Analogue Clock – MATLAB

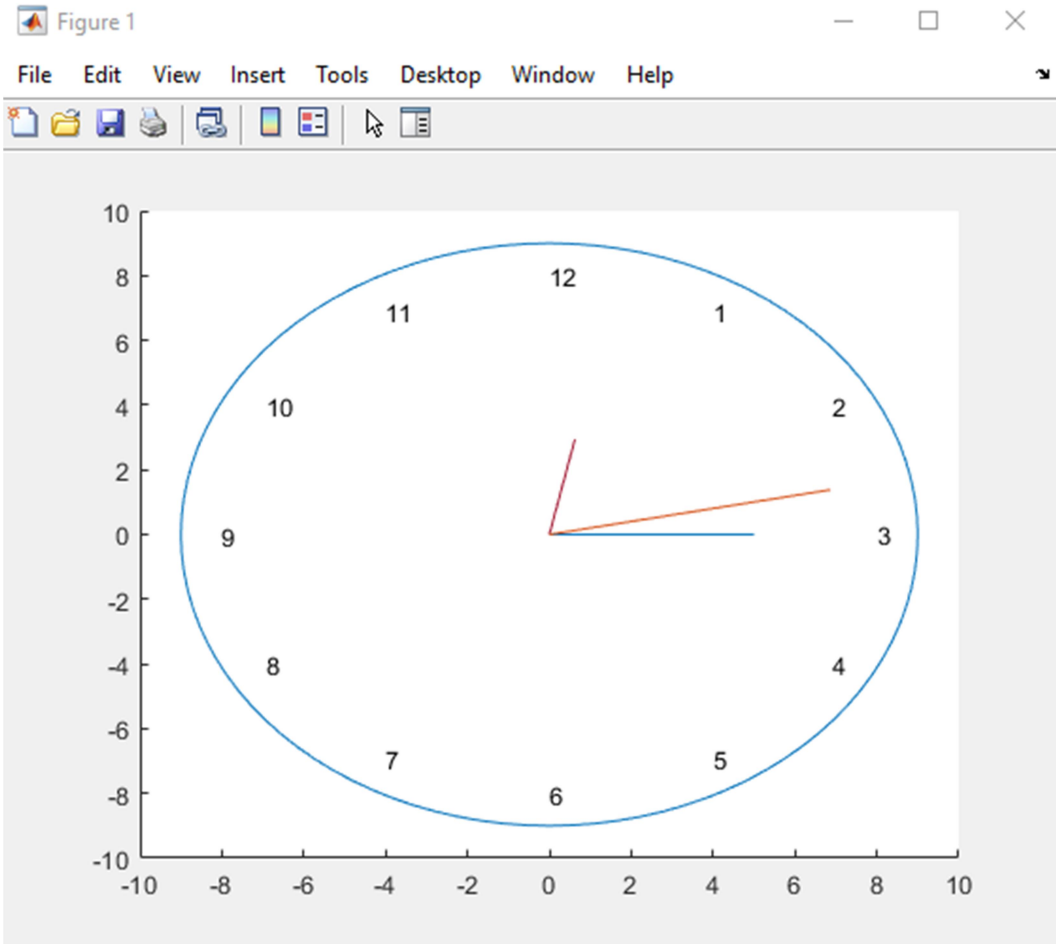
Course: Artificial Intelligence

Group: COM-18

Name: Betul Polat

Introduction

An analogue clock has moving hands, where the smaller one is the hours hand and travels 30° in one hour, and the longer one is the minutes hand and travels 360° in an hour. The seconds hand rotates with a step of 6°. Here we look to design an analogue clock using MATLAB.



Analogue clock using MATLAB

The coding is developed using MATLAB version R2021a. MATLAB has a good collection of graphics commands for plotting and analysing complex signals. The built-in functions reduce the size of the program and provide the desired output.

Program logic

- `x=0;y=0;r=9;`
  → Coordinates which is defined in the beginning of the code.
  Will gonna draw a circle with Radius 9

- `hold on;`
  → Means to overlap multiple outputs in a single frame.

