Modeling

from Physics to Languages and Software

Karl Johan Åström
Department of Automatic Control LTH
Lund University

1

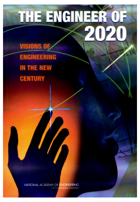
Modeling

- Essential for the development of science, example: Brahe, Kepler, Newton
- Essential element of all engineering
- Process design and optimization
- Insight and understanding
- Control design and optimization
- Implementation – The internal model principle
- Simulation: Illustrate behavior of model, HIL hardware in the loop, SIL software in the loop
- Validation and verification
- Diagnostics, fault detection, reconfiguration
- Digital twin

2

Modeling in Engineering

There will be growth in areas of simulation and modeling around the creation of new engineering "structures". Computer-based design-build engineering ... will become the norm for most product designs, accelerating the creation of complex structures for which multiple subsystems combine to form a final product.



NAE The Engineer of 2020

3

Drawings and Models

- Drawings show what systems look like
- Models show how systems behave
- Equally important
- Should be handled in the same way with respect to creation,development, documentation and intellectual property

4

Model Types

- Partial differential equations PDE
- Difference equations
- Ordinary differential equations ODE
- Differential algebraic equations DAE
- Discrete Event Systems DES

5

- Static models

$$f(x,u,v)=0, h(x,u,v)\leq 0$$

- Dynamic models ODE, DAE, PDE

$$\frac{dx}{dt}=f(x,u), F(\dot{z},z,u)=0, \frac{\partial T}{\partial t}=\frac{\partial^2 T}{\partial x^2}$$

- Logic and finite state machines

$$g_i(\alpha,\beta) \implies \alpha' = r_i(\alpha), i=1,\ldots,N.$$

- Discrete event systems, hybrid systems

$$\frac{dx}{dt}=f_\alpha(x,u,v),$$

$$g_i(x,\alpha,\beta) \implies \alpha' = r_i(x,\alpha)$$

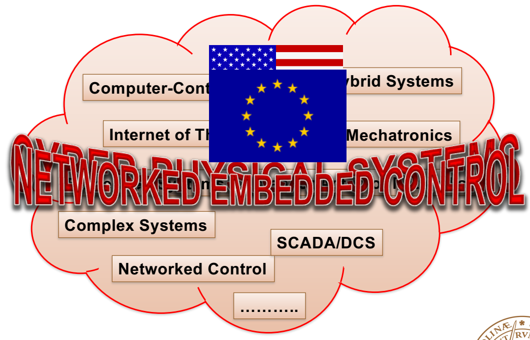
6

A Rich Field

- Mechanical
- Chemical
- Electrical
- Fluid
- Thermal
- Vehicles
- Multibody systems
- Robotics
- Mechatronics
- Networks
- Power systems
- Chemical Reactors
- Biological systems
- Ecosystems
- Pharmacokinetics
- Compartment Models

Essential for all engineering!

7



Slide from Karl Erik Årzén

8

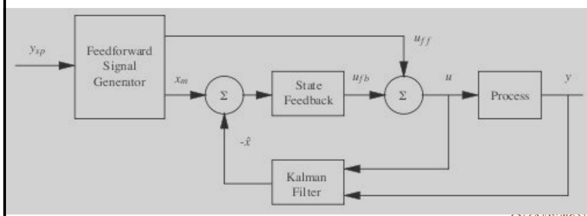
Control Applications

- Process and control design
- Simulation
- Performance guarantees
- Implementation
- Validation
- Fault detection
- Diagnostics
- Reconfiguration

9

The Internal Model Principle

A good controller contains a model of the process and its environment



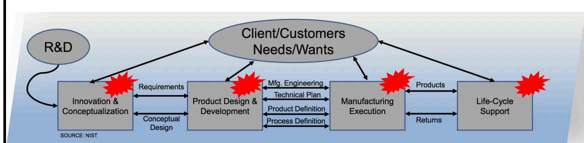
10

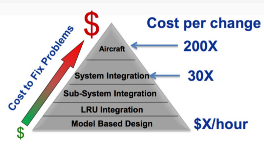
Model Based Systems Engineering

- From requirements to hardware in operation
- Static modeling and system design
- Integrated process and control design
- Architecture exploration
- Optimization
- Model reduction (reduced order models)
- Parameter estimation
- Embedded systems
- Incorporation in tool chain

11

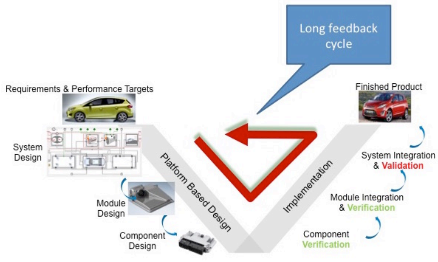
Model Based Systems Engineering





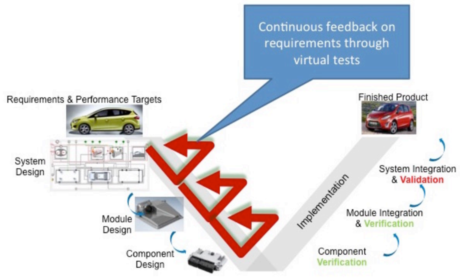
12

Design Flow



13

Model-Based Design Flow



14

Complexity

- Many different physical domains
- Large physical dimensions
- Large number of components
- Complex behavior
- Concurrent design of systems & control
- Tight coupling: Computers and physical devices, Recirculation, heat recovery, just-in-time production
- Mixed continuous- and discrete-time behavior: hybrid systems, Ex: gear box

15

16

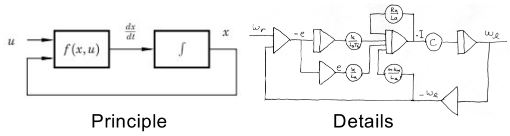
Analog Computing

- For a long time analog computing was the only way to simulate complex systems
- Used both for simulation and for controllers
- Simple, fast intuitive
- Easy to see effect of parameter changes
- Prediction of tides 1886
- Electrical systems 1940
- Widespread industrial use through 1950
- Formation of user groups
- Digital computers replaced analog 1960--

17

Analog Computing

- Solving ordinary differential equations ODE
- A feedback loop is used to solve ODE



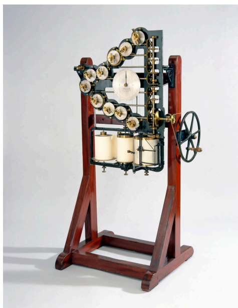
Principle Details

- Integrators, adders and function generation
- Parallelism
- Granularity and aggregation
- Alarms for algebraic loops (without integrators)
- Alarms for out of scale!!

