Real-Time Systems: Project Descriptions 2014

---

1. Administrative Information

A number of different projects are presented. The projects should be performed as team works with the size of four persons per team (in very special cases it is OK with smaller project teams). Constraints on hardware, processes and supervisors require synchronization among the projects. It may in certain cases be necessary for the project teams to reserve laboratory processes and/or computers using booking lists. More information about how this will be done in practice will be provided later.

> The project teams should be organized and have handed in a priority list of their desired projects by Tuesday **October 14**. The list should contain at least four projects in **priority order**. This should be done in the form of an email to **karlerik@control.lth.se** with the subject "Real-Time Project 2014". The email should contain the **names** and the **email** addresses of **all** the members of the group. On **Monday 20**, the projects will be assigned. We will contact you by email. Projects will be allocated to groups in a first-come, first-served manner.

This year it is not possible to do a joint project with the Predictive Control course unless you wait with your project intil Spring 2015 when it will again be possible.

If you cannot find a group to join, then send us an email with your personal preferences and we will try find a suitable group for you.

Suggested Solutions

A detailed suggested solution (3–4 pages) for the project should be presented to your supervisor no later than Wednesday **November 5**. You are not allowed to start coding until your suggested solution has been approved. The suggested solution is also a natural basis for the project report.

The suggested solution should contain:

1. Program structure. What threads, communication and synchronization will be used? What are the main classes, and what are their public interfaces? Use some kind of graphical notation, e.g., UML or the graphical notation used in the Buttons exercise.
2. Operator communication. Screen layout, available commands in different modes, plotters, etc. Which parameters can be changed on-line by the user?
3. Control principles. What controller structure will be used, how should the controller be designed, what reference generation will be used, etc.
4. Project plan. Time plan. Describe in which order the different parts will be implemented. Define what can be done in parallel.

Project Requirements

Your project will be accepted if it passes the following requirements:

- A **program** that fulfills the specifications should have been demonstrated to your project supervisor by Friday **December 12**. Each team member should be able to answer questions about the program structure, why a certain solution has been chosen, etc. All **code** must be submitted to your supervisor (.zip file)
- A project **report** (in Swedish or English) should be written (using some word processing system). The suggested solution provides a good structure for the project report. A first (but complete) version of the report should be handed in to your supervisor by Friday **December 12**. You are only allowed to hand in revisions of your report three times (including the first version). Spelling errors in the report are not accepted.

  The report should contain the following parts:

  - A cover page with the name of the course, name of the project, the names and email addresses of the involved students, and the name of the supervisor.
  - An introduction that states the problem that has been solved.
  - A section describing the main program structure, both from a class and and a real-time perspective. If possible illustrate this with some type of figure.
  - A section describing the control design aspects of the project.
  - A section describing the user interface in the project including a short HowTo description on how to start and operate the program.
  - A section containing the results. In case the project is a control-oriented project this should include plots of measurement signals, reference signal, and control signal. If the project is more of a real-time nature then this section could contain measurement results of different type.
  - A conclusion section.
- The project should be **demonstrated** during the project presentation lecture on Tuesday **December 16**, 15:15-17:00.
- An **oral presentation** (10 minutes) should be made on the oral presentation session, Tuesday **December 16**, 17.15–19.00. Prepare a few slides and show them on your own laptop or submit them to your supervisor at least one day before the presentation.

Project Suggestions

A number of different projects are proposed. The programming language used in the projects is Java, unless otherwise indicated. The character codes after the title have the following meaning:

P Real-Time Programming project. May not be selected by those that have taken the CS course on real-time programming.

C Programming in C required.

2 Small project. Only two persons.

S Special project. High risk factor. Only for highly motivated and/or skilled persons.

Each project is described by a brief presentation. A more detailed specification of each project will be available when the projects start.

2. Algorithm-Oriented Projects

Project 1 – Event-Based Sampling and PID Control (2)

Most control theory is based on the assumption that equidistant sampling is used. This is typically not the way manual control is performed. There, a new control signal is only generated when necessary, e.g., when the control error has changed more than a certain limit or when the error derivative has changed more than a certain limit. Implement a PID controller that consists of two parts: one part that samples the input and performs limit checking with high frequency, and another part that, on demand, calculates a new control signal. Try the controller on some process in our lab, e.g., the servo or the double tank.

