EEE391 - Computer Assignment 2

Çelik Köseoğlu - 21400196

(a)

Student Number: 21400196

```
y[-3]  = 2
y[-2]  = -1
y[-1]  = 0
y[0]   = 4
y[1]   = 0
y[2]   = 0
y[3]   = 0
y[4]   = -1
y[5]   = 9
y[6]   = 6
y[n]   = 0, otherwise
```

I have tested this part of the program using my Student ID. When you run my Matlab program, you have to select a pulse type to see its graph. Here are the screenshots for i-iv)

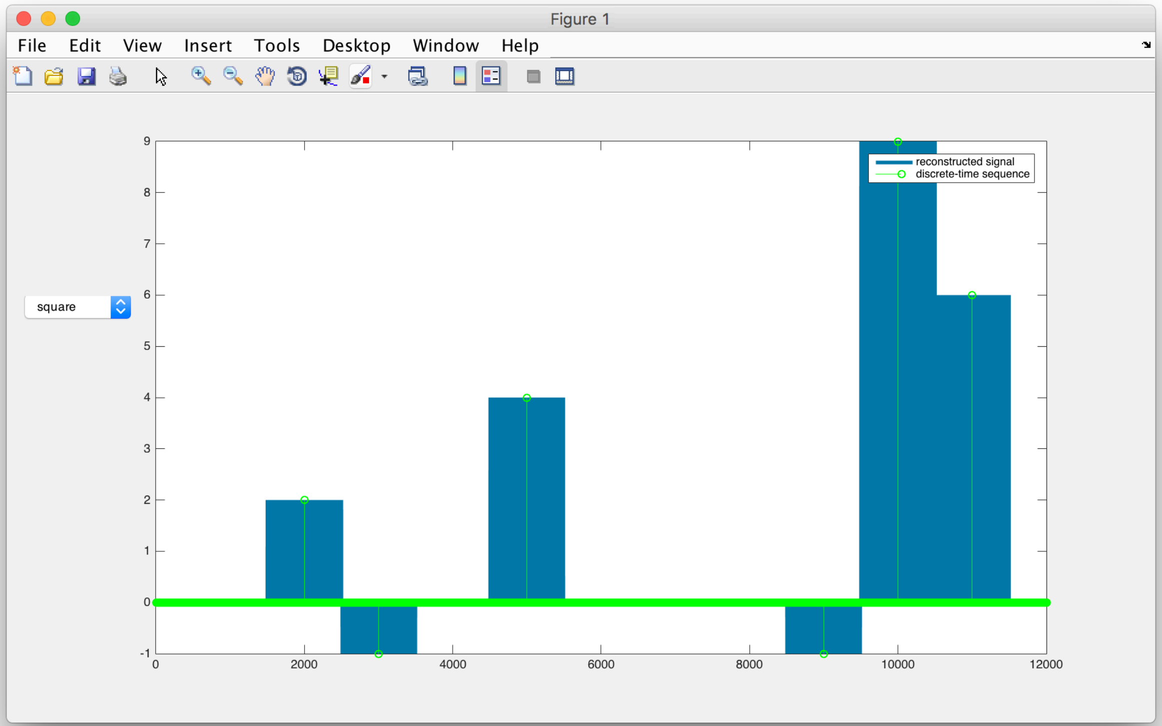

Figure 1 - Square Pulse

We could see that my student ID 21400196 is reflected on the graph. Discrete time sequences; y[n] values are highlighted with green.

