Group - 9
Dinesh Vaithyalingam Gangatharan
Harsha Nandeeshappa Krishnarajanagar (2677583)

DSP Lab - 1 Introduction to Matlab

Preparation

1. MATLAB - Matrix Laboratory
2. Scalar - a number or character.
   Vector - single row of character or number.
   Matrix - An mxn matrix.

3.

```
v = -1:0.1:1;
s = [];
for i = 1:numel(v)
    if(v(i) >= 0)
        s = [s '+'];
    else
        s = [s '-'];
    end
end
disp(s);
```

Output -
----------+++++++++++

4. - size(G) = 5 x 4, it is not a square matrix since no of rows are not equal to no of columns.
    - Indices of the elements that contain the value 0

```
[row,col] = find(G==0.5);
[row col]

4   ,  1
2   ,  2
4   ,  2
4   ,  4
5   ,  4
```

  - Indices of the element that contain negative values

```
[row,col] = find(G < 0);
    [row col]

     5     2
     2     3
     5     3
     1     4
     2     4
```

6 Difference between MATLAB Script and Function :
MATLAB script is a sequence of commands to be run in an order. Creating these in a file allows iterative running of the program in the particular order.
MATLAB function is a set of processes which is capable of accepting arguments to do computation and can return variables. Both are stored as .m file but function has additional syntax at the start of the file.

4.1 Magic Matrices

Code:

```
fprintf('\n--------4.1 Magic matrices--------\n');

M = magic(5);

fprintf('\n---SUM OF ROWS---\n');
fprintf('Code : sum(M)\n');
disp(sum(M));

fprintf('\n---SUM OF COLS---\n');
fprintf('Code : sum(M")\n');
disp(sum(M'));

fprintf('\n---VALUES IN FIRST ROW---\n');
fprintf('Code : M(1,:)\n');
disp(M(1,:));

fprintf('\n---VALUES IN THIRD COL---\n');
fprintf('Code : M(:,3)\n');
disp(M(:,3));

fprintf('\n---VALUES FROM COL ONE TO THREE---\n');
fprintf('Code : M(:,1:3)\n');
disp(M(:,1:3));

fprintf('\n---VALUES FROM ROW TWO TO END---\n');
fprintf('Code : M(2:end,:)\n');
disp(M(2:end,:));

fprintf('\n--INDICES OF ELEMENTS WITH VALUES >10--\n');
fprintf('Code : find(M>10)\n');
disp(find(M>10));

fprintf('\n--INDICES OF ELEMENTS <4--\n');
fprintf('Code : find(M<4)\n');
disp(find(M<4));
```

Result

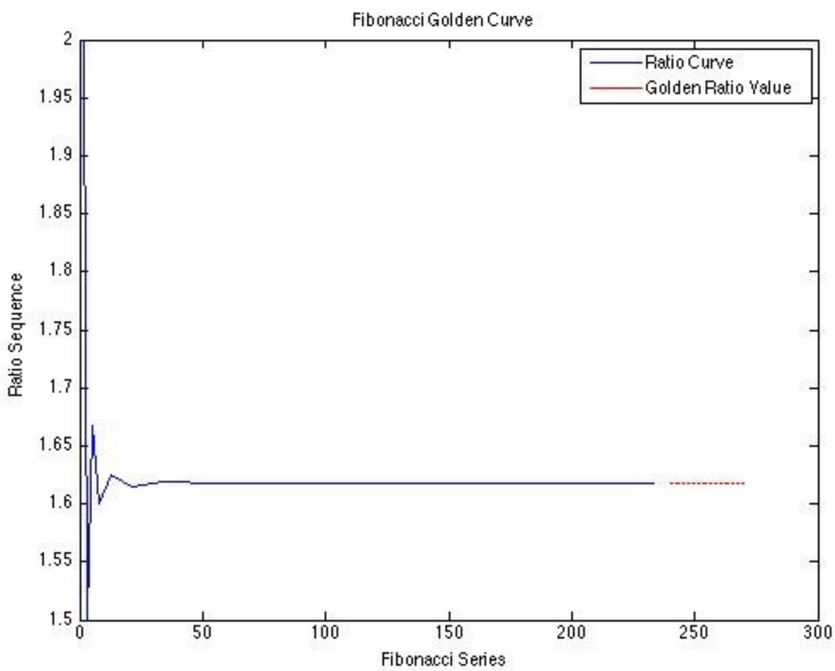
```
--------4.1 Magic matrices--------

---SUM OF ROWS---
Code : sum(M)
    65    65    65    65    65

---SUM OF COLS---
Code : sum(M")
    65    65    65    65    65

---VALUES IN FIRST ROW---
Code : M(1,:)
    17    24     1     8    15

---VALUES IN THIRD COL---
Code : M(:,3)
     1
     7
    13
    19
    25

---VALUES FROM COL ONE TO THREE---
Code : M(:,1:3)
    17    24     1
    23     5     7
     4     6    13
    10    12    19
    11    18    25

---VALUES FROM ROW TWO TO END---
Code : M(2:end,:)
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

--INDICES OF ELEMENTS WITH VALUES >10--
Code : find(M>10)
     1
     2
     5
     6
     9
    10
    13
    14
    15
    17
    18
    19
    21
    22
    23

--INDICES OF ELEMENTS <4--
Code : find(M<4)
    11
    20
    24
```

