DAE models and differential index

Lecture 1 – Simulation of differential-algebraic equations

Erik Frisk (erik.frisk@liu.se)
Department of Electrical Engineering
Linköping University

LINKÖPING UNIVERSITY

Is and is not

What this part of the course is (hopefully)

- Understand what a DAE is, characteristics, and structure
- Understand why they are useful
- Understand why they are (sometimes) more difficult to simulate than an ODE
- Understand the origins of the difficulties and how to detect them
- Know how and when one can expect your favorite solver for ODEs to work well also for DAEs
- How to simulate models described in object orients languages, like Modelica

What this part is not

- detailed derivations and analysis of specific methods for simulation of DAEs

Outline of the DAE module, lectures

1. Basic properties
   - principles
   - differences between ODEs and DAEs
   - differential index
2. Simulation methods
   - principal problems with high index problems
   - simulation of low-index problems
   - index reduction techniques
3. Adjoint sensitivity analysis, numerical code, and Modelica, simulation of object-oriented models
4. Simulation of object oriented models continued
   - Simulation of Modelica models, structural analysis
   - index reduction using dummy-derivatives

Introduction to differential-algebraic models

ODE and DAE

Ordinary Differential Equations - ODE

A system of ordinary differential equations

$$\frac{d}{dt}x(t) = f(t, x(t)), \quad x(0) = x_0$$

where $x(t) \in \mathbb{R}^n$ and $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$.

DAE – another, mathematically and numerically, convenient representation of a dynamical system

Differential-Algebraic Equations - DAE

A general DAE formulation

$$F(\frac{d}{dt}x(t), x(t), t) = 0, \quad x(0) = x_0, \ \dot{x}(0) = \dot{x}_0$$

where $x(t) \in \mathbb{R}^n$ and $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$.

Algebraic vs dynamic vs. state variables

In an ODE

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

the state is x but for a DAE

$$F(\dot{x}(t), x(t), t) = 0, \quad x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0$$

x is not exactly the state. It includes the state, but there are typically more variables than state-variables.

For that reason, it is sometimes beneficial to write a DAE as

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

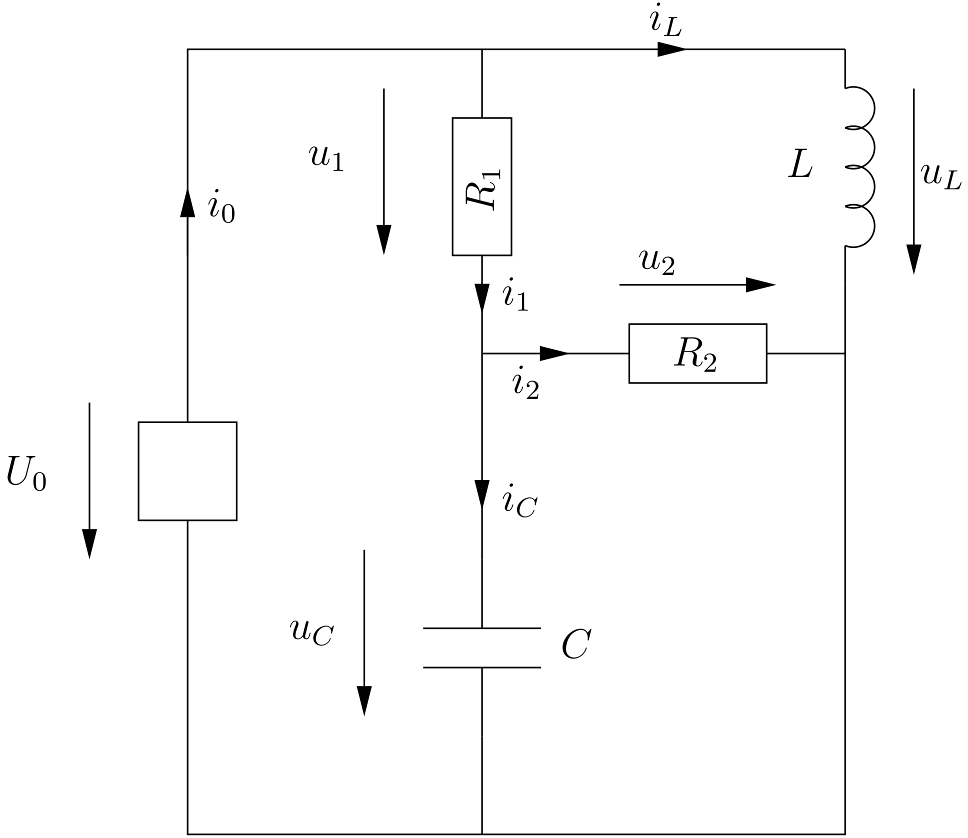
where $x(t)$ are the dynamic variables and $y(t)$ the algebraic variables.

Again: Note that $x(t)$ here is **not** the state (although it includes the state); more later.

Why DAE?

