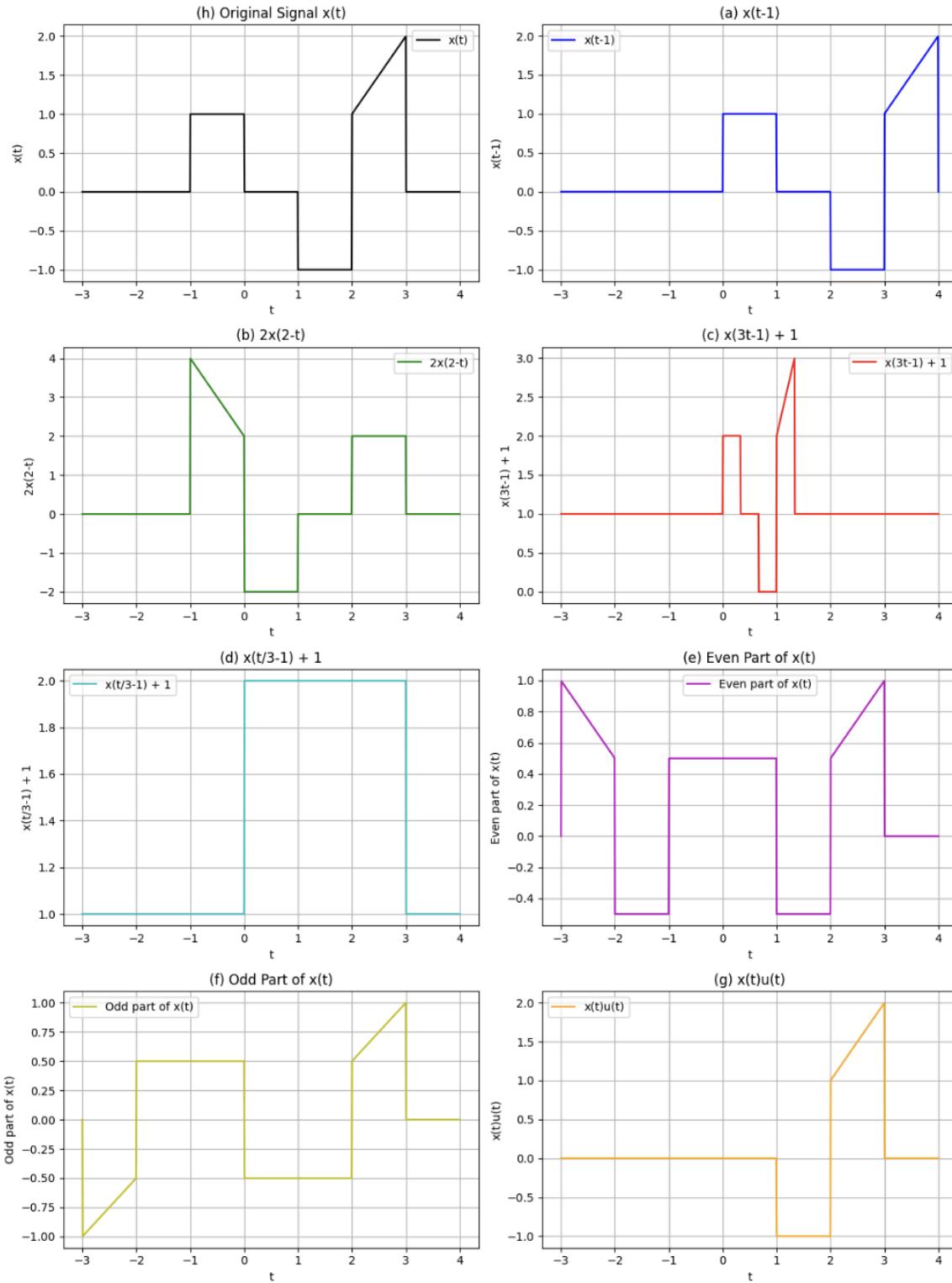
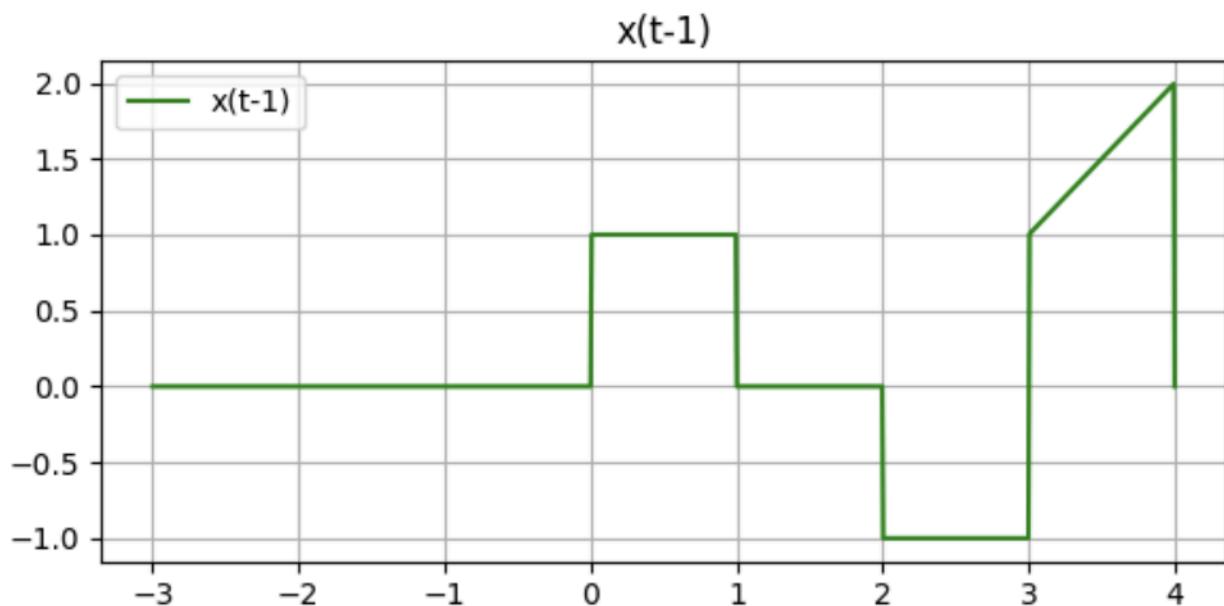


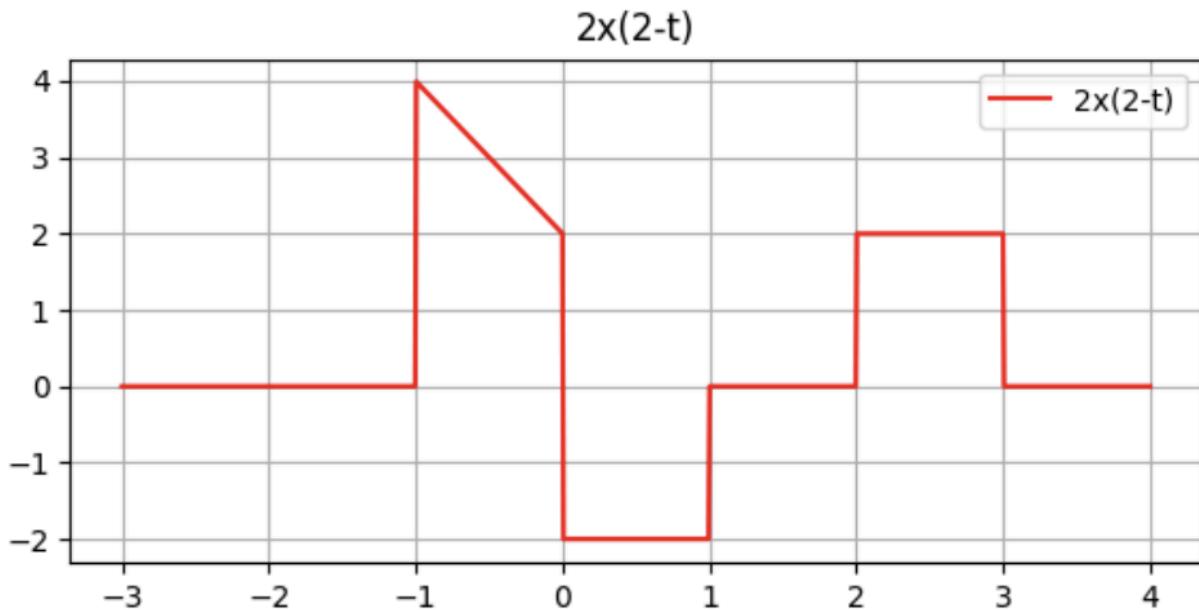
- 1) all signals plotted - d) is cut off by range but the full function is shown in part d)



