

Real-Time Systems

Laboratory 1

Filip Birkfeldt

1. Sample Time

If h very small, then we sample too often. This results in that the computer can no longer keep up with the computations that are required. Also, the systems become much more “accurate”, it will try to adjust every movement of the ball. Which results overshooting and rigid movement.

If we don't sample often enough, the system will not now where the ball actually is. It will try to adjust itself to a position where the ball no longer is. In return, the system will try to fix this by adjusting itself very radical. Which results in overshooting.

2. Three shared resources dealt with extra care

The inner and outer regulators should be dealt in with extra care, to avoid race-conditions. Also, the data that is being shared and dealt with in the synchronized methods. But also the reference generator.

3. Synchronized methods in PI & PID-classes

To avoid race conditions, for example if the program itself wants to update the parameters but the user too wants to update the parameters. Then, the synchronized methods will prevent a race conditions between the user and the program itself.

4. Poor design to add an integrator for the inner loop

Since integrators adds delay but also instability in the inner loop. The inner loop needs to be faster than the outer loop. Though, an integrator must be used at the outer loop to prevent stational errors.

5. Duration less than or equal to zero

The controller that makes the computations is too slow for the system. It will not keep up with the sampling time, which results in that the systems falls behind the sampling time.

6. Why we are using feedforward

We are using feedforward to lower the effect of measurable disturbances. But also to improve the response to the command signals.