A Modeling Methodology

1. Introduction
2. Representation of Models
3. Units
4. Schematic Diagrams
5. A Water Tank
6. Electrical Circuits
7. Summary

A Modeling Methodology

- Purpose of modeling: understanding, control design, diagnostics, ...
- Cut a system into subsystems
- Write mass, momentum and energy balances for each subsystem
- Discretize partial differential equations
- Add constitutive equations for material properties
- Validity ranges
- The model format is differential algebraic equations
- Use object orientation to structure the system
- Let software (Modelica) handle bookkeeping and transformations
- Build component libraries
- Organize for reuse

A Modeling Methodology

1. Introduction
2. Representation of Models
3. Units
4. Schematic Diagrams
5. A Water Tank
6. Electrical Circuits
7. Summary

Representation of Models

- Pictures and Graphs for overview
  - Schematic diagrams "Mickey Mouse Pictures"
  - Block diagrams
  - Flow sheets
  - Domains specific graphs
  - Bond graphs
- Mathematical models
- Multiple views
- Interactive environments
- Virtual reality

Mathematical Models

- Purpose
- Assumptions
- Equations
- Normalization
- Units
- Range of validity
- Inputs, outputs, states
- Parameters
- Steady state properties
- Nonlinear models
- Simulation
- Visualization

Model Types

Continuous

- Equations - Steady state
- Ordinary Differential Equations
- Differential Algebraic Equations
- Partial Differential Equations

Discrete

- State Machines
- Petri Nets
- Grafcet
- Grafchart

Hybrid

Approximations

- Physical simplifications
- Important phenomena
- Important parameters
- System theoretical
- Experimental - identification
- A family of models

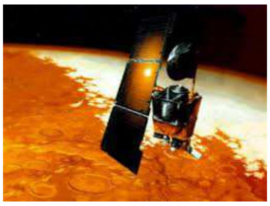
Linearized Models

- Equations
- Normalization, dimension free parameters
- Range of validity
- Transfer functions
- Frequency responses
- Time constants and gains
- RHP poles and zeros, time delays
- Relations to physical parameters

A Modeling Methodology

1. Introduction
2. Representation of Models
3. Units
4. Schematic Diagrams
5. A Water Tank
6. Electrical Circuits
7. Summary

Importance of Units - The Mars Climate Orbiter 1999



Mars Climate Orbiter Failure Board Release Report, Nov. 10, 1999: ... "The 'root cause' of the loss of the spacecraft was the failed translation of English units into metric units in a segment of ground-based, navigation- related mission software, as NASA has previously announced," said Arthur Stephenson, chairman of the Mars Climate Orbiter Mission Failure Investigation Board.

Modelica has excellent facilities for dealing with units!

SI Units

A fundamental reform of the SI-units was made in May 2019. All base units should be expressed in 7 physical quantities: kg, m, s, A, K, mol, cd.
All SI units are defined by declaring that seven defining constants have certain exact numerical values when expressed in terms of their SI units. These defining constants are the speed of light in vacuum c, the hyperfine transition frequency of caesium νCs, the Planck constant h, the elementary charge e, the Boltzmann constant k, the Avogadro constant NA, and the luminous efficacy Kcd. The nature of the defining constants ranges from fundamental constants of nature such as c to the purely technical constant Kcd. The last artefact used by the SI was the International Prototype of the Kilogram, a cylinder of platinum-iridium.

Units in Modelica

Three ways of introducing units i Modelica
parameter Modelica.SIunits.Mass m = 2; Modelica.SIunits.Velocity v(start=3);
import Modelica.SIunits; parameter SIunits.Mass m = 2; SIunits.Velocity v(start=3);
import SI = Modelica.SIunits; parameter SI.Mass m = 2; SI.Velocity v(start=3);
Modelica checks units in equations.

Example

```
model MovingMass2
parameter Real m(min=0,unit=kg) = 2;
parameter Real f(unit=N)
Real s;
Real v;
equation
v = der(s);
m*der(v) = f;
end MovingMass2;
```

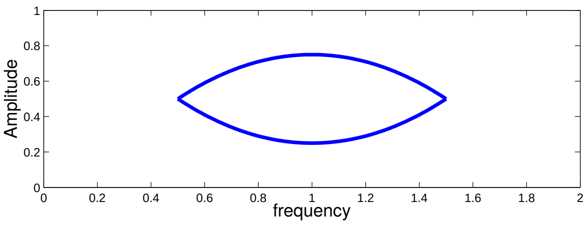
When units are introduced relations are also checked for correct dimension during compilation!

