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

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

In MATLAB, the row vector, **b**, stores the numerator coefficients, $b(1),b(2),\cdots,b(M),b(M+1)$, and the row vector, **a**, stores the denominator coefficients, $a(1),a(2),\cdots,a(N),a(N+1)$. 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

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

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, $e^{j\mathbf{w}}$, 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

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

where k is the system's gain, $q(1),q(2),\cdots,q(M)$ are the system's zeros and $p(1),p(2),\cdots,p(N)$ 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

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

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 $\begin{bmatrix}-2 & 0.5\end{bmatrix}^T$, and

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

generates the row vector, $\begin{bmatrix}1 & 1.5 & -1\end{bmatrix}$, 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

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

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 $0 \le n \le 50$.

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,

$$a = 0.9 \quad \text{and} \quad \omega_0 = \pi / 7 \text{ rad/sample.}$$

1. Determine the system's transfer function, $H(z)$.

2. Calculate $H(z)$ for $-2 \le \text{Re}\{z\} \le 2$ and $-2 \le \text{Im}\{z\} \le 2$. Plot $|H(z)|$ is dB $\left(20 * \log_{10}|H(z)|\right)$ using the *meshc*, *title*, *xlabel*, *ylabel* and *zlabel* functions.

3. Using the *freqz* function, calculate the frequency response, $H(e^{j\omega})$, for $-2\pi \le \omega \le 2\pi$. Plot $\left|H(e^{j\omega})\right|$ is dB and the phase of $H(e^{j\omega})$ in degrees using the *plot*, *title*, *xlabel*, *ylabel* and *subplot* 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.

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

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.