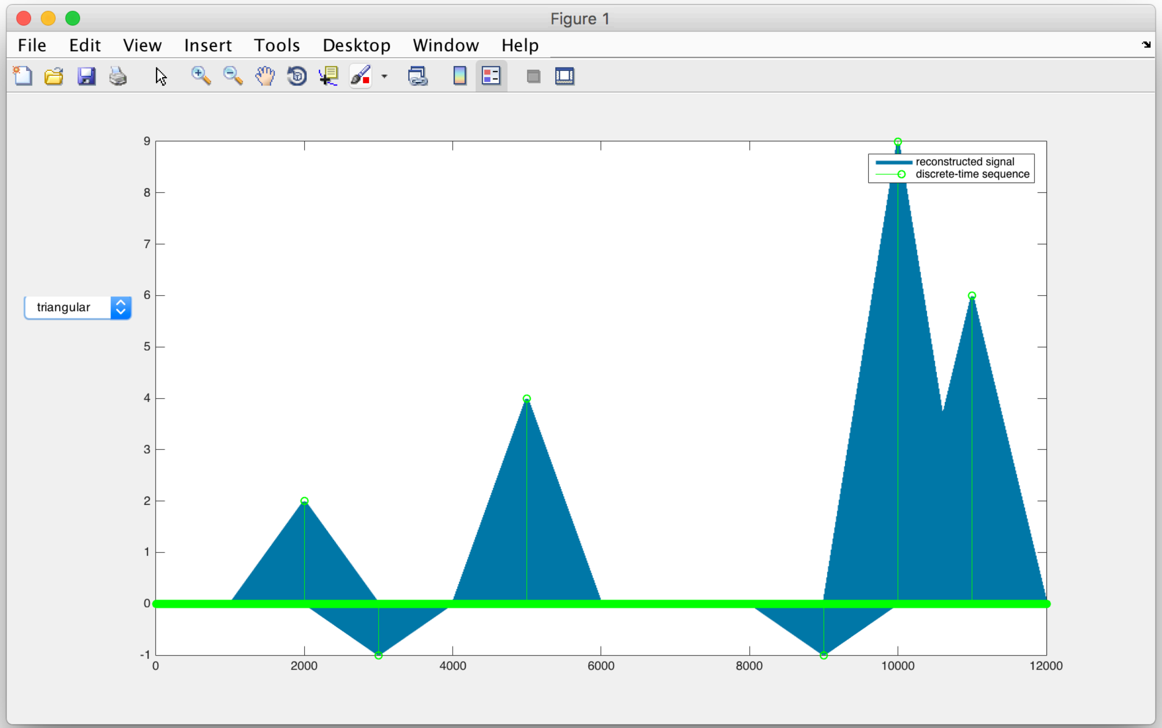

Figure 2 - Triangular Pulse

Again, my student number is represented using the square pulse. The highest point of my graph is the y[n] value. They are again, highlighted with green.

For the parabolic pulse, I have used the following function. The graph is on the next page.

