COT 5615 Math for Intelligent Systems Fall 2021 Homework #4

UFID: 96703101
Name: *Vyom Pathak*
Instructor: Professor Kejun Huang
Due Date: October 01, 2021

Problem 7.6

Rows of incidence matrix

Solution

According to the definition of incidence matrix, the sum of the rows will be equal to zero, $1^T A = 0$, and thus the rows of the incidence matrix are always linearly dependent.

Problem 7.8

Flow conservation with sources

Solution

Here, we can prove $1^T s = 0$ by the following derivation:

$$\begin{aligned} Ax + s &= 0 \\ 1^T(Ax + s) &= 1^T 0 \\ (1^T A)x + 1^T s &= 0 \\ 0 + 1^T s &= 0 \quad (\because 1^T A = 0) \\ \therefore 1^T s &= 0 \end{aligned}$$

Problem 7.15

Channel equalization

Solution

a Here, using the associative property of convolution we get the following result:

$$\begin{aligned} z &= h * y \\ z &= h * (c * u) \\ z &= (h * c) * u \\ z &\approx e_1 * u \quad (\because given) \\ z &\approx (u_1, u_2, \ldots, u_m, 0, 0, \ldots, 0) \end{aligned}$$

b Following is the code for calculation and plotting u, y, z with sizes 50, 52, and 59 respectively.

```
using LinearAlgebra
using DSP
using Random
using Plots
plotly()
```

```
rng = MersenneTwister(1234);
u = shuffle(rng, Vector(1:50))
X = -1;
Y = +1;
C = findall(u.<=25);
D = findall(u.>25);
u[C] .= X;
u[D] .= Y;
c = [1,0.7,-0.3];
h = [0.9, -0.5, 0.5, -0.4, 0.3, -0.3, 0.2, -0.1];
y = conv(c,u);
z = conv(h,y);
plot(1:length(u),u)
plot(1:length(y),y)
plot(1:length(z),z)
```

Following are the generated graphs:

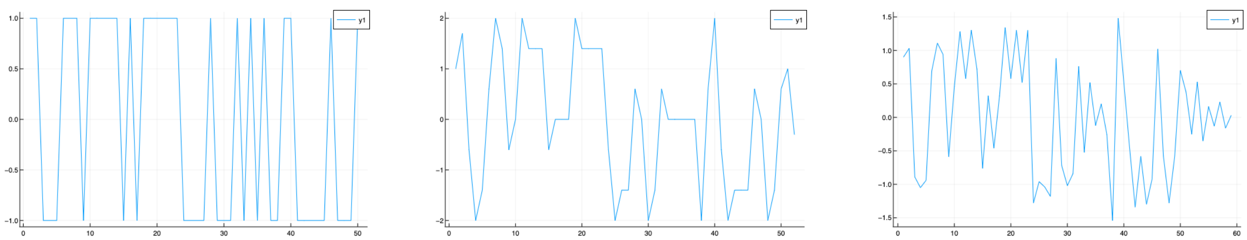

Figure 1: signal v/s t for u, y and z respectively.

Problem A7.1

Equalization in communication

Solution

```
include("channel_equalization_data.jl")
hc = conv(h,c);
y = conv(c,s);
y_tilda = conv(h,y);

plot(1:length(c), c)
plot(1:length(h), h)
plot(1:length(hc), hc)

plot(1:length(s[1:100]), s[1:100])
plot(1:length(y[1:100]), y[1:100])
plot(1:length(y_tilda[1:100]), y_tilda[1:100])

s_cap = 1*(y .> 0.5);
s_cap_eq = 1*(y_tilda .> 0.5);
BER = sum( broadcast(abs, s_cap[1:1000]-s))/1000;
BER_eq = sum( broadcast(abs, s_cap_eq[1:1000]-s))/1000;
```

```
println("Bit Error rate for non-equilized transmission signal: ",BER) #0.113
println("Bit Error rate for equilized transmission signal: ",BER_eq) #0.0
```

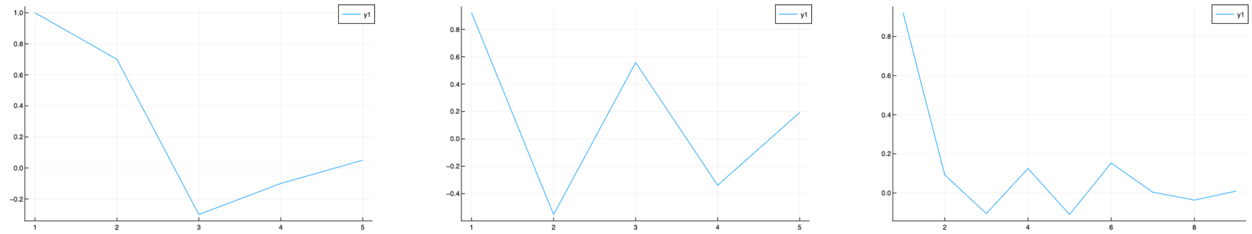

Figure 2: signal v/s t for c, h and h*c respectively.

