

MECHENG313 Real Time Software Design Assignment

Part2: Real Time Temperature Control with C#

Objectives:

- To be familiar with designing control system using PC-based interface
- Apply basic control theory to temperature control
- Develop codes that operate a real-time system using C#
- Graphical User Interface Programming using in C# forms application

Grouping and Project Weighting:

Students will work in pairs on the development of the software. A report per group will be submitted. The project (including the associated laboratory exercise) is worth 20% of the overall course mark.

Description:

The aim of this assignment is to develop programme using C# for real time digital control of temperature in a temperature chamber at a nearly constant level above room temperature. The control tasks involved in this project include: real time data sensing of temperature, signal conditioning using digital filters, and control of heating and cooling elements. A user interacts with the programme via a graphical interface. A temperature chamber will be provided.

The Temperature Chamber & The Elvis board:

- There are three devices in the temperature chamber; heating element, fan and temperature sensors. National Instruments provides a C# API that lets you set and read the input and output signals through the Elvis board.
- One digital output channel DI0 is needed to turn on/off the fan.
- One digital output channel DI1 is needed to turn on/off the heater.
- Three analogue input channels AI0, AI1, AI2 are needed to read the measurements of the three temperature sensors.
- Turning on the heater will decrease the values AI0 to AI2. Turning on the fan will increase the values AI0 to AI2.
- The measured values correspond to the voltage at the thermistors used for temperature measurement. In order to give a meaningful user feedback, the measurements must be converted into a temperature value given in degree Celsius. You will need to determine the voltage to temperature relationship and model it in your programme.

Interfacing with the Elvis board:

The interfacing with the Elvis board can be done using the "NationalInstruments.DAQmx" assembly for .NET. Lab sessions will be used to introduce the board and how to interface with it using C#.

Software Specifications

Tasks:

- Continuously read the values from the temperature sensors. The sampling rate of the signal reading must be 10 Hz or higher.
- Denoise (i.e. reduce noise contained in a signal) the signals read from the sensors by applying digital filters then convert them to temperatures.
- Display the current temperatures measured by the three sensors on screen in Celsius and store the history in separate files for further analysis.
- Allow the user to set the desired steady state temperature value range of 2° to 5°C above room temperature.
- Activate the heating element and/or fan to eventually maintain the temperature within $\pm 0.25^{\circ}\text{C}$ of the desired temperature. Initial rise time and overshoot should be minimised.
- See more complete specifications below for the user interface.

Before implementing the controller, you are required to characterise the inherent capability of the chamber. You have to determine the temperature ramp rate of the chamber with the heating element at full power and also the cooling rate of the chamber with the heating element turned off and with the fan turned on.

User Interface:

The user interface should be constructed using a C# Forms application. It should contain components that satisfy the following specifications:

- Display the status of the heater and fan at any time. Both the fan and heater should be off when the programme starts.
- Display the current temperature values from the sensors. It should be updated at 0.5 second intervals. Perturbation of the displayed temperature by noise should be minimised. Every reading should also be written to a file, for subsequent plotting (using Excel or Matlab for example.)
- Display and allow the user to set the desired temperature value.
- Allow the user to set the parameters of the digital filters by reading a parameter file.
- Allow the user to choose which locations of the sensors in the chamber the system controls.
- Allow the user to initiate and terminate the control process.
- Allow the user to interactively adjust the parameters of the control process and to reset any options with neither interrupting the control process while the user makes the changes nor causing stress for the user of waiting for the system to respond to.
- When programme terminates, the heater must be turned off and the fan must run enough time to return the temperature value to around room temperature.

Your interface should give user inputs high priority (i.e. should not freeze up at any state) and involve a minimum CPU overhead.

Assignment Specifications

Submissions:

The submissions will have two components.

1. Report

Write a report (maximum 5 pages including diagrams, graphs, images, and appendices; single line spacing in 12pt font) that describes how the control system was implemented and data or graphs confirming that you have characterised the behaviour of the chamber and that the control specification is met.

The report should cover your major design decisions, which include: i) justification of the chosen controller and digital filters; ii) explanation on how the readings from three temperature sensors were utilised in the control algorithm; iii) explanation on how the voltage-to-temperature mapping has been derived; iv) descriptions about the design of the user interface; and v) comment on how the concepts required in real-time system were deployed. The report should also give: vi) example of the output that demonstrates the correct functioning of the software.

Make sure your report has a clear structure and sections with headlines. One report is to be submitted per group.

2. Set of C# code

Submit a set of electronic copy of your C# code. Your code will be assessed based on the layout and readability (self-explanatory identifiers and good comments). You need to create a folder with your student ID's and save your files on your local drive, then copy the folder into the folder on the S-drive:

S:\Mech\Submit\Mecheng313

One set of code is to be submitted per group.

Deadline: Friday 21 October 2016, 5pm

Place of submitting report will be announced later.

Live demonstration:

Apart from the two deliverables above, each group will be asked to demonstrate the functionality and the user interface of its code. You are expected to give a brief presentation about your software to the assessors and answer any questions that they will ask. Make sure you show the same version of the code you submit to the S-drive.

Demonstration session: Afternoon Wednesday 19 October 2016

Detailed time slot for the demonstration will be announced later.