SunDAE: How to Schedule Multimode DAE Systems?

Albert Benveniste, Benoît Caillaud, Khalil Ghorbal, Marc Pouzet

figures/inria-en.pdf figures/ens.pdf

5 December 2016

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

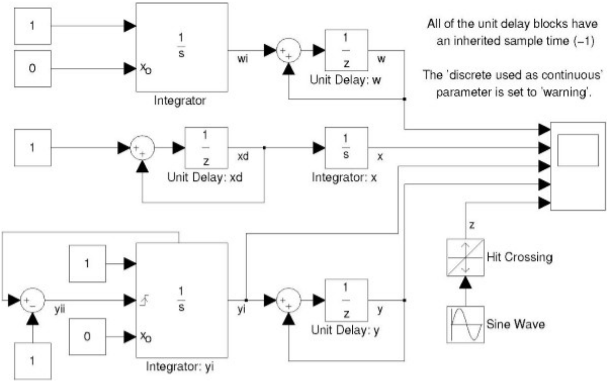
mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

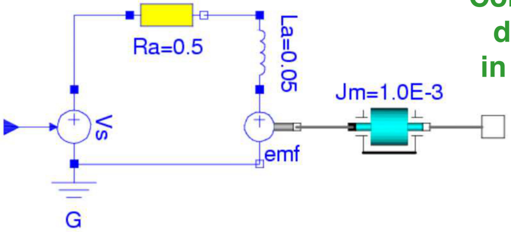
Compositionality and reuse: Simulink → Modelica

Simulink has become a central tool in systems design: Block Diagram

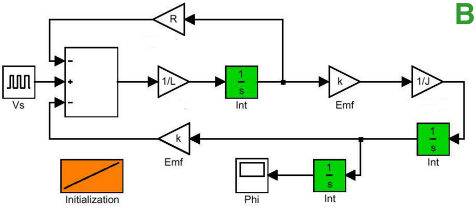


Compositionality and reuse: Simulink → Modelica

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		to Modelica (DAE): HS as physical balance equations
$\begin{cases} x' = f(x,u) \\ y = g(x,u) \end{cases}$	$\longrightarrow$	$\begin{cases} 0 = f(x',x,u) \\ 0 = g(x,u) \end{cases}$
the state space form depends on the context		Kirchhoff laws, bond graphs, multi-body mechanical systems
reuse is difficult		*reuse is much easier*

Compositionality and reuse: ODE $\rightarrow$ DAE

- Modeling tools supporting DAE
  - Proprietary languages: Mathworks/Simscape, LMS/AmeSim (bond graphs)
  - Modelica is a public standard https://www.modelica.org/ ;
  - EDA dedicated languages: VHDL AMS

A sketch of Modelica and its semantics [Fritzson]

```
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

```
1 = if g then x*x + y*y else y;
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 [Tim Bourke and Marc Pouzet]

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

Structural Analysis of DAE Systems

Aim:

- Determine the latent equations that are required to turn the DAE system into a determined system with ODEs
- Compute a scheduling of minimal blocks of equations

Two steps:

1. Index reduction: determine differentiation index and latent equations
2. Compute a scheduling: block triangular form (BTF) decomposition

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

Structural Differentiation Index

A classics: the pendulum example (T is an algebraic variable):

$$\left\{\begin{array}{lll} \ddot{x} & = & Tx \\ \ddot{y} & = & Ty - g \\ L^2 & = & x^2 + y^2 \end{array}\right. \quad \text{as a 1st order DAE:} \quad \left\{\begin{array}{l} 0 = \dot{x} - u \\ 0 = \dot{u} - Tx \\ 0 = \dot{y} - v \\ 0 = \dot{v} - Ty + g \\ 0 = -L^2 + x^2 + y^2 \end{array}\right.$$

This is not index 0 since the Jacobian with respect to $\dot{x}, \dot{u}, \dot{y}, \dot{v}$, T is singular:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -x \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -y \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Structural Differentiation Index

A classics: the pendulum example (T is an algebraic variable):

$$\left\{\begin{array}{lcl} \ddot{x} & = & Tx \\ \ddot{y} & = & Ty - g \\ L^2 & = & x^2 + y^2 \end{array}\right. \quad \text{as a 1st order DAE:} \quad \left\{\begin{array}{l} 0 = \dot{x} - u \\ 0 = \dot{u} - Tx \\ 0 = \dot{y} - v \\ 0 = \dot{v} - Ty + g \\ 0 = -L^2 + x^2 + y^2 \end{array}\right.$$

