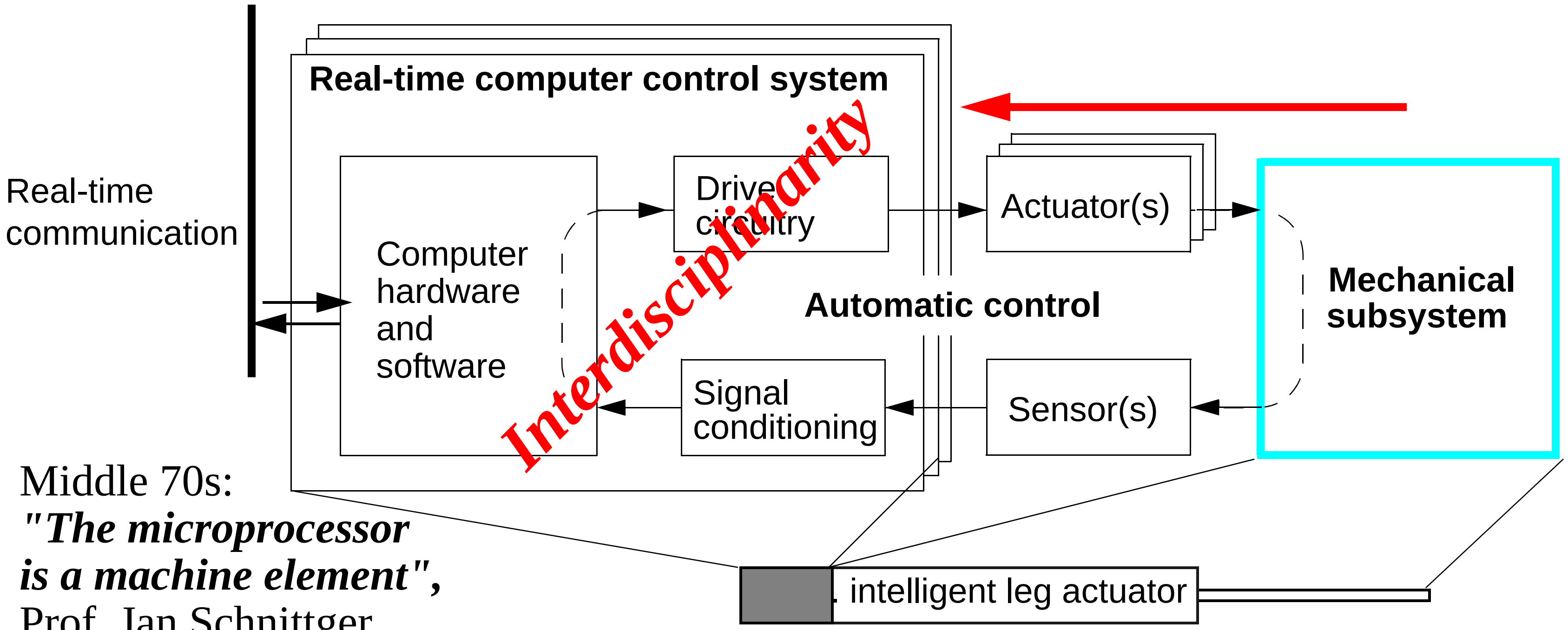
Dynamics and motion control

Lecture 1

Course introduction and overview

Bengt Eriksson
KTH, Department of Machine Design
Division of Mechatronics
e-mail: benke@md.kth.se

Mechatronics at Machine Design, KTH



Middle 70s:
"The microprocessor is a machine element",
Prof. Jan Schnittger

Motivation: multidisciplinarity is required!

Analysis and specifications

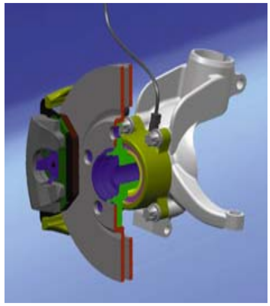
Conceptual design, functions, technical implementation

Mechanics development

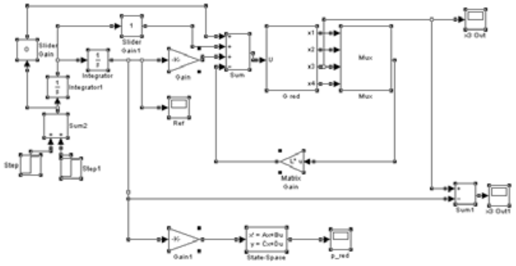
Control function development

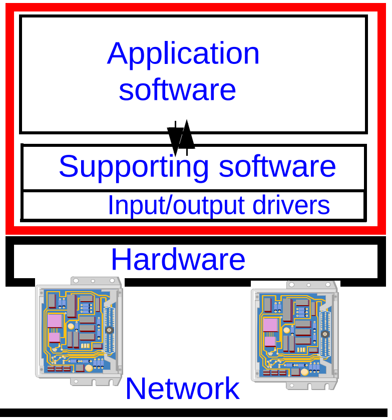
Software-development

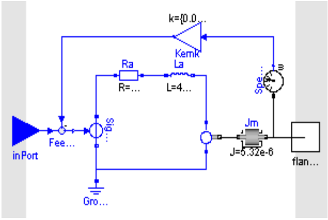
Electronics development



CAD
Dynamics
FEM







Integration

1.1. Lecture outline

- 1. Course goal
- 2. Course implementation
- 3. Overview of and introduction to modelling and control of mechanical systems

