Real-Time Systems

Course Introduction

Martina Maggio & Karl-Erik Årzén

21 January 2020

Lund University, Department of Automatic Control

http://control.lth.se/education/engineering-program/frtn01-real-time-systems

Real-Time Systems

A *real-time system* is a computer system that has to respond to externally generated events or inputs within a finite and specified time period.

All **control systems** are real-time systems.

Most real-time systems are *embedded systems*, i.e, the computer is an embedded, integrated part of some equipment or machinery.

Embedded Systems

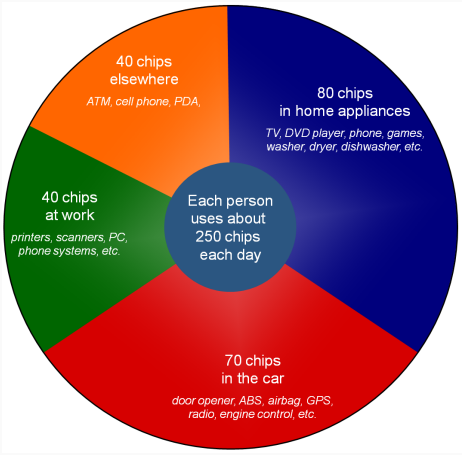
Embedded systems are by far the largest computer sector by volume.

A large part of all embedded systems are control systems with hard/soft real-time constraints.

- Vehicles
- Telecom
- Process & Manufacturing industry
- Intelligent Buildings
- ...

Application Examples

Embedded Systems



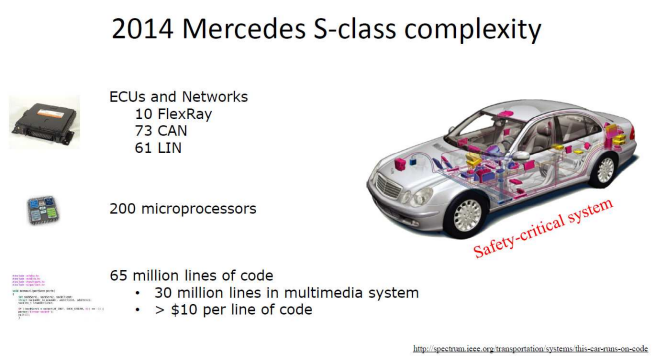
[Illustration courtesy of J. Sifakis]

Example: Car Industry

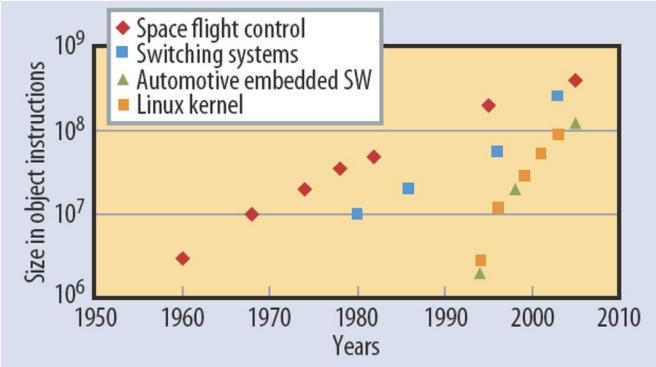
A Volvo 80:

- Contains more than 50 computers (ECUs) and several communication networks,
- Most of them for various control applications,
- 25–30% of the price,
- Software the largest part of the cost,
- Strong connections between control and software,
  - e.g., climate control system: 25,000 lines of C code.

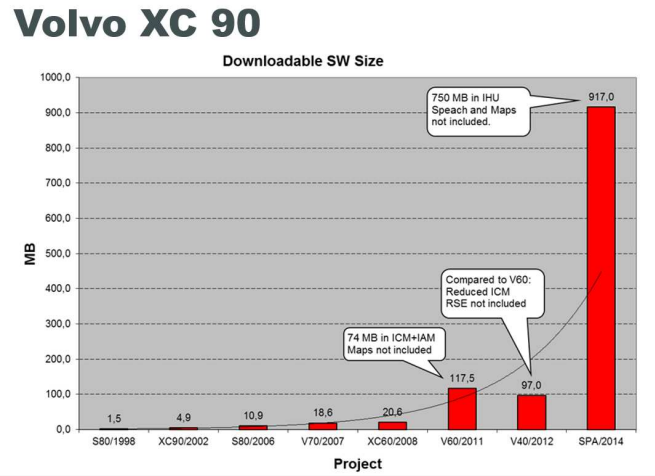
Example: Car Industry



Software Size



Software Size



Lines of Code Comparisons

Pacemaker	80k
Space Shuttle	400k
Windows 3.1 (1992)	2.5M
US military drone	3.5M
Mars Rover	5M
Google Chrome	6M
Firefox	9M
Android	12M
Boeing 787	14M
Linux 3.1	15M
F-35 fighter jet	24M
Microsoft Office 2013	44M
Facebook	61M
High-end car	100M
Mouse genome	120M
Human genome	3,300B

1M LOC = 18,000 pages of printed text

`www.informationisbeautiful.net/visualization/million-lines-of-code/`

Example: Process Automation

- "Industrial IT" buzzword used by ABB.
- Integration of automation and IT,
  - software, distributed systems, WWW, e-commerce.
- Focus on software rather than hardware.