4.2 Fibonacci Numbers
Code:

```
%----------Fibonacci Numbers-----------
clear all;
clc;

fprintf('\n--------4.2 Fibonacci Numbers--------\n');

N = 12;
fprintf('\nThe fibonacci series for first 12 numbers are \n');
fib = fibonacci(N)';
disp(fib);

ratio_sequence = zeros(N-1,1);

for i = 1 : length(ratio_sequence)
    ratio_sequence(i) = fib(i+1)/fib(i);
end
plot(fib(1,2:end),ratio_sequence);
title('Fibonacci Golden Curve');
xlabel('Fibonacci Series');
ylabel('Ratio Sequence');
legend('Golden Ratio Curve');
```

Result:

```
--------4.2 Fibonacci Numbers--------

The fibonacci series for first 12 numbers are
     1     2     3     5     8    13    21    34    55    89   144   233
```



4.3 Statistical Measurement

code:

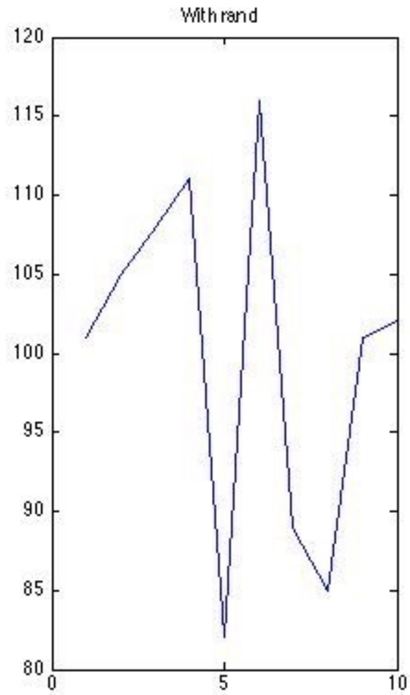
```
x = rand(1000,1);
subplot(1,2,1),plot(hist(x))
minx = min(x);
maxx = max(x);
meanx = mean(x);
stdx = std(x);

y = (4 * x) - 2;
miny = min(y);
maxy = max(y);
meany = mean(y);
stdy = std(y);

fprintf('With rand(x) \n');
fprintf('Mean of x  = %f \n',meanx);
fprintf('Standard Deviation of x = %f
\n',stdx);
fprintf('Minimum value of x = %f
\n',minx);
fprintf('Maximum value of x  = %f
\n',maxx);

fprintf('Mean of y  = %f \n',meany);
fprintf('Standard Deviation of y  = %f
\n',stdy);
fprintf('------------------------------
\n',maxy);


x = randn(1000,1);
subplot(1,2,2),plot(hist(x))
minx = min(x);
maxx = max(x);
meanx = mean(x);
stdx = std(x);

y = (4 .* x) - 2;
miny = min(y);
maxy = max(y);
meany = mean(y);
stdy = std(y);

fprintf('With randn(x) \n');
fprintf('Mean of x  = %f \n',meanx);
fprintf('Standard Deviation of x = %f
\n',stdx);
fprintf('Minimum value of x = %f
\n',minx);
fprintf('Maximum value of x  = %f
```

