

Real-Time Systems: Project Descriptions 2020

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 **March 5**. 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 2020”. The email should contain the **names** and the **email** addresses of **all** the members of the group. Please provide both the StiL addresses and the email addresses that you actually read. The email should also list all other control courses that the group members have taken. The projects will be assigned on March 6-9. 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 27** (but preferably as soon as possible). 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 exercise.
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. A Gantt chart is a good way to visualize the time plan.

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 18**. 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 commented, documented, and 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 Friday **May 18**. 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 same holds for compression artefacts and black background in Simulink plots. Figures and plots should have clear labels and titles. 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 25**, 15:15-17:00.
- An **oral presentation** (10 minutes) should be made on the oral presentation session, **May 25**, 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.

Important dates

- **March 5th**: group composition and project list due.
- **March 6th-9th**: project assignments.
- **March 27th**: suggested solution due.
- **May 18th**: deadline for project report and demonstration to supervisors.
- **May 25th**: demos and oral presentations.

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, Python, or some other programming language required.
- 2** Small project. Only two persons.
- S** Special or research-related project. High risk factor. Only for highly motivated persons or PhD students.

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 oscillatory modes.

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

Same as Predictive Project 1, but you should implement the system in a real-time environment and try it out on the real large Furuta 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 the Predictive Project 2, but you should implement the system in a real-time environment and try it out on the real large Furuta pendulum. Successful swingup (by any method) of the pendulum is not necessary (but is nice).

Project PC8 – Control of the Helicopter

Try MPC, LQG or adaptive control on the helicopter process. In particular MPC is a challenging, but rewarding, project.

Project PC9 – Adaptive Friction Compensation

Consider a controller that stabilizes the DC servo process. A simple model of friction leads to a problem linear in the parameters 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 PID Controllers

The goal of the project is to implement relay-based automatic tuning and test it on some of our laboratory processes.

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.

3. Algorithm-Oriented Projects

Project 1 – Event-Based Sampling and PID Control

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.

Project 5 – Bandwidth Control of Camera Network

Consider a system with many cameras looking at various different locations. Each camera captures a frame of an image and encodes it for transmission. The amount of bandwidth used by each camera can vary depending on the importance of the scene the camera is looking at. Your task is to determine the importance of each camera and regulate its bandwidth. Image encoding is performed using various control techniques. One solution is using OpenCV (open-source computer vision library) with your favorite language (Python, Java, C++) for object detection.

Project 6 – Multi-Process Control on a Single-Core System (S)

Control multiple (2 or 3) Furuta pendulums using only a single-core processor. This introduces problems such as contention for the communication channel and CPU time. The project is therefore complex in the scheduling, control, and communication aspects.

Project 7 – Robust Control of a Furuta Pendulum over Bluetooth (S)

Control of a Furuta pendulum is not a trivial problem. In this project we make it even more difficult by introducing faults in different parts of the wireless system. This project is highly relevant for industry as well as for research and requires a dedicated group.

4. Control-Oriented Projects

Project 8 – “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 9 – “Catch and Throw” Ball and Beam Process in JGrafchart

The same as project 8 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 10 – Stabilization of the Uni-directional MinSeg Segway

The MinSeg is a small Segway-process that is built using, and programmed with, an Arduino. The uni-directional MinSeg has only one motor controller, i.e., it can only drive along a straight line. Your task is to implement a stabilizing controller for this process. This project may not be chosen if you have already taken the Multi-Variable Control course. The system should use wireless Bluetooth communication to a GUI that resides either on a desktop or on a smart phone where signals can be plotted and setpoints and controller parameters changed.

Project 11 – Control of the Multi-directional MinSeg Segway

The MinSeg is a small Segway process that is built using, and programmed with, an Arduino. The multi-directional MinSeg has two motor controllers, i.e., it can also turn. Your task is to implement a controller for this process for. This project may not be chosen if you have already taken the Multi-Variable Control course. The system should use wireless Bluetooth communication to a GUI that resides either on a desktop or on a smart phone where signals can be plotted and setpoints and controller parameters changed.

Project 12 – Collaborating MinSeg Segways (C)

The MinSeg is a small Segway process that is built using, and programmed with, an Arduino. Your task is to implement a system where two MinSegs collaborate. Here wireless communication is necessary. One possibility is a follow-the-leader system where one MinSeg follows the other.

Project 13 – Collaborating MinSeg Segways for Balancing See-Saw (CS)

The MinSeg is a small Segway-process that is built using, and programmed with, an Arduino. Here wireless communication is necessary. The two MinSegs should cooperate in order to balance over a see-saw. Visual feedback from a camera can be used for relative positioning of the MinSegs.

Project 14 – Visual Control of Crane: Rotation/translation

In this project you should control the orientation of a hanging crane load and damp out oscillations in the translation and/or rotation using visual feedback. The control is to be implemented on a Raspberry Pi. The crane is realised by a small robot arm.

5. Real-Time Programming Projects

Project 15 – 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 16 – 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 17 – 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. In order to this project you must use your own Android phone(s).

Project 18 – Robotics as a Cloud Service

Cloud computing is rapidly expanding beyond classic IT web service and entering into the domain of industrial automation. This allows for new application domains to reap the benefit of large scale compute systems. In this project you should implement a cloud-based robotics service. The Robotics Operating System (ROS) will be used in combination with containers to build a cloud service that allows robots to connect to the cloud.

Project 19 – 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 20 – Evaluation of New Real-Time Kernel (PC)

There are many C-based real-time kernels (e.g., FreeRTOS, Zephyr, Miosix), that supports a wide range of microprocessors. Your task is evaluate how well suited one of these is for use in this course, e.g., as an alternative to Java and STORK.

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 based on RS-232 (serial cable) or Bluetooth.

Project 21 – 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 22 – 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 23 – 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 using a Raspberry Pi. Depending on your interest, you may modify the process to use some alternative actuator.

Project 24 – 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 HTC Vive virtual reality system is available. Your task is to develop a controller for the system that allows the UAV to hover and to move between waypoints.

Project 25 – Autonomous Control of the Huskvarna Mower Platform (CS)

The Huskvarna platform is the mobile robotics platform used in the Huskvarna lawn mowers. Your task is to implement a simple autonomous controller that avoids colliding with people (or hedgehogs).

7. Your Own Ideas

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