Related Modelica Features

Real variables have attributes: min, max, start, fixed, nominal, stateSelect.
A good feature for testing, compare alarms on analog computers.

Validity Ranges

- The analog computing heritage: scaling and alarm
- The uncertainty lemon: Gilles J.C., Decaulne P., Pelegrin M., Theorie des systemes asservis, Dunod, Paris, 1959



- Frequency can be replaced by $|dx/dt|$

Good reserach project: Capture uncertainty in Modelica models

A Modeling Methodology

1. Introduction
2. Representation of Models
3. Units
4. Schematic Diagrams
5. A Water Tank
6. Electrical Circuits
7. Summary

Schematic Diagrams – Draper “Mickey Mouse pictures”

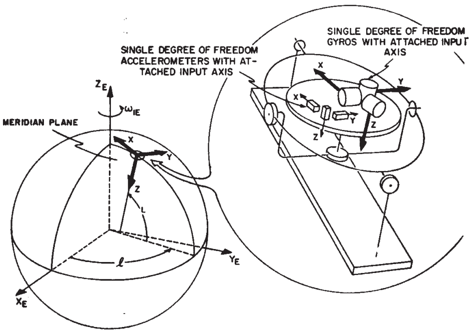


FIG. 14. *An instrumented model of local geographic co-ordinates.*

Schematic Diagrams – Draper “Mickey Mouse pictures”

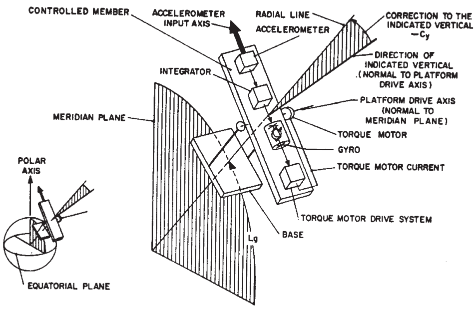
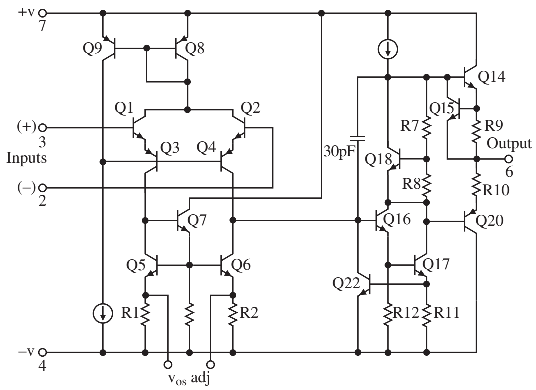
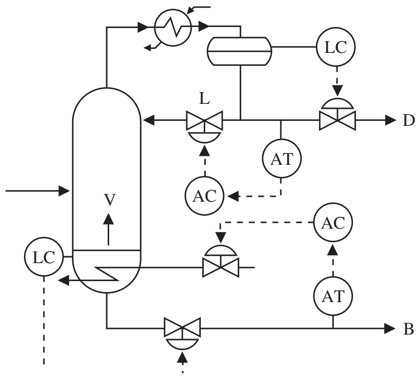


FIG. 13. *Schematic diagram of a single-axis vertical indicating system constrained to move in a meridian plane.*

