# Loopshaping Design for Fifth-Order Analog Plant

Digital Control System Design

Professor Fuller

# Final project Canvas assignment page

## Lab 4: Loopshaping control of 5th order plant (final project)

Published | Edit

Requirements:

1. **Unlike all previous labs and homeworks, collaboration with others on this project is prohibited.** Do not discuss anything related to this project with anyone, other than Professor Fuller or the TA Johannes James, prior to the time when reports for it are due.
2. Address your questions regarding this project exclusively to Johannes James at jmjames@uw.edu or Professor Fuller at minster@uw.edu.
3. Should you discover what you think might be an error in the project assignment or the MATLAB/Python and Simulink files provided, please notify the instructors over email as soon as possible.
4. Watch for announcements posted to the discussion board of corrections or clarifications.There may be updates to this page, but all updates will be listed at the top of the discussion board for this lab.
5. In your initial design work in Matlab/Python, you are to use the nominal component values for the resistors and capacitors comprising the analog plant. You may use measured values or other observations as part of your discussion of the differences between theoretical and HIL tests.

- Assignment
  - TBP ← PDF of this Document
- Support files
  - Python:
    - AEM581_lab4_class.ipynb ← Matlab and Python design files

# What to Submit

1. One PDF approximately one page of text (not including figures, equations, code, or appendices)
   a. Describe what you did
      i. Hardware build and brief description of the 5th order analog plant.
      ii. Loop-shaping design to meet design objectives. Include one frequency response plot each for $C(z)$ and $N(z)$. Use the plots to explain how you chose your $C(z)$, and $N(z)$.
      iii. Your *kff* gain, and $C(z)$, $N(z)$ transfer functions with corresponding difference equations. Make sure the coefficients are easy to copy-paste into python for instructors.
      iv. The python code snippet you used to implement your difference equations.
   b. The results you obtained
      i. Time domain response plots of the Hardware-In-Loop (HIL) system. Use the same *kff*, $C(z)$, and $N(z)$ you obtained from theoretical design which met the design objectives.
      ii. Explain why your HIL test did not perform the same as your theoretical design. Hint: this is expected!
   c. Your recommendation on the suitability of this control strategy

2. Attachments or appendices
   1. Your Circuit Python code which you ran on the Itsy Bitsy M4, as a .py or text in appendix.
   2. Any loopshaping design code you used.
   3. The .csv datafiles you used to generate response plots.
   4. Anything extra which you wish to include, these may or may not factor in grading if it demonstrates your understanding of the material taught in this course.

# Procedure

1. Read and understand this document and monitor the discussion board for assignment updates/clarifications. Reminder: do not discuss this final project with anyone except instructors.

2. Design your control using a loopshaping design for $C(z)$ and $N(z)$ to meet performance objectives
   *i.* $K_{ff}$ to be the inverse of the $u-\text{to}-y$ DC gain of your Analog Plant.
   *ii.* $N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of your Analog Plant and replaces them with poles having more desirable dynamics.
   *iii.* $C(z)$ to be a first-order transfer function.
   iv. Your goal is to meet design objectives, at least in theory, using your system model

3. Construct your $5^{th}$ order plant (diagram below) using your existing $4^{th}$ order system:
   i. Interpose the rolloff filter in series with the output $u(t)$ from pin A0.
   ii. Add the input noninverting summer. This will entail repurposing the single channel of the rail-to-rail MCP6004 quad-op IC which you used to buffer the analog output u(t) in the previous lab.
   iii. The components used are within 5--10% tolerance of nominal value. You may pick up the components per announcement from MEB115. Remote students have had them mailed.

4. Test your design HIL on your breadboarded $5^{th}$ order plant using your pyboard and the provided class_Lab4.py Following the procedure from previous labs.
   i. In particular, edit the code where indicated to:
      - add your $C(z)$ and $N(z)$ coefficients.
      - implement your $N(z)$ and $C(z)$ in discrete time, i.e, your $N(k)$ and $C(k)$ difference equations. The informational slides below show one recommendation.
      - change experiment parameters such as the reference and disturbance inputs.

# Design Objectives

1. Sampling period = 0.025 sec. ⟵ Important!

2. Peak overshoot of the $y(t)$ response to an $r(k)$ step < 2%.

3. Time for the $y(t)$ response to a unit step $r(k)$ input to settle to within 2% of its steady-state value ≤ 1 sec.

4. Peak $|u(k)|$ in response to an $r(k)$ unit step < 2+$|V_{OP}|$ volts.

5. Steady-state $y(t)$ response to a $d(t)$ step = 0+$|V_{OP}|$ volts.

6. Time for the $|y(t)|$ response to a $d(t)$ unit step to settle to within 0.1+$|V_{OP}|$ volts < 30 sec.

7. Peak of the $|u(k)|$ response to a $d(t)$ unit step < 2+$|V_{OP}|$ volts.

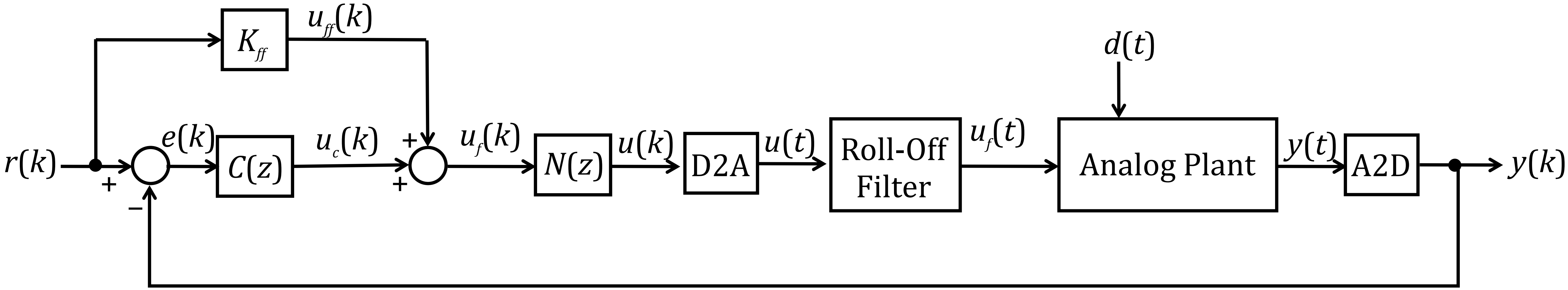
8. For the loop breaking point labeled $u(k)$ in the system diagram, Positive Gain Margin > 20 dB.

9. For the loop breaking point labeled $u(k)$ in the system diagram, Negative Gain Margin = $-\infty$ dB.

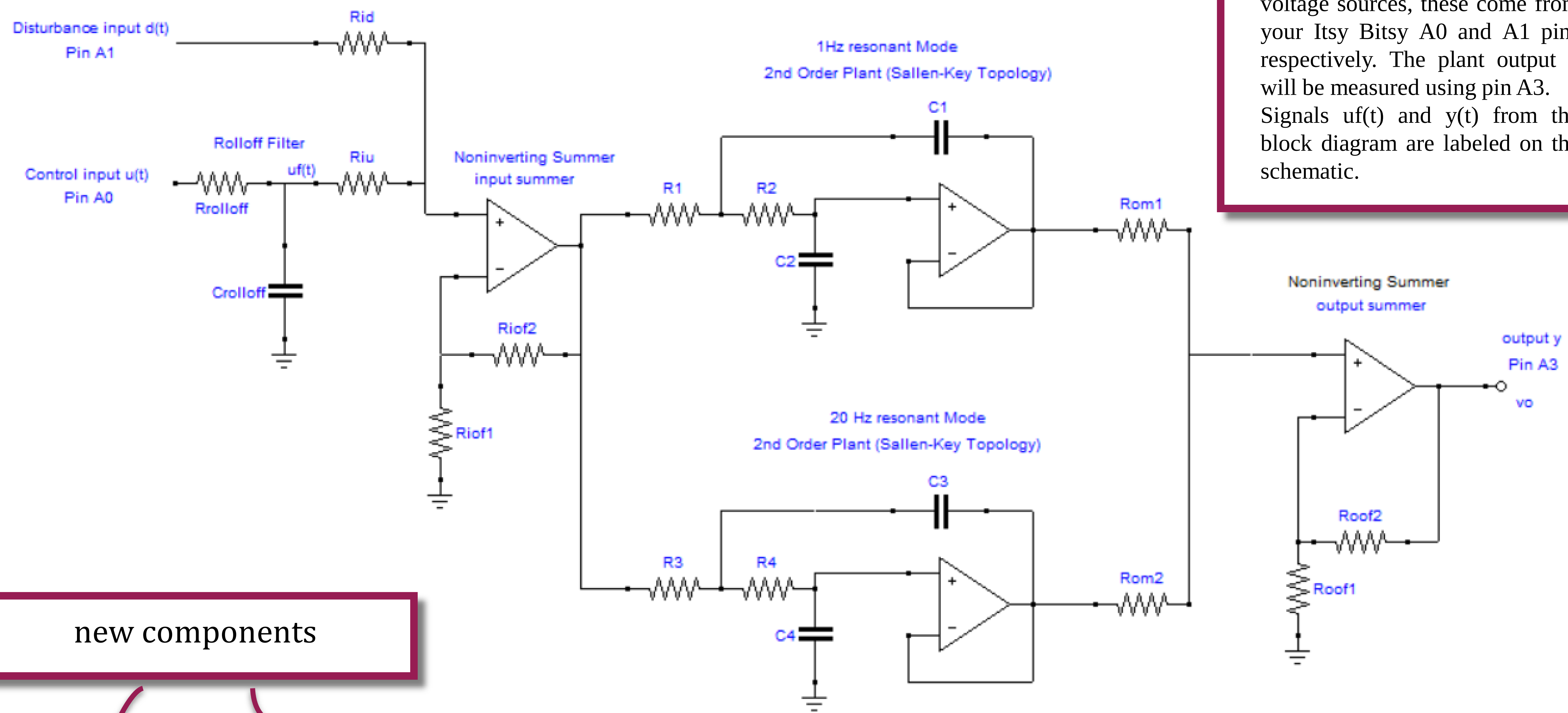
10. For the loop breaking point labeled $u(k)$ in the system diagram, Phase Margin > 90 degrees.

