
Homework 12

Due: Monday, December 4, 2006, at 5pm
Homework 12 GSI: June Wang

Please note that the due date is Monday (12/4), not Thursday. The due date is set such that you can get the solution early to study for the final.

Reading OWN Chapter 10.

Practice Problems (*Suggestions.*) OWN 10.1, 10.2, 10.3, 10.4, 10.12, 10.16, 10.17, 10.18, 10.20.

Problem 1 (*z-Transform Basics*)

(a) OWN 10.21 (b), (h)

(b) OWN 10.22 (b), (d)

Problem 2 (*Inverse z-Transform*)

(a) OWN 10.23, the first and second $X(z)$. No need to do the Taylor series method.

(b) OWN 10.26 (a), (b), (c).

Problem 3 (*Properties of the z-Transform*)

OWN 10.44 (a), (b), (c). In each case, also determine the region of convergence (in terms of the region of convergence R of the original signal $x[n]$).

Remark: These properties are the foundations of wavelets.

Problem 4 (*Properties of the z-Transform: minimum-phase system.*)

OWN 10.58

Problem 5 (*Discrete-time LTI system analysis.*)

A causal LTI system is described by the following system diagram, where b is a real number:

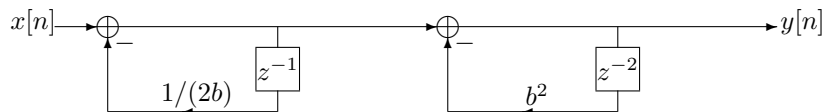


Figure 1: A causal discrete-time system.

Find the pole-zero plot and the range of b such that the system is stable.

Problem 6 (*Discrete-time LTI system.*)

OWN Problem 10.34

Problem 7 (*Discrete-time LTI system analysis.*)

OWN Problem 10.47

Problem 8 (*Unilateral Z-Transform.*)

OWN Problem 10.42 (b) and (c)

Problem 9 (*Fibonacci Numbers.*)

The original problem that Fibonacci investigated (in the year 1202) was about how fast rabbits could breed in ideal circumstances.¹ My math formula book describes the Fibonacci numbers as follows:

“Denote the n th Fibonacci number by F_n . These numbers are described by the following formulas:

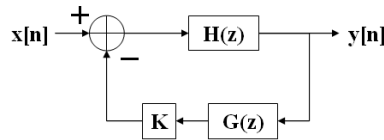
$$F_1 = 1, F_2 = 1, \quad \text{and} \quad F_{n+2} = F_{n+1} + F_n, n \geq 1. \quad (1)$$

(a) Find an explicit formula for F_n (in terms of n). *Hint:* Note that the description of the Fibonacci numbers is a difference equation, and use the unilateral z -transform.

(b) Use your formula from Part (a) to determine F_{21} .

Problem 10 (*Root locus of discrete-time systems.*)

Consider the following system



where $H(z) = \frac{1}{z+2}$, $G(z) = z^{-1}$ and K is the adjustable parameter. Plot the root locus of the overall system and specify for what range of K is the system stable.

Note: this problem replaces OWN Problem 11.25 (a).

Problem 11 (*Pole/Zero plots.*)

Match the pole/zero plots (a)-(e) with the corresponding magnitude responses (1)-(5). Provide a brief justification for each case. (Example: “must have 2 symmetric peaks, therefore can only be plot (x)”.)

(Optional) **Problem 12** (*Image Processing*)

In this homework problem, we explore some of the essential properties of the two-dimensional Fourier transform, defined simply as

$$X(\omega_1, \omega_2) = \int \int x(t_1, t_2) e^{-j\omega_1 t_1} e^{-j\omega_2 t_2} dt_1 dt_2 \quad (2)$$

¹You can learn more about the history of Fibonacci at <http://www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/fibnat.html#Rabbits>

