Real Time Systems TTK4147 - Mini project

Jon Eirik Lisle Andresen – 742410
Sondre Foslien – 750398

Oktober 2017

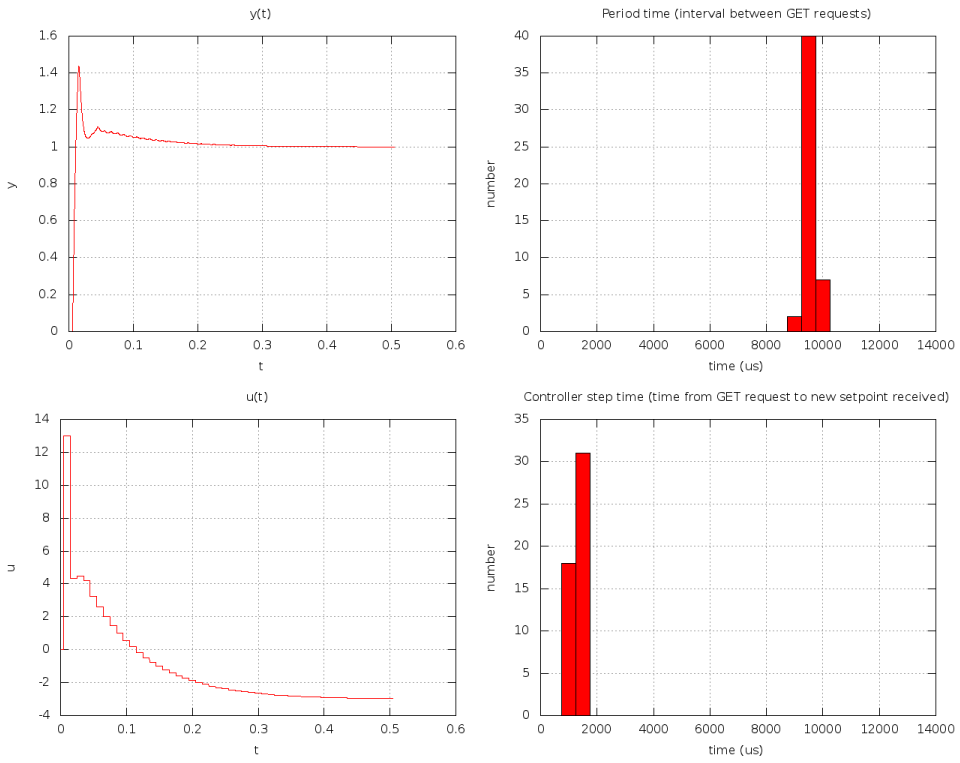

Figure 1: Figure of the controlled system output $y(t)$, the input $u(t)$, the interval between GET requests and interval from GET request to a new $u(t)$ was recieved.

1 Exercise 1

The controller period was chosen as $T_P = 0.01$s as this was largest period where we would still get a stable system. This can be explained by the fact that the smallest time constant of the system to be controlled was $T = 0.014$s and the controller needed to be faster than this to control the system. Also the period could not have been much shorter as the resolution chosen for the timer on the AVR was $T_r = 0.001$s and we wanted to have more than just one timer tick available between each calculation of control input.

We also chose to implement a PID rather than a PI to control the overshoot, but still achieve rapid convergence. The chose to use threads also in exercise 1 to allow for easier expansion.

The shared variable yMsg was protected by a mutex as it was our way of communicating between the UDPlistener thread and the controller. Condition variables were used to signal the controller, setU, that a new measurement of y had arrived and that a new u was to be calculated.

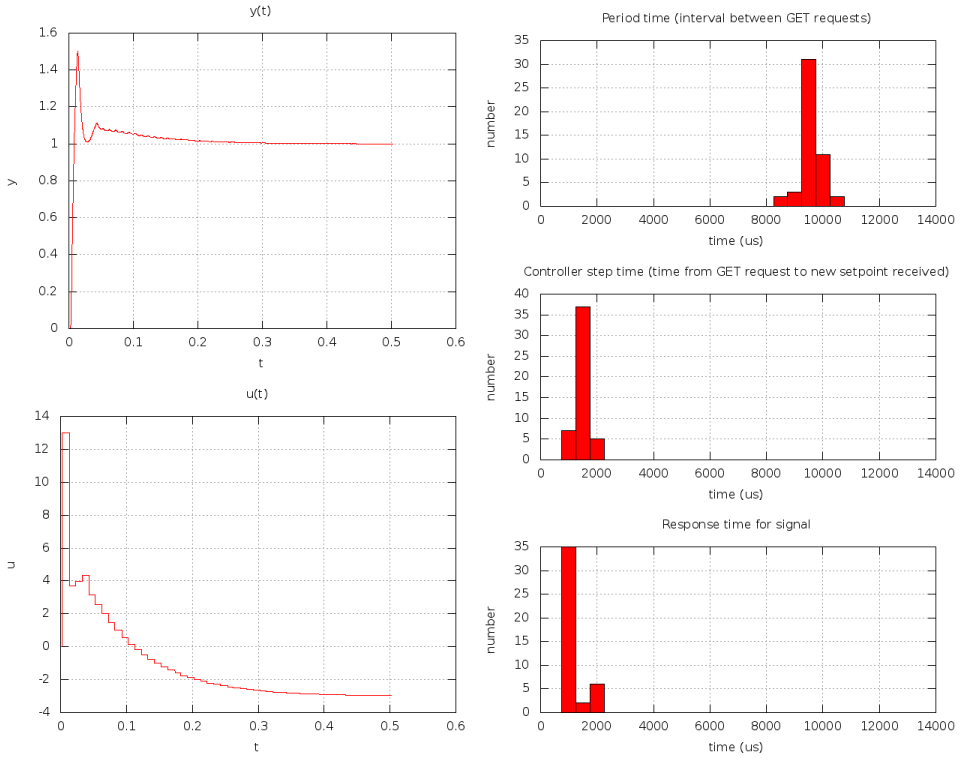

Figure 2: Figure of the controlled system output $y(t)$, the input $u(t)$, the interval between GET requests, interval from GET request to a new $u(t)$ was recieved and the response time of the signal.

2 Exercise 2

The design was the same as in part 1 except for a new thread which handled SIGNAL-messages.

The addition of signals affected the system by spreading out the times between GET-requests and the controller step-time. The controller was made worse by the addition of the signal, but not to a large degree.