Project 2 – General State Feedback Controller Using MATLAB Compute Engine

Develop a general state feedback controller with an observer that should work with (almost) any process in our lab. The user should be able to specify the process transfer function, the sampling interval, and the desired state feedback and observer pole placement in the user interface. The control design is then done on-line, using calls to a MATLAB Compute Engine. Try the controller on one or two processes in our lab, for instance the double tank and the DC servo.

Project 3 – Digital vs Analog Cascaded Controllers

In the course you implement a cascaded PID controller for the ball and beam process in laboratory 1. In this project you will explore more advanced design techniques, including direct digital design of cascaded controllers and state feedback with observers. Try the new controller(s) on the ball and beam process and compare the performance to the “old” cascaded PID controller.

Project 4 – Linear vs Nonlinear Reference Generation

In this project you should combine linear state feedback from an observer with linear or nonlinear feedforward from a reference generator (see lectures 8 and 10). A suitable target for this project is the ball and beam process. As a benchmark, you should try to move the ball from one end of the beam to the other as fast as possible without losing the ball. You may also use Optimica and JModelica in order to calculate optimal reference trajectories. Alternatively you may apply the theory to either the linear pendulum or the Furuta pendulum.

Project 5 – Compensation for Network Delays

When closing a control loop over a communication network, a delay is introduced in the feedback loop. Depending on the network protocol the delay is more or less deterministic. The Internet gives very non-deterministic delays. Your task in this project is to investigate the possibility to dynamically compensate for the network delay in the controller. In order to get sufficiently large and stochastic delays you may use the Internet. However, to simplify things we will implement both the sensing, control calculations, and the actuation on the same computer. The delays are obtained by sending out measurement values and actuator values as Ping-messages to some suitable URL on the network. The ping message will return with the same information attached and the round-trip delay for the message can be easily measured.Alternatively, you may implement the delay yourself rather than using the Internet. This gives better control of the the actual delay. In both cases you should try the controller on some process in our lab, for instance the ball and beam or the DC servo.

3. Process-Oriented Projects

Project 6 – "Catch and Throw" Ball and Beam Process

An older version of the ball and beam process is equipped with a ball magazine and a solenoid that automatically pushes a ball onto the beam. The process contains three different ball sizes. The task is to write a program that pushes a ball on to the beam, catches it, weighs the ball, and depending on the size of the ball either throws it into a waste basket or into a special ball basket mounted on the process. Handling of sequences and mode changes are important in the project.

Project 7 – "Catch and Throw" Ball and Beam Process in JGrafchart

The same as project 6 but you should use the Sequential Function Chart editor JGrafchart to implement the sequential parts of the control system, rather than doing it in Java. JGrafchart will communicate with a Java application where the feedback control loops are running using sockets.

Project 8 – Control of the Helicopter Process

Design and implement a digital controller for the helicopter process. To obtain better performance, you can use gain scheduling, where the controller parameters are changed depending on the current operating point.

Project 9 – Control of the Mass-Spring-Damper Process

Design and implement a digital controller for the mass-spring-damper process.

Project 10 – Control of the Linear Pendulum

Design and implement a controller for the linear pendulum process. The controller should able to swing up the pendulum automatically.

Project 11 – Control of the Furuta Pendulum

Design and implement a controller for the Furuta pendulum process. The controller should able to swing up the pendulum automatically.

Project 12 – Multivariable Control of the Batch Tank Process

The batch tank process is a tank with filling, emptying, heating, cooling, and mixing possibilities. In this project you should simultaneously control the level and temperature in the tank while simulating an exotermic chemical reaction.

Project 13 – Vision Feedback

In this project you should use a camera as the sensor in a feedback loop. A suitable process can be the ball and beam process or the Furuta pendulum. The vision sensor can either be a camera or a Kinect sensor (essentially a 3D-camera). In the ball and beam case instead of using the normal beam you should use a non-straight beam. For this project it is an advantage if you have taken the course in Image Analysis.

