**ENEL441**

**Lab 4 Digital Control loop implementation**

**2020**

**Introduction**

The objective of this lab is to introduce the digital implementation of a compensator for a SISO control loop. Methods of simulation, analysis and design will be developed based on using the Matlab control toolbox and Simulink. The design will follow the conventional methodology of starting with a continuous time compensator design and then converting the time domain state space formulation into a discrete time variant resulting in a recursive difference equation. This can be implemented directly into an embedded processor. From there on additional features can be added based on the simplicity of implementing digital processing control.

Consider the digital control loop is shown in Figure 1. The input reference is rk which is a time sampled sequence. The output of the plant is a continuous time function given as y(t). This goes into the ADC channel and is converted to a digital number representation. The error is formed from  and is represented by a signed integer. The object of the control loop is to make the magnitude of the error small after a short transient and with little overshoot. This will of course be limited by the dynamics of the reference sequence rk, the plant and the loop type.

Plant

Hp(s)

compensator

Hc(z)

sensor feedback

ADC

y(t)

+

-

rk

ek

yk

DAC

ck

u(t)

***Figure 1***  *Negative feedback loop based on a digital compensator*

The control loop compensator will typically be implemented based on a small microprocessor with an ADC sampling of a sensor input. The microprocessor will have a limited processing speed and therefore will be processing the feedback signal calculations based on fixed point integer arithmetic. We generally will not have the luxury of using full floating point sample representation. The compensator algorithm is usually based on a recursive matrix equation that can process a state space matrix representation. In this lab we will only consider a SISO problem and hence the compensator implements a recursive difference equation as discussed in the lecture. The output of the microprocessor is a DAC that converts the outcome of the compensator recursive equation into analog form. The DAC will include the equivalent of a ZOH (zero order hold) as the analog output is held approximately constant from one sample to the next.

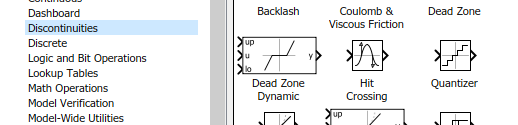
**Q1** Start with a continuous time Simulink simulation of a proportional negative feedback loop. Assume that we have a plant given by transfer function of

Determine the constant b such that the rise time is 10 msec. Set up a Simulink model to simulate the step response of  with a step generator source and a scope sink to verify the step response.

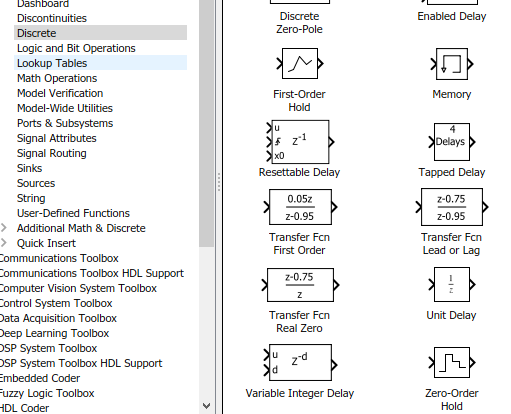
Next it is desired to have a proportional control feedback loop that speeds up the step response of the plant such that the closed loop response has a rise time of 2 msec. Simulate the negative feedback loop. In Matlab provide a comparison plot of the plant step response and the feedback closed loop response with the 2 msec rise time.

**Q2** Simulate the discrete time implementation of the control loop of **P1**  by assuming that the gain constant is implemented by the microcontroller. Also the reference in rk is discrete time. The feedback from the plant is based on an ADC with a resolution of .01 volts and the overall sampling rate is 25 kHz. Your Simulink output should compare the outputs from the discrete sampled loop and the continuous time output of part **P1**. Note observations regarding the comparison of the discrete time version and the continuous version.

For this part we can assume that the reference is a discrete time sampled function of rk where k is the time index and sampled at 25 kHz. The feedback from the output can also be assumed to be sampled at 25 kHz. In Simulink you can find Quantizer useful for quantizing the feedback and reference signal to .01 volts.