1.1.1 Overall course goal

To provide an understanding and the skills of modelling and design of motion control systems;

"from modelling to implementation"

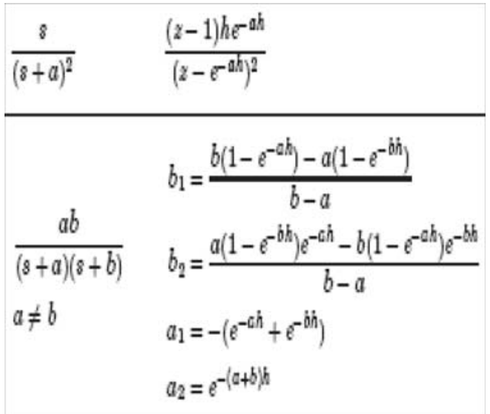
1.1.2 After the course you should be able to

- Specify overall performance requirements for a motion control system
- Derive models of typical mechatronic applications
- Find good parameters of dynamic models in both frequency and time domain
- Design model based feedback and feed forward control both in continuous and discrete time
- Simulate process and control system models in both continuous and discrete time
- Design and structure the controller software for microprocessor implementation
- Understand the limitations and restrictions due to computing speed.

Download and read more in the course PM from course hompage

1.1.3 Further course goals

Learning how your theoretical knowledge from earlier
courses in math, mechanics,
numeric methods and control should best be
used to avoid chaos in real world experiments.



$\frac{s}{(s+a)^2}$	$\frac{(z-1)he^{-ah}}{(z-e^{-ah})^2}$
$\frac{ab}{(s+a)(s+b)}$ $a \neq b$	$b_1 = \frac{b(1-e^{-ah}) - a(1-e^{-bh})}{b-a}$ $b_2 = \frac{a(1-e^{-bh})e^{-ah} - b(1-e^{-ah})e^{-bh}}{b-a}$ $a_1 = -(e^{-ah} + e^{-bh})$ $a_2 = e^{-(a+b)h}$

1.2. Lecture outline

- **1. Course goal**
- **2. Course implementation**
- **3. Overview of and introduction to modelling and control of mechanical systems**

1.2.1 Prerequisites

- **Some mechanical and electrotechnical knowledge**
- **Very litle programming knowledge**
- **Good control knowledge it should corespond to the content of a basic course in control engineering**

1.2.2 Course outline

- **2 two hour lectures per week (except first week -> three)**
  - Basic theory and calculated examples
- **2 three hour per week lab working**
  - Mainly computer simulation work, Matlab/Simulink
  - Woking with Exercises and Workshops
  - Assistance by

1.2.3 Literature and course material

- The course specification
- The reading material
- The lecture notes →
- Standard book in control
- The exercises
- The workshops

L2, modelling

L3, feedback design in continuous time

L4, feedback design in discrete time

L5, Model following control -servo

L6, Implementation on real-time HW

L7, Robustness issues

1.2.4 Examination

- **1. Completed exercises**
  - The exercises are completed by showing the results to an assistant in the lab.

    Exercises part 1, modeling of mechanical systems
    Exercises part 2, modeling of a DC motor system
    Exercises part 3, control of a DC motor system
- **2. Written reports on the workshops, A and B.**
- **3. Written exam.**

Grading: A-F according to ECTS based on a weighted combination of the bullets 2 and 3 above.

1.2.5 Exercises and laboratory facilities

- Working in groups of preferably three people in each group

- Where to work?
  - MMT computer rooms: Matlab/Simulink
  - Mechatronics-lab: ("FIM-labbet") Matlab/Simulink and experiments on real motor.
  - Wennström-lab: ("elektro-labbet") Matlab/Simulink
  - Room A319 is equiped with special HW/SW for RCP
  - Your own computers, KTH CD’n with Matlab/Simulink

- Mechatronics-lab: "passerkort": need card numbers

- The labs are free to use 7/24 if they not are booked for other courses, booking schedule is available from course home page.

1.2.6 Additional useful material

- **The course builds upon earlier courses!**
  - mechanics, machine elements, numerical methods, control engineering, computer science and programming, electrical engineering

- General textbooks in
  - Control engineering -from a first course in automatic control. e.g. *Glad och Ljung: Reglerteknik - Grundläggande teori*
  - Electric circuit theory -from a first course in electric circuits. e.g. Hans Johansson Elektroteknik
  - Mechanical enginering. -from a course in dynamics. e.g. ------

- Selected parts of **"Mechatronics System Design",** *Klaus Janscheck*
  - Available as E-book from the KTH Library

1.2.7 Learning philosophy

- It is about *your learning* - for which *You* is the boss!
- We employ feed forward and feedback to improve the learning process
- *Feed forward* concepts: material, lectures, the project etc.
- *Feedback* mechanisms
  1) Participant representatives: to collect important feedback info
  2) Lecture and workshop follow up questions
  3) Hand ins of project parts
  4) Course evaluation in the end

1.3. Lecture outline

- **1. Course goal**
- **2. Course implementation**
- **3. Overview of and introduction to control of mechanical systems**
  and a brief introduction to tools