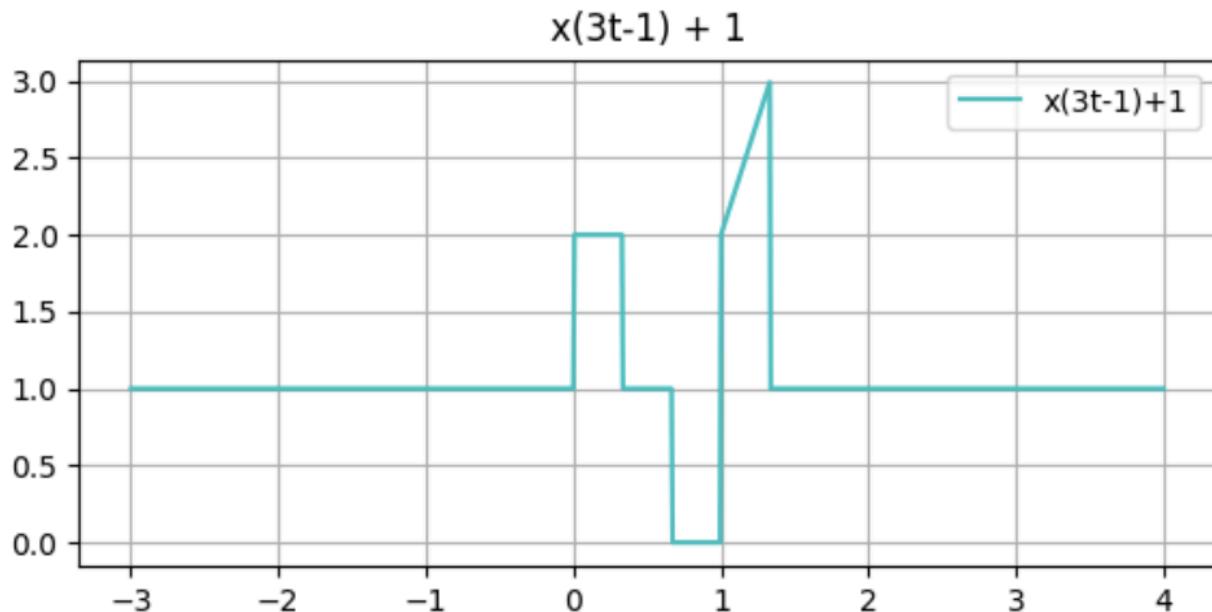
a) $x(t - 1)$



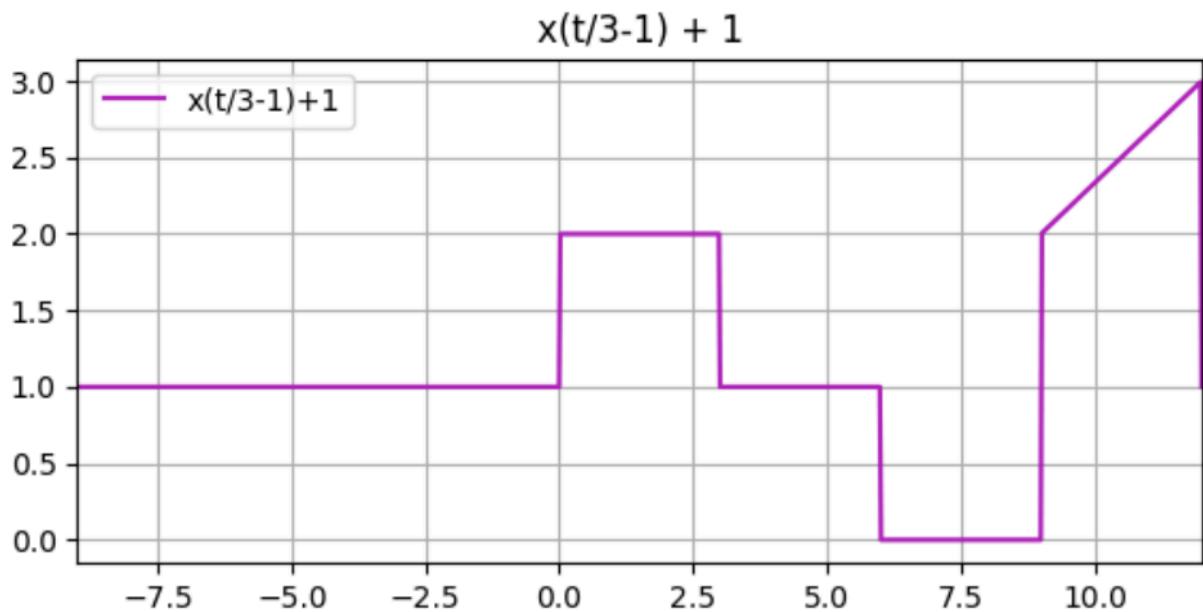
b) $2x(2 - t)$

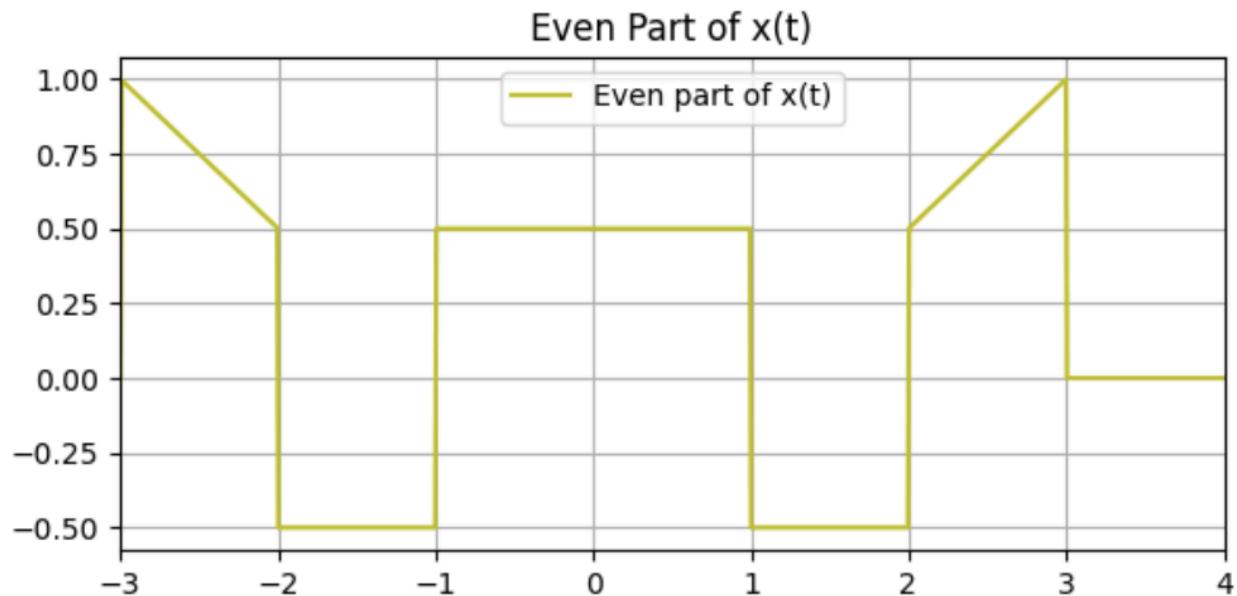
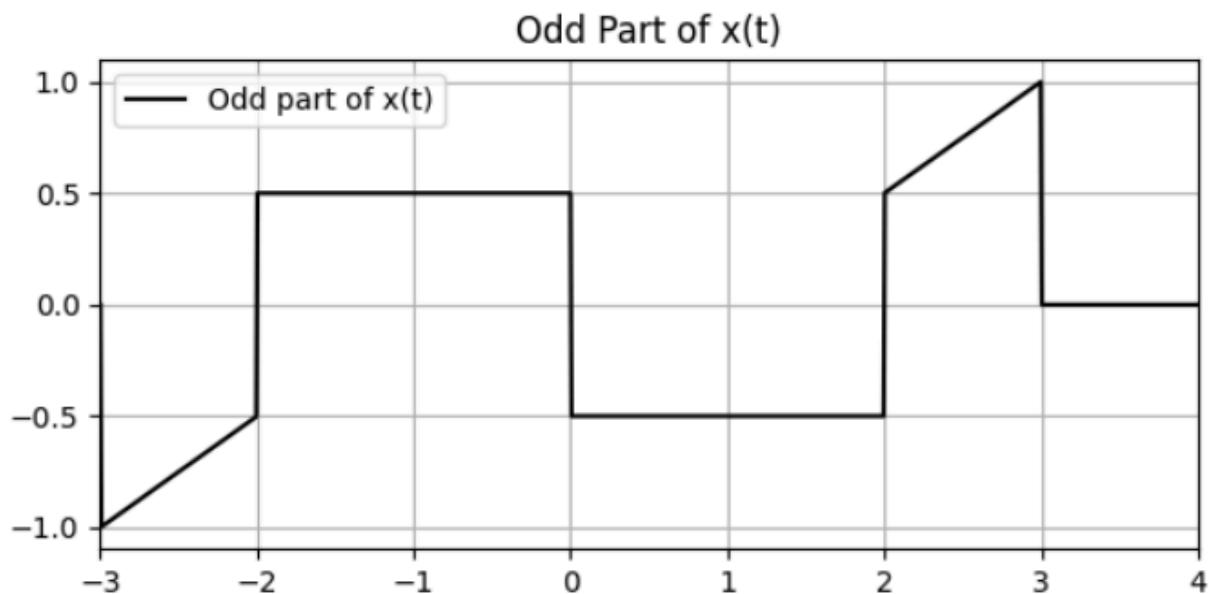