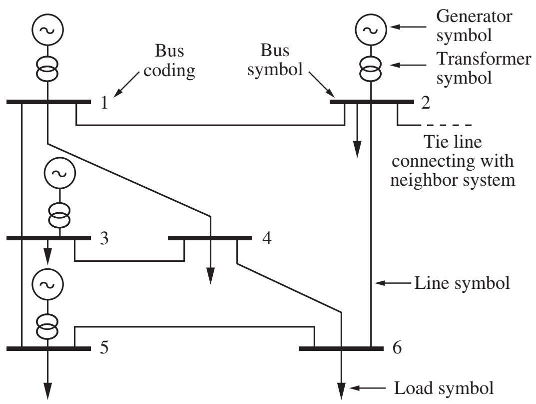
Circuit Diagram



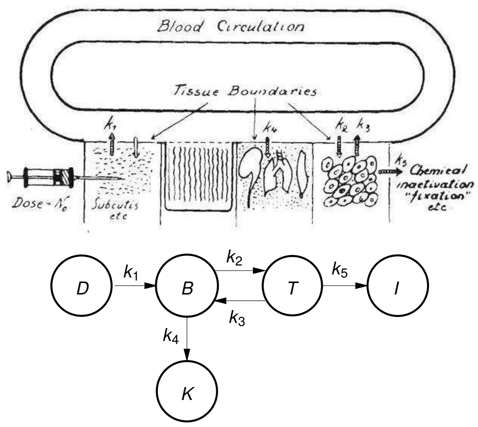
Piping and Instrumentation Diagram – P&I Diagram



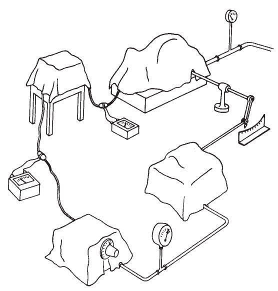
Power Networks



Compartment Models



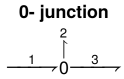
Blockdiagrams



Bondgraphs - Paynter MIT

- Flow (*f* current, flow) and through (*e* voltage, pressure) variables
- Useful when only the energy balance is important - signal converters
- States are obtained automatically
- Junctions

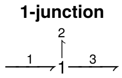
0- junction



$e_1 = e_2 = e_3$

$f_1 = f_2 + f_3$

1-junction



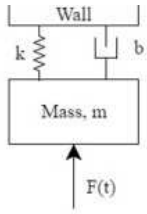
$e_1 = e_2 + e_3$

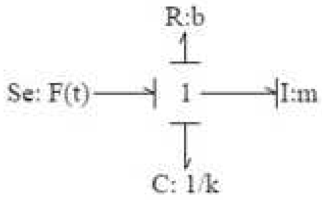
$f_1 = f_2 = f_3$

R, I and C elements

$$e(t) = Ri(t) \qquad m\frac{de}{dt} = f \qquad \frac{1}{C}\int^{T} e(\tau)d\tau = f(t)$$

Bond Graphs - Example





$$F(t) = bv + m\frac{dv}{dt} + k\int vdt$$

A Modeling Methodology

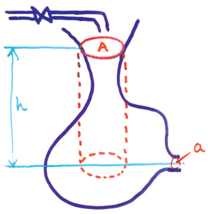
A Water Tank

How do level h and outflow q_{out} depend on the inflow q_{in}?
Assume: Constant density

$$\frac{dV}{dt} = q_{in} - q_{out} \qquad \text{Massbalance}$$

$$V = \int_0^h A(h)dh \qquad \text{Geometry}$$

$$q_{out} = a\sqrt{2gh} \qquad \text{Energybalance}$$



What parameters are relevant?
How do we determine them?
Several ways to choose the states!

Ordinary Differential Equations

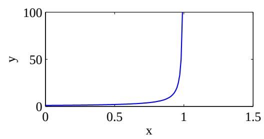
$$\frac{dx}{dt} = f(x,t), \qquad x(0) = a$$

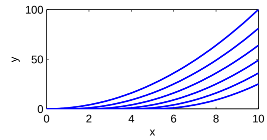
Existence and uniqueness - Lipschitz continuity

$$|f(x) - f(y)| < k|x - y|$$

Finite escape time or many solutions

$$\frac{dx}{dt} = x^2 \qquad \frac{dx}{dt} = \sqrt{x}$$





Analysis and Simplification

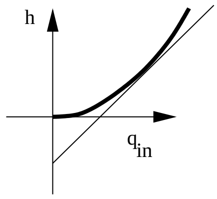
Choosing h as a state variable we find

$$\frac{dh}{dt} = \frac{1}{A(h)}(q_{in} - a\sqrt{2gh})$$
$$q_{out} = a\sqrt{2gh}$$

One function $A(h)$ and one parameter a.

Steady state relation

$$q_{out} = q_{in}$$
$$h = \frac{q_{in}^2}{2ga^2}$$



Not influenced by A!

