SIGNALS AND SYSTEMS I
Computer Assignment 6

Discrete systems can be designed, analyzed and simulated in MATLAB using the signal processing and controls toolboxes.

Signal Processing Toolbox

In MATLAB's signal processing toolbox, the transfer function is the principal system representation. For single input, single output (SISO) systems, MATLAB's signal processing toolbox assumes that the system's transfer function or system function, $H(z)$, has the form

$$H(z) = \frac{b(1) + b(2)z^{-1} + \cdots + b(M+1)z^{-M}}{a(1) + a(2)z^{-1} + \cdots + a(N+1)z^{-N}}$$

In MATLAB, the row vector, **b**, stores the numerator coefficients, $b(1), b(2), \ldots, b(M), b(M+1)$, and the row vector, **a**, stores the denominator coefficients, $a(1), a(2), \ldots, a(N), a(N+1)$. Thus, a system function is defined by two row vectors, one row vector for the numerator and one for the denominator.

If the system function has the from

$$H(z) = \frac{c(1)z^{N} + c(2)z^{N-1} + \cdots + C(N+1)}{d(1)z^{N} + d(2)z^{N-1} + \cdots + d(N+1)}$$

the system function can be evaluated for particular values of z using the *polyval* function. The *eqtflength* function can be used to equalize the order of the numerator and denominator polynomials so that a system function that has the form of (1) can be described by a system function that has the form of (2) . For example,

```
[c,d] = eqtflength(b,a)
H = polyval(c,z) ./ polyval(d,z)
```

evaluates the system function defined by the vectors, **b** and **a**, at the values in the matrix, **z**. The system function can be evaluated around the unit circle using the *freqz* function. For example,

```
H = freqz(b,a,w)
```

evaluates the system defined by the vectors, **b** and **a**, at the values in the vector, e^{jw}, where **w** is a real vector. Typically the system function, $H(z)$, is complex valued. The *abs* and *angle* functions can be used to generate the magnitude and phase of $H(z)$, respectively.

A system function that can be written in the form of (1) can also be written in the factored or zero pole gain form

$$H(s) = k\frac{[1 - q(1)z^{-1}][1 - q(2)z^{-1}]\ldots[1 - q(M)z^{-1}]}{[1 - p(1)z^{-1}][1 - p(2)z^{-1}]\ldots[1 - p(N)z^{-1}]}$$

where k is the system's gain, $q(1),q(2),\ldots,q(M)$ are the system's zeros and $p(1),p(2),\ldots,p(N)$ are the system's poles. By convention, MATLAB stores polynomial coefficients in row vectors and polynomial roots in column vectors. Therefore, functions, such as *zplane*, generate different results depending on whether the function's input is a row or column vector. For example,

```
b = [1  1.5  -1];
a = [1  0.25  -0.125];
zplane(b,a)
```

generates a pole zero plot of the system function

$$H(z) = \frac{1 + 1.5z^{-1} - z^{-2}}{1 + 0.25z^{-1} - 0.125z^{-2}}$$

where the zeros are at -2 and 0.5 and the poles are at -0.5 and 0.125. On the other hand,

```
b = [-2;0.5];
a = [-0.5;0.125];
zplane(b,a)
```

generates an identical plot. MATLAB's *poly* and *roots* functions can be used to convert between polynomial and root representations. For example,

```
roots([1  1.5  -1])
```

generates the column vector $[-2\ 0.5]^T$, and

```
poly([-2;0.5])
```

generates the row vector, $[1\ 1.5\ -1]$, that represents the polynomial, $1+1.5z^{-1} - z^{-2}$.

