) as an algebraic system of eqns: for a given (ω_1, ω_2) satisfying (e_3), find $(\omega_1', \omega_2', \tau_1, \tau_2)$ using eqns (e_1, e_2, e_3', e_4): structurally nonsingular.

The clutch example: separate analysis of each mode

$$\left\{\begin{array}{llll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \quad \text{clutch engaged} \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \quad \cdots \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \quad \text{clutch released} \\ & \texttt{and} & \tau_2 = 0 & (e_6) \quad \cdots \end{array}\right.$$

Mode $\gamma = \mathrm{T}$: it is now a DAE system

$$\left\{\begin{array}{ll} \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \omega_1 - \omega_2 = 0 & (e_3) \\ \omega_1' = \omega_2' & (e_3') \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right. \tag{4}$$

- The structural analyses we performed
  - in continuous time, and
  - in discrete time using Euler schemes

mirror each other (this is a general fact)

The clutch example: mode transitions, difficulties

$$
\left\{
\begin{array}{llll}
 & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\
 & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\
\texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \\
 & \texttt{and} & \omega_1' - \omega_2' = 0 & (e_3') \\
 & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \\
\texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \\
 & \texttt{and} & \tau_2 = 0 & (e_6)
\end{array}
\right.
$$

- Guess: structural analysis in each mode is enough
- Problems:
  - reset $\neq$ initialization
    (initialization has 1 degree of freedom in mode $\gamma = \text{T}$)
  - transition *released* $\rightarrow$ *engaged* has impulsive torques
    (to adjust the rotation speeds in zero time)

The results obtained by Modelica and Mathematica are interesting

The clutch example: mode transitions, difficulties

$$\left\{\begin{array}{rlll} & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \\ & \texttt{and} & \omega_1' - \omega_2' = 0 & (e_3') \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \\ & \texttt{and} & \tau_2 = 0 & (e_6) \end{array}\right.$$

- Guess: structural analysis in each mode is enough
- Problems:
  - reset $\neq$ initialization
    (initialization has 1 degree of freedom in mode $\gamma = \text{T}$)
  - transition *released* $\rightarrow$ *engaged* has impulsive torques
    (to adjust the rotation speeds in zero time)

The results obtained by Modelica and Mathematica are interesting

The clutch example: mode transitions, difficulties

$$\left\{\begin{array}{lll} & & \omega_1' = f_1(\omega_1, \tau_1) \qquad (e_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) \qquad (e_2) \\ \texttt{when}\ \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \qquad (e_3) \\ & \texttt{and} & \omega_1' - \omega_2' = 0 \qquad (e_3') \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \qquad (e_4) \\ \texttt{when not}\ \gamma & \texttt{do} & \tau_1 = 0 \qquad (e_5) \\ & \texttt{and} & \tau_2 = 0 \qquad (e_6) \end{array}\right.$$

- Guess: structural analysis in each mode is enough
- Problems:
  - reset $\neq$ initialization
    (initialization has 1 degree of freedom in mode $\gamma = \mathrm{T}$)
  - transition *released* $\rightarrow$ *engaged* has impulsive torques
    (to adjust the rotation speeds in zero time)

The results obtained by Modelica and Mathematica are interesting

The clutch example: mode transitions, difficulties

$$
\left\{
\begin{array}{rlll}
 & & \omega_1' = f_1(\omega_1, \tau_1) & (e_1) \\
 & & \omega_2' = f_2(\omega_2, \tau_2) & (e_2) \\
\texttt{when}\ \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \\
 & \color{red}{\texttt{and}} & \color{red}{\omega_1' - \omega_2' = 0} & \color{red}{(e_3')} \\
 & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \\
\texttt{when not}\ \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \\
 & \texttt{and} & \tau_2 = 0 & (e_6)
\end{array}
\right.
$$

- Guess: structural analysis in each mode is enough
- Problems:
  - reset $\neq$ initialization
    (initialization has 1 degree of freedom in mode $\gamma = \mathrm{T}$)
  - transition *released* $\rightarrow$ *engaged* has impulsive torques
    (to adjust the rotation speeds in zero time)

The results obtained by Modelica and Mathematica are interesting

The clutch in Modelica and Mathematica

Clutch

$$\left\{\begin{array}{rll} & & \omega_1' = f_1(\omega_1, \tau_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) \\ \texttt{when}\ \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \\ \texttt{when not}\ \gamma & \texttt{do} & \tau_1 = 0 \\ & \texttt{and} & \tau_2 = 0 \end{array}\right.$$

Changes γ : $\texttt{F} \to \texttt{T} \to \texttt{F}$ at $t = 5, 10$

When the clutch gets engaged, an impulsive torque occurs if the two rotation speeds differed before the engagement. The common speed after engagement should sit between the two speeds before it.

The clutch in Modelica and Mathematica

Clutch in Modelica

$$\begin{cases} & & \omega_1' = f_1(\omega_1, \tau_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \\ & \texttt{and} & \tau_2 = 0 \end{cases}$$

Changes γ : F $\rightarrow$ T $\rightarrow$ F at $t = 5, 10$

```
The following error was detected at time:
5.002
Error:  Singular inconsistent scalar system
for f1 = ((if g then w1-w2 else 0.0))/(-(if
g then 0.0 else 1.0)) = -0.502621/-0
Integration terminated before reaching
"StopTime" at T = 5
```

```
model ClutchBasic
  parameter Real w01=1;
  parameter Real w02=1.5;
  parameter Real j1=1;
  parameter Real j2=2;
  parameter Real k1=0.01;
  parameter Real k2=0.0125;
  parameter Real t1=5;
  parameter Real t2=7;
  Real t(start=0, fixed=true);
  Boolean g(start=false);
  Real w1(start = w01, fixed=true);
  Real w2(start = w02, fixed=true);
  Real f1;
  Real f2;
equation
  der(t) = 1;
  g = (t >= t1) and (t <= t2);
  j1*der(w1) = -k1*w1 + f1;
  j2*der(w2) = -k2*w2 + f2;
  0 = if g then w1-w2 else f1;
  f1 + f2 = 0;
end ClutchBasic;
```

The clutch in Modelica and Mathematica

Clutch in Modelica

$$\left\{\begin{array}{lll} & & \omega_1' = f_1(\omega_1, \tau_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \\ & \texttt{and} & \tau_2 = 0 \end{array}\right.$$

Changes γ : F $\rightarrow$ T $\rightarrow$ F at $t = 5, 10$

The reason is that Dymola has symbolically pivoted the system of equations, regardless of the mode.
By doing so, it has produced an equation defining `f1` that is singular in mode `g`.

```
model ClutchBasic
  parameter Real w01=1;
  parameter Real w02=1.5;
  parameter Real j1=1;
  parameter Real j2=2;
  parameter Real k1=0.01;
  parameter Real k2=0.0125;
  parameter Real t1=5;
  parameter Real t2=7;
  Real t(start=0, fixed=true);
  Boolean g(start=false);
  Real w1(start = w01, fixed=true);
  Real w2(start = w02, fixed=true);
  Real f1;
  Real f2;
equation
  der(t) = 1;
  g = (t >= t1) and (t <= t2);
  j1*der(w1) = -k1*w1 + f1;
  j2*der(w2) = -k2*w2 + f2;
  0 = if g then w1-w2 else f1;
  f1 + f2 = 0;
end ClutchBasic;
```