$$\frac{2 - 2\cos x}{x^2}$$

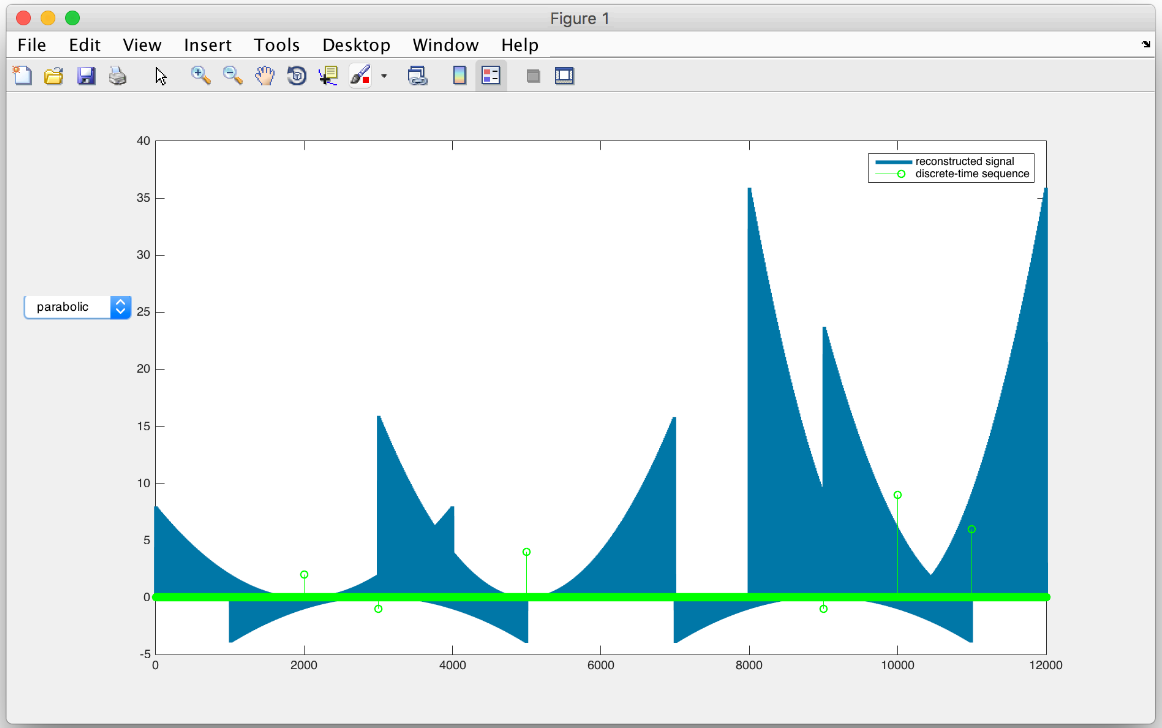

Figure 3 - Parabolic Pulse

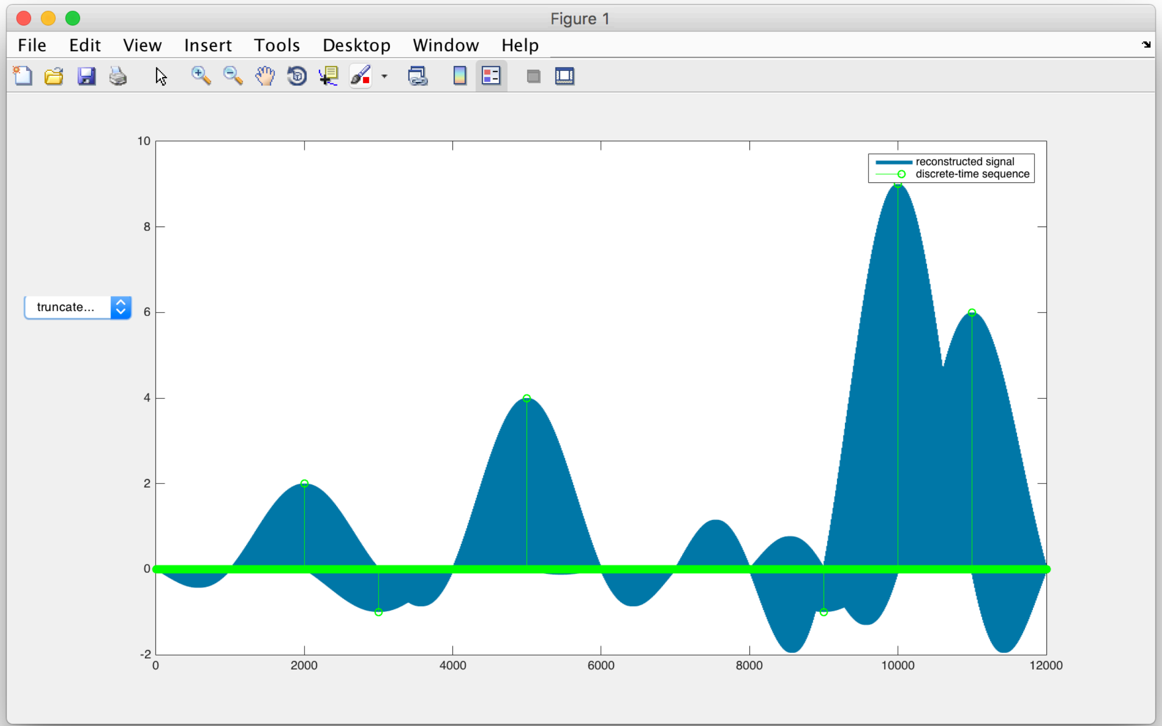

Figure 4 - Truncated Pulse

(b) the selection of the pulse type has significant effects on the output pulse. Plots with the provided partial function are shown below