1.3.1 Forecasts!

- *"I think there is a world market for about five computers", Tomas J Watson Sr, IBM 1943*
- *"There are no reasons for any individuals to have a computer in their home", Ken Olson, Digital Equipment 1977*
- *"The current rate of progress cannot continue much longer", various computer technologists, 1950*

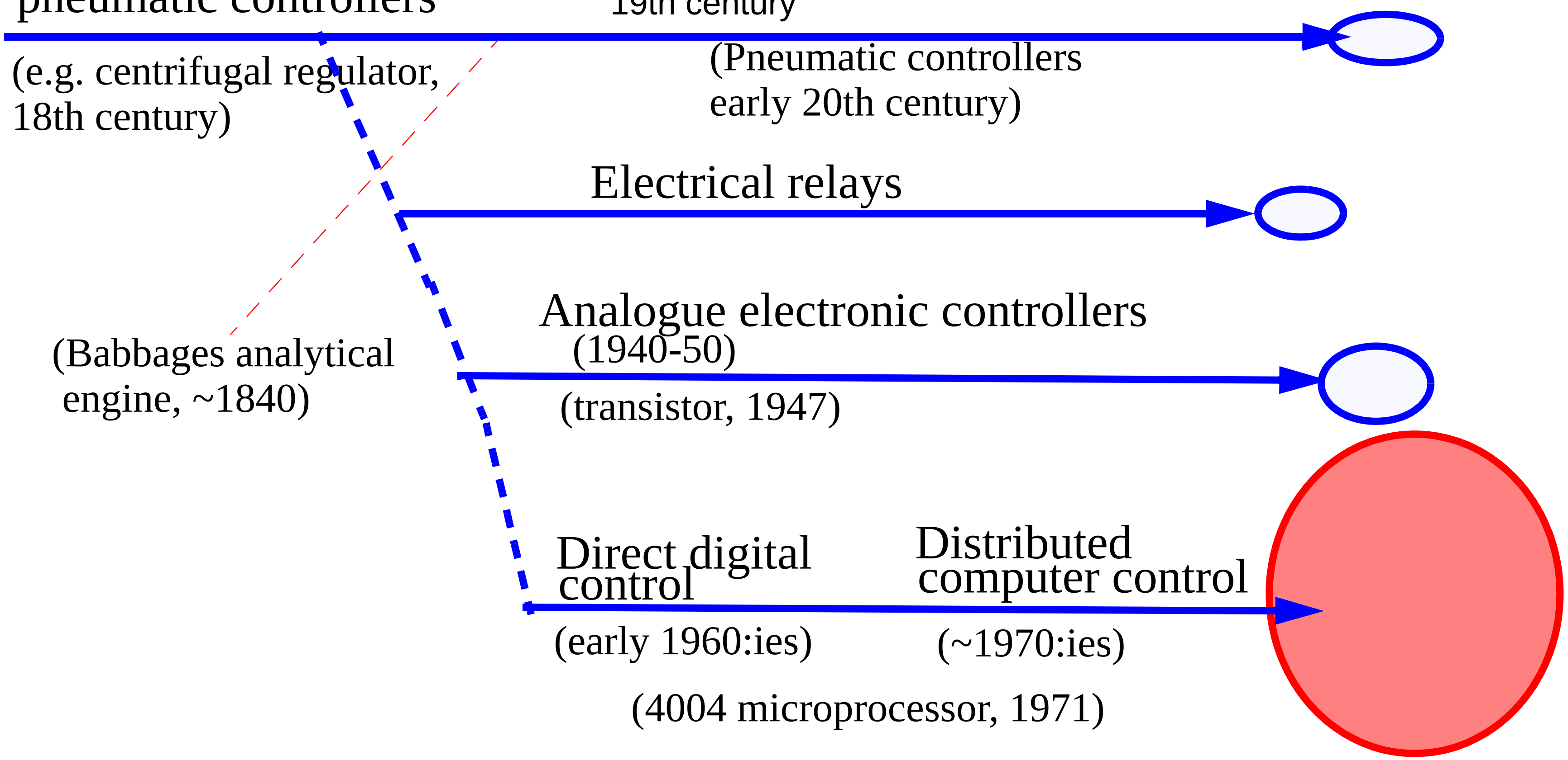
’Moore’s law’ (Intel, 1965):

Microelectronics (MOS) performance is ~doubled every 18 months and chip size is reduced by 50%

Intel 4004/1971 vs. Intel Pentium/1996: from 2300 to 5.5 million transistors

One prediction for "2006: One chip with 200M transistors with 2GHZ frequency"

1.3.2 Control technologies over time



Mechanical, hydraulic and pneumatic controllers

Combustion engine
19th century

(e.g. centrifugal regulator, 18th century)

(Pneumatic controllers early 20th century)

Electrical relays

(Babbages analytical engine, ~1840)

Analogue electronic controllers
(1940-50)

(transistor, 1947)

Direct digital control

(early 1960:ies)

Distributed computer control

(~1970:ies)