- Object oriented modelling
- Basic physics
- structure and numerics
- Invariants
- Simplification of an ODE, e.g., assume a physical connection is stiff instead of flexible. Can result in a DAE that is significantly simpler to solve than the original ODE
- Singular perturbation problems (SPP)
- Inverse problems, given $y(t)$, simulate corresponding u
- Many names: singular, implicit, descriptor, generalized state-space, non-causal, semi-state, …

A simple electrical circuit



The ODE is direct to derive (do it!)

$$\frac{d}{dt}u_{C(t)} = -\frac{R_1 + R_2}{CR_1R_2}u_C(t) + \frac{1}{CR_1}f(t)$$

$$\frac{d}{dt}i_L = \frac{1}{L}f(t)$$

- Some loss of structure; for a small example it doesn't matter
- For larger models this matters; analysis and simulation
- Wouldn't it be great if we could simulate models in the original formulation directly?

Modelica model of the circuit

```
model Circuit
  import Modelica.Electrical.Analog.Basic.*;
  import Modelica.Electrical.Analog.Sources.*;
  Resistor R1;
  Resistor R2;
  Capacitor C;
  Inductor L;
  Ground G;
  SineVoltage src;
equation
  connect(G.p, src.n);
  connect(src.p, R1.p);
  connect(src.p, L.p);
  connect(R1.n, R2.p);
  connect(R1.n,C.p);
  connect(L.n, R2.n);
  connect(L.n, C.n);
  connect(C.n, G.p);
end Circuit;
```

Equations generated from the Modelica model (33 eqs.)

```
R1.R * R1.i = R1.v;
R1.v = R1.p.v - R1.n.v;
0.0 = R1.p.i + R1.n.i;
R1.i = R1.p.i;
R2.R * R2.i = R2.v;
R2.v = R2.p.v - R2.n.v;
0.0 = R2.p.i + R2.n.i;
R2.i = R2.p.i;
C.i = C.C * der(C.v);
C.v = C.p.v - C.n.v;
0.0 = C.p.i + C.n.i;
C.i = C.p.i;
L.L * der(L.i) = L.v;
L.v = L.p.v - L.n.v;
0.0 = L.p.i + L.n.i;
L.i = L.p.i;
G.p.v = 0.0;
```

```
src.signalSource.y = sin();
src.v = src.signalSource.y;
src.v = src.p.v - src.n.v;
0.0 = src.p.i + src.n.i;
src.i = src.p.i;
L.n.i + R2.n.i + C.n.i + G.p.i
+ src.n.i = 0.0;
L.n.v = R2.n.v;
R2.n.v = C.n.v;
C.n.v = G.p.v;
G.p.v = src.n.v;
R1.n.i + R2.p.i + C.p.i = 0.0;
R1.n.v = R2.p.v;
R2.p.v = C.p.v;
src.p.i + R1.p.i + L.p.i = 0.0;
src.p.v = R1.p.v;
R1.p.v = L.p.v;
```

Differential-algebraic models

A general DAE in the form

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

is kind of similar to an ODE

$$\dot{y} = f(y, t)$$

How big difference could there be?

Why not apply, e.g., an Euler-forward or Euler-backward

$$F\left(\frac{y_t - y_{t-h}}{h}, y_{t-h}, t-h\right) = 0, \qquad F\left(\frac{y_t - y_{t-h}}{h}, y_t, t\right) = 0$$

and solve for y_t? Identical procedure as for an ODE.

Unfortunately, it is not that simple! (in general) (but sometimes!)

A simple case

Assume a DAE

$$\dot{x} = f(x, y, t), \quad 0 = g(x, y, t)$$

If you can solve for y in the second equation $y = g^{-1}(x, t)$, you'll have an ODE

$$\dot{x} = f(x, g^{-1}(x, t), t)$$

Loss of structure when transforming into an ODE (e.g. the simple circuit on slide 7).

As on last slide, we can also apply Euler-backwards directly (maybe?)

$$F(y_n, (y_n - y_{n-1})/h, t_n) = 0$$

But … what happens with the mathematically well formulated model

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

$$0 = g(x, t)$$

Differential-algebraic models

A general DAE

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

is pretty similar to an ODE

$$\dot{y} = f(y, t)$$

What is the difference? When can an ODE solver work also for DAEs?

Answer: Sometimes

This first lecture deals with these differences, characteristics of DAEs and when ODE methods can be directly applied

Next time more on how to simulate DAEs and how to transform them into a form suitable for an ODE solver.

A super simple example

The DAE below can easily be transformed into an ODE

$$\begin{aligned}\dot{x}(t) &= -x(t) + y(t) \\ 0 &= x(t) + y(t) - u(t)\end{aligned}$$

but for illustration, a directly applied backward Euler gives

$$\begin{aligned}\frac{x_{t+1} - x_t}{h} &= -x_{t+1} + y_{t+1} \\ 0 &= x_{t+1} + y_{t+1} - u_{t+1}\end{aligned}$$

