Experiences of Teaching Real-Time Systems to Control Engineers

Karl-Erik Årzén
Dept of Automatic Control
Lund University

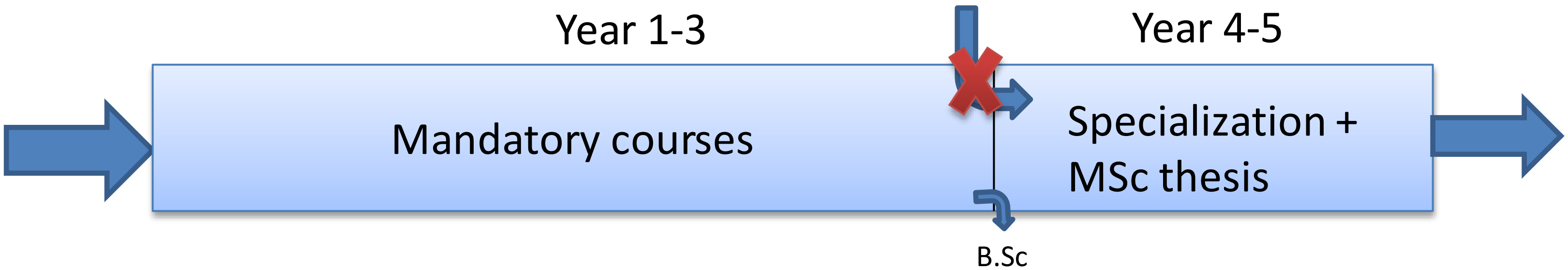
LUND UNIVERSITY

Outline

- The Lund Education Systems
- The Real-Time Systems Course
- A MBSE Perspective on the course

Education System

- LTH (Faculty of Engineering at Lund University) follows the traditional ”pre-Bologna” model
- Five year integrated engineering programs



  - Elec. Eng, Comp. Eng., Mech. Eng, Eng. Phys., Eng. Math., Chem.Eng, …
  - Around 18 programs

- A few International 2-year Master’s Programs
  - E.g., SoCWare, Wireless Systems
  - Aimed at non-European students

Departments vs Programs

- Departments do not have their ”own” students
- Matrix structure

Engineering Programs

Departments

	E	M	F	D	K	…….
Aut.Control	X	X	X	X	X	
Comp.Science						
Electro & IT						
Math						
Physics						
…..						
…..						

Course offerings

Specializations

- 4-12 specializations per program
- Each specialization contains a course package from which a student has to select sufficiently many
- Example of specialisations that contain our courses:
  - ”Control Systems”, ”Systems, Signals, and Control”, ”Control and Automation”, ”Embedded Systems”, ”Mechatronics”, ….
- Specializations correspond in some sense to masters programs

Basic Level Courses

Basic level courses:

- Basic Course in Automatic Control
  - Mandatory for Eng. Phys, Eng. Math, Elec Eng, Comp Eng, Mech Eng, Ind Eng, NanoPhysiscs, and Info and Comm Eng
  - 2nd or 3rd year
  - 600 students per year
- System Engineering
  - Mandatory for Environmental Eng
- Process Control
  - Elective for Chem Eng and BioChem Eng
- Physiological Models and Computations
  - Mandatory for Medical Eng

Advanced Level Courses

- Mostly students from Eng Phys, Eng Math, Elec Eng, Comp Eng, Mech Eng
- Around 40-60 students per course and year
- Elective

- Multivariable Control
- Nonlinear Control
- Predictive Control
- System identification
- Control Theory
- Network Dynamics
- Market-Driven Systems

Advanced Level Courses

- Real-Time Systems
- Project Course in Control

Outline

- The Lund Education Systems
- The Real-Time Systems Course
- A MBSE Perspective on the course

Real-Time Systems

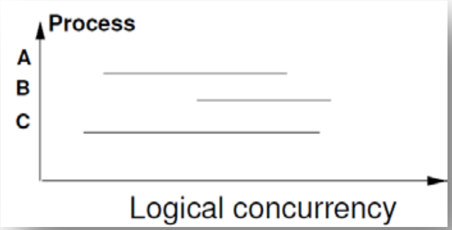
- Largest elective control course
  - 80-90 students per year
  - 10 ECRTS credits over 14 weeks
- Students from Engng Physics, Elec Engng, Comp Engng, Mech Engng, Engng Math specializing in control
  - Different background
  - The same mandatory basic control course
- Oldest course (40 years)
- Format:
  - 17 lectures
  - Exercises (problem-solving and computer)
  - Three 4 hour laboratories
  - 2-3 week project
  - Written open-book exam

