Improvements of the Initialization Method of DAEs in OpenModelica

Lennart A. Ochel  Bernhard Bachmann  Willi Braun
FH Bielefeld - University of Applied Sciences
Outline

⇒ Mathematical Formulation of Hybrid DAEs
⇒ Symbolic Transformation Steps
⇒ Initialization in Modelica (Conventional)
⇒ Initialization in OpenModelica
Mathematical Formalism

general representation of hybrid DAEs:

\[ 0 = f(t, \dot{x}, x, y, u, q, q_{pre}, c, p) \]

<table>
<thead>
<tr>
<th>( t )</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \dot{x}(t) )</td>
<td>vector of differentiated state variables</td>
</tr>
<tr>
<td>( x(t) )</td>
<td>vector of state variables</td>
</tr>
<tr>
<td>( y(t) )</td>
<td>Vector of algebraic variables</td>
</tr>
<tr>
<td>( u(t) )</td>
<td>vector of input variables</td>
</tr>
<tr>
<td>( q(t_e); q_{pre}(t_e) )</td>
<td>vectors of discrete variables</td>
</tr>
<tr>
<td>( c(t_e) )</td>
<td>vector of condition expressions</td>
</tr>
<tr>
<td>( p )</td>
<td>vector of parameters/ constants</td>
</tr>
</tbody>
</table>
Principles of Numerical Integration Methods
(Example: Explicit Euler Method)

Integration of explicit ordinary differential equations (ODEs):

\[ \dot{x}(t) = f \left( t, x(t), u(t), p \right), \quad x(t_0) = x_0 \]

Numerical approximation of the derivative and/or right-hand-side:

\[ \dot{x}(t_n) \approx \frac{x(t_{n+1}) - x(t_n)}{t_{n+1} - t_n} \approx f \left( t_n, x(t_n), u(t_n), p \right) \]

Iteration scheme:

\[ x(t_{n+1}) \approx x(t_n) + \left( t_{n+1} - t_n \right) \cdot f \left( t_n, x(t_n), u(t_n), p \right) \]

Calculating an approximation of \( x(t_{n+1}) \) based on the values of \( x(t_n) \)

Here:
Explicit Euler integration method

Convergence?
Symbolic Transformation Steps

Transform to explicit state-space representation:

\[ 0 = f(t, \dot{x}, x, y, u, q, q_{\text{pre}}, c, p) \quad \rightarrow \quad 0 = f(t, z, x, u, q, q_{\text{pre}}, c, p) \quad z = \begin{pmatrix} \dot{x} \\ y \end{pmatrix} \]

\[ z = \begin{pmatrix} \dot{x} \\ y \end{pmatrix} = g(t, x, u, q, q_{\text{pre}}, c, p) \quad \rightarrow \quad \dot{x} = h(t, x, u, q, q_{\text{pre}}, c, p) \quad y = k(t, x, u, q, q_{\text{pre}}, c, p) \]

Implicit function theorem:

• Necessary condition for the existence of the transformation is that the following matrix is regular at the point of interest:

\[ \det \left( \frac{\partial}{\partial z} f(t, z, x, u, q, q_{\text{pre}}, c, p) \right) \neq 0 \]
Initialization in Modelica

Initialization of “free” state variables
• same number of “free” states and additional equations

\[ 0 = f(t, \dot{x}, x, y, u, q, q_{pre}, c, p) \]

Initialization of “free” parameters
• same number of “free” parameters and additional equations

\[ z = \begin{pmatrix} \dot{x} \\ y \end{pmatrix} = g(t, x, u, q, q_{pre}, c, p) \]

Initialization mechanism in Modelica
• initial equation/algorithm sections
• attribute fixed, start and nominal

\[ \dot{x} = h(t, x, u, q, q_{pre}, c, p) \]
\[ y = k(t, x, u, q, q_{pre}, c, p) \]
Initialization in OpenModelica

Nonlinear system of equations

- $m$, number of equations
- $n$, number of variables
- $m \geq n$, over-determined
- $m \leq n$, under-determined

Corresponding minimization problem

- solution solves the nonlinear system of equations

Derivative-free method in OpenModelica

- based on simplex method of Nelder and Mead

$$
\begin{align*}
G_1(z_1, \ldots, z_n) &= 0 \\
& \vdots \\
G_m(z_1, \ldots, z_n) &= 0 \\
\sum_{i=1}^{m} G_i(z_1, \ldots, z_n)^2 &\rightarrow \min \\
0 &= f(t, \dot{x}, x, y, u, q, q_{pre}, c, p)
\end{align*}
$$
Initialization in OpenModelica

- **start**
- Setup with start-values
- Zerocrossings
- Initialized?
- Optimization
- Event iteration
- End
Full Support of Start-Values

**initial equation/algorithm**

equations are valid at the end of the initialization

\[
\begin{align*}
G_1(z_1, \ldots, z_n) &= 0 \\
\vdots \\
G_m(z_1, \ldots, z_n) &= 0
\end{align*}
\]

**attribute: start**

guesses are valid at the beginning of the initialization

\[
z_i - \text{start}(z_i) = 0
\]