The clutch in Modelica and Mathematica

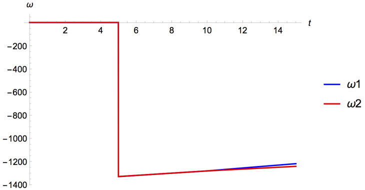
Clutch in Mathematica

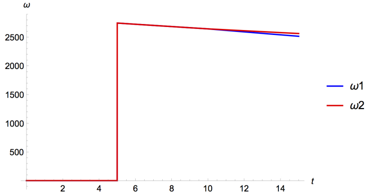
$$\left\{ \begin{array}{lll} & & \omega_1' = f_1(\omega_1, \tau_1) \\ & & \omega_2' = f_2(\omega_2, \tau_2) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \\ & \texttt{and} & \tau_2 = 0 \end{array} \right.$$

Changes γ : F $\rightarrow$ T $\rightarrow$ F at $t = 5, 10$

The simulation does not crash but yields meaningless results highly sensitive to little variations of some parameters.
This suggests that a cold restart, not a reset, is performed.

```
NDSolve[{
   w1'[t] == -0.01 w1[t] + t1[t],
   2 w2'[t] == -0.0125 w2[t] + t2[t],
   t1[t] + t2[t] == 0,
   s[t] (w1[t] - w2[t]) + (1 - s[t]) t1[t] == 0,
   w1[0] == 1.0, w2[0] == 1.5, s[0] == 0,
   WhenEvent[t == 5,
         s[t] -> 1
         ] },
   w1, w2, t1, t2,s,
   t, 0, 7, DiscreteVariables -> s]
```





The clutch example

Separate analysis of each mode

The mode transitions: difficulties

The clutch example: a comprehensive approach

Nonstandard structural analysis

Back-Standardization

Results and code for the clutch

The Cup-and-Ball example: handling transient modes

Making this work in general

Structural analysis

Standardization

Further results

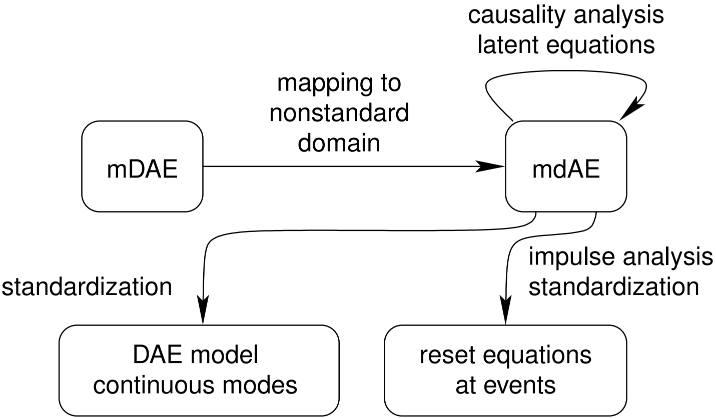
Escaping from standardization

What if we change the nonstandard representation of x'?

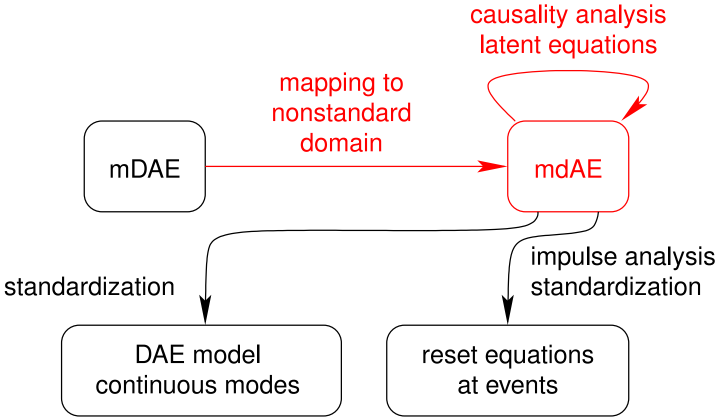
Correctness of our notion of solution of multimode DAE

Conclusion

Overview of our approach



Nonstandard structural analysis



Nonstandard analysis

- Nonstandard reals ${}^{\star}\mathbb{R} \supset \mathbb{R}$: conservative extension offering infinitesimals (smaller in size than any $\epsilon{>}0$) and infinities; $+, \times$, etc. extend to ${}^{\star}\mathbb{R}$
- Standardization: say $x \approx y$ if $x - y$ is infinitesimal
  Each finite $x \in {}^{\star}\mathbb{R}$ has a unique standard part $st(x) \in \mathbb{R}$ such that $x \approx st(x)$
- Nonstandard integers ${}^{\star}\mathbb{Z} \supset \mathbb{Z}$: conservative extension offering infinities

Using it

- Pick $\partial > 0$ infinitesimal and consider $\mathbb{T} = \{n\partial \mid n \in {}^{\star}\mathbb{Z}\}$; $\mathbb{T}$ is both
  - discrete: each $n\partial$ has a predecessor $(n{-}1)\partial$ and a successor $(n{+}1)\partial$
  - dense in $\mathbb{R}$: each $t \in \mathbb{R}$ has a $\tau \in \mathbb{T}$ such that $t \approx \tau$
- We will "faithfully" approximate multimode DAE systems by multimode dAE systems over $\mathbb{T}$; reverse mapping by performing standardization
  - "faithful" means "up to an infinitesimal error"
  - this way we unify continuous dynamics and mode changes

Nonstandard analysis

- Nonstandard reals ${}^\star\mathbb{R} \supset \mathbb{R}$: conservative extension offering infinitesimals (smaller in size than any $\epsilon{>}0$) and infinities; $+, \times,$ etc. extend to ${}^\star\mathbb{R}$
- Standardization: say $x \approx y$ if $x - y$ is infinitesimal
  Each finite $x \in {}^\star\mathbb{R}$ has a unique standard part $st(x) \in \mathbb{R}$ such that $x \approx st(x)$
- Nonstandard integers ${}^\star\mathbb{Z} \supset \mathbb{Z}$: conservative extension offering infinities

Using it

- Pick $\partial > 0$ infinitesimal and consider $\mathbb{T} = \{n\partial \mid n \in {}^\star\mathbb{Z}\}$; $\mathbb{T}$ is both
  - discrete: each $n\partial$ has a predecessor $(n{-}1)\partial$ and a successor $(n{+}1)\partial$
  - dense in $\mathbb{R}$: each $t \in \mathbb{R}$ has a $\tau \in \mathbb{T}$ such that $t \approx \tau$
- We will “faithfully” approximate multimode DAE systems by multimode dAE systems over $\mathbb{T}$; reverse mapping by performing standardization
  - “faithful” means “up to an infinitesimal error”
  - this way we unify continuous dynamics and mode changes

