Report: Miniproject

TTK4147 Real-time Systems

By:
Thorbjørn Sømod, 742379
Andreas Nordby Vibeto, 742387

November 17, 2015

Part 1: PID-controller

We chose to split the project into two different modules in addition to the given source code; message handling and the controller itself. The message handling module initializes the UDP-socket and handles sending and receiving. The controller module consists of two functions; one main loop handling duration and period, while the other function calculates the output of the controller. In this task we use two threads; one for the controller and one for receiving messages.

For the controller parameters we used the ones given in the task, and chose not to use a derivative term. We chose the controller period first by trial and error, and then we tuned it by looking at the graphs generated by the server. By making sure that the period time between GET requests always stays above the controller step time from GET request to new setpoint received, we know that no deadlines are missed. Our final period was 5.5ms. This way we get a good response for our system, while we still have a good margin before any deadlines were missed.

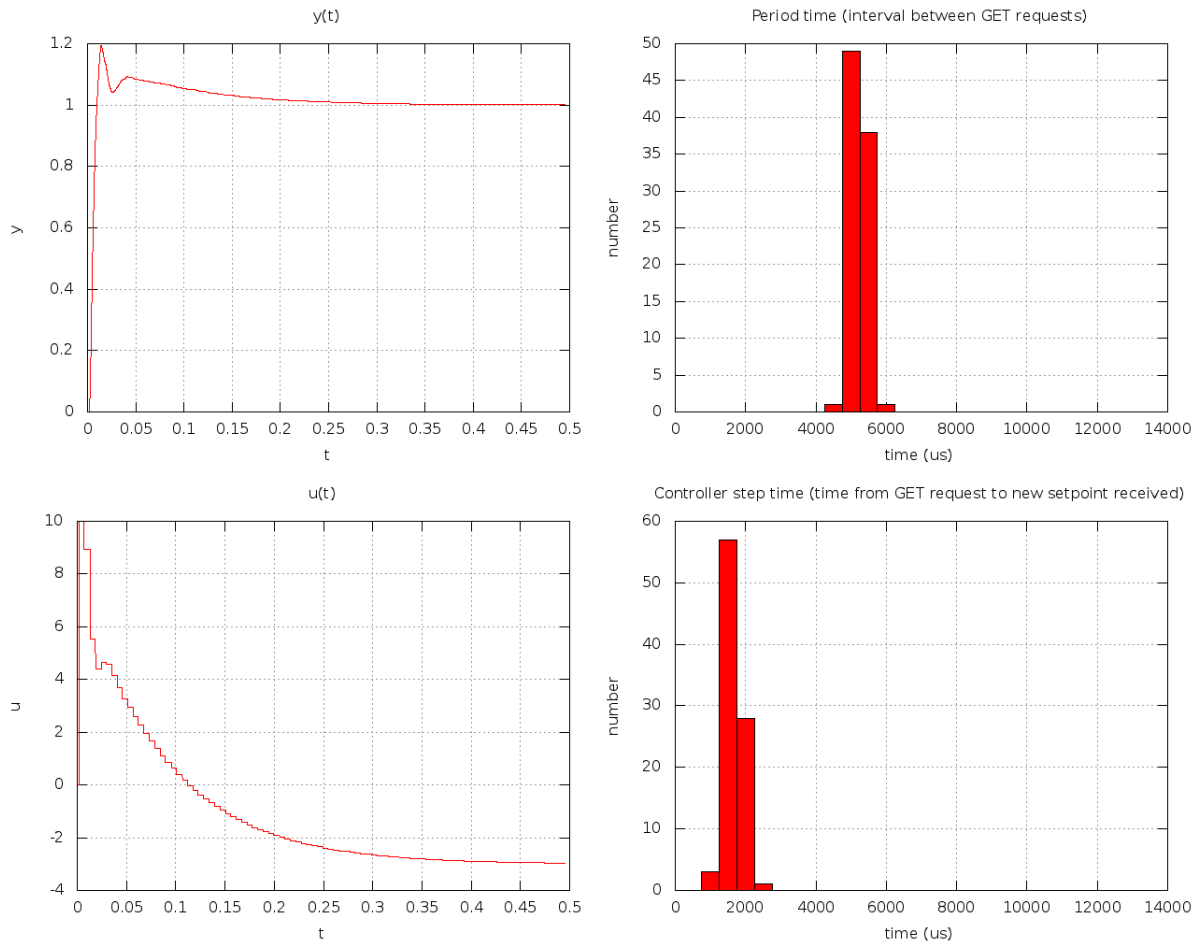

Figure 1: Plot for task 1

Part 2: Signal response

In this task we added a thread to the message module. This thread sends a response to the server when it receives a signal. This is done by using a semaphore that is signalled when the listening thread receives signal from the server.