which can be solved analytically as

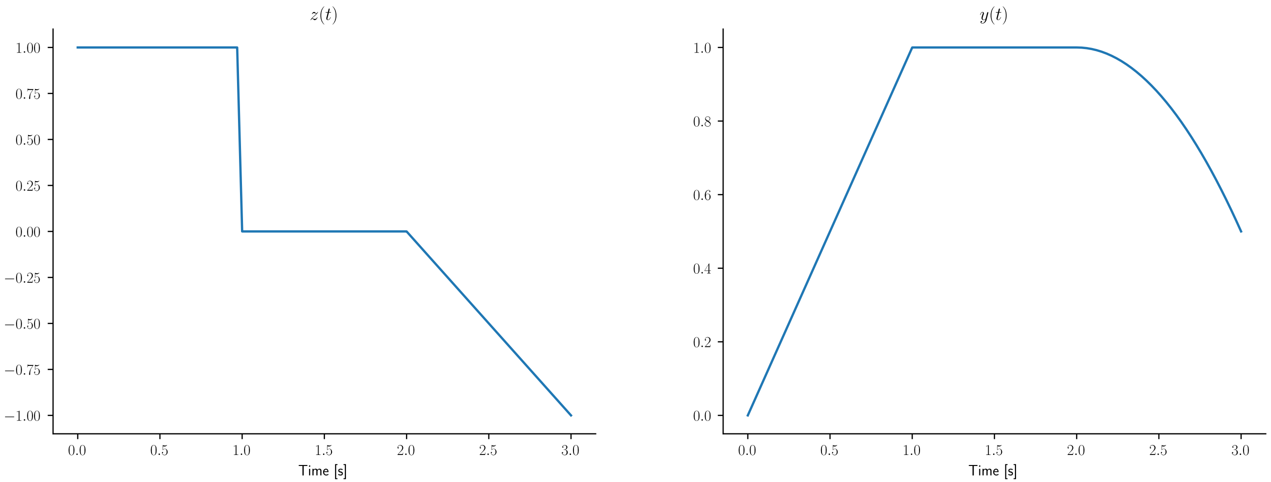
$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \frac{1}{1+2h} \begin{pmatrix} x_t + h\ u_{t+1} \\ -x_t + (1+h)\ u_{t+1} \end{pmatrix}$$

Briefly; solution to differential-algebraic equations

DAE and ODE

$$\dot{y}(t) = z(t)$$

- Integration, gives smoother solutions; differentiation gives more non-smooth solutions.
- Differentiation is “simpler” than integration analytically; numerically the other way around
- ODE - pure integration.
  DAE - mix between integration and differentiation



LINKÖPING
UNIVERSITY

Solutions

Assume a DAE

$$z_1(t) = g(t)$$
$$\dot{z}_1(t) = z_2(t)$$

It is direct that there are no continuous solutions $(z_1(t), z_2(t))$ if the function $g(t)$ has discontinuities.

For ODEs the situation is more simple

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

solutions are always smoother than forcing function.

Different DAE formulations

Implicit ODE

$$F(y, \dot{y}, t) = 0, \quad F_{y'} \text{ invertible}$$

Linear time-invariant DAE

$$E\dot{y} = Ay, \quad E \text{ singular}$$

Semi-explicit DAE

$$\dot{x} = f(x, y, t)$$
$$0 = g(x, y, t)$$

Solvability/solutions

Definitions on solvability for DAE is similar to solvability for ODEs.

Require consistency! (we will talk more about what this means)

One difference worth noting: An ODE solution is always at least once differentiable, this is not true for DAEs and all components are not as smooth.

Consider

$$\begin{aligned}\dot{y} &= x \\ y &= v(t)\end{aligned} \Leftrightarrow \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ v(t) \end{pmatrix}$$

where $v(t) \in \mathcal{C}^1$. Then y will be 1 time differentiable and x not differentiable.

Solvability

A linear and time-invariant DAE

$$A\dot{y} + By = f(t) \sim (sA + B)Y(s) = F(s)$$

is solvable if and only if $\lambda A + B$ has full rank for any $\lambda \in \mathbb{C}$ (think Laplace-transform).

It can be tricky if not LTI: The time-varying DAE

$$\begin{pmatrix} -t & t^2 \\ -1 & t \end{pmatrix} \dot{y} + y = A\,\dot{y} + y = 0,$$

is not solvable on the interval $t > 0$ even though $|\lambda A(t) + B(t)| \equiv 1$.

Something to think about at home ... figure out why.

Summary: Characterizing solvability and solutions for time-variable/non-linear DAE's can be complex, and not something we dig further into in this course.

DAE vs. stiff problems

A semi-explicit DAE

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ 0 &= f_2(x_1, x_2, t) \end{aligned}$$