Nonstandard structural analysis everywhere

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{lll}
 & & \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1, \tau_1) \qquad (e_1^\partial) \\
 & & \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2, \tau_2) \qquad (e_2^\partial) \\
\texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \qquad (e_3) \\
 & \texttt{and} & \omega_1^\bullet - \omega_2^\bullet = 0 \qquad (e_3^\bullet) \\
 & \texttt{and} & \tau_1 + \tau_2 = 0 \qquad (e_4) \\
\texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \qquad (e_5) \\
 & \texttt{and} & \tau_2 = 0 \qquad (e_6)
\end{array}\right.$$

Latent equations were added for each mode

Nonstandard structural analysis everywhere

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$
\left\{
\begin{array}{lll}
 & & \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1, \tau_1) \quad (e_1^\partial) \\
 & & \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2, \tau_2) \quad (e_2^\partial) \\
\texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \quad (e_3) \\
 & \texttt{and} & \omega_1^\bullet - \omega_2^\bullet = 0 \quad (e_3^\bullet) \\
 & \texttt{and} & \tau_1 + \tau_2 = 0 \quad (e_4) \\
\texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \quad (e_5) \\
 & \texttt{and} & \tau_2 = 0 \quad (e_6)
\end{array}
\right.
$$

Latent equations were added for each mode

This is ok for each of the two modes:

- $\gamma = \texttt{T}$: activate $(e_3, e_3^\bullet, e_4)$ and disable (e_5, e_6)
- $\gamma = \texttt{F}$: disable $(e_3, e_3^\bullet, e_4)$ and activate (e_5, e_6)

Nonstandard structural analysis everywhere

∂ infinitesimal; ${}^\star\mathbb{T} \;=_{\text{def}}\; \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{lll}
 & & \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1, \tau_1) \qquad (e_1^\partial) \\
 & & \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2, \tau_2) \qquad (e_2^\partial) \\
\texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \qquad (e_3) \\
 & \texttt{and} & \omega_1^\bullet - \omega_2^\bullet = 0 \qquad (e_3^\bullet) \\
 & \texttt{and} & \tau_1 + \tau_2 = 0 \qquad (e_4) \\
\texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \qquad (e_5) \\
 & \texttt{and} & \tau_2 = 0 \qquad (e_6)
\end{array}\right.$$

Latent equations were added for each mode

Structural conflict at mode change γ : F $\rightarrow$ T, between “past” and “present”:

$$\left\{\begin{array}{ll}
\omega_1 = {}^\bullet\omega_1 + \partial.f_1({}^\bullet\omega_1, {}^\bullet\tau_1) & ({}^\bullet e_1^\partial) \\
\omega_2 = {}^\bullet\omega_2 + \partial.f_2({}^\bullet\omega_2, {}^\bullet\tau_2) & ({}^\bullet e_2^\partial) \\
\omega_1 - \omega_2 = 0 & (e_3)
\end{array}\right.$$

Nonstandard structural analysis everywhere

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{lllr} & & \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1,\tau_1) & (e_1^\partial) \\ & & \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2,\tau_2) & (e_2^\partial) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 & (e_3) \\ & \texttt{and} & \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \\ & \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 & (e_5) \\ & \texttt{and} & \tau_2 = 0 & (e_6) \end{array}\right.$$

Latent equations were added for each mode

Structural conflict at mode change γ : F $\rightarrow$ T:

$$\left\{\begin{array}{lr} \omega_1 = {}^\bullet\omega_1 + \partial.f_1({}^\bullet\omega_1, {}^\bullet\tau_1) & ({}^\bullet e_1^\partial) \\ \omega_2 = {}^\bullet\omega_2 + \partial.f_2({}^\bullet\omega_2, {}^\bullet\tau_2) & ({}^\bullet e_2^\partial) \\ \omega_1 - \omega_2 = 0 & (e_3)\text{: inhibited} \end{array}\right.$$

Priority is given to the past: no UNDO

Nonstandard structural analysis everywhere

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{rlr} & \omega_1 = {}^\bullet\omega_1 + \partial.f_1({}^\bullet\omega_1, {}^\bullet\tau_1) & ({}^\bullet e_1^\partial) \\ & \omega_2 = {}^\bullet\omega_2 + \partial.f_2({}^\bullet\omega_2, {}^\bullet\tau_2) & ({}^\bullet e_2^\partial) \\ & \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1, \tau_1) & (e_1^\partial) \\ & \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2, \tau_2) & (e_2^\partial) \\ & & \\ & \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \\ \texttt{and} & \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right.$$

Resulting code at mode change $\gamma : \mathtt{F} \rightarrow \mathtt{T}$:

- context inherited from the past
- latent equations
- other equations
- (e_3) inhibited

Nonstandard structural analysis: summary

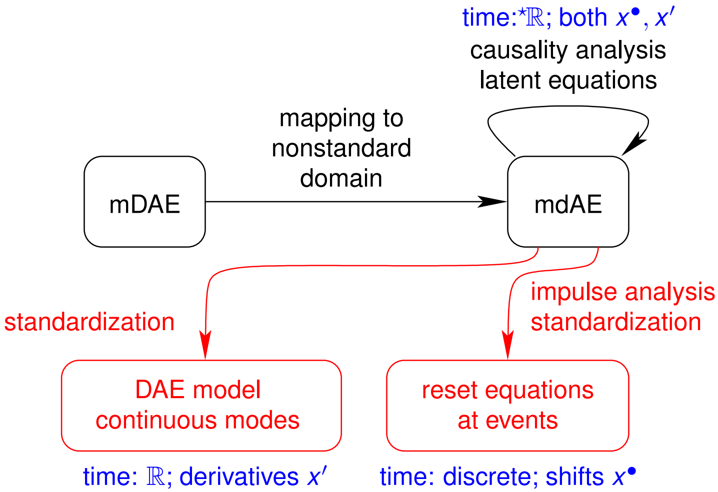
1. Each continuous mode comes with its own structural analysis

2. Conflicts may occur at mode changes, between:
   - state predictions from the previous mode, and
   - consistency conditions of the current mode

3. Resolve such possible conflicts by removing, at the instant of mode change, the consistency equations conflicting with the predictions from previous mode

   - As a result, conflicts are postponed for finitely many nonstandard instants (which accounts for zero standard time)

