Real-Time Systems: Project Descriptions 2017

---

1. Administrative Information

The projects should be performed as team works with the size of four persons per team. (In special cases it may be OK with smaller project teams.) Constraints on hardware, processes and supervisors may 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 **February 28**. The list should contain 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 2017". The email should contain the **names** and the **email** addresses of **all** the members of the group. The projects will be assigned on March 2-3 . We will contact you by email once this is done to plan the first meeting with your supervisor(s).

If you cannot find a group to join, then send us an email with your personal preferences and we will 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 **March 24**. You are not allowed to start working in the lab 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. Introduction. What is the overall problem to be solved? What hardware (computers and processes), what operating system, and what programming language will be used in the implementation?
2. 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
3. Operator communication. Screen layout, available commands in different modes, plotters, etc. Which parameters can be changed on-line by the user?
4. Control principles (if applicable). What controller structure will be used, how will the controller be designed, what reference generation will be used, etc.
5. 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 **May 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. The report should be handed in to your supervisor by Thursday **May 12**. The supervisor will read the report and request revisions, if necessary. You are only allowed to hand in 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 **May 16**, 15:15-17:00.
- An **oral presentation** (10 minutes) should be made on the oral presentation session, **May 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 below. The programming language used in the projects is Java, unless otherwise indicated. The character codes after the title have the following meaning:

P Pure real-time programming project. May not be selected by those who have taken the CS course EDA040 Concurrent and Real-Time Programming.

C Programming in C required.

2 Small project. Only two persons.

S Special project. High risk factor. Only for highly motivated persons.

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

2. Predictive Control (PC) Joint Projects

The following projects can be performed as joint projects with the Predictive Control course. It is not necessary that all students in a group performs the project as a joint project.

Project PC4 – Mass-Spring-Damper System

A mass-spring-damper system arranged for linear acceleration is available in our laboratory. Apply adaptive control for improved damping of oscillation modes.

Project PC5 – Control of an Inverted Pendulum (discrete time)

Same as Predictive Project 1 but implement the system in a real-time environment and try it out on the real pendulum. Successful swingup (by any method) of the pendulum is not needed (but is nice).

Project PC6 – Control of an Inverted Pendulum (continuous time)

Same as Project 2 but implement the system in a real-time environment and try it out on the real pendulum. Approximate the continuous time controller by sampling fast and run the parameter estimator at a slower sampling rate. Successful swingup (by any method) of the pendulum is not needed (but is nice).

Project PC7 – Adaptive Control of the See-saw Process

Try indirect adaptive control of the see-saw process.

Project PC8 – Control of the Helicopter

Try MPC, LQG or adaptive control on the helicopter process.

Project PC9 – Adaptive Friction Compensation

Consider a controller that stabilizes an inverted pendulum or motor process. A simple model of friction leads to a piece-wise linear systems for which the standard adaptive techniques apply. Implement an adaptive friction compensator and explore its properties.

Project PC10 – Model Predictive Control Using CVXGEN or QPgen

Implement a model predictive controller for any suitable lab process using CVXGEN or QPgen to generate fast 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.

Project PC11 – Autotuning of Robust PID Controllers

The goal of the project is to implement automatic tuning on a process with time delay. The project involves use of a new Matlab program for derivation of optimal robust PID controllers, that have been developed at the department. The incorporated PID design method has several advantages to existing methods in industrial autotuners. The program has, however, so far only been used in simulations on models and the project is therefore interesting from a research point of view.

Project PC12 – Control of Ball-and-Beam Process

Try MPC, LQG or adaptive control of the ball-and-beam process. With MPC or LQG you could try to move the ball as quickly as possible between the two end points of the beam. With adaptive control the goal could be to get the adaptive controller to converge before the ball falls off.

Project PC13 – MPC Control of Ball-and-Plate Process

Use MPC, e.g., utilizing the CVXGEN or QPgen toolboxes, to control the position of a ball on a plate. The position of the ball is measured by a camera. The aim of the task is to design the controller so that the ball follows some predefined trajectory, e.g., a circle.

3. 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 DC 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 model, 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, e.g., 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 – 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. 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.

Project 4 – 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). For this project it is an advantage if you have taken the course in Image Analysis.

4. Control-Oriented Projects

Project 5 – "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 6 – "Catch and Throw" Ball and Beam Process in JGrafchart

The same as project 5 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 7 – 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 8 – Control of the Mass-Spring-Damper Process

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

Project 9 – Control of the Linear Pendulum

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

Project 10 – 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 11 – 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.

5. Real-Time Programming Projects

Project 12 – Linux for Control (2PC)

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 simply implement the GUI in Java.

Project 13 – Wireless Control over 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 (e.g., the ball and beam) in Java where the communication is performed via Bluetooth.

Project 14 – Wireless Control Using Android (P)

Smartphones based on Android provide 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 via the Atmel/AVR boards and a Bluetooth dongle, or via the the desktop PCs and a USB Bluetooth dongle.