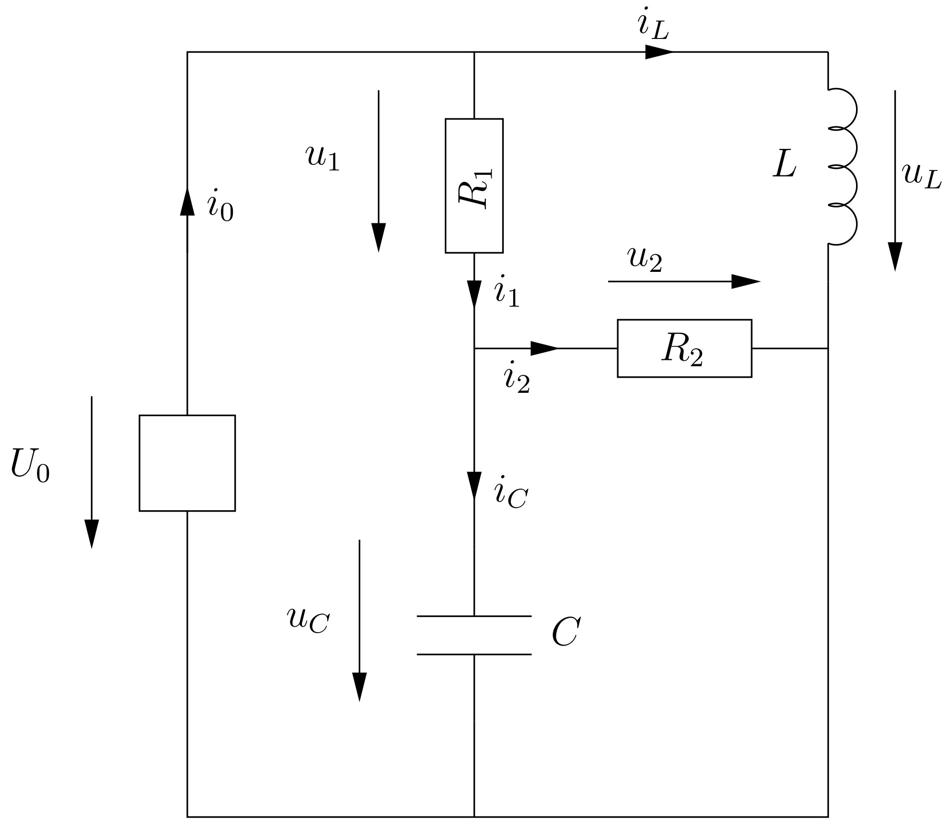
is similar to the stiff ODE (ε small)

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ \varepsilon\, \dot{x}_2 &= f_2(x_1, x_2, t) \end{aligned}$$

- similarities and differences
- when do ODE methods work for DAEs?
- In this presentation, I will for simplicity mainly illustrate using one-step Euler-backwards

Illustrative example in three acts

The simple circuit model, act 1



$$u_0 = f(t)$$
$$u_1 = R_1 i_1$$
$$u_2 = R_2 i_2$$
$$i_C = C\frac{du_c}{dt}$$
$$u_L = L\frac{di_L}{dt}$$
$$i_0 = i_1 + i_L$$
$$i_1 = i_2 + i_C$$
$$u_0 = u_1 + u_C$$
$$u_L = u_1 + u_2$$
$$u_C = u_2$$

$$x_1 = (u_C, i_L), x_2 = (u_0, u_1, u_2, u_L, i_0, i_1, i_2, i_C)$$

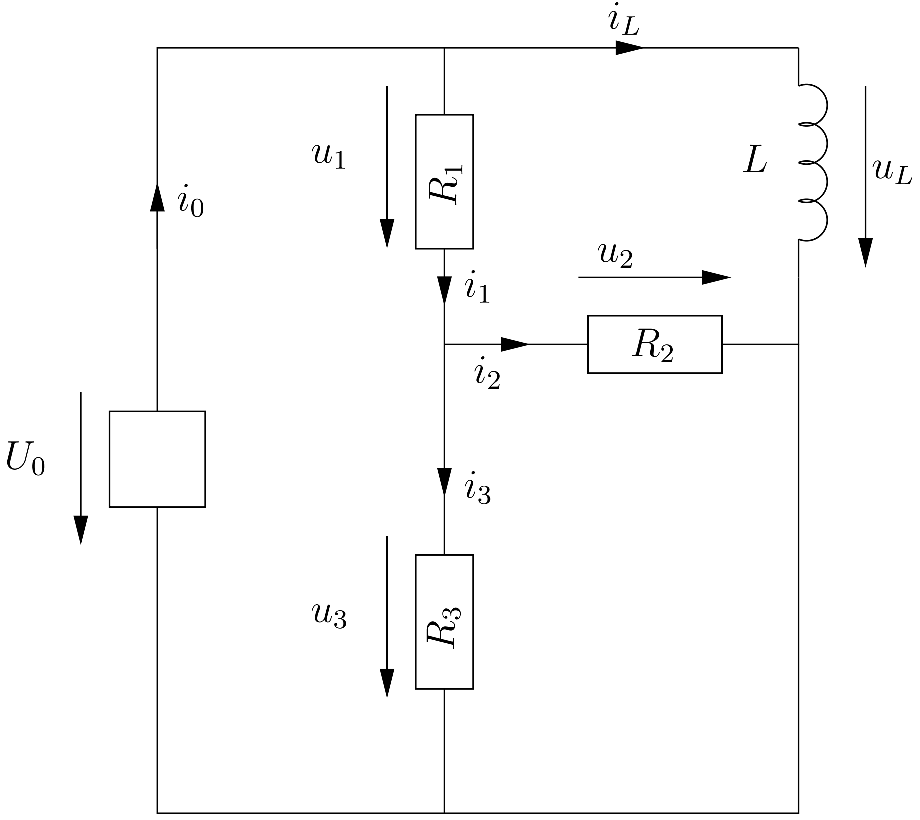
Reformulate equations into computational form