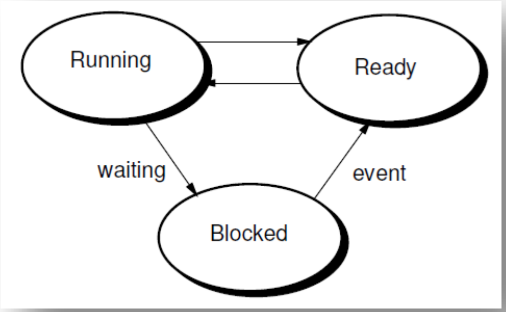
Real-Time Systems: Objectives

1. Implement real-time (control) applications using concurrent programming
2. Understand how an RTOS is implemented
3. Sampled discrete-time control theory
4. Discretization of continuous-time controllers
5. PID control
6. Discrete-Event Control
7. Implementation aspects of controllers
8. Control and scheduling co-design

1. Concurrent Programming

- Threads and processes
- Preemption and context switches
- Mutual exclusion and synchronization
  - Semaphores
  - Monitors with condition variables
  - Message passing
- Deadlocks, priority inversion, priority inheritance
- Interrupts
- Timing primitives
- Basic scheduling theory
- Java, Linux, Stork



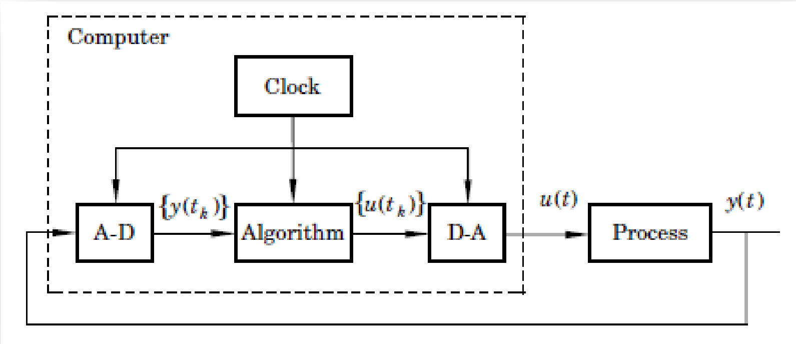


2. Understand how an RTOS is implemented

- STORK
  - Real-Time kernel
- Old, obsolete, Modula 2
- But very pedagogical
  - Every real-time primittive fits on a single slide
- Students should understand, but need not program using STORK

3. Sampled Discrete-Time Control

- ZOH-sampling
- z-transforms, shift operators
- Pulse transfer functions
- State feedback and observers
- Reference signals



4. Discretization of Continuous-Time Controllers

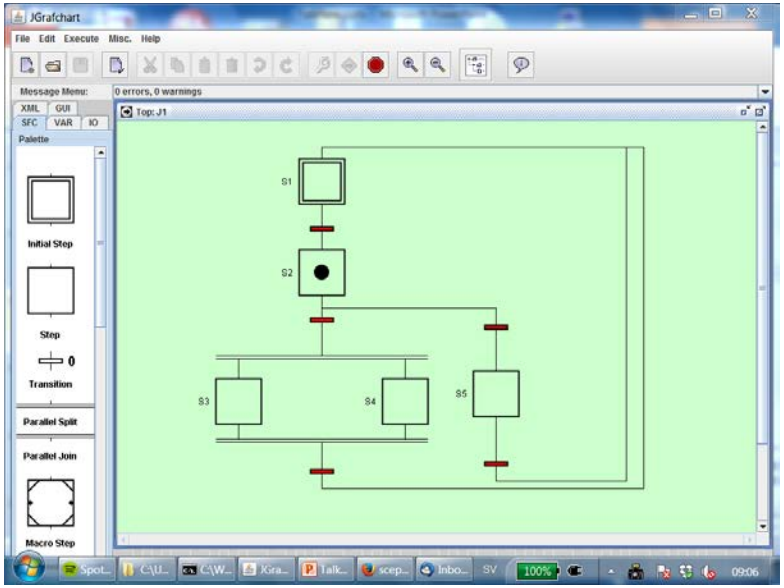
- Difference approximations
- Tustin approximations
- Frequency warping