Back-Standardization



Back-Standardization

∂ infinitesimal; ${}^{\star}\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^{\star}\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{lll} & & \omega_1^{\bullet} = \omega_1 + \partial.f_1(\omega_1, \tau_1) \quad (e_1^{\partial}) \\ & & \omega_2^{\bullet} = \omega_2 + \partial.f_2(\omega_2, \tau_2) \quad (e_2^{\partial}) \\ \texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \quad (e_3) \\ & \texttt{and} & \omega_1^{\bullet} - \omega_2^{\bullet} = 0 \quad (e_3^{\bullet}) \\ & \texttt{and} & \tau_1 + \tau_2 = 0 \quad (e_4) \\ \texttt{when not } \gamma & \texttt{do} & \tau_1 = 0 \quad (e_5) \\ & \texttt{and} & \tau_2 = 0 \quad (e_6) \end{array}\right.$$

OK inside each individual mode:

- Apply usual continuous time structural analysis for each mode, separately
- This can be justified by the standardization of the nonstandard model, which consists in moving backward, from nonstandard Euler scheme to DAE
- The usual numerical schemes for DAE can then be applied

Back-Standardization

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial.f_1(\omega_1, \tau_1) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial.f_2(\omega_2, \tau_2) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right.$$

This is the nonstandard dynamics at mode change $\gamma : \mathtt{F} \to \mathtt{T}$.

How to map it to effective code?

- again by standardization, by, however
- targeting discrete time dynamics (for stepwise computing restart values)
- the issue: getting rid of the infinitesimal ∂ acting in space in $(e_1^\partial, e_2^\partial)$

Let us proceed for the simple case where $f_i(\omega_i, \tau_i) = a_i\omega_i + b_i\tau_i$

Back-Standardization

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial(a_1\omega_1 + b_1\tau) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial(a_2\omega_2 - b_2\tau) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \end{array}\right.$$

1st step: impulse analysis:

- Since ∂ occurs in $(e_1^\partial, e_2^\partial)$, $\omega_1^\bullet - \omega_2^\bullet = 0$ and $\omega_1 - \omega_2 \neq 0$ together require either $(a_1\omega_1 + b_1\tau)$ or $(a_2\omega_2 - b_2\tau)$ to be infinite (in nonstandard setting)
- Requires τ to be infinite, expressing that τ is impulsive

Back-Standardization

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial(a_1\omega_1 + b_1\tau) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial(a_2\omega_2 - b_2\tau) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \end{array}\right.$$

2nd step: eliminating impulsive variables:

- Since ∂ occurs in $(e_1^\partial, e_2^\partial)$, $\omega_1^\bullet - \omega_2^\bullet = 0$ and $\omega_1 - \omega_2 \neq 0$ together require either $(a_1\omega_1 + b_1\tau)$ or $(a_2\omega_2 - b_2\tau)$ to be infinite (in nonstandard setting)
- Requires τ to be infinite, expressing that τ is impulsive
- We cannot directly set $\partial \leftarrow 0$ because this yields a singular system

$\Rightarrow$ Need to eliminate τ. Having done this, setting $\omega^\bullet =_{\text{def}} \omega_1^\bullet = \omega_2^\bullet$, we get:

$$(b_1 + b_2)\omega^\bullet = b_2(1+\partial.a_1)\omega_1 + b_1(1+\partial.a_2)\omega_2$$

Back-Standardization

∂ infinitesimal; ${}^\star\mathbb{T} =_{\text{def}} \{n.\partial \mid n \in {}^\star\mathbb{N}\}$; nonstandard clutch model:

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial(a_1\omega_1 + b_1\tau) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial(a_2\omega_2 - b_2\tau) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \end{array}\right.$$

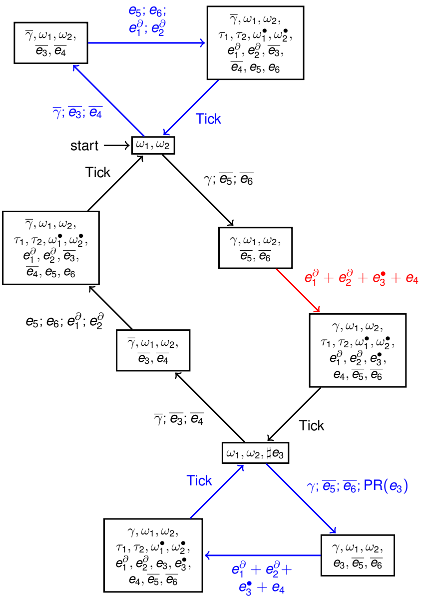
3rd step: perform safe standardization of

$$(b_1 + b_2)\omega^\bullet = b_2(1{+}\partial.a_1)\omega_1 + b_1(1{+}\partial.a_2)\omega_2$$

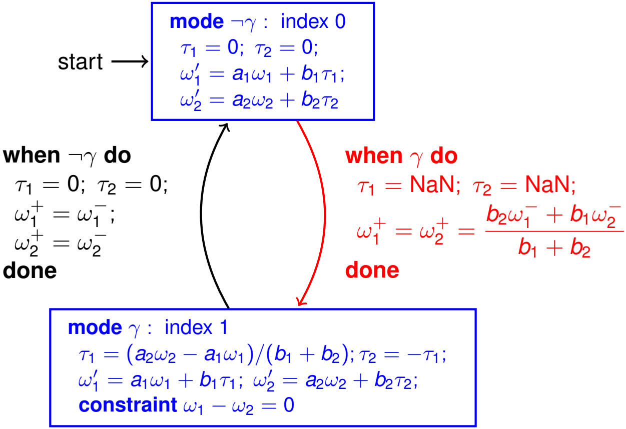
Now we can safely set $\partial \leftarrow 0$, which yields

$$\begin{array}{lrcl} & (b_1 + b_2)\omega^\bullet & = & b_2\omega_1 + b_1\omega_2 \\ \Rightarrow \text{ reset equation} & (b_1 + b_2)\omega^+ & = & b_2\omega_1^- + b_1\omega_2^- \end{array}$$

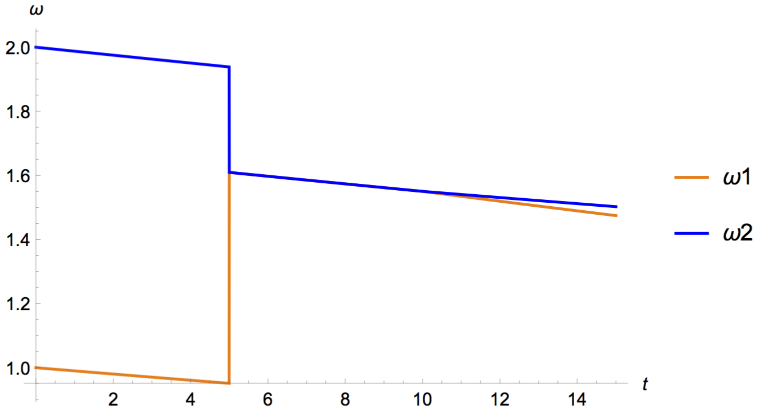
Results and code for the clutch: structural analysis