$$\begin{aligned}
e_1 &: u_0 = f(t) \\
e_2 &: u_1 = R_1 i_1 \\
e_3 &: u_2 = R_2 i_2 \\
e_4 &: i_C = C\frac{du_c}{dt} \\
e_5 &: u_L = L\frac{di_L}{dt} \\
e_6 &: i_0 = i_1 + i_L \\
e_7 &: i_1 = i_2 + i_C \\
e_8 &: u_0 = u_1 + u_C \\
e_9 &: u_L = u_1 + u_2 \\
e_{10} &: u_C = u_2
\end{aligned}$$

$$\Rightarrow$$

$$\begin{aligned}
e_{10} &: u_2 := u_C \\
e_3 &: i_2 := \frac{1}{R_2}u_2 \\
e_1 &: u_0 := f(t) \\
e_8 &: u_1 := u_0 - u_C \\
e_9 &: u_L := u_1 + u_2 \\
e_2 &: i_1 := \frac{1}{R_1}u_1 \\
e_7 &: i_C := i_1 - i_2 \\
e_6 &: i_0 := i_1 + i_L \\
e_4 &: \frac{du_c}{dt} = \frac{1}{C}i_C \\
e_5 &: \frac{di_L}{dt} = \frac{1}{L}u_L
\end{aligned}$$

The simple circuit model, act 2 ($C \to R_3$)



$$u_0 = f(t)$$
$$u_1 = R_1 i_1$$
$$u_2 = R_2 i_2$$
$$u_3 = R_3 i_3$$
$$u_L = L\frac{di_L}{dt}$$
$$i_0 = i_1 + i_L$$
$$i_1 = i_2 + i_3$$
$$u_0 = u_1 + u_3$$
$$u_L = u_1 + u_2$$
$$u_3 = u_2$$

$$x_1 = i_L, x_2 = (i_3, u_2, i_2, u_0, u_1, u_L, i_1, i_C, i_0)$$

Reformulate equations into computational form

$$\frac{di_L}{dt} = \frac{1}{L} u_L$$

$$u_0 := f(t)$$

Solve for $\{u_1, u_2, u_3, i_1, i_2, i_3\}$ (6 unknowns, 6 equations)

$$u_1 = R_1 i_1$$

$$u_2 = R_2 i_2$$

$$u_3 = R_3 i_3$$

$$i_1 = i_2 + i_3$$

$$u_0 = u_1 + u_3$$

$$u_3 = u_2$$

$$i_0 := i_1 + i_L, \quad u_L := u_1 + u_2$$

Reformulate equations into computational form

$$\frac{di_L}{dt} = \frac{1}{L} u_L$$

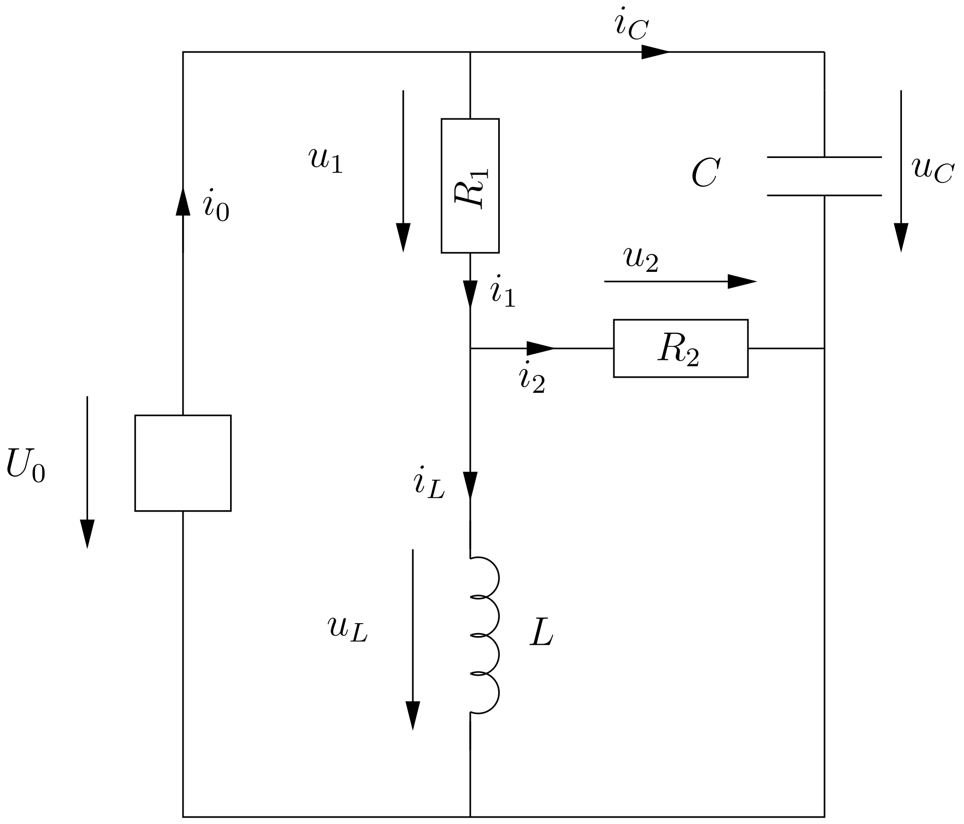
$$u_0 := f(t)$$

Solve for $\{u_1, u_2, u_3, i_1, i_2, i_3\}$ (6 unknowns, 6 equations)