In design work, use $V_{OP} = 0$ volts. In HIL testing, it is advised to use the non-zero default value, $V_{OP} = 1.5$ volts.

# Block diagram of this lab



# Circuit schematic for Analog Plant




Inputs u(t) and d(t) shown as voltage sources, these come from your Itsy Bitsy A0 and A1 pins respectively. The plant output y will be measured using pin A3. Signals uf(t) and y(t) from the block diagram are labeled on the schematic.

new components

Roll Off Filter
Rrolloff = 5 kΩ
Crolloff = 10 uF

Input Summer
Riu = 200 kΩ
Rid = 200 kΩ
Riof1 = 10 kΩ
Riof2 = 10 kΩ

Same as lab 3

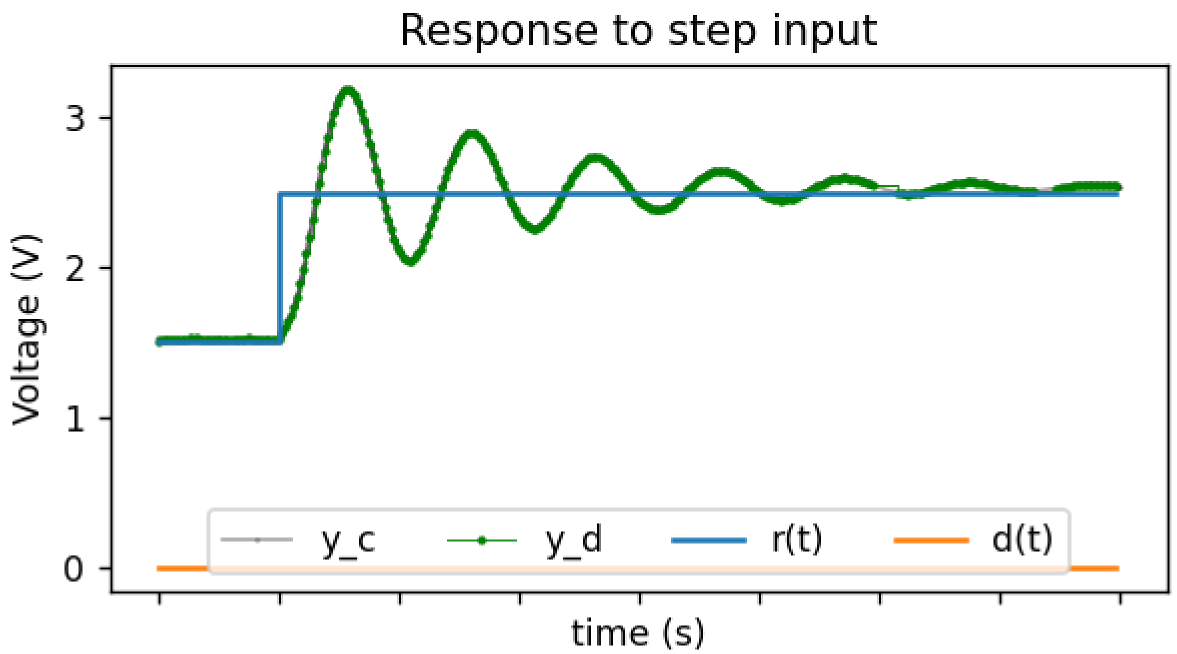
1Hz Vibration Mode
R1 = 160 kΩ
R2 = 200 kΩ
C1 = 10 uF
C2 = 0.082 uF

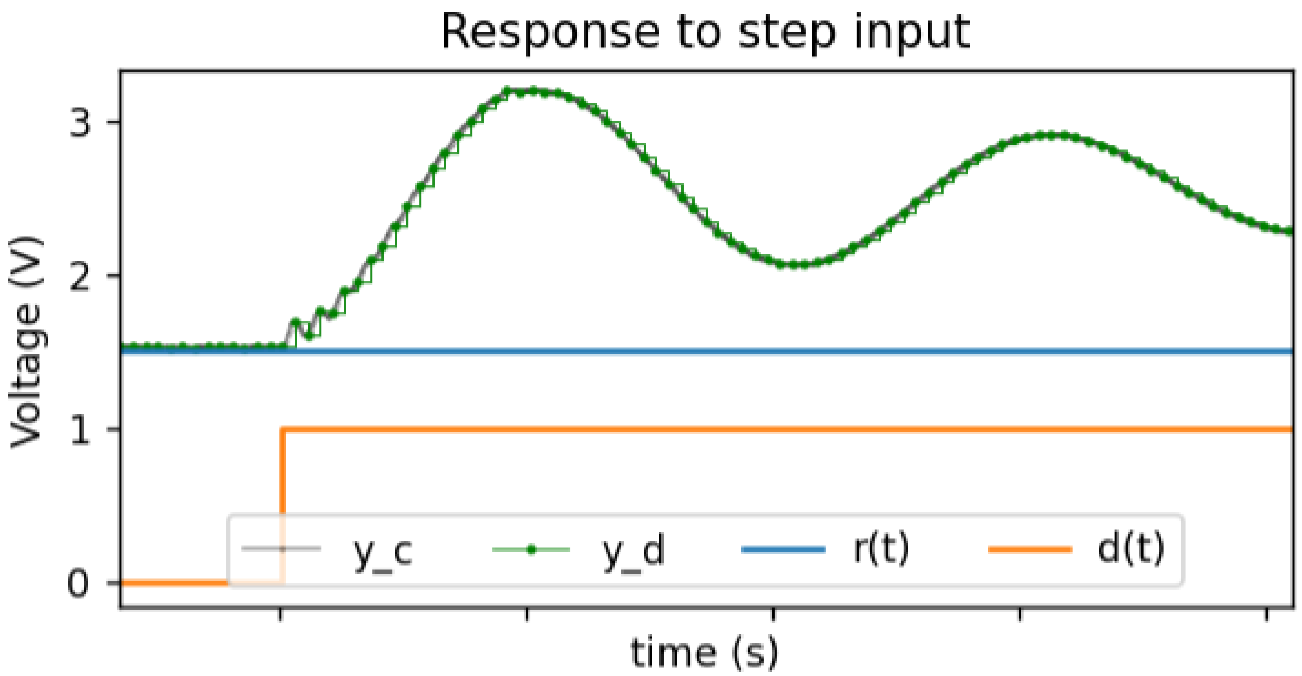
20Hz Vibration Mode
R3 = 68 kΩ
R4 = 13 kΩ
C3 = 6.8 uF
C4 = 0.01 uF