5: PID Control

- Textbook algorithm
- Algorithm extensions
- Anti-windup
- Bumpless mode changes
- Discretization
- Code

6: Discrete Event Control

- Moore and Mealy machines
- Statecharts
- Grafcet/SFC
  - JGrafchart tool

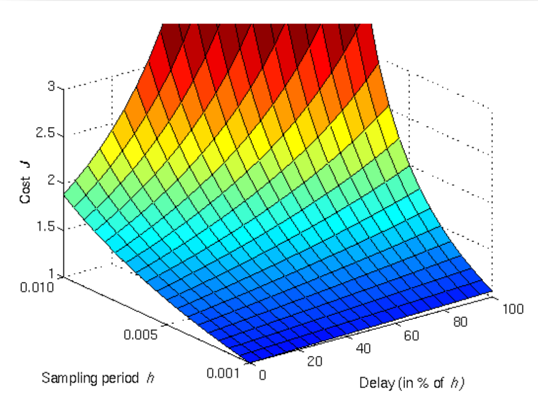


7: Implementation Aspects of Controllers

- Aliasing
- Sampling period selection
- Effects of delay and jitter
- Fixed point arithmetic
- Real-time networks and Networked control

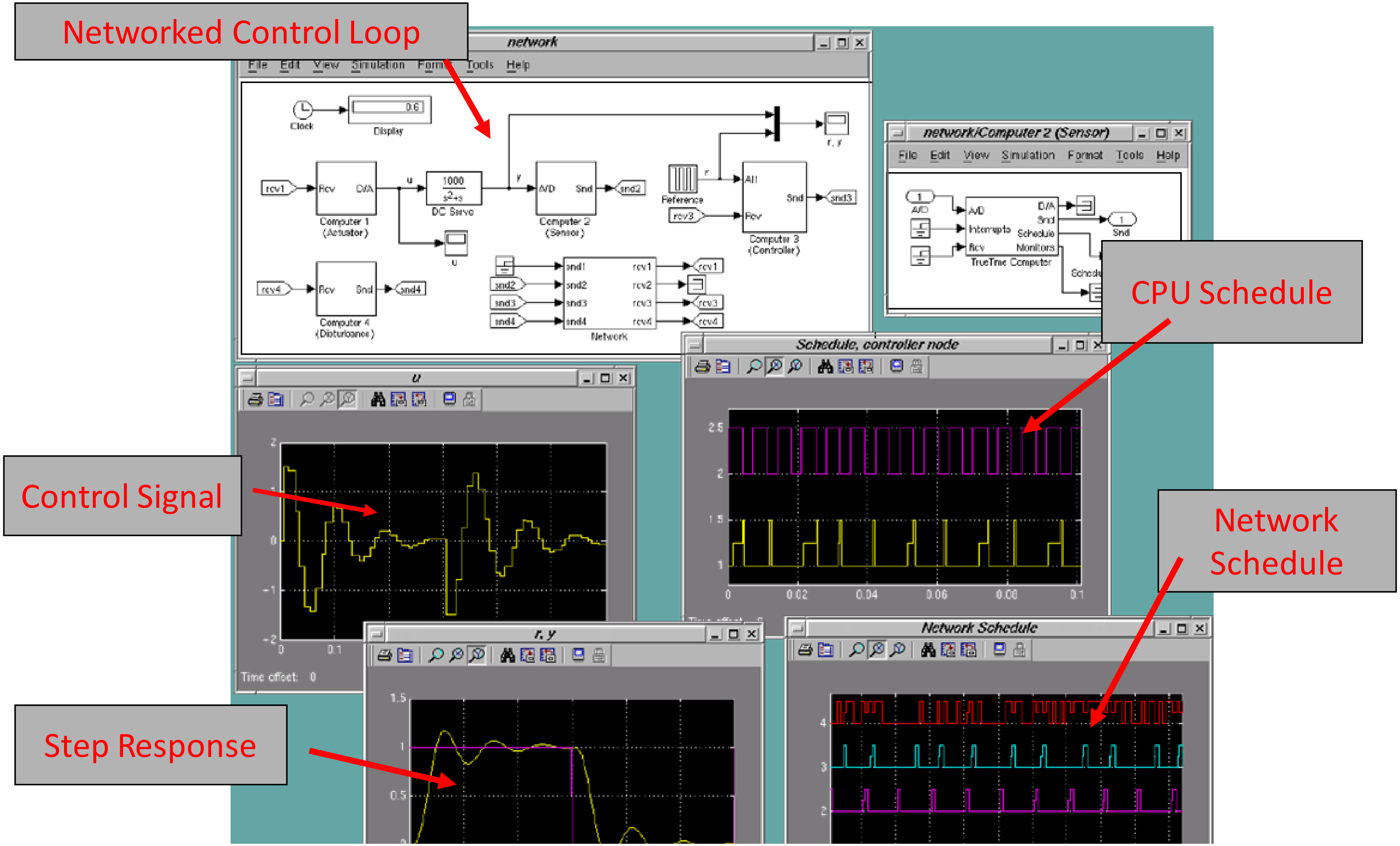
8: Control and scheduling codesign

- How sheduling-induced delays with jitter effects the control performance
- In-house tools
  - Jitterbug
    - Average case analytic evaluation of how stochastic delays effect control performance



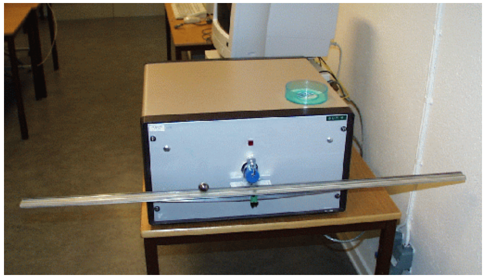
  - TrueTime
    - Simulation of real-time kernels and networks as S-functions within Simulink

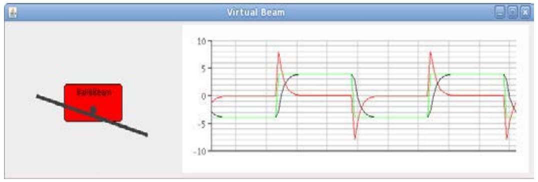
TrueTime