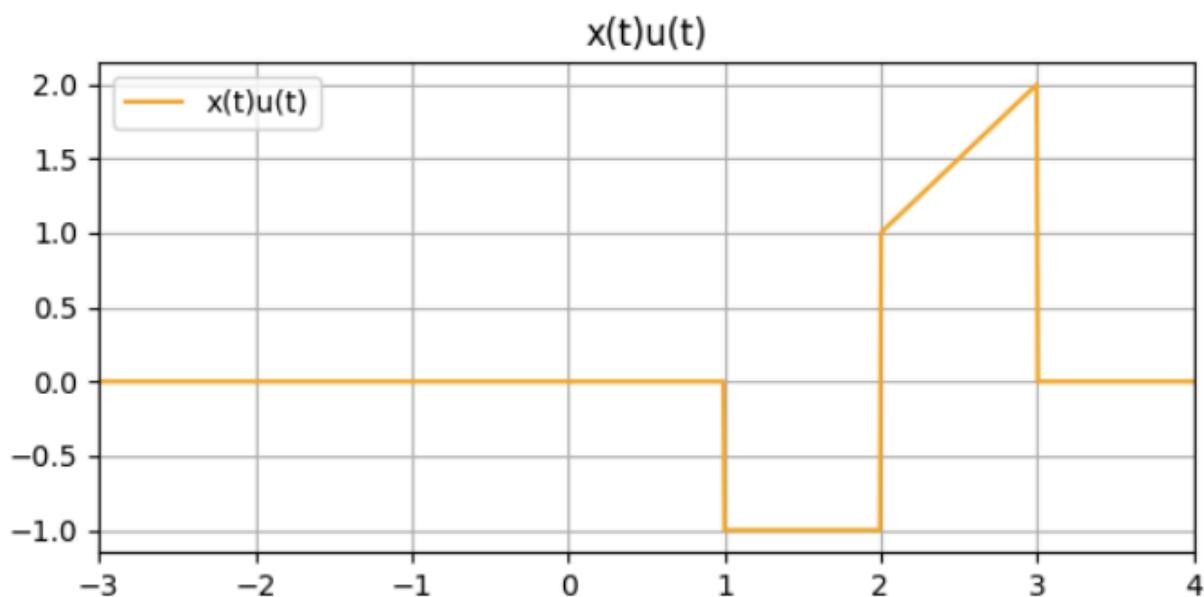
c) $x(3t - 1) + 1$



d) $x(t/3 - 1) + 1$

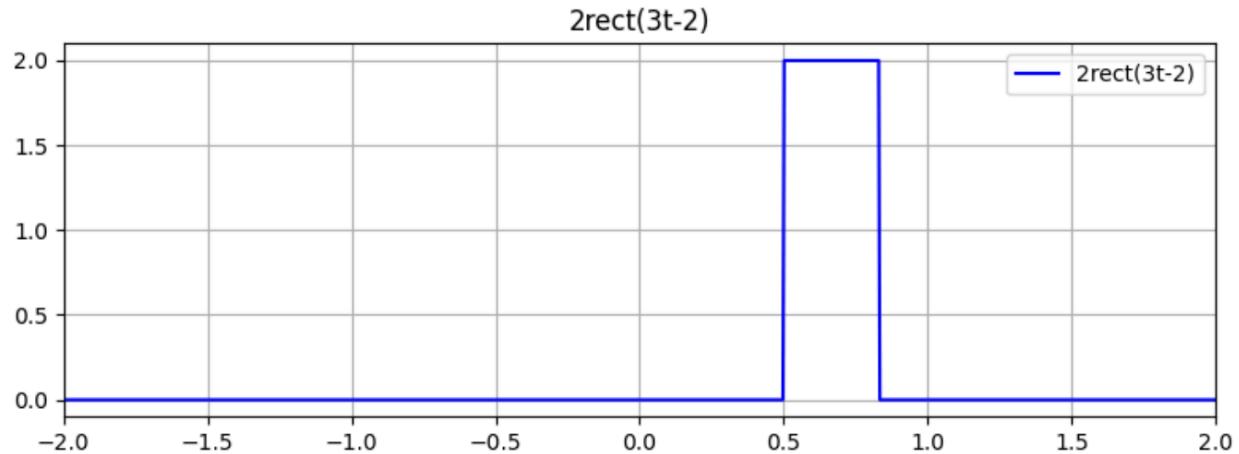


e) Even part of $x(t)$ f) Odd part of $x(t)$ 

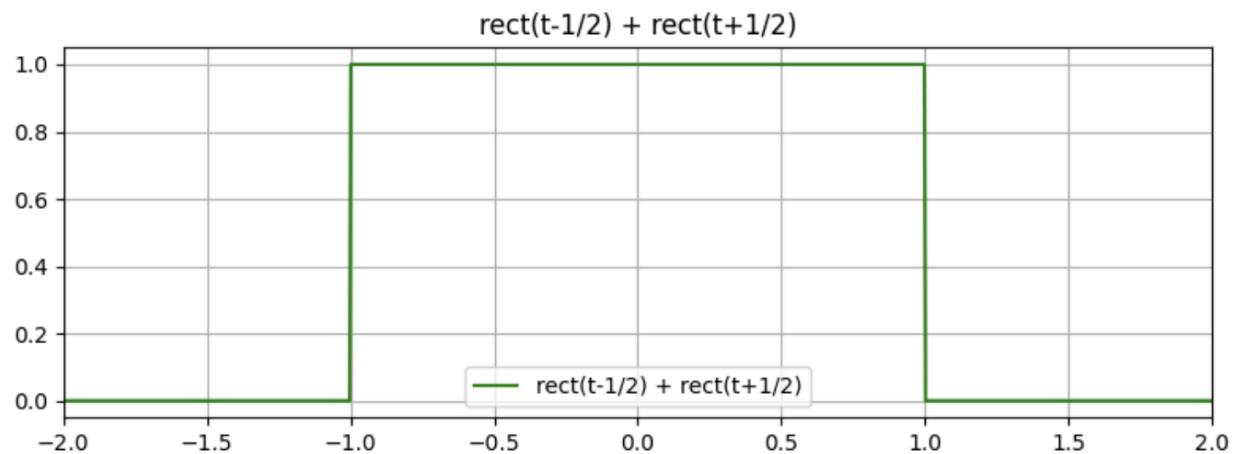
g) $x(t)u(t)$ 

2)

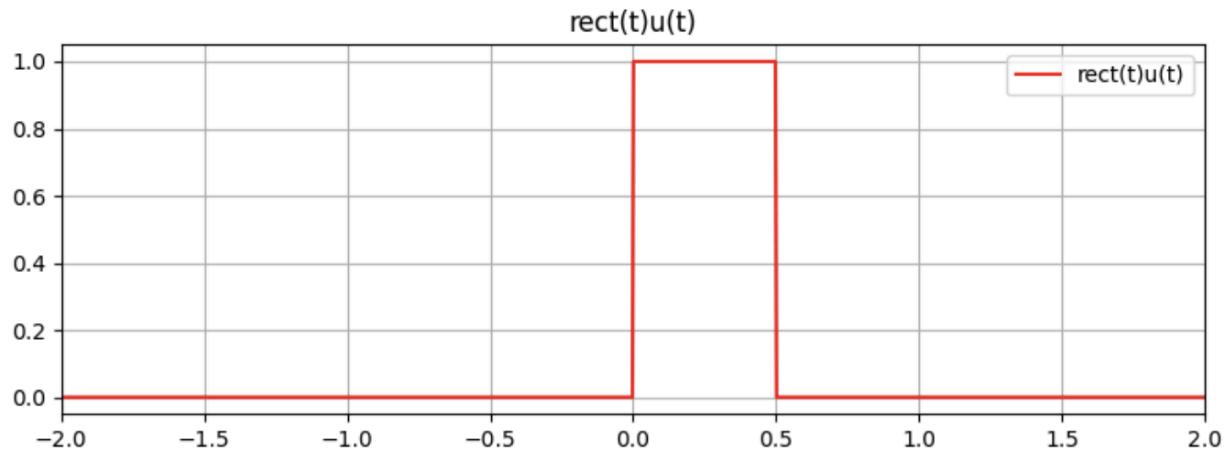
a) $2\text{rect}(3t - 2)$



b) $\text{rect}(t - 1/2) + \text{rect}(t + 1/2)$



c) $\text{rect}(t)u(t)$



3)

a) $V_S = A\angle 0$

b) $V_L = Z_E V_S$

$$V_{total} = R + jwL - \frac{j}{wC}$$

$$V_L = I * Z_L$$

$$V_L = \frac{V_S}{Z_{total}} * jwL$$

$$V_E = \frac{V_L}{V_S} = \frac{jwL}{Z_{total}}$$

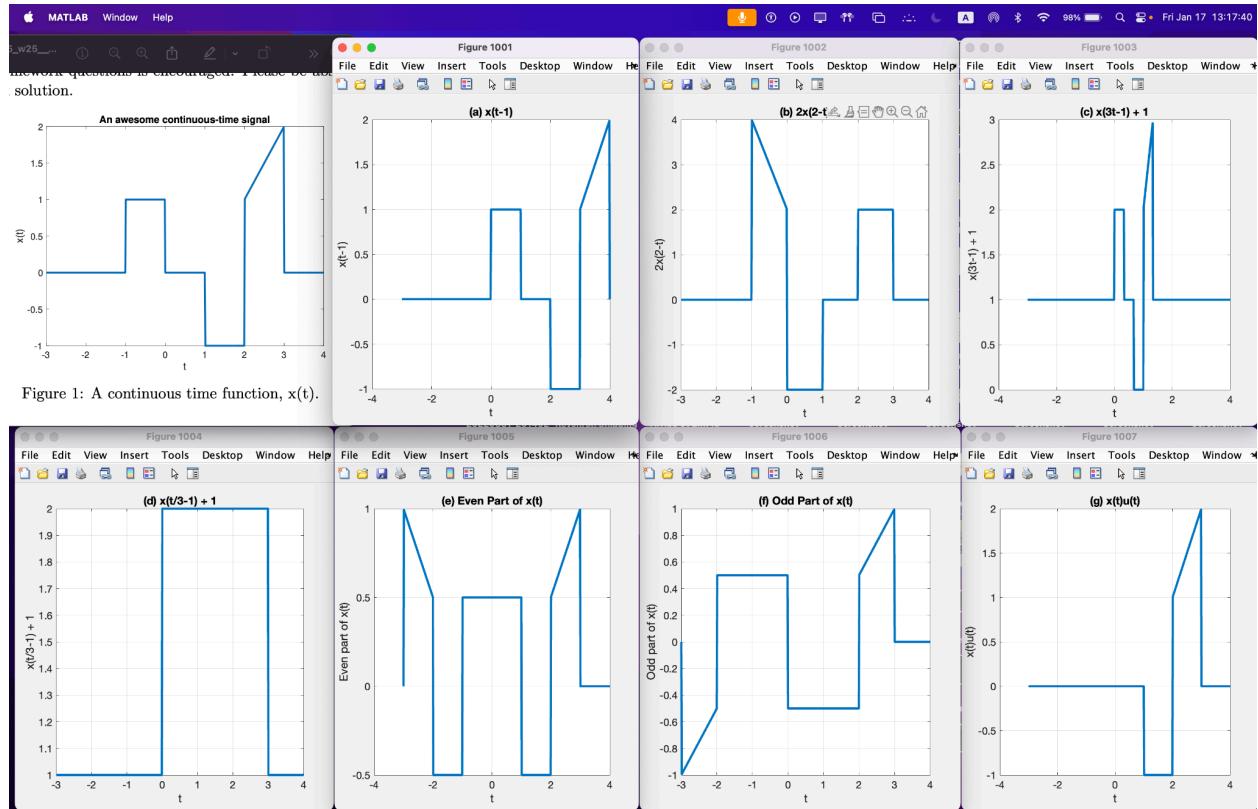
c) $V_L = (R + j(wL - \frac{1}{wC}))A\angle 0$

d) $|Z_E| = \sqrt{R^2 + (wL - \frac{1}{wC})^2}$

e) $\theta_{Z_E} = \tan^{-1}(\frac{wL - \frac{1}{wC}}{R})$

f) $v_L(t) = A|Z_E| \cos(wt + \theta_{Z_E})$

4) MATLAB script



Contents

- [Problem 1](#)
 - [Put my functions here](#)
-

```
% hw2_ece_45_w25_v1.m
%
% Script to produce the required outputs for the most amazing course in the
% ECE curriculum worldwide, ECE 45
%
% Created by Andrew Onozuka
%
% Change history
%   January 8, 2025 - initial version
%
% Pending items to finish
%   N/A
%
```

