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

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

Lines of Code Comparisons

Example: Process Automation

Real-Time Systems in Sweden
Course Aims

Study methods for design and implementation of computer control systems.
Focused on embedded control systems.
Two parts:
1. Real-time programming,

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

Claudio Mandrioli
Teaching Assistant
claudio.mandrioli@control.lth.se

Gautham Nayak Seetanadi
Teaching Assistant
gautham.nayak_seetanadi@control.lth.se

Nils Vreman
Teaching Assistant
nils.vreman@control.lth.se

Lectures

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Topic</th>
<th>Lecturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Jan 21</td>
<td>10-12</td>
<td>M:B</td>
<td>Introduction</td>
<td>Both</td>
</tr>
<tr>
<td>L2</td>
<td>Jan 22</td>
<td>15-17</td>
<td>M:012B</td>
<td>Extra: Introduction to Java</td>
<td>Martina</td>
</tr>
<tr>
<td>L3</td>
<td>Jan 23</td>
<td>10-12</td>
<td>M:B</td>
<td>Concurrent programming</td>
<td>Martina</td>
</tr>
<tr>
<td>L4</td>
<td>Jan 24</td>
<td>8-10</td>
<td>M:B</td>
<td>Process communication 1</td>
<td>Martina</td>
</tr>
<tr>
<td>L5</td>
<td>Jan 28</td>
<td>10-12</td>
<td>M:B</td>
<td>Process communication 2</td>
<td>Martina</td>
</tr>
<tr>
<td>L6</td>
<td>Jan 31</td>
<td>8-10</td>
<td>M:B</td>
<td>Interrupts and time</td>
<td>Martina</td>
</tr>
<tr>
<td>L7</td>
<td>Feb 4</td>
<td>10-12</td>
<td>M:B</td>
<td>Sampling of linear systems</td>
<td>Martina</td>
</tr>
<tr>
<td>L8</td>
<td>Feb 6</td>
<td>10-12</td>
<td>M:E</td>
<td>Input-output models</td>
<td>Martina</td>
</tr>
<tr>
<td>L9</td>
<td>Feb 11</td>
<td>10-12</td>
<td>M:B</td>
<td>Approx. of analog controllers, PID</td>
<td>Martina</td>
</tr>
<tr>
<td>L10</td>
<td>Feb 13</td>
<td>10-12</td>
<td>M:E</td>
<td>State feedback and observers</td>
<td>Martina</td>
</tr>
<tr>
<td>L11</td>
<td>Feb 21</td>
<td>10-12</td>
<td>M:B</td>
<td>Feedforward design</td>
<td>Martina</td>
</tr>
<tr>
<td>L12</td>
<td>Feb 25</td>
<td>10-12</td>
<td>M:E</td>
<td>Implementation aspects</td>
<td>Martina</td>
</tr>
<tr>
<td>L13</td>
<td>Feb 27</td>
<td>10-12</td>
<td>M:E</td>
<td>Scheduling theory</td>
<td>Martina</td>
</tr>
<tr>
<td>L14</td>
<td>Mar 5</td>
<td>10-12</td>
<td>M:E</td>
<td>Project specifications</td>
<td>Both</td>
</tr>
<tr>
<td>L15</td>
<td>Mar 10</td>
<td>10-12</td>
<td>M:E</td>
<td>Real-time networks</td>
<td>Martina</td>
</tr>
<tr>
<td>L16</td>
<td>Mar 26</td>
<td>15-17</td>
<td>M:E</td>
<td>Hot research topics</td>
<td>Martina</td>
</tr>
<tr>
<td>L17</td>
<td>Apr 2</td>
<td>15-17</td>
<td>M:E</td>
<td>Extra: Repetition lecture</td>
<td>Martina</td>
</tr>
<tr>
<td>LY</td>
<td>May 25</td>
<td>15-17</td>
<td>M:E</td>
<td>Project demos &amp; oral presentations</td>
<td>Martina</td>
</tr>
</tbody>
</table>
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

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

Laboratory Sessions

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

<table>
<thead>
<tr>
<th>Lab</th>
<th>Dates</th>
<th>Topic</th>
<th>Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab 1</td>
<td>March (ini)</td>
<td>Ball and beam</td>
<td>Nils Vreman</td>
</tr>
<tr>
<td>Lab 2</td>
<td>March (mid)</td>
<td>Bead sorter</td>
<td>Gautham Nayak Seetanadi</td>
</tr>
<tr>
<td>Lab 3</td>
<td>March (end)</td>
<td>Rotating servo</td>
<td>Claudio Mandrioli</td>
</tr>
</tbody>
</table>

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 Grafet 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.

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

- “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.


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.
- 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