Project 14 – Model Predictive Control Using CVXGEN – any suitable process

Implement a model predictive controller for a suitable lab process, e.g., the quad-tank process, using CVXGEN to generate fast C code for embedded systems. Investigate the effects of prediction horizon on code size, execution speed and performance. You can also experiment with the use of constraints on the control signal and the output. The project can either be implemented on PCs in the lab, e.g., in Java (JNI to interface with the C code), or on a Raspberry Pi. (Since CVXGEN generates floating point code small microprocessors without hardware-support for floating point and with a small amount of flash memory cannot be used.)

Project 15 – Python Control

The interpreted language Python is gaining in popularity for scientific and engineering applications. In this project you should investigate the suitability of implementing feedback controllers in Python. The task is to implement a controller of the type used in Laboratory 1 entirely in Python utilizing Python's thread support and support for GUIs.

4. Real-Time Programming Projects

Project 15 – Distributed Control with Java RMI (P)

Design and implement a distributed control system where the inter-process communication is done via the Remote Method Invocation and the Java Object Serialization mechanism. Evaluate the benefits and drawbacks of this approach as compared to the exercise project and to project 15, if possible.

Project 16 – Distributed control using MPI (P)

Design and implement a distributed control system where the inter-process communication is done via MPI (Message Passing Interface). MPI is a standardized and portable message-passing system designed to function on a wide variety of parallel computers. The standard defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in different computer programming languages such as Fortran, C, C++ and Java. Open MPI is am open source implementation of MPI that supports Java. Evaluate the benefits and drawbacks of this approach as compared to the exercise project.

Project 17 – Networked Control with Android (P)

Modern technologies allow to create a networked control system with off-the-shelf mobile devices, see `http://goo.gl/xwIhyr`. This project aims at the creation of a simulator over the network of a controlled system. Two mobile devices should connect to each other and communicate over a WiFi. One mobile is simulating a process to be controlled (e.g., the ball and beam), it sends the computed output to the other device and it receives the next input. The other mobile is receiving the output of the simulated process, it is computing a control law (different ones should be implemented) and it is sending the new control input.

Project 18 – Mobile Power Consumption (P)

Managing the power consumption in an efficient way, especially for mobile devices is a crucial problem. The target device for the project is a Sony Xperia T2 Ultra running Android. The purpose of this project is to develop an application (similar to the one proposed in `http://goo.gl/FsKwt2`) that logs in real-time power consumption data. Then, different consumption models (see `http://goo.gl/oFNH1e`) can be tested against the results.

Project 19 – Distributed Wireless Control with Bluetooth (P)

Bluez (`http://www.bluez.org`) is the official Linux Bluetooth protocol stack. The project aims at designing and implementing a distributed control system for one of our lab-processes in Java where the communication is performed via Bluetooth.

Project 20 – Tiny EDF-Based Real-Time Kernel (CP)

TinyRealTime (`http://www.control.lth.se/Staff/AntonCervin/tinyrealtime`) is an event-based kernel for the ATMEL AVR Mega8 processor. The kernel implements the earliest-deadline-first (EDF) task scheduling policy and provides semaphores for task synchronization. Currently, the memory footprint of the kernel is approximately 1200 bytes (of the available 8K bytes). This project aims at improving and extending the implementation of the kernel.

Project 21 – Adding a New Scheduler in a RTOS (CPS)

This project aims at implementing a new control-based scheduler in a RTOS, see `http://dl.acm.org/citation.cfm?id=2560015`. A possible target RTOS for the project could be TinyRealTime (`http://www.control.lth.se/Staff/AntonCervin/tinyrealtime`). The implementation should then be tested against other scheduling techniques. This project requires some abilities and expertise in modifying parts of a (small) kernel.

Project 22 – Control Using Android (P)

Smartphones based on Android provides a nice multi-thread Java environment, sensors, e.g., accelerometers, that allow gesture-based interaction, and a touch screen. In this project the task is to implement a controller in Java on an Android phone for one of our laboratory processes. External sensors can be connected either via the ATMEL/AVR boards and a Bluetooth dongle, or via the Android IOIO board together with a USB-Bluetooth converter.