Output Summer
Rom1 = 1 kΩ
Rom2 = 10 kΩ
Roof2 = 100 Ω
Roof1 = 10 kΩ

The remainder of the slides are information and suggestions

# Expected response with correct circuit and default class_Lab4.py code

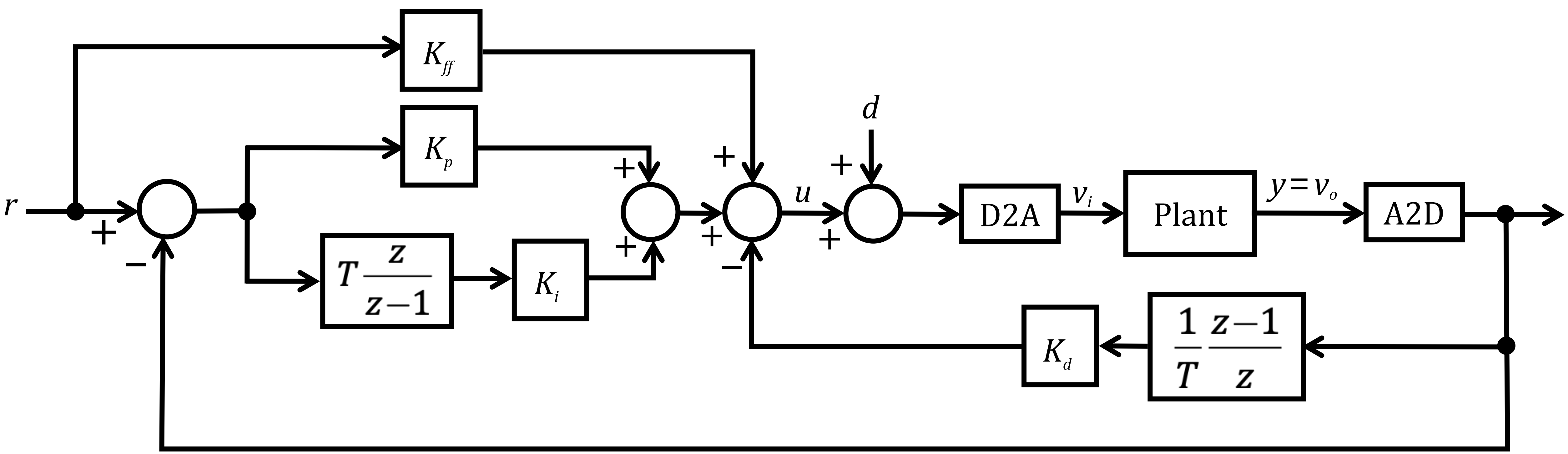


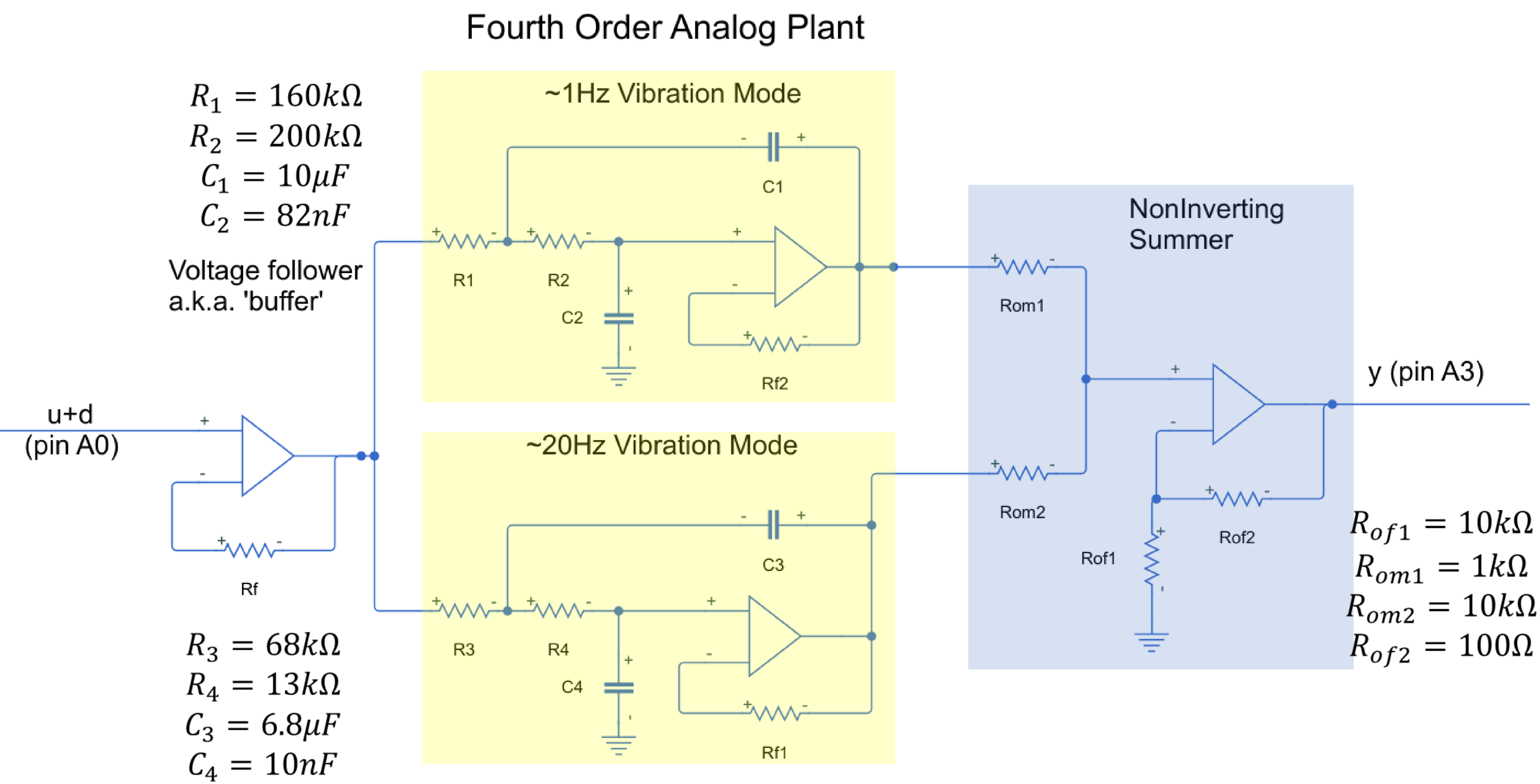

Response to step inputs with default $N(z)$ and $C(z)$ and difference equation stand-in code.

- Reference step (left) features expected ~1 Hz underdamped mode ringing, but the ringing of the ~20 Hz mode is much reduced due to the roll-off filter having correctly attenuated the higher frequency components of the $u(t)$ input, thereby reducing the excitation of the 20 Hz mode.
- Disturbance step (right) shows much more ringing of the ~20 Hz vibration mode because the disturbance input is delivered directly to the plant and does not pass through any roll-off filter.
- Both responses start at the default *Vop*=1.5v and both appear to be settling to a DCgain of about 1.0v/v, indicating that the input summer circuit (and output summer circuit and rolloff filter for that matter) are correct.
- The output summer should be unchanged from last lab, and its expected behavior is the observed weighted addition of the two modes, ~10-to-1 weight of the 1 Hz mode to the 20 Hz mode.