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ i_1 \\ i_2 \\ i_3 \end{pmatrix} := \frac{1}{R_1 R_2 + R_1 R_3 + R_2 R_3} \begin{pmatrix} R_1(R_2 + R_3) \\ R_2 R_3 \\ R_2 R_3 \\ R_2 + R_3 \\ R_3 \\ R_2 \end{pmatrix} u_0$$

$$i_0 := i_1 + i_L, \quad u_L := u_1 + u_2$$

The simple circuit model, act 3 ($C \to L$)



$$x_1 = (u_C, i_L), x_2 = (u_2, i_2, u_0, u_1, u_L, i_1, i_C, i_0)$$

$$u_0 = f(t)$$
$$u_1 = R_1 i_1$$
$$u_2 = R_2 i_2$$
$$i_C = C\frac{du_C}{dt}$$
$$u_L = L\frac{di_L}{dt}$$
$$i_0 = i_1 + i_C$$
$$i_1 = i_2 + i_L$$
$$u_0 = u_1 + u_L$$
$$u_C = u_1 + u_2$$
$$u_L = u_2$$

Reformulate equations into computational form

It is not possible to, in the same way as before, to obtain a computational form. If you write the model in the form

$$\dot{x}_1 = g(x_1, x_2)$$
$$0 = h(x_1, x_2)$$

where $x_1 = (u_C, i_L)$ och $x_2 = (u_0, u_1, u_2, u_L, i_0, i_1, i_2, i_C)$. Then

$$\text{rank }\ h_{x_2} = \text{rank }\ \frac{\partial h(x_1, x_2)}{\partial x_2} = \text{rank }\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -R_1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -R_2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} = 7 < 8$$

Summary of the three acts

- Act 1: simple, very similar to an ODE
- Act 2: a bit more difficult, took some algebra, but we were OK
- Act 3: significantly more difficult

The difference between these three acts were changes in components.

Important: All three are mathematically well-formed models!

A main property that separates them is: *differential index*

Transfer functions for model 1

The three models are linear, i.e., we can compute the transfer functions to show what is happening.

$$u_C = \frac{R_2}{R_1 + R_2 + sCR_1R_2} f, \quad u_L = f$$

$$i_L = \frac{1}{sL} f, \quad i_0 = \frac{R_1 + R_2 + s(L + CR_1R_2 + CLR_2 s)}{sL(R_1 + R_2 + CR_1R_2 s)} f$$

$$u_0 = f, \quad i_1 = \frac{1 + sCR_2}{R_1 + R_2 + sCR_1R_2} f$$

$$u_1 = \frac{R_1 + sCR_1R_2}{R_1 + R_2 + sCR_1R_2} f, \quad i_2 = \frac{1}{R_1 + R_2 + sCR_1R_2} f$$

$$u_2 = \frac{R_2}{R_1 + R_2 + sCR_1R_2} f, \quad i_C = \frac{sCR_2}{R_1 + R_2 + sCR_1R_2} f$$

Transfer functions for model 2

The three models are linear, i.e., we can compute the transfer functions to show what is happening.

$$i_L = \frac{1}{s} f, \qquad u_2 = \frac{R_2 R_3}{R_2 R_3 + R_1(R_2 + R_3)} f$$

$$u_L = f, \qquad i_3 = \frac{R_2}{R_2 R_3 + R_1(R_2 + R_3)} f$$

$$i_1 = \frac{R_2 + R_3}{R_2 R_3 + R_1(R_2 + R_3)} f, \qquad u_3 = \frac{R_2 R_3}{R_2 R_3 + R_1(R_2 + R_3)} f$$

$$u_1 = \frac{R_1(R_2 + R_3)}{R_2 R_3 + R_1(R_2 + R_3)} f, \qquad u_0 = f$$

$$i_2 = \frac{R_3}{R_2 R_3 + R_1(R_2 + R_3)} f, \qquad i_0 = \frac{R_1(R_2 + R_3) + sLR_3 + R_2(R_3 + sL)}{sL(R_2 R_3 + R_1(R_2 + R_3))} f$$

Transfer functions for model 3

The three models are linear, i.e., we can compute the transfer functions to show what is happening.

$$u_C = f, \qquad u_L = \frac{sLR_2}{R_1R_2 + sL(R_1+R_2)} f$$

$$i_L = \frac{R_2}{R_1R_2 + sL(R_1+R_2)} f, \qquad i_C = sCf$$

$$u_0 = f, \qquad i_0 = \frac{R_2 + sCR_2(R_1+sL) + sL(1+sCR_1)}{sLR_2 + R_1(R_2+sL)} f$$

$$u_1 = \frac{R_1(R_2+sL)}{sLR_2 + R_1(R_2+sL)} f, \qquad i_1 = \frac{R_2+sL}{sLR_2 + R_1(R_2+sL)} f$$

$$u_2 = \frac{sLR_2}{R_1R_2 + sL(R_1+R_2)} f, \qquad i_2 = \frac{sL}{R_1R_2 + sL(R_1+R_2)} f$$

Differential index

Index, one example

A linear example that illustrates an important difference between a DAE and an ODE

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \dot{x}_1 &= \dot{f}_2 - \ddot{f}_3 \\ \dot{x}_2 &= -x_1 + f_2 \\ \dot{x}_3 &= x_1 - f_2 - \ddot{f}_2 + \dot{f}_1 - f_3^{(3)} \end{aligned}$$