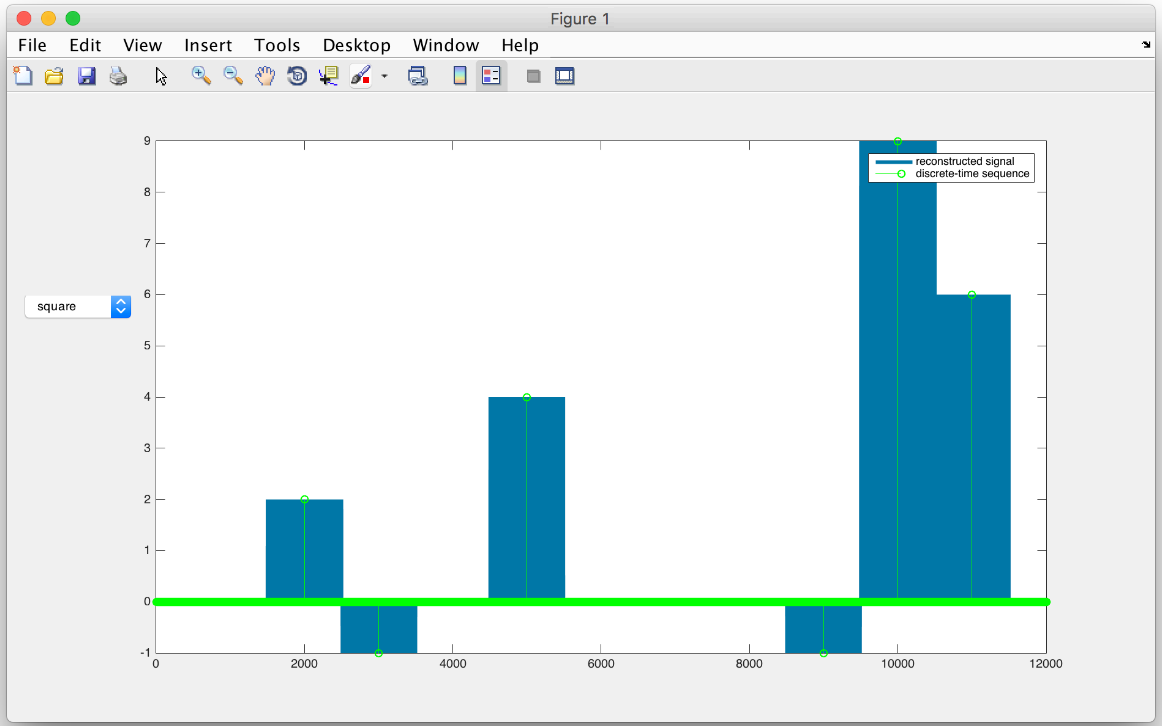

Figure 5 - Double Sided Square Pulse

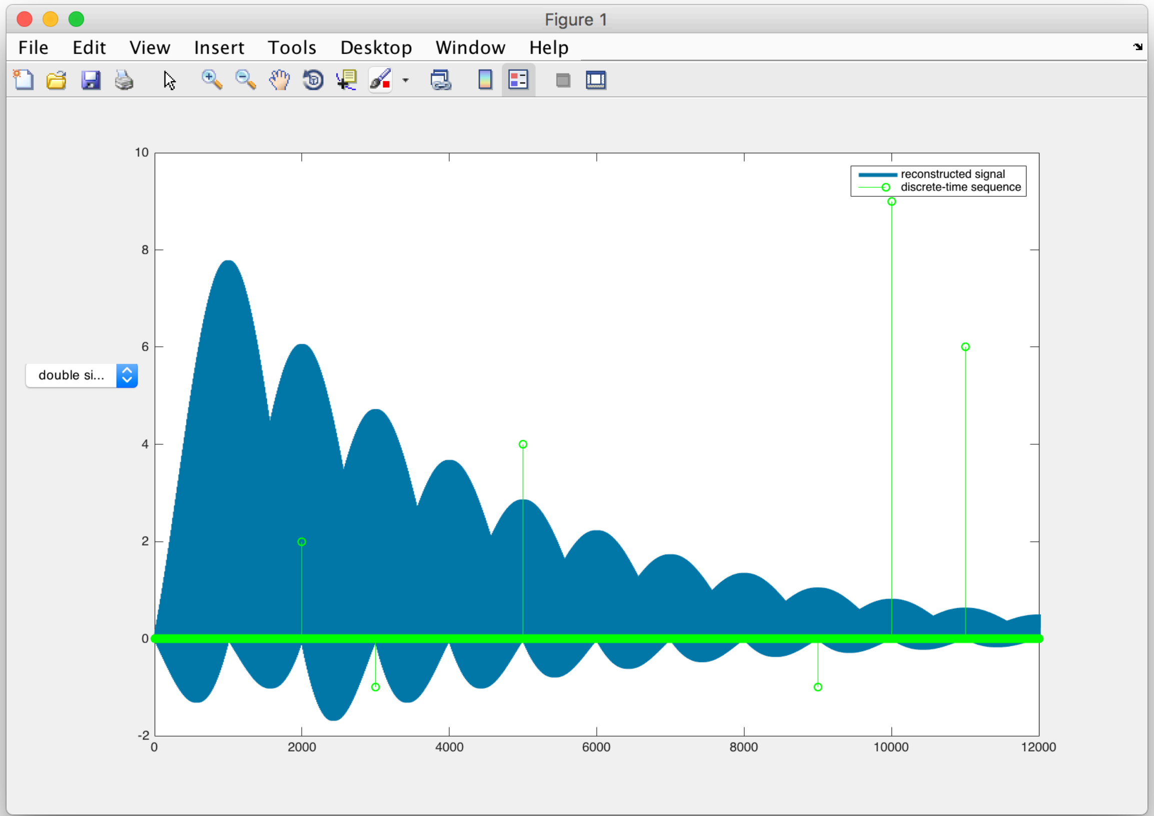

Figure 6 - Double Sided Truncated Pulse

(c)