When adding this third thread we had to increase the period for the controller to be sure that no deadlines were missed. We increased the period to 8ms, which still gave us a good response for the system and no deadlines were missed.

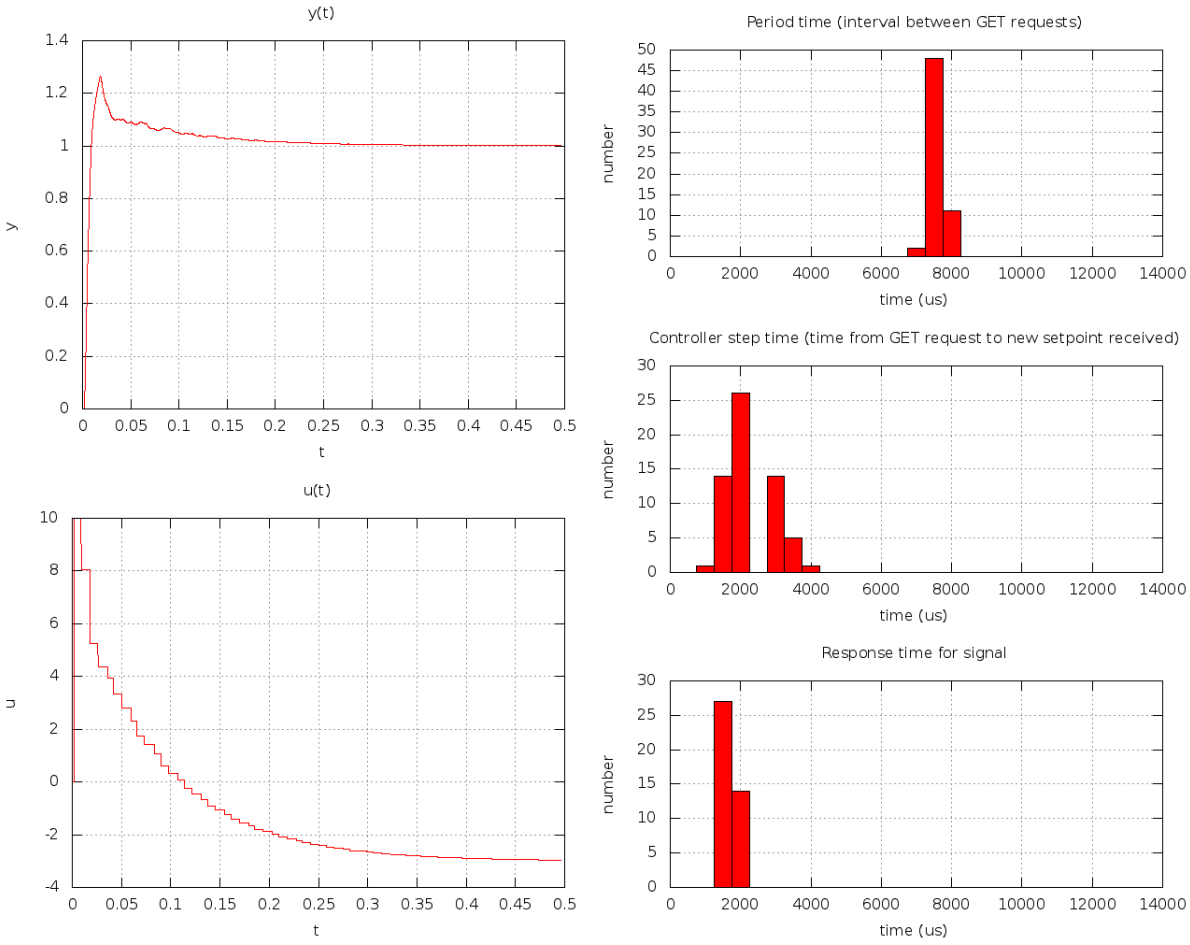

Figure 2: Plot for task 2