Process Modeling - A PhD Course

Nilsson, Bernt

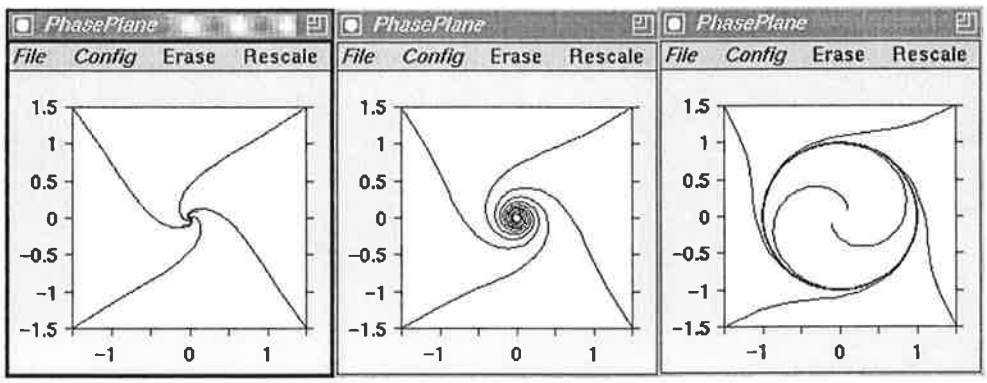
1995

Document Version:
Publisher's PDF, also known as Version of record

ISSN 0280–5316
ISRN LUTFD2/TFRT--7536--SE

Process Modelling — a PhD course

Bernt Nilsson



Department of Automatic Control
Lund Institute of Technology
June 1995

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
TECHNICAL REPORT

Date of issue
June 1995

Document Number
ISRN LUTFD2/TFRT--7536--SE

Author(s)
Bernt Nilsson

Supervisor

Sponsoring organisation

Title and subtitle
Process Modelling – a PhD course

Abstract

This report contains a set of lecture notes of a PhD course in process modelling for primary PhD students in Chemical Engineering. The course gives an overview and covers following areas: physical modelling, computer-aided modelling, system modelling, linear systems, nonlinear systems, distributed parameter systems, discrete time systems, system identification, simulation and model approximation.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280–5316

ISBN

Language
English

Number of pages
79

Recipient's notes

Security classification

Process Modelling

PhD course at Department of Chemistry and Chemical Engineering
Spring 1995

Aim
The aim with the course is to give an overview on how models are created, developed and used and how modelling relates to other subjects in science. The course focus up on modelling based on first principles, and the properties of different classes of models, and modern computer tools.

Course plan
Lectures (Wednesday 13-15, K:L (except for 26/4)):

1. *Models and Modelling*, **moved to 25/4 13-15**
   - Model types,
   - Model use,
   - The modelling process,
   - Models, problem formulations and problem solving,
2. *Physical Modelling*, **OBS! Wednesday 3/5 10-12!**
   - Laws of physics, conservation laws,
   - Mass, energy and momentum balances,
   - Lumped and distributed parameter descriptions,
   - System descriptions.
   - *To read: Luyben, part I, (chap 2 and 3).*
   - Lecture problems
3. *Computer Aided Modelling*, **3/5**
   - CSSL language
   - Graphical modelling,
   - Model structuring concepts,
   - Object-oriented modelling.
   - Differential-algebraic equation system.
   - *To read: Ljung & Glad, chap 6 (Bond graphs).*
   - Lecture problems
4. *Linear and Nonlinear Models*, **10/5**
   - Systems of ODE,
   - Eigenvalues and eigenvectors,
   - Phase plane analysis,
   - Linearization.
   - *To read: Strang, chap 6.1 (ODE) and 6.2 (p492-501).*
   - Lecture problems
5. *Nonlinear Models and Distributed Models*, **17/5**
   - Periodic solutions, limit cycles and strang attractors,
   - Bifurcation and chaos,
   - Classification of PDE:s,
   - Diffusion type PDE:s,
   - Wave equations.
   - *To read: Strang, chap 6.2(p 502-510) and 6.4.*
   - Lecture problems
6. *Discrete Time Models and Identification*, **24/5**
   - Difference equations,
   - Eigenvalues and stability,
   - Parameter estimation,
   - Linear regression,
   - System identification.
   - State estimation (reconstruction),
   - *To read: Ljung & Glad, chap 9 and 10.*
   - Lecture problems
7. *Simulation and Model Approximation*, **31/5**
   - Algebraic equation systems, sparce matrices.

- Continuous time simulation, initial-value problem.
- Accuracy, stability, complexity and stiffness,
- Discretization in space,
- Finite difference and finite element methods,
- Methods of lines.
- *To read: Strang, chap 6.5.*
- Lecture problems

8. *Project Presentations*, **7/6**

Guest-Lecture: (Wednesday 15.30-16.30, K:L, *after the lecture*)

24/5; **Hilding Elmqvist**, Dynasim
Elmqvist developed the simulation language Simnon during the early seventies. His thesis is one of the most important contributions in computer aided modelling and the Dymola language is described. During the eighties Elmqvist was one of the project leader for the SattLine development at SattControl (graphic based real time control system). Now Elmqvist has his own company at Ideon for development of simulation software.

31/5; **Magnus Pettersson**, KAT
Magnus is going to talk about his experiences of developing dynamic model of a wet system for flue gas cleaning (master thesis at ABB Fläkt). Magnus used MATLAB/SimuLink. The model contains static scrubber model, dynamic buffer tank and pH calculations of a complex mixture.

31/5; **Michael Grimsberg**, Kem. Tekn.
Michael makes a demonstration of **DIFFPAR**, a MATLAB toolbox for parameter estimation in continuous time models.

Exercises: (Thursday 10-12, Lutetia (except for 29/5))

1. Physical Modelling. **4/5**
   - Set up models of different phenomena and complexity,
   - Case study: a batch reactor.
   - Simulation in MATLAB and SIMULINK.
2. Linear Models and Analysis.**11/5**
   - Set up linear models,
   - Linear analysis (Matlab).
   - Phase plane analysis (Matlab).
3. Nonlinear Models and Analysis. **18/5**
   - Set up nonlinear models,
   - Linearization and transient responses (Matlab),
   - Multiple steady states.
   - Phase plane analysis (Matlab).
4. System Identification. **29/5**
   - Discrete time models.
   - Stochastic discrete time models.
   - Identification using System Identification Toolbox in Matlab.
5. Project exercise. **1/6**
   - Your own work on your hand-in problem.

Your-own-problem:

1. Define a modelling problem in your own research area. Use the course material that are useful for the problem.
2. Write a short report.
3. Make a short presentation of your work. **7/6**

Literature:

1. Luyben, W.; *Process Modeling, Simulation and Control for Chemical Engineers,* Part I. (lecture 1-2)
2. Strang, G.; *Introduction to Applied Mathematics,* chap. 6, (lecture 4-5,7)
3. Ljung, L. and T. Glad; *Modellbygge och simulering* (Modeling and Simulation), chap. 6 and 9-10, (lecture 3 and 6)

Examination:

1. *Passive:* the lectures and one-day take-home-exam (alt. hand-in on the lecture problems) (3 marks).
2. *Active:* 1 **and** short report (or presentation) on your-own-problem (alt. report on exercise) (5 marks).

Exam:

1. One-day take-home-exam is possible to do during week 24 (June 8-16).

Process Modelling ++: *In the case of interest, a continuation of the course is possible. The course will then be a set of seminars with active students. Examples on seminars:*

- *Tensor based models (Bird, Stewart and Lightfoot).*
- *PDE approximations.*
- *Bond graph theory. (Cellier)*
- *Object-oriented modelling theory. (Marquardt)*

Bernt Nilsson
Dept. of Automatic Control
Lund Institute of Technology
Box 118, 221 00 LUND, Sweden

Hämtställe: 9
phone: +46 46 108784, fax: +46 46 138118
E-mail: `bernt@control.lth.se`
URL: `http://www.control.lth.se/~bernt`

This page has the following URL:

`http://www.control.lth.se/~bernt/pmcourse.html`

Last update: June 7, 1995.
Bernt Nilsson
bernt@control.lth.se

PROCESS MODELLING

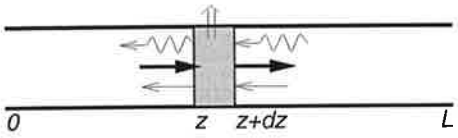
PhD course 1995

Bernt Nilsson
Automatic Control, LTH

PROCESS MODELLING

Physical Modelling

Models based on the application of physical and chemical laws on the system being studied.



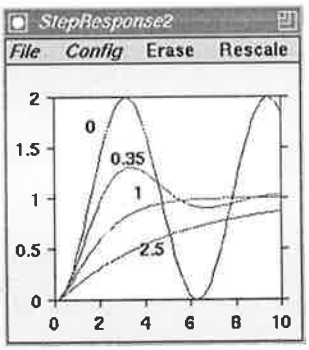
$$\frac{\partial c}{\partial t} = -\frac{\partial (vc)}{\partial z} + D\frac{\partial^2 c}{\partial z^2} + r$$

$$C_p \frac{\partial (\rho T)}{\partial t} = -C_p \frac{\partial (\rho v T)}{\partial z} + \frac{k_T}{A}\frac{\partial^2 T}{\partial z^2} + Q_p - \frac{2}{r_r} h_T (T - T_0)$$

PROCESS MODELLING

System Modelling

Models described in a mathematical framework capturing the system behaviour.



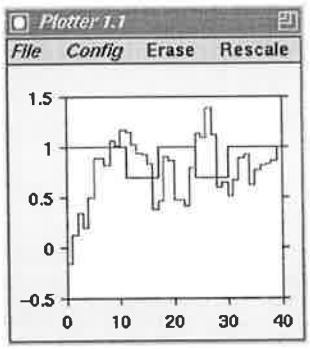
$$\frac{dx}{dt} = \begin{bmatrix} -2\xi\omega & 1 \\ -\omega^2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

PROCESS MODELLING

Identification

Models that are fit to measurment data

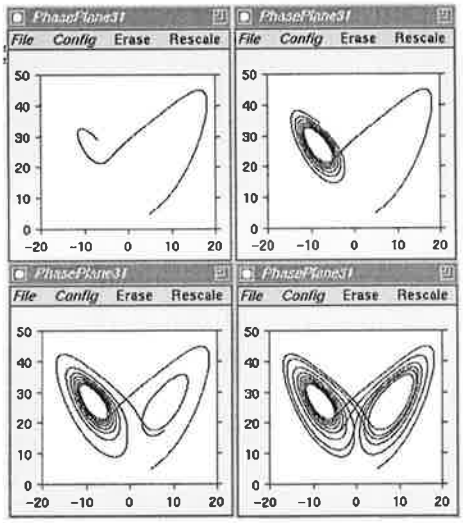


$$y_{n+3} + a_1 y_{n+2} + a_2 y_{n+1} + a_3 y_n = b_1 u_{n+1} + b_2 u_n + c_1 e_n$$

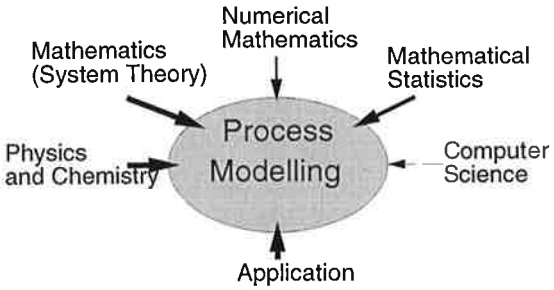
PROCESS MODELLING

Simulation

Models are approximated to generate a numerical solution.



Process Modelling



Courses at LTH:

- Physical Modelling ?
- System Modelling (Matematik)
  - Linjära system,
  - Kontinuerliga system,
  - Olinjära system (FK),
- Identification
  - Tidsserieanalys,
  - Processidentifiering,
- Simulation
  - Numerisk analys (AK och FK),
  - SAM: Simulering.

PROCESS MODELLING

Content

Lectures:

- Overview
  1. Models and Modelling
- Physical Modelling
  2. Physical Modelling
  3. Computer Aided Modelling
- System Modelling
  4. Linear and Nonlinear Models
  5. Nonlinear and Distributed Models
- Identification
  6. Discrete Models and Estimation
- Simulation
  7. Methods and Approximations.

PROCESS MODELLING

Content

Active part:

- Your-own-problem
  - Define your own problem,
  - Apply relevant parts of the couse
- Exersice-problem
  1. Physical Modelling
  2. Linear Analysis
  3. Nonlinear Analysis
  4. Identification
  5. Extra time

PROCESS MODELLING

Examination

- Lectures (passive part):
  - Take-home-exam **or**
  - Hand-in-problems.
- Active part:
  - Short report **or**
  - Short presentation.

PROCESS MODELLING

Information

- Literature
  - Luyben; part I (chap 1-3)
    Process Modeling, Simulation and Control for Chemical Engineers.
  - Strang; chap 6
    Introduction to Applied Mathematics.
  - Ljung, Glad; chap 6, 9 and 10
    Modellbygge och simulering.
- Communication
  - Info on WWW (netscape)
  - email
- Dates
  - lectures; wednesday 13-25,
    except 26/4 moved to 24/4 (same time)
  - exercises; thursday 10-12,
    except 25/5 moved to 29/5 (same time)
  - presentations; 7/6
  - exam; week 23.

PROCESS MODELLING

lecture I

Models

and Modelling

What is a model?

A model (M) for a system (S) and an experiment (E) is anything to which E can be applied in order to answer questions about S.

Marvin Minsky 1965

Model Types I

Types of Models:

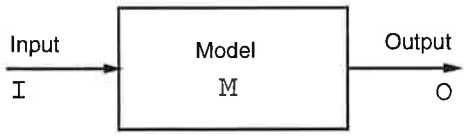
- Intuitive
- Verbal
- Causal
- Qualitative
- Quantitative

Model Types II

Quantitative Model Types

- Static vs. Dynamic
  - transients
- Lumped vs Distributed
  - space description
- Deterministic vs. Stochastic
  - noice
- Continuous vs Discrete
  - sampling or events
- Linear vs Nonlinear
  - qualitative behaviour
- Black Box vs State Space
  - internal behaviour
- Time vs Frequency
  - time scale

What is a model used for?



Models are used i different problem formulations:

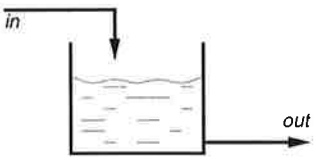
- **Direct**: apply I on M, study O.
- **Inverse**: apply O on M, study I.

If I and O are known:

- **Identification**: find structure and parameters in M.
- **Estimation**: find internal states in M (if internal structure of M is known).
- **Design**: study parameters in M (if structure and internal states are known)

Model Use I

Example: CSTR



Continuous Stirred Tank Reactor with simple first order reaction kinetics.

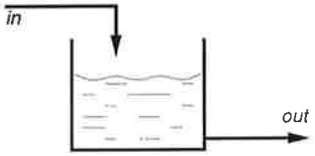
- Assume isothermic and isobaric conditions.
- Flow and volume are also assumed to be constant.

Model:

In + Production	=Out	+Accumulation
$qc_{in} + (-kc)V$	$=qc$	$+\frac{d(Vc)}{dt}$

Model Use II

Example: CSTR



Problem formulation I: the Direct problem (apply I on M).

Dynamic behaviour a):

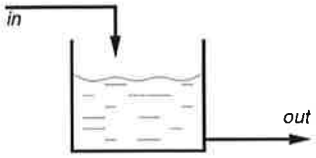
$$\frac{dc}{dt} = \frac{q}{V}(c_{in} - c) - kc$$
$$c(0) = c_0$$

Static behaviour:

$$c = \frac{q}{q + kV} c_{in}$$

Model Use III

Example: CSTR



Problem formulation II: the Inverse problem (apply O on M).

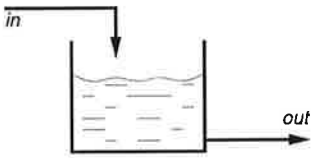
Inverse problem:

$$c_{in}(t) = (1 + \frac{V}{q}k)c(t) + \frac{V}{q}\frac{dc(t)}{dt}$$

- Dynamic inverse problem needs differentiation of $c(t)$.
- Static inverse problem $c_{in} = (1 + \frac{V}{q}k)c$.

Model Use IV

Example: CSTR



Problem formulation V: the Design problem (apply O and I on M).

Design problem a: (V is unknown in M)

$$V = \frac{q}{k}(\frac{c_{in}}{c} - 1)$$

Optimiztion problem b: (V and q are unknown in M)

$$\frac{V}{q} = \frac{1}{k}(\frac{c_{in}}{c} - 1)$$
$$F_{loss} = f(V, q)$$

- three unknowns: V, q and F_{loss}.
- two equations.
- select V and q that minimize (maximize) F_{loss}.

Model Use V

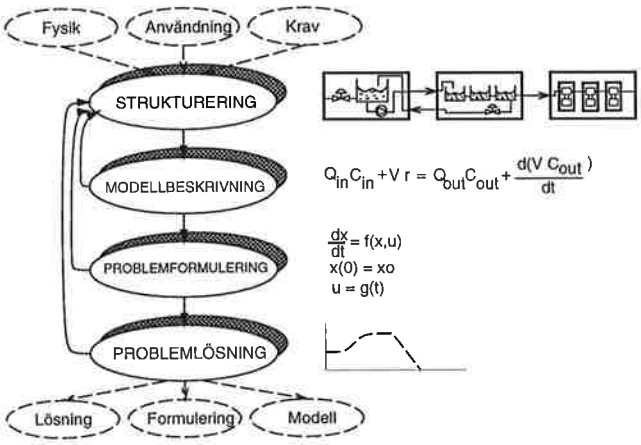
Conclusions:

- Model is invariant.
- Models are transformed (and manipulated) to fit problem formulations.
- Problem formulation is the input to the problem solving tool.
- Problem solving often approximate the mathematical PF.

Demands on the Model

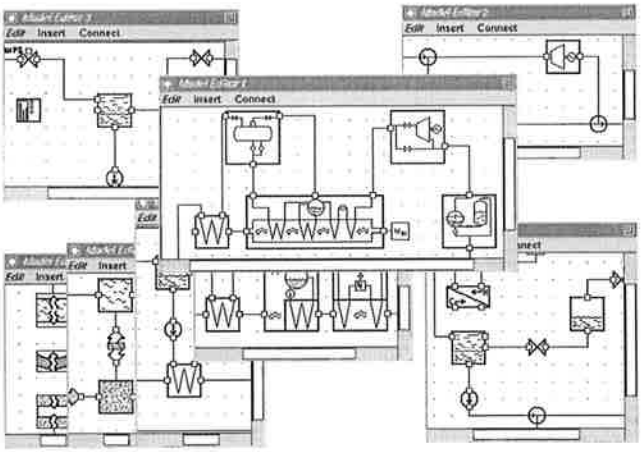
- accuracy,
  - no more and no less.
  - quantitative and qualitative.
  - (**not** the same as tool accurcy)
- validity,
  - range,
  - operation conditions,
  - transient operation,
  - internal properties.
- complexity,
  - simple (macroscopic),
  - detailed (microscopic),
  - phenomena oriented.

Modelling Process I



Modelling Process II

Structuring



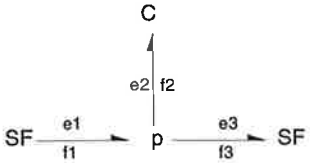
Modelling Process III

Model Description

- Mathematical equations

$$q_{in}c_{in} + Vr = q_{out}c + \frac{(Vc)}{dt}$$

- Graphical formulations



- Modelling assistent
  - menu-driven
  - application oriented description

Modelling Process IV

Problem Formulation

- Simulation
  - Direct dynamic (integration)
  - Invers dynamic (derivation)
  - Static (equation solving)
- Analysis
  - Characteristics (calculations)
  - Graphical characteristics (plots)
- Design
  - Parameter simulation (= static simulation)
  - Optimization

Modelling Process V

Problem Solving

Simulation

- *Programming*

  PF and PS are integrated
- *CSSL language* (Simnon)

  PF and PS are separated

  PF is the description
- *CSSL with graphics and libraries* (SimuLink)

  Structuring of PFs

  Reuse of predefined PF
- *Object-Oriented Modelling Language* (Dymola)

  Equation based models

  Reuse of models

  Automatic manipulation to PF and PS

Model Developers and Users

- *System models*
  - handle the complexity
  - use of predefined unit models
- *Unit models*
  - special designed model
  - for a purpose
- *Phenomena models*
  - small scale model
  - "general models"

Conclusions

- Models are invariant
- Different model types
- Choice of model
  - accurcy
  - validity
  - complexity
- Modelling process
  - structuring
  - model description
  - problem formulation
  - problem solving

PROCESS MODELLING

lecture II

Physical Modelling

PROCESS MODELLING

Content

- Principles of Physical Modelling
- Continuity Equations
  - Mass Balances
  - Energy Balances
  - Momentum Balances
  - Mechanical Energy Balances
- Transport Phenomena and Reaction Kinetics
- System Descriptions and Problem Formulations

Physical Modelling

Principles of Formulation

Basis: models are based on fundamental physical and chemical laws.

Assumptions: (engineering compromises)

- carefully considered and listed.
- used to tune accuracy, validity and complexity of models.

Consistency: (model verification)

- degree-of-freedom ($n_{variables} = n_{equations}$).
- check units and dimensions.

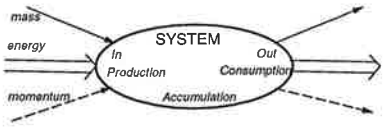
Solution: techniques and tools demand problem formulations on particular forms (models + manipulations).

Verifications: test the model against data (model validation).

Fundamental Laws

1. Continuity equations:
   - mass balances,
   - energy balances,
   - momentum balances.
2. Transport phenomena:
   - mass transport,
   - energy transport,
   - momentum transport.
3. Equilibrium descriptions:
   - phase equilibrium,
   - chemical equilibrium.
4. Kinetic descriptions.
5. State equations (or relations).

Continuity Equations I



Dynamic balances over a system:

$$Acc = In - Out + Prod - Cons$$

Balances:

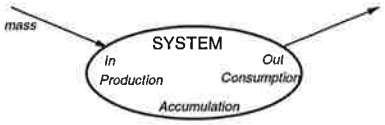
- mass (chemical components),
- energy and
- momentum.

Or as "Kemisk Teknologi" would like to express it

$$In + Prod/Cons = Out + Acc$$

Continuity Equations II

Mass Balances



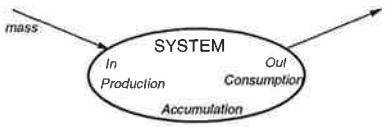
Dynamic total mass balances over a system:

$$\begin{aligned} Acc &= In - Out + Prod/Cons \\ \frac{dm}{dt} &= w_{in} - w_{out} + r \end{aligned}$$

- m is the total mass in the system.
- w are the mass flows entering and leaving.
- r is the mass production. In our applications $r = 0$ in the total mass balance.

Continuity Equations III

Component Mass Balances



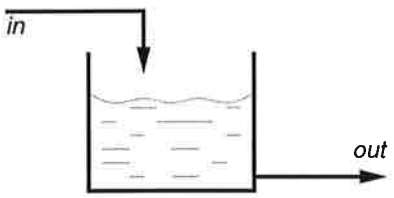
Dynamic mass balance of component i over the system: *(mole balance instead of mass balance $n_i = \frac{m_i}{M_i}$)*

$$\begin{aligned} Acc &= In - Out + Prod \\ \frac{dn_i}{dt} &= \omega_{i_{in}} - \omega_{i_{out}} + r_i \end{aligned}$$

- n_i is the number of mole of species i in the system.
- ω_i is the mole flow rate of species i.
- r_i is the mole production (consumption) rate of species i.

