Modelica – numerické metody

12. dubna 2013

Soustava diferenciálních a algebraických rovnic – DAE

Diferenciální rovnice

$$\bar{F}(\bar{x}, \dot{\bar{x}}, \bar{y}, t) = \bar{0}$$

Algebraické rovnice

$$\bar{G}(\bar{x}, \bar{y}, t) = \bar{0}$$

$\bar{\mathbf{x}}$ .. stavové proměnné – jsou derivované

- stejný počet stavových proměnných jako diferenciálních rovnic

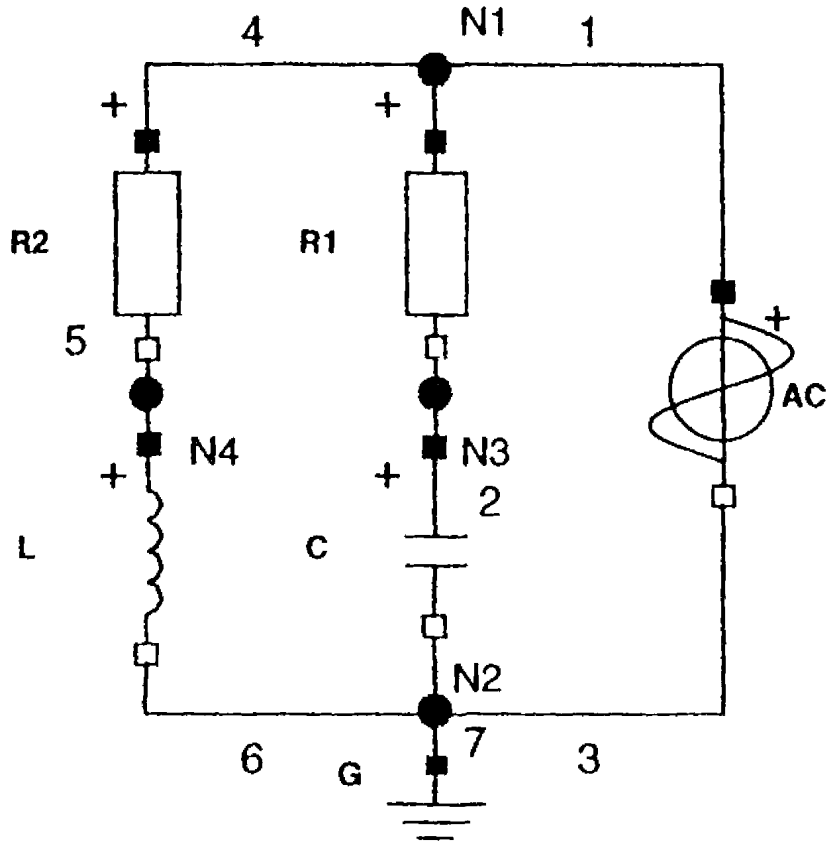
$\dot{\bar{\mathbf{x}}}$ .. derivace stavových proměnných

$\bar{\mathbf{y}}$ .. algebraické proměnné

- stejný počet algebraických proměnných jako algebraických rovnic

Příklad – schéma

RLC obvod:



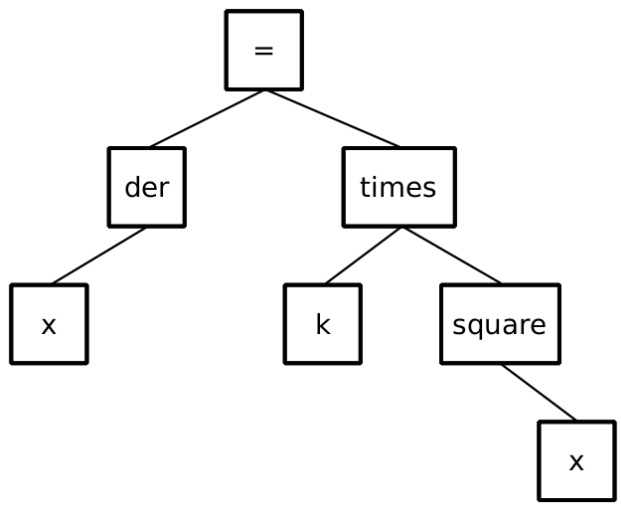
Příklad – rovnice

$$
\begin{aligned}
der(L.i) * L.L - L.v &= 0 \\
der(C.v) * C.C - C.i &= 0 \\
C.v - R1.p.v + R1.v &= 0 \\
C.i - R1.v / R1.R &= 0 \\
u(time) - R1.p.v &= 0 \\
R1.p.v - R2.v - L.v &= 0 \\
R2.v - R2.R * L.i &= 0
\end{aligned}
$$