Problem 1

```
% Use this template below to do some plotting

t = linspace(-3, 4, 500); % Time vector for plotting
ct_fun = ct_function(t);

% (a) x(t-1)
figure(1001)
x_a = ct_function(t - 1);
plot(t, x_a, 'LineWidth', 3);
xlabel('t'); ylabel('x(t-1)');
title('(a) x(t-1)');
grid on; set(gca, 'FontSize', 14);

% (b) 2x(2-t)
figure(1002)
x_b = 2 * ct_function(2 - t);
plot(t, x_b, 'LineWidth', 3);
xlabel('t'); ylabel('2x(2-t)');
title('(b) 2x(2-t)');
grid on; set(gca, 'FontSize', 14);

% (c) x(3t-1) + 1
figure(1003)
x_c = ct_function(3 * t - 1) + 1;
plot(t, x_c, 'LineWidth', 3);
xlabel('t'); ylabel('x(3t-1) + 1');
title('(c) x(3t-1) + 1');
grid on; set(gca, 'FontSize', 14);

% (d) x(t/3-1) + 1
figure(1004)
x_d = ct_function(t / 3 - 1) + 1;
plot(t, x_d, 'LineWidth', 3);
xlabel('t'); ylabel('x(t/3-1) + 1');
```

```

title('(d) x(t/3-1) + 1');
grid on; set(gca, 'FontSize', 14);

% (e) Even part of x(t)
figure(1005)
x_even = (ct_function(t) + ct_function(-t)) / 2;
plot(t, x_even, 'LineWidth', 3);
xlabel('t'); ylabel('Even part of x(t)');
title('(e) Even Part of x(t)');
grid on; set(gca, 'FontSize', 14);

% (f) Odd part of x(t)
figure(1006)
x_odd = (ct_function(t) - ct_function(-t)) / 2;
plot(t, x_odd, 'LineWidth', 3);
xlabel('t'); ylabel('Odd part of x(t)');
title('(f) Odd Part of x(t)');
grid on; set(gca, 'FontSize', 14);

% (g) x(t)u(t)
figure(1007)
u_t = t >= 0; % Unit step function
x_g = ct_function(t) .* u_t;
plot(t, x_g, 'LineWidth', 3);
xlabel('t'); ylabel('x(t)u(t)');
title('(g) x(t)u(t)');
grid on; set(gca, 'FontSize', 14);

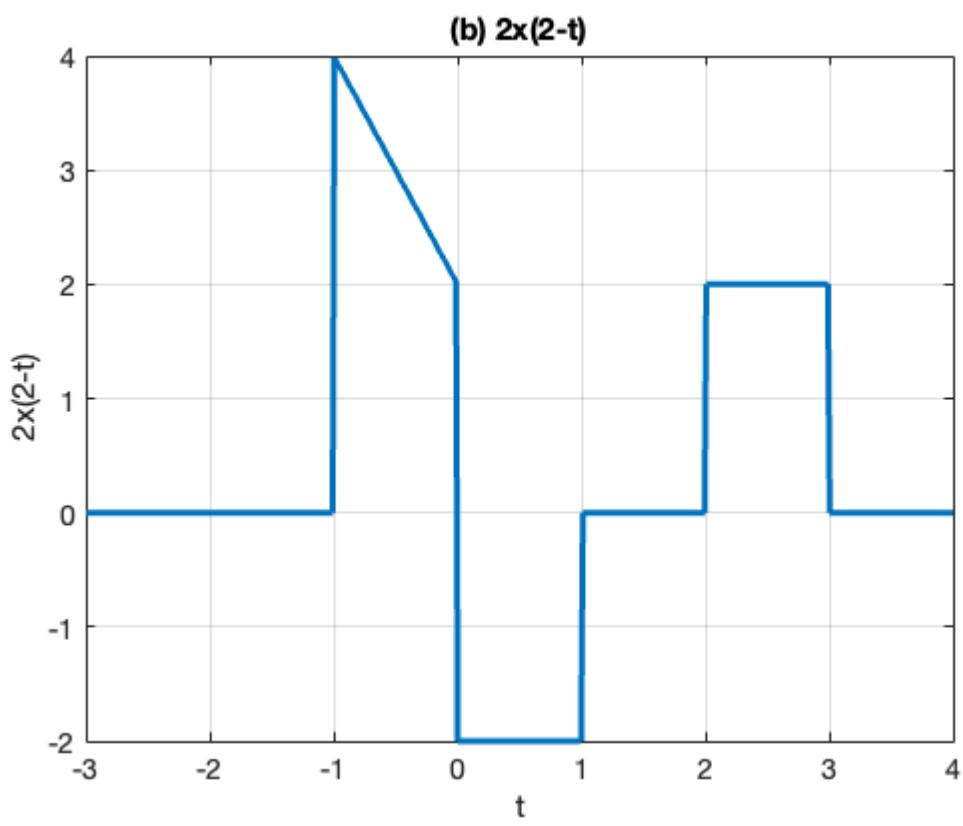
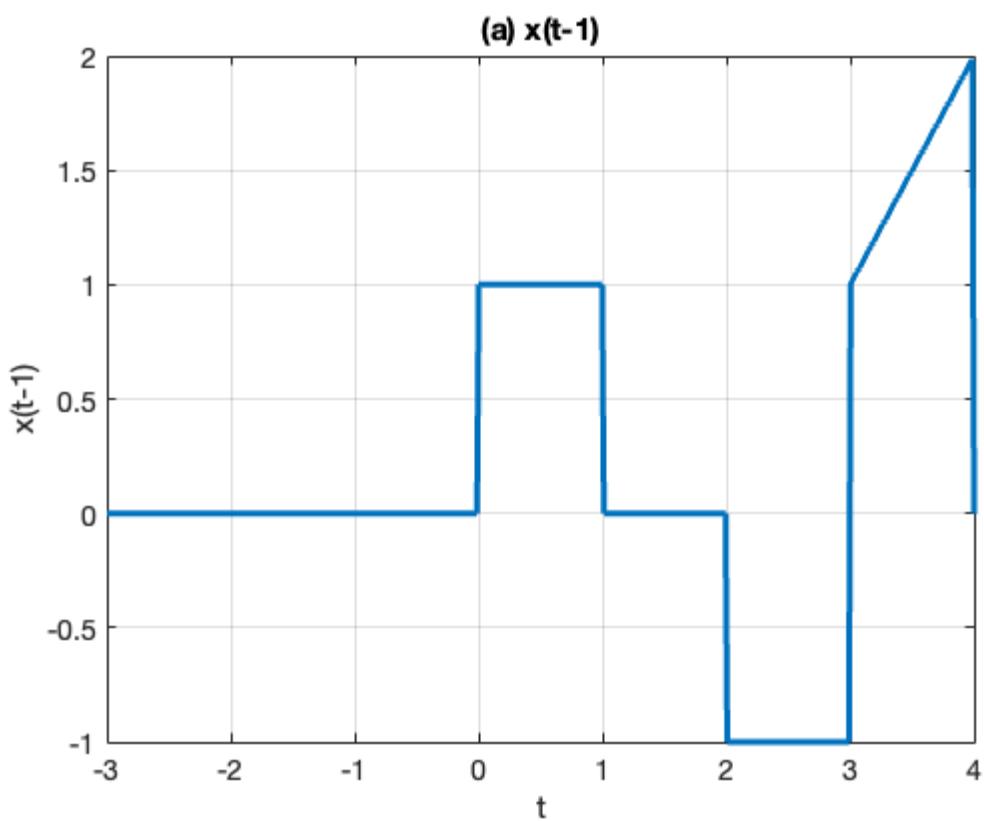
```

Put my functions here

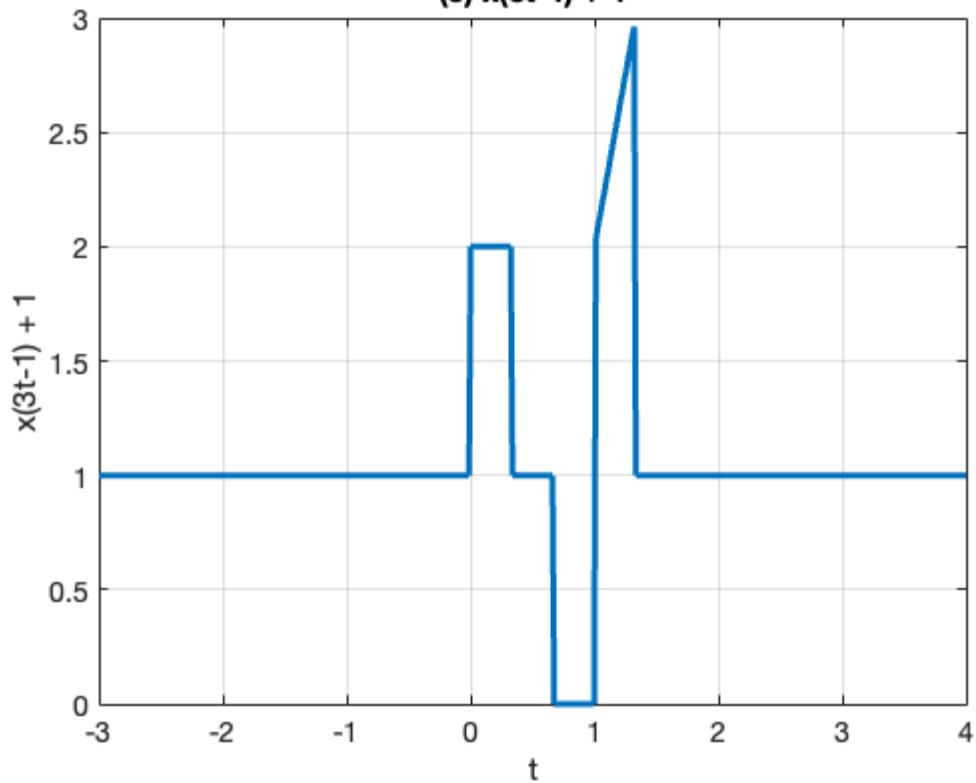
```

function my_ct_fun = ct_function(t)
% Function to compute x(t) based on defined piecewise intervals
my_ct_fun = zeros(size(t)); % Preallocate for efficiency
for c = 1:length(t)
    switch true % Use switch to handle intervals
        case t(c) < -1
            my_ct_fun(c) = 0;
        case t(c) >= -1 && t(c) < 0
            my_ct_fun(c) = 1;
        case t(c) >= 0 && t(c) < 1
            my_ct_fun(c) = 0;
        case t(c) >= 1 && t(c) < 2
            my_ct_fun(c) = -1;
        case t(c) >= 2 && t(c) < 3
            my_ct_fun(c) = t(c) - 1;
        case t(c) >= 3
            my_ct_fun(c) = 0;
    end
end
end

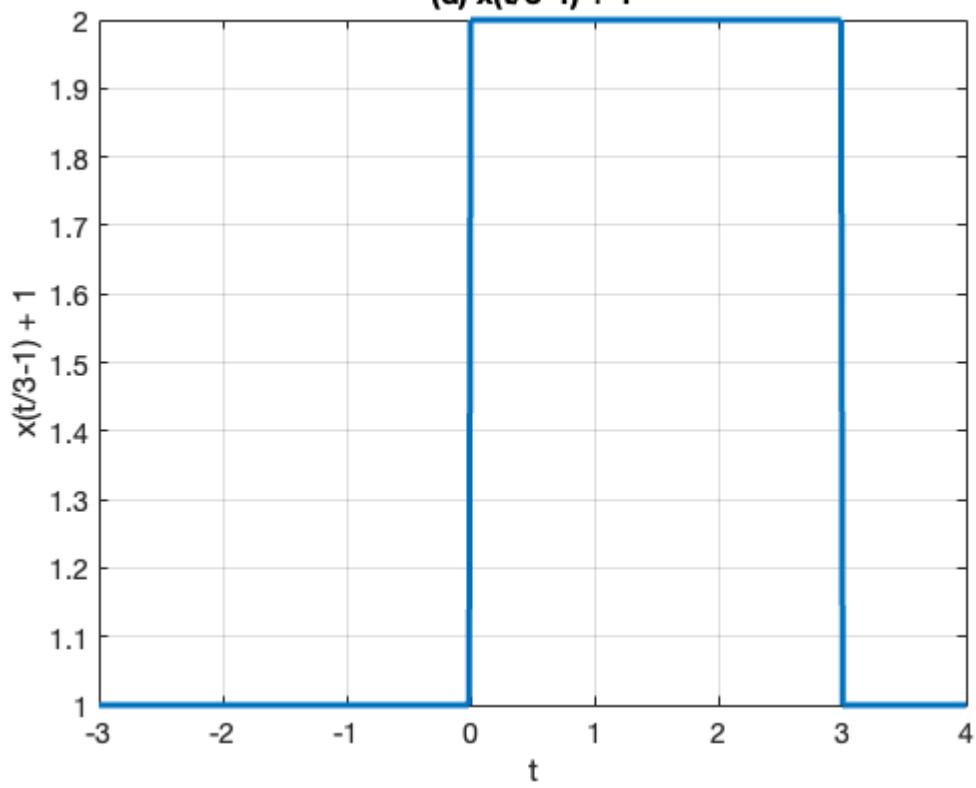
```



