

1. Change the sample time of the system. What happens with the system dynamics when the sample time becomes very small and very large?

When the sample time is set to very small the console will start printing “lagging behind...”. This means that the execution time is longer than the sampling time and as a result we might not be updating the states correctly. The control signal also becomes more jittery as the sample time becomes smaller.

When the sample time becomes very large the system starts to oscillate and becomes completely unstable. We might also get aliasing since we might be sampling the reference signal too slow compared to how long the reference signal itself updates.

2. Give at least three shared resources that we have to handle with extra care.

- The controller states
- The PID and PI controller parameters
- The current mode

3. Why do we put the synchronized keyword on every method in the PI and PID classes? Motivate.

Because all the methods in the PI and PID classes use the controller parameters. For example one method in PID/PI is used to set new parameter values and another method is used to get the parameter values. In that case if the setParameters method is called and then the getParameters method is called before the setParameter method is finished setting all the new parameters we don't want getParameters to return execute until setParameters is finished setting all the new parameters. Otherwise getParameters might return some of the new parameters and some of the old. The synchronized classes make sure that only one thread can access the shared resources at a time.

4. Why would it be poor design to add an integrator to the inner loop controller?

The beam process is already modeled as a pure integrator and therefore, if we were to add an integrator to the inner loop, it would add another pole at the origin of the system which would result in an unstable system.

5. **In the course so far we usually sleep threads using the following code snippet:**

```
t += h; // t was the previous release time and h is sample time  
duration = t - system.currentTimeMillis();  
if (duration > 0) {  
    try { sleep(duration);  
    } catch (InterruptedException x) {  
        // Do something  
    }  
}
```

Explain what it means for the controller that duration is less than or equal to zero.

It means that the execution time is longer than or equally long as the sampling time. If the duration is less than zero this will result in the system to start lagging behind which might cause problems later on when the controller states are updated, as stated in question 1. If the duration is equal to zero that means that the execution time is exactly as long as the sample time.

6. **Briefly describe why we are using feedforward.**

Feedforward is a way of predicting ahead of time what controller output is required based on your knowledge of the setpoint change. This means that feedforward can give a quicker and better response to a change of reference value since it will account for the change in the reference value before it has actually affected the system.

We use feedforward since it as stated above gives a faster and more accurate response to a reference signal/setpoint change. Especially in the time optimal case feedforward can be used to optimize the control of the system since the reference signal changes slowly in a predictable way. But when the reference signal instead is set to square it can be seen that the system does not follow the reference signal as well as in the time optimal case which is because the change in reference signal happens instantly.