Mass Balances I

Tank Example



Total mass balance over a simple tank. *Assumptions:*

- constant density, $m = \rho V$, $w = \rho q$.
- cylidrical geometry, $V = Ah$.

$$\begin{aligned} Acc &= In - Out \\ \frac{dm}{dt} &= w_{in} - w_{out} \\ \frac{d(\rho A h)}{dt} &= \rho q_{in} - \rho q_{out} \\ \frac{dh}{dt} &= \frac{1}{A}(q_{in} - q_{out}) \end{aligned}$$

Mass Balances II

Tank Example cont.

Component mole (mass) balance over a simple tank. *Assumptions:*

- no production/consumption, $r_i = 0$.
- homogeneous mixing, $c_i = c_{i_{out}}$
- constant volume, $q_{in} = q_{out}$.

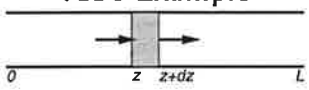
$$\begin{aligned} Acc &= In - Out + Prod \\ \frac{dn_i}{dt} &= \omega_{i_{in}} - \omega_{i_{out}} + r_i \\ \frac{d(Vc_i)}{dt} &= q_{in}c_{i_{in}} - q_{out}c_i + 0 \\ \frac{dc_i}{dt} &= \frac{q}{V}(c_{i_{in}} - c_i) \end{aligned}$$

or direct on vector form $c = [c_1 \quad c_2 \ldots c_n]^T$,

$$\frac{dc}{dt} = \frac{q}{V}(c_{in} - c)$$

Mass Balances III

Tube Example



Total mass balance over a volume element in a tube. *Assumptions:*

- no mass production.
- constant cross area, $V = Adz$, $q_z = Av_z$.
- space dependent density, $m = \int_z^{z+dz} \rho A dx$.

$$\begin{aligned} Acc &= In - Out \\ \frac{dm}{dt} &= w_{z_{in}} - w_{z+dz_{out}} \\ \frac{d}{dt}\int_z^{z+dz} \rho A dx &= \rho_z A v_z - \rho_{z+dz} A v_{z+dz} \end{aligned}$$

Move the time derivative inside integral, divide with the volume, let dz go to zero.

$$\frac{1}{dz}\int_z^{z+dz} \frac{\partial \rho}{\partial t} dx = -\frac{v_{z+dz}\rho_{z+dz} - v_z \rho_z}{dz}$$

$$\frac{\partial \rho}{\partial t} = -\frac{\partial (v\rho)}{\partial z}$$

Continuity Equations IV

Energy balances

Dynamic energy balance over a system:

$$\begin{aligned} Acc &= In - Out \\ \frac{dE}{dt} &= e_{in} + Q - e_{out} - W \end{aligned}$$

Terms:

$E = U_{tot} + K_{tot} + \Phi_{tot}$, sum of internal, kinetic and potential energy.

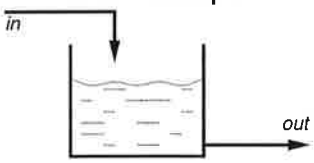
$e = (u + \frac{v^2}{2} + gh)w$, flow terms of internal, kinetic and potential energy.

$Q = Q_a - Q_w$, heat added or withdrawn by conduction, radiation and reaction.

$W = W_{shaft} + \frac{w_{out}}{\rho_{out}} p_{out} - \frac{w_{in}}{\rho_{in}} p_{in}$, work done by the system on surroundings, shaft work and PV work.

Energy Balances I

Tank Example



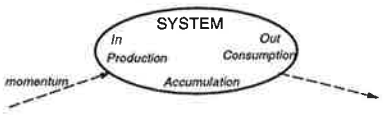
Total energy balance over a simple tank. *Assumptions:*

- Q and W_{shaft} are zero.
- for liquids: $U \gg K$ and $U \gg \Phi$.
- use $mh = mu + pV$ and $h = u + \frac{p}{\rho}$.
- neglect the change of the pV work.

$$\begin{aligned} \frac{d(U+K+\Phi)}{dt} &= w_{in}(u_{in} + K_{in} + \Phi_{in}) \\ &\quad - w_{out}(u_{out} + K_{out} + \Phi_{out}) \\ &\quad - (\frac{w_{out}}{\rho_{out}} p_{out} - \frac{w_{in}}{\rho_{in}} p_{in}) \\ \frac{dU}{dt} &= w_{in}(u_{in} + \frac{p_{in}}{\rho_{in}}) - w_{out}(u_{out} + \frac{p_{out}}{\rho_{out}}) \\ \frac{d(mh)}{dt} &= w_{in}h_{in} - w_{out}h_{out} \end{aligned}$$

Continuity Equations V

Momentum Balances



Dynamic momentum balance (or force balance):

$$\begin{aligned} Acc &= In- \quad Out+ \quad Prod- \quad Cons \\ \frac{d(mv)}{dt} &= F_{in}- \quad F_{out}+ \quad G- \quad L \end{aligned}$$

$F = wv + pA$, forces from convection and pressure,

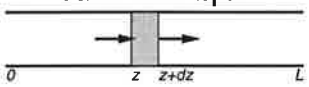
Gravitation is a production term, mg.

Losses are consumption terms.

Note that this describes the acceleration and the position is the integral of the velocity, $\frac{dx}{dt} = v$, which results in a second order system.

Momentum Balances I

Tube Example



Momentum balance over a tube element.
Assumptions:

- constant cross area, $w = \rho A v$.
- space dependent density, $m = \int_z^{z+dz} \rho A dx$.
- losses expressed as $L = -\tau A$.

$$\frac{d(mv)}{dt} = F_{in} - F_{out} + G - L$$

$$\frac{d}{dt}\int_z^{z+dz} \rho A v dx = w_z v_z - w_{z+dz} v_{z+dz}$$

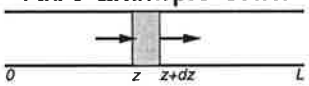
$$+p_z A - p_{z+dz} A + \int_z^{z+dz} \rho A dx g + tau_z A - tau_{z+dz} A$$

$$\frac{d}{dt}\int_z^{z+dz} \rho A v dx = A\rho_z v_z^2 - A\rho_{z+dz} v_{z+dz}^2$$

$$+p_z A - p_{z+dz} A + \int_z^{z+dz} \rho A dx g + tau_z A - tau_{z+dz} A$$

Momentum Balances II

Tube Example cont.



Move time derivative inside integral, divide with the volume, let dz go to zero.

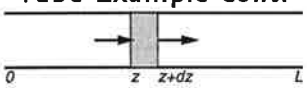
$$\begin{aligned} \frac{1}{dz}\int_z^{z+dz} \frac{\partial(\rho v)}{\partial t} dx = &- \frac{v_{z+dz}^2 \rho_{z+dz} - v_z^2 \rho_z}{dz} \\ &- \frac{p_{z+dz} - p_z}{dz} \\ &+ \rho g \\ &- \frac{\tau_{z+dz} - \tau_z}{dz} \end{aligned}$$

This gives the following equation:

$$\frac{\partial(\rho v)}{\partial t} = -\frac{\partial(\rho v^2)}{\partial z} - \frac{\partial p}{\partial z} + \rho g - \frac{\partial \tau}{\partial z}$$

Momentum Balances III

Tube Example cont.



2:nd Assumptions:

- Newtonian fluid, $\tau = -\mu \frac{dv}{dz}$.
- constant density and viscosity.

$$\rho \frac{\partial v}{\partial t} = -\rho \frac{\partial v^2}{\partial z} - \frac{\partial p}{\partial z} + \rho g + \mu \frac{\partial^2 v}{\partial z^2}$$

This is the *Navier-Stokes equation* in one space dimension.

In three space dimensions it becomes

$$\rho \frac{\partial v}{\partial t} = -\rho \nabla v v - \nabla p + \rho g + \mu \Delta v$$

Continuity Equations VI

Mechanical Energy Balance

Mechanical energy balance appears if the momentum balance is multiplied by the velocity. (for differential balances)

$$\frac{\partial(\rho v^2)}{\partial t} = -\frac{\partial(\rho v^3)}{\partial z} - v\frac{\partial p}{\partial z} - v\frac{\partial \tau}{\partial z} + \rho v g$$

$$\frac{\partial(\rho v^2)}{\partial t} = -\frac{\partial(\rho v^3)}{\partial z} - \frac{\partial(pv)}{\partial z} + p\frac{\partial v}{\partial z} - \frac{\partial(\tau v)}{\partial z} + \tau\frac{\partial v}{\partial z} + \rho v g$$

(see BSL pp. 81)

Continuity Equations VII

Mechanical Energy Balance cont.

Macroscopic system with isothermal properties:

$$\begin{array}{rcccc} Acc = & In- & Out+ & Prod- & Cons \\ \frac{d(K+\Phi+A)}{dt} = & E_{in}- & E_{ut}- & W- & L_e \end{array}$$

$K + \Phi + A$ are kinetic, potential and (Helmholtz) free energy.

$E = w(\frac{1}{2}v^2 + gh + \frac{p}{\rho})$ are kinetic, potential and pressure energy from convection.

W are work done by the system

L_e are friction losses where mechanical energy irreversible converts to thermical energy.

Mechanical Energy Balances

Tank Example

Macroscopic mechanical energy balance over a tank. *Assumptions:*

- isothermal.
- no work and losses.
- constant density, $A = 0$.
- kinetic energy, $K = \frac{1}{2}\rho V v^2$ (v is tank velocity).
- potential energy, $\Phi = \rho V g x$ (x is tank position, $\frac{dx}{dt} = v$).

$$\frac{d(K+\Phi)}{dt} = w_{in}(\frac{1}{2}v_{in}^2 + gx_{in} + \frac{p_{in}}{\rho}) - w_{out}(\frac{1}{2}v_{out}^2 + gx_{out} + \frac{p_{out}}{\rho})$$

$$\rho\frac{d(\frac{1}{2}Vv^2 + Vgx)}{dt} = w_{in}(\frac{1}{2}v_{in}^2 + gx_{in} + \frac{p_{in}}{\rho}) - w_{out}(\frac{1}{2}v_{out}^2 + gx_{out} + \frac{p_{out}}{\rho})$$

Transport Phenomena

Molecular transport (microscopic phenomena):

Quantity	Heat	Mass	Momentum
Flux	q	N_A	τ_z
Force	$\frac{\partial T}{\partial z}$	$\frac{\partial C_A}{\partial z}$	$\frac{\partial v_z}{\partial z}$
Property	Conductivity	Diffusivitity	Viskosivity
	k_T	D_A	μ
Law	Fourier	Fick	Newton
Relation	$q = k_T\frac{\partial T}{\partial z}$	$N_A = D_A\frac{\partial C_A}{\partial z}$	$\tau_z = \mu\frac{\partial v_z}{\partial z}$

Overall transport (macroscopic phenomena):

Quantity	Heat	Mass	Momentum
Flux	q	N_A	τ_z
Force	ΔT	ΔC_A	ΔP
Property	Heat transfer	Mass transfer	Friction
	h_T	k_L	***
Relation	$q = h_T\Delta T$	$N_A = k_L\Delta C_A$	***

see table 2.1 in Luyben

Kinetic Description

Chemical Reaction

Simple kinetic relations:

$$n_A A \longrightarrow n_B B$$

$$r = \frac{1}{n_A} k c_A^n \quad ; \quad r_A = n_A r \quad ; \quad r_B = n_B r$$

$$k = k_0 e^{-\frac{E_a}{RT}}$$

Michaelis-Menten-kinetics (enzyme catalysed kinetics):

$$S + E \longleftrightarrow ES \longrightarrow P + E$$

$$v = \frac{v_{max} s}{K_m + s} \quad ; \quad K_m = \frac{k_1}{k_{-1}} \quad ; \quad v_{max} = \alpha e_0$$

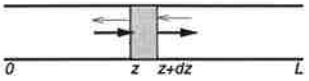
Monod-kinetics (cell growth):

$$\mu = \frac{\mu_{max} s}{K_s + s}$$

Kinetic expressions are very application dependent and particular multi phase system. Geometry, side-reactions, inhibations make the expressions much more complicated in practice.

Mass Balance IV

Tube with Diffusion and Reaction



Component balance over a volume element in a tube. *Assumptions:*

- constant cross area, $V = Adz$.
- Fick's law, $N = D\frac{\partial c}{\partial z}$.
- space dependent n, $n = \int_z^{z+dz} Acdx$.

$$\frac{\partial}{\partial t}\int_z^{z+dz} Acdx = (qc)_z + AN_{z+dz} - (qc)_{z+dz} - AN_z + Vr$$

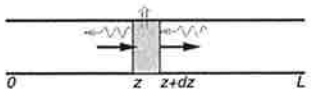
Move time derivative inside integral, divide with the volume, let dz go to zero:

$$\frac{1}{dz}\int_z^{z+dz} \frac{\partial c}{\partial t} dx = -\frac{((vc)_{z+dz} - (vc)_z)}{dz} + D\frac{(\frac{\partial c}{\partial z}_{z+dz} - \frac{\partial c}{\partial z}_z)}{dz} + r$$

$$\frac{\partial c}{\partial t} = -\frac{\partial (vc)}{\partial z} + D\frac{\partial^2 c}{\partial z^2} + r$$

Energy Balance II

Tube with Heat Transfer and Reaction



Energy balance over a volume element in a tube. *Assumptions:*

- constant cross area, $V = Adz (= 2\pi r_r dz)$.
- no shaft work.
- Fourie's law, $q = k_T \frac{\partial T}{\partial z}$.
- neglect potential and kinetic energies in total balance.
- space dependent e, $e = \int_z^{z+dz} \rho A u dx$.
- heat from reaction, $Q_p = \Delta H_r r$.
- heat transfer out, $q_t = A_c h_T (T - T_0)$.

$$\frac{\partial}{\partial t}\int_z^{z+dz} \rho A u dx = (wh)_z + Aq_{z+dz} - (wh)_{z+dz} - Aq_z + Q_p A dz - q_t$$

Energy Balance III

Tube with Heat Transfer and Reaction cont.

Move time derivative inside integral, divide with the volume, let dz go to zero:

$$\frac{1}{dz}\int_z^{z+dz} \frac{\partial(\rho u)}{\partial t} dx = -\frac{((\rho vh)_{z+dz} - (\rho vh)_z)}{dz} + k_T \frac{(\frac{\partial T}{\partial z}_{z+dz} - \frac{\partial T}{\partial z}_z)}{dz} + Q_p - \frac{A_c}{Adz} h_t (T - T_0)$$

$$\frac{\partial(\rho u)}{\partial t} = -\frac{\partial(\rho vh)}{\partial z} + k_T \frac{\partial^2 T}{\partial z^2} + Q_p - \frac{2}{r_r} h_T (T - T_0)$$

Additional Assumptions:

- constant heat capacity, $h = C_p T$.
- constant pressure and use $h = u + \frac{p}{\rho}$.

$$\frac{\partial(\rho T)}{\partial t} = -\frac{\partial(\rho v T)}{\partial z} + \frac{k_T}{C_p}\frac{\partial^2 T}{\partial z^2} + \frac{Q_p}{C_p} - \frac{2h_T}{r_r C_p}(T - T_0)$$

System Descriptions I

"Macroscopic Models"

Put up the continuity balances and write them on *state space form* of ordinary differential equations, ODE:s.

$$\frac{dx}{dt} = f(x, t, u)$$

x are the system state vector (mass, component, temperature, velocity, position etc.).

Static models has zero left hand side resulting in a nonlinear equation system.

$$0 = f(x, u)$$

Special case: f is linear

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

A static linear model becomes $0 = Ax + Bu$ or rewritten $Ax = b$.

System Descriptions II

"Microscopic Models"

Systems with space dimensions are described by partial differential equations, PDE:s.

$$\frac{\partial x}{\partial t} = f(\frac{\partial x}{\partial z}, \frac{\partial^2 x}{\partial z^2}, x, t, u)$$

Nonlinear PDE:s are in general hard to solve.

Some linear PDE:s are more easy to handle:

- $\frac{\partial x}{\partial t} = k\frac{\partial^2 x}{\partial z^2}$ is a parabolic PDE or diffusion equation (tube with diffusion).
- $\frac{\partial^2 x}{\partial t^2} = k\frac{\partial^2 x}{\partial z^2}$ is a hyperbolic PDE or wave equation
- $\frac{\partial x}{\partial t} = k\frac{\partial x}{\partial z}$ is also a hyperbolic PDE (mass balance in tube)
- $0 = k\frac{\partial^2 x}{\partial z^2}$ is a elliptic PDE or Laplace equation (static PDE in space).

Physical Modelling

Assumptions

- System boundary, conceptual decomposition.
- Quantities to model. Choice of state.
- Neglected properties
  - dynamics,
  - flow terms,
  - production/consumption terms.
- Medium properties.
  - characteristic dependency.
- Geometry
- Lumped or distributed
  - Space dimensions,
  - Choice of coordinates,

Physical Modelling

Conclusions

1. Put up *physical and chemical laws*,
   - Continuity equations,
   - Flow terms,
   - Medium descriptions,
2. Make *assumptions*,
   - Accuracy,
   - Validity,
   - Complexity,
3. Generate *system descriptions*,
   - Algebraic equations, AE,
   - ODE:s,
   - PDE:s.
4. *Problem Formulation*
5. *Problem Solving*

PROCESS MODELLING

lecture III

Computer Aided Modelling

Computer Aided Modelling

Outline:

- System Descriptions and Problem Formulations
- CSSL languages
- Graphical Modelling Tools
  - Analog computer description
  - Block diagram
- Object-Oriented Modelling
  - Structure decomposition
  - Equation-based description
  - Reuse and inheritance
- Bond graphs
- Differential-algebraic equations, DAE

Computer Aided Modelling I

Systems, Problems and Solutions

1. System Description (Mathematical models)
   - AE
   - ODE
   - PDE
2. Problem Formulation
   - direct (simulation)
   - inverse
   - design
3. Problem Solving
   - equation solving
   - integration

Computer Aided Modelling II

Problem Solving Tools

Dynamic Simulation

- *Programming*
  PF and PS are integrated
- *CSSL language*
  (Simnon, ACSL, SpeedUp, etc)
  PF and PS are separated
  PF is the description
- *CSSL with graphics and libraries*
  (SimuLink, SystemBuild, etc)
  Structuring of PFs
  Reuse of predefined PF
- *Object-Oriented Modelling Language*
  (Dymola, Omola, ASCEND, etc)
  Equation based models
  Reuse of models and Inheritance
  Automatic manipulation to PF and PS

CSSL Language I

Problem Formulation

Problem formulation oriented model description.

A set of ODE:s on state space form (explicit):

$$\frac{dx}{dt} = f(x, t, u)$$

The assignments are sorted to generate a calculation procedure.

CSSL Language II

Example: MATLAB

```
function xdot = tankmodel(t,x,c)
qin  = 1;
qout = 1;
cin  = 1;
xdot(1) = qin - qout;
c = x(2)/x(1);
xdot(2) = qin*cin + qout*c;
```

No sorting of the assignment. Pure calculation procedure.

CSSL Language III

Example: Simnon

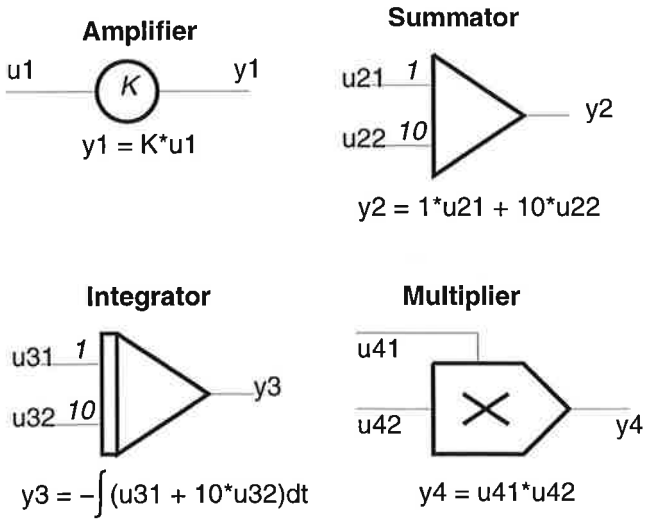
```
CONTINUOUS SYSTEM TankModel
STATE V Vc
DER dV dVc

"  equations
dV  = qin - qout
dVc = qin*cin + qout*c
c   = Vc/V

"  parameters
qin  : 1
qout : 1
cin  : 1

"  initial values
V  : 1
Vc : 1
END
```

Sorting of the assignments.

Graphical Modelling I

Analog Computer Descriptions

Graphical description based on a set of electrical object with given behaviour.



Analog Computer Description I

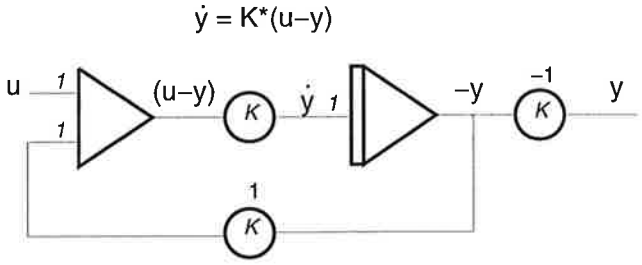
Tank Example

The simple tank example from lecture II.

- component mole balance
- constant volume and flow,
- homogeneous mixed,
- no reaction,

$$\dot{c} = \frac{q}{V}(c_{in} - c)$$

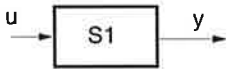
This is the corresponding analog computer configuration.



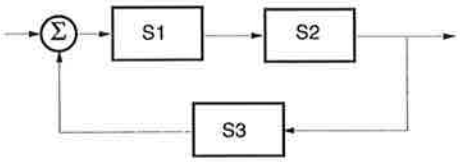
Graphical Modelling II

Block Diagram Descriptions

Each block or system is a representation of the transformation of inflow information, **u**, to the outflow, **y**.



Block diagrams are graphical description of the *information flow*.



Graphical Modelling III

Transfer Functions

A common way to describe blocks are by *transfer functions*. Transfer function is the *Laplace transform* of the corresponding differential equation.

$$\dot{c}(t) = \frac{q}{V}(c_{in}(t) - c(t))$$

$$L\{\dot{c}(t)\} = \frac{q}{V}(L\{c_{in}(t)\} - L\{c(t)\})$$

$$sC(s) - c(0) = \frac{q}{V}(C_{in}(s) - C(s))$$

$$(s + \frac{q}{V})C(s) = \frac{q}{V}C_{in}(s) + c(0)$$

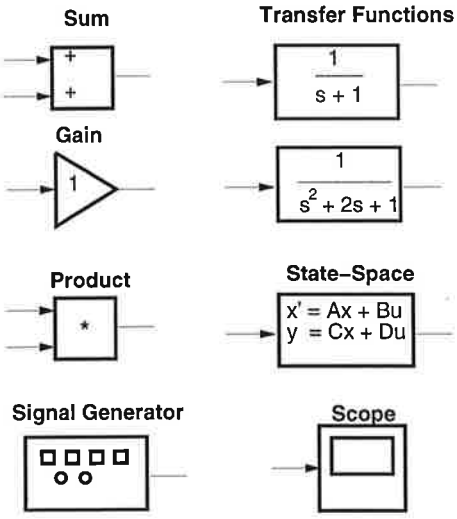
$$C(s) = \frac{\frac{q}{V}}{(s + \frac{q}{V})}C_{in}(s) + \frac{1}{(s + \frac{q}{V})}c(0)$$

For convenience the *initial condition* is assumed to be *zero* and the transfer function for the differential equation becomes:

$$C(s) = \frac{\frac{q}{V}}{(s + \frac{q}{V})}C_{in}(s)$$

Graphical Modelling IV

Block Diagram Descriptions
a la SIMULINK



Block Diagram Descriptions

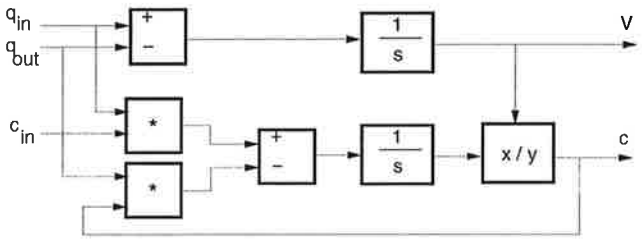
Tank Example

A simple tank example. *Assumptions:*

- constant density
- isothermic
- ⋆ dynamic mass balance

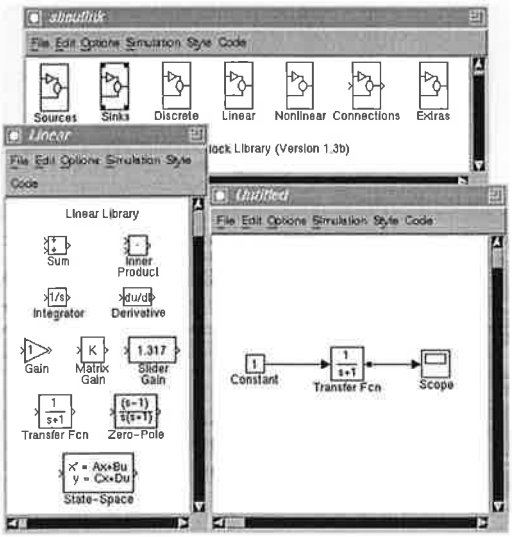
  $\dot{V} = q_{in} - q_{out}$
- ⋆ dynamic component balance

  $(\dot{V}c) = q_{in}c_{in} - q_{out}c$



SIMULINK

CSSL Language with Graphics



SIMULINK is put upon MATLAB.