- What are allowed initial conditions? For an ODE they are free
- Not the case for a DAE, there might be “hidden” algebraic constraints

$$\begin{aligned} x_1 &= f_2 - \dot{f}_3 \\ x_2 &= f_3 \\ x_3 &= f_1 - \dot{f}_2 - f_3 + \ddot{f}_3 \end{aligned}$$

Something called *(differential) index* characterize DAEs

(Differential-) Index

A DAE is almost an ODE, just need some differentiation

$$\dot{x} = f(x, y)$$
$$0 = g(x, y)$$

Differentiate the second equation

$$0 = g_x \dot{x} + g_y \dot{y} = g_x f + g_y \dot{y}$$

If g_y^{-1} exists we can rewrite as

$$\dot{x} = f(x, y)$$
$$\dot{y} = -g_y^{-1} g_x f$$

Comments: solutions sets, equivalence.

Index, cont.

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

Definition

The minimum number of times the DAE has to be differentiated with respect to t to be able to determine $\dot{y}$ as a function of t och y is called the (differential-) index of the DAE.

- index might be solution dependent, uniform index
- There are several types of index, the above is called differential index.
- Perturbation index
- variants of the above (see paper)

Anyhow: index is a measure how far from an ODE the DAE is.

Linear constant DAEs of any index

$$E\dot{x} = Jx + Ku$$

Then there exists a non-singular matrix P and a change of variables $z = Qx$ such that

$$\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \begin{pmatrix} B \\ D \end{pmatrix} u$$

Where matrix N is nilpotent, i.e., there is an integer m such that $N^i \neq 0$ for $i < m$ and $N^m = 0$.

A simple algebra exercise gives that the solution to the DAE is

$$\dot{x}_1 = Az_1 + Bu, \quad z_2 = -\sum_{i=0}^{m-1} N^i D u^{(i)}$$

How is the degree of nilpotency m related to the index? Transfer function, how does it relate to the degrees of numerators and denominators?

Sufficient condition for index

$$\begin{aligned} F(y,\dot{y}) &= 0 \\ \frac{d}{dt}F(y,\dot{y}) &= 0 \\ &\vdots \\ \frac{d^{j-1}}{dt^{j-1}}F(y,\dot{y}) &= 0 \end{aligned} \quad \sim \quad \boldsymbol{F}_j(t, y, \boldsymbol{y}_j) = 0$$

Algebraically $\boldsymbol{F}_j(t, y, \boldsymbol{y}_j) = 0$ consists of nj equations in $nj + n$ unknown variables.

A sufficient condition for $\dot{y}$ is a unique function (locally) if t and y is that

$$\frac{\partial \boldsymbol{F}_j}{\partial \boldsymbol{y}_j}$$

is 1-full column rank

The DAE has index no larger than v if $\partial \boldsymbol{F}_{v+1} / \partial \boldsymbol{y}_{v+1}$ has 1-full rank and $\boldsymbol{F}_{v+1} = 0$ is consistent.

1-full rank

When has the equation

$$(A_1 \ \ A_2)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = b$$

a unique solution for x_1?

Unique x_1 solution if and only if

$$\text{rank } A = n_1 + \text{rank } A_2$$

Example:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = b$$

Now, back to the last slide, what does 1-full rank mean there?

Common forms for differential equations

- **ODE**

$$\dot{y} = f(y, t)$$

- **Hessenberg index 1/semi-explicit index 1**

$$\dot{x} = f(x, z, t)$$
$$0 = g(x, z, t), \quad g_z \text{ nonsingular for all } t$$

- **Hessenberg index 2**

$$\dot{x} = f(x, z, t)$$
$$0 = g(x, t), \quad g_x f_z \text{ nonsingular for all } t$$

Our index-2 equation, all algebraic variables are “index-2” variables.

Remainder of the lecture

The remainder of the lecture will introduce some important differences between ODEs and DAEs from a simulation perspective. We will come back to these in detail in upcoming lectures.

1) Initial conditions
2a) Simulation of equations with index 0 and 1
2b) Simulation of equations with index ≥ 2

Initial conditions

Bullet 1: Initial conditions

For an ODE

$$\dot{y}(t) = f(t, y(t))$$

it sufficient that the initial condition $y(0) = y_0$ satisfies

$$\dot{y}(0) = f(0, y_0)$$

For a DAE

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

is it sufficient that the initial conditions $y(0)$ and $\dot{y}(0)$ satisfies

$$F(0, y(0), \dot{y}(0)) = 0?$$

Remember the model that had no degrees of freedom

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned}$$

- Index and “hidden” conditions
- Methods to determine consistent initial conditions
- Pantelides algorithm

Initial conditions, cont.

What degrees of freedom do we have for the initial condition? In the equations

$$\begin{aligned} \dot{x}_1 + x_2 + x_3 &= f_1 \\ \dot{x}_2 + x_1 &= f_2 \\ x_2 &= f_3 \end{aligned}$$

there is no freedom at all and the solution was uniquely determined (in the class of smooth functions) directly by the equations.