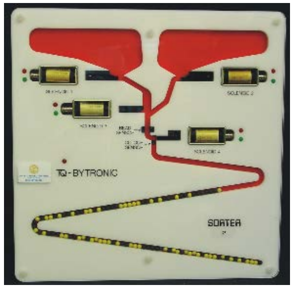

Networked Control Loop
CPU Schedule
Control Signal
Network Schedule
Step Response
network
network/Computer 2 (Sensor)
Schedule, controller node
u
r, y
Network Schedule
Clock
Display
DC Servo
Computer 1 (Actuator)
Computer 2 (Sensor)
Computer 3 (Controller)
Computer 4 (Disturbance)
Reference
Network
TrueTime Computer

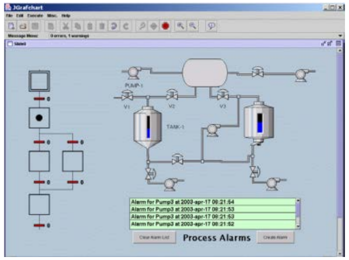
Laboratory Sessions

1. Control of a Ball-and-Beam Process
   - Java on desktop PC
   - From virtual process to physical
2. Discrete-event control of bead sorter process
   - JGrafchart
3. Fixed-point arithmetics control of a servo
   - C on Atmel AVR Mega 8/16









Projects

Two types:

1. Implement some type of controller for a lab process
   – MPC, self-tuner, LQG, PID
2. Extend or evaluate some RTOS

Also Networked control, Smartphone-based control, cloud control, vision feedback, quadcopters, …..