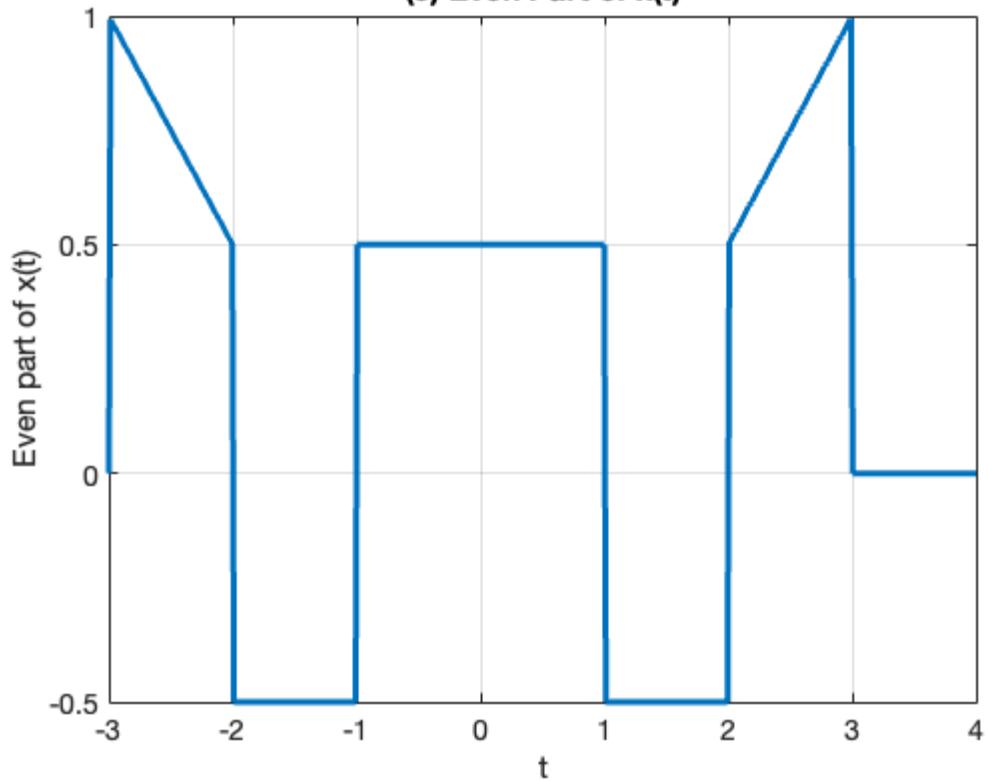
(c) $x(3t-1) + 1$



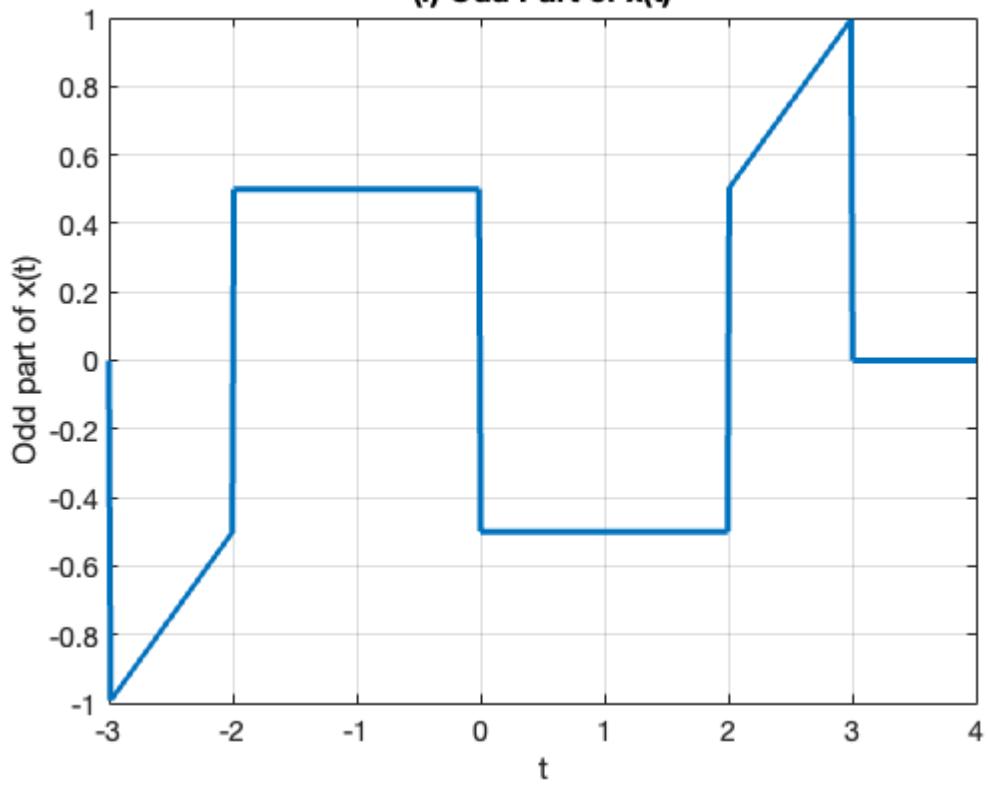
(d) $x(t/3-1) + 1$

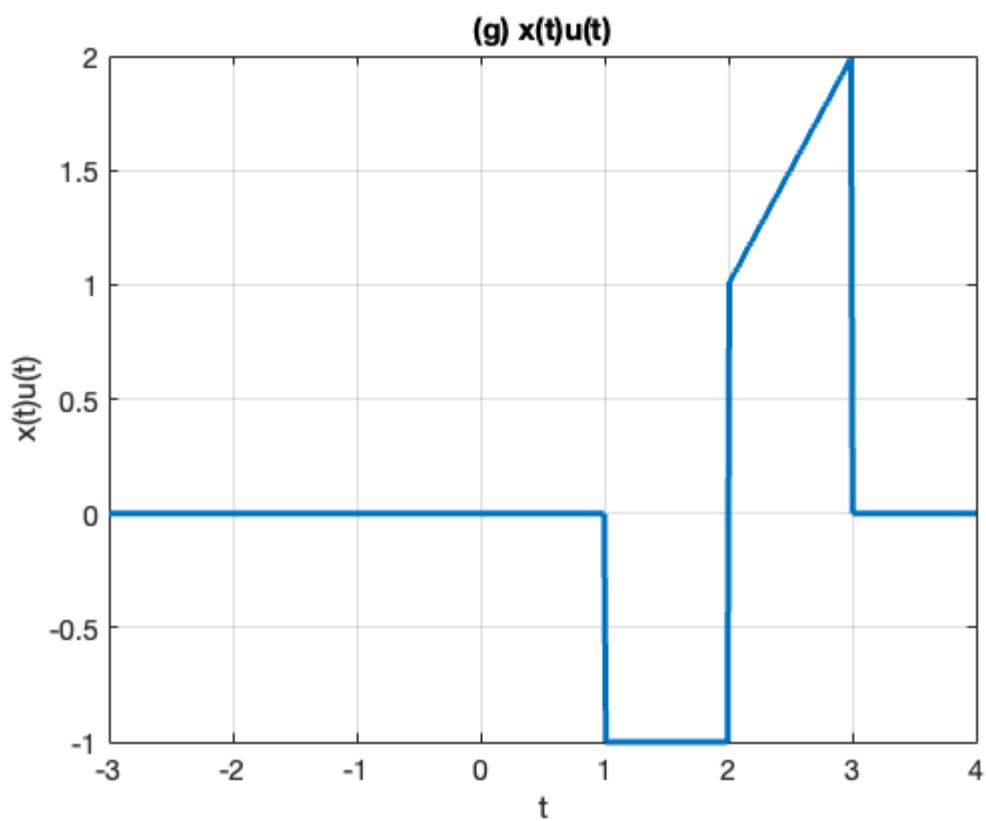


(e) Even Part of $x(t)$



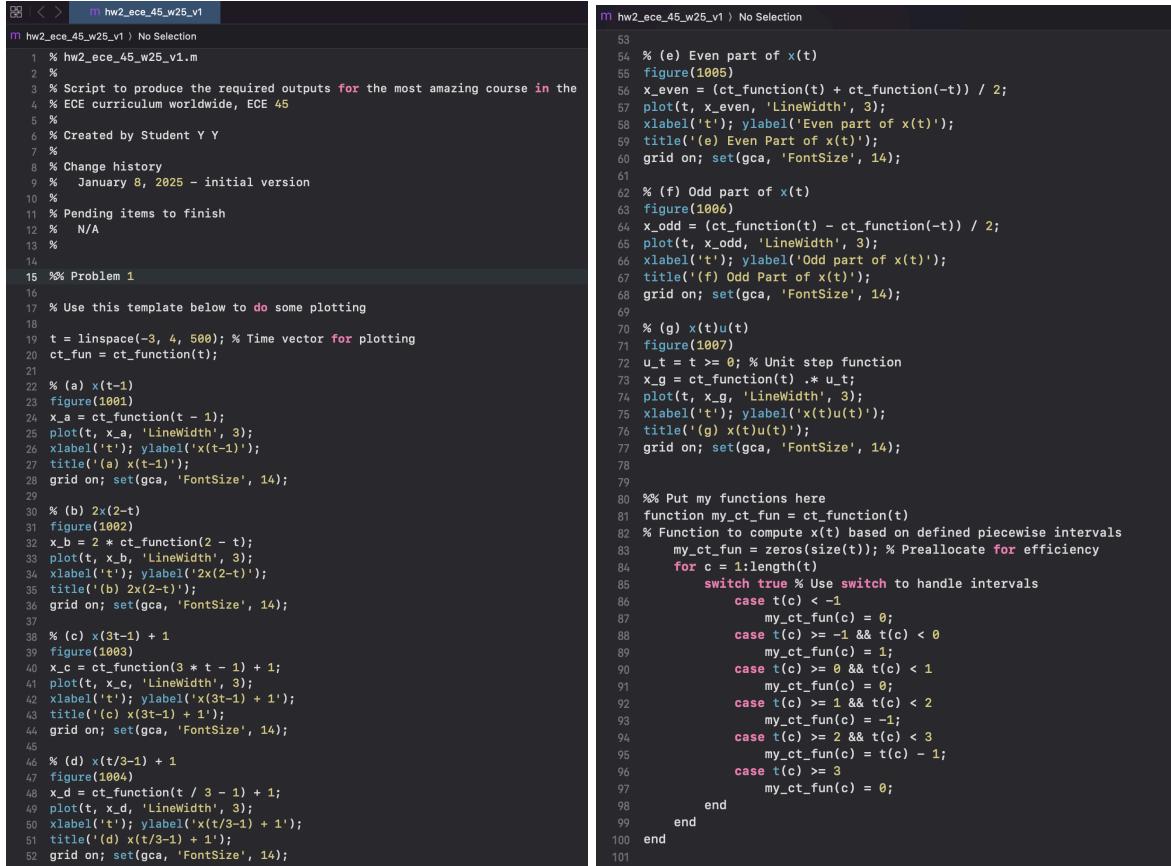
(f) Odd Part of $x(t)$





Published with MATLAB® R2024b

- 5) MATLAB live script - MATLAB was down so I included what should be the correct code
 (modified from the python notebook code I made to generate all of the plots above)



```

m hw2_ece_45_w25_v1 > No Selection
1 % hw2_ece_45_w25_v1.m
2 %
3 % Script to produce the required outputs for the most amazing course in the
4 % ECE curriculum worldwide, ECE 45
5 %
6 % Created by Student YY
7 %
8 % Change history
9 % January 8, 2025 - initial version
10 %
11 % Pending items to finish
12 % N/A
13 %
14 %% Problem 1
15
16 % Use this template below to do some plotting
17
18 t = linspace(-3, 4, 500); % Time vector for plotting
19 ct_fun = ct_function(t);
20
21 % (a) x(t-1)
22 figure(1001)
23 x_a = ct_function(t - 1);
24 plot(t, x_a, 'LineWidth', 3);
25 xlabel('t'); ylabel('x(t-1)');
26 title('(a) x(t-1)');
27 grid on; set(gca, 'FontSize', 14);
28
29 % (b) 2x(2-t)
30 figure(1002)
31 x_b = 2 * ct_function(2 - t);
32 plot(t, x_b, 'LineWidth', 3);
33 xlabel('t'); ylabel('2x(2-t)');
34 title('(b) 2x(2-t)');
35 grid on; set(gca, 'FontSize', 14);
36
37 % (c) x(3t-1) + 1
38 figure(1003)
39 x_c = ct_function(3 * t - 1) + 1;
40 plot(t, x_c, 'LineWidth', 3);
41 xlabel('t'); ylabel('x(3t-1) + 1');
42 title('(c) x(3t-1) + 1');
43 grid on; set(gca, 'FontSize', 14);
44
45 % (d) x(t/3-1) + 1
46 figure(1004)
47 x_d = ct_function(t / 3 - 1) + 1;
48 plot(t, x_d, 'LineWidth', 3);
49 xlabel('t'); ylabel('x(t/3-1) + 1');
50 title('(d) x(t/3-1) + 1');