- Click and drag
- Menu driven interaction
- "Block" libraries

Object-Oriented Modelling I

Equation Based Description

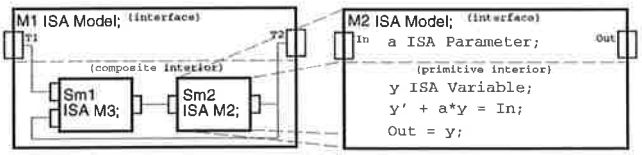
Real modelling tools are problem independent. This means:

- Model describes relations, not execution.
- Unknown causality (what are in and out?).
- Equation based description.

```
TankModel ISA Model WITH
  qin, qout, cin, k ISA Parameter;
  V, c, cout, r     ISA Variable;
  % In    + Prod = Out       + Acc
  qin              = qout     +     V';
  qin*cin + V*r    = qout*cout + (V*c)';
  % assumptions
  cout = c;
  % reaction kinetics
  r    = -k*c;
END;
```

Object-Oriented Modelling II

Abstraction



A model object is described in two parts.

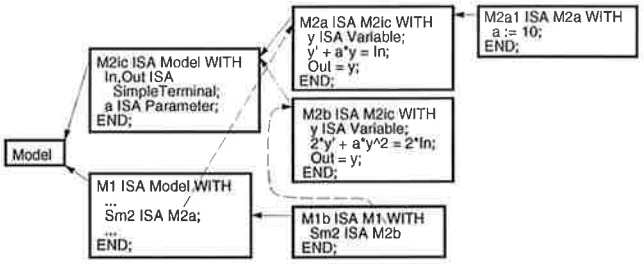
- **Interface** describes the interaction with other models and model user.
- **Interiour** describes the model behaviour, composite or primitive.

Structuring properties.

- Encapsulated models,
- Decomposition into structure hierarchy,
- Graphical representation.

Object-Oriented Modelling III

Class inheritance

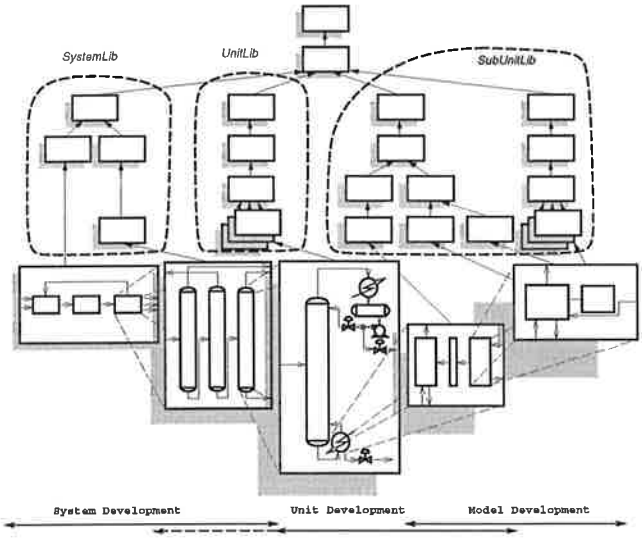


Inheritance facilitates development.

- **Reuse** of models in composite models.
- **Specialization** of a predefined model into a new one.
- **Polymorphism** for the reuse of structures.

Object-Oriented Modelling IV

Library Abstraction



Abstraction of library contents:

- Model development of subunits,
- Unit development using subunits,
- System development using units.

Object-Oriented Modelling V

Current Research

- **ASCEND** from CMU (A. Westerberg). Textual language. Equation solver. (public domain)
- **MODEL.LA.** from MIT (Stephanopoulos). Graphical and inside KEE. Design.
- **gPROMS** from Imperial Collage (Pantelides). New generation of SpeedUp. Textual language. Handle DAE, events and PDE.
- **VeDa** from Aachen/Stuttgart (Marqhart). Language on top of DIVA. Handle AE, DAE, PDE events.
- **Omola** from Lund (Mattsson). Textual and graphical. Handle DAE and events.
- **Dymola** from Dynasim, Lund (Elmqvist). Textual and graphical. Handle DAE and events.

Object-Oriented Modelling VI

Omola and OmSim

CACE group; S.E. Mattsson, M. Andersson, B. Nilsson, T. Schönthal and J. Eborn.

Omola is the modelling language and *OmSim* is the simulation environment.

- Modelling
  - Graphical editor for structures,
  - Textual editor (emacs) for equations,
  - Browser for library display,
- Problem formulation ("automatic")
  - Order and sort equations,
  - Manipulation and index reduction,
  - Generate simulation code,
- Problem solving
  - Simulator for dynamic studies,
    - Parameter sheets
    - Plot windows

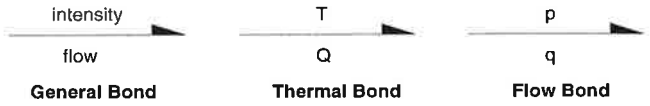
⋆ C++, Unix and Sun/HP. (public domain)

Bond Graphs I

Graphical Modelling

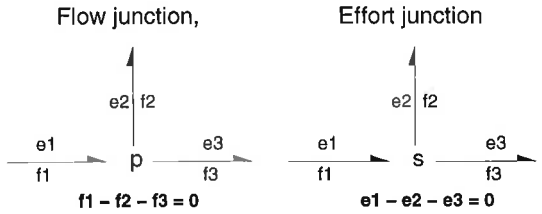
Introduced by Paynter, MIT.

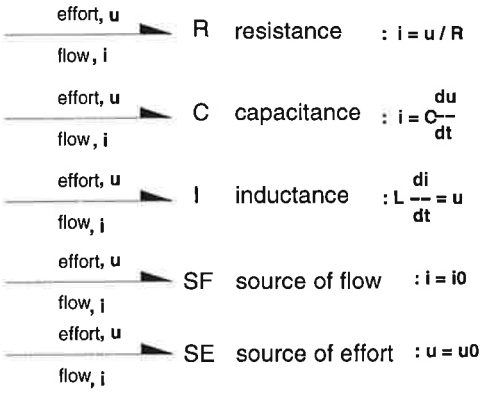
Idea: the product of the terms *flow* and *effort* is power.



Bond Graphs II

Bond Graph Elements



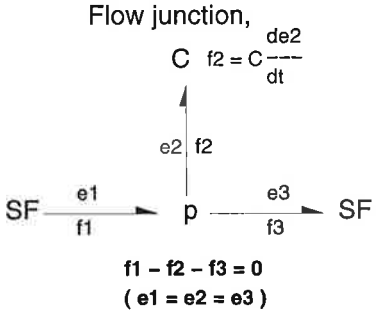


Bond Graphs III

Tank Example

A simple tank with a dynamic mass balance.
Assumptions:

- constant density.
- dynamic mass (volume) balance, $\dot{V} = q_i - q_o$.
- static momentum balance, $p_o = \rho g \frac{V}{A}$.
  - product ef is power, f is volumetric flow and e is pressure.



Transformation:

f1 = qi
f3 = qo
e2 = p

$C = \frac{A}{\rho g}$

Interpretation:

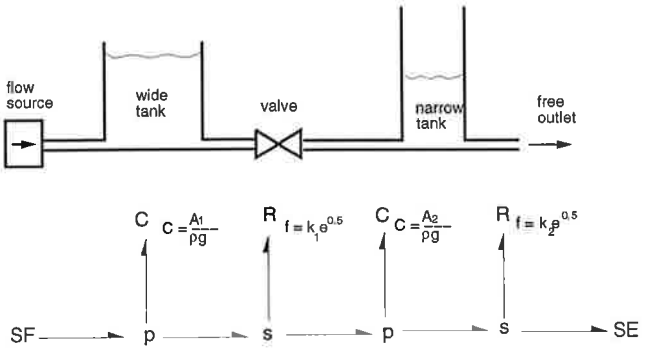
qi – f2 – qo = 0

$\frac{dp}{dt} = \frac{f2}{C}$

e1 = e3 = p

Bond Graphs IV

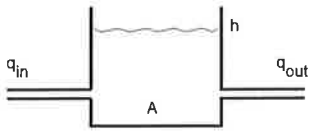
Flow System Example



Object-Oriented Modelling V

DAE system

Library definition: `SimpleTankModel`

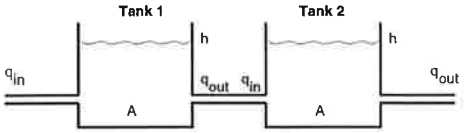


- Interface
  - terminals: q_{in} and q_{out},
  - parameter: A,
- Interiour
  - state: h,
  - equation: $\frac{dh}{dt} = \frac{1}{A}(q_{in} - q_{out})$

If the interface is known the interiour is well defined.

Object-Oriented Modelling VI

DAE system cont.



$$\frac{dh_1}{dt} = \frac{1}{A_1}(q_{1_{in}} - q_{1_{out}})$$
$$\frac{dh_2}{dt} = \frac{1}{A_2}(q_{2_{in}} - q_{2_{out}})$$

The connection gives an additional equation:

$$q_{2_{in}} - q_{1_{out}} = 0$$

Degree of freedom analysis:

- $n_{equations} = 3$
- $n_{variables} = 8$
  - $n_{knowns} = 4$
  - $n_{unknowns} = 4$

Result: One equation is missing

Object-Oriented Modelling VII

DAE system cont.

The missing equation: flow between the tanks.

1. rate equation: $q_{out_1} = A\sqrt{2\rho(h_1 - h_2)}$.
   - 2:nd order ODE.
   - (or DAE index 0)
2. equilibrium equation 1: $\dot{h_1} = \dot{h_2}$.
   - differential-algebraic equation system, DAE.
   - initialization $h_1(0) = h_2(0)$.
   - manipulation $=$ 1 ODE + 1 AE.
   - (DAE index 1)
3. equilibrium equation 2: $h_1 = h_2$.
   - DAE system
   - derivation to reach 2.
   - (DAE index 2)

Conclusions

- CSSL language
- CSSL with graphics
  - Block diagram descriptions
- Object-Oriented Modelling
  - Equation based description
  - Abstraction and encapsulation
  - Inheritance and reuse
  - Current research
- Bond graph modelling
- DAE problems

PROCESS MODELLING

lecture IVa

Linear Systems

Content

- Linear system descriptions,
- First order systems,
- Eigenvalue problem,
- Stability,
- Second order systems,
- Phase plan analysis,
- Chemical reaction example.

Linear System I

General Description

Systems on state space form

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

- x is a state vector, $(n, 1)$.
- A is a matrix, (n, n).
- u is a vector with m input signals, $(m, 1)$.
- B is a matrix, (n, m).
- y is a vector of k measurements, $(k, 1)$.
- C is a matrix, (k, n).
- D is a matrix, (k, m).

Special case: Single input/single output (SISO), one input signal (B is a column and u is a scalar) and one measurement (C is a row and y is a scalar) and no direct term (D is zero).

Linear System II

Steady-State Solution

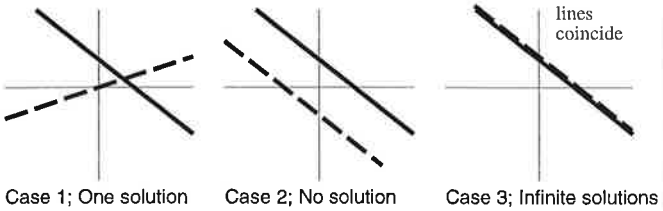
Steady-state means that the left hand side is zero:

$$0 = Ax + Bu$$
$$y = Cx + Du$$

Rewrite the first equation and assume that $b = -Bu$, then

$$Ax = b$$

- $x = A^{-1}b$ if A is invertible, (`x = A \ b`).
  - mathematical: A has full rank (`rank(A)`).
  - numerical: low condition number, (`cond(A)`).



Case 1; One solution | Case 2; No solution | Case 3; Infinite solutions

Linear System III

First order system

Solve the system equation (when $u(t) = u_0$ för $t > 0$)

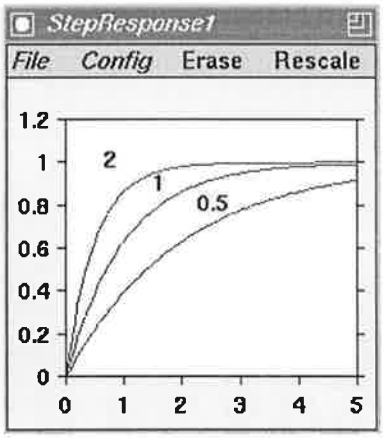
$$\begin{aligned}
\dot{x} &= ax + bu \\
e^{-at}(\dot{x} - ax) &= e^{-at}bu \\
e^{-at}x - x_0 &= \int_0^t e^{-as}bu(s)ds \\
x(t) &= e^{at}x_0 + \int_0^t e^{a(t-s)}bu(s)ds \\
x(t) &= e^{at}x_0 + b\int_0^t e^{a(t-s)}dsu_0 \\
x(t) &= e^{at}x_0 + \frac{b}{a}(e^{at} - 1)u_0
\end{aligned}$$

We see that if $a < 0$ then:

- Homogenous solution is an exponential and the initial conditions disappear.
- Nonhomogenous solution: $x(\infty) \approx -\frac{b}{a}u(\infty)$.
- if $a > 0$ then the solution goes to infinity.

Linear System IV

First order system cont



Step responses with $-a = 0.5, 1, 2$ for the system:

$$\frac{dx}{dt} = ax + bu$$

a is that same as the eigenvalue λ and $-\frac{1}{T}$.

Linear System V

Time Constant

Assume constant input u_0 and define $a = -\frac{1}{T}$:

$$\begin{aligned}
\dot{x} &= ax + bu \\
x(t) &= e^{at}x_0 + b\int_0^t e^{a(t-s)}dsu_0 \\
&= e^{at}x_0 + \frac{b}{a}(e^{at} - 1)u_0 \\
&= e^{-\frac{t}{T}}x_0 - bT(e^{-\frac{t}{T}} - 1)u_0
\end{aligned}$$

At $t = T$ the states are:

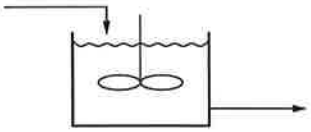
$$\begin{aligned}
x(t = T) &= e^{-1}x_0 - bT(e^{-1} - 1)u_0 \\
&\approx 0.37x_0 + (1 - 0.37)bTu_0
\end{aligned}$$

T is called the *time constant* of the system and when the time is $t = T$:

- The initial condition is 37%.
- The step response is 63% ($x_0 = 0$).
- ($-\frac{1}{T}$ is that same as the eigenvalue λ).

First Order System I

Mixing Tank Example



Assumptions:

- Constant density, volume and flow,
- Isothermal conditions.

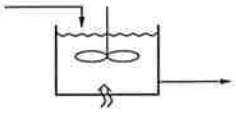
Component balance:

$$\dot{c} = -\frac{q}{V}c + \frac{q}{V}c_{in}$$

- Correspondes to $\dot{x} = ax + bu$ and
- Time constant: $-\frac{1}{T} = a = -\frac{q}{V}$ or $T = \frac{V}{q}$.

First Order System II

Heating Tank Example



Assumptions:

- Constant density, volume and flow,
- Constant heat capacity, $h = C_p T$,
- Heat transfer desciption, $q = \kappa A \Delta T$.

Energy balance:

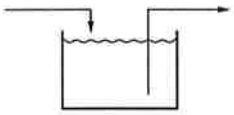
$$\rho V C_p \dot{T} = \rho q C_p (T_{in} - T) + \kappa A (T_h - T)$$

$$\dot{T} = -(\frac{q}{V} + \frac{\kappa A}{\rho V C_p})T + \frac{q}{V} T_{in} + \frac{\kappa A}{\rho V C_p} T_h$$

- Corresponds to $\dot{x} = ax + b_1 u_1 + b_2 u_2$
- Time Constant: $-\frac{1}{T} = -(\frac{q}{V} + \frac{\kappa A}{\rho V C_p})$ or $T = \frac{V}{q} + \frac{\rho V C_p}{\kappa A}$

First Order System III

Buffer Tank Example II



Assumptions:

- Inflow and outflow are independent.

Total mass balance:

$$\dot{m} = w_{in} - w_{out}$$

- Corresponds to $\dot{x} = b_1 u_1 + b_2 u_2$.
- Time Constant: $-\frac{1}{T} = a = 0$ or infinite.
- This is a pure *integrator*.

Linear System V

Higher order system

Solve the system equation (when $u(t) = u_0$ för $t > 0$)

$$\dot{x} = Ax + Bu$$

$$e^{-At}(\dot{x} - Ax) = e^{-At} Bu$$

$$e^{-At} x - x_0 = \int_0^t e^{-As} Bu(s) ds$$

$$x(t) = e^{At} x_0 + \int_0^t e^{A(t-s)} Bu(s) ds$$

$$x(t) = e^{At} x_0 + \int_0^t e^{A(t-s)} ds B u_0$$

We see then that

- The intial conditions depends on e^{AT}.
- The input signal is first scaled by B and multiplied by the integral of e^{At}.

Eigenvalue Problem I

Eigenvalues

$$Az = z\lambda$$

- λ are the eigenvalues of A.
- z are the eigenvectors of A.

This means that the following applies:

$$(A - \lambda I)z = 0 \quad \Rightarrow \quad det(\lambda I - A) = 0$$

which results in a polynomial in λ.

Example:

$$A = \begin{bmatrix} -3 & 1 \\ -2 & 0 \end{bmatrix}$$

$$\det(\lambda I - A) = \det(\lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -3 & 1 \\ -2 & 0 \end{bmatrix}) =$$

$$\det(\begin{bmatrix} \lambda + 3 & -1 \\ 2 & \lambda \end{bmatrix} = \lambda(\lambda + 3) - (-1)2 =$$

$$\lambda^2 + 3\lambda + 2 = 0$$

$$\lambda_1 = -1 \qquad \lambda_2 = -2$$

Eigenvalue Problem II

Eigenvectors

When the eigenvalues are known one can find the eigenvectors:

$$(A - \lambda I)z = 0$$

For each eigenvalue, λ_i, there is a corresponding eigenvector, z_i.

Example:

$$(\lambda_1 I - A)z_1 = \begin{bmatrix} -1-(-3) & -1 \\ -(-2) & -1 \end{bmatrix} z_1 = \begin{bmatrix} 2 & -1 \\ 2 & -1 \end{bmatrix} z_1 = 0$$

$$(\lambda_2 I - A)z_2 = \begin{bmatrix} -2-(-3) & -1 \\ -(-2) & -2 \end{bmatrix} z_2 = \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} z_2 = 0$$

$$z_1 = c\begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad z_2 = c\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Eigenvalue Problem III

Diagonalization

Create a diagonal matrix of the eigenvalues, Λ, and a corresponding matrix of eigenvectors, S. The following are eqvivalent.

$$AS = S\Lambda \quad \Leftrightarrow \quad A = S\Lambda S^{-1} \quad \Leftrightarrow \quad S^{-1}AS = \Lambda$$

Example:

$$S = [z_1 \quad z_2] = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ & -2 \end{bmatrix}$$

Note:

$$e^{At} = e^{(S^{-1}\Lambda S)t} = S^{-1}e^{\Lambda t}S$$

Stability I

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

Properties that are characterized by the eigenvalues:

- Real eigenvalues gives exponential solutions.
- Complex eigenvalues gives oscillatory solution.
  - Eigenvalues < 0 gives asymptotically stable systems.
  - Eigenvalues > 0 gives unstable systems.

Stability II

Second order system

An axample of a second order system:

$$\frac{dx}{dt} = \begin{bmatrix} -2\xi\omega & 1 \\ -\omega^2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u$$

$$y = [1 \quad 0]\, x$$

Eigenvalues:

$$det(\lambda I - \begin{bmatrix} -2\xi\omega & 1 \\ -\omega^2 & 0 \end{bmatrix}) = det(\begin{bmatrix} \lambda + 2\xi\omega & -1 \\ \omega^2 & \lambda \end{bmatrix})$$

$$= \lambda(\lambda + 2\xi\omega) + \omega^2 = 0$$

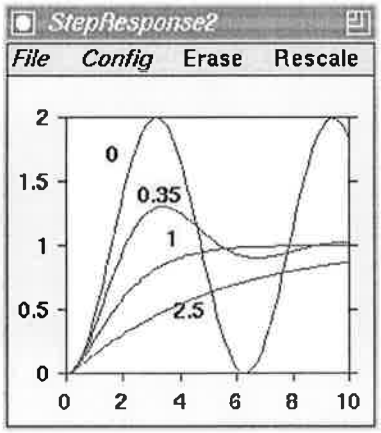
$$\lambda = \omega(-\xi \pm \sqrt{\xi^2 - 1})$$

It can be transformed into input/output from:

$$\ddot{y} + 2\xi\omega\dot{y} + \omega^2 y = \omega^2 u$$

Stability III

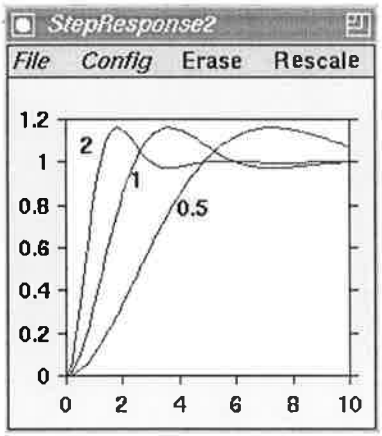
Second order system



Step responses with $\xi = 0, 0.35, 1, 2.5$ and $\omega = 1$.

Stability IV

Second order system



Step responses with $\xi = 0.5$ and $\omega = 0.5, 1, 2$.

Second Order System I

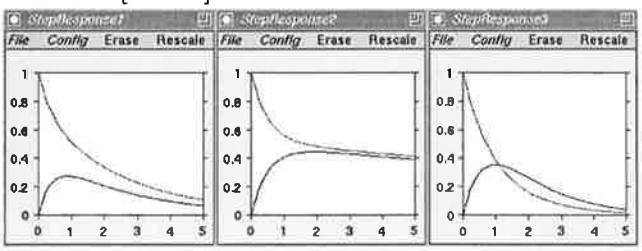
Chemical Reaction

$$A \longleftrightarrow B \longrightarrow C$$

Component balances over a batch reactor:

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \end{bmatrix} = \begin{bmatrix} -k_1 & k_{-1} \\ k_1 & -(k_{-1}+k_2) \end{bmatrix}\begin{bmatrix} c_A \\ c_B \end{bmatrix}$$

Homogenous solutions with the initial condition $c = [1 \quad 0]^T$.