Results and code for the clutch: standardization



Results and code for the clutch: simulation results



mode changes γ : F $\rightarrow$ T $\rightarrow$ F at $t = 5, 10$

The Cup-and-Ball example

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = \left[L^2 - (x^2 + y^2)\right] \times \lambda & (\kappa_3) \end{cases}$$

$$x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2}$$

The Cup-and-Ball example

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \quad x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2} \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = \left[L^2 - (x^2 + y^2)\right] \times \lambda & (\kappa_3) \end{cases}$$

$$\begin{cases} & 0 = x'' + \lambda x & (e_1) \\ & 0 = y'' + \lambda y + g & (e_2) \\ & \gamma = [s \leq 0] & (\kappa_0) \\ \texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) \\ \texttt{and} & 0 = \lambda + s & (\kappa_2) \\ \texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) \\ \texttt{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) \end{cases}$$

Summary:

1. Causality problem: must redefine $\gamma = [s^- \leq 0]$
2. Corrected model accepted, with index $2/0$ for mode $\gamma = \mathrm{T}/\mathrm{F}$
3. New difficulty related to the kind of impact model inelastic/elastic

The Cup-and-Ball example

$$\left\{\begin{array}{lll} 0 = x'' + \lambda x & (e_1) & x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2} \\ 0 = y'' + \lambda y + g & (e_2) & \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) & \\ 0 \leq \lambda & (\kappa_2) & \\ 0 = \left[L^2 - (x^2 + y^2)\right] \times \lambda & (\kappa_3) & \end{array}\right.$$

$$\left\{\begin{array}{rll} & 0 = x'' + \lambda x & (e_1) \\ & 0 = y'' + \lambda y + g & (e_2) \\ & \gamma = [s \leq 0] & (\kappa_0) \\ \texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) \\ \texttt{and} & 0 = \lambda + s & (\kappa_2) \\ \texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) \\ \texttt{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) \end{array}\right.$$

- The mode is defined by γ, which depends on s, whose evaluation is guarded by γ: causality circuit prevents from evaluating γ; see what can be evaluated
- At initialization of instant, only $(e_1), (e_2)$ are active, with 3 dependent variables $\lambda, x^{\bullet 2}, y^{\bullet 2}$: underdetermination $\Rightarrow$ model rejected at compile time
- Solution: breaking the circuit

The Cup-and-Ball example

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \quad x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2} \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = \left[L^2 - (x^2 + y^2)\right] \times \lambda & (\kappa_3) \end{cases}$$

$$\begin{cases} & 0 = x'' + \lambda x & (e_1) \\ & 0 = y'' + \lambda y + g & (e_2) \\ & \gamma^{\bullet} = [s \leq 0]; \gamma(0) = \mathsf{F} & (\kappa_0) \\ \texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) \\ \texttt{and} & 0 = \lambda + s & (\kappa_2) \\ \texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) \\ \texttt{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) \end{cases}$$

- Note that the mode is known at the beginning of the instant
- $\gamma = \mathsf{F}$: $(e_1, e_2, \kappa_3, \kappa_4)$ is an ODE system; easy
- $\gamma = \mathsf{T}$: $(e_1, e_2, \kappa_1, \kappa_2)$ is a DAE; (κ_1) must be shifted twice (index 2)

The Cup-and-Ball example

$$\begin{cases} 0 = x'' + \lambda x & (e_1) \quad x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2} \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = \left[L^2 - (x^2 + y^2)\right] \times \lambda & (\kappa_3) \end{cases}$$

$$\begin{cases} & 0 = x'' + \lambda x & (e_1) \\ & 0 = y'' + \lambda y + g & (e_2) \\ & \gamma^{\bullet} = [s \leq 0]; \gamma(0) = \mathrm{F} & (\kappa_0) \\ \texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) \\ \text{and} & 0 = L^2 - (x^2 + y^2)^{\bullet} & (\kappa_1^{\bullet}) \\ \text{and} & 0 = L^2 - (x^2 + y^2)^{\bullet 2} & (\kappa_1^{\bullet 2}) \\ \text{and} & 0 = \lambda + s & (\kappa_2) \\ \texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) \\ \text{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) \end{cases}$$

- $\gamma = \mathrm{F}$: $(e_1, e_2, \kappa_3, \kappa_4)$ is an ODE system; easy
- $\gamma = \mathrm{T}$: $(e_1, e_2, \kappa_1, \kappa_2)$ is a DAE; (κ_1) must be shifted twice (index 2)
  - $(e_1, e_2, \kappa_1^{\bullet 2}, \kappa_2)$ is structurally nonsingular;
  - solve conflicts possibly caused by $(\kappa_1), (\kappa_1^{\bullet})$ as for the clutch example

The Cup-and-Ball example

$$\left\{\begin{array}{ll} 0 = x'' + \lambda x & (e_1) \\ 0 = y'' + \lambda y + g & (e_2) \\ 0 \leq L^2 - (x^2 + y^2) & (\kappa_1) \\ 0 \leq \lambda & (\kappa_2) \\ 0 = \left[L^2 - (x^2+y^2)\right] \times \lambda & (\kappa_3) \end{array}\right. \qquad x'' \leftarrow \frac{x^{\bullet 2} - 2x^{\bullet} + x}{\partial^2}$$

$$\left\{\begin{array}{rll} & 0 = x'' + \lambda x & (e_1) \\ & 0 = y'' + \lambda y + g & (e_2) \\ & \gamma^{\bullet} = [s \leq 0]; \gamma(0) = \mathrm{F} & (\kappa_0) \\ \texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2+y^2) & (\kappa_1) \\ \texttt{and} & \color{red}{0 = L^2 - (x^2+y^2)^{\bullet}} & \color{red}{(\kappa_1^{\bullet})} \\ \texttt{and} & \color{red}{0 = L^2 - (x^2+y^2)^{\bullet 2}} & \color{red}{(\kappa_1^{\bullet 2})} \\ \texttt{and} & 0 = \lambda + s & (\kappa_2) \\ \texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) \\ \texttt{and} & 0 = (L^2 - (x^2+y^2)) - s & (\kappa_4) \end{array}\right.$$

- Resulting code keeps the rope straight until tension λ reaches zero: inelastic impact—yet, this was not explicitly specified!
- How can we capture elastic impact? In this case, the mode $\gamma = \mathrm{T}$ is transient (zero duration): prohibits the consideration of $(\kappa_1^{\bullet 2})$. MMMhhhhh??