# Previous lab
# PID control of 4th order analog plant



## Previous lab

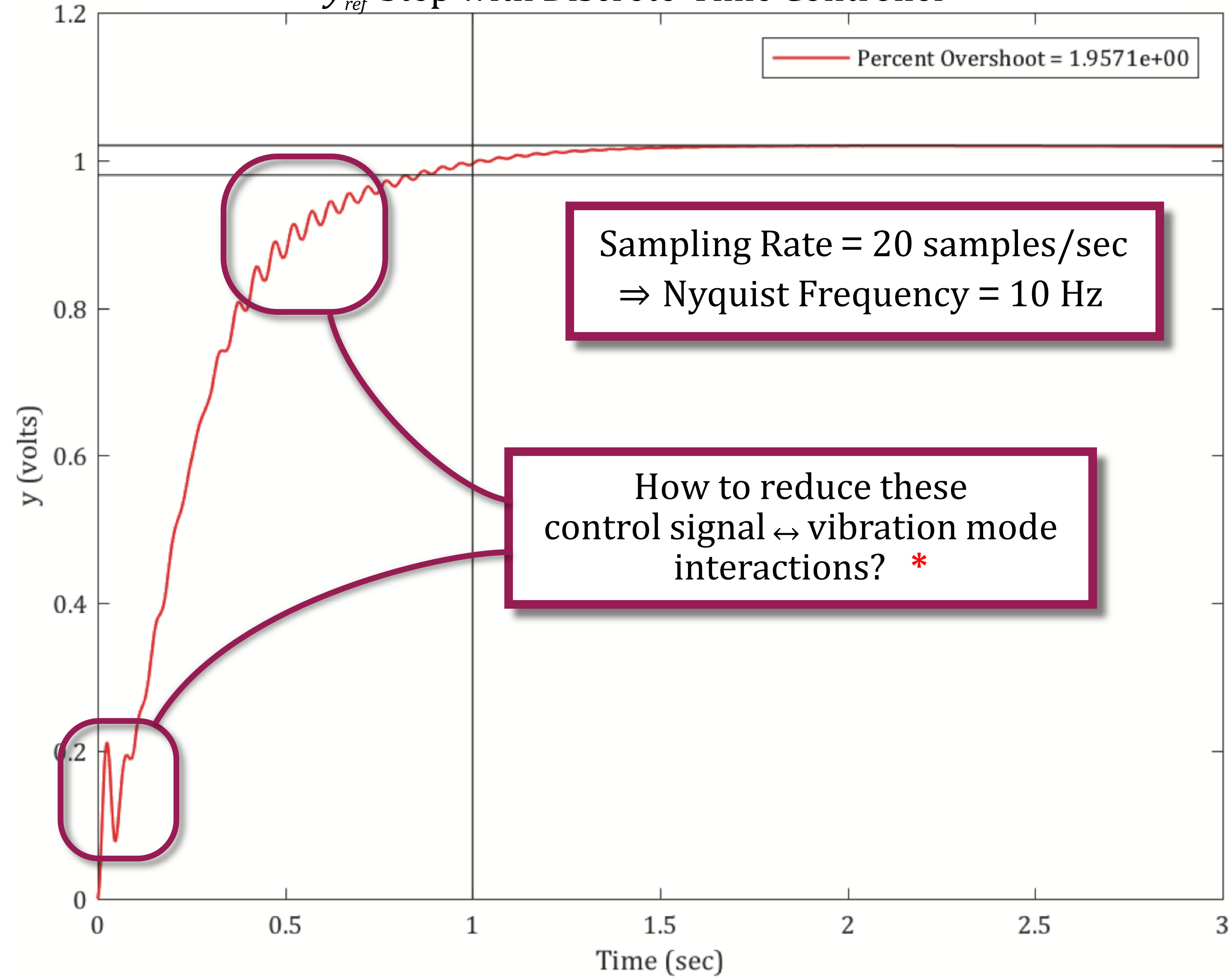




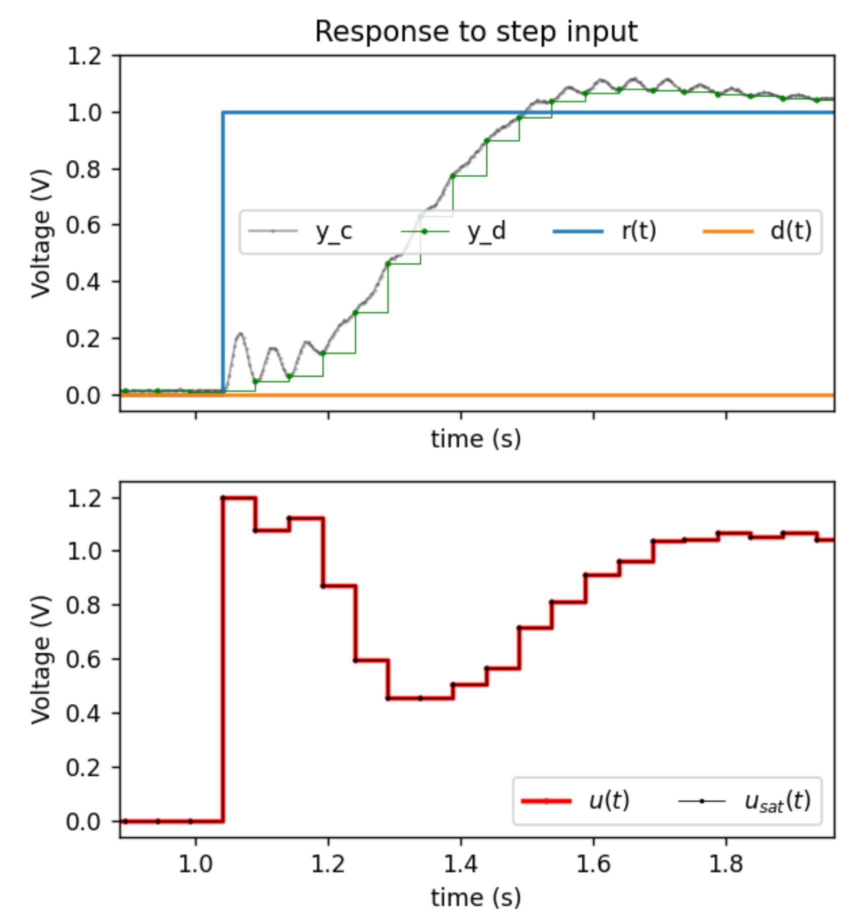

Electrical schematic for your fourth-order analog plant. The heart of the 4th order plant consists of the two individual 2nd order oscillators. The inverting summer circuit (blue) has a transfer function which achieves the addition operation of the outputs from each of the vibration modes, weighted according to the $R_{om2}$ and $R_{om1}$ resistors. Two input signals, $v_D$ and $v_U$ are summed together in the software before the DAC. The 2nd order oscillators each have their own characteristics as shown in the expanded form of the plant transfer function below.

# Previous lab
# PID control of 4th order analog plant

Closed-loop $y$ ContinuousResponse of 4th Order Analog Plant to a $y_{ref}$ Step with Discrete-Time Controller

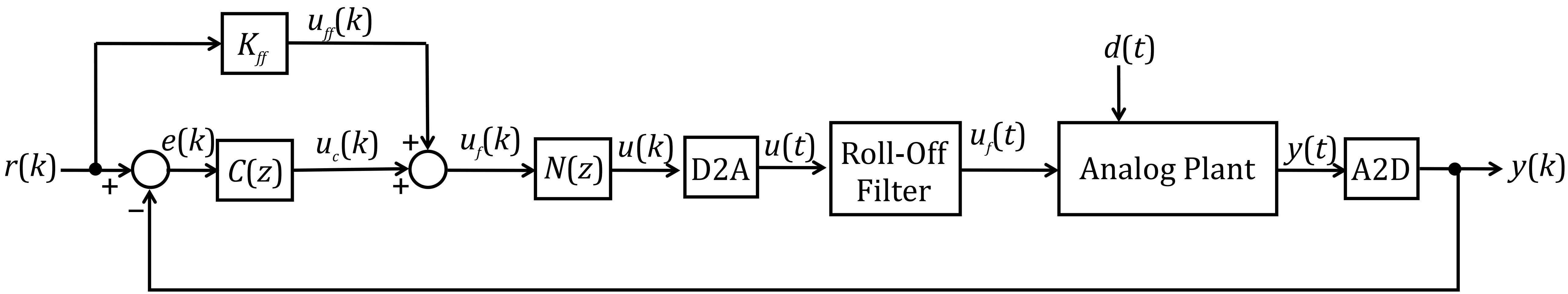



HIL response showing discrete controller response




* Review the “PID4thOrderSolutionDiscussion.pdf”