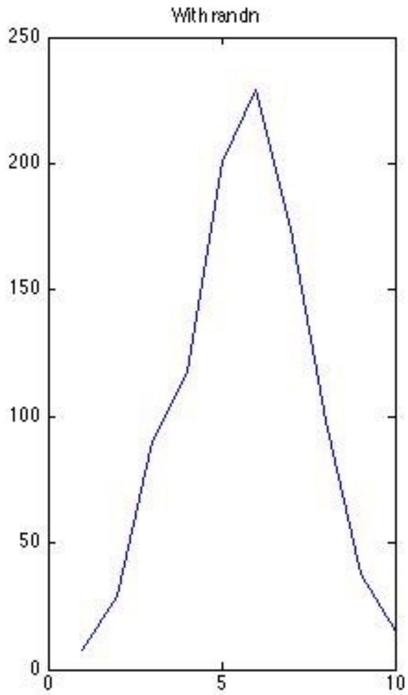
result -
Yes could have predicted the approximate result of y by multiplying the **mean of x** and 4 and subtracting 2

With rand(x)
Mean of x = 0.487812
Standard Deviation of x = 0.289792
Minimum value of x = 0.001722
Maximum value of x = 0.999119
Mean of y = -0.048752
Standard Deviation of y = 1.159168

With randn(x)
Mean of x = 0.053188
Standard Deviation of x = 1.009429
Minimum value of x = -3.534966
Maximum value of x = 3.482697
Mean of y = -1.787246
Standard Deviation of y = 4.037716

randn generated normally distributed pseudorandom numbers.