The Cup-and-Ball example: transient mode γ:F→T→F

$$\left\{\begin{array}{rlllll}
 & 0 = x'' + \lambda x & (e_1) & (e_1) & (e_1) \\
 & 0 = y'' + \lambda y + g & (e_2) & (e_2) & (e_2) \\
 & \gamma^\bullet = [s \leq 0]; \gamma(0) = \mathrm{F} & (\kappa_0) & & \\
\texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) & (\kappa_1) & \\
\texttt{and} & 0 = \lambda + s & (\kappa_2) & (\kappa_2) & \\
\texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) & & (\kappa_3) \\
\texttt{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) & & (\kappa_4) \\
 & & & S(\mathrm{T}) & S(\mathrm{F})
\end{array}\right.$$

1. Replace our technique of index reduction by the consideration of

$$\text{array } \mathcal{A}_n = \left[S(\mathrm{T}), S^\bullet(\mathrm{F}), S^{\bullet 2}(\mathrm{F}), \ldots, S^{\bullet n}(\mathrm{F})\right]$$

2. Find n such that $\mathcal{A}_n$ determines leading variables $\lambda, s, x^{\bullet 2}, y^{\bullet 2}$?
3. Alas, no such n exists, since $S^\bullet(\mathrm{F}), S^{\bullet 2}(\mathrm{F}), \ldots,$ are all ODE systems

The above model is underspecified at mode $\gamma = \mathrm{T}$ in cascade $\gamma : \mathrm{F}, \mathrm{T}, \mathrm{F}, \mathrm{F} \ldots$

The Cup-and-Ball example: transient mode γ:F$\rightarrow$T$\rightarrow$F

$$\left\{\begin{array}{rlllll}
 & 0 = x'' + \lambda x & (e_1) & (e_1) & (e_1) \\
 & 0 = y'' + \lambda y + g & (e_2) & (e_2) & (e_2) \\
 & \gamma^\bullet = [s \leq 0]; \gamma(0) = \mathtt{F} & (\kappa_0) & & \\
\texttt{when } \gamma \texttt{ do} & 0 = L^2 - (x^2 + y^2) & (\kappa_1) & (\kappa_1) & \\
\texttt{and} & 0 = \lambda + s & (\kappa_2) & (\kappa_2) & \\
\texttt{when not } \gamma \texttt{ do} & 0 = \lambda & (\kappa_3) & & (\kappa_3) \\
\texttt{and} & 0 = (L^2 - (x^2 + y^2)) - s & (\kappa_4) & & (\kappa_4) \\
 & & & S(\mathtt{T}) & S(\mathtt{F})
\end{array}\right.$$

1. Replace our technique of index reduction by the consideration of

$$\text{array } \mathcal{A}_n = \left[S(\mathtt{T}), S^\bullet(\mathtt{F}), S^{\bullet 2}(\mathtt{F}), \ldots, S^{\bullet n}(\mathtt{F})\right]$$

2. Find n such that $\mathcal{A}_n$ determines leading variables $\lambda, s, x^{\bullet 2}, y^{\bullet 2}$?
3. Alas, no such n exists, since $S^\bullet(\mathtt{F}), S^{\bullet 2}(\mathtt{F}), \ldots,$ are all ODE systems

The above model is underspecified at mode $\gamma = \mathtt{T}$ in cascade $\gamma : \mathtt{F}, \mathtt{T}, \mathtt{F}, \mathtt{F} \ldots$

The Cup-and-Ball example: transient mode γ:F→T→F

Discussion

- Mode $\gamma = \texttt{T}$ having > 0 duration means inelastic impact
- We return to the user the info: if mode $\gamma = \texttt{T}$ is transient, then it is incompletely specified (restart values for the next mode $\gamma = \texttt{F}$ are missing)
- User specifies impact law on velocities $\Rightarrow$ compilation succeeds

- For transient modes, replace index reduction by the (time-varying) array $\mathcal{A}$ and otherwise reuse the same algorithm for handling conflicts between previous and current nonstandard instants

- **The information {nontransient / transient} must be specified by the user as it is physical knowledge that the compiler cannot infer**
- **We expect a practical modeling language to provide different primitives for the two cases (e.g., "`if`" for a long mode vs. "`when`" for an event)**

The Cup-and-Ball example: transient mode γ:F$\rightarrow$T$\rightarrow$F

Discussion

- Mode $\gamma = \texttt{T}$ having > 0 duration means inelastic impact
- We return to the user the info: if mode $\gamma = \texttt{T}$ is transient, then it is incompletely specified (restart values for the next mode $\gamma = \texttt{F}$ are missing)
- User specifies impact law on velocities $\Rightarrow$ compilation succeeds

- For transient modes, replace index reduction by the (time-varying) array $\mathcal{A}$ and otherwise reuse the same algorithm for handling conflicts between previous and current nonstandard instants

- The information {nontransient / transient} must be specified by the user as it is physical knowledge that the compiler cannot infer
- We expect a practical modeling language to provide different primitives for the two cases (e.g., "`if`" for a long mode vs. "`when`" for an event)

The Cup-and-Ball example: transient mode γ:F$\rightarrow$T$\rightarrow$F

Discussion

- Mode $\gamma = \texttt{T}$ having > 0 duration means inelastic impact
- We return to the user the info: if mode $\gamma = \texttt{T}$ is transient, then it is incompletely specified (restart values for the next mode $\gamma = \texttt{F}$ are missing)
- User specifies impact law on velocities $\Rightarrow$ compilation succeeds
- For transient modes, replace index reduction by the (time-varying) array $\mathcal{A}$ and otherwise reuse the same algorithm for handling conflicts between previous and current nonstandard instants
- **The information {nontransient / transient} must be specified by the user as it is physical knowledge that the compiler cannot infer**
- **We expect a practical modeling language to provide different primitives for the two cases (e.g., "`if`" for a long mode vs. "`when`" for an event)**

Structural analysis: the Σ-method [Pryce2001]

Consider DAE $F = 0$: f_j(the x_i and their derivatives) $= 0$ for $i, j = 1, \dots, n$

- $\mathcal{G}_F$ weighted bipartite graph of F
  $(f, n, x) \in \mathcal{G}_F$ iff x has differentiation degree n in function f; let $d_{fx} := n$
- Find a complete matching $\mathcal{M} \subseteq \mathcal{G}_F$ (bijection $F \mapsto X$)
  and integer offsets $(c_f)_{f \in F}$ and $(d_x)_{x \in X}$, such that

$$\begin{array}{rcll} d_x - c_f & \geq & d_{fx} & \text{with equality iff } (f, d_{fx}, x) \in \mathcal{M} \\ c_f & \geq & 0 & \end{array} \qquad (\dagger)$$

- Differentiating c_f times each f yields an algebraic system of equations F_Σ(leading/other vars) $= 0$ that is structurally nonsingular with respect to leading variables (F_Σ is equivalent to an ODE)
- John Pryce found a linear programming problem equivalent to $(\dagger)$

Structural analysis: the Σ-method [Pryce2001]

Consider DAE $F = 0$: f_j(the x_i and their derivatives) $= 0$ for $i, j = 1, \ldots, n$

- $\mathcal{G}_F$ weighted bipartite graph of F
  $(f, n, x) \in \mathcal{G}_F$ iff x has differentiation degree n in function f; let $d_{fx} := n$