18

Prediction of Tides

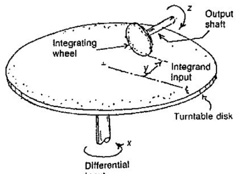
- Sir William Thomson (Lord Kelvin)1872
- Machine for predicting for computing 8 tidal components1873
- Larger machine for 10 tidal components 1875-76
- Machine for 20 tidal components for Government of India in 1879
- 24 harmonic components 1881



19

Analog Computers

- Ball and disc integrator
- Lord Kelvin tide predictor 1872
- Vannevar Bush differential analyzer 1927
- Pneumatic PID 1930
- Fire control systems
- V2 missile 1943
- Electronic computers
- EAI, Philbrick
- Digital emulators



20

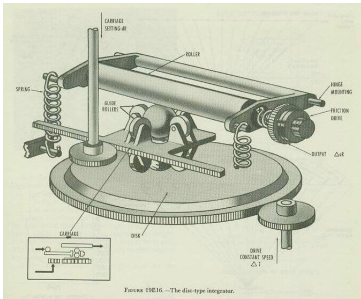
Vannevar Bush 1927 - 41

Engineering can progress no faster than the mathematical analysis on which it is based. Formal mathematics is frequently inadequate for numerous problems, a mechanical solution offers the most promise. => The mechanical differential analyzer (6 integrators)

. V. Bush and H. Hazen, "The differential analyzer: A new machine for solving differential equations," *J. Franklin Inst.*, vol. 212, no. 4, pp. 447–488, Oct. 1931
Ball and disc integrator and torque amplifier.

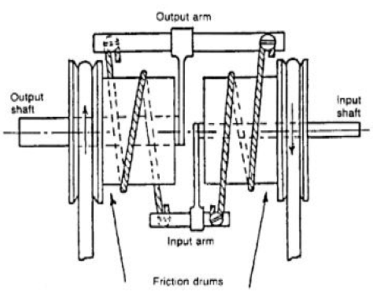
21

Kelvin Integrator 1886



22

Torque Amplifier



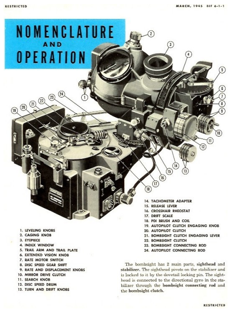
23

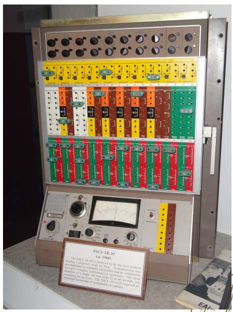
Early MDAs

- Bush and Hazen MIT 1928 6 integrators
- Two 10 integrator copies to Ballistics Research Lab Aberdeen and Moore School University of Pennsylvania
- Two 8 integrator copies to universities in Cambridge and Manchester
- University of Oslo1938-42 12 integrators
- University of Leningrad 6 integrators

24

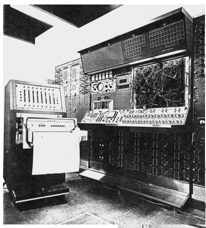
Analog Computing



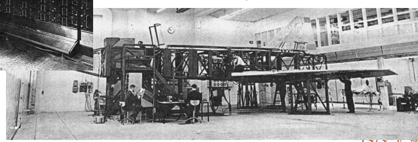


25

Analog Simulation & HIL



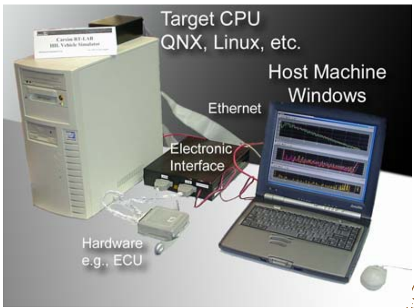
- Ordinary differential equations dx/dt=f(x,p)
- Scaling, patching, potentiometer setting
- Algebraic loops
- Set initial conditions and parameters
- Direct manipulation of parameters
- Print results
- Simulation centers: competence, interactions
  Skandinaviska Analogimaskin Sällskapet 1957
- Hardware in the loop simulation HIL



Saab 1950s

26

HIL Today



27

The Whirlwind Computer

- Flight trainer-analyzer 1944-56
- MIT Servomechanism Laboratory Forrester 1944
- From analog to digital computing decided on digital
- Core memory 1953 (Jay Forrester)
- Ken Olsen Digital Equipment 1957
- PDP 8 1965

28

Computing

- Vannevar Bush 1927. Engineering can progress no faster than the mathematical analysis on which it is based. Formal mathematics is frequently inadequate for numerous problems, a mechanical solution offers the most promise.
- Herman Goldstine IBM 1962. When things change by two orders of magnitude it is revolution not evolution.
- Gordon Moore Intel 1965: The number of transistors per square inch on integrated circuits has doubled in approximately 18 months. A revolution every 10 years!
- Strong potential, but so far algorithms and software have not delivered corresponding productivity increases!