If we have m equations/variables, it holds that the degrees of freedom l that $0 \leq l \leq m$ and it is not trivial to find *consistent* initial conditions.

$$\begin{aligned} \dot{x} &= f(x, y) \\ 0 &= g(x, y) \end{aligned}$$

Pantelides algorithm

We will come back to a possible solution later

Simulation of DAEs with low index

Bullet 2a: Index 1 “as easy” as ODE

Will come back to this, but the basic principle is easily illustrated.

Assume a semi-explicit DAE in the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, t) \\ 0 &= f_2(x_1, x_2, t) \end{aligned}$$

with index 1. Then,

$$\frac{\partial f_2}{\partial x_2}$$

has full column rank and it exists a (local) inverse w.r.t. x_2.

The algebraic variable can then be inserted in the dynamic equation resulting in an ODE which can be solved using any standard ODE method.

Bullet 2a: Index 1 “as easy” as ODE, cont.

Consider an implicit index-1 DAE

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

Apply a basic implicit Euler backward

$$F\left(\frac{x_t - x_{t-1}}{h_t}, x_t, t\right) = 0$$

and solve numerically for x_t. Index-1 property ensures that a solution exists.

Important note: Procedure *no different* than implicit Euler for ODEs.

A simple example

Consider the DAE

$$\dot{x} = -x + y \quad \Rightarrow \quad x(t) = -y(t) = x(0)e^{-2t}$$
$$0 = x + y$$

A Backward Euler step gives

$$\frac{x_{t+1} - x_t}{h} = -x_{t+1} + y_{t+1} \quad \sim \quad x_{t+1} = \frac{1}{1+2h} x_t$$
$$0 = x_{t+1} + y_{t+1} \qquad\qquad y_{t+1} = -x_{t+1}$$

which is exactly what you would've gotten for BE for the original

$$\dot{x} = -2x, \quad \text{and} \quad y(t) = -x(t)$$

Now, take a look at Forward Euler and we'll see that it doesn't even make sense.

Bullet 2a: Index 1 “as easy” as ODE, cont

One conclusion: BDF and other typical implicit solvers will work approximately the same for DAEs of index 1 as for ODEs.

There are practical differences though, see Hairer/Wanner and the following papers for further details

- Petzold, “*Differential/algebraic equations are not ODEs*”
- Brenan, Campbell and Petzold Petzold, “*Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*”

Bullet 2b: Why is index > 1 so difficult?

Equations you, generally, can solve using basic ODE methodology is

- Index 1 DAEs (more to follow)
- Linear DAEs with constant coefficients of any index (kind of)

$$A\dot{y} + By = f$$

  Will not pursue this here. More details in "*ODE methods for the solution of differential/ algebraic systems*".

- For index > 1, direct ODE methodology does not work at all. We need new techniques and index reduction is one possibility we will discuss a lot in upcoming lectures.

Implicit and semi-explicit forms

LINKÖPING
UNIVERSITY

Implicit and semi-explicit forms

A fully implicit DAE

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

can always be rewritten as a semi-explicit DAE by introducing a new variable x' (algebraic, should not be confused with $\dot{x}$)

$$\begin{aligned} \dot{x} &= x' \\ F(x', x) &= 0 \end{aligned}$$

Q

Does this mean that we can forget about implicit forms and focus on semi-explicit?

A

No, not really.

An implicit example

Consider the implicit index-1 DAE

$$\begin{aligned} e_1 : \quad \dot{x}_1 + \dot{x}_2 &= u_1 \\ e_2 : \quad x_1 - x_2 &= u_2 \end{aligned}$$

From equations $(e_1, e_2, \dot{e}_2)$ we can solve for the highest derivatives.

Transform the DAE into a semi-explicit DAE by introducing x_1' and x_2'

$$\begin{aligned} e_1 : \quad x_1' + x_2' &= u_1 \\ e_2 : \quad x_1 - x_2 &= u_2 \\ e_3 : \quad \frac{d}{dt}x_1 &= x_1' \\ e_4 : \quad \frac{d}{dt}x_2 &= x_2' \end{aligned}$$

Q

What is the (differential-)index of the semi-explicit formulation?

An implicit example, cont'd

Turns out that

$$\begin{aligned} e_1 : &\quad x_1' + x_2' = u_1 \\ e_2 : &\quad x_1 - x_2 = u_2 \\ e_3 : &\quad \frac{d}{dt} x_1 = x_1' \\ e_4 : &\quad \frac{d}{dt} x_2 = x_2' \end{aligned}$$

has index 2.

Assignment: Verify that you need $(e_1, \dot{e}_1, e_2, \dot{e}_2, e_3, \dot{e}_3, e_4, \dot{e}_4, \ddot{e}_4)$ to be able to solve for highest derivatives.

Rule of thumb

Going from fully implicit to semi-explicit increases index by 1

Summary

Take-aways

- Differential-algebraic equations, a natural way to model and a natural result from, e.g., object-oriented modeling in Modelica
- Model equations keep structure better than ODE reformulations
- Differential-index – a key characteristic of a DAE
  - Low index (≤ 1) shares many characteristics with ODEs and can also be directly simulated with (implicit) ODE methods
  - High index (> 1) requires special treatment; such models are not uncommon
- Feasible initial conditions more complex than for ODEs

liu.se