Project 15 – Wireless Control over 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 at the Ericsson Research datacenter in Lund. The project might be performed on a Raspberry Pi or on an ordinary lab PC, in both cases with an LTE-dongle (one for the sensing side and one for the actuation side).

Project 16 – Control over IoT Using Calvin (CS)

Calvin is a Internet-of-thing (IoT) framework supported by Ericsson Research here in Lund. Calvin is based on dataflow (or actors) programming model where each individual actor is written in Python. In the project you should implement a controller in Calvin and compare the performance obtained when the controller is executing locally on, e.g., a PC, or when it executes in the Ericsson Research datacenter. Optionally your solution should support online migration of your code between the datacenter and the local client. It is an advantage if you have some prior experience of Python.

Project 17 – Evaluation of FreeRTOS (PC)

FreeRTOS (`http://www.freertos.org`) 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 18 – Evaluation of Zephyr (PC)

Zephyr (`http://www.zephyrproject.org`) is a new, lightweight real-time operating system for use on severely resource-constrained platforms. Your task is to evaluate how well suited Zephyr is for use in this course. You should deploy Zephyr on an Arduino board or some other suitable platform. On this board you should implement a small multi-threaded control application.

Project 19 – Effects of SCHED_DEADLINE on the implementation of control strategies (PC2)

In this project you should do an investigation of the effect of the SCHED_DEADLINE scheduling policy on the implementation of control strategies. You should implement one or more controllers for an existing process in C, and to launch the implemented controller with tracing facilities (like trace-cmd) and obtain a trace of the events that occurred during the controller execution. The controller should fire an event every time it starts executing the control algorithm and every time it finishes executing it. The project consists in specifying different scheduling parameters for the controller thread, and in the collection of log of events. These logs can be analyzed in terms of stability of the controlled plant, to find out what is the effect of the scheduling parameters and the scheduling policy on the stability of the closed loop system.

6. Embedded Projects

In these projects you will implement your controller in an embedded computer, using a Linux PC for development and cross-compilation. On the 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 cable) or Bluetooth.

Project 20 – 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 21 – 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 22 – Embeddded Control of the Tiny Demo Process (C)

In the FRT090 project course a tiny servo process was recently developed that easily can be connected to and controlled from a laptop, to be used for demos during, e.g., control lectures. In this project you should instead implement the controller in the Arduino that currently only is used as a signal interface, and have the GUI on another computer.

Project 23 – 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.

The default application is to design and control a Lego mini-Segway. This can be combined with using an Android phone for setpoint generation. You may also 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 24 – Reversing a Lego Mindstorm Trailer Truck

Your task is to design a Lego Mindstorm controller and sensor and actuator system that allows a Lego truck with a trailer to reverse along to some predefined trajectory. A project of this type has previously been performed in the FRT090 course and part of it can be used as a starting point. Unless some other approach is found a vision system with Open CV can be used to measure the angle between the truck and the trailer.

Project 25 – Embedded Control of the Balanduino Balancing Robot (CS)

Design and implement a digital control system for the Balanduino balancing robot. The code will run on the Arduino platform with an ATmega1284P CPU. Use fixed-point arithmetic for the control calculations.

Project 26 – Embedded Control of a Ball in Pipe Process (CS)

A Ball in Pipe Process was recently constructed by students in FRT090 Projects in Automatic Control. The position of the ball is determined using an ultrasound sensor/received at the end of the pipe. Design and implement a digital control system for this process. Depending on your interest, you may modify the process to use some alternative actuator.

Project 27 – Control of the Crazyflie Quadcopter (CS)

Crazyflie (http://www.bitcraze.io/crazyflie-2) is a small open-source quadcopter. On the Crazyflie there is an Arm Cortex processor running FreeRTOS and the motherboard is a Raspberry Pi. A localization system based on UWB anchor nodes is availabe. You task is to develop a controller for the system that allows the UAV to hover and to move between waypoints.

Project 28 – Localization of the Parrot BeBop UAV (CS)

The Parrot BeBop is a larger UAV than the Crazyflie. The task in this project is to port the Crazyflie UWB localization system so that it can be used also on the BeBop. The system should be demonstrated using some simple control scenario.

Project 29 – Autonomous Control of the TurtleBot (CS)

The TurtleBot (http://www.turtlebot.com) is a small ROS-based open source mobile robot equipped with a camera and different types of touch sensors. Your task is to implement a simple autonomous controller that avoids colliding with people (think in terms of an autonomous lawn mower).

Project 30 – Autonomous Racing: Trajectory Control of a Small Scale Racing Car (CS)

Take part in the f1tenth competition (http://f1tenth.org/). This project's task is to write a trajectory controller for an autonomous 1/10th scale F1 racing car. The car is programmed using the Robotic Operating Systems (ROS), with the simple race objective of not crashing and minimizing laptime. The project uses a LIDAR sensor to obtain information about the environment and integrate the received information with a path planner to follow a predetermined trajectory in the smallest possible time.

7. Your Own Ideas

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