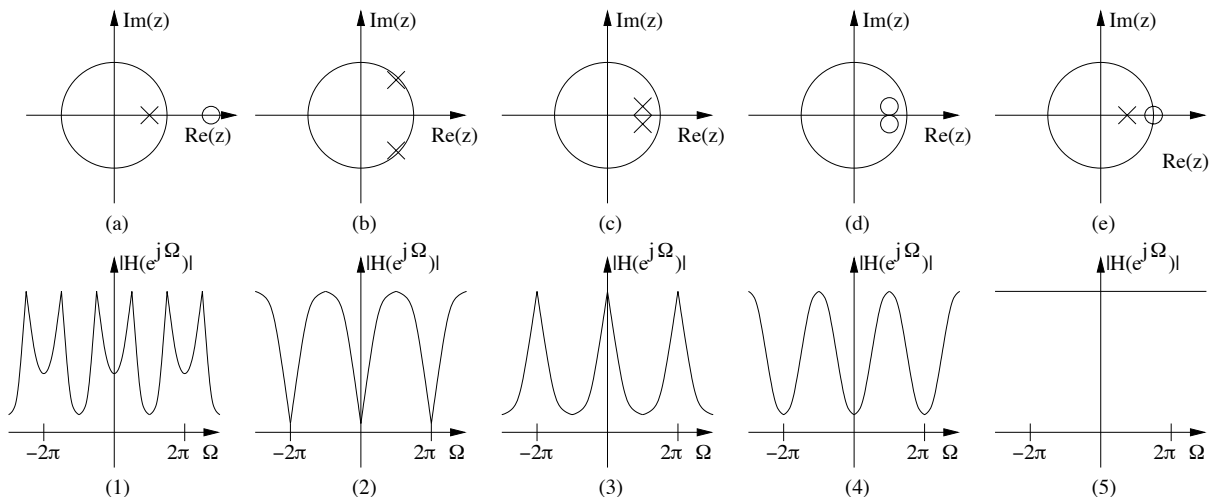


Figure 2: Matching of pole/zero plots and frequency responses.

(a) Consider the signal

$$x(t_1, t_2) = \cos(t_1/3) \cos(t_2/5). \quad (3)$$

Use matlab to get a sense for what the signal $x(t_1, t_2)$ looks like:

```
t1 = [-20:0.1:20]; % If your computer is slow, you may want to reduce the resolution.
t2 = [-20:0.1:20];
mesh(t1, t2, cos(t1/3)*cos(t2/5));
```

Turn in a printout of this three-dimensional plot of $x(t_1, t_2)$.

Then, calculate (by hand) $X(\omega_1, \omega_2)$ and sketch (by hand) a two-dimensional figure with the two axes corresponding to ω_1 and ω_2 and appropriate symbols representing the value of $X(\omega_1, \omega_2)$. *Hint:* First determine $X(\omega_1, \omega_2)$. Looking at the formula, it should become clear at once how to sketch it... ;-)

(b) Repeat (a) for the signal

$$x(t_1, t_2) = \text{sinc}(t_1/3) \text{sinc}(t_2/5). \quad (4)$$

Again, sketch (by hand) $X(\omega_1, \omega_2)$. In the two-dimensional plane with axes ω_1 and ω_2 , where are the low frequencies?

(c) One way to do image compression is to take the (two-dimensional) Fourier transform of the image, and then only retain the low frequencies as human eyes are most sensitive to low frequency content of an image.

To explore this idea, download the jpeg file brain.jpg from the class web page. In matlab, try the following:

```
I = imread('brain.jpg');
colormap('gray');
image(I);
```

Write the following m-file, called lowpass.m:

```
function Ic = lowpass(I, k);
if sum(size(I)==[256 256]) == 2,
if k > 128, k = 128; end;
if k < 1, k = 1; end;
Ifourier = fftshift(fft2(double(I)));
Ifouriercompressed = zeros(size(I));
Ifouriercompressed(128-k+1:128+k, 128-k+1:128+k) = Ifourier(128-k+1:128+k, 128-k+1:128+k);
```

```

Ic = ifft2(fftshift(Ifouriercompressed));
Ic = real(Ic);
else
Ic = 0; 'Code works only for image size 256 by 256 pixels!'
end;

```

Try the program for different values of k , and understand what the program is doing. Turn in the following:

- Plots for three representative values of k (ie high, medium, and low compression). Describe (in words) what the compression algorithm does to the image. *Hint:* It's pretty cool to check out, one after the other, $k = 1, 2, 3, 4, 5$, etc....
- In the two-dimensional Fourier plane (corresponding to the two-dimensional Fourier transform of the image), explain what the program does in order to compress the image. Why will the compressed image yield a smaller file?

(d) A frequent task in image processing is to remove noise from an image. Noise may be due to many different sources, such as dust on the optical lens or quantization in digital photography. In some cases, this noise is nearly independent from pixel to pixel, which means that it is “fast-varying”, that is, of high frequency. Based on this insight, people have tried to get rid of the noise simply by low-pass filtering the image.

To add noise to the image, proceed (in matlab) as follows:

```

I = imread('brain.jpg');
colormap('gray');
image(I);
In = uint8( double(I) + 10*(2*(rand(size(I))-0.5)) );
figure(2);
image(In)

```

You should clearly see the noise in the image. To remove the noise, use the tunable low-pass filter. Try for example:

```
image(lowpass(In, 90))
```

Of course, 90 may not be the best cut-off frequency.

- Turn in a printout of the image with the noise added in and a printout of the denoised image, where you choose the cut-off frequency of the tunable low-pass filter in a good way. What is your choice?
- (Essay question) Briefly describe the advantages and disadvantages of making the cut-off frequency larger and smaller, respectively (solely based on your observations from playing with the matlab code; no analytical justification is required).

Remark: The matlab command `rand` returns a matrix filled with random numbers between zero and one. For this homework exercise, the noise added to the pixels is a random number between -10 and 10 . The correct noise model, of course, depends on the nature of the noise source.