1. $k_{reac} = (1, 1, 1) \Rightarrow \lambda \approx (-0.38, -2.62)$
2. $k_{reac} = (1, 1, 0.1) \Rightarrow \lambda \approx (-0.05, -2.05)$
3. $k_{reac} = (1, 0.1, 1) \Rightarrow \lambda \approx (-0.73, -1.37)$

Phase Plane Analysis I

Phase plane is a plot where one state is plotted as a function of the other state (second order system). For a third order system there is the corresponding phase space.

The phase portreit is a set of trajectories that are plotted in the phase plane.

Examples of charateristics:

- *Node*, trajectories attracted (stable) or repealled (unstable) by the steady state.
- *Saddle point*, trajectories attracted (stable) in one direction and repealled (unstable) in another.
- *Focus*, trajectories that are attracted (repealled) on the same time as it circle around the steady state.
- *Center*, trajectories that circle around the steady state.

Phase Plane Analysis II

Second order system

An example of a second order system:

$$\frac{dx}{dt} = \begin{bmatrix} -2\xi\omega & 1 \\ -\omega^2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

Eigenvalues:

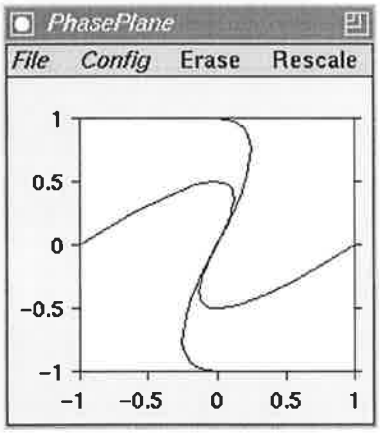
$$\lambda = \omega(-\xi \pm \sqrt{\xi^2 - 1})$$

Eigenvectors:

$$(A - \lambda I)z = \begin{bmatrix} -2\xi\omega - \lambda & 1 \\ -\omega^2 & -\lambda \end{bmatrix} \begin{bmatrix} z_{i1} \\ z_{i2} \end{bmatrix}$$
$$= \begin{bmatrix} -(\lambda + 2\xi\omega)z_{i1} + z_{i2} \\ -\omega^2 z_{i1} - \lambda z_{i2} \end{bmatrix} = 0$$
$$S = \begin{bmatrix} 1 & 1 \\ (\xi + \sqrt{\xi^2 - 1})\omega & (\xi - \sqrt{\xi^2 - 1})\omega \end{bmatrix}$$

Phase Plane Analysis III

Node



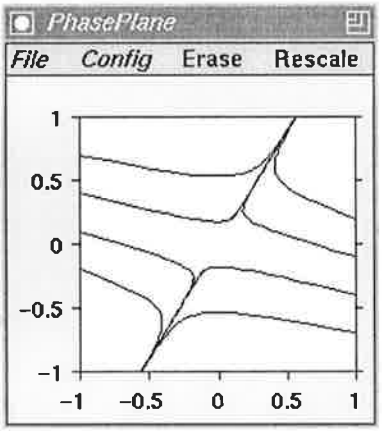
Second order system:

$$\frac{dx}{dt} = \begin{bmatrix} -3 & 1 \\ -2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$$

- eigenvalues in -1 and -2
- eigenvectors, $z = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $z = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Phase Plane Analysis IV

Saddle point



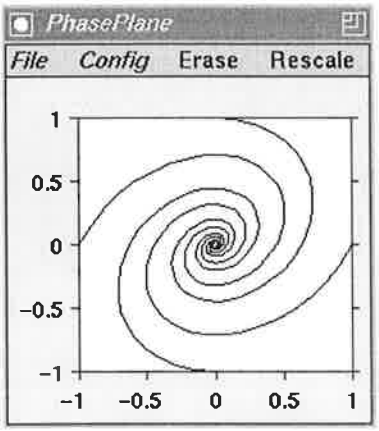
Second order system

$$\frac{dx}{dt} = \begin{bmatrix} -1.5 & 1 \\ 0.5 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u$$

- eigenvalues in ≈ 0.28 and ≈ -1.78
- eigenvectors, $z \approx \begin{bmatrix} 1 \\ 1.78 \end{bmatrix}$ and $z \approx \begin{bmatrix} 1 \\ -0.28 \end{bmatrix}$

Phase Plane Analysis V

Focus



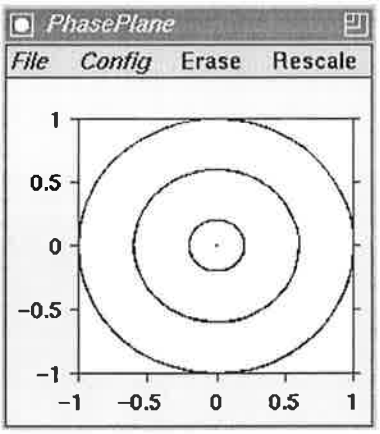
Second order system

$$\frac{dx}{dt} = \begin{bmatrix} -0.7 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

- eigenvalues in $\lambda = 0.35 \pm 0.94i$.
- eigenvectors has complex cofficients.

Phase Plane Analysis VI

Center



Second order system

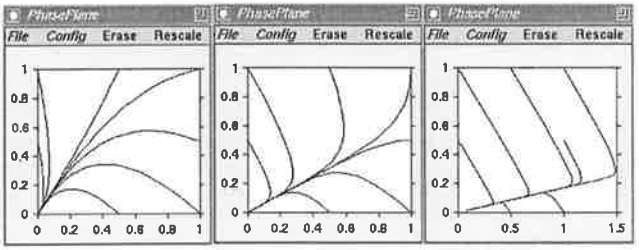
$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

- eigenvalues in $\lambda = \pm i$.

Chemical Reaction I

Phase Plane

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \end{bmatrix} = \begin{bmatrix} -1 & k_{-1} \\ 1 & -(k_{-1}+1) \end{bmatrix} \begin{bmatrix} c_A \\ c_B \end{bmatrix}$$



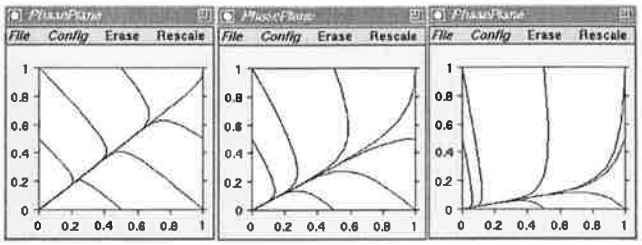
Phase plane plots for $k_{-1} = 0.2, 1, 5$.

1. Eigenvalues $\lambda \approx \begin{bmatrix} -0.64 & -1.56 \end{bmatrix}^T$
2. Eigenvalues $\lambda \approx \begin{bmatrix} -0.38 & -2.62 \end{bmatrix}^T$
3. Eigenvalues $\lambda \approx \begin{bmatrix} -0.15 & -6.85 \end{bmatrix}^T$

Chemical Reaction II

Phase Plane

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -(1+k_2) \end{bmatrix} \begin{bmatrix} c_A \\ c_B \end{bmatrix}$$



Phase plane plots for $k_2 = 0.2, 1, 5$.

1. Eigenvalues $\lambda \approx \begin{bmatrix} -0.10 & -2.10 \end{bmatrix}^T$
2. Eigenvalues $\lambda \approx \begin{bmatrix} -0.38 & -2.62 \end{bmatrix}^T$
3. Eigenvalues $\lambda \approx \begin{bmatrix} -0.81 & -6.19 \end{bmatrix}^T$

Conclusions

Linear systems:

- Eigenvalues determine the properties of the transient response.
- Eigenvalues determine the stability of the linear system.
- Eigenvectors determine the balance between states.
- Phase plane analysis illustrate the properites of the system.

PROCESS MODELLING

lecture IVb

Nonlinear Systems

Content

- Nonlinear system descriptions,
- Steady state solutions,
- Linearisation,
- Periodic solutions,
- Limit cycles,
- Bifurcation,
- Chaos.

Nonlinear Systems I

System Descriptions

Systems on state space form

$$\frac{dx(t)}{dt} = f_1(x(t), u(t))$$
$$y_1(t) = g_1(x(t), u(t))$$

In nonlinear theory this is often rewritten into an *Automonous System*:

$$\frac{dz(t)}{dt} = f_2(z(t))$$
$$y_2(t) = g_2(z(t))$$

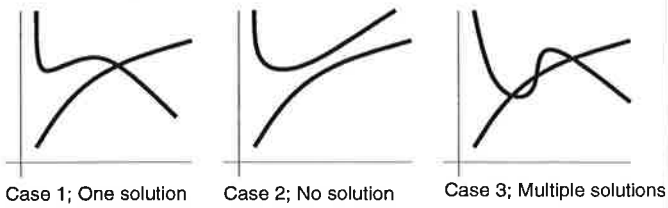
Nonlinear Systems II

Steady State Solutions

Steady state means that the derivative is zero:

$$0 = f(x, u)$$

- To solve the general nonlinear equations is not easy.
- Notice that a nonlinear f can have *multiple* solutions.



Case 1; One solution Case 2; No solution Case 3; Multiple solutions

Linearisation I

Taylor expansion of f.

$$\frac{dx(t)}{dt} = f(x(t)) \approx f(x^o) + \frac{\partial f}{\partial x}(x(t) - x^o)$$

If the linearisation point is a steady state then the nonlinear system become a linear system ($z = x - x^o$):

$$\frac{dz}{dt} = Az$$

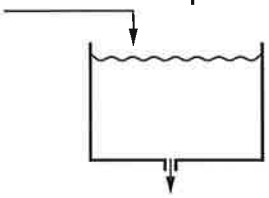
The system matrix A is composed of the partial derivatives at steady state, the *Jacobian*

$$A = \frac{\partial f}{\partial x}$$

Poincaré-Bendixsson: The nonlinear system has the same properties as the linerized system close to the steady state.

Linearisation II

Tank Example



A tank with a free outlet. *Assumptions:*

- Constant density and cylindrical geometry.
- Constant temperature and pressure.
- Outflow, $q_{out} = av_{out} = a\sqrt{2gh}$, (Bernoulli).

Total mass balance:

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

Linearisation III

Tank Example cont.

Linearization (new state $z = h - h^o$ and input $u = q_{in} - q_{in}^o$) give the following linearized system:

$$\dot{z} = az + bu$$
$$a = \frac{\partial}{dh}(-\frac{a}{A}\sqrt{2gh}) = -\frac{a}{A}\frac{2g}{2\sqrt{2gh}} = -\frac{a}{A}\sqrt{\frac{g}{2h}}$$
$$b = \frac{1}{A}$$

The eigenvalue for the system is

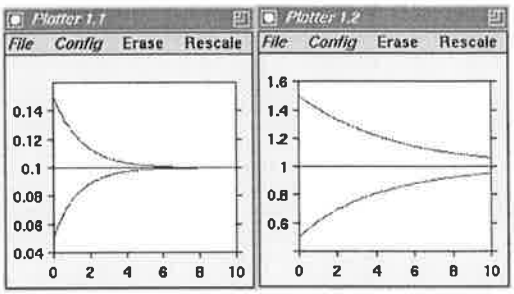
$$\lambda = -\frac{a}{A}\sqrt{\frac{g}{2h^o}}$$

This is a real negative eigenvalue that decrease proportional to the square root of the height.

Linearisation IV

Tank Example cont.

A tank with a free outlet.



Steady state:

- left: $h^o = 0.1$ with $\lambda = -0.7$,
- right: $h^o = 1$ with $\lambda = -0.22$.

Conclusions

Nonlinear Systems, part I

- Nonlinear systems can have multiple steady states.
- A nonlinear system can always be linearized.
- A linearized system has the same properties as the nonlinear system *close* to the steady state.
- *Weak* nonlinear system can be characterized by its linearized approximation.

  (weak = one steady state, eigenvalues do not change characters.)

PROCESS MODELLING

lecture Va

Nonlinear Systems

part II

Content

- Nonlinear system descriptions,
- Steady state solutions,
- Linearisation,
- Periodic solutions,
- Bifurcation,
- Limit cycles,
- Strange attractors and chaos.

Periodic Solutions I

Predator-Pray Model

A chemical reaction turned into a predator-pray model:

$$A + C \longrightarrow 2A + D$$
$$A + B \longrightarrow 2B + E$$
$$B \longrightarrow F + G$$

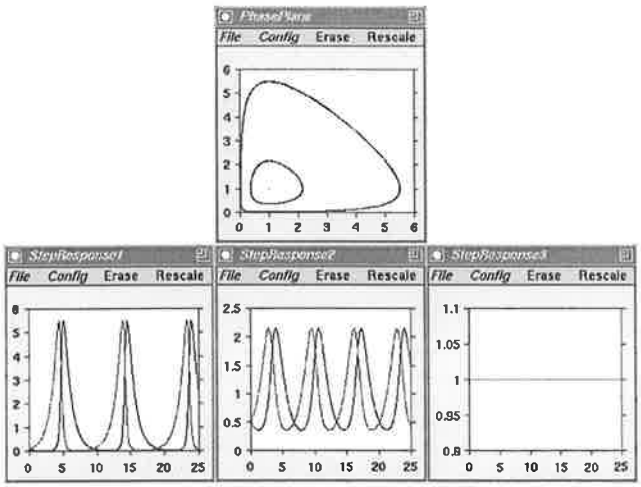
Assume that this reaction take place in a batch reactor:

$$\frac{dA}{dt} = k_1 A - k_2 AB$$
$$\frac{dB}{dt} = k_2 AB - k_3 B$$

- Steady state in $(\frac{k_3}{k_2}, \frac{k_1}{k_2})$ (and $(0,0)$).
- Linearisation gives $A = \begin{bmatrix} 0 & -k_3 \\ k_1 & 0 \end{bmatrix}$.
- Eigenvalues in $\lambda = \pm\sqrt{k_3 k_1}\, i$
- Phase plane will be a center.

Periodic Solutions II

Predator-Pray Model cont.



Phase plane plot and three time response plots for $k_1 = k_2 = k_3 = 1$ and different initial conditions, $(0.1, 0.1)$, $(0.5, 0.5)$ and $(1, 1)$.

Bifurcation I

Hopf equations

$$\dot{x_1} = -x_2 + x_1(a - (x_1^2 + x_2^2))$$
$$\dot{x_2} = x_+ x_1(a - (x_1^2 + x_2^2))$$

Steady state is $(0, 0)$. Linearization derives the following system matrix:

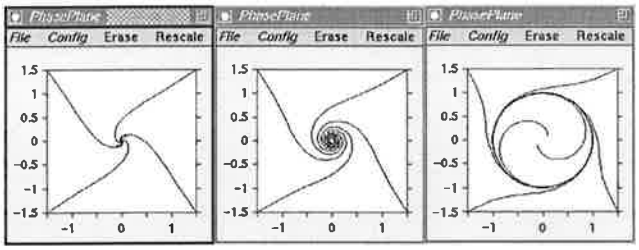
$$A = \begin{bmatrix} a & -1 \\ 1 & a \end{bmatrix}$$

Eigenvalues are $\lambda = -a \pm i$.

Hopf bifurcation: the change of the a parameter results in the change of a damped oscillation, $a > 0$, into an undamped oscillation, $a < 0$.

Bifurcation II

Hopf equations cont.

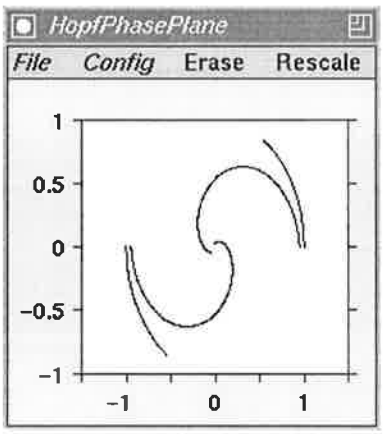


Phase plane plots for the Hopf equation system with $a = -1, 0, 1$.

- *Linear analysis* implies stable focus, center and unstable focus.
- *Nonlinear analysis* gives stable focus, stable focus and limit cycle (local unstable focus).

Limit cycles I

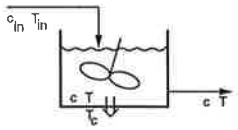
- Stable limit cycles are called *super critical*. Trajectories are attracted towards the cycle.
- *Subcritical* limit cycles are unstable and trajectories are repelled.



The modified Hopf example (change of sign on the quadratic term) with $a = -1$ The subcritical cycle is the unit circle.

Bifurcation III

Exothermic CSTR



Assumptions:

- Homogeneous mixing,
- Constant hold-up (volume and flow),
- Constant density,
- First order reaction (temp. dep.),
- Neglected kinetic and potential energies and pressure-volume change.
- Constant heat capacity,

$$\frac{dc_A}{dt} = (c_{A_{in}} - c_A)\frac{q}{V} - k_0 e^{-\frac{E_a}{RT}} c_A$$
$$\frac{dT}{dt} = (T_{in} - T)\frac{q}{V} - \frac{\Delta H_r}{C_p \rho} k_0 e^{-\frac{E_a}{RT}} c_A - \frac{\kappa A_c}{V C_p \rho}(T - T_c)$$

Bifurcation IV

Exothermic CSTR cont.

The Jacobian or system matrix is:

$$A = \begin{bmatrix} \frac{\partial f_c}{\partial c}_{c^o,T^o} & \frac{\partial f_c}{\partial T}_{c^o,T^o} \\ \frac{\partial f_T}{\partial c}_{c^o,T^o} & \frac{\partial f_T}{\partial T}_{c^o,T^o} \end{bmatrix}$$

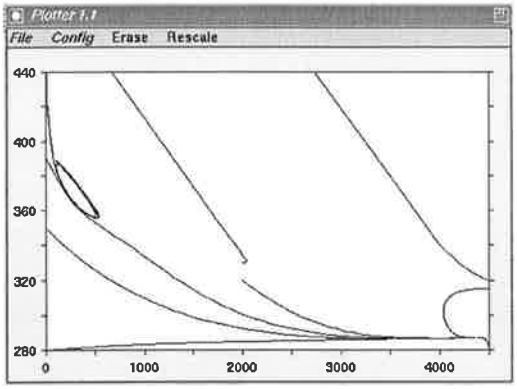
and the eigenvalues of the linearized system becomes:

$$\lambda^2 + a_1\lambda + a_2 = 0$$
$$\lambda_{1,2} = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} - a_2}$$

- If the square root is negative the system has complex eigenvalues,
- a_1 determines the stability of the system.
- $a_1 = 2\frac{q}{V} + \frac{\kappa A}{VC_p\rho} + k_0(1 + \frac{\Delta H_r}{C_p\rho}\frac{E}{RT^{o^2}}c^o)e^{-\frac{E}{RT^o}}$, where the last term is negative,
- *Hopf bifurcation* if $a_1 = 0$ and $a_2 > 0$.

Limit cycles II

Exothermic CSTR cont.



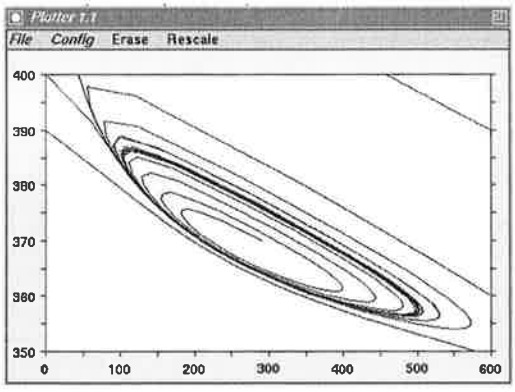
The exothermic CSTR has three steady states:

- 1: $(4387, 287)$ is a stable node.
- 2: $(2337, 327)$ is a saddle point.
- 3: $(272, 368)$ is a unstable focus and results in a limit cycle

  3 can be a stable focus for other parameters.

Limit cycles III

Exothermic CSTR cont.

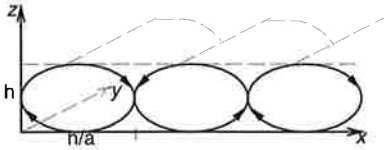


The limit cycle in the exothermic CSTR example.

- eigenvalues are $0.32 \pm 4.62i$.
- oscillations with period ≈ 1.55 hours.

Strange Attractors I

Rayleigh-Bérnard experiment



- Momentum balance: Navier-Stokes eq.,
- Energy balance: $\frac{\partial T}{\partial t} = \kappa\Delta T$,
- Mass continuity equation,
- Simplifications and transformation,

$$\dot{x_1} = s(-x_1 + x_2)$$
$$\dot{x_2} = rx_1 - x_2 - x_1x_3$$
$$\dot{x_3} = -bx_3 + x_1x_2$$

Parameters:

- $s = \frac{\nu}{\rho}$ is the Prandl number,
- $r = \frac{R}{R_c}$ where $R = \frac{gah^3}{\kappa\nu}\Delta T$ and $R_c = \frac{\pi^4}{a^2}(1 + a^2)^3$.
- $b = \frac{4}{1+a^2}$

Strange Attractors II

The Lorentz Equation

$$\dot{x_1} = s(-x_1 + x_2)$$
$$\dot{x_2} = rx_1 - x_2 - x_1x_3$$
$$\dot{x_3} = -bx_3 + x_1x_2$$

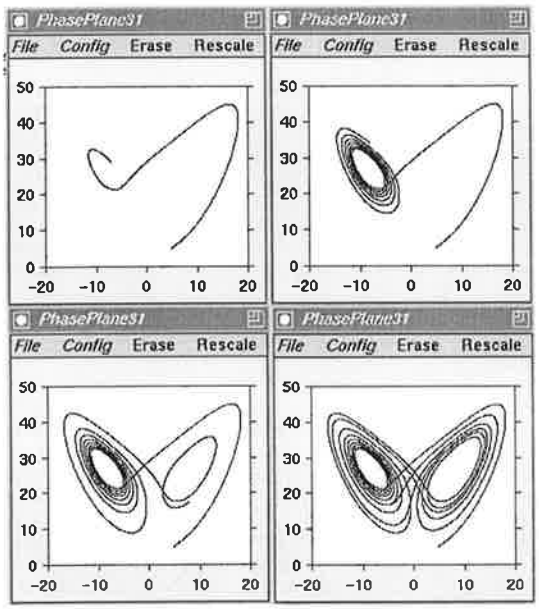
- Steady states are $(\pm\sqrt{b(r-1)}, \pm\sqrt{b(r-1)}, r-1)$ (and $(0,0,0)$ and some other).
- Eigenvalues indicate unstable focuses.
- Linearization derive the following system matrix

$$A = \begin{bmatrix} -s & s & 0 \\ 1 & -1 & -\sqrt{b(r-1)} \\ \sqrt{b(r-1)} & \sqrt{b(r-1)} & -b \end{bmatrix}$$

Strange Attractors III

The Lorentz Equation cont.

Parameters are $s = 10$, $b = \frac{8}{3}$ and $r = 28$. Initial conditions are $x_1(0) = x_2(0) = x_3(0) = 5$.

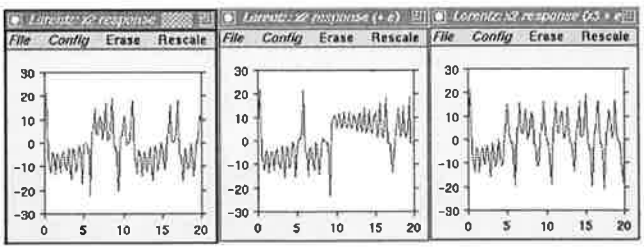


Simulations after 1, 5, 7 and 11 time units.

Chaos

The Lorentz Equation cont.

Parameters are $s = 10$, $b = \frac{8}{3}$ and $r = 28$. Initial conditions are $x_1(0) = x_2(0) = x_3(0) = 5$.