## New Configuration

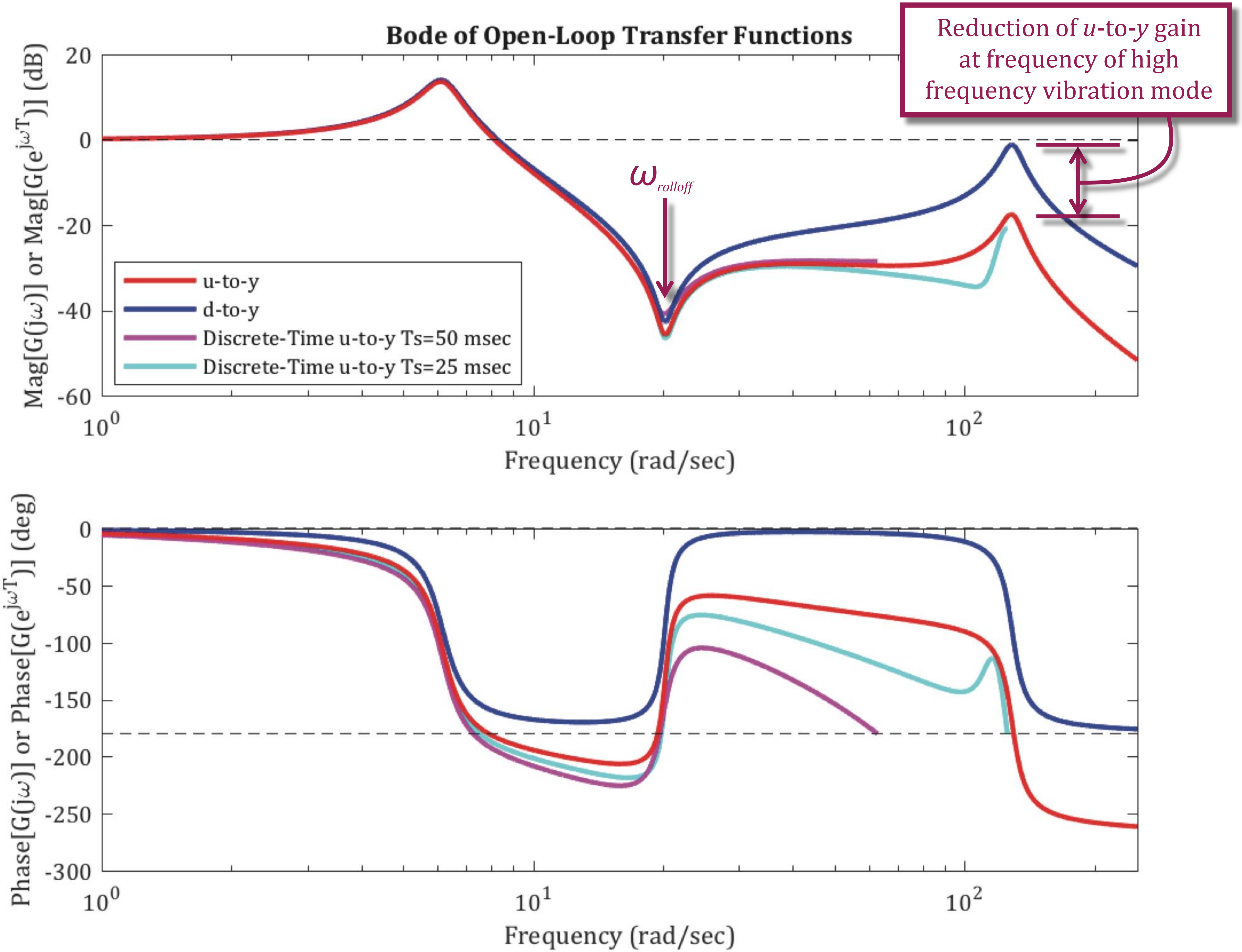



The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

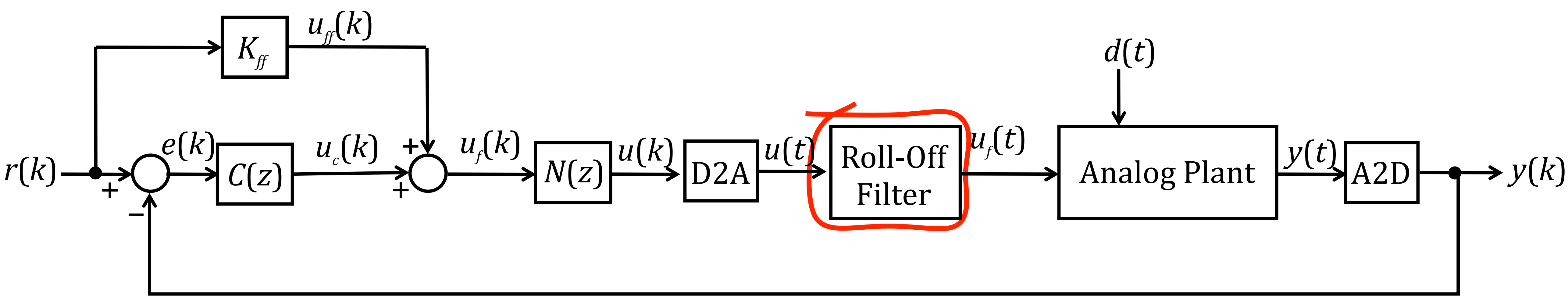
$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the analog plant and poles that have more desirable dynamics.


Bode of Open-Loop Transfer Functions
Reduction of u-to-y gain at frequency of high frequency vibration mode
ω rolloff
u-to-y
d-to-y
Discrete-Time u-to-y Ts=50 msec
Discrete-Time u-to-y Ts=25 msec
Mag[G(jω)] or Mag[G(e^{jωT})] (dB)
Frequency (rad/sec)
Phase[G(jω)] or Phase[G(e^{jωT})] (deg)
Frequency (rad/sec)

## New Configuration



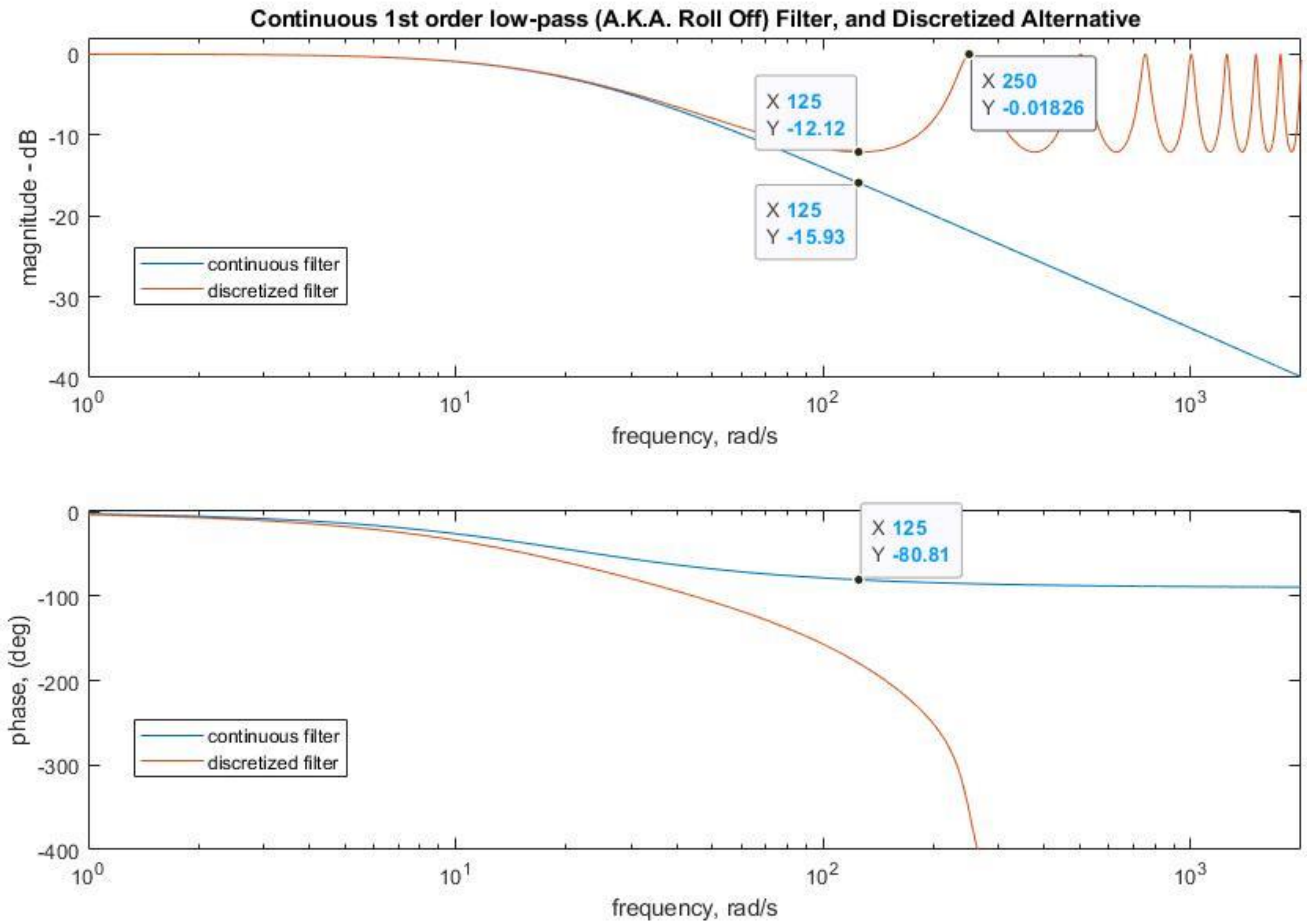

The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the analog plant and poles that have more desirable dynamics.