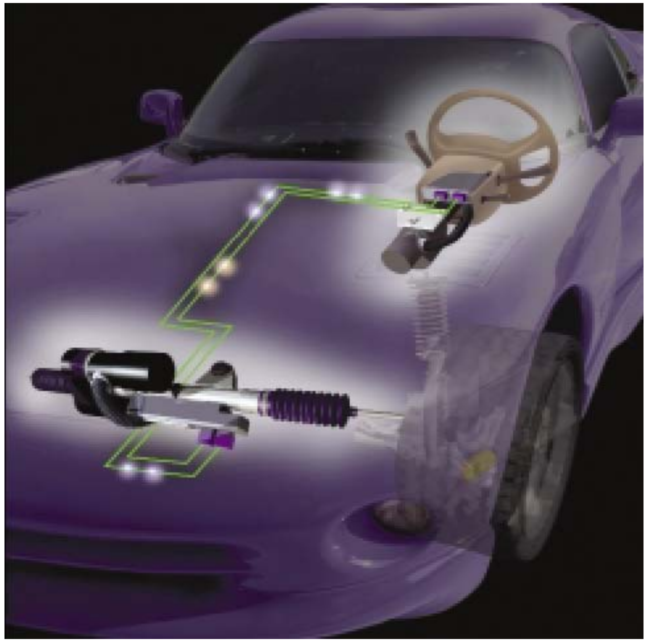
(4004 microprocessor, 1971)

1.3.3 Control development over time

- All mechanical controllers, e.g. the centrifugal governor (wind mills, steam engines, telephones), from 1700 and onwards.
- Early theoretical results, e.g Routh, Hurwitz and Lyapunov on stability, late 1800’s.
- Ziegler’s and Nichols’ method for tuning of PID controllers in 1942.
- Computers in control from the mid 50’s.
- New control design methods in the late 20th century: Optimal, Adaptive, Non-linear, Robust, Fuzzy, Neural, etc.
- 95% of all process controllers are of PID type

1.3.4 The paradigm shift

Servo steering with
mechanical backup



Steer by wire

⇨ **More digital control!**

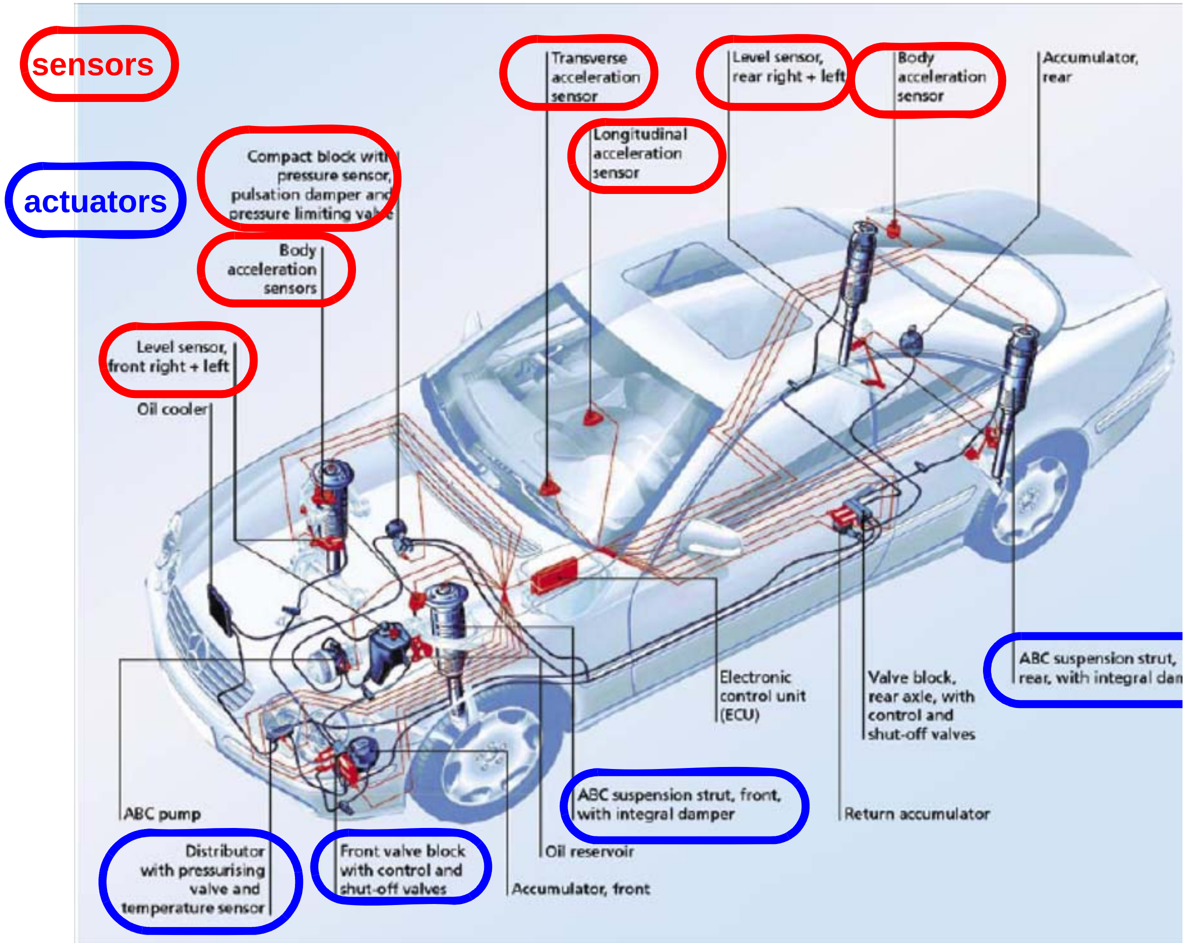
1.3.5 What can be achieved?