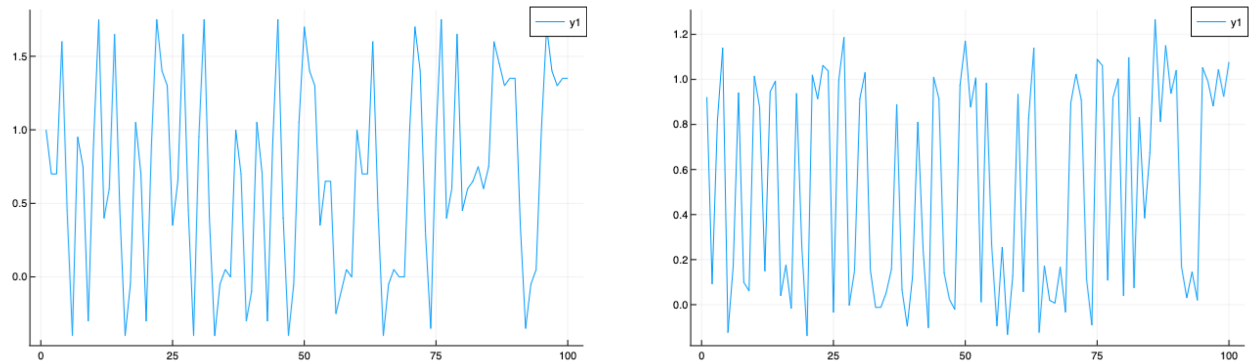

Figure 3: signal v/s t for y and y_tilda respectively.

a From figure 2 we can see that channel impulse response c has some amount of noise. After equalization of the channel impulse response $h * c$, we can see that sufficient amount of noise is reduced while minimizing the effects of the interference.

b From figure 3, it is a little bit clear that $\hat{s}$ will be a worse estimate of s than $\hat{s}^{eq}$.

c The BER for $\hat{s}$=0.113 and BER for $\hat{s}^{eq} = 0.0$, which proves that equalized signal is better than non-equalized signal for estimating the transmitted message.

Problem A7.3

Audio Filtering

Solution

```
using WAV
x, f = wavread("audio_filtering_original.wav");
x = vec(x);
#a)
h_smooth = 1 / 44 * ones(44);
output = conv(h_smooth, x);
```

```
wavplay(output, f); #Play Audio
#b)
IR = zeros(convert(Int32,round(0.03125*f,digits=0))); #Impulse Response
IR[1,1] = 1;
d = f*0.25
k = 0.5
drypath = vcat(IR,zeros(convert(Int32,d)));
wetpath = vcat(zeros(convert(Int32,d)),IR);
out = zeros(size(drypath));
for n in 1:length(drypath)
    out[n,1] = drypath[n,1]+k*wetpath[n,1];
end
#Here out is the h_echo filter
echo_IR = conv(out,x); #h_echo*x
wavplay(echo_IR, f); #Play Audio
echo_IR_2 = conv(out,echo_IR); #h_echo*h_echo*x
wavplay(echo_IR_2, f); #Play Audio
```

a After applying the smooth filter using convolution, the volume of the audio has decreased as well as the sharpness of the audio decreases.

b The h_{echo} is an impulse response filter developed to generate an impulse at a given delayed position of given attenuation value. The code for the same can be found above. After applying the echo filter again, more delayed audio with same amplitude is superimposed on the audio and we can hear the echo more clearly.

Problem 8.7

Interpolation of polynomial values and derivatives

Solution

The four equations can be written as follows:

$$
\begin{aligned}
c_1 &= 0 \\
c_2 &= 0 \\
c_1 + c_2 + c_3 + c_4 + c_5 &= 1 \\
c_2 + 2c_3 + 3c_4 + 4c_5 &= 0
\end{aligned}
$$

These equations can be written as $Ac = b$ as follows:

$$A = (1, 0, 1, 0; 0, 1, 1, 1; 0, 0, 1, 2; 0, 0, 1, 3; 0, 0, 1, 4), b = (0, 0, 1, 0)$$

The set of equations are undetermined as there are only 4 equations and 5 unknown variables.

Problem 8.11

Location from range measurements

Solution

By squaring and expanding on the RHS we get the following:

$$x^T x - 2a_i^T + a_i^T a_i = \rho_i^2, \; i = 1, 2, 3, 4$$

Now subtracting equation 1 from each of the other equations, the term $x^T x$ can be eliminated and the final set of linear equations can be expressed as follows:

$$(-2(a_2 - a_1), -2(a_3 - a_1), -2(a_4 - a_1))x = (\rho_2^2 - \rho_1^2 - \|a_2\|^2 + \|a_1\|^2, \rho_3^2 - \rho_1^2 - \|a_3\|^2 + \|a_1\|^2, \rho_4^2 - \rho_1^2 - \|a_4\|^2 + \|a_1\|^2)$$

Problem 9.5

Fibonacci sequence

Solution

Here, as we want the form $x_t = (y_t, y_{t-1})$, $x_1 = (1, 0)$. For t=2,3,... the formula is $x_{t+1} = x_t + x_{t-1}$, which can be represented as follows:

$$x_{t+1} = (1, 1; 1, 0)x_t, \quad x_1 = (1, 0)$$

The values of y_t for $t = 0, ..., 20$ are as follows:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181$$

The modified fibonacci equation can be expressed in linear dynamic system as follows:

$$x_{t+1} = (1, -1; 1, 0)x_t, \quad x_1 = (1, 0)$$