FRTN15 Predictive Control—Home Work 1
Signals and Systems

In this homework exercise we recapitulate theory for discrete time signals and systems in assignments 1-3. Recursive Least square estimation (RLS) is treated in assignment 4. The exercise also gives the opportunity to practice Julia/Matlab/Simulink.

E-mail your detailed and motivated solutions in pdf-format to christian.rosdahl@control.lth.se. Attach any code and models you might have used. Include figures in the pdf report, and verify that any included code is formatted such that copy-paste works (this is not always the case when mcode.sty is used).

1. Sample the (continuous time) system

$$G_{y,u}(s) = \frac{s+1}{s^2}.$$

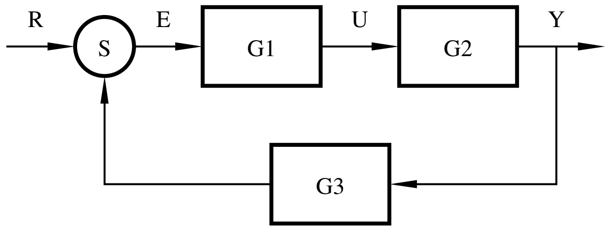
Provide the transfer function and a state space representation. Conduct the calculations

a. by hand, with a parametrized sample period h.

b. using julia/matlab, with sample period $h = 0.1$.

c. How does the sampling period affect the result?

2. Give the transfer function $H(z)$ from r to y of the interconnection below, where $H_1(z) = z + 2$, $H_2(z) = \frac{1}{z^2+2z+1}$ and $H_3(z) = \frac{z}{z+2}$. Also plot the response y when r is a step function.



3.

a. Show that the system H given by

$$x(kh+h) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix} x + \begin{pmatrix} \frac{h^2}{2} \\ h \end{pmatrix} u(kh)$$
$$y(kh) = \begin{pmatrix} 1 & 0 \end{pmatrix} x(kh)$$

is not asymptotically stable. Here $h > 0$ denotes the sampling period.

b. Stabilize H by means of linear state feedback. The resulting system should have all eigenvalues placed in the origin; use $h = 0.1$. (Hint: The command `acker` in the matlab control system toolbox can be useful.)

c. Placing all eigenvalues in the origin is referred to as deadbeat control. Give an interpretation, motivating this nomenclature. What potential drawbacks are there, compared to less aggressive eigenvalue placements? Hint: Study how the feedback gain changes when h approaches zero.

4.

a. Use Julia/matlab/Simulink to demonstrate RLS identification of $\Theta = [a\ b]^T$ for $y_k = ay_{k-1} + bu_{k-1} + w_k$, where w_k is a normal white noise process.

Hint: If you use Simulink, the the block 'Matlab function' might be useful. Also type '`help persistent`. at the Matlab prompt.

b. Comment on the choice of input signal and how it affects the result.

c. Assume a is time varying, how can the method be modified to identify a? How do you use information about typical speed of change of a to tune the algorithm?