**resulting optimization:**

\[
\begin{align*}
\min \left\{ \lambda \sum_{i=1}^{m} G_i^2 + (1 - \lambda) \sum_{i} (z_i - \text{start}(z_i))^2 \right\} \\
\text{s.t.: } 0 = f(t, \dot{x}, x, y, u, q, q_{\text{pre}}, c, p)
\end{align*}
\]
Full Support of Start-Values

```model StartValue
    Real x;
    Real y(start=-3);

    initial equation
        x^2 = 10;

    equation
        der(x) = time;
        y = x;

    end StartValue;
```

<table>
<thead>
<tr>
<th>var</th>
<th>expected solution</th>
<th>solution 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>$-\sqrt{10}$</td>
<td>$\sqrt{10}$</td>
</tr>
<tr>
<td>y</td>
<td>$-\sqrt{10}$</td>
<td>$\sqrt{10}$</td>
</tr>
</tbody>
</table>

min\(\lambda \sum_{i=1}^{m} G_i^2 + (1 - \lambda) \sum_{i} (z_i - \text{start}(z_i))^2\)

s.t.: \(0 = f(t, \dot{x}, x, y, u, q, q_{\text{pre}}, c, p)\)
Under-Determined Initialization

Fewer equations than unfixed states/parameters:

\[ \Rightarrow \text{Mechanism to auto-fix free states/parameters} \]

1. **Symbolic analysis of dependencies**
   \[ \Rightarrow \text{future work} \]

2. **Numerical analysis of dependencies**
   \[ \Rightarrow \text{implemented} \]
Scaling of Initial Residuals

Initialization ⇔ \( 0 = \min \{ F(z) \} \)

s.t.: \( 0 = f(t, \dot{x}, x, y, u, q, q_{pre}, c, p) \)

\[
F(z) = \lambda \sum_{i=1}^{m} G_i^2 + (1 - \lambda) \sum_i (z_i - \text{start}(z_i))^2
\]
Scaling of Initial Residuals

Example: \[ F(z_1, z_2) = \left( \frac{z_1 - 10^{-6}}{G_1} \right)^2 + \left( \frac{z_2 - 10^6}{G_2} \right)^2 \]

- \[ F(10^{-6}, 10^6) = 0 \]
- \[ F(1.1 \cdot 10^{-6}, 10^6) = 10^{-14} \] 10% deviation in \( z_1 \)
- \[ F(10^{-6}, 1.1 \cdot 10^6) = 10^{10} \] 10% deviation in \( z_2 \)
Scaling of Initial Residuals

Example: \[ F(z_1, z_2) = (z_1 - 10^{-6})^2 + (z_2 - 10^6)^2 \]

- \( F(10^{-6}, 10^6) = 0 \)
- \( F(1.1 \cdot 10^{-6}, 10^6) = 10^{-14} \) 10% deviation in \( z_1 \)
- \( F(10^{-6}, 1.1 \cdot 10^6) = 10^{10} \) 10% deviation in \( z_2 \)

With Scaling: \[ \tilde{F}(z) = \lambda \sum_{i=1}^{m} K_i^{-1} \cdot G_i^2 + (1 - \lambda) \sum_i K_i^{-1} \cdot (z_i - \text{start}(z_i))^2 \]

- \( \tilde{F}(10^{-6}, 10^6) = 0 \)
- \( \tilde{F}(1.1 \cdot 10^{-6}, 10^6) = 0.1 \) 10% deviation in \( z_1 \)
- \( \tilde{F}(10^{-6}, 1.1 \cdot 10^6) = 0.1 \) 10% deviation in \( z_2 \)
Scaling of Initial Residuals

Example: \[ \tilde{F}(z_1, z_2) = K_1^{-1} \cdot (z_1 - 10^{-6})^2 + K_2^{-1} \cdot (z_2 - 10^6)^2 \]

How to choose \( K_i^{-1} \)? (linear dependencies of one variable)
  - \( K_1 = \text{nominal}(z_1) \)
  - \( K_1 = \text{nominal}(z_2) \)

How to handle in general?
  - nonlinear equations
  - multiple dependencies
Scaling of Initial Residuals

How to choose scaling coefficients in general?

\[ \tilde{F}(z) = \lambda \sum_{i=1}^{m} K_i^{-1} \cdot G_i^2 + (1 - \lambda) \sum_{i} K_i^{-1} \cdot (z_i - \text{start}(z_i))^2 \]

\[ K_i = \max_j \left\{ \text{nominal}(x_j) \cdot \left| \frac{\partial G_i(\text{nominal}(x))}{\partial x_j} \right|, K_{min} \right\} \]

(derived from differential error analysis)

\[ K_{min}, \text{heuristically treatment for small } K_i \]
Conclusions

- Reliable initialization of standard models (OMC test suite)
- Initialization of consistent over-determined systems
- Initialization of under-determined systems
- Full support of start values for all variables
  - see Modelica specification
- Numerical improvements by robust scaling techniques
- First tests with real-world-problems:
  - First tests with models from Siemens Power library successfull
Future Work

- **Efficiency improvements**
  - Implementation of more advanced optimization algorithms
  - Involve boundary conditions (min/max-values)
  - Symbolic preprocessing of initialization problem

- **Initialization of under-determined systems**
  - Symbolic analysis of dependencies between states and initial equations

- **Real-World-Problems:**
  - More advanced tests with models from Siemens Power library
  - Full support of Modelica Standard Library (OMC functionality)