Real-Time Systems in Sweden

Real-Time and Embedded Systems are very strong in Sweden (and Lund).

Research:

- LUCAS: Center for Applied Software Research at LTH,
  - Computer Science and Automatic Control.
- EASE: Industrial Excellence Center for Embedded Applications Software Engineering.
- ELLIIT: The Lund-Linköping Initiative on IT and Mobile Communications.
- WASP: Wallenberg Autonomous Systems and Software Program.

Industry:

- Ericsson, ABB, Volvo, Scania, SAAB, ...

Course Aims

Study methods for design and implementation of computer control systems.

Focused on embedded control systems.

Two parts:

1. Real-time programming,
2. Design and Implementation of Digital Control Systems.

Programming Languages

Java as the main programming language.

However, not a Java course.

We assume basic knowledge of:

- Java,
- Object-oriented programming concepts.

Code examples written **Modula-2** (similar to C and Pascal).

One laboratory session and some of the projects will use **C**.

Relation to EDA040, Concurrent Programming

Students who have taken "Concurrent/Real-Time Programming" at Computer Science will recognize some parts of the first lectures.

During the lectures we will also describe how real-time programming is performed with a conventional real-time programming language (Modula-2) and how a conventional real-time kernel (Stork) is implemented (you do not have to program in Modula-2).

Deeper understanding and repetition.

Students who have taken the Concurrent Programming course will do a special version of Lab 1 in which LJRT is used. They also must do a control-oriented project.

Staff

Karl-Erik Årzén
Course Responsible and Lecturer

karlerik@control.lth.se
046-222 87 82

Martina Maggio
Course Responsible and Lecturer

martina@control.lth.se
046-222 87 77

Mika Nishimura
Course Administrator

mika@control.lth.se
046-222 87 85

Staff

Claudio Mandrioli
Teaching Assistant

Gautham Nayak Seetanadi
Teaching Assistant

Nils Vreman
Teaching Assistant

E-mail:

- claudio.mandrioli@control.lth.se
- gautham.nayak_seetanadi@control.lth.se
- nils.vreman@control.lth.se

Lectures

Lecture	Date	Time	Room	Topic	Lecturer
L1	Jan 21	10-12	M:B	Introduction	Both
LX	Jan 22	15-17	M:2112B	Extra: Introduction to Java	Martina
L2	Jan 23	10-12	M:B	Concurrent programming	Martina
L3	Jan 24	8-10	M:B	Process communication 1	Martina
L4	Jan 28	10-12	M:B	Process communication 2	Martina
L5	Jan 31	8-10	M:B	Interrupts and time	Martina
L6	Feb 4	10-12	M:B	Sampling of linear systems	Martina
L7	Feb 6	10-12	M:E	Input-output models	Martina
L8	Feb 11	10-12	M:B	Approx. of analog controllers, PID	Martina
L9	Feb 13	10-12	M:E	State feedback and observers	Martina
L10	Feb 18	10-12	M:B	Feedforward design	Martina
L11	Feb 20	10-12	M:E	Implementation aspects	Martina
L12	Feb 25	10-12	M:B	Scheduling theory	Martina
L13	Feb 27	10-12	M:E	Project specifications	Both
L14	Mar 5	10-12	M:E	Discrete-event control	Martina
L15	Mar 10	10-12	M:E	Real-time networks	Martina
L16	Mar 26	15-17	M:E	Hot research topics	Martina
LY	Apr 2	15-17	M:E	Extra: Repetition lecture	Martina
L17	May 25	15-17	M:E	Project demos & oral presentations	Martina

Computer Exercises

Nils Vreman: exercise sessions on Tuesdays 15-17.

Gautham Nayak Seetanadi: exercise sessions on Wednesdays 10-12.

Claudio Mandrioli: exercise sessions on Wednesdays 13-15.

- Five computer exercises (C1–C5)
- One extra Java exercise (C0 – 2020-01-27)

The last two problem solving exercise only have two sessions.
In order to balance the load on the exercise groups you must register for the group that you would like to follow.

All exercises are held in Department of Automatic Control, Lab A.

Problem Solving Exercises

Gautham Nayak Seetanadi: exercise sessions on Wednesdays 08-10.

Claudio Mandrioli: exercise sessions on Thursdays 08-10.

Nils Vreman: exercise sessions on Fridays 10-12.

- Six problem-solving exercises (P1–P6)
- One extra Matlab exercise (P0)

All exercises are held in Department of Automatic Control, Lab A.