2-3 weeks in groups of 4

Project Outcome

- Mostly very good results

Course Results

- Two group of students
  - Computer nerds
    - Focus on the programming parts of the course
    - Flunk the exam
  - Control theory nerds
    - Focus on the control theory parts of the course
    - Flunk the exam
- In order to pass the exam with good grades both parts are needed
  - Strong correlation between good grade and PhD student enrollment
- The course has good reputation among regional industry
- The course is popular among the students but still considered to be difficult
  - Work with physical processes rather than simulations

Project Course in Automatic Control

- 7 week project course (7.5 ECTS credits)
- Projects in teams of 4
- Go all the way from Modeling, Identification, Analysis, Control Design, Implementation, to Testing
- Often projects of the same type as in RTS but more elaborate

Outline

- The Lund Education Systems
- The Real-Time Systems Course
- A MBSE Perspective on the course

System or Software Engineering?

- In the majority of the projects the plant is given beforehand
- In the rest the design of the process to be controlled is also a part of the project
  - Mostly Lego Mindstorm projects
  - However > 90% are different versions of the Segway processes
- More Software Engng than Systems Engng

Model-Based?

- Yes and no!
- Models used:
  - Continuous and/or discrete-time linear or nonlinear plant models
  - Real-time task models for schedulability theory
- Software models such as UML may be used by the students (those with prior exposure to it) but no requirement (and hence not used)
- Same things with tools such as Eclipse
  - If used, only as a code editor and debugger
- SysML nobody has heard of

Architectures?

- Not very much
- A simple architecture suitable for rather small control systems derived during Laboratory 1 and the computer exercises
- Most students re-use this also in the projects

Virtual Simulation-Based Development?

- Strongly encouraged
- Most teams start developing and evaluating the controller against a simulation model
- Simulink rather than Modelica
  - Prior knowledge
  - Smaller entry threshold

Requirements Engineering?

- No
- Requirements on textual form
  - In the case they exist
- Sometimes the projects are of an explorative nature
  - Investigate if it is possible

Verification?

- No requirements on formal verification or informal testing
- The basic verification constraint is simply that the control performance is good enough
- Sometimes the project advisor performs code inspection
- Finding errors in computer-based control systems non-trivial
  - Real-time bug (locking, priorities, starvation, …..)
  - Algorithm bug
  - Wrong parameter values

Design-Space Exploration

- Trade-off analysis, variant analysis, …..
- Not formalized
- May happen in the projects that also do the plant design
  - However, the students are typically not aware of that this is what they are doing

Development Processes

- No requirement
- However, students that are familiar with, e.g., agile methods, are encouraged to use this also here

Automatic Code Synthesis?

- No!!
- We want the students to learn how to code things manually
- Once they master this then one could consider code generation techniques
- Compare fixed-point arithmetics

Scope and Size of the Projects

- The scope and size of the projects in the courses are typically not large enough to really motivate MBSE
  - Typically only one or a few use cases
- MBSE needs to be introduced at a larger scale than only in one-two indivdiual courses
- Still the Real-Time Systems and the Project Course contain many of the MBSE elements

Conclusions

- The Real-Time Systems and the Project Course contain some elements of MBSE but cannot be considered as truly MBSE courses
- MBSE requires more than modifications of a few individual courses
- How this should be done is to me an open question

The 5 Year Constraint

- Technogy advances but universities are constrained by the requirement that the engineering education may not exceed five years (3+2 or 5)
- Difficult to include the latest developments in the curriculum
  - In RTS, e.g., formal verification and multi-core scheduling theory
  - Narrow the education programs (not desirable)
  - Remove certain basic elements (makes it brittle)

Some thoughts

- To be a good systems engineer (bookkeeper/conductor/....) you need to be a domain expert on something
  - Modelica, model-checkers, multi-criteria optimization, ……
- Somewhat sceptical about MBSE MSc programs
  - At least not with our BSc students
- Students prefer the real thing over simulations
  - But MBSE requires large scale examples
  - How should we combine this?

One Year Program

- Year 6
- Recruit two student types:
  - Students with a master in systems, signals and/or control
    - Background in dynamics, behaviour models, …
  - Students with a master in software engineering
    - Knowledge of UML, requirements engingeering, testing, …
- Joint one semester project

Academic Rigidity

- Introducing a new course is extremely difficult at Lund University currently
  - Even a small change in the topic can take several years
- Introducing a new specialization or a new MSc program is much more difficult

Questions