- parametry, vstupy
- stavy
- algebraické
- derivace

Překlad modelu

1. Flatenizace modelu – hierarchie objektů nahrazena jedinou soustavou rovnic (viz. předchozí příklad)
2. Vytvoření AST (abstract syntax tree) modelu (parsování textových rovnic)
3. Analyzer – algebraické manipulace – zjednodušení rovnic, BLT transformace
4. Generování kódu v C (C#), slinkování se solverem $\rightarrow$ simulace



$$\dot{x} = kx^2$$

BLT transformace – incidenční matice

	$der(L.i)$	$der(C.v)$	$L.v$	$C.i$	$R1.p.v$	$R1.v$	$R2.v$
$der(L.i) * L.L - L.v = 0$	**1**	0	**1**	0	0	0	0
$der(C.v) * C.C - C.i = 0$	0	**1**	0	**1**	0	0	0
$C.v - R1.p.v + R1.v = 0$	0	0	0	0	**1**	**1**	0
$C.i - R1.v/R1.R = 0$	0	0	0	**1**	0	**1**	0
$u(time) - R1.p.v = 0$	0	0	0	0	**1**	0	0
$R1.p.v - R2.v - L.v = 0$	0	0	**1**	0	**1**	0	**1**
$R2.v - R2.R * L.i = 0$	0	0	0	0	0	0	**1**

- Incidenční matice - jednička, pokud rovnice obsahuje proměnnou
- Známe: parametry, vstupy, stavy
- Potřebuji spočítat: algebraické (abych mohl spočítat derivace), derivace (abych mohl udělat časový krok - spočítat změnu stavů)
- Prohazováním řádků a sloupců se snažíme převést na dolní trojúhelníkovou - BLT (Block Lower Triangular) transformace

Transformovaná incidenční matice – vnáší kauzalitu

	$R2.v$	$R1.p.v$	$L.v$	$R1.v$	$C.i$	$der(L.i)$	$der(C.v)$
$R2.v - R2.R * L.i = 0$	**1**	0	0	0	0	0	0
$u(time) - R1.p.v = 0$	0	**1**	0	0	0	0	0
$R1.p.v - R2.v - L.v = 0$	**1**	**1**	**1**	0	0	0	0
$C.v - R1.p.v + R1.v = 0$	0	**1**	0	**1**	0	0	0
$C.i - R1.v/R1.R = 0$	0	0	0	**1**	**1**	0	0
$der(L.i) * L.L - L.v = 0$	0	0	**1**	0	0	**1**	0
$der(C.v) * C.C - C.i = 0$	0	0	0	0	**1**	0	**1**

- Vzniká kauzalita:
  - Rovnice vyhodnocujeme odshora dolů
  - Proměnné vlevo od diagonály známe z vyhodnocení předchozích rovnic - vstupy
  - Proměnná na diagonále je z rovnice určena - výstup

Strong-componenty

Převod na LT není vždy možný, proto BLT – např. (jen algebraický systém)

$$
\begin{array}{rl|cccc}
 & & a & b & c & d \\
\hline
b+c+d & = -1 & 0 & 1 & 1 & 1 \\
b-c & = 2 & 0 & 1 & 1 & 0 \\
d & = 1 & 0 & 0 & 0 & 1 \\
a+ \quad d & = -3 & 1 & 0 & 0 & 1
\end{array}
$$

převedu na

$$
\begin{array}{rl|cccc}
 & & \mathrm{d} & \mathrm{b} & \mathrm{c} & \mathrm{a} \\
\hline
d & = 1 & 1 & 0 & 0 & 0 \\
d+c+b & = -1 & 1 & \mathbf{1} & \mathbf{1} & 0 \\
-c+b & = 2 & 1 & \mathbf{1} & \mathbf{1} & 0 \\
d \quad +a & = -3 & 1 & 0 & 0 & 1
\end{array}
$$

2. a 3. rovnice – systém rovnic – není možný rozřešit prohazováním řádků a sloupců - *strong-componenta* - řeší se numericky v každé iteraci výpočtu. BLT transformace - Tarjánův algoritmus z teorie grafů. Snaha vytvořit co nejmenší strongcomponenty.

Numerický výpočet (Eulerova metoda)

$$\bar{x} \rightarrow \boxed{\text{vyhodnocení rovnic, řešení strong-component}} \rightarrow \dot{\bar{x}}$$