# Why not a digital roll-off filter?




1) A digital filter would give you a “filtered” output which is a periodic function of frequency. Therefore it cannot possibly attenuate signals having higher frequencies than its own sampling rate. Those frequencies get aliased, and as far as the filter is concerned, they are ‘low’ frequency (on the primary strip), so it doesn’t attenuate them!
← *That* happens

2) The output would still be a staircase function!

## From LoopshapingDiscClass.m

```matlab
[Ap,Bp,Cp,Dp]=analogplant;
plant=ss(Ap,Bp,Cp(1,:),Dp(1,:)); % Drop ydot output of the plant

% Choose the sampling period
Ts=inptdf('\nEnter the sampling period in msec [25]',25)/1000;

% Get ZOH-equivalent discrete-time model of continuous-time plant
plantdisc=c2d(plant,Ts,'zoh');

% Define global variables for Simulink model
global Tsslink
global Kffslink
global Acslink Bcslink Ccslink Dcslink
global Afslink Bfslink Cfslink Dfslink
global Apslink Bpslink Cpslink Dpslink
global yrefslink dslink

% Set feed-forward control gain to the inverse of the DC gain
% of the plant's u-to-y transfer function
Kff=1/bode(plant(1,1),0);
```

# Design Procedure

2. Design your control using a loopshaping design for $C(z)$ and $N(z)$ to meet performance objectives
   *i.* $\boxed{K_{ff} \text{ to be the inverse of the } u-\text{to}-y \text{ DC gain of your analog plant.}}$
   *ii.* $N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of your analog plant and replaces them with poles having more desirable dynamics.
   *iii.* $C(z)$ to be a first-order transfer function.

## From LoopshapingDiscClass.m

```
[Ap,Bp,Cp,Dp]=analogplant;
plant=ss(Ap,Bp,Cp(1,:),Dp(1,:)); % Drop ydot output of the plant

% Choose the sampling period
Ts=inptdf('\nEnter the sampling period in msec [25]',25)/1000;

% Get ZOH-equivalent discrete-time model of continuous-time plant
plantdisc=c2d(plant,Ts,'zoh');

% Define global variables for Simulink model
global Tsslink
global Kffslink
global Acslink Bcslink Ccslink Dcslink
global Afslink Bfslink Cfslink Dfslink
global Apslink Bpslink Cpslink Dpslink
global yrefslink dslink

% Set feed-forward control gain to the inverse of the DC gain
% of the plant's u-to-y transfer function
Kff=1/bode(plant(1,1),0);
```

# Design Procedure

2. Determine:

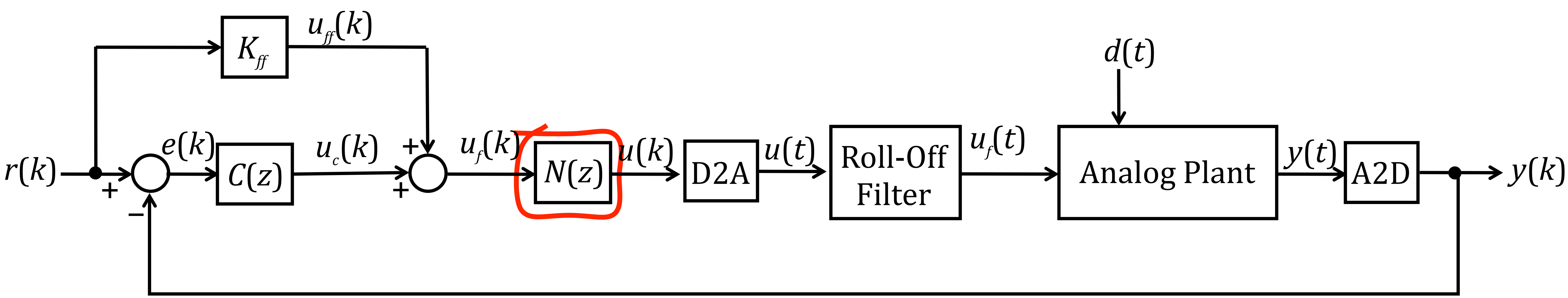
   *(a)* $K_{ff}$ to be the inverse of the $u-\text{to}-y$ DC gain of your Analog Plant proceeded by the roll-off filter . This value will depend upon the resistances in your analog plant, particularly the feedback resistors in the summing amplifier circuits which determine the gain of the summing amplifier.

   *(b)* $N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of your analog plant and replaces them with poles having more desirable dynamics.

   *(c)* $C(z)$ to be a first-order transfer function.

   to meet, on-paper (i.e., based solely on analyses and simulations using your very best model of your analog plant and Roll-Off Filter), all of these design objectives:

# Configuration

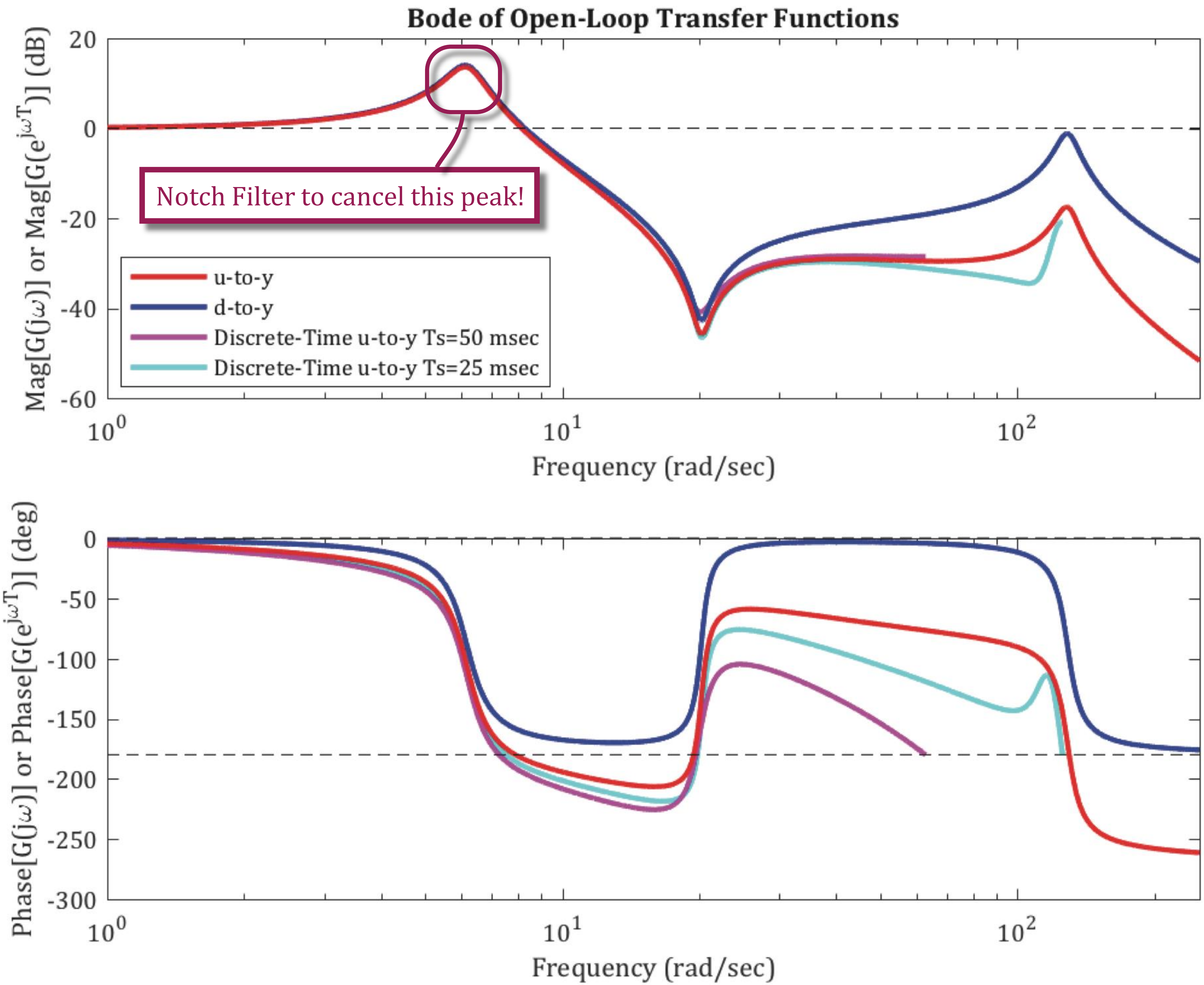



The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

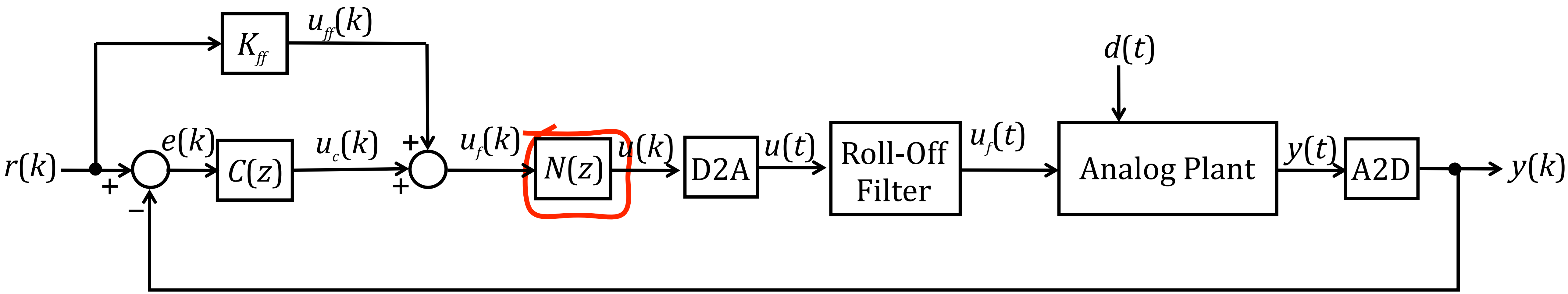
$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the analog plant and poles that have more desirable dynamics.


Bode of Open-Loop Transfer Functions
Mag[G(jω)] or Mag[G(e^{jωT})] (dB)
Notch Filter to cancel this peak!
u-to-y
d-to-y
Discrete-Time u-to-y Ts=50 msec
Discrete-Time u-to-y Ts=25 msec
Frequency (rad/sec)
Phase[G(jω)] or Phase[G(e^{jωT})] (deg)
Frequency (rad/sec)

## Configuration




The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the analog plant and poles that have more desirable dynamics.

# From LoopshapingDiscClass.m

```
%###################################
%######  Notch Filter Design #######
%###################################
% Add notch filter that cancels the open-loop plant's low-frequency
% vibration mode poles with zeros and replaces same with a complex
% conjugate pair of poles having specified damping ratio and natural
% frequency.

omegan_notch_zeros=inptdf('\nEnter desired natural frequency for notch filter zeros (rad/sec) [6.1733]',6.1733);
zeta_notch_zeros=inptdf('\nEnter desired damping ratio for notch filter zeros (dimensionless) [0.091118]',0.091118);
omegan_notch_poles=inptdf('\nEnter desired natural frequency for notch filter poles (rad/sec) [6.1733]',6.1733);
zeta_notch_poles=inptdf('\nEnter desired damping ratio for notch filter poles (dimensionless) [0.091118]',0.091118);

notch_numerator  =[1  2*zeta_notch_zeros*omegan_notch_zeros  omegan_notch_zeros^2]/(omegan_notch_zeros^2);
notch_denominator=[1  2*zeta_notch_poles*omegan_notch_poles  omegan_notch_poles^2]/(omegan_notch_poles^2);
notch=tf(notch_numerator,notch_denominator);

options = c2dOptions('Method','tustin','PrewarpFrequency',omegan_notch_zeros);
notchdisc=c2d(notch,Ts,options)
```

# Design Procedure

subset of full design procedure (end of document)

2. Determine:

   *(a)* $K_{ff}$ to be the inverse of the $u-\text{to}-y$ DC gain of your analog plant proceeded by the roll-off filter. This value will depend upon the $R_{outnum}$, $R_{outden1}$ and $R_{outden2}$ resistances in your analog plant.

   *(b)* $N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of your analog plant and replaces them with poles having more desirable dynamics.

   *(c)* $C(z)$ to be a first-order transfer function.

   to meet, on-paper (i.e., based solely on analyses and simulations using your very best model of your analog plant and Roll-Off Filter), all of these design objectives:

## From LoopshapingDiscClass.m

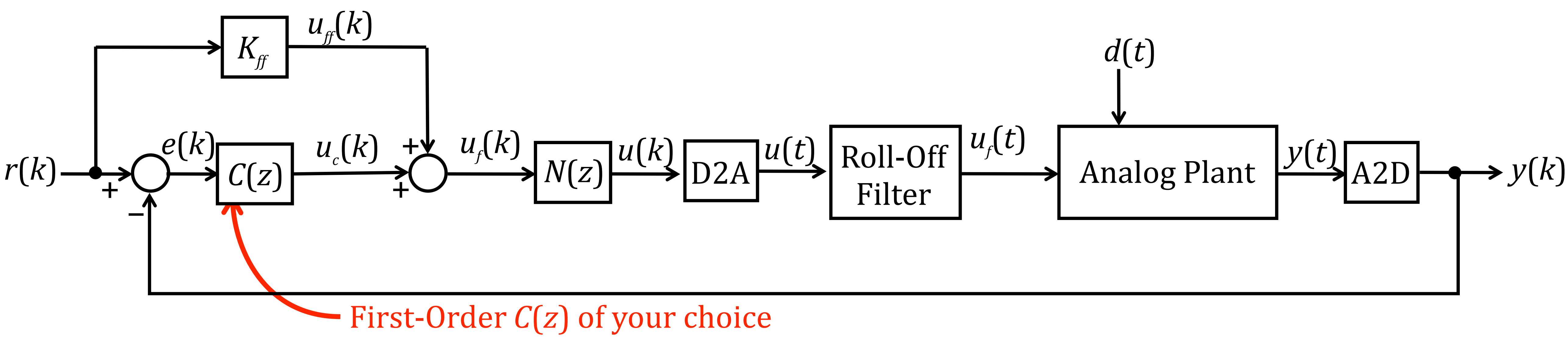
```
%##################################
%######  Compensator Design #######
%##################################
% Add code that develops the discrete-time compensator (C(z)) here.
% Or you can modify this entirely generic code:

b1 =                  1;  % Numerator's coefficient of z^1
b0 =               -0.5;  % Numerator's coefficient of z^0

a1 =                  1;  % Denominator's coefficient of z^1
a0 =               -0.5;  % Denominator's coefficient of z^0

compensatordisc=tf([b1  b0],[a1  a0],Ts) % compensatordisc is C(z)
```

# Configuration




The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the Simscape Electrical Analog Plant and poles that have more desirable dynamics.

# Implementing difference equations

You put a bunch of work into designing a digital controller or a filter, potentially something very high order.

But how do you actually code up a transfer function in 'z' to use it?

Here is where "the rubber hits the road"!

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_{n-1} z^1 + b_n z^0}{a_0 z^n + a_1 z^{n-1} + \cdots + a_{n-1} z^1 + a_n z^0}$$

Assume a transfer function with a numerator and denominator of the same order (same power of $z$)

- Proper (and therefore "causal")
- If the numerator is lower order than the denominator, you will need to set certain $b$ coefficients to zero.

$$a_0 y(k+n) + a_1 y(k+n-1) + \cdots + a_{n-1} y(k+1) + a_n y(k) = b_0 x(k+n) + b_1 x(k+n-1) + \cdots + b_{n-1} x(k+1) + b_n x(k)$$

Inverse z-transform and rearrange.

- ***Assume all initial conditions are zero***

$$y(k+n) = \frac{1}{a_0}\Big[\big(b_0 x(k+n) + b_1 x(k+n-1) + \cdots + b_{n-1} x(k+1) + b_n x(k)\big) - \big(a_1 y(k+n-1) + \cdots + a_{n-1} y(k+1) + a_n y(k)\big)\Big]$$

Solve for the most recent output

$$y(k) = \frac{1}{a_0}\Big[\big(b_0 x(k) + b_1 x(k-1) + \cdots + b_{n-1} x(k-n+1) + b_n x(k-n)\big) - \big(a_1 y(k-1) + \cdots + a_{n-1} y(k-n+1) + a_n y(k-n)\big)\Big]$$

Shift in time

$$= \frac{1}{a_0}\left[\sum_{i=0}^{n} b_i x(k-i) - \sum_{i=1}^{n} a_i y(k-i)\right]$$