- Find a complete matching $\mathcal{M} \subseteq \mathcal{G}_F$ (bijection $F \mapsto X$)
  and integer offsets $(c_f)_{f \in F}$ and $(d_x)_{x \in X}$, such that

$$\begin{array}{rcl} d_x - c_f & \geq & d_{fx} \quad \text{with equality iff } (f, d_{fx}, x) \in \mathcal{M} \\ c_f & \geq & 0 \end{array} \qquad (\dagger)$$

- Differentiating c_f times each f yields an algebraic system of equations F_Σ(leading/other vars) $= 0$ that is structurally nonsingular with respect to leading variables (F_Σ is equivalent to an ODE)

- John Pryce found a linear programming problem equivalent to $(\dagger)$

Structural analysis: the Σ-method [Pryce2001]

Consider DAE $F = 0$: f_j(the x_i and their derivatives) $= 0$ for $i, j = 1, \ldots, n$

- $\mathcal{G}_F$ weighted bipartite graph of F
  $(f, n, x) \in \mathcal{G}_F$ iff x has differentiation degree n in function f; let $d_{fx} := n$

- Find a complete matching $\mathcal{M} \subseteq \mathcal{G}_F$ (bijection $F \mapsto X$)
  and integer offsets $(c_f)_{f \in F}$ and $(d_x)_{x \in X}$, such that

$$\begin{array}{rcll} d_x - c_f & \geq & d_{fx} & \text{with equality iff } (f, d_{fx}, x) \in \mathcal{M} \\ c_f & \geq & 0 & \end{array} \qquad (\dagger)$$

- Differentiating c_f times each f yields an algebraic system of equations F_Σ(leading/other vars) $= 0$ that is structurally nonsingular with respect to leading variables (F_Σ is equivalent to an ODE)

- John Pryce found a linear programming problem equivalent to (†)

Solving conflicts between past and present

1. W dependent variables in: $F(W,X) = F^{\text{past}}(W,X) \uplus F^{\text{pres}}(W,X)$

2. Dulmage-Mendelsohn decomposition:

$$F \quad = \quad \underbrace{F_H}_{\text{underdetermined}} \uplus \underbrace{F_S}_{\text{regular}} \uplus \underbrace{F_V}_{\text{overdetermined}}$$

3. In F, replace $F_V \leftarrow F_V \setminus F^{\text{pres}}$

   Thm: F^{past} has no overdetermined part
   $\Rightarrow F$ reduced has no overdetermined part

   This solves the conflict

4. The final question is: is the so reduced F structurally regular?
   - Yes: we solve $F{=}0$
   - No: the model lacks information and we inform the designer

Solving conflicts between past and present

1. W dependent variables in: $F(W,X) = F^{\text{past}}(W,X) \uplus F^{\text{pres}}(W,X)$
2. Dulmage-Mendelsohn decomposition:

$$F \quad = \quad \underbrace{F_H}_{\text{underdetermined}} \uplus \underbrace{F_S}_{\text{regular}} \uplus \underbrace{F_V}_{\text{overdetermined}}$$

3. In F, replace $F_V \leftarrow F_V \setminus F^{\text{pres}}$

   Thm: F^{past} has no overdetermined part
   $\Rightarrow F$ reduced has no overdetermined part

   This solves the conflict

4. The final question is: is the so reduced F structurally regular?
   - Yes: we solve $F{=}0$
   - No: the model lacks information and we inform the designer

Solving conflicts between past and present

1. W dependent variables in: $F(W,X) = F^{\text{past}}(W,X) \uplus F^{\text{pres}}(W,X)$
2. Dulmage-Mendelsohn decomposition:

$$F \quad = \quad \underbrace{F_H}_{\text{underdetermined}} \uplus \underbrace{F_S}_{\text{regular}} \uplus \underbrace{F_V}_{\text{overdetermined}}$$

3. In F, replace $F_V \leftarrow F_V \setminus F^{\text{pres}}$

   Thm: F^{past} has no overdetermined part
   $\Rightarrow F$ reduced has no overdetermined part

   This solves the conflict

4. The final question is: is the so reduced F structurally regular?
   - Yes: we solve $F{=}0$
   - No: the model lacks information and we inform the designer

Standardization

- Inside continuous modes, easy:
  - target continuous time, map backward $\frac{x^\bullet - x}{\partial} \mapsto x'$
- At mode changes, more difficult:
  - We target discrete time and must get rid of ∂ in space in: $x^\bullet = x + \partial x'$
  - If there are impulsive variables, setting $\partial = 0$ yields a structurally singular system. Hence we proceed as follows:
    1. We identify impulsive variables by performing impulse analysis and then eliminate them (if possible)
    2. Having done this we can safely set $\partial = 0$ and produce the code that computes restart values for non impulsive variables (enough)
  - Alternatively, we solve the algebraic system with $\partial := \delta$ positive small; **Thm:** the non impulsive variables converge to their due restart values

Standardization

- Inside continuous modes, easy:
  - target continuous time, map backward $\frac{x^\bullet - x}{\partial} \mapsto x'$
- At mode changes, more difficult:
  - We target discrete time and must get rid of ∂ in space in: $x^\bullet = x + \partial x'$
  - If there are impulsive variables, setting $\partial = 0$ yields a structurally singular system. Hence we proceed as follows:
    1. We identify impulsive variables by performing impulse analysis and then eliminate them (if possible)
    2. Having done this we can safely set $\partial = 0$ and produce the code that computes restart values for non impulsive variables (enough)
  - Alternatively, we solve the algebraic system with $\partial := \delta$ positive small; **Thm:** the non impulsive variables converge to their due restart values

Standardization

- Inside continuous modes, easy:
  - target continuous time, map backward $\frac{x^\bullet - x}{\partial} \mapsto x'$
- At mode changes, more difficult:
  - We target discrete time and must get rid of ∂ in space in: $x^\bullet = x + \partial x'$
  - If there are impulsive variables, setting $\partial = 0$ yields a structurally singular system. Hence we proceed as follows:
    1. We identify impulsive variables by performing impulse analysis and then eliminate them (if possible)
    2. Having done this we can safely set $\partial = 0$ and produce the code that computes restart values for non impulsive variables (enough)
  - Alternatively, we solve the algebraic system with $\partial := \delta$ positive small; **Thm:** the non impulsive variables converge to their due restart values

Standardization

- Inside continuous modes, easy:
  - target continuous time, map backward $\frac{x^\bullet - x}{\partial} \mapsto x'$
- At mode changes, more difficult:
  - We target discrete time and must get rid of ∂ in space in: $x^\bullet = x + \partial x'$
  - If there are impulsive variables, setting $\partial = 0$ yields a structurally singular system. Hence we proceed as follows:
    1. We identify impulsive variables by performing impulse analysis and then eliminate them (if possible)
    2. Having done this we can safely set $\partial = 0$ and produce the code that computes restart values for non impulsive variables (enough)
  - Alternatively, we solve the algebraic system with $\partial := \delta$ positive small; **Thm:** the non impulsive variables converge to their due restart values

Escaping from standardization

Standardization at mode changes is a demanding compilation step. It requires identifying impulsive variables and eliminating them, which requires computer algebraic manipulations. This is costly beyond linear systems, and even impossible in some cases. Can we avoid it? Yes, we can!

