Lab Mathematical Modeling 4

Joël M.H. Karel

January 7, 2021

Exercise 1

Consider the following state-space system in continuous time:

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

met

$$\begin{aligned} A &= \begin{pmatrix} 4 & 1 & 0 \\ -7 & -1 & -1 \\ 119 & 32 & -8 \end{pmatrix} \\ B &= \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \\ C &= \begin{pmatrix} 227 & 55 & -13 \end{pmatrix} \\ D &= -2 \end{aligned}$$

a)

Enter this system into Matlab. Call it `system1` and compute its associated transfer function $H(s)$, using the *Fadeev* algorithm. (You already should have that from lab 3, hence this is one minute of work)

b)

Compute the poles of `system1` and show that the system is asymptotically stable.

c)

Choose the initial state $x(0) = 0$ and the input function $u(t) = \sin(4t)$. Simulate the corresponding output $y(t)$ using the routine `lsim`. Display the functions $u(t)$ and $y(t)$ jointly in a single plot. Note that for large values of t it holds approximately that $y(t) \approx K \sin(4t + \varphi)$. Determine these values for K and φ graphically, using the plot.

d)

Choose $s = 4i$ and compute the complex number $H(4i)$. Then rewrite $H(4i)$ into polar coordinates as $H(4i) = \tilde{K}e^{i\tilde{\varphi}}$. (Matlab can help you with this!) Now compare the values of $\tilde{K}$ and $\tilde{\varphi}$ to the values of K and φ previously obtained under c).

Exercise 2 - the effect of poles and zeros

We're now taking a closer look at how the location of the poles and zeros affect the frequency domain properties of a linear system.

One thing that you should first be aware of is that the Fourier coefficients of real valued signals are symmetric. This means that for a real signal it holds for the Fourier coefficients that $F(\omega) = F(-\omega)^*$ in the Fourier domain, hence $F(i\omega) = F(-i\omega)^*$ in the Laplace domain. (remember that Fourier is $s = i\omega$, hence limits to the complex axis.) The consequence of this is that we have to ensure that the frequency response our filters are symmetric in the real axis.

Next, assume that we have a continuous-time linear system with a transfer function of the form

$$H(s) = \frac{(s - z_1)(s - z_2)\dots(s - z_n)}{(s - p_1)(s - p_2)\dots(s - p_n)}$$

and thus a frequency response of

$$H(\omega) = H(i\omega) = \frac{(i\omega - z_1)(i\omega - z_2)\dots(i\omega - z_n)}{(i\omega - p_1)(i\omega - p_2)\dots(i\omega - p_n)}.$$

This means that if we have a pole at $p_k = i\omega$, then at frequency ω, we are dividing by zero, hence signals of frequency ω will blow up. This is not a desirable effect and we would like to have a stable filter for obvious reasons, hence we could place the pole just left of that location in the complex plane at $p_k = -\epsilon + i\omega$ such that one gets division by a small number and the corresponding frequencies are amplified.

If one has a zero at $z_k = i\omega$, this means that signals of frequency ω are filtered out. Both poles and zeros have a radius effect, meaning that frequencies close to the zero are attenuated.

a)

We first construct a test signal. It is convenient if all frequencies are present in the test signal. That's why we start with a signal consisting of white Gaussian noise. However we also add a frequency to that signal to ensure that we can also add some structure. We'll add a 15Hz frequency. Note that so far we haven't defined any units for frequency, and we have to multiply by 2π to work here in Hz. The signal is real, hence the Fourier spectrum should be symmetric. You can do all this with the following code:

```
fs=100; % sampling frequency
t=0:1/fs:5; % time vector
s1=randn(size(t)) + sin(2*pi*15*t); %signal
S1=fft(s1); % Fourier transform
omega=linspace(0,fs,length(t)+1);omega(end)=[]; % generate vector of frequencies
figure;plot(omega,abs(S1)); % plot Fourier magnitude spectrum
```

Notice the symmetry in the spectrum in 50Hz. Since we sampled at 100Hz, the frequencies from 50Hz to 100Hz are equal to the frequencies of -50Hz to 0Hz.

b)

Next create a system `lpf` with transfer function

$$H(s) = \frac{2\pi 50}{s + 2\pi 50}.$$

The coefficient in the numerator is just a normalisation. Find the poles and zeros to see that there is a single stable negative pole. The frequency response can be found by the Bode magnitude plots. The latter are typically in dB, which might not be very intuitive. For the plots the Matlab function `ltiview` can help you. For Octave to my knowledge such convenient function is not available.

This system only has a single pole that in terms of the imaginary axis is closest to the 0 frequency, hence it acts as a low-pass filter, with a very crappy attenuation to higher frequencies.

Check with the sinusoidal fidelity principle how frequencies of 0 and 15Hz are affected in terms of magnitude. `poly` and `polyval` can help you here.

c)

Simulate the system `lpf` with the data generated under a) and plot the frequency content of the input signal and output signal in a single plot to verify that we have a low-pass filter.

d)

Next create a system `lpf2` with transfer function that has a *double* pole at the same location as the system `lpf`, two zeros at $\pm 40 \cdot 2\pi i$ and of which the transfer function is scaled such that $H(0) = 1$. Repeat b) and c) for this system and find the frequency response at 40Hz. Observe that the zeros attenuate frequencies, but the poles amplify them and the poles and zeros interact. The stopband now got more pronounced, however in the next step we'll look at a more extreme case.

e)

We are now going to create a notch filter. In the signal we inserted a 15Hz sinusoid that we will now filter out. To filter it out we have to put a zero at $s = 15 \cdot 2 \cdot \pi i$ and since we want real valued output at $s = -15 \cdot 2 \cdot \pi i$. To obtain a proper system, we need at least two poles. If we would place them on the real axis we would get a low-pass filter, which we do not want. We want the poles to cancel the effect of the zeros for frequencies other than 15Hz. To do that we place them just off the imaginary axis, close to the zeros at $s = \pm 15 \cdot 2 \cdot \pi i - 10$. Construct system `nf1` and Repeat b) and c) for this system. Do this as well for system `nf2` that has its poles at $s = \pm 15 \cdot 2 \cdot \pi i - 1$. Observe that the notch then becomes narrower. If you want to see the cost of this look at the impulse responses. The frequency domain properties improve at the cost of the time domain properties as the Heisenberg uncertainty principle.

f)

As final step, we want to see the absolute values of the frequency responses in a single plot such that we can better see what we did. Just run the following code snippet:

```
f=linspace(0,50,100);
[b1,a1] = tfdata(lpf);
Hiw1=polyval(b1{1},f*2*pi*1i)./polyval(a1{1},f*2*pi*1i);
[b2,a2] = tfdata(lpf2);
Hiw2=polyval(b2{1},f*2*pi*1i)./polyval(a2{1},f*2*pi*1i);
[b3,a3] = tfdata(nf1);
Hiw3=polyval(b3{1},f*2*pi*1i)./polyval(a3{1},f*2*pi*1i);
[b4,a4] = tfdata(nf2);
Hiw4=polyval(b4{1},f*2*pi*1i)./polyval(a4{1},f*2*pi*1i);
figure; plot(f,abs(Hiw1),f,abs(Hiw2),f,abs(Hiw3),f,abs(Hiw4));
legend({'lpf','lpf2','nf1','nf2'})
```