A system function that can be written in the form of (1) also has a corresponding partial fraction expansion or residue representation of the form

$$H(z) = \frac{r(1)}{1 - p(1)z^{-1}} + \cdots + \frac{r(N)}{1 - p(N)z^{-1}} + k(1) + k(2)z^{-1} + \cdots + k(M - N + 1)z^{-(M-N)}$$

if multiple roots do not exist. MATLAB's *residuez* function can be used to determine a system function's residue representation. From this representation, the system's impulse response can be calculated. For example,

```
b = [1  2];
```

```
a = [1  0.25  -0.125];
[R,P,K] = residuez(b,a)
n = 0:50;
h = [P(1).^n P(2).^n]*R;
```

generates the impulse response of the system function,

$$H(z) = \frac{1 + 2z^{-1}}{1 + 0.25z^{-1} - 0.125z^{-2}}$$

for $0 \leq n \leq 50$.

For linear time invariant systems,

$$Y(z) = H(z)X(z)$$

where $Y(z)$ is the z transform of the system's output, $H(z)$ is the system's transfer function and $X(z)$ is the z transform of system's input. To generate $Y(z)$, the numerator polynomials of $H(z)$ and $X(z)$ must be multiplied together and the denominator polynomials of $H(z)$ and $X(z)$ must be multiplied together. In Matlab, the *conv* function can be used to multiply polynomials.

Exercises

For Exercises 1 - 8, use the discrete system described by the difference equation,

$$y(n) - 2r\cos(\omega_0)y(n-1) + r^2 y(n-2) = x(n) - r\cos(\omega_0)x(n-1)$$

where $x(n)$ is the system's input, $y(n)$ is the system's output,

$$Radius\ r = 0.9 \text{ and } \omega_0 = \frac{\pi}{4} \text{ rad/sample.}$$

1. Via hand calculations, determine the system's transfer function, $H(z)$. Where are the poles and zeros? Hint: for the poles, it may be easier to use Euler's equation. Sketch and label the poles and zeros on a Z-Plane. What angle (in radians) does the poles make with respect to the real axis?

2. Calculate $H(z)$ for $-2 \le Re\{z\} \le 2$ and $-2 \le Im\{z\} \le 2$. Using MATLAB, plot $H(z)$ in decibels (dB): $\text{Power} = 20 \cdot \log_{10}|H(s)|$. Use the functions *meshc*, *title*, *xlabel*, *ylabel*, *zlabel*, *title*, and ***log10***.

3. Using the *freqz* function, calculate the frequency response, $H(e^{j\omega})$,for$-2\pi \le \omega \le 2\pi$. Plot $|H(e^{j\omega})|$, in decibels (dB) and the phase of $H(e^{j\omega})$ in degrees using the *plot*, *title*, *xlabel*, *ylabel*, *subplot* and ***log10*** functions. (You should generate 2 plots on 1 page.)
4. Using the *roots* function, determine the system function's poles and zeros. Using these poles and zeros, use the *poly* function to generate the system function. Do the poles and zeros match your hand calculations from Exercise 1? If not, check your work.

5. Using the *zplane* function, generate a pole zero plot of $H(z)$. Does this match with your hand calculations from Ex. 1?

6. Using the *residuez* function, generate the partial fraction expansion representation of $H(z)$. Using this representation, generate the system's impulse response, $h(n)$, for $0 \le n \le 50$. Plot $h(n)$ for $0 \le n \le 50$ using the *stem*, *title*, *xlabel*, and *ylabel* functions.

7. Using the *conv* function, generate $Y(z)$ when $x(n) = u(n)$. Using the *residuez* function, generate the partial fraction expansion representation of $Y(z)$. Using this representation, generate the system's step response, $y(n)$, for $0 \le n \le 50$. Plot $y(n)$ for $0 \le n \le 50$ using the *stem*, *title*, *xlabel*, and *ylabel* functions.

   What value does the system settle to?

Controls Toolbox

Although MATLAB's controls toolbox allows discrete systems to be modeled as transfer functions in rational and factored representations, the state space representation is the principal system representation in MATLAB's controls toolbox. For example, many of MATLAB 's functions which accept rational, factored and partial fraction expansion system representations as inputs use a state space representation of the system internally. Regardless of the system representation, MATLAB's controls toolbox requires that the model be defined.