1) x_2 time response.
2) x_2 time response with a small perturbation in the initial condition of x_2.
3) x_2 time response with a small perturbation in the initial condition of x_3.

Deterministic chaotic behaviour or the *butterfly effect*.

Conclusion

- Nonlinear systems can have *multiple steady states* with *different dynamic properties*.
- Stable limit cycles.
- Bifurcation: number of steady states and property of a steady state are parameter dependent.
- Very nonlinear systems can be characterized by strange attractors.
- Deterministic nonlinear systems can behave unpredictable, *chaotic* behaviour.

PROCESS MODELLING

lecture Vb

Distributed Parameter Systems

Content

- Classifications of PDE.
- Diffusion-type problems,
- Wave equations,
- Nonlinear PDE:s.

Distributed Parameter System I

Classification

General second-order linear PDE in two variables:

$$A\frac{\partial^2 u}{\partial t^2} + B\frac{\partial^2 u}{\partial t \partial x} + C\frac{\partial^2 u}{\partial x^2} + D\frac{\partial u}{\partial t} + E\frac{\partial u}{\partial x} + Fu = G$$

Three basic types of linear PDE:s.

1. *Parabolic* PDE satisfy $B^2 - 4AC = 0$.
2. *Hyperbolic* PDE satisfy $B^2 - 4AC > 0$.
3. *Elliptic* PDE satisfy $B^2 - 4AC < 0$.

Distributed Parameter System II

Classification cont.

Initial condition:

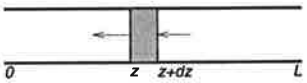
$$u(x,0) = u_0(x) \quad ; \quad \frac{\partial u}{\partial t}(x,0) = u_1(x)$$

Different boundary conditions:

- *Dirichlet* condition $u = f$ on the boundary.
- *Neumann* condition $\frac{\partial u}{\partial n} = f$ on the boundary.
- *Robin* condition or mixed condition $au + b\frac{\partial u}{\partial n} = f$ on the boundary.

Diffusion I

Physical Modelling



Component balance over a volume element in a tube. *Assumptions:*

- constant cross area, $V = Adz$.
- Fick's law, $N = D\frac{\partial c}{\partial z}$.
- space dependent n, $n = \int_z^{z+dz} Acdx$.
- no forced flow.
- no reaction.

$$\frac{\partial}{\partial t}\int_z^{z+dz} Acdx = AN_{z+dz} - AN_z$$
$$\frac{1}{dz}\int_z^{z+dz} \frac{\partial c}{\partial t}dx = D\frac{(\frac{\partial c}{\partial z}_{z+dz} - \frac{\partial c}{\partial z}_z)}{dz}$$
$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial z^2}$$

Diffusion II

A Parabolic PDE

Diffusion in time and one space dimension:

$$\frac{\partial u(x,t)}{\partial t} = \alpha^2 \frac{\partial^2 u(x,t)}{\partial x^2}$$

It is also called the **Heat Equation**.

There is a number of different methods for analysis and solution of PDE:s.

- Separation of variables,
- Fourie's method (eigenfunction expansion),
- Integral transforms,
- etc.

Diffusion III

Separation of Variables

Assume that the solution of a PDE is separated into one time dependent and one space dependent part:

$$u(x,t) = T(t)X(x)$$

Apply to the diffusion equation:

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$$
$$X(x)\frac{\partial T(t)}{\partial t} = \alpha^2 T(t)\frac{\partial^2 X(x)}{\partial x^2}$$
$$\frac{1}{\alpha^2 T(t)}\frac{\partial T(t)}{\partial t} = \frac{1}{X(x)}\frac{\partial^2 X(x)}{\partial x^2}$$

Assume that the two sides are equal to $-k^2$ which then result in two ODE:s

$$\frac{\partial T(t)}{\partial t} = -k^2\alpha^2 T(t)$$
$$\frac{\partial^2 X(x)}{\partial x^2} = -k^2 X(x)$$

The *eigenvalue problem* for PDE:s.

Diffusion IV

Eigenfunctions

Solutions to the two ODE:s.

$$T(t) = Ae^{-k^2\alpha^2 t}$$
$$X(x) = B\sin(kx) + C\cos(kx)$$

Apply boundary conditions, $u(t,0) = u(t,1) = 0$:

- $u(t,0) = T(t)X(0) \Rightarrow X(0) = C = 0$.
- $u(t,1) = T(t)X(1) \Rightarrow X(1) = B\sin k = 0$, which means that $k = \pm n\pi$.

$\Rightarrow$ $u_n(t,x) = A_n e^{-(n\pi\alpha)^2 t}\sin(n\pi x)$ where $\sin(n\pi x)$ are called *eigenfunctions* in x and $e^{-(n\pi\alpha)^2 t}$ an *eigenfunction* in t.

Apply initial condition, $u(0,x) = \Phi(x)$:

- $\Phi(x) = T(0)X(x) = \sum_{n=1}^{\infty} A_n \sin(n\pi x)$

$\Rightarrow$ $A_n = 2\int_0^1 \Phi(x)\sin(n\pi x)dx$

The solution becomes:

$$u(t,x) = \sum_{n=1}^{\infty} A_n e^{-(n\pi\alpha)^2 t}\sin(n\pi x)$$

Diffusion V

Diffusion out from a Tube

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial z^2}$$

Initial and boundary conditions are $c(0,x) = 1$, $c(t,0) = 0$ and $c(t,1) = 0$.

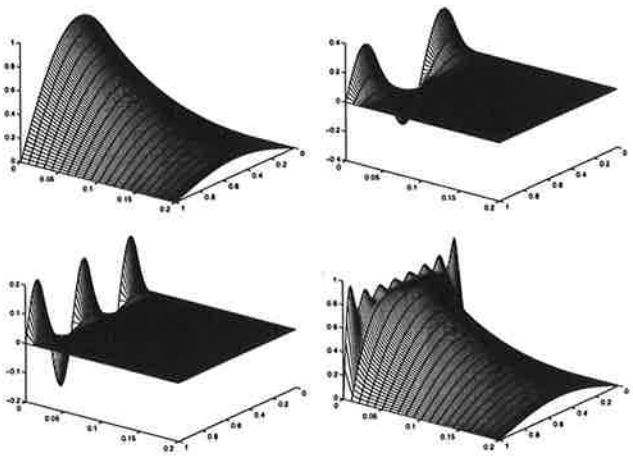
Solution:

$$c(t,x) = \sum_{n=1}^{\infty} A_n e^{-(n\pi\sqrt{D})^2 t}\sin(n\pi x)$$
$$= \frac{4}{\pi}(e^{-\pi^2 Dt}\sin(\pi x) + \frac{1}{3}e^{-3^2\pi^2 Dt}\sin(3\pi x) + ...)$$

- Higher harmonics decay very fast.
- Low harmonics dominate.
- The initial condition is smoothed out and destroyed.

Diffusion VI

Diffusion out from a Tube cont.



The harmonics number 1, 3 and 5 of diffusion problem and the sum of the first 15 harmonics. (time: $0 - 0.2$, space: $0 - 1$).

Diffusion VII

Diffusion with a Point Source

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial z^2}$$

Initial condition is $c(0,x) = \delta(x)$.

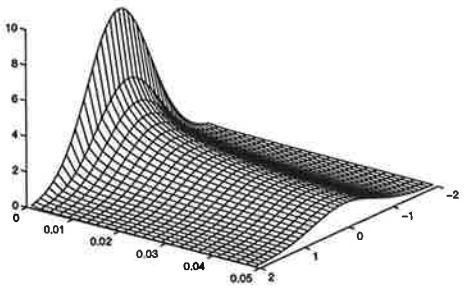
(Fundamental) Solution:

$$c(t,x) = \frac{1}{2\sqrt{\pi t}}e^{-\frac{x^2}{4}Dt}$$

- $c(0,0) = \infty$,
- A point source is spread out in space as a function of time.

Diffusion VIII

Diffusion with a Point Source cont.



(time: $0 - 0.2$, space: $0 - 1$).

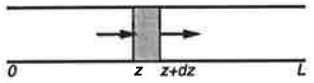
Waves I

Hyperbolic PDE:s

- First order: $\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x}$
  - Solution: $u(t,x) = u_0(x+ct)$.
  - The **wave moves** to the left with the **speed c**.
  - ⋆ Example: mass balance, momentum balance, ...
- Second order: $\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}$
  - General solution: $u(t,x) = F_1(x+ct) + F_2(x-ct)$.
  - The **two waves move** in opposite directions with the **speed c**.
  - ⋆ Example: strings, membrane dynamics, ...

Waves II

Mass Balance in a Tube



Total mass balance over a volume element in a tube. *Assumptions:*

- no mass production.
- constant cross area, $V = Adz$, $q_z = Av_z$.
- space dependent density, $m = \int_z^{z+dz} \rho A dx$.

$$\begin{aligned} Acc &= In - Out \\ \frac{dm}{dt} &= w_{z_{in}} - w_{z+dz_{out}} \\ \frac{d}{dt}\int_z^{z+dz} \rho A dx &= \rho_z A v_z - \rho_{z+dz} A v_{z+dz} \end{aligned}$$

Move the time derivative inside integral, divide with the volume, let dz go to zero.

$$\frac{1}{dz}\int_z^{z+dz} \frac{\partial \rho}{\partial t} dx = -\frac{v_{z+dz}\rho_{z+dz} - v_z\rho_z}{dz}$$

$$\frac{\partial \rho}{\partial t} = -\frac{\partial(v\rho)}{\partial z}$$

Waves III

Mass Balance in a Tube cont.

Assume constant velocity through out the tube.

$$\frac{\partial \rho}{\partial t} = -\frac{\partial \rho v}{\partial z} = -v\frac{\partial \rho}{\partial z} - \rho\frac{\partial v}{\partial z} = -v\frac{\partial \rho}{\partial z}$$

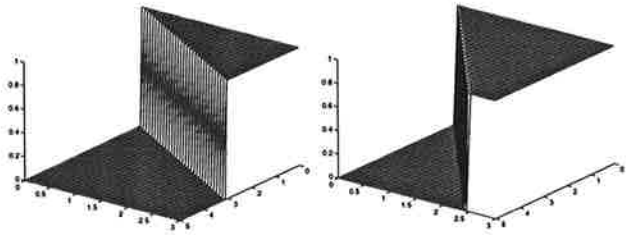
Initial condition: $\rho(0,x) = \theta(x)$ (step function)

Solution becomes $\rho(t,x) = \theta(x - vt)$

- The *density wave moves* with the speed v.
- The *density profile is unchanged* when passing through the tube.

Waves IV

Mass Balance cont.



A wave propagates with the speed 0.5 and 1. (time: $0-3$, space: $0-5$)

Systems of PDE:s

System with first-order (linear) PDE:s ("state space description")

$$\frac{\partial u(t,x)}{\partial t} = G\frac{\partial u(t,x)}{\partial x}$$

The solution to a first-order system consists of n waves - one from each eigenvector of G

- $u(t,x)$ is a vector.
- G is a quadratic matrix.

Systems of PDE:s

Gas Dynamics in a Tube

Assumptions:

- no friction losses,
- pressure linearly proportional to density, $\Delta p = c^2 \Delta \rho$.

The mass and momentum balances are

$$\frac{\partial \rho}{\partial t} = -\rho_0 \frac{\partial u}{\partial x}$$
$$\rho_0 \frac{\partial u}{\partial t} = -c^2 \frac{\partial \rho}{\partial x}$$

u is the velocity

Rewrite on matrix form:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ u\end{bmatrix} = \begin{bmatrix}0 & \rho_0 \\ -\frac{c^2}{\rho_0} & 0\end{bmatrix}\frac{\partial}{\partial x}\begin{bmatrix}\rho \\ u\end{bmatrix}$$

The eigenvalues are $\pm c$ with the eigenvectors $\begin{bmatrix}\frac{\rho_0}{c} \\ 1\end{bmatrix}$ and $\begin{bmatrix}-\frac{\rho_0}{c} \\ 1\end{bmatrix}$.

Mixed PDE:s

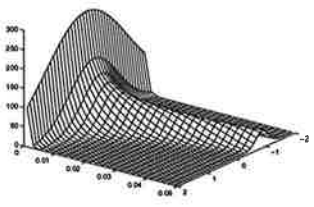
Convection-Diffusions Problem

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + \frac{1}{P}\frac{\partial^2 u}{\partial x^2}$$

The solution travels downstream as it diffuses.

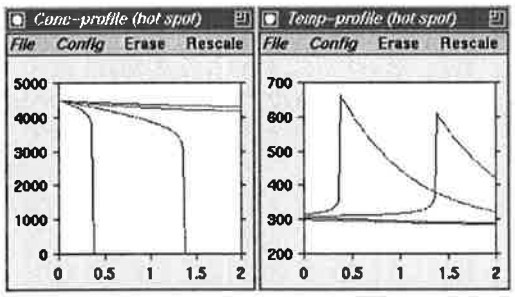
Fundamental solution:

$$u(t,x) = \frac{\sqrt{P}}{2\sqrt{\pi t}} e^{-\frac{(x+t)^2 P}{4}t}$$



Nonlinear PDE:s

Exothermic tube



An example of a exothermic plug flow tubular reactor. Note: x-axis is the space dimension.

- Small changes in the boundary conditions create drastic changes in the static profile.
- Chaotic behaviour?
- Hard to solve in the general case!
- Relyable simulation tools?

Conclusion

- Diffusion and heat equation are parabolic PDE:s.
- Parabolic PDE destroy and smooth out the initial condition.
- Mass and momentum balances are wave equations or hyperbolic PDE:s.
- Waves move the initial conditions through out the space without destroying it.
- (Very) nonlinear PDE:s are hard to handle.

PROCESS MODELLING

lecture VI

Discrete Time Systems

System Identification

Kalman Filter

Content

- Difference equations,
- Eigenvalues and stability,
- Discretization methods,
- Stochastic representations,
- Parameter Estimation,
- Linear Regression,
- System Identification,
- Kalman Filter.

Difference Equations I

State Space Form

Linear difference equation:

$$x(t+h) = \Phi x(t) + \Gamma u(t)$$
$$y(t) = Cx(t) + Du(t)$$

The new state is a function of the old state and the old input. t is the currant time and h is the sampling interval.

Nonlinear difference equation:

$$x(t+h) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t))$$

Difference Equations II

Input/Output Form

Linear difference equation:

$$x(t+h) = \Phi x(t) + \Gamma u(t)$$
$$y(t) = Cx(t) + Du(t)$$

Introduce the shift operator q with the property $qx(t) = x(t+h)$ and $q^{-1}x(t) = x(t-h)$. Eliminate the state vector and express the output y in the input u.

$$qx(t) = \Phi x(t) + \Gamma u(t)$$
$$(qI - \Phi)x(t) = \Gamma u(t)$$
$$x(t) = (qI - \Phi)^{-1}\Gamma u(t)$$
$$y(t) = (C(qI - \Phi)^{-1}\Gamma + D)u(t)$$

This can be rewritten as two polynomials:

$$y(t) = \frac{B(q)}{A(q)}u(t)$$

or as

$$A(q)y(t) = B(q)u(t)$$

Eigenvalues

Stability

As in the continuous case the eigenvalues of the system matrix has a meaning:

$$\Phi z = z\lambda \Rightarrow \det(\lambda I - \Phi) = 0$$

The stability for a difference equation are

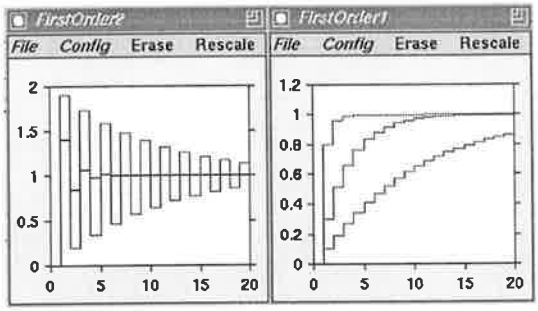
- Eigenvalues inside the unit disc or $|\lambda| < 1$ are stable.
- Eigenvalues outside the unit disc or $|\lambda| > 1$ are unstable.
- Eigenvalues on the unit disc or $|\lambda| = 1$ oscillates.

Stability I

First Order Example

Assume the following first order difference equation:

$$x(t+h) = \Phi x(t) + \Gamma u(t)$$
$$y(t) = Cx(t) + Du(k)$$



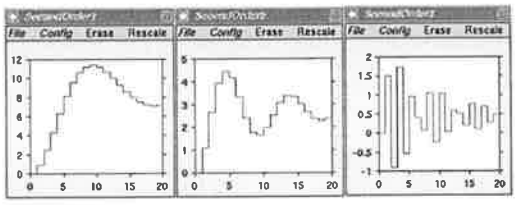
1. Negative real eigenvalues, $\lambda = -0.9, -0,4$.
2. Positive real eigenvalues, $\lambda = 0.9, 0,7, 0.2$.

Stability II

Second Order Example

Assume the following second order difference equation:

$$x(t+h) = \begin{bmatrix} a_{11} & 1 \\ a_{21} & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$



1. Eigenvalues are $\approx 0.85 \pm 0.3i$
2. Eigenvalues are $\approx 0.7 \pm 0.6i$
3. Eigenvalues are $\approx -0.8 \pm 0.4i$

Discretization Methods I

Exact Discretization

A continuous linear system on state space form:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Assume that the input signal is constant between the sampling points and solve the ODE over the sampling interval.

$$x(t+h) = e^{A(t+h-t)}x(t) + \int_t^{t+h} e^{A(t+h-s')}Bu(s')ds'$$

$$x(t+h) = e^{Ah}x(t) + \int_t^{t+h} e^{A(t+h-s')}ds'Bu(t)$$

The exact difference equation for the linear system is then

$$x(t+h) = \Phi x(t) + \Gamma u(t)$$
$$\Phi = e^{Ah}$$
$$\Gamma = \int_t^{t+h} e^{A(t+h-s')}ds'B = \int_0^h e^{As}dsB$$

Discretization Methods II

Time Discretization

The general problem can not be discretized exact. Instead must the time derivative be approximated. The simplest approximation is *forward Euler*:

$$\frac{dx(t)}{dt} = ax(t) \quad \Rightarrow \quad \frac{x(t+h)-x(t)}{h} = ax(t)$$

Rewrite to a difference equation form

$$x(t+h) = x(t) + ahx(t) = (1+ah)x(t)$$

Stability for Euler approximation $|1+ah| < 1$ or $h < \frac{1}{a}$.

Other approximations methods:

- Backward Euler, $\frac{x(t+h)-x(t)}{h} = ax(t+k)$.
- Trapezoidal rule (Tustin), $\frac{x(t+h)-x(t)}{h} = a\frac{x(t+k)+x(t)}{2}$.
- many more advanced approximations .

Discretization Methods III

Exact Discretization Example

First order continuous time model: (mixing tank)

$$\begin{aligned}\dot{c} &= \frac{q}{V}(c_{in}-c) = -\frac{q}{V}c + \frac{q}{V}c_{in} \\ &= Ac + Bc_{in}\end{aligned}$$

Discrete time model: (exact discretization of the mixing tank)

$$\begin{aligned}c(t+h) &= \Phi c(t) + \Gamma c_{in}(t) \\ \Phi &= e^{Ah} = e^{-\frac{q}{V}h} \\ \Gamma &= \int_0^h e^{As}ds = \frac{1}{A}\left[e^{As}\right]_0^h = \frac{1}{A}(e^{Ah}-1) \\ &= \frac{V}{q}(1-e^{-\frac{q}{V}h})\end{aligned}$$

- System is sampling rate, $\frac{1}{h}$, dependent.

Discretization Methods IV

Approximation Example

First order continuous time model: (mixing tank)

$$\begin{aligned}\dot{c} &= \frac{q}{V}(c_{in}-c) = -\frac{q}{V}c + \frac{q}{V}c_{in} \\ &= Ac + Bc_{in}\end{aligned}$$

Euler discretized model: $\dot{c} \approx \frac{c(t+h)-c(t)}{h}$

$$\begin{aligned}c(t+h) &\approx c(t) + \frac{qh}{V}(c_{in}(t)-c(t)) \\ &= (1-\frac{qh}{V})c(t) + \frac{qh}{V}c_{in}(t)\end{aligned}$$

Difference Equations III

Input/Output Description example

Mixing tank:

$$\begin{aligned}c(t+h) &= \Phi c(t) + \Gamma c_{in}(t) \\ y(t) &= Cc(t)\end{aligned}$$

Eliminate c using the shift operator q.

$$\begin{aligned}qIc &= \Phi c + \Gamma c_{in} \\ (qI-\Phi)c &= \Gamma c_{in} \\ y_c &= C(qI-\Phi)^{-1}\Gamma c_{in}\end{aligned}$$

This results in the following

$$\begin{aligned}y_c(t) &= 1(q-e^{-\frac{q}{V}h})^{-1}(1-e^{-\frac{q}{V}h})c_{in}(t) \\ &= \frac{b}{q-a}c_{in}(t) \\ &= \frac{B(q)}{A(q)}c_{in}(t)\end{aligned}$$

Stochastic Difference Equation I

Input/Output Models

Instead of an input signal to the difference equation we **add noise**, $e(t)$.

AR-model, (Auto regression):

$$A(q)y(t) = e(t)$$

MA-model, (Moving average):

$$y(t) = C(q^{-1})e(t)$$

ARMAX-model, (X for extra input u):

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

OE-model, (Output error model):

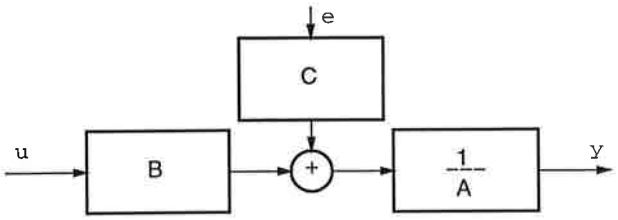
$$y(t) = \frac{B(q)}{F(q)}u(t) + e(t)$$

BJ-model, (Box-Jenkins):

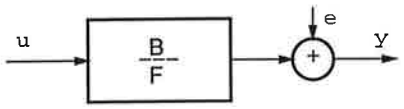
$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

Stochastic Difference Equation II

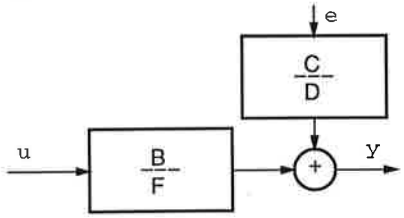
Graphical interpretations



ARMAX: same dynamics for e and u



Output-Error: noice added to the measurement



Box-Jenkins: different dynamics for e and u

Stochastic Difference Equation III

Mixing Tank Example

$$c(t+h) = e^{-\frac{q}{V}h}c(t) + (1 - e^{-\frac{q}{V}h})c_{in}(t)$$
$$y(t) = c(t)$$

Noise on the measurement: OE-model

$$c(t+h) = ac(t) + bc_{in}(t)$$
$$y(t) = c(t) + e(t)$$
$$y(t) = \frac{b}{q-a}c_{in}(t) + e(t)$$

Noise on the input: ARMAX-model

$$c(t+h) = ac(t) + b(c_{in}(t) + e(t))$$
$$y(t) = c(t)$$
$$(q-a)y(t) = bc_{in}(t) + be(t)$$

Conclusions

Discrete Time Models

- *Difference equations* corresponds to differential equations.
- Stable difference equations have eigenvalues inside the *unit disc*.
- Differential equations can be turned into discrete time difference equations:
  - linear systems can exactly be discretized.
  - nonlinear systems can only be approximated.
- Stochastic difference equations add noise to the system description.
- ARMAX model is a stochastic difference equation with one deterministic and one stochastic input.