Hy-Wire from GM (autonomy 2)

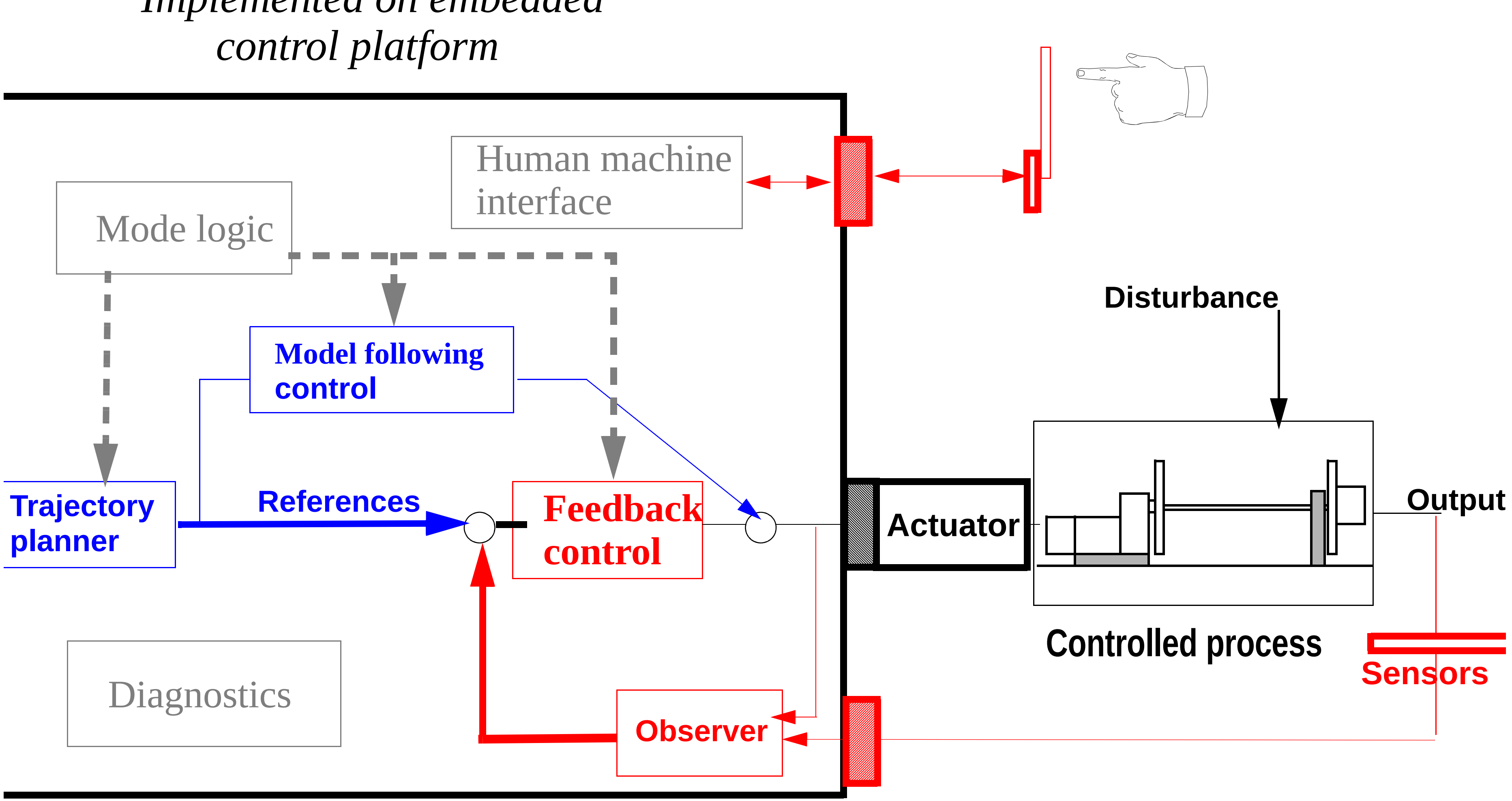
Skateboard concept
Fuel cell 94kW
Integrated distributed control
Weight: 2 tons, Distance: 480kn

1.3.6 Why models? - active body control

- Many sensors
- 4 actuators
- Complex dynamics
- Mechanical and control design not separable
- Simulation necessary



1.3.7 More than just feedback



1.3.8 Example motion control applications

- Industrial robot control
- Vehicular systems (suspension, brakes, engine control...)
- Aircraft control and space applications
- Axis control in various production machines
- Hydraulic servo, e.g. in forest harvester or excavator
- Master and slave control in teleoperated systems
- Mobile robots/machines