MATLAB's controls toolbox assumes that a system's transfer function or system function, $H(z)$, has the form

$$H(z) = \frac{b(1)z^N + b(2)z^{N-1} + \cdots + b(N+1)}{a(1)z^N + a(2)z^{N-1} + \cdots + a(N+1)}$$

The *eqtflength* function can be used to convert a transfer function in the signal processing toolbox form to a transfer function in the controls toolbox form. A discrete rational transfer function can then be defined as

```
sys_tf = tf(b,a,-1)
```

A factored system representation can be defined as

```
sys_zpk = zpk(z,p,k,-1)
```

and a state space model of the form,

$$\begin{aligned}\mathbf{q}(n+1) &= \mathbf{A}\mathbf{q}(n) + \mathbf{B}\mathbf{x}(n)\\ \mathbf{y}(n) &= \mathbf{C}\mathbf{q}(n) + \mathbf{D}\mathbf{x}(n)\end{aligned}$$

can be defined as

```
sys = ss(A,B,C,D,-1)
```

After the system has been defined, the *freqresp*, *pole*, *zero*, *dcgain*, *impulse*, and *step* functions can be used to determine the system's frequency response, poles, zeros, gain, impulse response and step response, respectively. Systems can also be combined using addition and multiplication. Addition performs a parallel interconnection, and multiplication performs a series interconnection. For example,

```
sys1 = tf([1 0],[1 0.5],-1);
sys2 = tf([1 0],[1 0.25],-1);
sys3 = sys1*sys2
```

Generate the system function,

$$H_3(z) = H_1(z)H_2(z) = \frac{1}{1 + \frac{1}{2}z^{-1}} \cdot \frac{1}{1 + \frac{1}{4}z^{-1}} = \frac{1}{1 + \frac{3}{4}z^{-1} + \frac{1}{8}z^{-2}}.$$

Exercises

For Exercises 8 - 15, again use the discrete system described by the differential equation,

$$y(n) - 2r\cos(\omega_0)y(n-1) + r^2 y(n-2) = x(n) - r\cos(\omega_0)x(n-1)$$

where $x(n)$ is the system's input, $y(n)$ is the system's output,

$$r = 0.9 \text{ and } \omega_0 = \pi/4 \text{ rad/sample.}$$

8. Draw (and label) a Direct Form II block diagram of the system, and using this block diagram, write out the state space equations of the system **in matrix form**.

9. Using the *tf2ss* and *eqtflength* functions, generate a state space model from the transfer function model that you developed in Exercise 1. Compare the result to your state space model in Exercise 8.

10. Using the *freqresp* function, calculate the frequency response, $H\left(e^{j\omega}\right)$,, for $-2\pi \le \omega \le 2\pi$. Plot $|H\left(e^{j\omega}\right)|$, in dB and the phase of $H\left(e^{j\omega}\right)$,in degrees using the *plot*, *title*, *xlabel*, *ylabel*, *subplot* and ***log10*** functions. (You should generate 2 plots on 1 page.)

11. Using the *zero*, *pole* and *dcgain* functions, determine the system's poles, zeros and gain.

12. Using the *pzmap* function, generate a pole zero plot of $H(z)$.

13. Using the *impulse* function, generate the system's impulse response, $h(n)$ for $0 \le n \le 50$. Plot $h(n)$ for $0 \le n \le 50$ using the *stem*, *title*, *xlabel*, and *ylabel* functions.

14. Using the *step* function, generate the system's step response, $s(n)$ for $0 \le n \le 50$. Plot $s(n)$ for $0 \le n \le 50$ using the *stem*, *title*, *xlabel*, and *ylabel* functions.

15. Generate a system model for the step function and multiply it with your system model. Using the resulting model and the *impulse* function, generate the system's step response, $s(n)$ for $0 \le n \le 50$. Plot $s(n)$ for $0 \le n \le 50$ using the *stem*, *title*, *xlabel*, and *ylabel* functions.