Sanntid Miniproject

Eivind Eigil Bøhn, Mads Åsholt Rolsdorph, Tharald Jørgen Stray

November 2016

1 Part 1

We used the provided controller parameters, and found that a shorter period gave better performance, i.e. less overshoot oscillations, with the trade-off being some missed controller deadlines. We therefore settled on a period of 2500 μs, resulting in some oscillations but no missed deadlines. We settled on 3 threads, one for listening on the UDP connection, one for sending over UDP, and one for the controller. We decided to keep a separate sender thread so the other thread didn't have to wait for the connection to be available and could continue with their work. We created a message queue using the FIFO module from exercise 9, that the other threads put their messages in, and the sender thread reads from. The queue is controlled by a semaphore. We also implemented a function that calculates remaining time to sleep to ensure a consistent period for the controller.

2 Part 2

The code in this part is pretty much the same as in part one. We extended the listener to put signal_ack messages into the queue when signal messages were received. We increased the period to 3000 μs to compensate for the slightly increased computing time of processing the signal messages. With this period we experienced more oscillations but we were very fast to respond to the signals, and we still didn't miss any deadlines. In both parts we achieve the reference after approximately 0.2 seconds with zero steady-state error. We are confident the oscillations could be lessened/removed by further tuning of the controller, which is outside the scope of this miniproject.

Figure 1: Part 1: Results with a period of 2500 μs

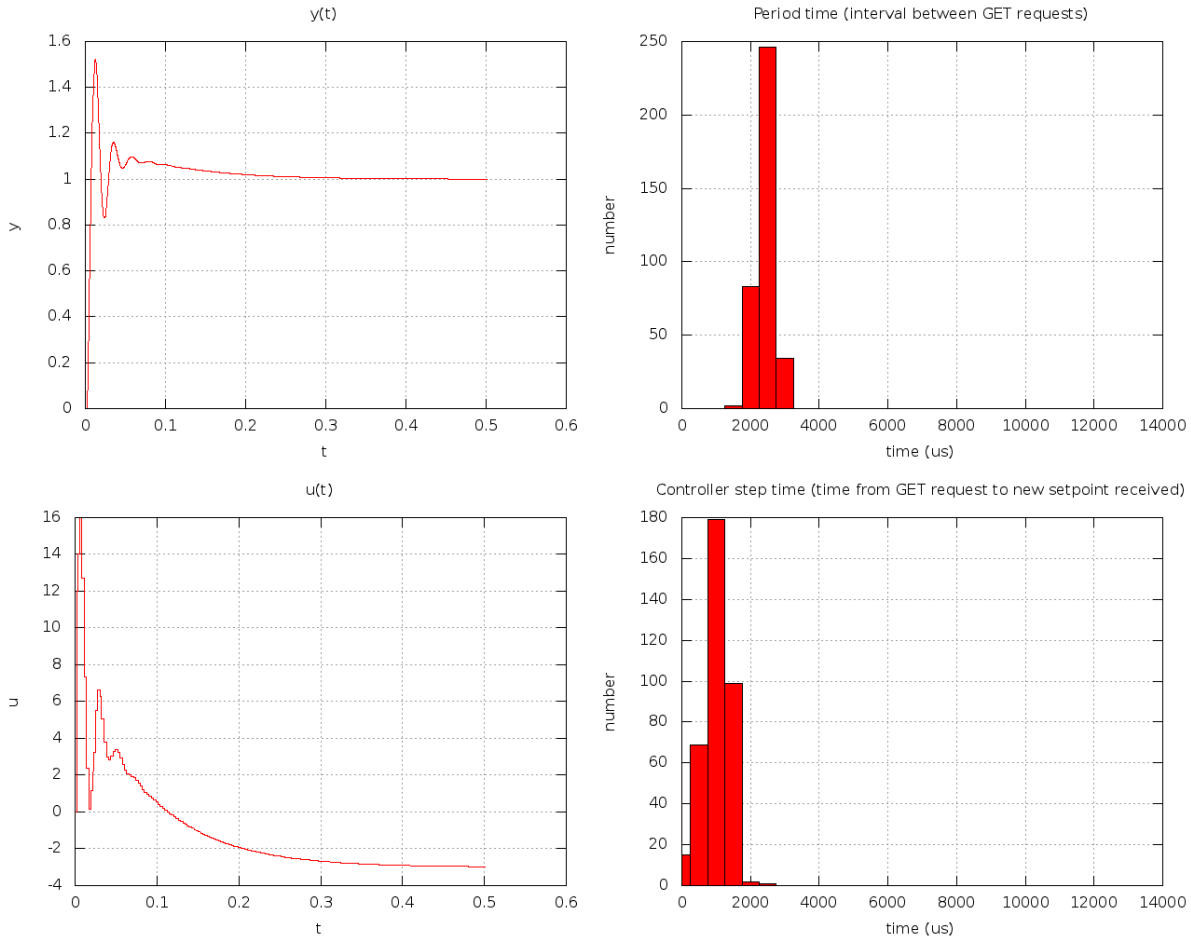

Figure 2: Part 2: Results with a period of 3000 μs