- `th = 0:pi/50:2*pi;`
  → Time period 0 to 2 pi means 0 to 360 degree and in the span of pi by 50 means 180/50

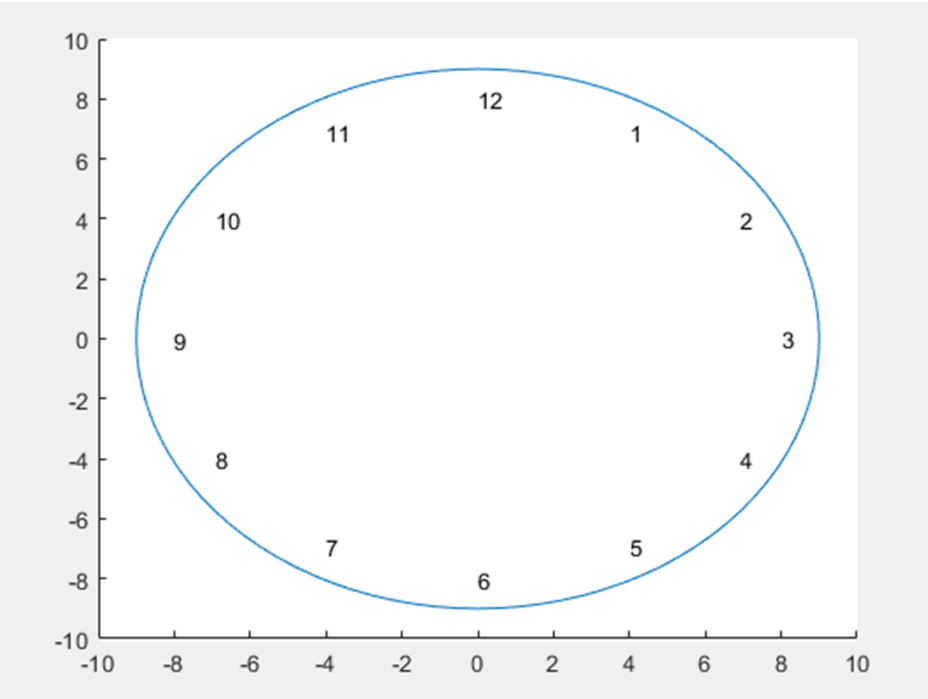
- ```
  xunit = r*cos(th) + x;
  yunit = r * sin(th) + y;
  ```
  → In a XY direction will give the units or into cos theta(max value of cos and sine is: 1 and -1 in the negative direction as well)

- `h = plot(xunit,yunit);`
  → Will plot X and Y. Then we get circle of Radius 9 with units.

- `r=8; s=[1];`
  ➔ We have to plot the hours 1,2,3 till 12. For plotting we have to come inside the circle for that we take the lower Radius and string = 1.
  ```

```
for th = pi/6: pi/6: 2*pi
        ytemp = r* cos(th);
        xtemp = r*sin(th);
        text(xtemp,ytemp,num2str(s));
        s = s + 1;
end
```

- Than we have to give time period PI by 6 that is nothing but 180/6 = 30. 30 is 1 so PI/6 and in the span of 30 till 360 we have to write the values
- ytemp and xtemp are COORDINATES. So we will use sine and cos once again. To locate the position we have to give outputs Y coordinates and number2str. Whichever number is there will convert it to string. String position is 's' we add increment to work.Till 2pi it will increment (In the span of 30 degree)



Logic of clock

```
while(1) → Clock should be always running, condition is 1 means 'always
true'

c = clock; → Function, it will take the system time, date

c = c(1,4:6); -> Time is stored as (hrs: min: sec)

min = c(1,2); sec = c(1,3); →Minute and second information load

if(c(1,1)>12) → c(1,1)== hours from matrix

    hr = c(1,1)-12; → am/pm denote

else
    hr = c(1,1);
end
min1 = ceil(min/12); → Ceil == round up to 12, divided by 12

theta = (hr*pi)/6 + (min1*pi)/30;

f = figure(1); hold on; → overlap the output

ytemp = 3 *cos(theta); Y = [0 ytemp]; → 0 ytemp = short minute, long
second
xtemp = 3 *sin(theta); X = [0 xtemp]; → 0 xtemp = lines

p=plot(X,Y); hold on; → overlap the output

theta1 = (min*pi)/30; → The lines are going to be at the center

ytemp1 = 5*cos(theta1); Y1 = [0 ytemp1]; → should rotate by 30 degree
xtemp1 = 5*sin(theta1); X1 = [0 xtemp1]; → degree
30 degree per hour, 360 per hour

p1=plot(X1,Y1);
theta2 = (sec*pi)/30; → sec will rotate one by one

ytemp2 = 7*cos(theta2); Y2 = [0 ytemp2]; →minute shoud rotate by 360
xtemp2 = 7*sin(theta2); X2 = [0 xtemp2];
p2=plot(X2,Y2);

pause(1); → Will cause a delay of one second in the rotation
delete(p2); →We have to delete whatever the plot was there each time in
the loop, and new location will be added.
delete(p1);
delete(p);
end
```