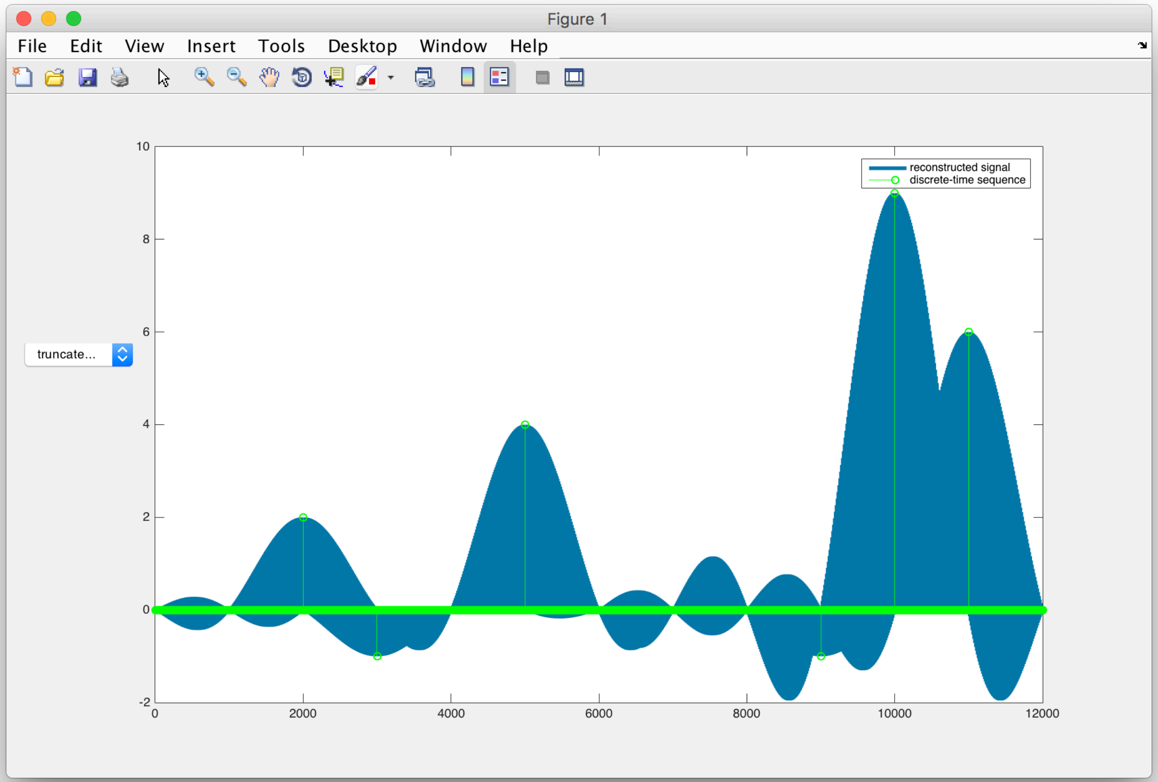

Figure 7 - Truncated Pulse 5T

The difference may not be visible on this graph but when we do zoom in, we see that the graph starts to lose its curves and behave more like a square wave.

Since I choose to sample a lot of points, the difference is not really visible on this one.