Project 23 – Real-Time Control in the Cloud Using 4G/LTE (2C)

In this project you should evaluate the performance that can be achieved for a networked control in which the controller executes somewhere in the cloud. The sensor and actuator node communicate with the controller using a wireless 4G/LTE link. The aim of the project is to evaluate the performance that can be achieved when, e.g., the controller is hosted either on an external cloud provider, e.g., Amazon, or within the Internet service provider (the operator). Another focus is to compare the performance obtained for GSM, 3G and 4G. The project will be performed on a Raspberry Pi with an LTE-dongle (one for the sensing side and one for the actuation side).

Project 24 – FreeRTOS evaluation (C)

FreeRTOS is a small C-based real-time kernel that supports a wide range of microprocessors. Your task is evaluate how well suited FreeRTOS is for use in this course, e.g., as an alternative to Java and STORK. You should deploy FreeRTOS on a Texas board with an ARM Cortex M4 processor. On this board you should implement a small multi-threaded control application.

Project 25 – Linux for Control (2CSP)

In the course we have implemented controllers in standard Java in Linux. Java is far from ideal with respect to real-time performance. In this project you will instead implement a laboratory process control system directly in Linux using C. There are several options. One is to use the POSIX pthreads library for the threads and use some graphics framework such as gtk for the GUI. Alternatively one can use Python (pygtk) for the GUI or simpluy implement the GUI in Java. Another possibility, with even better real-time performance, is to implement the controllers in Xenomai, a real-time extension to Linux. This project requires prior experience of Linux programming.

5. Embedded Projects

In the embedded projects you will implement your controllers on a small ATMEL AVR Mega8 or Mega16 processor using the Linux PC for development and cross-compilation. On the Linux PC you will also run the GUI for the controller implemented in Java. The on-line communication between the host and the target machine should be be based on RS-232 (serial line).

Project 26 – Embedded Control of the Ball and Beam Process (C)

Design and implement a digital control system for the ball and beam process using the ATMEL AVR processor. The controller should use state feedback from an observer and include integral action. Use fixed-point arithmetic for the control calculations.

Project 27 – Embedded Control of the Mass-Spring-Damper process (C)

Design and implement a control system for the mass-spring-damper process using the ATMEL AVR processor. The controller should use state feedback from an observer and include integral action. Use fixed-point arithmetic for the control calculations.

Project 28 – Lego Mindstorm Projects

The task in this project is to design, build, and control a process using Lego Mindstorm. Various sensors are available, e.g., gyro, touch sensor, light sensor, ultrasonic sensor, and accelerometer. Different programming languages are available for the Mindstorm. It is, for example, possible to use Java or C.

Project 28a - Lego Segway Design and control a Lego mini-Segway. This cna be combined with using an Android phone for setpoint generation.

Project 28b - Debian Linux for EV3 (C) A Debian Linux installation is available for the latest Mindsorm version called EV3. The task in this project to deploy this, called ev3dev, get RTLinux or something similar running and make a small demo utilizing hard real-time scheduling.

Project 28c - Yor own suggestion Here you may yourself suggest a process that you should design, build, and control. A requirement here, though, is that the feedback control element of the design is sufficiently advanced.

Project 29 – Sensor Fusion for Quadcopter (CS)

The Crazyflie is a tiny open-source quadcopter, see
http://www.linuxuser.co.uk/reviews/crazyflie-6-dof-review-fly-away-now-fly-away
On the Crazyflie there is an Arm Cortex processor running FreeRTOS and the motherboard is a Raspberry Pi. The task in this project is to investigate how to apply sensor fusion utilizing both the pressure sensor and the accelerometers to get a good and robust estimate of the altitude of the quadcopter.

Project 30 – Synchronization Protocol in Wireless Sensor Networks (PS)

FLOPSYNC-2 is a control-based low-power synchronization protocol for wireless sensor networks, see `http://goo.gl/2bH3bT`. The purpose of this project is to implement on a Sony Xperia T2 Ultra with Android the synchronization scheme, and test it in real operating conditions, in order to get some robustness measurements in presence of delays and interference of WiFi networks.

6. Your Own Ideas

If you feel that you have an idea for a suitable project, it can always be discussed.