Differentiating the third equation twice yields *two latent constraints*:

$$\begin{array}{l} 0 = \dot{x} - u \\ 0 = \dot{u} - Tx \\ 0 = \dot{y} - v \\ 0 = \dot{v} - Ty + g \\ 0 = -L^2 + x^2 + y^2 \\ 0 = \dot{x}x + \dot{y}y \\ 0 = \dot{u}x + \dot{x}^2 + \dot{y}^2 + \dot{v}y \end{array}$$

Jacobians show that $\dot{x}, \dot{u}, \dot{y}, \dot{v}, T$ are uniquely determined: *the index is* 2.
Algorithms: Diff. index, consistent initialization [Pantelides 88], Σ-method (linear programming) [Pryce 01], dummy derivatives [Matsson et al. 93]

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

Decomposition into Block Triangular Form (BTF)

- Bipartite graph: incidence relation ρ between $E = \{e_1, \dots e_n\}$ and $X = \{x_1, \dots x_m\}$
- BTF = decomposition into minimal structurally invertible blocks & partial order between blocks
- Essential step in Modelica compilers
- Modelica models are structurally determined: $n = m$

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ e_1 & & X & X & X & \\ e_2 & X & X & X & & \\ e_3 & & & & X & X \\ e_4 & & & & X & X \\ e_5 & X & & & & X \end{bmatrix}$$

Decomposition into Block Triangular Form (BTF)

- BTF = decomposition into minimal structurally invertible blocks & partial order between blocks
- BTF is unique
- Classic method for sparse matrices [Duff et al. 1986]

	x_1	x_2	x_3	x_4	x_5
e_1		X	X	X	
e_2	X	X	X		
e_3				X	X
e_4				X	X
e_5	X				X

$\mapsto$

	x_4	x_5	x_1	x_2	x_3
e_3	X	X			
e_4	X	X			
e_5		X	X		
e_1	X			X	X
e_2			X	X	X

Decomposition into Block Triangular Form (BTF)

- BTF = decomposition into minimal structurally invertible blocks & partial order between blocks
- BTF is unique
- Classic method for sparse matrices [Duff et al. 1986]

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ e_1 & & X & X & X & \\ e_2 & X & X & X & & \\ e_3 & & & & X & X \\ e_4 & & & & X & X \\ e_5 & X & & & & X \end{bmatrix} \mapsto \begin{bmatrix} & x_4 & x_5 & x_1 & x_2 & x_3 \\ e_3 & X & X & & & \\ e_4 & X & X & & & \\ e_5 & & X & X & & \\ e_1 & X & & & X & X \\ e_2 & & & X & X & X \end{bmatrix}$$

Scheduling: solve e_3, e_4 for x_4, x_5; solve e_5 for x_1; solve e_1, e_2 for x_2, x_3

Reduction to Block Triangular Form (BTF)

Two steps:

1. Compute a transversal: minimal vertex cover, defining a bijection between E and X. Depth-first search algorithm [Duff, Gustavson 72–81]. Complexity $O(n|\rho|)$
2. Compute an orientation of the bipartite graph, based on the transversal. Defines a BTF decomposition (blocks are the strongly connected components) [Sargent, Westerberg 64] [Tarjan72]. Complexity $O(|\rho|)$

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 \\ e_1 & & X & X & X & \\ e_2 & X & X & X & & \\ e_3 & & & & X & X \\ e_4 & & & & X & X \\ e_5 & X & & & & X \end{bmatrix}$$

Reduction to Block Triangular Form (BTF)

- Two steps:
  1. Compute a transversal: minimal vertex cover, defining a bijection between E and X. Depth-first search algorithm [Duff, Gustavson 72–81]. Complexity $O(n|\rho|)$
  2. Compute an orientation of the bipartite graph, based on the transversal. Defines a BTF decomposition (blocks are the strongly connected components) [Sargent, Westerberg 64] [Tarjan72]

$$
\begin{bmatrix}
 & x_1 & x_2 & x_3 & x_4 & x_5 \\
e_1 & & X & X & X & \\
e_2 & X & X & X & & \\
e_3 & & & & X & X \\
e_4 & & & & X & X \\
e_5 & X & & & & X
\end{bmatrix}
\mapsto
\begin{bmatrix}
 & x_3 & x_2 & x_4 & x_5 & x_1 \\
e_1 & \color{red}{X} & X & X & & \\
e_2 & X & \color{red}{X} & & & X \\
e_3 & & & \color{red}{X} & X & \\
e_4 & & & X & \color{red}{X} & \\
e_5 & & & & X & \color{red}{X}
\end{bmatrix}
$$