In discrete you find the ZOH which sets up the sampling rate of 25kHz. As well this is the fundamental block to implement a DAC response sampled at 25 kHz.



The details of the Simulink library may look slightly different due to the different versions of Matlab.

**Q3** As before we assume that we have a plant given as

where the time constant is 10 msec. Design a compensator that will result in a type-1 loop where the compensator is of the continuous time form of



The target rise time is 2 msec with as small an overshoot of less than 5%. Use SISOtool to design this.

**Q4)** Take the continuous time compensator from before and convert this into a discrete time representation. Use c2d() to convert the compensator into a discrete time version that is sampled at 25 kHz. Write out the recursion equation and explain how this can be implemented in a microprocessor. Show the discrete time system response to a step input in Simulink comparing the continuous and the discrete time responses.

The Matlab routine c2d(), translates the transfer function into Z domain transfer function . From this a recursion equation can be determined. For instance suppose we have and error as and the plant drive signal given as . Then we have the transform relation in the z domain as



where  is the z domain transfer function corresponding to the c2d() output.

As an example, suppose that the conversion from the continuous time compensator transfer function to a discrete time Z transform transfer function (based on using c2d()) of



converting into a recursive equation we get

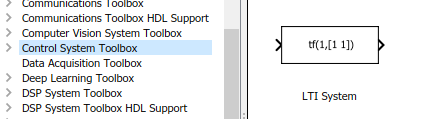


which gives the recursive equation of

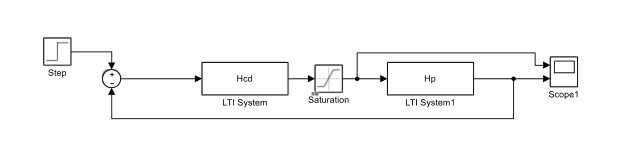


The update of the present compensator output value in terms of the previous compensator output and the present and previous values of the error.

Note that for the actual Simulink simulation the control toolbox has a block of an LTI system which is convenient to use for this question but not necessary.



**Q5** Next generate a plot of the output of the control loop at the output of the plant and a plot of the drive signal into the plant. Note that the initial drive voltage is large due to the effect of speeding up the step response. Now consider saturating the plant input drive signal at a maximum of 2 and a minimum of 0 volts and note the difference in the plant output. You can use the saturation block as in the Simulink loop below



**Q6 Design of Drone height control**

Consider a simplified drone model for height control that has a propeller with electric motor for vertical thrust and an ultrasound sensor for measuring of vertical distance from the ground. The frequency domain model of the vertical thrust motor is given as

in terms of newtons of thrust. The input is current that is source from a programmable current source. That is for a steady current of 1 ampere, the thrust will settle to 1 newton of force. The motor and propeller are approximated as being a linear system. Assume that the mass of the drone is 1 kg.

The ultrasonic height measurement system transmits and processes 100 pulses per second and provides a precise measurement of the height.

Design a height flight controller for the drone that is stable based on feedback from the ultrasonic every 100 msec. The criteria for the controller are:  
a) must accept a desired reference height as input.  
b) have zero steady state error (for a step reference input)  
c) be stable when it hovers with a constant input height reference.   
d) have minimum overshoot when it reaches the reference height  
e) have the shortest transient   
f) current into drone motor cannot exceed 300 amps  
g) assume that the microcontroller can solve an update for the recursive control equation every 10 msec.

1. Start with a design based on a continuous time compensator.
2. Then convert this into discrete time and determine the recursion equations that need to be implemented.
3. Complete a Simulink simulation showing the step response of the drone height transient. Note the sensitivity to the current limit.

One of the advantages of the digital control is that you can put in additional processing rules that will mitigate such things as the erratic behavior when the current limit is applied. For instance when the difference between the reference input and the actual height is larger than a certain amount you may turn on the motors for a specific time duration regardless of the feedback. That is you over-rule the negative feedback control. See if you can devise a rule that will mitigate the poor drone control performance when current limiting is added.