A Difficulty - Physics changes for low levels

The equation
$$\frac{dh}{dt} = \frac{1}{A(h)}(q_{in} - a\sqrt{2gh})$$

Equation not Lipschitz for $h = 0$! Trouble!
When water level is below the upper part of the hole we get

$$q_{out} = \int_0^h \sqrt{2gx}\, dA(x)$$

Rectangular cross section of width b give $dA(x) = bdx$, hence

$$q_{out} = \int_0^h \sqrt{2gx}bdx = \frac{2}{3}bh\sqrt{2gh}$$

Effective outlet area

Linearization

Equations

$$\frac{dh}{dt} = \frac{1}{A(h)}(q_{in} - a\sqrt{2gh})$$
$$q_{out} = a\sqrt{2gh}$$

Linearized equation

$$\frac{d\delta h}{dt} = -\frac{q_0}{2A(h_0)h_0}\delta h + \frac{1}{A(h_0)}\delta q_{in}$$
$$\delta q_{out} = \frac{q_0}{2h_0}\delta q_{in}$$

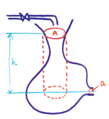
Transfer functions

$$\frac{H}{Q_in} = \frac{2h_0}{q_0}\frac{1}{1+sT}$$
$$\frac{Q_{out}}{Q_in} = \frac{1}{1+sT}$$

Static gain $\frac{2h_0}{q_0}$.

Time constant $T = \frac{2Ah_0}{q_0}$

- Physical interpretation



Solving the Equation

For zero inflow we have
$$\frac{dh}{dt} = -\frac{a}{A}\sqrt{2gh}$$

One solution is

$$h = \left(\sqrt{h_0} - \frac{a}{A}\sqrt{g/2t}\right)^2 = h_0\left(1 - \frac{t}{T}\right)^2$$
$$q_{out} = a\sqrt{2gh_0}\left(1 - \frac{t}{T}\right) = q_0\left(1 - \frac{t}{T}\right)$$

Summary of Simple Water Tank

- Simple prototype modeling
- Equations difficult to solve analytically even in this simple case (cf Vannevar Bush)
- Mathematical conditions (Lipschitz good indicators)
- Linearization and steady state solutions helpful for insight
- Special cases useful to validate results
- Physical interpretation of parameters useful
- Think about good transformations

A Modeling Methodology

Electrical Circuits

- A good place to start
- We all know the basic models
- Components
  - Resistor $V = R\,I$
  - Capacitor $I = C\,\frac{dV}{dt}$
  - Inductor $V = L\,\frac{dI}{dt}$
- Combination rules
  - Kirchhoff's current law
  - Kirchhoff's voltage law

Component Hierarchy

Signal sources

- Voltage and current sources

Two port components

- Conductor, Resistor, Capacitor, Inductor,

Four port components

- Transformers, Gyrators

Special devices

- Ground, Pins, Ports, Interfaces, Meters

Modelica Standard Library

Top Level

Blocks, Constants, Electrical, Icons,
Math, Mechanics, SIunits

Modelica.Electrical.Analog

Basic, Examples, Ideal, Interfaces,
Lines, Semiconductors, Sensors, Sources

Modelica.Electrical.Analog.Basic

Ground, Resistor, Conductor, Capacitor, Inductor,
Transformer, Gyrator, EMF,
Voltage and current sources VCV, VCC, CCV, CCC

OnePort

```
partial model OnePort “Component with two electrical
pins p and n and current i from p to n”
SIunits.Voltage v
“Voltage drop between the two pins (= p.v - n.v)”
```

Component Code

```
model Resistor “Ideal linear electrical resistor”
extends Modelica.Electrical.Analog.Interfaces.OnePort;
parameter SIunits.Resistance R=1 "Resistance";
equation
R*i = v;
end Resistor

model Capacitor “Ideal linear electrical capacitor”
extends Modelica.Electrical.Analog.Interfaces.OnePort;
parameter SIunits.Capacitance C=1 "Capacitance";
equation
i = C*der(v);
end Capacitor
```

A Modeling Methodology

A Modeling Methodology

- Purpose of modeling: understanding, control design, diagnostics, ...
- Cut a system into subsystems
- Write mass, momentum and energy balances for each subsystem
- Discretize partial differential equations
- Add constitutive equations for material properties
- Validity ranges
- The model format is differential algebraic equations
- Use object orientation to structure the system
- Let software (Modelica) handle bookkeeping and transformations
- Build component libraries
- Organize for reuse