51 grid on; set(gca, 'FontSize', 14);
52
53 % (e) Even part of x(t)
54 figure(1005)
55 x_even = (ct_function(t) + ct_function(-t)) / 2;
56 plot(t, x_even, 'LineWidth', 3);
57 xlabel('t'); ylabel('Even part of x(t)');
58 title('(e) Even Part of x(t)');
59 grid on; set(gca, 'FontSize', 14);
60
61 % (f) Odd part of x(t)
62 figure(1006)
63 x_odd = (ct_function(t) - ct_function(-t)) / 2;
64 plot(t, x_odd, 'LineWidth', 3);
65 xlabel('t'); ylabel('Odd part of x(t)');
66 title('(f) Odd Part of x(t)');
67 grid on; set(gca, 'FontSize', 14);
68
69 % (g) x(t)u(t)
70 figure(1007)
71 u_t = t >= 0; % Unit step function
72 x_g = ct_function(t) .* u_t;
73 plot(t, x_g, 'LineWidth', 3);
74 xlabel('t'); ylabel('x(t)u(t)');
75 title('(g) x(t)u(t)');
76 grid on; set(gca, 'FontSize', 14);
77
78
79 %% Put my functions here
80 function my_ct_fun = ct_function(t)
81 % Function to compute x(t) based on defined piecewise intervals
82 my_ct_fun = zeros(size(t)); % Preallocate for efficiency
83 for c = 1:length(t)
84 switch true % Use switch to handle intervals
85 case t(c) < -1
86     my_ct_fun(c) = 0;
87 case t(c) >= -1 && t(c) < 0
88     my_ct_fun(c) = 1;
89 case t(c) >= 0 && t(c) < 1
90     my_ct_fun(c) = 0;
91 case t(c) >= 1 && t(c) < 2
92     my_ct_fun(c) = -1;
93 case t(c) >= 2 && t(c) < 3
94     my_ct_fun(c) = 1;
95 case t(c) >= 3
96     my_ct_fun(c) = 0;
97 end
98 end
99 end
100 end

```

Homework 2

ECE 45 - Andrew Onozuka

Table of Contents

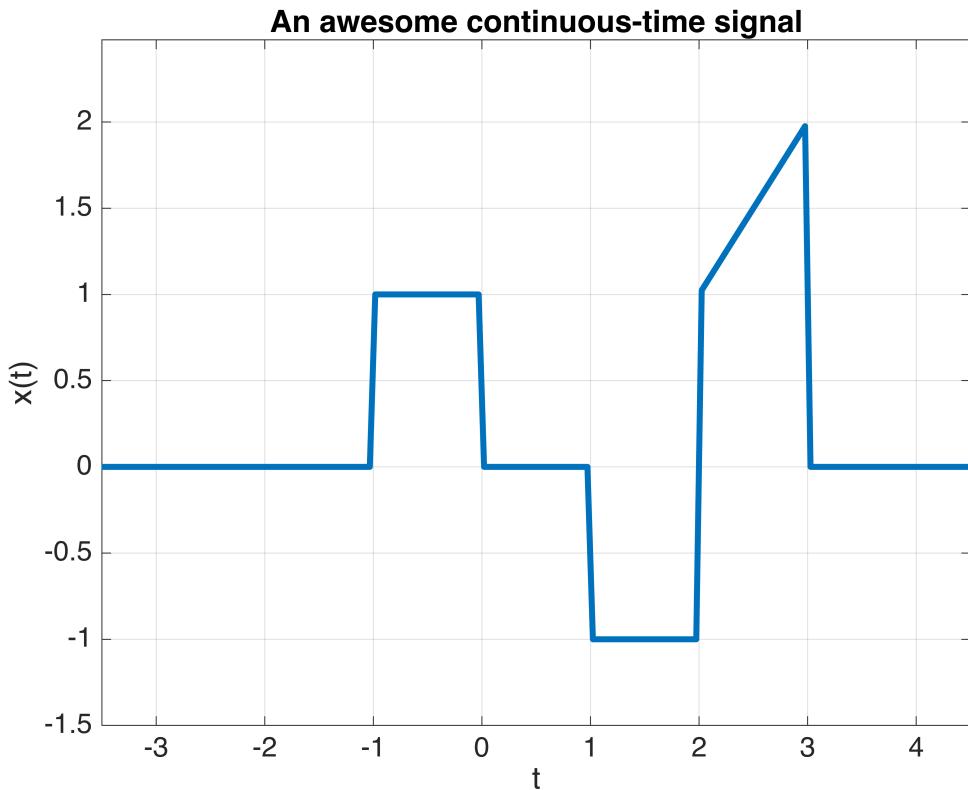
Problem 1.....	1
Part (a).....	2
Part (b).....	3
Part (c).....	4
Part (d).....	5
Part (e).....	6
Part (f).....	7
Part (g).....	8
Appendix - Functions.....	9

New text here

Problem 1

Consider the following CT signal .