Parameter Estimation

1. *Tailor-made models* are based on physical modelling. Estimation of parameters with physical interpretation.
2. *Ready-made models* are general and problem independent (black-box models) and are often stochastic difference equations.
3. *Physical experiment* based estimation are problem, technology and application dependent (not addressed here)

Linear Regression I

Linear Regression Model

Linear discrete time system:

$$A(q)y(t) = B(q)u(t)$$
$$q^n y_t + a_1 q^{n-1} y_t + ... + a_n y_t =$$
$$q^m b_1 u_t + ... + b_m u_t$$
$$y_t = -a_1 y_{t-1} - ... - a_n y_{t-n}$$
$$+ b_1 u_{t+m-n} + ... + b_m y_{t-n}$$

Rewrite as a linear regression model:

$$y_p = \theta^T \varphi$$

Parameter vector:

$$\theta = [a_1 \quad a_2 \quad ... \quad a_n \quad b_1 \quad ... \quad b_m]^T$$

Regression vector:

$$\varphi =$$
$$[-y_{t-1} \quad ... \quad -y_{t-n} \quad u_{t-k} \quad ... \quad u_{t-k-m+1}]$$

Parameter Estimation I

Linear regression method

Prediction error

$$\varepsilon = y - y_p = y - \theta^T \varphi$$

The sum of prediction errors:

$$V_N = \frac{1}{N}\sum \varepsilon^2 = \frac{1}{N}\sum (y - y_p)^2$$
$$= \frac{1}{N}\sum (y^2 - 2\theta^T f_N + \theta^T R_N \theta)$$
$$= \frac{1}{N}\sum (y^2 - f_N^T R_N^{-1} f_N$$
$$+ (\theta - R_N^{-1} f_N)^T R_N (\theta - R_N^{-1} f_N)$$

where R_N (matrix) and f_N (column vector) are

$$R_N = \frac{1}{N}\sum \varphi\varphi^T \qquad f_N = \frac{1}{N}\sum \varphi y$$

The prediction error are minimized by the following choice:

$$\theta = R_N^{-1} f_N = (\varphi\varphi^T)^{-1}\varphi y$$

Parameter Estimation II

Iteration method

Other types of models, rewritten as $g(x) = 0$, are estimated by the used on iteration methods like Newton-Raphson.

$$x_{i+1} = x_i - \mu [g'(x_i)]^{-1} g(x_i)$$

x is decreased by the function g divided by the Jacobian of g. μ is a step length parameter.

It is used to solve

$$\frac{d}{d\theta} V_N(\theta) = 0$$

Linear Regression III

Mixing Tank Example

Assume noise on the input resulting in an ARMAX model

$$(q+a)y_t = bc_{in_t} + e(t)$$
$$y_t = ay_{t-1} + bc_{in_{t-1}} + e_t$$

Linear regression formulation:

$$y_t = \theta^T\varphi = [a \quad b]\begin{bmatrix} -y_{t-1} \\ c_{in_{t-1}} \end{bmatrix}$$

This result in the following parameter estimation:

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix} = (\varphi\varphi^T)^{-1}\varphi y$$

where

$$\varphi = \begin{bmatrix} -y_{t-1} & -y_{t-2} & \cdots \\ c_{in_{t-1}} & c_{in_{t-2}} & \cdots \end{bmatrix} \qquad y = \begin{bmatrix} y_t \\ y_{t-1} \\ \cdots \end{bmatrix}$$

Model Properties

- *Bias error* is the convergence error of the estimation. The estimation converge to the *best* estimation if the number of data increase. The best estimation is not the true estimation.
- *Variance error* is the variance of the parameter estimation. The variance error decrease with increased number of data.
- *Identifiability* means that there by occur algebraic constraints between parameters. Can not be estimated on the same time.

System Identification I

Working procedure

- *Input signal* must contain all needed frequencies. Persistently excitation of the system
- *Sampling interval* can not be to short and not to long. Recommendation is 4-8 sample on the step response.
- *Post treatment:* remove outliers and trends.
- *Model structure:* tailor-made or ready-made model? What kind of ready-made models, ARMAX, BJ etc.?
- *Model validation:* model "stability" for different data and identification methods.

System Identification II

Basic concepts

S: is the *system* which is a mathematical description of the process.

M: is the *model structure* for which parameters are estimated.

I: is the *identification method* used in the parameter estimation.

X: is the *experimental conditions* used to produce data.

These four basic concepts can be changed in different directions independently of each other

Conclusions

System Identification

- Good experimental data.
- Choice of model structure:
  - tailor-made, (physical modelling),
  - ready-made, examples are ARX, ARMAX, OE and BJ.
- Identification methods:
  - numerical demanding,
  - MATLAB toolbox.

Kalman Filter I

State Estimation

Assume a linear difference equation

$$x_{t+1} = \Phi x_t + \Gamma u_t$$
$$y_t = C x_t$$

A perfect model of the system can be used to estimate the states of the system. The guarantee that the estimates converge we have to feedback the estimation error through some gain coefficients.

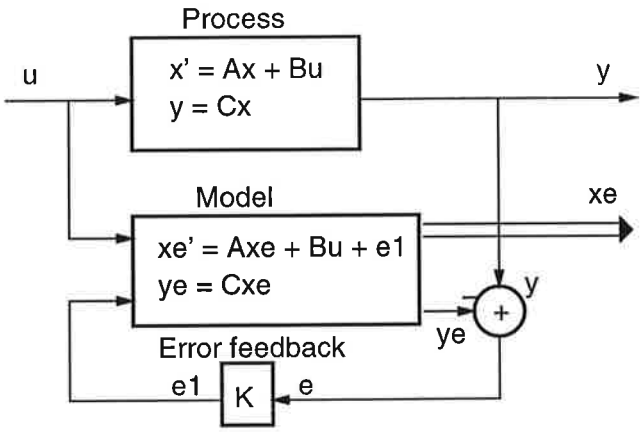
$$x_{e_{t+1}} = \Phi x_{e_t} + \Gamma u_t + K(y_t - y_{e_t})$$
$$y_{e_t} = C x_{e_t}$$

The state estimation error will be

$$\begin{aligned}\varphi_{t+1} &= x_{t+1} - x_{e_{t+1}} \\ &= \Phi x_t + \Gamma u_t - \Phi x_{e_t} - \Gamma u_t - KC(x_t - x_{e_t}) \\ \varphi_{t+1} &= (\Phi - KC)\varphi_t\end{aligned}$$

If $\Phi - KC$ has its eigenvalues inside the unit disc the estimation error goes to zero.

Kalman Filter II



- Model-based control,
- Model-based sensors,
- Supervisory control,
- Parameter estimation.

Kalman Filter III

Parameter Estimation

Assume the following process model:

$$\theta_{t+1} = \theta_t$$
$$y_t = \varphi^T \theta_t + e(t)$$

A Kalman filter will look like:

$$\theta_{e_{t+1}} = \theta_{e_t} + K(y_t - y_{e_t})$$
$$y_{e_t} = \varphi^T \theta_{e_t}$$

- Recursive parameter estimation,
- Real-time identification,

Conclusions

Kalman Filter

- Kalman filter has many names
  - state reconstruction,
  - state observer,
  - estimation filter.
- Deterministic models
  - pole placement of the observer gains.
- Stochastic models
  - Gain K calculated from noise models,
  - the Riccati equation,
  - both stationary and time varying K:s.
- Kalman filter can be used both for
  - state estimation by the used of a good model.
  - parameter estimation by the used of a model structure.

PROCESS MODELLING

lecture VII

Simulation

Model Approximations

Content

- Algebraic equation systems,
  - Linear equations,
  - Sparse matrices,
  - Nonlinear equations.
- Continuous time simulation,
  - explicit and implicit methods,
  - multistep methods,
  - Runge-Kutta methods,
  - Stiffness.
- PDE simulations,
  - Discretization in space and time,
  - Finite difference and finite element methods.
  - Method of lines.
- Other Model Approximations

Linear Equations

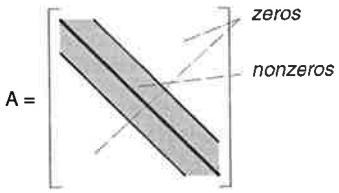
Direct Method

$$Ax = b$$

1. LU-factorization and Gauss elimination results in $A = P^{-1}LU$
2. solve $Lz = Pb$ (forward)
3. solve $Ux = z$ (backward)

- (MATLAB: $x = A\backslash b$)

Sparse Matrices



- many zero elements
- *sparse matrix technique*
  store and handle only nonzeros
- nonzero density, nnz
- (MATLAB can handle sparse matrices)

Nonlinear Equations

Newton's Method

$$g(x_1, ..., x_n) = 0$$

Taylor expansion equal to zero

$$g(x^{i+1}) \approx g(x^i) + g'(x^i)(x^{i+1} - x^i) = 0$$
$$g'(x^i)(x^{i+1} - x^i) = -g(x^i)$$
$$x^{i+1} = x^i - g'(x^i)^{-1} g(x^i)$$

- Iterative procedure
- $A^i = g'(x^i)$ and $b^i = g'(x^i)x^i - g(x^i)$ each iteration becomes: $A^i x^{i+1} = b^i$
- A^i is the Jacobian
- modified Newton methods
  - Other Jacobian calculations
  - Step size control

Continuous Time Simulation

Content

- Explicit and implicit methods,
- Multistep methods,
- Runge-Kutta methods,
- Stiffness.

Explicit Methods

Euler Approximation

Make a Euler approximation of the derivative of a continuous system:

$$\frac{dx}{dt} = -ax$$
$$\frac{x_{t+1} - x_t}{h} = -ax_t$$
$$x_{t+1} = (1 - ah)x_t$$

- Stability: $|1 - ah| < 1$ or $h < \frac{2}{a}$,
- Complexity: one function evaluation, $x_{t+1} = x_t + hf(x_t)$.

Implicit Methods I

Backward Euler

Make a backward approximation of the derivative of a continuous system:

$$\frac{dx}{dt} = -ax$$
$$\frac{x_{t+1} - x_t}{h} = -ax_{t+1}$$
$$(1 + ah)x_{t+1} = x_t$$
$$x_{t+1} = \frac{1}{1 - ah} x_t$$

- Stability: $\frac{1}{|1-ah|} < 1$ valid for all h. *A-stability*

Implicit Methods II

Backward Euler cont.

In the general nonlinear case we can not make an explicit formula of the new state.

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{x_{t+1} - x_t}{h} &= f(x_{t+1}) \\ x_{t+1} - hf(x_{t+1}) - x_t &= 0 \\ F(x_{t+1}) &= 0 \end{aligned}$$

- Nonlinear equation in each time step.
- Newton based method
  $x^{i+1} = x^i - [\, F(x^i)' \,]^{-1} F(x^i)$.
- Complexity; one function evaluation + one jacobian generation.

Multistep Methods I

Adam methods

Approximate the RHS with a weighted sum of old function evaluations from previous steps.

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{x_{t+1} - x_t}{h} &= c_1 f(x_t) + c_2 f(x_{t-1}) \end{aligned}$$

- 2-step explicit method,
- multistep explicit method are called *Adam methods*,
- increase accuracy without increase of complexity (only use of old data),
- decrease of stability regions, smaller step sizes.

Multistep Methods II

Gear Methods

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{x_{t+1} - x_t}{h} &= c_0 f(x_{t+1}) + c_1 f(x_t) + c_2 f(x_{t-1}) \end{aligned}$$

- 3-step implicit method,
- multistep implicit methods are called *Gear methods*,
- better stability to the prize of increased stability

Predictor-Corrector method:

P: use explicit method to predict x^*_{t+1},

E: use x^*_{t+1} to evaluate $f^*(x^*_{t+1})$,

C: use $f^*(x^*_{t+1})$ in the implicit method to correct x_{t+1} (Newton method).

Runge-Kutta Methods I

A One-step Method

Make better approximation of the RHS at current step.

$$\begin{aligned} \frac{dx}{dt} &= f(x) \\ \frac{x_{t+1} - x_t}{h} &= \frac{1}{2}(f(x_t) + f(x_t + hf(x_t))) \end{aligned}$$

- one-step method,
- order-2,
- error estimate of order 3,
  $e = y - y_e = -\frac{h^2}{2} f_y f - \frac{h^3}{4} f_{yy} f f$,
- error estimate is used for step size control.
- Runge-Kutta 23-method.
  - standard MATLAB: `ode23`, `ode45`
  - SIMNON: `Dopri45`, `RKF45`

Stiffness

$$\dot{x} = Ax$$

- A has eigenvalues with different magnitudes.
- Step size is controlled by the largest eigenvalue, i.e. fastest mode.
- Small stability regions means many small time steps, *(simulation takes forever)*.
- **Use implicit methods.**

DAE-Methods

$$\frac{dx}{dt} = f_1(x) \quad ; \quad 0 = f_2(x)$$
$$\frac{x_{t+1} - x_t}{h} = f_1(x_{t+1}) \quad ; \quad 0 = f_2(x_{t+1})$$
$$x_{t+1} - hf_1(x_{t+1}) - x_t = 0 \quad ; \quad f_2(x_{t+1}) = 0$$
$$F(x_{t+1}) = 0$$

- Nonlinear equation in each step,
- Modified implicit method, DASSL.
- Often better to make a symbolic manipulation of the DAE:s into ODE-form, e.g. Dymola.

Conclusions

- Explicit methods has lower complexity,
- Implicit methods has better stability,
- Multistep methods increase accuracy to the prize of stability,
- Runge-Kutta methods increase accuracy for 1-step methods,
- Stiffness means that there are different time scales in the system. Use implicit methods.
- Large sparse problems require special implementations of these solvers.

PDE Approximations

Content

- Discretization in space and time,
- Finite difference and finite element methods.
- Method of lines.

Finite Difference Approximation I

Space Discretization

$$\frac{\partial^2 u}{\partial x^2} - f(u) = 0$$

Approximate the function in space with a set of points.

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\frac{\partial u}{\partial x}_{i+1} - \frac{\partial u}{\partial x}_{i}}{h}$$

$$\approx \frac{\frac{u_{i+1}-u_i}{h} - \frac{u_i-u_{i-1}}{h}}{h} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$= \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & . & & \\ & & & . & 1 \\ & & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ u_n \end{bmatrix} = \begin{bmatrix} f(h) \\ f(2h) \\ . \\ . \\ f(nh) \end{bmatrix}$$

Finite Difference Approximation II

Space Discretization cont.

The finite difference approximation results in

$$Au = b$$

- Large linear equation system
- Sparse matrix problem
- Can be used in 2 and 3 dimensions
- Generation of the grid points
- Automatic regridding
  - moving grid points
  - generation of new grid points

Finite Element Approximation I

Function Approximation

Approximation of the function with a set of piecewise polynomials (pp-approximation)

$$u(x) = \Sigma_{j=1}^{m} a_j \Phi_j(x)$$

Piecewise polynomials (Basis functions):

- linear $w_j = \frac{x - x_{j-1}}{x_{j+1} - x_j}$
- Hermite cubic $g_j = 2x^3 + 3x^2$
- B-spline

Dominating methods (to find a):

- Collocation (approx. in a set of points)
- Galerkin (approx. of the function)

Finite Element Approximation II

Collocation Method

$$\frac{\partial^2 u}{\partial x^2} - f(u) = 0$$

The pp-approximation must be valid in the collocation points, $i = 1..m$.

$$\frac{\partial^2 u}{\partial x^2} - f(u) = 0$$
$$u''(x_i) - f(x_i) = 0$$
$$\Sigma_{j=1}^{m} a_j \Phi_j''(x_i) - f(x_i) = 0$$

This can be rewritten as

$$A^c a = f^c$$
$$A^c_{ij} = \Phi_j''(x_i)$$
$$a = [\, a_1 \quad a_2 \quad \ldots \quad a_m \,]^T$$
$$f = [\, f(x_1) \quad f(x_2) \quad \ldots \quad f(x_m) \,]^T$$

- Galerkin also results in a linear equation system $A^G u = f^G$.

Method of Lines I

Parabolic PDE can be handled in the following way:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} - v\frac{\partial u}{\partial x} + f(u)$$

- Make a discretization
  - finite difference in space *or*
  - finite element of solution

  $\Rightarrow$ a set of ODE:s
- Solve with an ODE-solver
  - Stiff problem
  - Sparse matrices

  $\Rightarrow$ sparse implicit solver

Method of Lines II

Dynamic Finite Difference

Approximate the PDE with a set of ODE:s in time

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u)$$
$$\frac{du_i}{dt} = \frac{D}{h^2}(u_{i+1} - 2u_i + u_{i-1}) + f(u_i)$$

Rewrite on matrix form

$$\frac{du}{dt} = \frac{D}{h^2}Au + f$$
$$u = [\,u_1 \quad u_2 \quad . \quad . \quad u_n\,]^T$$
$$A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & . & & \\ & & & . & 1 \\ & & & 1 & -2 \end{bmatrix}$$
$$f = [\,f(u_1) \quad f(u_2) \quad . \quad . \quad f(u_n)\,]^T$$

Method of Lines III

Dynamic Finite Difference cont.

The boundary conditions must be fulfilled. Assume Dirichlet condition

$$u(t,0) = u0(= x_0)$$
$$u(t,L) = uL(= x_{n+1})$$

Space difference with n grid points

$$\frac{du_1}{dt} = \frac{D}{h^2}(u_2 - 2u_1 + u0) + f(u_1)$$
$$\frac{du_n}{dt} = \frac{D}{h^2}(uL - 2u_n + u_{n-1}) + f(u_n)$$

On matrix form (u, A and f as above)

$$\frac{du}{dt} = \frac{D}{h^2}Au + f + b$$
$$b = \frac{D}{h^2}[\,u0 \quad 0 \quad \ldots \quad 0 \quad uL\,]^T$$

Boundary conditions:

- Dirichlet add a b-vector
- Neumann also change the A-matrix

Method of Lines IV

Dynamic Finite Difference cont.

Trapezoidal approximation of the time derivative

$$\frac{du}{dt}_{t+\frac{h}{2}} = \frac{u_{t+h} - u_t}{h}$$

The space discretization must also be based on the midpoints.

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u)$$
$$\frac{u_{i,t+h} - u_{i,t}}{h} = \frac{D}{h^2}\Big(\frac{u_{i+1,t+h} + u_{i+1,t}}{2} - 2\frac{u_{i,t+h} + u_{i,t}}{2} + \frac{u_{i-1,t+h} + u_{i-1,t}}{2}\Big) + f\Big(\frac{u_{i,t+h} + u_{i,t}}{2}\Big)$$

- **Crank-Nicolson** method
- implicit problem

Method of Lines V

Dynamic Finite Element

Approximate the solution with trial functions in space and parameters in time

$$u(t,x) = \Sigma_{j=1}^{m} a_j(t)\Phi_j(x)$$

Use it to approximate the parabolic PDE in a set of points (collocation)

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + f(u)$$
$$\Sigma(a'\Phi) = D\Sigma(a\Phi'') + f(\Sigma(a\Phi))$$

This results in the following nonlinear ODE:s

$$Aa'(t) = F(t,a)$$
$$A = \Phi_j(x_i)$$
$$a = [\,a_1 \quad a_2 \quad . \quad . \quad a_n\,]^T$$
$$F = [\,f(\Sigma a\Phi(x_1)) \quad . \quad . \quad f(\Sigma a\Phi(x_n))\,]^T$$

Conclusions

- *Finite difference* approximates the PDE into a set of *points.*
- *Finite element* approximates the PDE into a set of *trial functions.*
- Both FD and FE result in linear equation systems
- Method of lines, **MOL**:
  - make a FD or FE
  - solve the initial value problem (ODE)
  - parabolic PDE:s
- Waves or Hyperbolic PDE:s are not stable
- MOL should not be used for waves
- Efficient code requires regridding

Model Approximations

Summary

- Space discretization of PDE into ODE,
- PDE-approximations into harmonics (relaxation)
- Linearization of nonlinear to linear,
- Time discretization of continuous to discrete,
- Model reduction (state reduction)
  - Balanced realizations

PROCESS MODELLING

Content

1. Physical Modelling
2. Computer Aided Modelling
3. Linear Models
4. Nonlinear Models
5. Distributed Parameter Models
6. Discrete Time Models
7. Stochastic Discrete Time Models
8. System Identification
9. Model Approximations in simulation

PROCESS MODELLING

What is **not** a part of the course

- Tensor based models
- Stochastic models
- Nonparametric models
  - Spectrum and FFT (Fast Fourie Transform)
  - Frequency response models
- Fuzzy logic based models
- Neural net based models
- Discrete event models
- etc.

PROCESS MODELLING

the END

(only the exam left)

Process Modelling

Lecture 2 - problems: Physical Modelling

2.1) Make problem 2.1 in Luyben, page 38. List your assumptions. (Component mass balances over a continuous stirred tank reactor, CSTR.)

2.2) A stirred tank (with no reaction) has two inflows, F1 and F2, and one outflow, F3. The tank is heated with a electrical heater, Q. Assume that the flows and the tank volume are constant. Describe the liquid temperature in the tank. List your assumptions. (see example 2.6 in Luyben, page 23).

2.3) Make problem 2.4 in Luyben, page 38. List your assumptions. (Energy balance over a plug flow tubular reactor, PFTR.)

2.4) Substrate, S, is pumped into a fed-batch reactor where it is consumed by the biomass, X. The biomass growth is described by Monod kinetics and the consumption of S is proportional to the biomass growth. Put up a model describing the volume of the reactor and the concentrations of X and S. List your assumptions.

2.5) An ice cube is dropped into a hot,perfectly mixed, insulated cup of coffee. Develop the equations describing the dynamics of the system. List all assumptions and define all terms. (Problem 3.9 in Luyben)

Last update: April 24, 1995.

Bernt Nilsson
bernt@control.lth.se

Process Modelling

Lecture 3 - problems: Computer Aided Modelling

3.1) Study the gas-phase, pressurized CSTR on in section 3.5, p. 45-46 in Luyben.

a) Write down the dynamic model of the system.

b) Make a direct problem formulation (dynamic simulation). Which variables are known parameters (time invariant), known variables and unknown variables?

c) Write a MATLAB function (or calculation procedure) that can be used to simulate the system.

3.2) Make a SIMULINK model (or block diagram description), using predefined modules, of the tank reactor series presented in section 3.2 in Luyben, p 41-43. Hints: transform the differential equations into transfer functions and draw a block diagram. Match with SIMULINK modules.

3.3) Put up a bond graph for the two heated tanks in section 3.4 in Luyben, p 44-45.

3.4) The single-component vaporizer in section 3.7 in Luyben, p. 51-54, is described by four different models. Model C is based on a rate equation and Model D is based on equilibrium. Assume equilibrium, T(liquid) = T(vapour). Rewrite the DAE-system into a ODE-system, i.e. verify Model D.

3.5) Make the problem 3.12 on page 80 in Luyben.

a) Model a semibatch reactor with evaporation (do not model the pressure and the temperature controllers).

b) The reactor is assumed to be isothermal. How will the model look like if this assumption is changed?

c) If we model the pressure dynamics in the vapour and describes the evaporation with a flux expression then we get yet another differential equation. If we describes it with an equilibrium expression instead then we get an algebraic equation system. Show it!

Last update: April 27, 1995.
Bernt Nilsson
bernt@control.lth.se

Process Modelling

Lecture 4 - problems: Linear Models

4.1) Make a model of two tank reactors in series (two CSTR). Assume constant flows and volume in the CSTRs and first order kinetic. List your other assumptions. (See the tank series in section 3.2 in Luyben)

a) Write your model on matrix based state space form. Use the concentrations as states.

b) Calculate the eigenvalues and eigenvectors of the corresponding system matrix.

c) How does the dynamic response depend on the reactor volumes? Discuss.

d) Scetch the phase plane.

4.2) Make a model of the gas-phase, pressurized CSTR discussed in secion 3.5 in Luyben, p 45-46. (See also lecture-problem 3.1)

a) Calculate the steady state.

b) Calculate the Jacobian of the system and make a linearization. Write down the linearized model.

c) How does the linear model dynamics depend on the steady state.