Express as summation

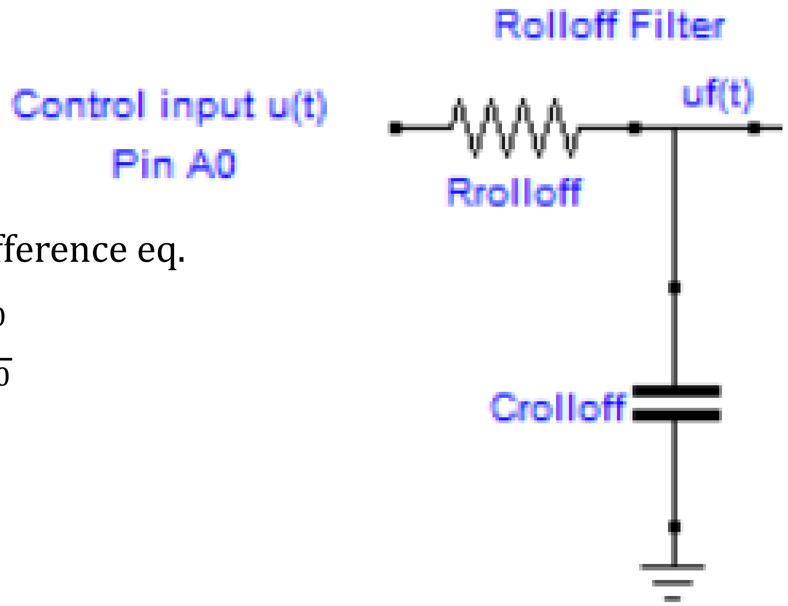
```
def dot(a, b): # dot product of two lists
    out = 0
    for idx in range(len(a)):
        out += a[idx] * b[idx]
    return out
```

One possible way to compute these sums is to recognize that they are the dot product (or "inner product") of the polynomial coefficients and the signal history, e.g. `dot(num, x_history)`

Note: because numerical precision negatively impacts high order difference equations, higher order DE's are often decomposed into multiple second-order systems which are cascaded together to yield an equivalent result. This technique is called a "Bi-quad cascade" of second-order sections.

# Implementing difference equations

Example: a <u>digital</u> 1st order low-pass filter




$$G(s) = \frac{Y(s)}{X(s)} = \frac{1}{\tau s + 1}$$ $\tau = RC$ is the "RC time constant"

Discretize with Tustin's method, evaluate for $T = 0.004$, $\tau = 0.05$, and write difference eq.

$$G(z) = G(s)\big|_{s=\frac{T}{2}\left(\frac{z+1}{z-1}\right)} = \frac{z+1}{\left(\frac{2\tau}{T}+1\right)z+\left(1-\frac{2\tau}{T}\right)} = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_{n-1} z^1 + b_n z^0}{a_0 z^n + a_1 z^{n-1} + \cdots + a_{n-1} z^1 + a_n z^0}$$

$n = 1$
$b_0 = 1$
$b_1 = 1$
$a_0 = \left(\frac{2\tau}{T} + 1\right) = 26.0$
$a_1 = \left(1 - \frac{2\tau}{T}\right) = -24.0$

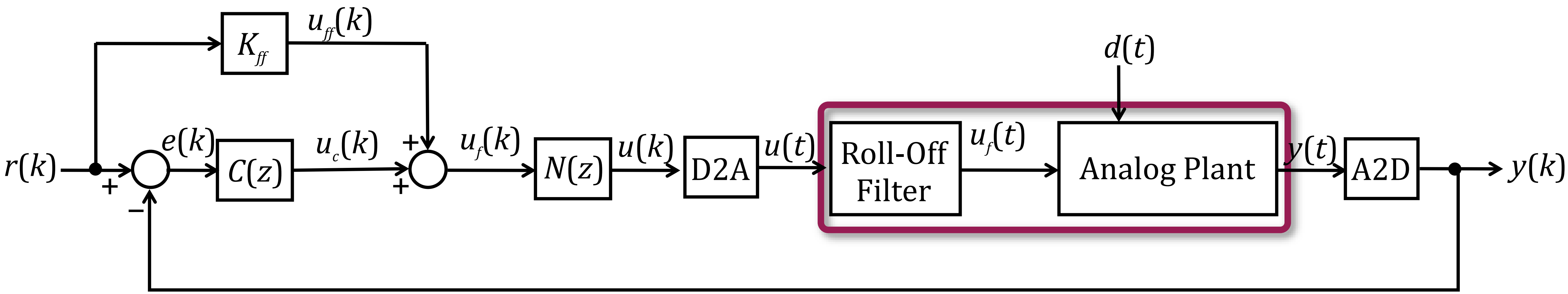
$$y(k) = \frac{1}{a_0}[b_0 x(k) + b_1 x(k-1) - a_1 y(k-1)]$$

Python code (where `x` and `y` are lists of previous values ordered from newest to oldest):

```
new_y = 1/a0 * (b0 * x[0] + b1 * x[1] - a1 * y[1])
      = 1/den[0] * (dot(num, x) - dot(den[1:], y))
```

**Remarks:**
1) Consider writing your DE's in a general form that can accommodate a transfer function of arbitrary order
2) In Lab 4, we used a *continuous-time* 1st low-pass filter because a discrete-time filter would not be able to attenuate (roll-off) high frequency components above the Nyquist frequency.

## Configuration




The configuration for the closed-loop system for this project is shown above. The D2A block is a zero-order-hold-type digital-to-analog converter. The roll-off filter is to filter the $u(t)$ signal to lessen the excitation of the analog plant's high frequency vibration mode. The transfer function for the roll-off filter is to be

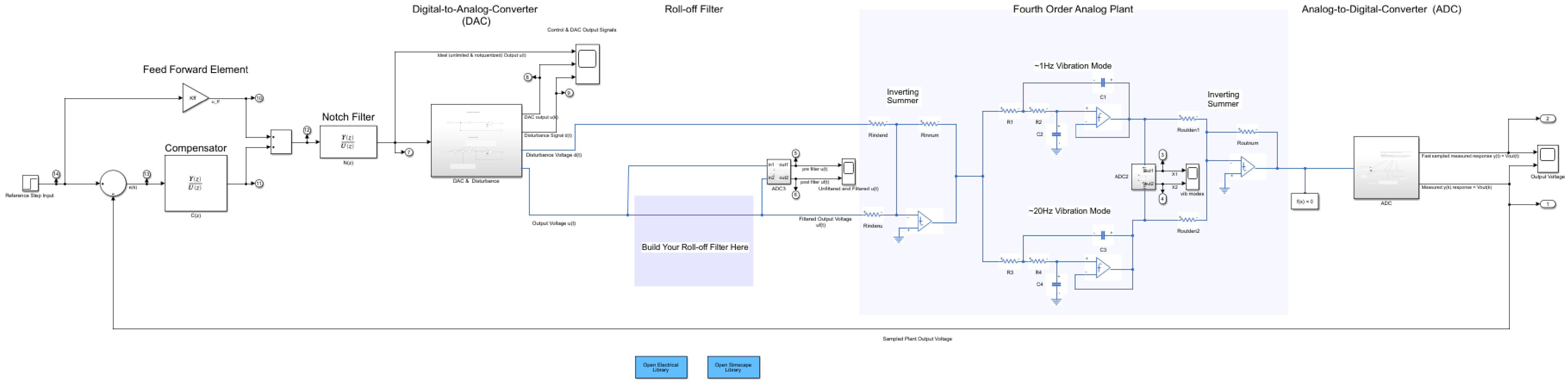
$$\frac{U_f(s)}{U(s)} = \frac{\omega_{rolloff}}{s + \omega_{rolloff}}$$

The break frequency for the rolloff filter, $\omega_{rolloff}$, is targeted to be 20.232 rad/sec, which is the damped natural frequency of the zeros of the nominal analog plant's $u_f - \text{to} - y$ transfer function.

$N(z)$ to be a second-order notch filter with zeros that cancel the low-frequency vibration mode poles of the analog plant and poles that have more desirable dynamics.

# OPTIONAL: aem581_lab4_LoopShaping_5thO_class.slx

Matlab Simulink Simscape version of this lab available on canvas. Simulink and simscape simulation allows you to probe internal states of the plant inaccessible on the breadboard. The ability to simulate multi-domain problems using Simscape is potentially useful.



# Suggestion

To better understand a likely source of differences between your virtual hardware-in-the-loop closed-loop system's $y(t)$ response to a unit step $r(k)$ input and that expected based upon your model-based design work:

Modify the provided MATLAB or Python software to obtain plots of the simulated $y(t)$ response of your closed-loop system to a unit step $r(k)$ input when the natural frequency of the modeled analog plant's low-frequency vibration mode is 10% higher and, separately, 10% lower, than that assumed for your model-based design work.

# Grading

Based upon the extent that what you submit demonstrates understanding of the subject matter of this course.

*Approaches that use a bona fide loop shaping approach to controller design, rather than some other means, will receive a better score.* A solution based upon trial and error ad infinitum does little to demonstrate understanding of the subject matter of this course.