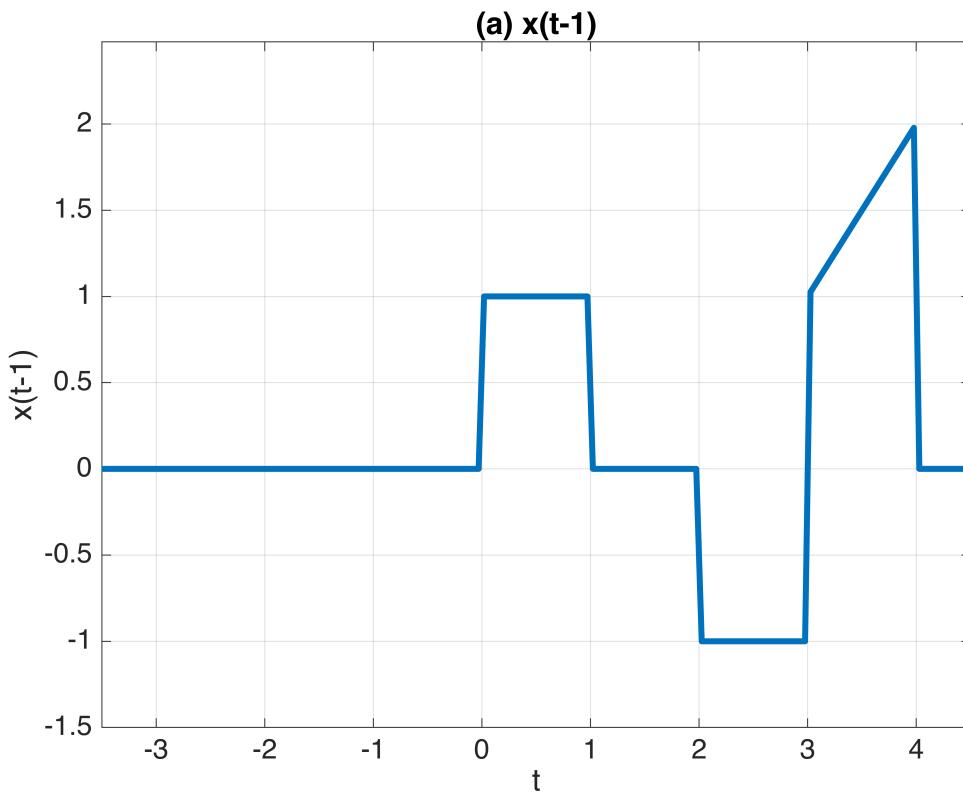
```
figure; clf;
t = linspace(-10, 15, 500);
ct_fun = ct_function(t);
plot(t, ct_fun, '-', 'LineWidth', 3);
xlabel('t');
ylabel('x(t)');
set(gca, 'FontSize', 14);
title('An awesome continuous-time signal');
xlim([-3.5, 4.5]);
ylim([min(ct_fun) - 0.5, max(ct_fun) + 0.5]);
grid on;
```



Part (a)

Plot $x(t - 1)$

```
% (a) x(t-1)
figure; clf;
x_a = ct_function(t - 1);
plot(t, x_a, '-', 'LineWidth', 3);
xlabel('t');
ylabel('x(t-1)');
set(gca, 'FontSize', 14);
title('(a) x(t-1)');
xlim([-3.5, 4.5]);
ylim([min(ct_fun) - 0.5, max(ct_fun) + 0.5]);
grid on;
```

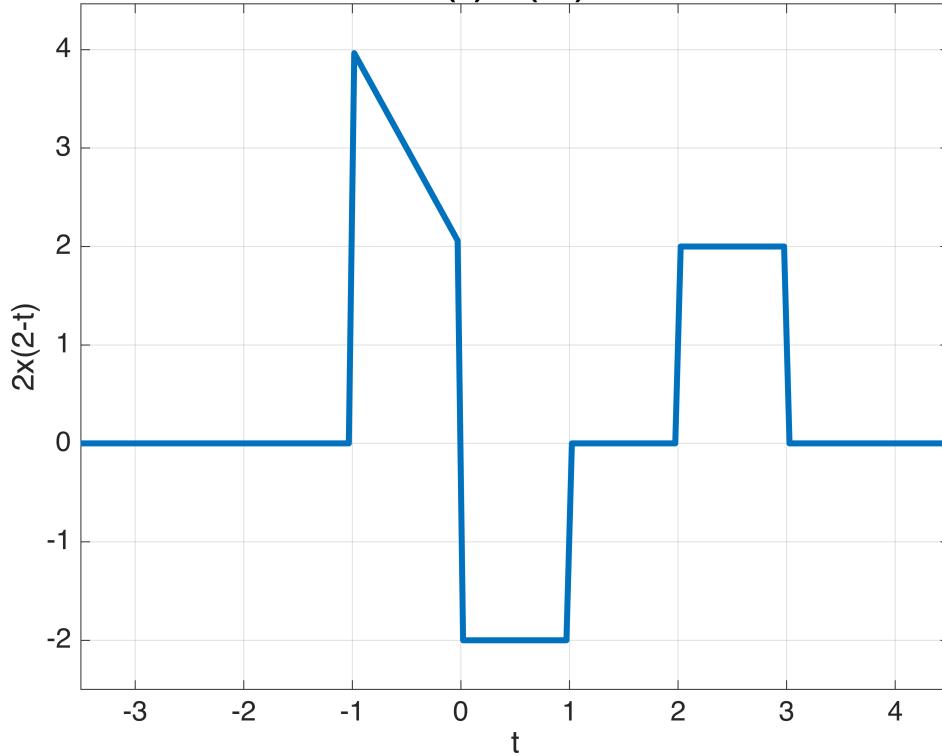


Part (b)

Plot $2x(2 - t)$

```
% (b) 2x(2-t)
figure; clf;
x_b = 2 * ct_function(2 - t);
plot(t, x_b, '-', 'LineWidth', 3);
xlabel('t');
ylabel('2x(2-t)');
set(gca, 'FontSize', 14);
title('(b) 2x(2-t)');
xlim([-3.5, 4.5]);
ylim([min(x_b) - 0.5, max(x_b) + 0.5]);
grid on;
```

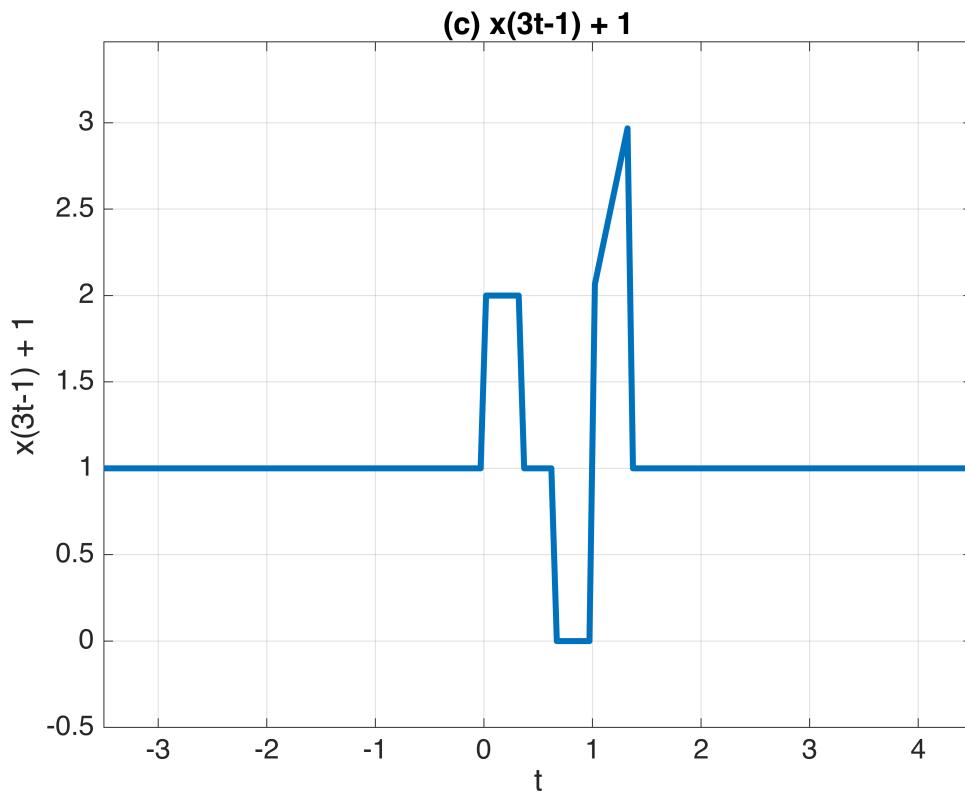
(b) $2x(2-t)$



Part (c)

Plot $x(3t - 1) + 1$

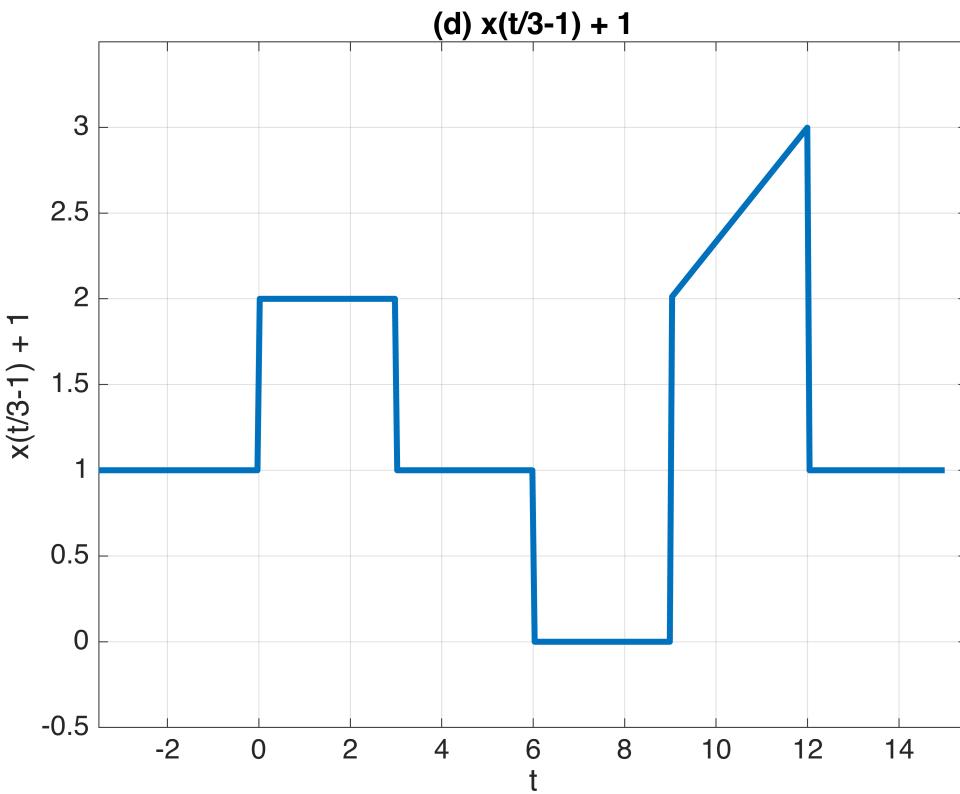
```
% (c) x(3t-1) + 1
figure; clf;
x_c = ct_function(3 * t - 1) + 1;
plot(t, x_c, '-', 'LineWidth', 3);
xlabel('t');
ylabel('x(3t-1) + 1');
set(gca, 'FontSize', 14);
title('(c) x(3t-1) + 1');
xlim([-3.5, 4.5]);
ylim([min(x_c) - 0.5, max(x_c) + 0.5]);
grid on;
```



Part (d)