29

The Swedish Scene

- Stig Ekelöf CTH Prof Electrical Engineering 1938
  "Time to get mechanical differential analyzer"
  MDA with 4 integrators ready by 1950
- KTH Aero EDA, Qvarnström and Lundin 1949
- FOA Freda, Stemme and Wikland 1950
- Saab SEDA 1950
- CTH Wallman EDA 1953
- Analog computing centers: Axel Johnson Institutet, ASEA, Bofors, FOA, Saab, Vattenfall
- Skandinaviska Analogi Maskin Sällskapet (SAMS) 1958
- Matematikmaskinnämnden MMN, IVA, KTH, FOA
  Marinförvaltningen, BARK 1948
- BESK 1953, SMIL 1955-56, SARA Saabs
  Räkneautomat 1957, Facit EDB 1958

30

Digital Emulators

- CSMP IBM 1962
- MIMIC Wright-Patterson 1965
- Babels tower > 30 emulators by 1965
- CSSL Simulation Council 1967
- ACSL Gauthier and Mitchell 1975
- SIMNON Elmqvist 1975 Masters thesis
- MATLAB Cleve Moler 1980
- System Build, MatrixX 1984
- LabView 1986
- Simulink 1991

31

CSSL - From computing to problem description

```
PROGRAM test
INITIAL
  a=1
END $ "of initial"
DYNAMIC
  DERIVATIVE
    f=a*x^2+b
    b=1
    xdot=f
  END $ "of DERIVATIVE"
  termt(t.ge.tmx or x.gt.1)
END $ "of DYNAMIC"
END $ "of PROGRAM"
```

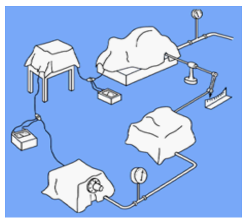
- Declarative
- Sorting
- Algebraic loops
- FORTRAN preprocessor
- FORTRAN compilation
- Predefined functions termt
- Run time library
- Explicit integration algorithm
- Parameter changes without recompilation

32

33

Block Diagram Modeling

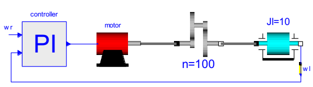
- Information hiding
- Causal inputs-output blocks
- Blocks have state (integrators) and are described by ODE
- Very useful abstraction
- Essential for control
- Base for analog computing
- BUT not for serious physical modeling - states may diappear when connecting two systems

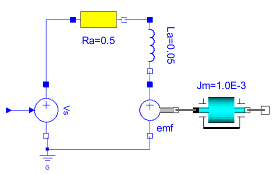


Oppelt 1954 *Kleines Handbuch technischer Regelvorgänge* 1954

34

Significant Effort to Obtain State Models (ODE)





35

Mathematical Model

Physical model DAE

$$J_\ell \frac{d\omega_\ell}{dt} = T_g$$
$$J_m \frac{d\omega_m}{dt} = k_m I - T_m$$
$$\omega_m = n\omega_\ell$$
$$T_g = nT_m$$
$$L_a \frac{dI}{dt} + R_a I = V_s - k_m\omega_m$$
$$V_s = k\Big(\omega_r - \omega_\ell + \frac{1}{T_i}\theta\Big)$$
$$\frac{d\theta}{dt} = \omega_r - \omega_\ell$$

Transform to state model ODE

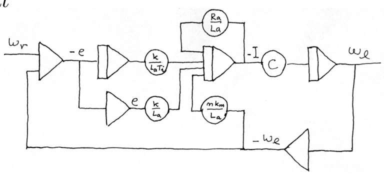
$$\frac{d\omega_\ell}{dt} = \frac{nk_m}{J_\ell + n^2 J_m} I$$
$$\frac{dI}{dt} = -\frac{R_a}{L_a} I + \frac{k}{L_a}\Big(\omega_r - \omega_\ell + \frac{1}{T_i}\theta\Big) - \frac{nk_m}{L_a}\omega_\ell$$
$$\frac{d\theta}{dt} = \omega_r - \omega_\ell$$

7 equations
4 differentiated variables
3 algebraic equations
3 states in ODE
Parameters and physics are aggregated

36

Analog Circuit Diagram

$$\frac{d\omega_\ell}{dt} = \frac{nk_m}{J_t + n^2 J_m} I$$
$$\frac{dI}{dt} = -\frac{R_a}{L_a} I + \frac{k}{L+a}\left(\omega_r - \omega_\ell + \frac{1}{T_i}\theta\right) + \frac{nk_m}{L_a}\omega_\ell$$
$$\frac{d\theta}{dt} = \omega_r - \omega_\ell$$



37

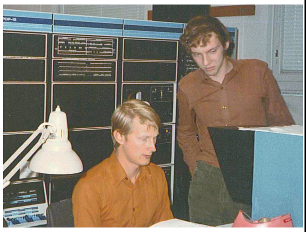
LTH in the 70s

- New department (1965) at new school LTH (1961) associated with an old university (Lund University 1666)
- Research program in Control Department: Optimization, Computer Control, System Identification, Adaptive Control + Honest applications = real industrial projects => Computer Aided Control Engineering, CACE
- Research program on interactive computing: INTRAC, SYNPAC, IDPAC, MODPAC (Wieslander), FORTRAN, widely distributed to industry and universities
- A nonlinear simulator was missing
- IBM San Jose and CSMP development (Brennan)

38

Wieslander and Elmqvist

PDP 15 16k words core,
256 k disk,
Tektronics display
inkjet printer



Hilding Elmqvist and Johan Wieslander. INTRAC A Commmunication Module for Interactive Programs – Language Manual. *Research Reports TFRT-3149 Department of Automatic Control LTH 1978*