Last update: May 3, 1995.

Bernt Nilsson
bernt@control.lth.se

Process Modelling

Lecture 5 - problems: Nonlinear Models and Distributed Models

---

5.1) A heated tank has a constant volume of liquid. The tank is heated by an electrical heater with the effect Q.

a) Set up an energy balance over the tank. Assume constant feed temperature and variable feed flow. Hints: see Energy Balance I in lecture 2 and First Order System II in lecture 4a.

b) Calculate the steady state and linearize the system.

c) Calculate the time constant and the steady state gain, K = T/qin.

d) Scetch the time responses for two different feed flow changes. The feed flow is changed as a step function from qin to 1.1*qin. The first step with qin= q and the second with qin=q/2.

5.2) Study a vertial heated water tube. Assume constant flow through the tube.

a) Make proper assumption and put up a physical model and rewrite it as a system description. Hint: Energy balances II and III in lecture 2.

b) What kind of PDE do you get in subproblem **a)**?

c) What kind of PDE do we get if we remove the external added heat?

d) What kind of PDE do we get if we remove the forced flow?

e) What kind of PDE do we get if we remove the heat conduction in space dimension?

f) What kind of PDE do we get if we remove the time dynamics?

g) What happens if the water starts to boil? Discuss! Model?

Last update: May 16, 1995.
Bernt Nilsson
bernt@control.lth.se

Process Modelling

Lecture 6 - problems: Discrete Time Models and Identification

---

6.1)

Hot liquid enters a buffer tank with the volume 2 m^3. The flow through the tank is assumed to be constant, q = 0.01. A simple model is a first order system, T' = (q/V) (Tin - T).

a) Make an exact discretization of the system.

b) Calculate the parameters for two different sampling intervals. Choose the sampling intervall 40 and 2.

c) A simple rule says that it is proper to sample 4-8 times on a step response. What does this mean in this case?

6.2)

a) Assume that the measurement of the tank temperature is noicy, e.g., add noise on the measurement y. Write down the input/output representation. What kind of model is this?

b) Assume that the temperature in the inflow is noicy, e.g., add noise on the input Tin. Write down the input/output representation. What kind of model is this?

Last update: May 23, 1995.
Bernt Nilsson
bernt@control.lth.se

Process Modelling

Lecture 7 - problems: Simulation and Model Approximation

7.1)

Lets go back ones again to the mixing tank with constant hold-up, c' = q/V(cin - c).

a) Make a forward Euler approximation of the linear mixing tank model. What is the largest time step for a stable approximation.

b) Make a backward Euler approximation and calculate the stability.

c) Two mixing tanks in series can be described as a linear system, c' = A c + B cin. Assume that the second tank has a volume that is 50 times larger then the first one. Make an forward Euler approximation and calculate the stability margin of the time step.
(Hint: stiff problem)

7.2) extra problem for the interested

From lecture 2, slide Mass Balance IV, a tubular reactor model is developed. Assume constant velocity, v, first order reaction, r=-kc and isothermal conditions then we get dc/dt = D d^2c/dx^2 - v dc/dx - k c.

a) Make a finite difference approximation of the space dependency and write it as a MOL problem (method of lines). For simplicity make a two point grid (x0=0, x1=L/3, x2=2L/3, x3=L) where x0 and x3 are the boundary points.
Hints: d^2c/dx^2 = (c(x_i+1) - 2c(x_i) + c(x_i-1))/h^2 and dc/dx = (c(x_i+1) - c(x_i-1))/h

b) Choose a Dirichlet boundary condition, c(x_0) = a and c(x_3) = b. Use it in the discretization of the boundaries in a). What is a proper boundary condition for this problem, Dirichlet, Neumann or Robin?

c) This becomes a second order linear system on state space form. Calculate the eigenvalues.

Last update: May 31, 1995.
Bernt Nilsson
bernt@control.lth.se

Process Modelling

Exercise 1: Physical Modelling Thursday 4/5, 10-12, Lutetia

This first exercise in the course is focused upon physical modelling but first of all short description how to use MATLAB and SIMULINK.

The following chemical reaction occure in a batch reactor; *A -> B -> C*

1) Model the component balances in the reactor and write down the differential equations for component A and B. Assume:

1. Homogenous mixing,
2. Isothermal conditions,
3. Atmospheric pressure,
4. Constant volume,
5. No other side reactions, and
6. 1:st order kinetics.

---

2) MATLAB

a) Write a MATLAB function with the calculation procedure for the simulation problem for the model in 1). Example:

```
function xder = batch(t,x)
x(1) = ...;
x(2) = ...;
```

b) Simulate the reactor.

- Parameters: k1 = k2 = 1.
- Initial value: cA(0) = 1, cB(0) = 0.

```
[t,x] = ode23('batch',0,3,[1,0]);
plot(t,x)
```

c) Make simulations with different parameters and inital values and study the difference.

---

3) SIMULINK

a) Describe the model in 1) as a block diagram with transfer functions.

b) Use the SIMULINK libraries to generate a graphical block diagram of the batch reactor model.

c) Make corresponding simulation as in 2).

d) Are there other ways to describe the same problem in SIMULINK? Discuss!

---

4) Physical Modelling

A batch reactor model is discussed in section 3.9, p. 57-62, in Luyben. A case study is also found in section 5.7, p. 150-157 (not in hand-outs).

- initial value of A: cA0 = 0.8 lb mol / ft^3
- first reaction: *r1 = a1*e^(-E1/(R*T))*
  - a1 =729.55 min^-1
  - E1 =15 000 Btu/(lb mol)
  - R =1.9858 Btu/(lb mol F)
  - Reaction heat: L1 = -40 000 Btu/(lb mol)
- second reaction: *r2 = a2*e^(-E2/(R*T))*
  - a2 =6567.6 min^-1
  - E2 =20 000 Btu/(lb mol)
  - Reaction heat: L2 = -50 000 Btu/(lb mol)
- reactor volume: V = 42.5 ft^3
- jacket volume: Vj = 18.83 ft^3
- metal volume: Vm = 9.42 ft^3
- liquid: Cp =1 Btu/(lb F)
- jacket: Cpj =1 Btu/(lb F)
- metal: Cm =0.12 Btu/(lb F)
- liquid density =50 lb/ft^3
- jacket density =62.3 lb/ft^3
- metal density =512 lb/ft^3
- Heat transfer: liquid to wall, *qi = Ai hi (Tw - Tj)*
  - Ai = 56.5 ft^2
  - hi = 160 Btu/(h F ft^2)
- Heat transfer: wall to jacket, *qos = Aos hos (T - Tw)*
  - Aos = 56.5 ft^2
  - hos = 1000 Btu/(h F ft^2)
- Jacket inflow temperature: Tj0 = 80 F

a) Make a SIMULINK model (MATLAB or other simualtion tool) of the batch reactor with with the physical data above. Assume isothermical conditions (T = 250 F).

b) Change the assumption 2 and assume instead variable temperature in the reactor. The reactor is cooled by a jacket (Tj = 250 F). In other words describe the dynamic energy balance over the reactor. Assume constant temperature in the jacket and no metal wall accumulation. List all other assumptions. Add your energy description to the model in a).

c) Describe also the energy balance over the jacket. List all assumptions. Add the jacket energy balance to the SIMULINK model. What is a proper coolant flow through the jacket.

d) Assume also that the metal wall of the reactor accumulate heat.

---

5) Project Part

a) Make the same simulation of the batch reactor which is done in Luyben, section 5.7. Heating of the reactor by steam to start the reaction then filling the jacket with coolant and then cool the reactor by controlling the flow.

b) Change the 4th assumption and assume variabel volume. In other words model a dynamic mass balance. Component A enters the reactor under a time intervall, x. (Make a semi-batch sequence).

c) Change the 5:th and 6:th assumptions and add side-reactions and nonlinear kinetic expressions.

d) Nonhomogenous mixing and discuss other assumptions that can be violated in the model.

Last update: May 3, 1995.
Bernt Nilsson
bernt@control.lth.se

TABLE 5.11 (*continued*)

TIME	N	X	Y	T	P	V	L	M
2.000	0	.00058	.00162	106.4	245.5	16.9	12.1	89.5
	1	.00119	.00331	105.8	240.3	16.9	29.0	7.2
	2	.00217	.00605	105.0	235.1	16.8	28.9	7.2
	3	.00376	.01047	104.3	229.8	16.8	28.9	7.2
	4	.00633	.01759	103.5	224.6	16.7	28.8	7.2
	5	.01046	.02891	102.6	219.4	16.6	28.8	7.2
	6	.01700	.04662	101.5	214.1	16.5	28.7	7.2
	7	.02720	.07354	100.3	208.9	16.4	28.6	7.2
	8	.04267	.11282	98.7	203.6	16.3	28.5	7.2
	9	.06514	.16670	96.9	198.4	16.2	28.4	7.2
	10	.09584	.23467	94.6	193.2	16.1	28.3	7.2
	11	.13443	.31182	92.1	188.0	16.0	28.2	7.2
	12	.17812	.38954	89.5	182.7	16.0	28.1	7.2
	13	.22206	.45890	86.9	177.5	15.9	28.0	7.2
	14	.26125	.51442	84.6	172.3	17.3	28.0	7.2
	15	.26106	.51479	83.9	168.1	17.3	11.4	5.4
	16	.26204	.51672	83.1	163.8	17.3	11.4	5.4
	17	.26541	.52184	82.3	159.5	17.4	11.4	5.4
	18	.27368	.53326	81.2	155.2	17.4	11.4	5.4
	19	.29164	.55651	79.8	150.8	17.4	11.5	5.4
	20	.32778	.59958	77.8	146.4	17.5	11.5	5.4
	21	.39443	.66915	74.8	142.0	17.7	11.6	5.5
	22	.50184	.76054	70.8	137.5	17.9	11.7	5.6
	23	.64212	.85199	66.3	133.0	18.3	12.0	5.7
	24	.78094	.92078	62.3	128.4	18.6	12.3	5.8
	25	.88386	.96172	59.3	123.6	18.8	12.7	5.9
	26	.94433	.98261	57.1	118.8	19.0	12.9	6.0
	27	.97492	.99239	55.5	113.9	19.1	13.0	6.0
	28	.98917	.99676	54.2	108.9	19.1	13.1	6.1
	29	.99553	.99868	52.9	103.9	19.2	13.2	6.1
	30	.99831	.99950	51.6	98.7	19.2	13.2	6.1
		.99950	.99986	49.4	90.0	5.9	13.2	65.0

5.7 BATCH REACTOR

Let us consider the batch reactor modeled in Sec. 3.9 (Fig. 3.9). Steam is initially fed into the jacket to heat up the system to temperatures at which the consecutive reactions begin. Then cooling water must be used in the jacket to remove the exothermic heats of the reactions.

The output signal of the temperature controller goes to two split-ranged valves, a steam valve and a water valve. The instrumentation is all pneumatic, so the controller output pressure P_c goes from 3 to 15 psig. The valves will be adjusted so that the steam valve is wide open when the controller output pressure P_c is at 15 psig and is closed at $P_c = 9$ psig (i.e., half the full range of the controller output). The water valve will be closed at $P_c = 9$ psig and wide open at $P_c = 3$ psig. The reason for hooking up the valves in this manner is to have the correct fail-safe action in the event of an instrument air failure. The steam valve takes air pressure to open it and therefore it will fail closed. We call this an "air-to-open" (AO) valve. On the other hand, the water valve takes air pressure to close it and therefore it will fail open. This is an "air-to-close" (AC) valve. If an emergency occurs, we want to remove the source of heat (steam) and go to full cooling.

Controller output range (psig)	3	9	15
Steam valve fraction open X_s:		0 (closed)	1 (open)
Water valve fraction open X_w:	1 (open)	0 (closed)	

The equations for the reaction liquid inside the tank and the vessel metal are

$$\frac{dC_A}{dt} = -k_1 C_A \tag{5.35}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B \tag{5.36}$$

$$\frac{dT}{dt} = \frac{-\lambda_1}{\rho C_p} k_1 C_A - \frac{-\lambda_2}{\rho C_p} k_2 C_B - \frac{Q_M}{V \rho C_p} \tag{5.37}$$

$$Q_M = h_i A_i (T - T_M) \tag{5.38}$$

$$\frac{dT_M}{dt} = \frac{Q_M - Q_J}{\rho_M C_M V_M} \tag{5.39}$$

The equations for the jacket are different for the three phases of the batch cycle.

A. With steam in the jacket (35 psia supply pressure steam):

$$V_J \frac{d\rho_s}{dt} = w_s - w_c \tag{5.40}$$

$$\rho_s = \frac{MP_J}{R(T_J + 460)} \tag{5.41}$$

$$P_J = \exp\left(\frac{A_{vp}}{T_J + 460} + B_{vp}\right) \tag{5.42}$$

$$w_s = C_{Vs} X_s \sqrt{35 - P_J} \tag{5.43}$$

$$Q_J = -h_{os} A_{os} (T_J - T_M) \tag{5.44}$$

$$w_c = -\frac{Q_J}{H_s - h_c} \tag{5.45}$$

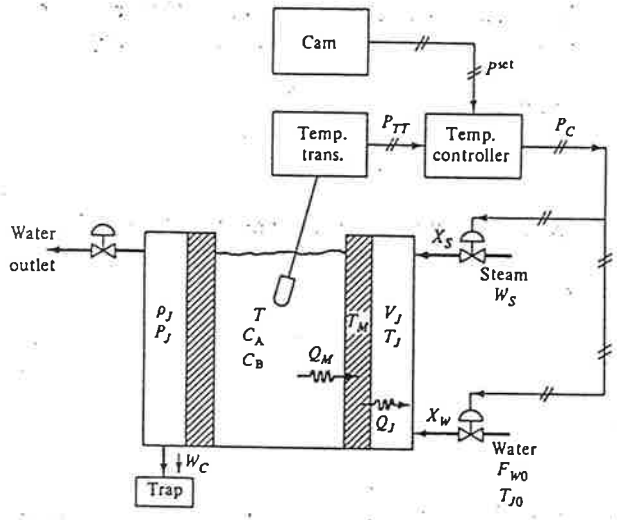


FIGURE 5.7
Batch reactor.

B. During filling with water (20 psig water header pressure):

$$A_o = \left(\frac{A_o}{V_J}\right)_{\text{total}} V_J \tag{5.46}$$

$$\frac{dV_J}{dt} = F_{WO} \tag{5.47}$$

$$\frac{d(V_J T_J)}{dt} = F_{WO} T_{JO} + \frac{Q_J}{\rho_J C_J} \tag{5.48}$$

$$Q_J = h_{ow} A_o (T_M - T_J) \tag{5.49}$$

$$F_{WO} = C_{Vw} X_w \sqrt{20} \tag{5.50}$$

C. When the jacket is full of water:

$$\frac{dT_J}{dt} = \frac{F_{WO}}{V_J}(T_{JO} - T_J) + \frac{Q_J}{C_J V_J \rho_J} \tag{5.51}$$

The system is sketched in Fig. 5.7, and numerical values of parameters are given in Table 5.12. The digital program is given in Table 5.13. Plotted results are shown in Fig. 5.8.

The temperature transmitter has a range of 50 to 250°F, so its output pneumatic pressure signal goes from 3 psig at 50°F to 15 psig at 250°F.

$$P_{TT} = 3 + (T - 50)\frac{12}{200} \tag{5.52}$$

TABLE 5.12
Parameters for batch reactor

α_1	729.55 min^{-1}	V_J	18.83 ft^3
α_2	6567.6 min^{-1}	C_{Vw}	100 gpm/$psi^{0.5}$
E_1	15,000 Btu/lb·mol	T_{JO}	80°F
E_2	20,000 Btu/lb·mol	A_l	56.5 ft^2
A_{vp}	−8744.4°R	λ_1	−40,000 Btu/lb·mol
B_{vp}	15.70	λ_2	−50,000 Btu/lb·mol
C_{AO}	0.80 lb·mol A/ft^3	C_p	1 Btu/lb_m °F
T_0	80°F	V	42.5 ft^3
K_c	10 psi/psi	ρ	50 lb_m/ft^3
C_{Vs}	112 lb_m/min $psi^{0.5}$	C_M	0.12 Btu/lb_m °F
h_{os}	1000 Btu/h °F ft^2	V_M	9.42 ft^3
h_{ow}	400 Btu/h °F ft^2	ρ_M	512 lb_m/ft^3
h_i	160 Btu/h °F ft^2	ρ_J	62.3 lb_m/ft^3
A_O	56.5 ft^2	C_J	1 Btu/lb_m °F
$H_s - h_c$	939 Btu/lb_m		

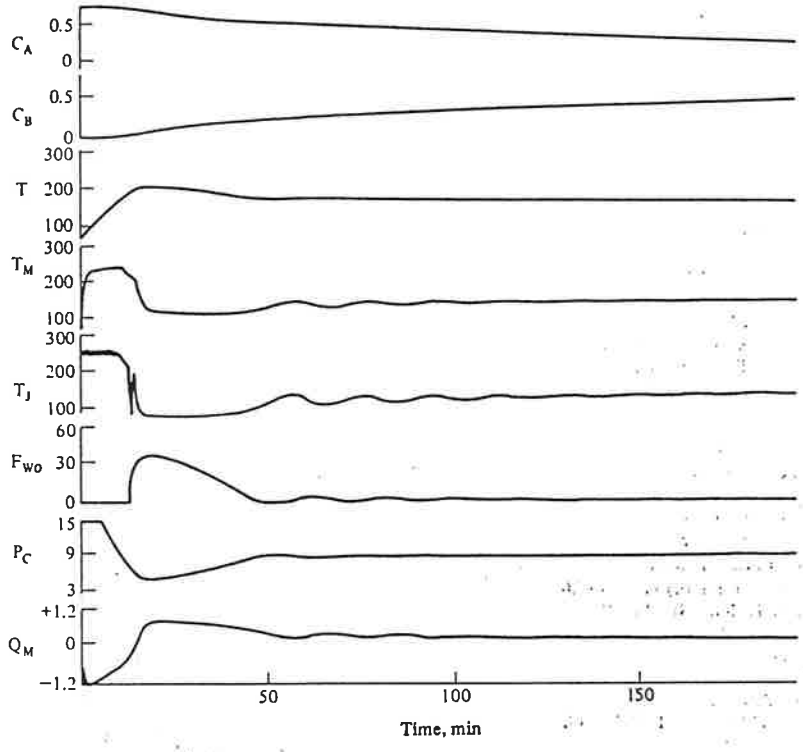


FIGURE 5.8
Plotted results for batch reactor.

TABLE 5.13
Batch reactor simulation

```
      REAL K1,K2,KC
      ALPHA1=729.5488
      ALPHA2=6567.587
C VAPOR PRESSURE CONSTANTS FOR STEAM
      AVP=-8744.4
      BVP=15.70036
C USE SMALL DELTA DURING STEAM PERIOD
      DELTA=.002
      KC=2.
      TPRINT=1.
C INITIAL CONDITIONS
      CA=0.8
      CB=0.
      TIME=0.
      T=80.
      TM=80.
      TJ=259.
      PJ=34.4
      DENS=18.*PJ*144./(1545.*(TJ+460.))
      PSET=12.6
      START=1.
      FWO=0.
      FULL=-1.
      VJ=0.
      VJTJ=0.
      TFLAG=0.
      CAM=-1.
      RAMP =.005
      WRITE(6,54)
   54 FORMAT(' TIME    CA     CB    XS  XW    T     TM    TJ    FWO
     +  QJ    QM')
C  MAIN LOOP FOR EACH STEP IN TIME
  100 K1=ALPHA1*EXP(-15000./(1.99*(T+460.)))
      K2=ALPHA2*EXP(-20000./((T+460.)*1.99))
C   TRANSMITTER
      PTT=3.+(T-50.)*12./200.
C  CONTROLLER
      PC=7.+KC*(PSET-PTT)
      IF(PC.GT.15.) PC=15.
      IF(PC.LT.3.) PC=3.
C   VALVES
      XS=(PC-9.)/6.
      XW=(9.-PC)/6.
      IF(XS.GT.1.) XS=1.
      IF(XS.LT.0.) XS=0.
      IF(XW.GT.1.) XW=1.
      IF(XW.LT.0.) XW=0.
C   TEST FOR STEAM
      IF(START.LT.0.) GO TO 20
      IF(PJ.GE.35.) GO TO 40
      WS=XS*112.*SQRT(35.-PJ)
      GO TO 41
   40 WS=0.
   41 CONTINUE
      QJ=-1000.*56.5*(TJ-TM)/60.
      WC=-QJ/939.
      DENDOT=(WS-WC)/18.83
      DENS=DENS+DELTA*DENDOT
C ITERATIVE LOOP TO CALCULATE STEAM TEMPERATURE
```


TABLE 5.13 (*continued*)

```
C     AND PRESSURE FROM KNOWN DENSITY USING
C     INTERVAL HALVING
      FLAGP=-1.
      FLAGM=-1.
      DTJ=1.
      LOOP=0
   15 PJ=EXP(BVP+AVP/(TJ+460.))
      LOOP=LOOP+1
      IF(LOOP.GT.50)  GO TO 70
      DCALC=18.*PJ*144./(1545.*(TJ+460.))
      IF(ABS(DENS-DCALC).LT. .0011) GO TO 50
      IF(DENS.GT.DCALC) GO TO 17
      IF(FLAGM.LT.0.) GO TO 16
      DTJ=DTJ/2.
   16 TJ=TJ-DTJ
      FLAGP=1.
      GO TO 15
   70 WRITE(6,71)
   71 FORMAT(1X,'STEAM TEMP LOOP')
      STOP
   17 IF(FLAGP.LT.0.) GO TO 18
      DTJ=DTJ/2.
   18 TJ=TJ+DTJ
      FLAGM=1.
      GO TO 15
   20 FWO=100.*SQRT(20.)*8.33*XW/62.3
      WS=0.
      WC=0.
      DENS=0.
      PJ=0.
      XS=0.
C   TEST FOR JACKET FILLING
      IF(FULL.GT.0.) GO TO 30
      AO=VJ*56.5/18.83
      VJ=VJ+DELTA*FWO
      IF(VJ.GE. 18.83) FULL=1.
      QJ=400.*AO*(TM-TJ)/60.
      VJTJ=VJTJ+DELTA*(FWO*80.+QJ)
      IF(VJ.LE.0.) GO TO 25
      TJ=VJTJ/VJ
      GO TO 50
   25 TJ=80.
      GO TO 50
C   FULL JACKET
   30 QJ=400.*56.5*(TM-TJ)/60.
      TPRINT=2.
C USE BIGGER DELTA ONCE JACKET IS FULL OF WATER
      DELTA=.05
C  EVALUATE DERIVATIVES
      TJDOT=FWO*(80.-TJ)/18.83+QJ/(18.83*62.3)
      TJ=TJ+DELTA*TJDOT
   50 CADOT=-K1*CA
      CBDOT=K1*CA-K2*CB
      QM=160.*56.5*(T-TM)/60.
      TDOT=(K1*CA*40000.+K2*CB*50000.)/50.-QM/(42.4*50.)
      TMDOT=(QM-QJ)/(512.*.12*9.42)
C   INTEGRATION
      TIME=TIME+DELTA
      CA=CA+CADOT*DELTA
```

TABLE 5.13 (*continued*)

```
      CB=CB+CBDOT*DELTA
      T=T+TDOT*DELTA
      TM=TM+TMDOT*DELTA
      IF(T.GT.300.) STOP
      IF(T.GT.200.) START=-1.
      IF(T.GT.200.) CAM=1.
      IF(CAM.GT.0.) PSET=PSET-DELTA*RAMP
      IF(TIME.GT.100.) GO TO 56
      IF(TIME.LT.TFLAG) GO TO 100
      WRITE(6,55)TIME,CA,CB,XS,XW,T,TM,TJ,FWO,QJ,QM
   55 FORMAT(1X,F5.1,2F7.3,2F5.2,4F6.1,2E10.2)
      TFLAG=TFLAG+TPRINT
      GO TO 100
   56 STOP
      END
```