1. Nonstandard dynamics at mode change $\gamma : \mathrm{F} \rightarrow \mathrm{T}$ for the clutch:

$$\left\{ \begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial . f_1(\omega_1, \tau_1) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial . f_2(\omega_2, \tau_2) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array} \right.$$

2. Make it standard by substituting $\partial \leftarrow \delta$ where $\delta > 0$ is standard and small. Solving for $\omega_1^\bullet, \omega_2^\bullet, \tau_1, \tau_2$ yields $\omega_1^\bullet(\delta), \omega_2^\bullet(\delta), \tau_1(\delta), \tau_2(\delta)$.

3. **Thm:** $\omega_1^\bullet(\delta), \omega_2^\bullet(\delta)$ converge to ω_1^+, ω_2^+ when $\delta \rightarrow 0$.

Hint: Rescale impulsive variables τ_1, τ_2 to improve conditioning: $\widehat{\tau}_i = \delta \times \tau_i$.

Escaping from standardization

Standardization at mode changes is a demanding compilation step. It requires identifying impulsive variables and eliminating them, which requires computer algebraic manipulations. This is costly beyond linear systems, and even impossible in some cases. Can we avoid it? Yes, we can!

1. Nonstandard dynamics at mode change $\gamma : \texttt{F} \to \texttt{T}$ for the clutch:

$$\left\{\begin{array}{ll} \omega_1^\bullet = \omega_1 + \partial . f_1(\omega_1, \tau_1) & (e_1^\partial) \\ \omega_2^\bullet = \omega_2 + \partial . f_2(\omega_2, \tau_2) & (e_2^\partial) \\ \omega_1^\bullet - \omega_2^\bullet = 0 & (e_3^\bullet) \\ \tau_1 + \tau_2 = 0 & (e_4) \end{array}\right.$$

2. Make it standard by substituting $\partial \leftarrow \delta$ where $\delta > 0$ is standard and small. Solving for $\omega_1^\bullet, \omega_2^\bullet, \tau_1, \tau_2$ yields $\omega_1^\bullet(\delta), \omega_2^\bullet(\delta), \tau_1(\delta), \tau_2(\delta)$.
3. **Thm:** $\omega_1^\bullet(\delta), \omega_2^\bullet(\delta)$ converge to ω_1^+, ω_2^+ when $\delta \to 0$.

Hint: Rescale impulsive variables τ_1, τ_2 to improve conditioning: $\widehat{\tau}_i = \delta \times \tau_i$.

What if we change nonstandard representation of x'?

We used the following nonstandard representation of the derivative:

$$x' \leftarrow \frac{x^{\bullet} - x}{\partial}$$

We could, however, use equally well

$$x' \leftarrow \frac{1}{\partial} \sum_n \alpha_n \left(x^{\bullet} - x\right)^{\bullet n} \quad \text{where } \sum_n \alpha_n = 1$$

Thm: The code we generate does not depend on the particular $(\alpha_n)_{M \leq n \leq N}$

About our notion of solution of multimode DAE

How to compare the solutions we construct with other definitions?

- Problem: there is no mathematical definition for what a solution of multimode DAE system is, in the general case
- So we are unable to qualify the solutions we construct, in the general case
- Still, we have such results for a particular (nontrivial) subclass of multimode DAE systems, possibly involving impulsive behaviors at mode changes
- This subclass was recently identified by Hilding Elmqvist and Martin Otter, we call it the semi-linear multimode DAE systems. This subclass contains in particular multibody mechanics with contact.
- For semi-linear multimode DAE systems, we prove that our approach actually implements the same schemes as the ones that were directly derived by Elmqvist-Otter.

About our notion of solution of multimode DAE

How to compare the solutions we construct with other definitions?

- Problem: there is no mathematical definition for what a solution of multimode DAE system is, in the general case
- So we are unable to qualify the solutions we construct, in the general case
- Still, we have such results for a particular (nontrivial) subclass of multimode DAE systems, possibly involving impulsive behaviors at mode changes
- This subclass was recently identified by Hilding Elmqvist and Martin Otter, we call it the semi-linear multimode DAE systems. This subclass contains in particular multibody mechanics with contact.
- For semi-linear multimode DAE systems, we prove that our approach actually implements the same schemes as the ones that were directly derived by Elmqvist-Otter.

Semi-linear multimode DAE systems

Def:[Elmqvist-Otter2017] For each mode μ, the active DAE system takes the form

$$\left\{ \begin{array}{l} 0 = A(X_s)X' + B_\mu(X) \\ 0 = C_\mu(X) \end{array} \right. , \text{ where}$$

- matrix $A(X_s)$ is mode-independent and X_s) are the smooth components of X
- $A(.), B_\mu(.), C_\mu(.)$ are smooth functions of their arguments
- The following Jacobian with respect to X is regular:

$$\left[\begin{array}{c} A(X_s) \\ C'_\mu(X) \end{array} \right]$$

Thm:[Elmqvist-Otter2017] At mode changes, the restart conditions X^+ are determined from the left-limits X^- by using the following scheme:

$$\left\{ \begin{array}{l} 0 = A(X_s^-)(X^+ - X^-) \\ 0 = C_{\mu^+}(X^+) \end{array} \right.$$

Thm: The above scheme actually coincides with our method.

Concluding remarks

We proposed a systematic approach

- for the compilation of multimode DAE system models,
- from source specification to simulation code

Our approach is physics-agnostics

- better than multi-physics?

Concluding remarks

Our structural analysis rejects some spurious models:

- overconstrained (within modes or at mode changes)
- underspecified (some missing information at restart events)
- causality cycle: var→guard→eqn→var,

  a logical/numerical fixpoint equation that we do not support
  (a new issue arising in multi-mode systems)

  alternative for some cases: nonsmooth systems solvers
  handling complementarity conditions directly (V. Acary)

Concluding remarks

We rely on nonstandard analysis

- to formalize structural analysis everywhere
  (long and transient modes, and mode changes)
- to justify final code generation — standardization was essential, even if not used in practice

Some numerical difficulties remain:

- chattering, sliding modes
- nonlinear equations at restart events (we have no close guess)

How does this compare with existing approaches? (for some subclasses of systems)

- Answer was given for the semi-linear systems ($\supseteq$ contact mechanics)

Concluding remarks

Computational efficiency? Not addressed in this talk

Two alternative approaches:

- At each instant,
  1. evaluate all guards and then
  2. perform run time mode dependent structural analysis [Trenn][Höger]
- Perform compile time mode dependent structural analysis, by not enumerating the modes

Next talk by Benoit Caillaud

nonstandard problems
require
nonstandard solutions