SIGNALS AND SYSTEMS I
Computer Assignment 4

Analog 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 assumes that the system's transfer function or system function, $H(s)$, has the form

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

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.

The system function can be evaluated for particular values of s using the *polyval* function. For example,

```
H = polyval(b,s) ./ polyval(a,s)
```

evaluates the system function defined by the vectors, **b** and **a**, at the values in the matrix, **s**. The system function can be evaluated along the imaginary axis using the *freqs* function. For example,

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

evaluates the system function defined by the vectors, **b** and **a**, at the imaginary values in the vector, **w**. By convention, the values in **w** are real; however, the function *freqs* evaluates the system function at the values *j***w**. Typically the system function, $H(s)$, is complex valued. The *abs* and *angle* functions can be used to generate the magnitude and phase of $H(s)$, 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

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

where k is the system's gain, $z(1), z(2), \cdots, z(M)$ 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 3 2];
```

```
a = [1 7 12];
zplane(b,a)
```

generates a pole zero plot of the system function

$$H(s) = \frac{s^2 + 3s + 2}{s^2 + 7s + 12}$$

where the zeros are at -1 and -2 and the poles are at -3 and -4. On the other hand,

```
b = [-1;-2];
a = [-3;-4];
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 3 2])
```

generates the column vector $[-2 \quad -1]^T$, and

```
poly([-2;-1])
```

generates the row vector, $[1 \quad 3 \quad 2]$, that represents the polynomial, $s^2 + 3s + 2$.

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

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

if multiple roots do not exist. MATLAB's *residue* 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 3];
a = [1 7 12];
[R,P,K] = residue(b,a)
t = 0:0.025:5;
h = R.' * exp(kron(P,t));
```

generates the impulse response of the system function

$$H(s) = \frac{s + 3}{s^2 + 7s + 12}$$

for $0 \le t \le 5$.

For linear time invariant systems,

$$Y(s) = H(s)X(s)$$

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

Exercises

For Exercises 1 - 8, use the analog system described by the differential equation,

$$\frac{d^2y(t)}{dt} + 2a\frac{dy(t)}{dt} + \left(a^2 + w_0^2\right)y(t) = \frac{dx(t)}{dt} + ax(t)$$

where $x(t)$ is the system's input, $y(t)$ is the system's output and

$$a = 1.5 \text{ and } \omega_0 = 7 \text{ rad/sec.}$$

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

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

3. Using the *freqs* function, calculate the frequency response, $H(j\omega)$, for $-10 \le \omega \le 10$. Plot $|H(j\omega)|$ is dB and the phase of $H(j\omega)$ 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 *tf2zp* function, generate the factored form of the system function. Using the *zp2tf* function, convert the factored form of the system function back into its original form.

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

7. Using the *residue* function, generate the partial fraction expansion representation of $H(s)$. Using this representation, generate the system's impulse response, $h(t)$, for $0 \le t \le 5$. Plot $h(t)$ for $0 \le t \le 5$ using the *plot*, *title*, *xlabel*, and *ylabel* functions.

8. Using the *conv* function, generate $Y(s)$ when $x(t) = u(t)$. Using the *residue* function, generate the partial fraction expansion representation of $Y(s)$. Using this representation, generate the system's step response, $y(t)$, for $0 \le t \le 5$. Plot $y(t)$ for $0 \le t \le 5$ using the *plot*, *title*, *xlabel*, and *ylabel* functions.