Plot $x(t/3 - 1) + 1$

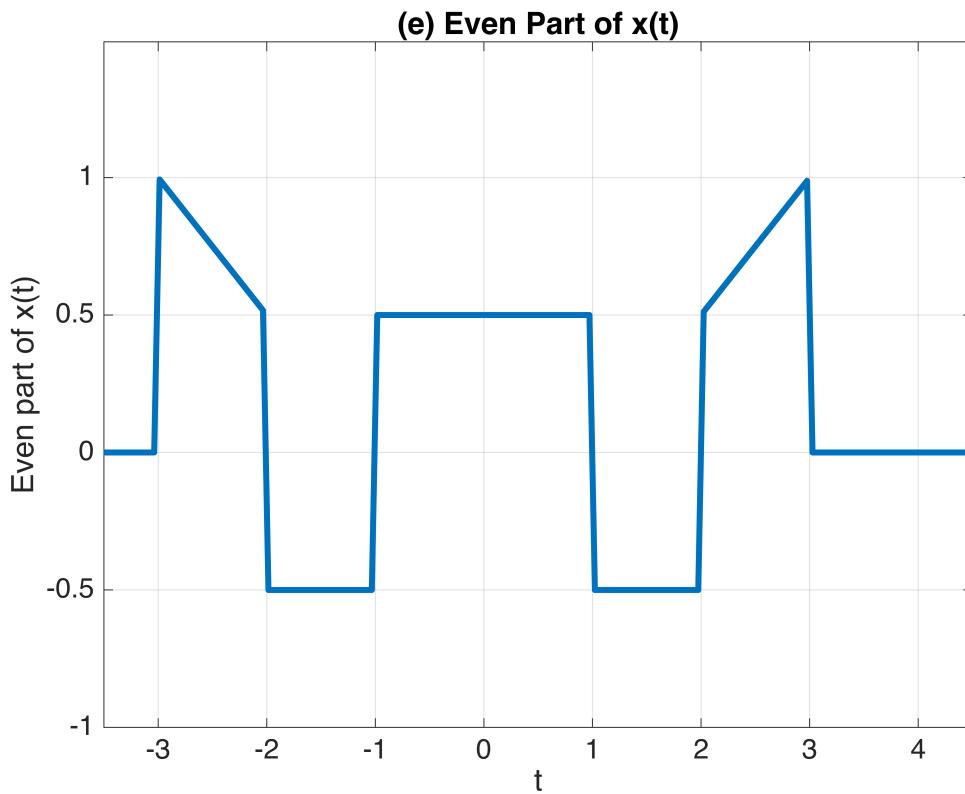
```
% (d) x(t/3-1) + 1
figure; clf;
x_d = ct_function(t / 3 - 1) + 1;
plot(t, x_d, '-', 'LineWidth', 3);
xlabel('t');
ylabel('x(t/3-1) + 1');
set(gca, 'FontSize', 14);
title('(d) x(t/3-1) + 1');
xlim([-3.5, 15.5]);
ylim([min(x_d) - 0.5, max(x_d) + 0.5]);
grid on;
```



Part (e)

Even part of $x(t)$

```
% (e) Even part of x(t)
figure; clf;
x_even = (ct_function(t) + ct_function(-t)) / 2;
plot(t, x_even, '-', 'LineWidth', 3);
xlabel('t');
ylabel('Even part of x(t)');
set(gca, 'FontSize', 14);
title('(e) Even Part of x(t)');
xlim([-3.5, 4.5]);
ylim([min(x_even) - 0.5, max(x_even) + 0.5]);
grid on;
```

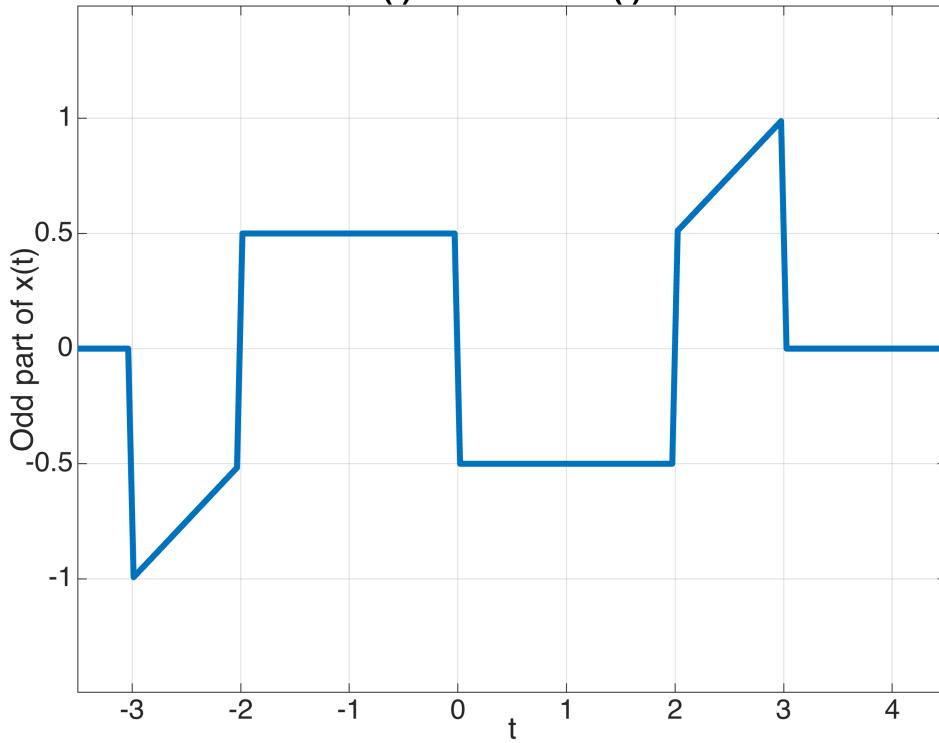


Part (f)

Odd part of $x(t)$

```
% (f) Odd part of x(t)
figure; clf;
x_odd = (ct_function(t) - ct_function(-t)) / 2;
plot(t, x_odd, '-', 'LineWidth', 3);
xlabel('t');
ylabel('Odd part of x(t)');
set(gca, 'FontSize', 14);
title('(f) Odd Part of x(t)');
xlim([-3.5, 4.5]);
ylim([min(x_odd) - 0.5, max(x_odd) + 0.5]);
grid on;
```

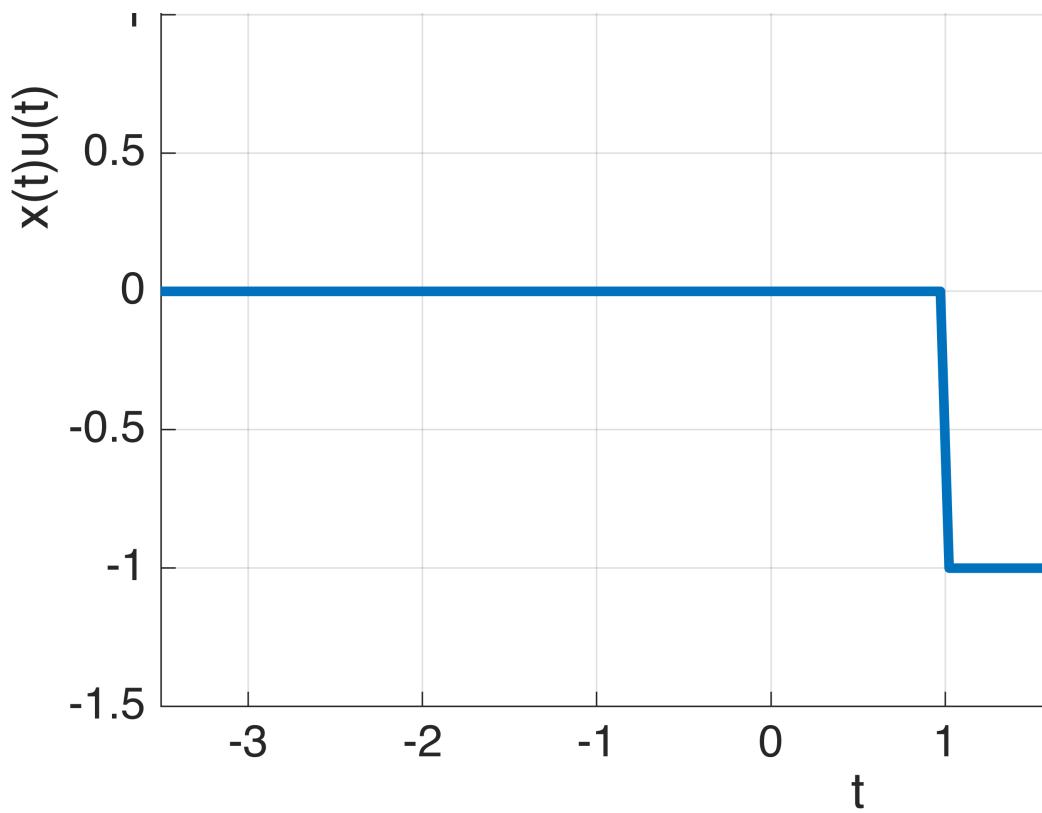
(f) Odd Part of $x(t)$



Part (g)

$x(t)u(t)$

```
% (g) x(t)u(t)
figure; clf;
u_t = t >= 0; % Unit step function
x_g = ct_function(t) .* u_t;
plot(t, x_g, '-', 'LineWidth', 3);
xlabel('t');
ylabel('x(t)u(t)');
set(gca, 'FontSize', 14);
title('(g) x(t)u(t)');
xlim([-3.5, 4.5]);
ylim([min(x_g) - 0.5, max(x_g) + 0.5]);
grid on;
```



Appendix - Functions

```

function my_ct_fun = ct_function(t)
    % Initialize the output variable
    my_ct_fun = zeros(size(t)); % Preallocate for efficiency

    % Iterate through each value of t
    for c = 1:length(t)
        switch true % Evaluate conditions for intervals
            case t(c) < -1
                my_ct_fun(c) = 0;
            case t(c) >= -1 && t(c) < 0
                my_ct_fun(c) = 1;
            case t(c) >= 0 && t(c) < 1
                my_ct_fun(c) = 0;
            case t(c) >= 1 && t(c) < 2
                my_ct_fun(c) = -1;
            case t(c) >= 2 && t(c) < 3
                my_ct_fun(c) = t(c) - 1;
            case t(c) >= 3
                my_ct_fun(c) = 0; % Explicitly set x(t) = 0 for t >= 3
        end
    end
end

```

```
    end  
end  
end
```

ECE 45 WI25 Professor Heath

January 16, 2025 11:59PM

Andrew Onozuka A16760043

Homework #2

6) Previous HW

HW1

● Graded

Select each question to review feedback and grading details.

Student

Andrew Onozuka

Total Points

100 / 100 pts

Question 1

Q1 40 / 40 pts

Question 2

Q2 40 / 40 pts

Question 3

Q3 10 / 10 pts

Question 4

Q4 10 / 10 pts