

Homework 2 - Implementation of plant and controller with Pthreads

Mikael Tulldahl, 901007
Björn Nilsson, 880302

Saturday 23rd May, 2015

1 Implementation

Two Pthreads were implemented to get the behaviour asked for. The states of the controller is maintained inside the controller thread so that the simulated plant can be replaced by a physical plant later on. The implementation was pretty straight forward and the code can be viewed in Chapter ??, "*Implemented C-code*", below.

To achieve a constant *looptime*, `clock_nanosleep` with `TIMER_ABSTIME` flag was used to get absolute time instead of relative as with *nanosleep* which was supplied in the template code.

The use of a *semaphore* made sure that the shared variables was not accessed at the same time by the process and the controller. The semaphore was initialized with "`sem_init(&s, 0, 1);`" saying that it is of type *boolean* and was set to "busy" with the command "`sem_wait(&s);`"

A PID controller was used as a controller and it was tuned by manual tuning using an training set of setpoint values containing steps and noise. The tuning was performed iteratively and was aimed to get as quick step response as possible without overshoot and with minimal oscillations.

2 Results

The resulting behaviour to an reference input can be seen in Figure 1 below. Here we can observe that the controller manages to regulate the output very well to what is demanded even though the reference has discontinuous steps.

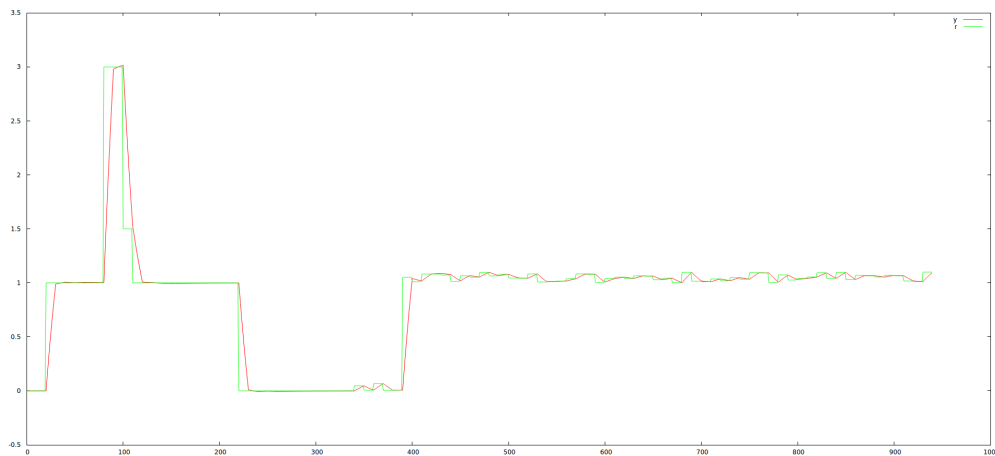


Figure 1: Plot with the reference and the output