Results

TIME	CA	CB	XS	XW	T	TM	TJ	FWO	QJ	QM
0.0	0.800	0.000	1.00	0.00	80.0	80.6	251.0	0.0	-0.17E+06	0.00E+00
1.0	0.799	0.001	1.00	0.00	86.3	211.9	265.0	0.0	-0.43E+05	-0.19E+05
2.0	0.799	0.001	1.00	0.00	96.4	234.2	261.0	0.0	-0.27E+05	-0.21E+05
3.0	0.798	0.002	1.00	0.00	106.7	239.0	260.0	0.0	-0.22E+05	-0.20E+05
4.0	0.797	0.003	1.00	0.00	116.7	240.8	262.0	0.0	-0.22E+05	-0.19E+05
5.0	0.796	0.005	1.00	0.00	126.3	242.6	261.0	0.0	-0.19E+05	-0.18E+05
6.0	0.794	0.006	1.00	0.00	135.7	243.8	265.0	0.0	-0.21E+05	-0.16E+05
7.0	0.792	0.008	0.97	0.00	144.7	244.9	260.0	0.0	-0.16E+05	-0.15E+05
8.0	0.789	0.011	0.80	0.00	153.5	245.9	262.0	0.0	-0.12E+05	-0.14E+05
9.0	0.787	0.014	0.62	0.00	162.1	246.7	263.0	0.0	-0.11E+05	-0.13E+05
10.0	0.783	0.017	0.46	0.00	170.6	247.3	259.0	0.0	-0.12E+05	-0.12E+05
11.0	0.779	0.021	0.29	0.00	178.9	248.0	259.0	0.0	-0.10E+05	-0.10E+05
12.0	0.775	0.026	0.12	0.00	187.3	248.8	259.0	0.0	-0.96E+04	-0.93E+04
13.0	0.769	0.031	0.00	0.04	195.6	243.4	244.0	0.0	-0.54E+03	-0.72E+04
14.0	0.763	0.037	0.00	0.20	203.2	232.6	212.7	11.9	0.17E+04	-0.44E+04
15.0	0.756	0.044	0.00	0.34	210.4	222.9	203.2	20.1	0.51E+04	-0.20E+04
17.0	0.741	0.059	0.00	0.53	219.7	159.1	98.5	31.8	0.23E+05	0.89E+04
19.0	0.724	0.076	0.00	0.59	222.4	132.9	88.5	35.3	0.17E+05	0.13E+05
21.0	0.707	0.092	0.00	0.61	223.0	127.1	86.9	36.2	0.15E+05	0.14E+05
23.0	0.691	0.108	0.00	0.61	222.9	125.8	86.6	36.3	0.15E+05	0.15E+05
25.0	0.675	0.123	0.00	0.60	222.5	125.5	86.5	36.0	0.15E+05	0.15E+05
27.0	0.659	0.138	0.00	0.59	221.8	125.3	86.6	35.4	0.15E+05	0.15E+05
29.0	0.644	0.152	0.00	0.58	220.9	125.1	86.7	34.5	0.14E+05	0.14E+05
31.0	0.630	0.166	0.00	0.55	219.6	124.9	86.9	33.2	0.14E+05	0.14E+05
33.0	0.616	0.178	0.00	0.53	218.0	124.7	87.1	31.5	0.14E+05	0.14E+05
35.0	0.603	0.191	0.00	0.49	216.1	124.4	87.5	29.5	0.14E+05	0.14E+05
37.0	0.591	0.202	0.00	0.45	214.0	124.1	87.9	27.1	0.14E+05	0.14E+05
39.0	0.579	0.213	0.00	0.41	211.5	123.8	88.5	24.4	0.13E+05	0.13E+05
41.0	0.568	0.223	0.00	0.36	208.9	123.6	89.3	21.4	0.13E+05	0.13E+05
43.0	0.558	0.232	0.00	0.30	206.0	123.4	90.4	18.2	0.12E+05	0.12E+05
45.0	0.549	0.241	0.00	0.25	203.0	123.5	91.9	14.7	0.12E+05	0.12E+05
47.0	0.540	0.249	0.00	0.19	199.9	123.9	94.0	11.2	0.11E+05	0.11E+05
49.0	0.531	0.257	0.00	0.13	196.8	124.9	97.2	7.7	0.10E+05	0.11E+05
51.0	0.523	0.264	0.00	0.07	193.9	126.8	101.9	4.4	0.94E+04	0.10E+05
53.0	0.516	0.270	0.00	0.03	191.3	130.2	109.1	1.5	0.80E+04	0.92E+04
55.0	0.509	0.277	0.00	0.00	189.3	135.8	119.6	0.0	0.62E+04	0.81E+04
57.0	0.503	0.282	0.00	0.00	188.3	142.5	129.0	0.0	0.51E+04	0.69E+04
59.0	0.496	0.288	0.00	0.00	188.1	148.5	137.0	0.0	0.43E+04	0.60E+04
61.0	0.490	0.294	0.00	0.00	188.7	153.9	143.9	0.0	0.38E+04	0.53E+04

TABLE 5.13 (*continued*)

63.0	0.483	0.300	0.00	0.01	189.9	158.6	148.9	0.7	0.36E+04	0.47E+04
65.0	0.477	0.305	0.00	0.05	191.4	159.1	143.9	2.8	0.57E+04	0.48E+04
67.0	0.470	0.311	0.00	0.07	192.5	153.8	132.9	4.3	0.78E+04	0.58E+04
69.0	0.464	0.317	0.00	0.08	192.6	146.8	123.7	4.7	0.87E+04	0.69E+04
71.0	0.457	0.322	0.00	0.07	191.8	141.9	119.4	3.9	0.85E+04	0.75E+04
73.0	0.451	0.328	0.00	0.04	190.5	140.1	119.9	2.5	0.77E+04	0.76E+04
75.0	0.445	0.333	0.00	0.02	189.1	141.3	124.3	1.1	0.65E+04	0.72E+04
77.0	0.439	0.338	0.00	0.00	188.0	145.2	131.8	0.0	0.51E+04	0.65E+04
79.0	0.434	0.343	0.00	0.00	187.7	150.4	139.5	0.0	0.41E+04	0.56E+04
81.0	0.428	0.348	0.00	0.00	188.0	155.4	145.7	0.3	0.36E+04	0.49E+04
83.0	0.422	0.352	0.00	0.02	188.8	158.1	146.3	1.5	0.44E+04	0.46E+04
85.0	0.417	0.357	0.00	0.04	189.7	156.6	140.8	2.7	0.59E+04	0.50E+04
87.0	0.411	0.362	0.00	0.06	190.0	152.2	133.4	3.3	0.71E+04	0.57E+04
89.0	0.406	0.366	0.00	0.05	189.6	147.8	128.3	3.1	0.73E+04	0.63E+04
91.0	0.401	0.371	0.00	0.04	188.7	145.3	127.1	2.2	0.69E+04	0.65E+04
93.0	0.395	0.375	0.00	0.02	187.6	145.5	129.7	1.1	0.60E+04	0.64E+04
95.0	0.390	0.379	0.00	0.00	186.8	148.0	135.3	0.3	0.48E+04	0.59E+04
97.0	0.385	0.383	0.00	0.00	186.4	152.3	142.2	0.0	0.38E+04	0.52E+04
99.0	0.381	0.387	0.00	0.01	186.6	156.6	147.2	0.4	0.35E+04	0.45E+04

A proportional feedback controller is used with its output biased at 7 psig (i.e., its output pressure is 7 psig when there is zero error).

$$P_c = 7 + K_c(P^{\text{set}} - P_{TT}) \tag{5.53}$$

The setpoint signal P^{set} comes from a pneumatic function generator. When the process temperature gets up to 200°F the P^{set} signal is ramped slowly downward to prevent too much loss of component B, as discussed in Sec. 3.9.

$$P^{\text{set}} = 12 - \text{RAMP}(t - t_{200}) \tag{5.54}$$

where RAMP = rate of P^{set} change with time, psi/min
t = batch time, min
t_{200} = time when process temperature T reaches 200°F

5.8 TERNARY BATCH DISTILLATION WITH HOLDUP

The model of a multicomponent batch distillation column was derived in Sec. 3.13. For a simulation example, let us consider a ternary mixture. Three products will be produced and two "slop" cuts may also be produced. Constant relative volatility, equimolal overflow, constant tray holdup, and ideal trays are assumed.

Table 5.14 gives a digital computer FORTRAN program for this three-component batch distillation dynamic simulation. The specific example is a column with 20 trays and relative volatilities of 9, 3, and 1. The vapor flow rate is constant at 100 mol/h.

The column starts up on total reflux (no distillate is withdrawn) until the distillate composition reaches the desired purity level. The time at total reflux is called TE in the program. Then distillate is withdrawn at a fixed rate of 40 mol/h.

Process Modelling

Exercise 2: Linear Models and Analysis, Thursday 11/5, 10-12, Lutetia

---

This exercise is focused upon linear models and linear analysis. The numerical problems require the use of MATLAB or corresponding software.

---

1)

Two CSTR is connected in series. They have constant hold-up (constant volume and flow). In the reactor a simple reaction occure $R \rightarrow P$, which is described by first order kinetics. Assume isothermal conditions. (Use the following data: q = V1 = V2 = k1 = k2 = 1)

a) Put up the component mole balances of component R over the reactors. Write the linear model on state space form (using matrices). Assume that the inflow concentration of R is the input signal.

b) Calculate the eigenvalues and eigenvectors for the system matrix A. (both for homogenous case (q = 0) and nonhomogenous (q = 1)).

c) Draw a phase plane plot.

d) How does it change if the reactor volumes are changed and when then reaction coefficient is changed?

---

2)

Use a proportional controller to control the concentration in the second CSTR by the use of the inflow concentration of R.

a) Rewrite the linear model with the control law $Cin = u = K(Cref - C2)$

b) Calculate the eigenvalues for the controlled reactors. Choose K = 1.

c) Draw a phase plane plot. Study also a step response.

Project part.

3)

A cocurrant heat exchanger can be modelled by two PDE:s. The heat transfer between to two sides are modelled with an overall heat transfer expression. One way to approximate the PDE is by the *method of lines*, MOL. The heat exchanger is cut into slices, each slice is composed of one hot side and one cool side. These sides are constant volume compartments.

a) Put up the energy balances over the heat exchanger and writew them as PDE:s.

b) Approximate the PDE by the method of lines and rewrite the model on matrix based state space form. Begin by only making one slice. Simulate the response to changes in the inflow temperatuers.

c) Put a model for three slices. Simulate.

d) Put a model for ten slices. Simulate.

e) Compare the simulations of different approximations. Make linear analysis and study the eigenvalues for models. The change of some assumptions will make the model nonlinear. Which?

Last update: May 10, 1995.
Bernt Nilsson
bernt@control.lth.se

Process Modelling

Exercise 3: Nonlinear Models and Analysis, Thursday 18/5, 10-12, Lutetia

---

This exercise is focused upon nonlinear model, linear analysis and phase plane analysis. The numerical problems require the use of MATLAB or corresponding software.

---

1) In a stirred tank bioreactor microorganisms, X, is produced under the consumption of substrate, S. The specific growth rate of X can be described by a Monod expression, mu = (mu_max * S)/(Km + S); (s^-1). The consumption of S is proportional to the growth rate, muS = k1*mu. Data:

- D = q/V = 0.2 h^-1
- Sin = 5 g/l
- k1 = 2
- Km = 10 g/l
- mu_max = 6.3 h^-1

a) Put up the mass balances over the microorganisms, X, and substrate, S. Assume constant volume, flow, temperature and other conditions (pH, O2, etc.). The feed flow has the concentration Sin of substrate.

b) Calculate steady state.

c) Linearize the system, calculate the Jacobian and its eigenvalues.

d) Draw a phase plane plot.

---

2) Change the specific growth rate and add inhibition, mu = (mu_max * S)/(Km + S + S^2/Ki); (s^-1).

- Ki = 0.1 g/l

b) Calculate the steady states (three).

c) Linearize the system, calculate the Jacobian and its eigenvalues for the different steady states.

d) Draw a phase plane plot. Categorize the different steady states.

Project part.

3)

A CSTR with an exothermic reaction can behave very perculiar. A model based on one mass and one energy balances show up bifurcation phenomena and limit cycles. A PI-controlled CSTR (three states, an extra state from the integral part of the controller) can have chaotic behaviour. The project is to simulate a system like this. Ask me (Bernt) for references to papers with data.

alternative:

A very famous chemical reaction set is called **Belusov-Zabotinskii** reaction. The components of the reactions are cerium sulphate, sodium bromate, malonic acid and sulphuric acid. A simplified model of the kinetics are as follows

x' = x(x - xy + y -qx^2)
y' = 1/s*(-y - xy + fz)
z' = w(x - z)

This model can show up bifurcation, supercritical and subcritical limit cycles and chaotic behaviour (choice of parameters s, q, f, w). The project is to simulate a system like this. Ask me (Bernt) for references to papers with data.

Last update: May 16, 1995.

Process Modelling

Exercise 4: System Identification, Thursday 29/5, 10-12, Lutetia

---

This exercise is focused upon system identification and discrete time systems. The numerical problems require the use of MATLAB with system ideentification toolbox, SITB.

---

1) Simulate different discrete time systems using the idsim command in SITB. The command sequence need for a simualtion of a discrete system with or without noise is found on page 1-59.

a) Simulate the mixing tank example used in the lecture.

b) Add noise to the measurement and simulate.

c) Add noise to the input signal (concentration variations in the inflow) and simulate.

d) Simulate a ARMAX model and a Box-Jenkins model.

---

2) In the SITB maunual there is An Introductory Example on page 1-5 to 1-7.

a) Make a system identification by just follow the introductory example (ARX identification).

b) Test other kinds of models, like an ARMAX model.

Project part:

3)

Select a simple process example and put up a system description based on physical modelling. Simulate it, add noise and take out (noisy) sample data. Make a system identification on the simulated data using SITB. Validate the estimated model against the model with or without noise.

Last update: May 24, 1995.

Bernt Nilsson
bernt@control.lth.se

3. An Introductory Example

A demonstration M-file called `iddemo.m` provides several examples of what might be typical sessions with the System Identification Toolbox. To start the demo, execute `iddemo` from inside MATLAB.

Before giving a formal treatment of the capabilities and possibilities of the toolbox, this example is designed to get you using the software quickly. This example is essentially the same as demo #1 in `iddemo`. You may want to invoke MATLAB at this time, execute `iddemo1`, and follow along.

Data have been collected from a laboratory scale process. (Feedback's Process Trainer PT326; see page 440 in Ljung, 1987.) The process operates much like a common hand-held hair dryer. Air is blown through a tube after being heated at the inlet to the tube. The input to the process is the voltage applied to a mesh of resistor wires that constitutes the heating device. The output of the process is the air temperature at the outlet measured in volts by a thermocouple sensor.

One thousand input-output data points were collected from the process as the input was changed in a random fashion between two levels. The sampling interval is 80 ms. The data were loaded into MATLAB in ASCII form and are now stored as the vectors `y2` (output) and `u2` (input) in the file `dryer2.mat`.

First load the data:

```
load dryer2
```

This example selects the first 300 data points for building a model. For convenience, the input-output vectors are merged into a matrix:

```
z2 = [y2(1:300) u2(1:300)];
```

Take a look at the data,

```
idplot(z2)
```

The toolbox makes frequent use of default arguments. Default values are used when trailing arguments are omitted. In the case above, by default, all data points are graphed and the sampling interval is one time unit.

You can select the values between sample numbers 200 and 300 for a close-up, and at the same time obtain correct time scales, with


```
idplot(z2,200:300,0.08)
```

Remove the constant levels and make the data zero mean with

```
z2 = dtrend(z2);
```

Now, fit to the data a model of the form:

$$y(t) + a_1 y(t-T) + a_2 y(t-2T) = b_1 u(t-3T) + b_2 u(t-4T) \quad (3.1)$$

where T is the sampling interval (here 0.08 seconds). This model, known as an ARX model, tries to explain or compute the value of the output at time t, given previous values of y and u. Later on, especially in Section 10, are discussions of how to decide upon suitable model orders (structures).

The best values of the coefficients a_1, a_2, b_1 and b_2 can be computed with

```
th = arx(z2,[2 2 3]);
```

The numbers in the second argument tell `arx` to find a model (3.1) with two a-parameters, two b-parameters and three delays. The result is stored in the matrix `th` in a somewhat coded form. To specify the actual sampling interval, enter

```
th = sett(th,0.08);
```

There are several ways to display and illustrate the computed model. With

```
present(th)
```

the coefficient values of (3.1) and their estimated standard deviations are presented on the screen.

Next, you might ask how to evaluate how well the model fits the data. A simple test is to run a simulation whereby real input data is fed into the model, and to compare the simulated output with the actual measured output. For this, select a portion of the data that was not used to build the model, for example, from sample 700 to 900:

```
u = dtrend(u2(700:900));
y = dtrend(y2(700:900));
ysim = idsim(u,th);
plot([y(100:200) ysim(100:200)])
```

You see that the model is quite capable of describing the system, even for data that were not used in calculating the fit.

To compute and graph the poles and zeros of the model, use

```
zpth = th2zp(th);
zpplot(zpth)
```

If you want to know the frequency response, you can compute the frequency function of the model and present it as a Bode plot by entering

```
gth = th2ff(th);
bodeplot(gth)
```

Compare this transfer function with a transfer function obtained from a nonparametric, spectral analysis method. Such an estimate is obtained directly from the data

```
gs = spa(z2);
gs = sett(gs,0.08);
```

The sampling interval, 0.08, is set by the second command in order to obtain correct frequency scales. The function spa also allows you to select window sizes, frequency ranges, etc. All these have here been given default values.

You can compare the estimate of the transfer function obtained by spectral analysis and the one obtained from the model (3.1) with

```
bodeplot([gs gth])
```

The agreement is quite good.

Finally, plot the step response of the model. The model comes with an estimate of its own uncertainty. Ten different step responses are computed and graphed. They correspond to "possible" models, drawn from the distribution of the true system (according to our model):

```
step = ones(30,1);
idsimsd(step,th)
```


11. A Typical Session

A typical system identification session involves several attempts with different model structures, followed by comparisons and validation tests. This section gives an example of a computer session, where for simplicity only a few structures are tested. It includes simulation of the data, so you can use the whole sequence of commands to gain familiarity with the System Identification Toolbox . It is also included with the software as a demo and can be run by executing iddemo and selecting case # 2.

Start by forming some simulated data. Here are the numerator, denominator, and noise coefficients of the z-transform model of a system:

```
B = [0 1 0.5];  A = [1 -1.5 0.7];
C = [1 -1 0.2];
```

These can be put into the special theta matrix format used by the toolbox:

```
th0 = poly2th(A,B,C);
```

Now generate an input signal u, a disturbance signal e, and simulate the response of the model to these inputs:

```
rand('normal')
u = sign(rand(300,1));
e = rand(300,1);
y = idsim([u e],th0);
```

Graph the 50 first values of the input u and the output y:

```
idplot([y u],1:50)
```

Now that you have noise corrupted, simulated, data, you can estimate models and make comparisons. Start with correlation analysis to get an idea about the impulse response of the system:

```
z = [y u];
ir = cra(z);
```

Compute an estimate of the frequency function and the noise spectrum using spectral analysis:

```
[GS,NSS] = spa(z);
bodeplot(GS)
```

Use the ARX method to find a model with two poles, one zero, and a single delay on the input:

```
a2 = arx(z,[2 2 1]);
present(a2)
```

Now use instrumental variables (IV4):

```
i2 = iv4(z,[2 2 1]);
present(i2)
```

Calculate transfer functions for the two models obtained by ARX and IV4. Show Bode plots comparing the Spectral Analysis estimate with the ARX and IV4 estimates:

```
[Ga2,NSa2] = th2ff(a2);
Gi2 = th2ff(i2);
bodeplot([GS Ga2 Gi2])
```

Calculate and graph residuals for the model obtained by IV4:

```
e = resid(z,i2);
plot(e), title('Residuals, IV4 method')
```

Now compare ARMAX with Box-Jenkins estimates:

```
am2 = armax(z,[2 2 2 1]);
bj2 = bj(z,[2 2 2 2 1]);
[Gam2,NSam2] = th2ff(am2);
[Gbj2,NSbj2] = th2ff(bj2);
bodeplot([Gam2 Gbj2 Gi2 GS])
bodeplot([NSam2 NSbj2 NSS])
```

Plot residuals for ARMAX and Box-Jenkins:

```
e1 = resid(z,am2);
e2 = resid(z,bj2);
plot([e1 e2])
```


Simulate the ARMAX and Box-Jenkins models with the real input:

```
yham2 = idsim(u,am2);
yhbj2 = idsim(u,bj2);
plot([y yham2 yhbj2])
```

Plot poles and zeros of the ARMAX and Box-Jenkins models:

```
zpam2 = th2zp(am2);
zpbj2 = th2zp(bj2);
zpplot([zpam2 zpbj2])
```

AM2 and BJ2 both give good residuals and simulated outputs. They are also in good mutual agreement. Check which has the smaller FPE:

```
[am2(2,1) bj2(2,1)]
```

Since you generated the data, you can compare the model to the true system:

```
[G0,NS0] = th2ff(th0);
zp0 = th2zp(th0,1);
bodeplot([G0 Gam2])
bodeplot([NS0 NSam2])
zpplot([zp0 zpam2])
```

The System Identification Toolbox comes with a number of data sets in MAT-files `iddata1` to `iddata8`, disk space permitting. You are encouraged to try your hand with these. Write to The MathWorks, Inc. for solutions when you think you are done or when you need help. You can also go through some of the *case studies* that are included among the demos. Execute `iddemo` and select the `cs` demo.

Process Modelling: Exam

---

1. Explain shortly the following terms:

a) Direct and inverse problem formulation.

b) Continuity equations.

c) Phase plane.

d) Deterministic chaos.

e) Parabolic PDE (example).

f) ARMAX model.

g) Stiffness and sparsness

2. A stirred tank (with no reaction) has two inflows, F1 and F2, and one outflow, F3. The tank is heated with a electrical heater, Q. Assume that the flows and the tank volume are constant. Describe the liquid temperature in the tank. List your assumptions. (see example 2.6 in Luyben, page 23).

3. Make problem 2.2 in Luyben, page 38. List your assumptions. (Component balances over a plug flow tubular reactor, PFTR.)

4. Make a SIMULINK model (or block diagram description), using predefined modules, of one heated tank. A modification of section 3.4 in Luyben, p 44-45. Hints: transform the differential equations into transfer functions and draw a block diagram. Match with SIMULINK modules.

5. Make a model of two tank reactors in series (two CSTR). Assume constant flows and volume in the CSTRs and first order kinetic. List your other assumptions. (See the tank series in section 3.2 in Luyben)

a) Write your model on matrix based state space form. Use the concentrations as states.

b) Calculate the eigenvalues and eigenvectors of the corresponding system matrix.

c) How does the dynamic response depend on the reactor volumes? Discuss.

d) Scetch the phase plane.

6. A heated tank has a constant volume of liquid. The tank is heated by an electrical heater with the effect Q.

a) Set up an energy balance over the tank. Assume constant feed temperature and variable feed flow.

b) Calculate the steady state and linearize the system.

c) Calculate the time constant and the steady state gain, K = T/qin.

7. Hot liquid enters a buffer tank with the volume 2 m^3. The flow through the tank is assumed to be constant, q = 0.01. A simple model is a first order system, T' = (q/V) (Tin - T).

a) Make an exact discretization of the system. Calculate the parameters for two different sampling intervals. Choose the sampling intervals 40 and 2.

b) A simple rule says that it is proper to sample 4-8 times on a step response in order to make system identification. What does this mean in this case?

Bernt Nilsson, June 7, 1995.