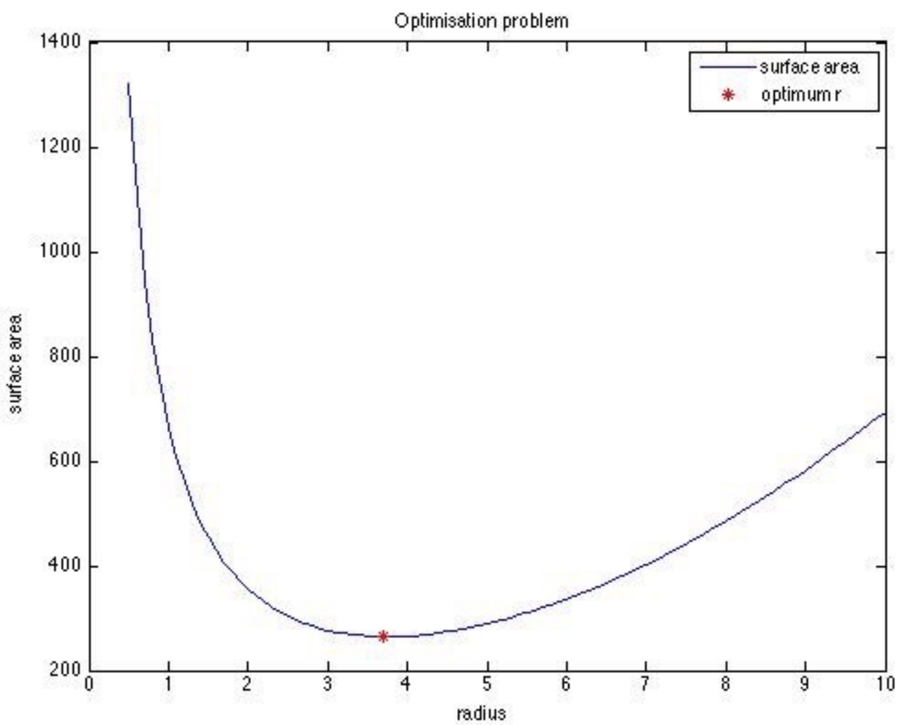


4.4 - An optimisation example

Code :

```
r = 0.5:0.1:10; % in cm
V = 330; % in cm3
aofr = (2 * pi * (r .* r)) + ( (2 * V)./r);
[minaofr,indminaofr] = min(aofr);
optr = r(indminaofr);
h = V/(pi * (r .* r));
plot(r,aofr);
hold on
plot(optr, minaofr,'*r');
title('Optimisation problem');
xlabel('radius'); ylabel('surface area');
fprintf('Minimum Surface area is %f and optimal r is %f and h is %f
\n’,minaofr,optr,h);
```

Result: **Minimum Surface area is 264.395185 and optimal r is 3.700000 and h is 28.0487**