39

Simnon Elmqvist 1973

Block diagram language, interactive simulator change parameters and initial conditions without recompilation. Continuous and discrete time systems. Formal syntax in Bachus Naur format. Only scalar real variables.
Scripting language: 6 basic commands: SYST, PAR, INIT SIMU, PLOT, AXES
8 auxiliary: STORE, SHOW, DISP, SPLIT, HCOPY, ALGOR, ERROR, MACRO

```
CONTINUOUS SYSTEM proc
Input u
Output y
State x
Der dx
dx=sat(u,0.1)
END

CONNECTING SYSTEM
yr(reg)=1; y(reg)=y(proc)
u(proc)=u(reg)
END
```

```
DISCRETE SYSTEM reg
Input yr y
Output u
State I
New nI
Tsamp ts
ts=t+h
v=k*e+I
u=sat(v.0.1)
nI=I+k*h*e/Ti+u-v
k:1
h:0.1
END
```

Hilding Elmqvist MS Thesis SIMNON –User's Guide *Research Report TFRT-3106 Department of Automatic Control LTH 1973*

40

Computer Aided Control Engineering CACE

- Design suite
- Modeling from data IDPAC
- Control design SYNPAC, POLPAC, …
- Simulation Simnon
- Common interface Intrac
- Distributed to academia and industry
- General Electric Research – John Cassidy
- Application projects
- Project terminated in early 1980's because of Fortran, Matlab and Simula

K.J. Åström, Computer aided modeling, analysis and design of control systems - a survey IEEE CSM 3:2 1983

41

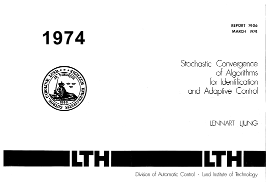
Matlab

- Linpac, Cleve Moler 1980: MATLAB – An Interactive Matrix Laboratory, Workshop on Numerical Methods in Automatic Control Lund Sept 1980. MATLAB users guide June 1980, Rev. June 1981 (FORTRAN)
- Systems Control Inc Palo Alto CTRL-C – Control extensions and graphics to Moler's MATLAB code
- Integrated Systems Inc (ISI), Matrix-X 1982, SystemBuild 1984, Code generation
- John Little, MathWorks, PC Matlab1984, Simulink 1991, Toolboxes
- Blaise 1984, Scilab INRIA 1994, Octave GNU 1993, SysQuake (interactive) Calerga 1998
- Comsol FemLab (PDE modeling) 2000

42

Matlab System ID Toolbox

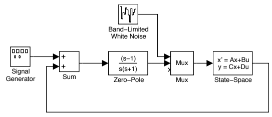
Lennart Ljung rewrote IDPAC in Matlab and improved it.



43

Simulink the Ultimate Block Diagram Tool

- Mimics the analog computer with more general blocks
- Each block a state model

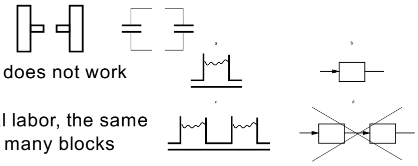


- MATLAB, Stateflow
- Granularity and Structuring
- Graphical aggregation and disaggregation
- Much manual manipulation from physics to blocks
- Syntax and semantics not publicly available

44

But!!

States may disappear when systems are connected – warning algebraic loop!



Composition does not work

Much manual labor, the same parameter in many blocks

Lesson 1: Block diagrams and Simulink are not suitable for serious physical modeling

Lesson 2: Don't stick to a paradigm based on old technology when new technology emerges!!

45

46

Physical Modeling

- Divide a system into subsystems, define interfaces
- Use object orientation for structuring
- Mass, momentum and energy balances for subsystems
- Interconnections naturally modeled by algebraic equations
- Material properties as functions or tables
- Resulting in a Differential Algebraic Equation DAE
- Create libraries for reuse
- Symbolic computation to generate code for simulation and optimization

47

Differential Algebraic Equation DAE

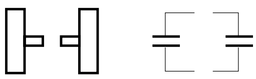
Differential algebraic equation

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

Special case

$$\frac{dx}{dt} = f(x, y), \qquad g(x, y) = 0$$

Natural to describe connections:



48

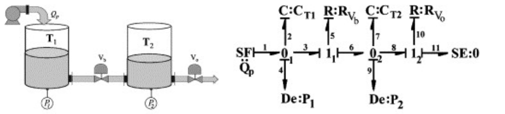
Electronics

- Electric circuits: multi port systems
- Component based: resistor, capacitor, inductor, transistor. Model components and connect them.
- Nagel, Peterson Berkeley 1973, SPICE2 1975, …
- Kirchoffs voltage and current laws
- Node equations
- Nonlinear equations – Tearing Gabriel Kron
- Differential algebraic equations DAE, Gear
- Important part of Tool Chain for VLSI design
- Spice Peterson Berkeley 1970 IEEE Milestone 2011

49

Bond Graphs

- Henry Paynter MIT, *Analysis and design of engineering systems, The M.I.T. Press, Boston, 1961*
- Graphical representation of bi-directional exchange of energy using across (V) and through (I) variables



- Difficult to handle more than one essential balance equation, not convenient for boiler model
- Did not work for boiler model, verified by Paynters alumni at UTexas

50

Some Modeling Tools

Program	Domain	Originator	Year
SAP	Structural Dynamics	Boeing, UCB	1956
SPEEDUP	Chemical Engineering	Imperial	1964
ECAP1	Electronics	IBM	1965
NASTRAN	Structural Dynamics	MSC/NASA	1968
SPICE	Integrated Circuits	UC Berkeley	1970
ANSYS	Structural Dynamics	SAS (ANSYS)	1970
SPICE	Electronics	UCB	1973
TRNSYS	Buildings	Wisconsin	1975
ABAQUS	Structural Dynamics	MARC	1978
Dymola	General Purpose	LTH	1978
Adams	Multibody Dynamics	Michigan	1981
AspenPlus	Chemical Engineering	Aspen	1982
gPROMS	Chemical Engineering	Imperial C	1994
Modelica	General Purpose	Modelica Ass	1996

