Real-Time Systems

Course Introduction

Anton Cervin & Martina Maggio
18 January 2016
Lund University, Department of Automatic Control

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

Embedded Systems

[Illustration courtesy of J. Sifakis]

Example - Car Industry

- A Volvo S80 contains $>$ 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 - 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 have a very strong position in Sweden and in 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

Industry:

- embedded systems and embedded control systems of vital importance to Swedish 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 (very similar to C, Pascal) will be shown.

One laboratory session and some of the projects will use C

Relation to EDA040 Concurrent Programming

The students who have taken the Concurrent/Real-Time Programming course 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

Students that have taken the Concurrent Programming course must do a control-oriented project.

Staff

Anton Cervin
Course responsible and lecturer

anton@control.lth.se
046-222 44 75

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

Manfred Dellkrantz
Teaching assistant

manfred@control.lth.se

Martin Karlsson
Teaching assistant

martinka@control.lth.se

Victor Millnert
Teaching assistant

victor@control.lth.se

Gabriel Ingesson
Teaching assistant (projects only)

gabriel@control.lth.se

Olof Troeng
Teaching assistant (labs only)

oloft@control.lth.se

Lectures

Lecture	Date	Time	Room	Topic	Lecturer
L1	Jan 18	10-12	E:A	Introduction	Both
L2	Jan 19	13-15	E:A	Concurrent programming	Martina
LX	Jan 20	17-19	M:2112b	Extra: Introduction to Java	Martina
L3	Jan 21	13-15	M:B	Process communication 1	Martina
L4	Jan 26	13-15	E:B	Process communication 2	Martina
L5	Jan 28	13-15	M:B	Interrupts and time	Martina
L6	Feb 2	13-15	E:B	Sampling of linear systems	Anton
L7	Feb 3	13-15	MH:B	Input-output models	Anton
L8	Feb 9	13-15	E:B	Approx. of analog controllers, PID	Anton
L9	Feb 11	13-15	M:B	State feedback and observers	Anton
L10	Feb 16	13-15	E:B	Feedforward design	Anton
L11	Feb 18	13-15	M:B	Implementation aspects	Anton
L12	Feb 23	13-15	E:B	Scheduling theory	Martina
L13	Feb 25	13-15	M:B	Real-time networks	Anton
L14	Mar 1	13-15	M:B	Discrete-event control	Martina
L15	Mar 3	13-15	MH:B	Project specifications	Both
L16	Mar 8	13-15	M:B	Hot research topics	Both
LY	May 3	13-15	MH:B	Extra: Repetition lecture	Both
L17	May 19	13-15	M:B	Project demos & oral presentations	-

Exercises

- Five computer exercises (C1–C5) + one extra Java exercise (C0)
- Six problem-solving exercises (P1–P6) + one extra Matlab exercise (P0)
- Room: Department of Automatic Control Lab A

Exercise	Dates	Topic
C0	Jan 21-22	Extra: Introduction to Java
C1	Jan 25-26	Threads
P0	Jan 28-29	Extra: Control in Matlab
C2	Feb 1-2	Synchronization
P1	Feb 4-5	Sampling of systems
C3	Feb 8-9	Controller implementation
P2	Feb 11-12	Input-output models
C4	Feb 15-16	Graphical user interface
P3	Feb 18-19	State feedback and observers
C5	Feb 22-23	Prepare Lab 1
P4	Feb 25-26	Discrete approximation, PID
P5	Mar 3-4	Fixed-point implementation
P6	Mar 8-9	Scheduling theory

Laboratory Sessions

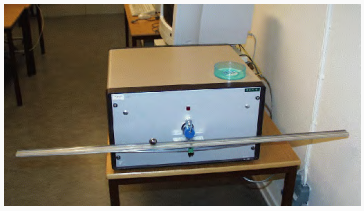
- Three manadatory laboratory sessions, 4 hours each
- The preparatory assigments will be checked at the beginning of each lab
- Room: Department of Automatic Control Lab A

Lab	Approx. dates	Topic	Sign-up	Responsible
Lab 1	Feb 24–Mar 2	Control of ball and beam	Feb 10	Victor Millnert
Lab 2	Mar 3–23	Sequence control of bead sorter	Feb 22	Martin Karlsson
Lab 3	Mar 29–Apr 5	Embedded control of rotating servo	Mar 15	Manfred Dellkrantz

Lab 1

Implementation of a control system for the ball & beam process

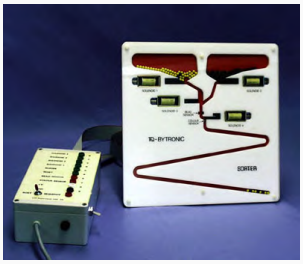
- Cascaded PID controllers
- Java or Java/LJRT with Swing-based GUI
- Prepared during the computer exercises



Lab 2

Sequence control of a bead-sorter process

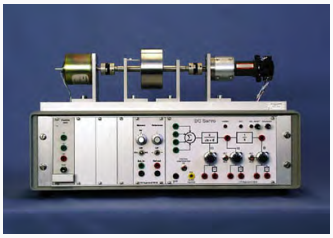
- Discrete-event controller
- JGrafchart – a Java-based Grafcet editor and run-time system



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
- etc.

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:

- March 3, at 13–15 (Lecture 15): Presentation of available projects
- March 9: Deadline for team formation and project selection
- April 6: Deadline for suggested solution
- May 16: Deadline for project report (10–15 pages, English/Swedish)
- May 19, at 13-15 (Lecture 18): Project demos (mandatory)
- May 19, at 15-17: Oral presentations (mandatory)

Very good projects will give one bonus point and excellent projects will be give two bonus points on the May and June 2016 exams.

Bonus points can help raise the grade from 3 to 4 or from 4 to 5, but not from Fail to 3.

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.

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.

Two exam opportunities in the spring:

- May 12 at 14-19
- June 1 at 14-19

Changes From Last Year

Small updates of the textbooks

New lecturers, new Hot Research Topics lecture

Always some new and exciting projects to choose from

Most students were happy about the course last year (+60 overall CEQ in 2015, +38 in 2014, +64 in 2013)

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