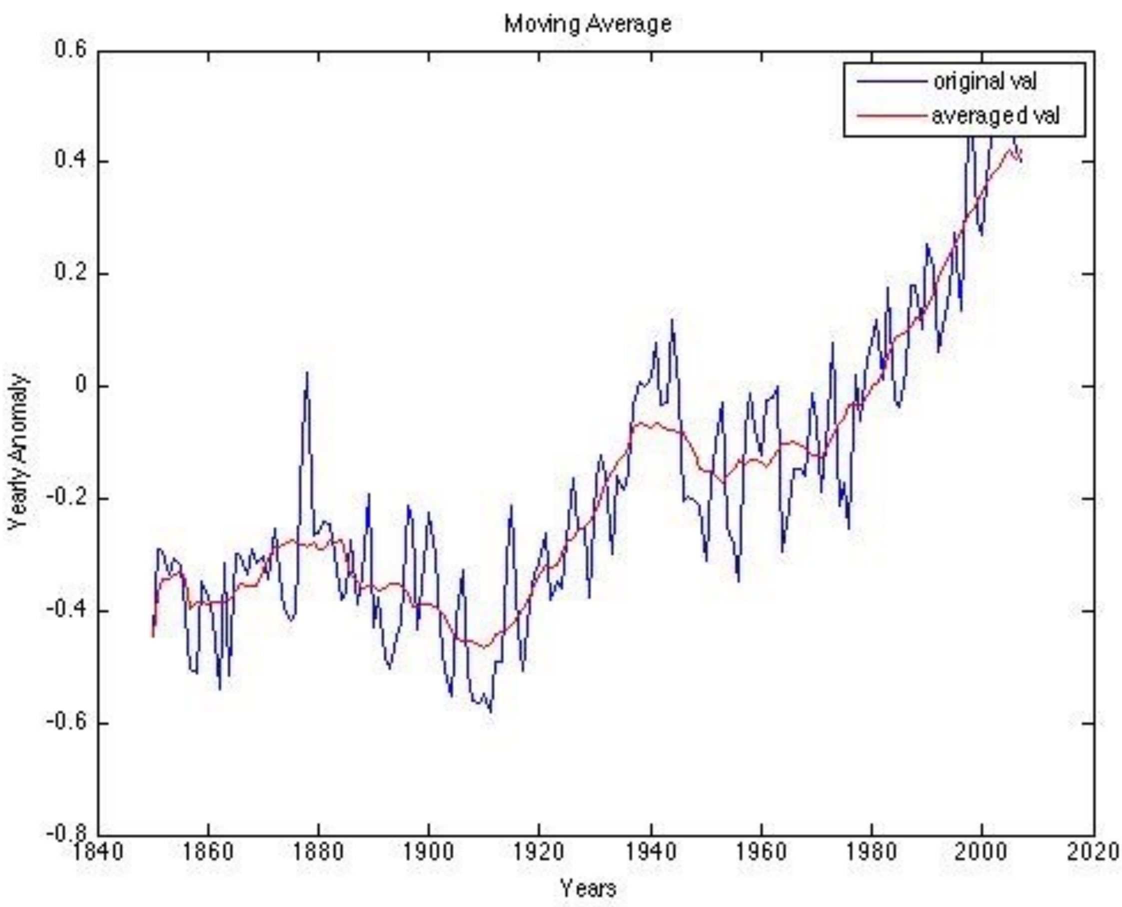


4.5 The moving average

Code:

```
load('glob_warm.mat');
m = 7;
N = numel(year);
mavgTa = zeros(numel(N));
for curyear = 1:N
    if(curyear < (m+1))
        mavgTa(curyear) = (sum(Ta(1:curyear))/curyear);
    elseif(curyear > (N-m))
        mavgTa(curyear) = (sum(Ta(curyear - m:N))/
numel((curyear - m):N));
    else
        mavgTa(curyear) = (sum(Ta(curyear - m: curyear +
m))/((2*m) + 1));
    end
end
plot(year,Ta,'b');
hold on
plot(year, mavgTa,'r');
title('Moving Average'); xlabel('Years'); ylabel('Yearly
Anomaly');
legend('original val','averaged val')
```

result :