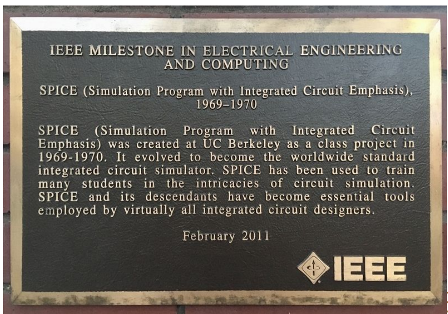
51

Electronics

- Electric circuits: multi-port systems
- Component based: resistor, capacitor, inductor, transistor. Model components and connect them.
- Nagel, Peterson Berkeley 1970, SPICE2 1975, …
- FORTRAN
- Kirchoff's voltage and current laws
- Node equations
- Nonlinear equations - Tearing
- Differential algebraic equations DAE, W. Gear
- Important part of Tool Chain for VLSI design
- IEEE Milestone 2011 Cory Hall Berkeley

52

SPICE IEEE Milestone

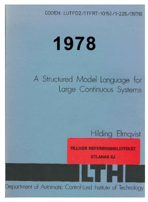


53

Dymola

- SPICE Electrical Circuits – Physics much richer
- DAE Solvers Gear Petzold
- SPEEDUP Chemical Engineering
- Multibody Mechanical Adams
- Dymola Elmqvist LTH BNF
  - Equation based
  - Object orientation (Simula Dahl-Nygaard)
  - Extensive symbolic computing
  - Boiler-turbine test case
- Great idea but premature due to limitations of computers and software

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



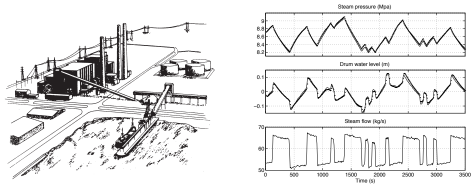
www.control.lth.se/Publication/elm78dis.html

54

Simula IEEE Milestone

55

Boiler Control Project



- Experiments, modeling, system identification
- Eklund Drum Boiler-Turbine Models 1971
- Lindahl Design and Simulation of a Coordinated Drum Boiler-Turbine Controller Dec 1976 (Simnon) 80 pages. Dymola Elmqvist 8 pages!)

56

Features

- Simula – Only object programming language available
- Real language syntax in Backus-Naur form
- About 128 kbyte (18bit word) on Univac-1108
- Later translated to Pascal for PDP Vax computer
- Boiler model coded in 8 pages
- 250 equations
- 11 systems of equations largest block 17 equations

57

Index - Linear DAE

$$E\frac{dx}{dt} = Ax + b(t)$$

DAE regular if the matrix pencil $\lambda E - A$ is regular for all λ. A regular matrix pencil can be transformed to the Weierstrass-Kronecker normal form

$$P(\lambda E - A)Q = \lambda\begin{pmatrix} I & 0 \\ 0 & N \end{pmatrix} + \begin{pmatrix} C & 0 \\ 0 & I \end{pmatrix}$$

where C is on Jordan form and $N = \mathrm{diag}(N_1, N_2, ..., N_k)$ is a block diagonal matrix where all elements are nonzero except the super-diagonal which are 1.

58

Linear ODE - Index

$$\frac{dx}{dt} = Cx + \xi(t)$$

$$N\frac{dy}{dt} = y + \eta(t)$$

The smallest integer m such that $N^m = 0$ is called the differentiation index of the ODE

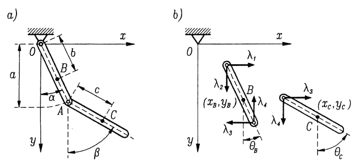
$$x = e^{Ct}x(0) + \int_0^t e^{A(t-s)}\xi(s)ds$$

$$y = -\eta(t) - N\frac{d\eta(t)}{dt} - \ldots - N^{m-1}\frac{d^{m-1}\eta(t)}{dt^{m-1}}$$

Index tells how many times input is differentiated

59

Computations



Eliminate trivial equation

Reduction to BLT (Block lower triangular form)

Analytic solution of linear equations

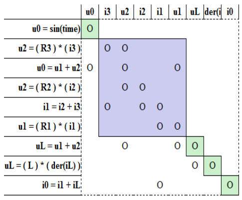
Solve nonlinear equations (Newton, Tearing)

Integrate DAEs (index reduction)

60

Tarjan's Algorithm

- Represent equations and variables as a directed graph
- Algorithm gives strongly connected components
- Transfrom to BLT (Block LowerTriangular) form
- Tarjan's algorithm



61

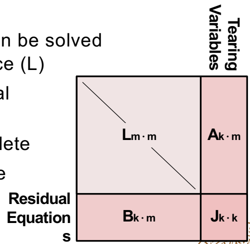
Kron's Tearing

- Solving large nonlinear equations
- Gabriel Kron: Diakoptics – The Piecewise Solution of Large-Scale Systems Electrical Journal London 1957-59
- Reduce a large equation to a set of smaller equations
- Select tearing variables and a corresponding set of equations whose errors are called residuals.
- Iterate until residuals are sufficiently small

62

Tearing

- Transform BLT-blocks (algebraic loops) into bordered triangular matrices
- Exploit that many equations can be solved analytically – diagonal incidence (L)
- Only a small number of residual equation remains (J)
- Finding smallest k is NP-complete
- Intuition: if tearing variables are known, L-part can be solved analytically in sequence
- Exploit physics, heuristics

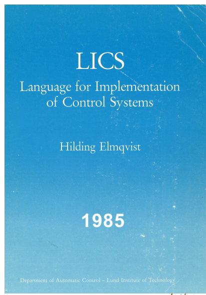


63

63

What happened to Hilding?