Exercises

Exercise	Topic
C0	Extra: Introduction to Java
C1	Threads
C2	Synchronization
C3	Controller implementation
C4	Graphical user interface
C5	Prepare Lab 1
P0	Extra: Control in Matlab
P1	Sampling of systems
P2	Input-output models
P3	State feedback and observers
P4	Discrete approximation, PID
P5	Fixed-point implementation
P6	Scheduling theory

Laboratory Sessions

- Three manadatory laboratory sessions, 4 hours each.
- Preparatory assigments checked at the beginning of each lab.
- Room: Department of Automatic Control Lab A.

Lab	Dates	Topic	Responsible
Lab 1	March (ini)	Ball and beam	Nils Vreman
Lab 2	March (mid)	Bead sorter	Gautham Nayak Seetanadi
Lab 3	March (end)	Rotating servo	Claudio Mandrioli

Lab 1

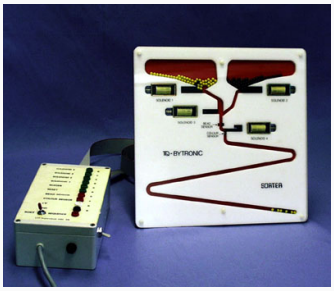
Implementation of a control system for the ball & beam process:

- Cascaded PID controllers,
- Java or Java/LJRT with Swing-based GUI,
- Prepared during the computer exercises.

Lab 2

Sequence control of a bead-sorter process:

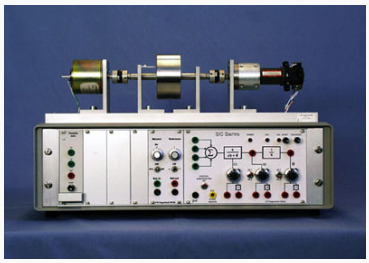
- Discrete-event controller,
- JGrafchart – a Java-based Grafcet editor and run-time system.



Lab 3

Fixed-point implementation of a DC-servo controller:

- State feedback controllers,
- C on ATMEL AVR Mega16.



Project

Projects are performed as team works with four persons per team (in special cases it is OK with smaller project teams).

Around 30 different projects to chose among:

- Control of ball and beam process,
- Control of inverted pendulum,
- Control of helicopter process,
- Real-time kernel projects,
- Embedded system projects using ATMEL AVR and C,
- Lego Mindstorm NXT projects.

If you are following the Predictive Control course it will be possible to do a joint project between the courses. Students that have taken EDA040 Concurrent Programming course must do a control-oriented project.

Project

Important dates:

- Feb 27: Presentation of available projects,
- Mar 5: Deadline for team formation and project selection,
- Mar 27: Deadline for suggested solution,
- May 18: Deadline for project report (10–15 pages, English/Swedish),
- May 25, at Lecture 18: Project demos (Mandatory),
- May 25: Oral presentations (Mandatory).

Literature

- K.-E. Årzén, "Real-Time Control Systems", 2015. KFS.
- B. Wittenmark, K.J. Åström, K-E Årzén, "Computer-Control: An Overview", Educational version 2016. KFS.
- "Real-Time Systems – Problem Solving Exercises", 2015. KFS.
- "Real-Time Systems Formula Sheet". Online.

The 2014 versions are very similar and also possible to use.

Examination

Mandatory parts:

- Three laboratory sessions, Project, Written exam (5 hours).

The exam consists of 25 points and gives the grade Fail, 3, 4, or 5:

- 2020-04-24, 8-13: Victoriastadion 1A, 1B,
- 2020-05-30, 8-13: Sparta:B,
- 2019-08-27, 8-13: MA 10F.

Accepted aid: The textbooks "Real-Time Control Systems" and "Computer Control: An Overview ", standard mathematical tables and authorized "Real-Time Systems Formula Sheet"; pocket calculator.

Registration

You must **register** for the course in Ladok. This is something you do yourself. In order to be able to this you must have signed up for the course ("anmält dig till kursen").

If you have not done that you should: 1) *send an email* to Martina or Karl-Erik asking if it is OK that you follow the course, 2) forward the positive reply email to your *program planner* and ask her/him to do a late sign up ("sen anmälan") for the course. Once that is done you can finally **register** for the course in Ladok.

Course History

-71-72 Control of LKAB iron ore crusher over modem, PDP 15

-73 "Computers in Control Systems", PDP 15, assembler

-79 "Computers in Control Systems 2", LSI-11, Concurrent Pascal

-81 Pascal + real-time kernel

-83 "Applied Real-Time Programming", IBM PC, Modula 2

-86-87 CS course on real-time programming. Focus on robotics.

-89 "Computer Implementation of Control Systems", VME 68020

-93 "Real-Time Systems". CS course no longer a prerequisite.

-96 Windows NT, Pentium, InTouch

-98 PowerPC, Migration to Java started

-00 Java, Linux, PC

-03 ATMEL AVR microprocessors introduced

-07 More focus on digital control and embedded systems