1.3.9 The complete *control* design process

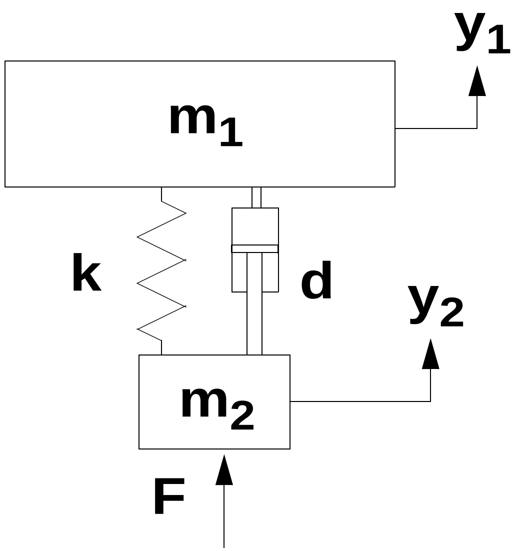
- Design of the mechanical system
  Either prior to, or in conjunction with control design
- **Modelling** of the system and in some cases also modelling of the system’s interaction with the environment
- Analysis of **dynamic properties** including simulation
- Control requirement **specifications**
- **Control design** including choice of related components such as sensors, control computer and sometimes actuator(s)
- Simulation, verification and **control prototyping**
- Real-time **implementation**

1.3.10 Mechanical system modelling

- **Degrees of freedom - DOF: The number of inertia in a dynamic system that can move relative each other**

- **The motion of each DOF is modelled by,** $m\ddot{x} = \sum F_y$**, or** $J\ddot{\varphi} = \sum M_y$**.**

Example:

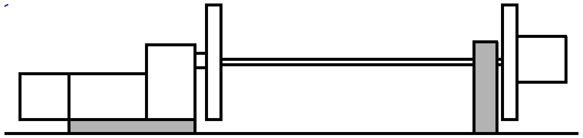


Two DOFs coupled by a weak link.
4:th order differential equation

1.3.11 Modelling and dynamic analysis of mechanical systems

- Putting up the differential equations
- Linearization
- Formulations:
  - State space; matrix equations
  - Input-output; transfer functions
- Different levels of simplification depending on purpose, e.g. simulation vs control design, feed forward vs feedback - *model validation*
- Finding the parameters; system identification
- Computer tools such as Maple and Matlab e.g. with the Identification toolbox

1.3.12 Different domains for different tasks

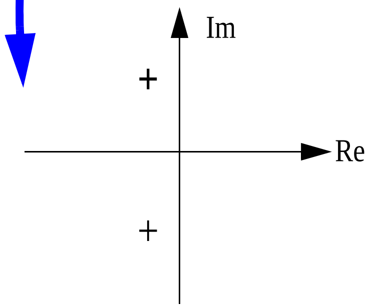


$$\frac{dx}{dt} = f(x, u) \qquad y = h(x)$$

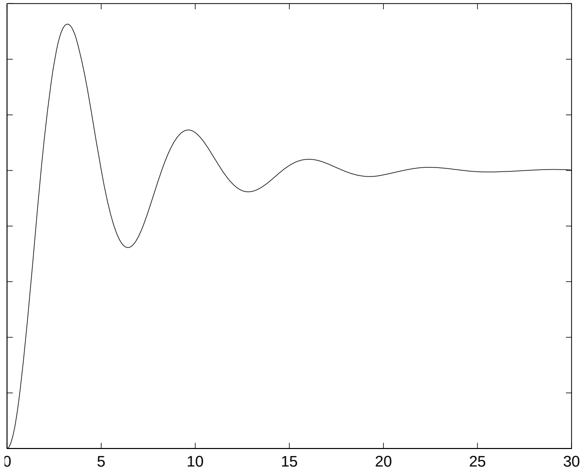
Time domain

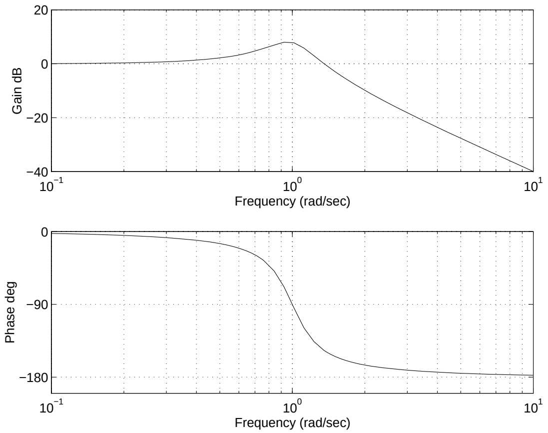
$$y(t) = \sum A_i e^{-\lambda_i t} \cos(\omega_i t + \phi_i)$$

$$G(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2}$$



Complex plane





Frequency domain

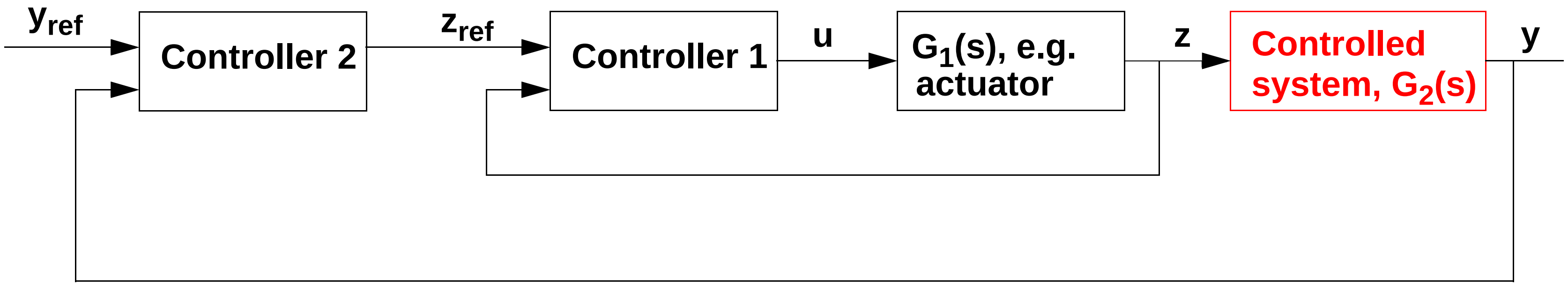
1.3.13 Feedback control

Different controller structures

- PID control:

$$u(t) = K_p \cdot e(t) + K_I \cdot \int_0^t e(s)ds + K_D \cdot \frac{\mathrm{d}}{\mathrm{d}t}e(t)$$

- Cascade control: inner loop viewed as an ideal servo by outer loop

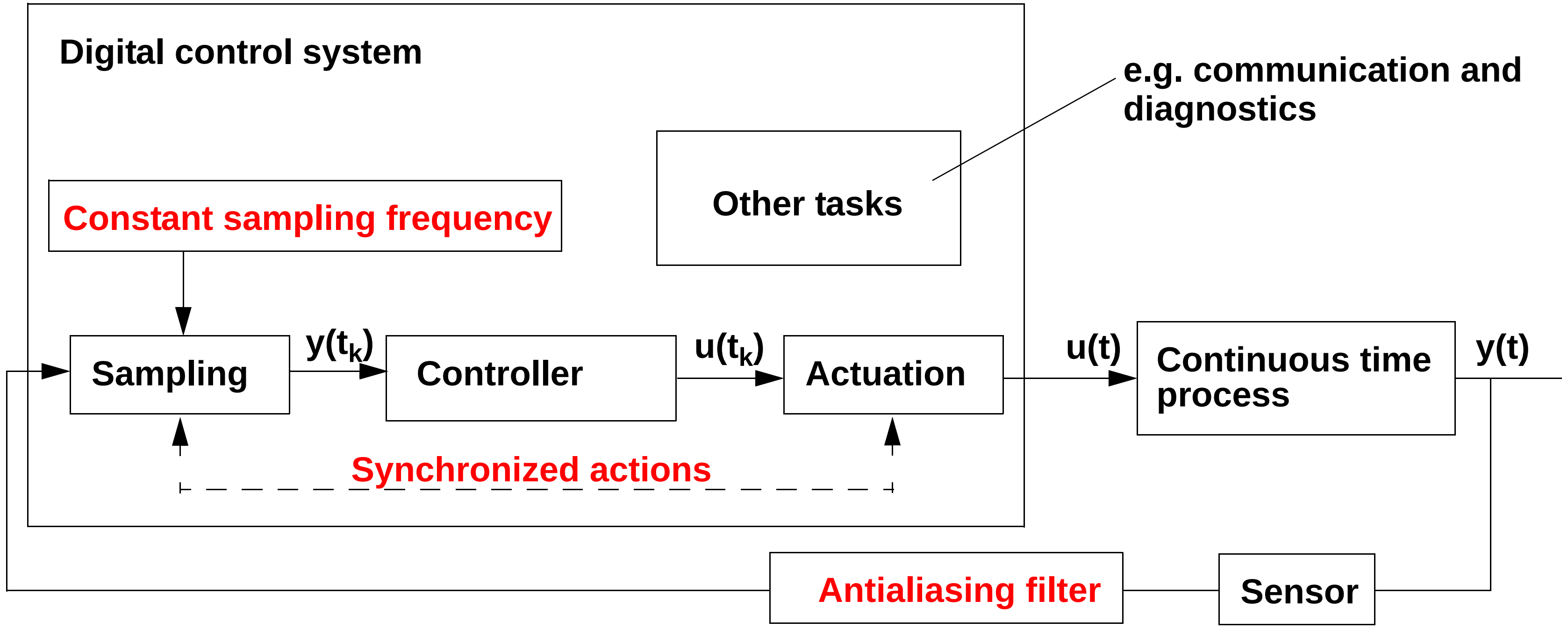


1.3.14 Feedback control properties

The main principle in control engineering

- Typically model based (but not required to be)
- Produces control signals after an error has occurred
- Disturbance rejection is achieved
- Effect of process parameter variations is reduced
- Leads to a closed loop
- Sensor noise may be amplified and deteriorate performance
- May lead to instability if designed incorrectly