- Postdoc at Stanford John L. Hennessy 1978-1979
- Lund 1979-1983 The LICS project - Language for Implementing Control Systems – DCS system
- Satt and SattLine 1984-1992
- ABB aquired Satt and Sattline
- Founded Dynasim 1992
- ABBs DCS system stiil reccognize as top-of-the-line



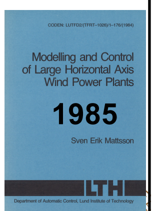
64

1 Introduction
2 Analog Computing
3 Block Diagram Modeling
4 Equation Based Modeling
5 Modelica
6 Summary

65

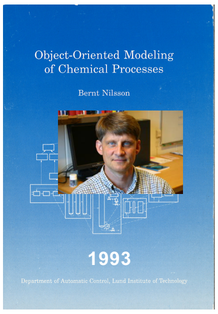
Omola-Omsim

- The CACE project was terminated in early 1980s because of software, FORTRAN and MATLAB
- New research project 1985 Object Oriented Modeling and Simulation: Sven Erik Mattsson, Mats Andersson, Bernt Nilsson, Dag Bruck
- Experimentation in Lisp & KEE
- C++ (standard 1985) was chosen
- Language (Omola) and simulator (OmSim)
- Refining symbolic manipulations (SEM)
- Index reduction for DAE
- Team merged with Dynasim in 2000

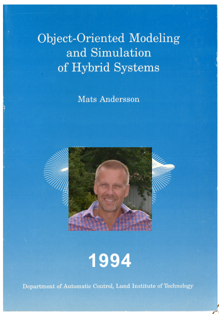


66

Bernt Nilsson & Mats Andersson



Prof Kemiteknik



Volvo

67

Dynasim 1992

- Founded by Hilding Elmqvist in Lund 1992 to develop the Dymola language
- Collaboration with LTH and DLR Martin Otter
- Extensively used in design of Toyota Prius from 1996
- Migration of researchers from LU to Dynasim: Dag Bruck, Sven Erik Mattson, …
- Acquired by Dassault Systèmes 2006
- Integration with Catia
- Synchronos extensions (real time computing) Hilding, French school Albert Benveniste, Marc Pouzet (kolla)

68

Value of Collaboration

- Wide range of physics and engineering: EE, ME, ChemE, BioEng, …
- Control and optimization
- Computer Science
- Numerical Mathematics
- Software
- Many tools are required
- Very strong incentive for collaboration

69

Modelica

- Lund University (Omola, Omsim) & Dynasim (Dymola)
- Discussions of future development
- ESPRIT Simulation in Europe, Lund Sept 2-6, 1996
- COSY meeting Lund Sept 5-7, 1996
- 23 participants from European groups: Dynasim and LTH Lund, ETH Zurich, INRIA Paris, DLR Munich, VTT Helsinki, Imperial College London, RWTH Aachen, universities of Barcelona, Groningen, Valencia, Wien
- Formation of the Modelica language group: define language
- First Modelica language specification Sept 1997
- 100th design meeting Lund 2019

70

Original Language Team

Martin Otter DLR, Alexandre Jeandel Gaz de France, Per Sahlin, Brisdata/Equa, Sven Erik Mattsson LTH, Bernt Nilsson LTH, Hilding Elmqvist Dynasim

71



Control Engineering Practice 6 (1998) 501–510

Physical system modeling with Modelica

Sven Erik Mattsson,[a,*] Hilding Elmqvist,[b] Martin Otter[c]

[a]Dept. of Automatic Control, Lund University, Box 118, SE-221 00 Lund, Sweden
[b]Dynasim AB, Research Park Ideon, SE-223 70 Lund, Sweden
[c]DLR Oberpfaffenhofen, D-82230 Wessling, Germany

Received 8 October 1997; accepted 9 February 1998

72

MODELICA

- **Modelica**® is a non-proprietary, object-oriented, equation based language for modeling complex physical systems
- **The Modelica Association** is a non-profit, non-governmental organization with the aim of developing and promoting the Modelica modeling language https://www.modelica.org
- **Research** projects within Europe spend more than 100 M€ in the period 2007-2022 to further improve the Modelica Language, Modelica Libraries, FMI, DCP and related technologies

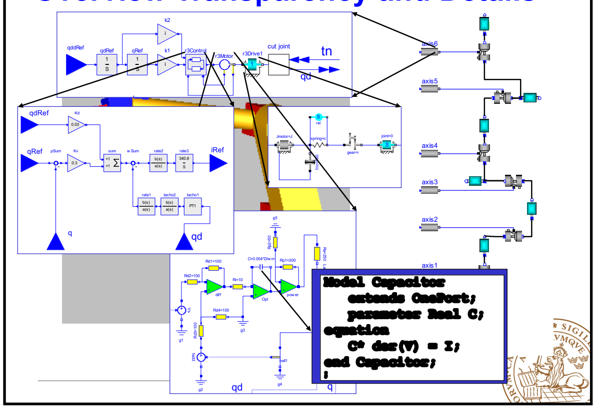
73

MODELICA

- 14th Modelica Conference Linköping 2021,
- 100th Modelica design meeting Lund 2019
- Modelica standard library
- Modelica Simulation Environments: CATIA Systems,CyModelica, Dymola, LMS AMESim, JModelica.org, MapleSim, OpenModelica, SCICOS, Simulation X, Vertex and Wolfram SystemModeler
- Home page www.modelica.org

74

Overview Transparency and Details



75

Animation

See videos
YouBot Ball Bounce.mp4
YouBotsGripping.mp4

76

FMI and FMU

The Functional Mock-up Interface (FMI) is a free standard that defines a container and an interface to exchange dynamic models using a combination of XML files, binaries and C code zipped into a single file. It is supported by 100+ tools and maintained as a Modelica Association
A component which implements the interface is called a Functional Mockup Unit (FMU)