Reduction to Block Triangular Form (BTF)

- Two steps:
  1. Compute a transversal: minimal vertex cover, defining a bijection between E and X. Depth-first search algorithm [Duff, Gustavson 72–81]. Complexity $O(n|\rho|)$
  2. Compute an orientation of the bipartite graph, based on the transversal. Defines a BTF decomposition (blocks are the strongly connected components) [Sargent, Westerberg 64] [Tarjan72]

$$\begin{bmatrix} & x_3 & x_2 & x_4 & x_5 & x_1 \\ e_1 & \color{red}{X} & X & X & & \\ e_2 & X & \color{red}{X} & & & X \\ e_3 & & & \color{red}{X} & X & \\ e_4 & & & X & \color{red}{X} & \\ e_5 & & & & X & \color{red}{X} \end{bmatrix} \mapsto \begin{bmatrix} & x_4 & x_5 & x_1 & x_2 & x_3 \\ e_3 & \color{blue}{X} & \color{blue}{X} & & & \\ e_4 & \color{blue}{X} & \color{blue}{X} & & & \\ e_5 & & X & \color{blue}{X} & & \\ e_1 & X & & & \color{blue}{X} & \color{blue}{X} \\ e_2 & & & X & \color{blue}{X} & \color{blue}{X} \end{bmatrix}$$

Scheduling: solve e_3, e_4 for x_4, x_5; solve e_5 for x_1; solve e_1, e_2 for x_2, x_3

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

mDAEs Example: A Simple Clutch

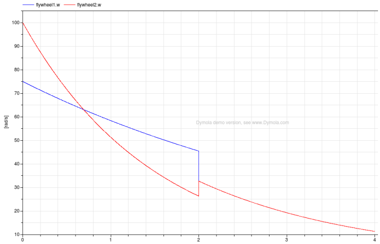
Adapting BTF Decomposition to mDAEs

Conclusion

A Simple Clutch

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

- ω_i, τ_i are the two speeds, torques
- Boolean γ is an input representing the engaged/disengaged mode



A Simple Clutch

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

- Mode `not` γ: index 0, only ODEs
- Mode γ: index 1, latent equation $\omega_1' - \omega_2' = 0$, must be entered with consistent state $\omega_1 - \omega_2 = 0$
- What happens at mode switchings?
- Albert's talk tomorrow: Structural analysis of mDAE systems

A Simple Clutch

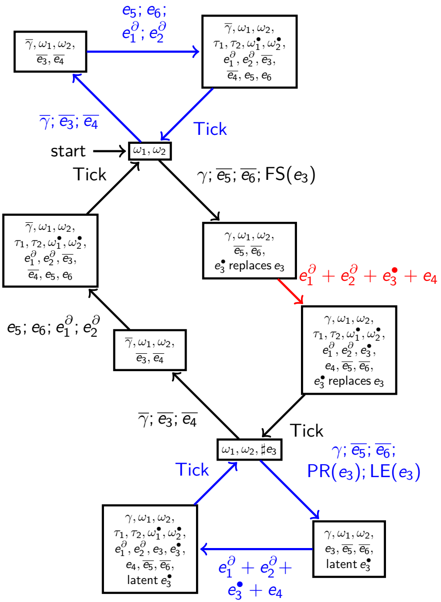
$$
\left\{
\begin{array}{rll}
 & & \omega_1^\bullet - \omega_1 = \partial \cdot f_1(\omega_1, \tau_1) \qquad (e_1^\partial) \\
 & & \omega_2^\bullet - \omega_2 = \partial \cdot f_2(\omega_2, \tau_2) \qquad (e_2^\partial) \\
\texttt{when } \gamma & \texttt{do} & \omega_1 - \omega_2 = 0 \qquad (e_3) \\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

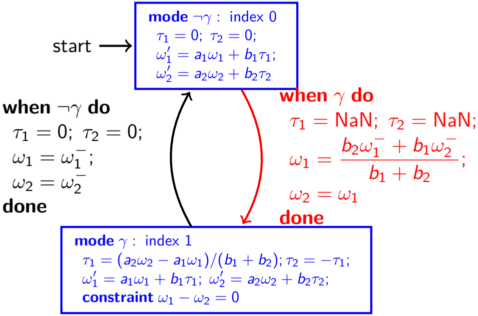
- Nonstandard time domain $\mathbb{T} = \{n\partial \mid n \in {}^\star\mathbb{N}\}$
- Transforms differential equations into *infinitesimal* difference equations:

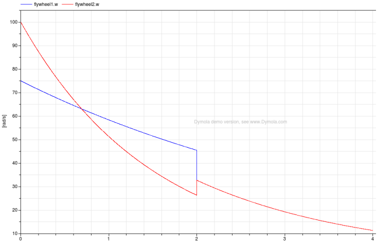
$$
x' =_{\text{def}} \frac{1}{\partial}(x^\bullet - x), \quad \text{where} \quad
\begin{array}{lll}
x^\bullet(\mathfrak{t}) & =_{\text{def}} & x(\mathfrak{t}^\bullet) \\
\text{and} \quad \mathfrak{t}^\bullet & =_{\text{def}} & \mathfrak{t} + \partial
\end{array}
$$

- Maps mDAE systems to discrete-time dynamical systems with algebraic equations

A Simple Clutch







Approach inherited from Synchronous Languages

- The structural analysis consists in searching for
  - the mode-dependent latent equations
  - a mode-dependent scheduling of blocks of equations, or block triangular form (BTF)
  - such that variables can be evaluated by solving blocks
- Adapted from the *constructive semantics* of synchronous languages [Berry1996, Benveniste et al.2003], which served as a mathematical basis for code generation.
- The structural analysis of multi-mode DAE systems we are proposing derives from the constructive semantics of synchronous languages.
- $\Longrightarrow$ **Albert's talk tomorrow** (don't miss it !)

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

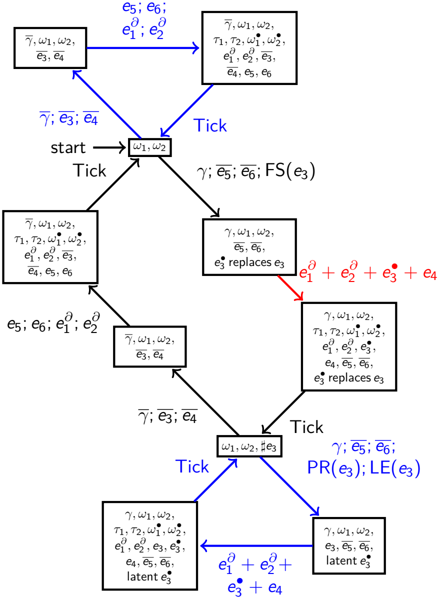
mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

Adapting BTF Decomposition to mDAEs

- Two types of dependencies: data and **control** (guarded equations)
- BTF can not be computed in one step
- SunDAE implements a variation of the transversal [Duff, Gustavson 72–81] / BTF [Tarjan72] algorithms:
  1. Transversal is updated as soon as equations are enabled / evaluated
  2. Lazy BTF decomposition: stops as soon as we have computed a minimal block



Adapting BTF Decomposition to mDAEs

- Contrarily to DAEs, mDAEs may lead to over-determined systems of equations ($n > m$, see Albert's talk).
- Transversal is not unique $\Rightarrow$ non-deterministic semantics
- Example:

$$\begin{bmatrix} & x_1 & x_2 & x_3 \\ e_1 & X & X & \\ e_2 & X & X & X \\ e_3 & & & X \\ e_4 & & X & X \end{bmatrix} \qquad \begin{bmatrix} & x_3 & x_1 & x_3 \\ e_3 & X & & \\ e_1 & X & X & X \\ e_2 & & X & X \\ e_4 & X & & X \end{bmatrix} \qquad \begin{bmatrix} & x_1 & x_2 & x_3 \\ e_1 & X & X & \\ e_2 & X & X & X \\ e_4 & & X & X \\ e_3 & & & X \end{bmatrix}$$

From DAEs to mDAES

Structural Analysis of DAE Systems

Strucutral Differentiation Index

Decomposition into Block Triangular Form (BTF)

mDAEs Example: A Simple Clutch

Adapting BTF Decomposition to mDAEs

Conclusion

Conclusion

- Constructive semantics to perform structural analysis of mDAE systems
- Inspired by: Constructive semantics of synchronous programming languages [Berry 1996]
- Main Result: mode-dependent index & causality analysis, inluding during mode switchings
- SunDAE, prototype implementation supports: Impulsive systems, varying index & dimension
- BTF decomposition : key to efficient implementation of the constructive semantics
- Transversal / BTF computed incrementally, as soon as equations become enabled
- **Open issues:** dealing with **over-determined** systems of enabled equations, **unilateral** constraints (complementarity conditions), **scalability** (state-space explosion), symbolic methods, just-in-time structural analysis [Modia], encoding state-machines into nonsmooth dynamical systems