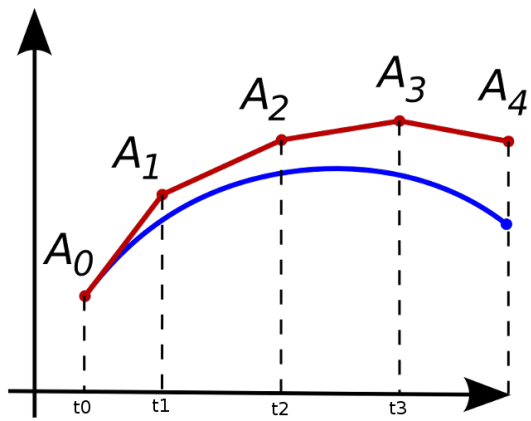
$$\dot{\bar{x}} = \bar{f}(\bar{x}, t)$$

Diskretizace proměnných a rovnice (náhrada derivace diferencí)

$$\frac{x_{n+1} - x_n}{\Delta t} = f(x_n, t_n)$$

Počítáme iterativně

$$x_{n+1} = x_n + \Delta t \cdot f(x_n, t_n)$$



Hybridní systémy - podmíněné výrazy a rovnice

Volný pád ve vzduchu a vodě. Vodní hladina $x = 0$:

$$
\begin{aligned}
inAir &= x > 0 \\
f &= -gm - (\text{if } inAir \text{ then } k_1 v \text{ else } k_2 |v| v) \\
\dot{v} &= f/m \\
\dot{x} &= v
\end{aligned}
$$

- Zero-crossing funkce $g(t) = x$
- Continuous time
  - řeší se diferenciální a algebraické rovnice a postupuje se v čase
  - nenastávají události, jsme jen v jedné větvi podmínky
  - sleduje se, jestli g nemění znaménko, událost $\rightarrow$přepnutí na
- discrete time
  - zpracovává se událost
    - ★ přesná detekce času události
    - ★ řeší se dohromady spojité i diskrétní rovnice (událost může vyvolat další, hledají se hodnoty diskrétních proměnných vyhovující rovnicím)
  - čase se nemění

Výpočet DAE systému – rekapitulace

- Při překladu – tzn. jen jednou
  - Algebraické úpravy rovnic a BLT transformace
- V každé iteraci výpočtu
  - vyhodnocování rovnic – $\bar{x} \rightarrow \dot{\bar{x}}$ – numerické řešení strong-component
    - ⋆ strong-componenty mohou být nelineární, veliké (pro nelineární $N \gtrsim 10$) $\Rightarrow$**může být časově velice náročné** $\Rightarrow$u velkých modelů **potřeba strongcomponenty minimalizovat**
  - výpočet nových stavů
  - detekce zero-crossingů
  - případně zpracování události

Inicializace

Před simulací probíhá inicializace - přiřazuje se hodnota všem proměnným (stavovým, derivacím, algebraickým, parametrům).
atribut `fixed`

- `true` - hodnota proměnné je neměnná
- `false` - proměnná vystupuje v inicializaci jako neznámá
- defaultně: parametry - `true`, stavy, algebraické (,derivace) - `false`

atribut `start`

- `fixed=true` - pevná počáteční hodnota
- `fixed=false` - počáteční odhad

sekce `initialEquation`

- rovnice, které mají být při inicializaci navíc splněny společně s rovnicemi modelu

Inicializace 2

Řeší se dohromady rovnice modelu a iniciální rovnice

$$F(x, \dot{x}, y, t) = 0$$
$$G(x, y, t) = 0$$
$$I(x, \dot{x}, y, t) = 0$$

Počet všech rovnic by měl odpovídat počtu „nonFixed" proměnných(+derivací). Pokud rovnic méně, nastaví se některé proměnné na fixed.
Pro všechny rovnice jsou vygenerovány residuální funkce

$$r(x, \dot{x}, y, t) = \begin{pmatrix} F(x, \dot{x}, y, t) \\ G(x, y, t) \\ I(x, \dot{x}, y, t) \end{pmatrix}$$

minimalizace čtverců residuálních rovnic - Simplexová metoda.
Pokud chci zadat počáteční hodnotu, je lepší použít `start = ...`, `fixed = true` **než používa pro přiřazení** `initial equation`.

Inicializace - příklad

```
model InitExample
  Real x(start = 1, fixed = true);
  Real y(start = 2); //fixed = false by default
initial equation
  der(y) = 0;
equation
  der(x) = x + y;
  der(y) = x - y);
end InitExample
```

Reziduální funkce

$$r_1 = \dot{y}$$
$$r_2 = \dot{x} - x - y$$
$$r_3 = \dot{y} - x + y$$

x je zadána napevno, hledáme $y, \dot{x}, \dot{y}$. Stejně rovnic jako neznámých.

$$\min_{(y,\dot{x},\dot{y})}(\dot{y}^2 + (\dot{x} - x - y)^2 + (\dot{y} - x + y)^2)$$

Materiály

- Specifikace Modeliky - www.modelica.org/documents
- Principles of Object-oriented modeling and simulation with Modelica (Peter Fritzson)
- články
  - Event Handling in the OpenModelica Compiler and Runtime System (Håkan Lundvall, Peter Fritzson, Bernhard Bachmann)
  - Dynamic Selection of States in Dymola (Sven Erik Mattsson, Hans Olsson and Hilding Elmqvist)
- Přeložené modely, zdrojový kód OpenModeliky