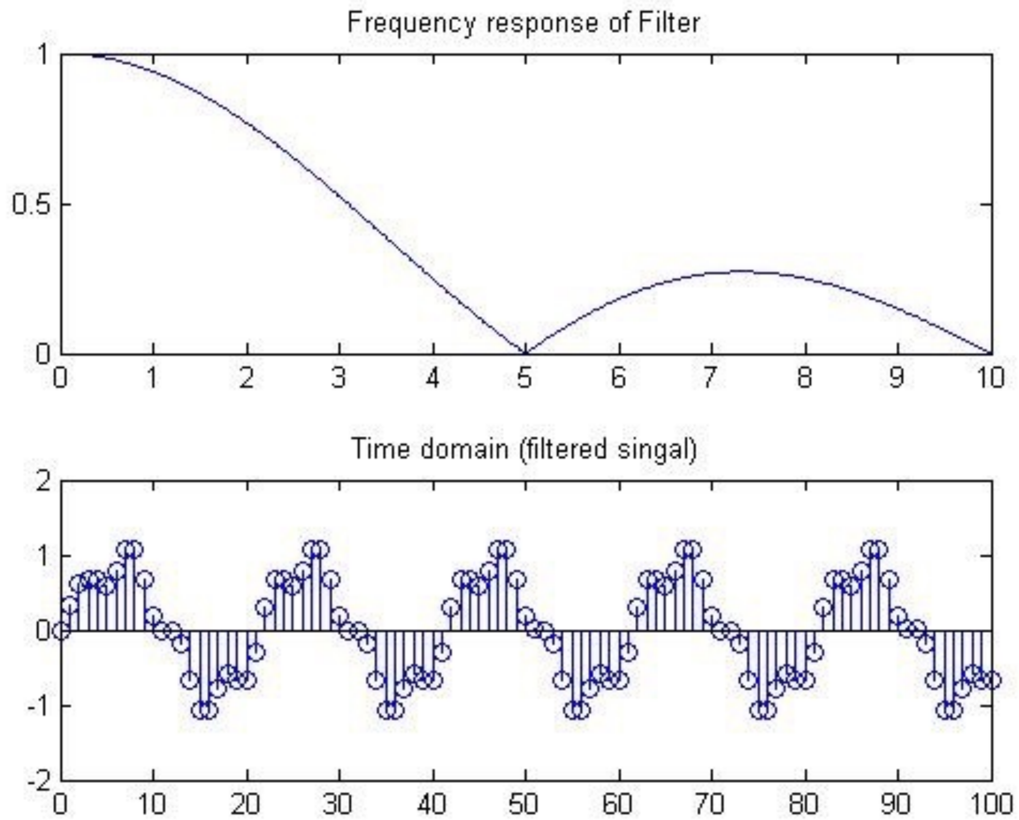


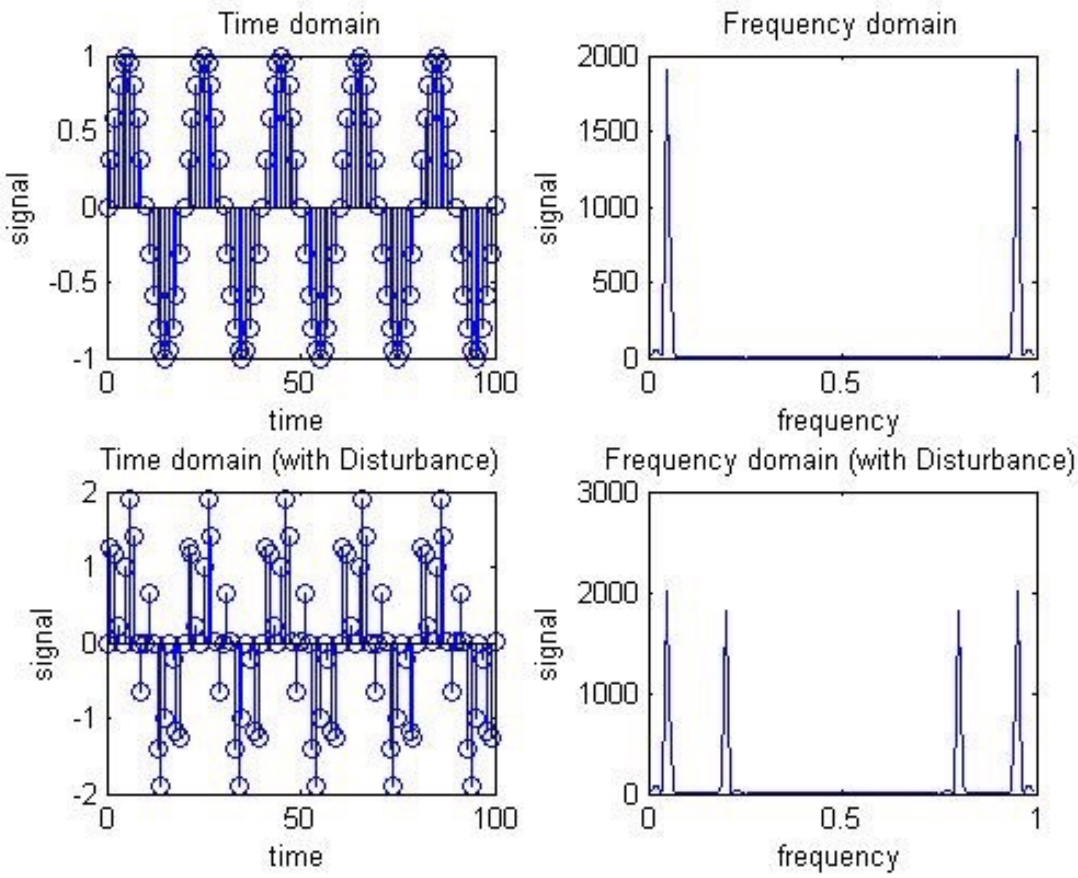
4.6 A signal Processing Example

Code:

```
%Signal Processing Example
clear all;
clc;

n = 0:100; %time vector
F = 1; %frenquency in Hz
T = 0.05; %time in seconds
f=(0:127)/128; %Normailized frequency vector

%% Ideal Signal

s = sin(2*pi*F*n*T);% signal
S=fft(s,128);% Singal in frequency domain
P=S.*conj(S);% Spectrum of the signal


%% Signal with noise

s2=s+sin(2*pi*4*n*T);
S2=fft(s2,128);
P2=S2.*conj(S2);

%% Plots

 figure

 subplot(2,2,1);
 stem(n,s);
 xlabel('time');
 ylabel('signal');
 title('Time domain');

 subplot(2,2,2);
 plot(f,P);
 xlabel('frequency');
 ylabel('signal');
 title('Frequency domain');

 subplot(2,2,3);
 stem(n,s2);
 xlabel('time');
 ylabel('signal');
 title('Time domain (with Disturbance)');

 subplot(2,2,4);
 plot(f,P2);
 xlabel('frequency');
 ylabel('signal');
 title('Frequency domain (with Disturbance)');
```

Result:


Frequency response of Filter
Time domain (filtered singal)


Time domain
Frequency domain
Time domain (with Disturbance)
Frequency domain (with Disturbance)
signal
time
frequency