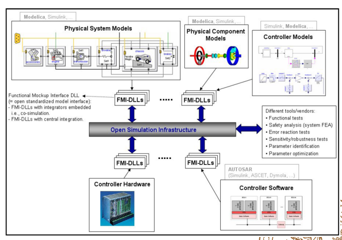
- Model description in XML
- Implementation in source or binary
- FMU can also include simulation code

77

Combining Tools

modelisar

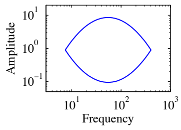
Functional mockup interface FMI



78

Validity ranges – A missing element

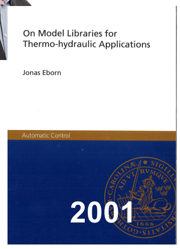
- Extend alarm bell in analog computers
- The uncertainty lemon Gille, Pelegrin, Decaulne 1959

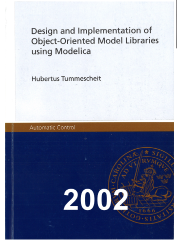


- Amplitude ranges easy
- Frequency ranges by using dy/dt=w*y

79

Eborn & Tummescheit

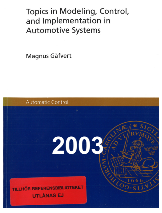




Scynamics HB 2001 Consulting based on Dymola & Material properties
Founders of Modelon AB 2004

80

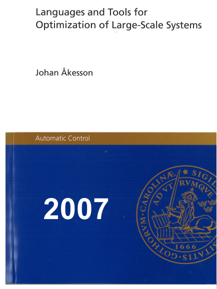
Magnus Gäfvert



Founder of Modelon 2004

81

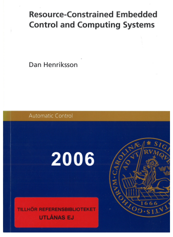
Johan Åkesson



Compiler technology
Optimization, Optimica
LTH and Modelon 2007

82

Dan Henriksson
Toivo Henningsson



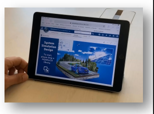
Dassault 2007 | Modelon 2013-16, Axis, Ericsson

83

Dynasim | Modelica Association | fmi | DASSAULT SYSTEMES

Dynasim AB was founded in 1992 by Hilding Elmqvist.

- The first commercial success of Dymola was the development of the hybrid drivetrain of the Toyota Prius.
- Dymola is still the premier Modelica (Gold Standard) tool, with a complete development and simulation environment, including support for FMI, and the largest selection of available libraries of all Modelica tools.
- Widely used in industry, first commercial success-story was the development of the hybrid drivetrain of the Toyota Prius.
- Acquired by Dassault Systèmes in 2006.



84

DASSAULT SYSTEMES

a **Scientific** company

Combining **Science, Technology** and **Art** for a sustainable society

15,000 passionate people

- 124 nationalities / 179 sites
- One global R&D / 64 labs
- Game changing **3D**EXPERIENCE solutions

210,000 enterprise customers

- 12 industries in 140 countries
- 25 million users

12,600 partners

- Software, Technology & Architecture
- Content & Online Services
- Sales
- Consulting & System Integrators
- Education
- Research

Long-term driven

- Majority shareholder control
- **Revenue**: €3,065.6 million*

* Figures as of FY 2016 / Non-IFRS

85

Modelon

System simulation without boundaries

- Simulation software solutions and expert services for model-based systems engineering from product concept to operation
- Dedicated to open technologies and standards (Modelica, FMI)
- Serving leading technology enterprises across multiple industries worldwide
- Founded 2004 in Lund as PhD spin-off from Lund University Hubertus Tummescheit, Hilding Elmqvist, Jonas Eborn, Johan Andreasson KTH, Magnus Gäfvert, Johan Åkesson creator of the OPTIMICA Compiler Toolkit (joined later)
- About 100 MSEK turnover, 100 employees (20% PhD) in 10 offices in Europe, North America, Asia – 15 years of profitable growth

86

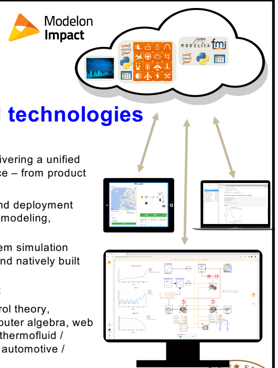
Modelon

Modelon History

- A company built upon research in system simulation and optimization at LTH and KTH started in 2004
- Hubertus Tummescheit and Jonas Eborn 2005 – Modelica libraries for air conditioning in cars (Dymola)
- Johan Andreasson and Magnus Gafvert 2006 – Modelica libraries for vehicle dynamics for Dymola
- Johan Åkesson 2008 – Compilers and optimization with JModelica.org (open source)
- Compiler and solver platform OPTIMICA Compiler Toolkit 2013
- Offices in Germany, USA, Japan and India 2008-
- Modelica library portfolio growth with 18+ libraries 2004-
- ANSYS integrates the OPTIMICA Compiler Toolkit in system simulation tool Twin Builder (2015)
- Cloud-based platform for modeling and simulation – Modelon Impact 2020
- IPO Nasdaq First North 2021

87

Modelon



Modelon products and technologies

- Modelon products enable companies to focus on delivering a unified picture of product system interaction and performance – from product concept to operation
- Modelon's 18+ model libraries, simulation solvers, and deployment solutions are leading products for industrial systems modeling, simulation and optimization
- Modelon Impact is a novel and disruptive SaaS system simulation platform based on Modelon open technology stack and natively built for web/cloud
- Modelon solutions based on unique mix of expertise:
  - Physical modeling, system dynamics and control theory, numerics & solvers, compiler technology, computer algebra, web & cloud front-end and back-end development, thermofluid / mechanical / chemical / electrical engineering, automotive / aerospace / energy industry expertise
- Modelon has been a leading contributor to Modelica and FMI open standards