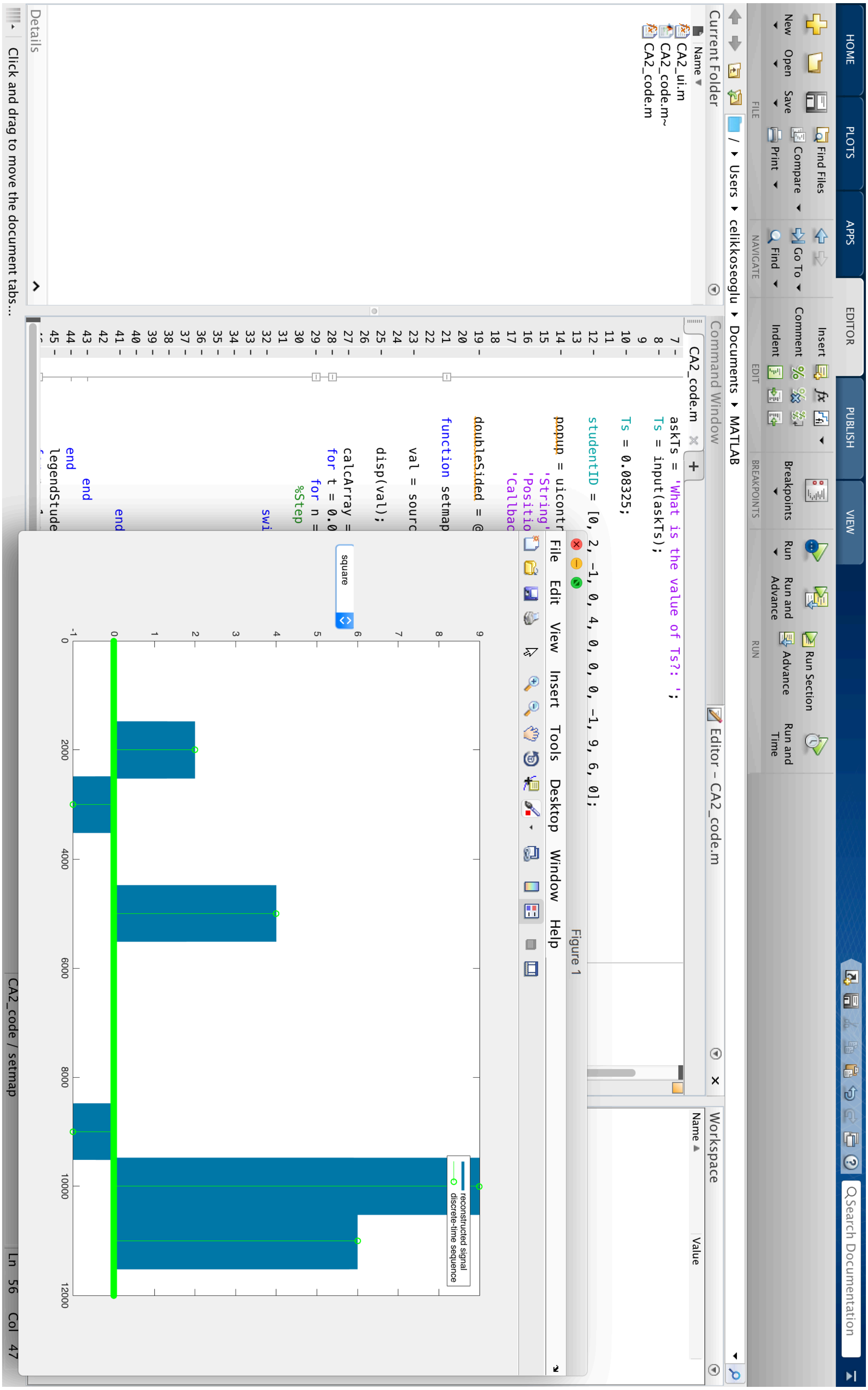

Figure 8 - MATLAB Working Environment

MATLAB Code:

```
function CA2_code

    clc;
    close all;

    %ask for user input Ts
    askTs = 'What is the value of Ts?: ';
    Ts = input(askTs);

    Ts = 0.08325;

    studentID = [0, 2, -1, 0, 4, 0, 0, 0, -1, 9, 6, 0];

    popup = uicontrol('Style', 'popup',...
           'String', {'square','triangular','parabolic','truncated
ideal', 'double sided'},...
           'Position', [10 340 100 10],... %margin-left, margin-bottom,
width, ??
           'Callback', @setmap);

    doubleSided = @(n) 10 * exp(-0.25*abs(n));

    function setmap(source, event)

        val = source.Value;

        calcArray = [];
        for t = 0.0 : 0.001 : 0.999
            for n = 1 : 12
             %Step by increments of 0.001, start from 0.0, end at 1

                switch val
                    case 1
                        calcArray = [calcArray, studentID(n) *
squareFunction(Ts, t-n*Ts)];
                    case 2
                        calcArray = [calcArray, studentID(n) *
triangleFunction(Ts, t-n*Ts)];
                    case 3
                        calcArray = [calcArray, studentID(n) *
parabolicFunction(Ts, t-n*Ts)];
                    case 4
                        calcArray = [calcArray, studentID(n) *
truncatedFunction(Ts, t-n*Ts)];
                    case 5