1.3.15 Discrete time control



Digital control system

e.g. communication and diagnostics

Other tasks

Constant sampling frequency

Sampling

$y(t_k)$

Controller

$u(t_k)$

Actuation

$u(t)$

Continuous time process

$y(t)$

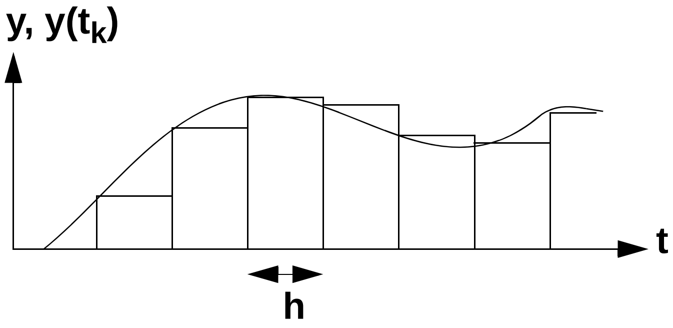
Synchronized actions

Antialiasing filter

Sensor

Sampling interval $h = t_k - t_{k-1}$

Sampling frequency $f = 1/h$



1.3.16 Brief introduction to tools

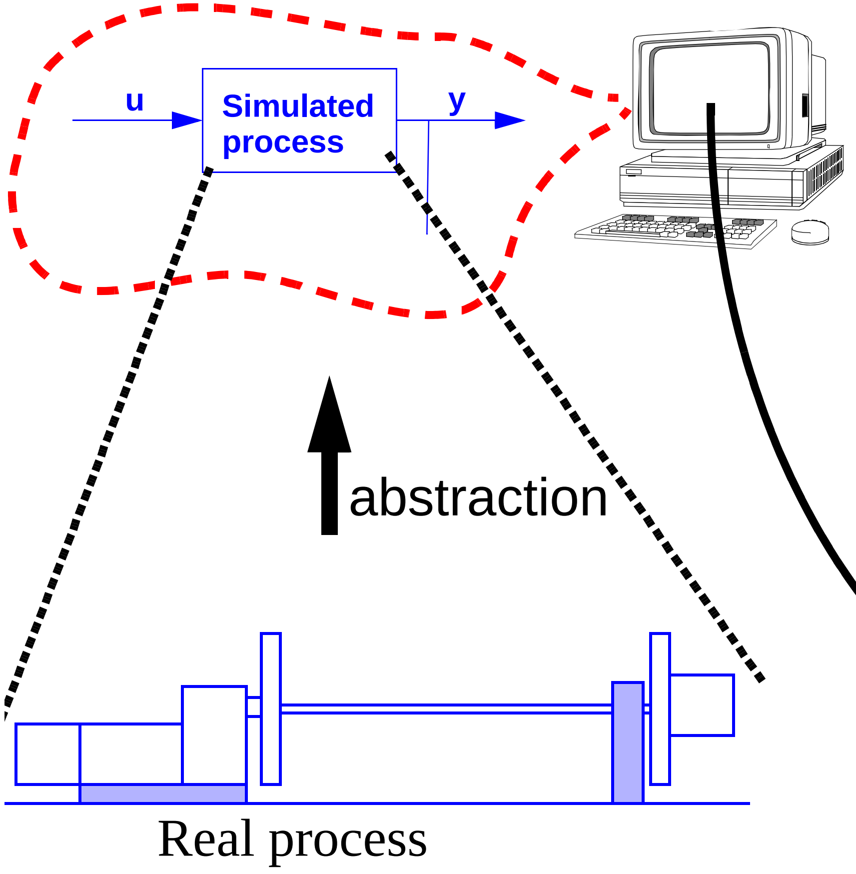
- **Modelling and simulation**
  + Matlab/Simulink
  + Purpose: Modeling, analysis and design + model verification

- **Rapid control prototyping (RCP), 2 different systems**

dSPACE

- Power PC on PCI bus inside PC
- I/O on the same PCI board
- Programmed with Simulink
- RTW, C-code generator
- Microtec C cross compiler
- 4 systems in lab
- Host PC env. ControlDesk

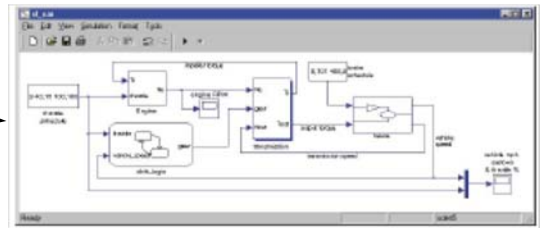
1.3.17 Modelling and simulation



$$\frac{dx}{dt} = f(x, u) \qquad y = h(x)$$

- Numerical solution: non real-time

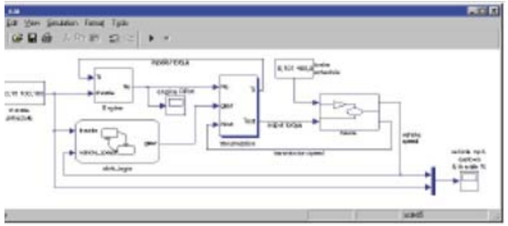
  Integration solvers, e.g., runge kutta order 2-5 using variable step size for efficient simulation time and stable solution.



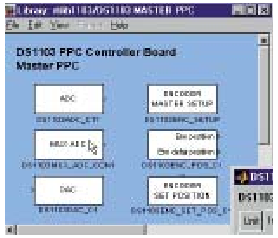
1.3.18 Rapid control prototyping (RCP)

Modelling and design

Non real-time simulation



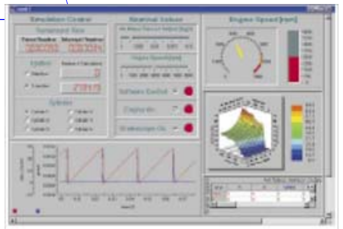
Include I/O drivers



Automatic C-code generation

Compile and down load on RCP HW

Instrumentation for tuning and measurement



Evaluate and redesign

Control implementation needs tools for quick iterations