88

Mogram & Modia

- Mogram founded by Hilding Elmqvist 2016 in Lund
- Modia is an experimental implementation of the modeling language Modlica created by Elmqvist, Mogram AB, Toivo Henningsson, Lund (earlier) and Martin Otter, DLR Oberpfaffenhofen

  https://github.com/ModiaSim/Modia.jl
- Modia is based on based on Julia, a high-level, high-performance, dynamic programming language

https://en.wikipedia.org/wiki/Julia_(programming_language)

89

Julia Computing

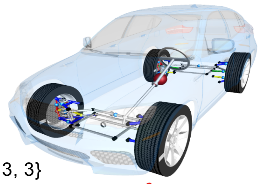
JuliaSim is a next generation cloud-based simulation platform, combining the latest techniques in SciML with equation-based digital twin modeling and simulation. Our modern ML-based techniques accelerate simulation by up to 500x, changing the paradigm of what is possible with computational design.

https://juliacomputing.com

90

Vehicle Simulation

Detailed vehichle model with chassi and power train
Original model:
Number of components: 6863
Variables: 53495
Differentiated variables: 1683



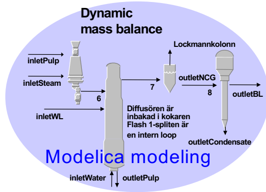
After reduction:
States: 330
Largest nonlinear systems: {3, 3, 3, 3}

100 x real time on Xeon E5649 with 4 cores

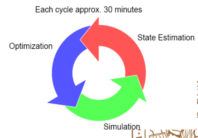
Slide courtesy of Johan Andreasson Modelon AB

91

Mill Wide Control



25	**Production units**
38	**Buffer tanks**
250	**Streams**
250	**Measurements**
2500	**Variables**

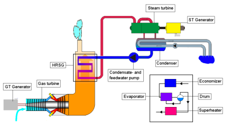


Slide courtesy of Alf Isaksson ABB

92

Combined Cycle Power Plant

Alexandra Lind and Elin Sällberg
Ms Thesis on Start-up Optimization LTH, Siemens, Modelon
www.control.lth.se/Publication/5900.html

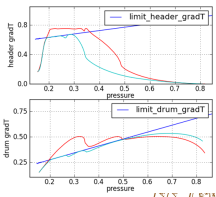


$$F(x(t), \dot{z}(t), y(t), u(t), t) = 0$$

$$\min_{u(t)} \int_{t_0}^{t_f} g(x(t), y(t), u(t))dt$$

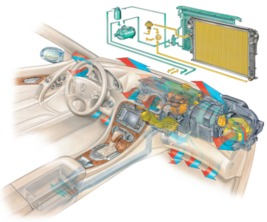
$$z^L \leq z(t) \leq z^U$$
$$y^L \leq y(t) \leq y^U$$
$$u^U \leq u(t) \leq u^U$$

Continuous states: 28
Scalar eq: 389
Algebraic eq: 361
Discrete NLP eq: 18772



93

Automotive Climate Control



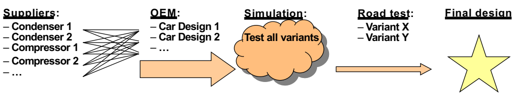
- *Audi, BMW, DaimlerCrysler, Volkswagen and their suppliers have standardized on Modelica*
- *Suppliers provide components and validated Modelica models based on the AirConditioning library from Modelon*
- *Car manufacturers evaluate complete system by simulation*
- *IP protected by extensive encryption*

Picture courtesy of Behr GmbH & Co.

94

Customer Value

- Automotive OEM customers:
  Daimler, BMW, AUDI, VW, Ford, Volvo, …
- Automotive supplier customers:
  Visteon, Valeo, Denso, Behrgroup, Modine, Showa Denko
- Calibrated models by suppliers mandatory in bid for hardware contract
- Substantial reduction of road & climate chamber testing



Successful model-based development process based on exchange of component and system models between suppliers and OEM

Slide from Modelon

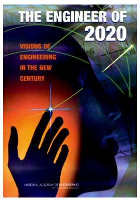
95

1 Introduction
2 Analog Computing
3 Block Diagram Modeling
4 Equation Based Modeling
5 Modelica
6 Summary

96

Modeling is Important

There will be growth in areas of simulation and modeling around the creation of new engineering "structures". Computer-based design-build engineering ... will become the norm for most product designs, accelerating the creation of complex structures for which multiple subsystems combine to form a final product.



NAE The Engineer of 2020

97

Drawings and Models

- Drawings show what systems look like
- Models show how systems behave
- Equally important
- Should be handled in the same way with respect to creation,development, documentation and intellectual property

98

Modeling

Solomon Golomb: Mathematical models – Uses and limitations. IEEE Transaction on Reliability, Aug. 1971, pp 130-131

Solomon Wolf Golomb (1932-2016) mathematician, engineer, professor of electrical engineering at the University of Southern California. Best known to the general public and fans of mathematical games as the inventor of polyominoes, the inspiration for the computer game Tetris. He has specialized in problems of combinatorial analysis, number theory, coding theory and communications.

99

Golomb On Modeling

- Don't apply a model until you understand the simplifying assumptions on which it is based and can test their applicability. Validity ranges
- Distinguish at all times between the model and the real world. You will never strike oil by drilling through the map
- Don't expect that by having named a demon you have destroyed him. Singularity, badly conditioned
- The purpose of notation and terminology should be to enhance insight and facilitate computation – not to impress or confuse the uninitiated. Gobbledygook

100

Lectures

- Introduction
- Modeling Methodology
- Discrete Systems (Årzen)
- DAE
- Modelica
- Circuit Theory (Pates)
- Power Systems (Samuelsson)
- Mechanical Systems
- Friction Models
- Bicycles
- Cars (Olofsson)
- Ships & Aircrafts
- Thermo Fluids
- A Thermal Boiler
- Compartment Models
- Neurons

101