                        calcArray = [calcArray, doubleSided(n) *
truncatedFunction(Ts, t-n*Ts)];
                end

            end
        end
        legendStudentID = [];
        for n = 1 : 12000
            legendStudentID = [0, legendStudentID];
```

```
        end
        for n = 1 : 12
            legendStudentID(n * 1000) = studentID(n);
        end
        dtx = 1:12000;
        hold off;
        plot(dtx ,calcArray, 'color', [0.066, 0.474, 0.654],
'LineWidth', 3); hold on;
        atx = 1:12000;
        stem(atx, legendStudentID, 'color', 'g');
        legend('reconstructed signal','discrete-time sequence')
    end

    function squarepT = squareFunction(Ts, t)
        if (-Ts/2) <= t && t <= (Ts/2)
            squarepT = 1;
        else
            squarepT = 0;
        end
    end

    function triangularpT = triangleFunction(Ts, t)
        if -Ts <= t && t <= Ts
            triangularpT = 1 - (abs(t)/Ts);
        else
            triangularpT = 0;
        end
    end

    function parabolicpT = parabolicFunction(Ts, t)
        if -2*Ts <= t && t <= 2*Ts
            parabolicpT = (2 - 2*cos(t))/(Ts*Ts);
        else
      parabolicpT = 0;
    end
  end

  function truncatedpT = truncatedFunction(Ts, t)
    if (-Ts*5) <= t && t <= (Ts*5) %change 3 with 5 for part C
      truncatedpT = sin((pi/Ts)*t)/((pi/Ts)*t);
    else
      truncatedpT = 0;
    end
  end
end
```