Principles of Communication

Systems Lab

Lab 3, 4th September 2019

Pratyush Nandi (IMT2017518)

Answer to Q1

Note: For question 1

u​c​(t) = Σ​n=1​N ​b​c​[n]p(t−n), u​s​(t) = Σ​n=1​N b​s​[n]p(t−n),

where p(t) = I​[0,1]​(t), m = 100, N = 100,

bc = [1,-1,1,-1,1,-1,1,-1,1,-1],

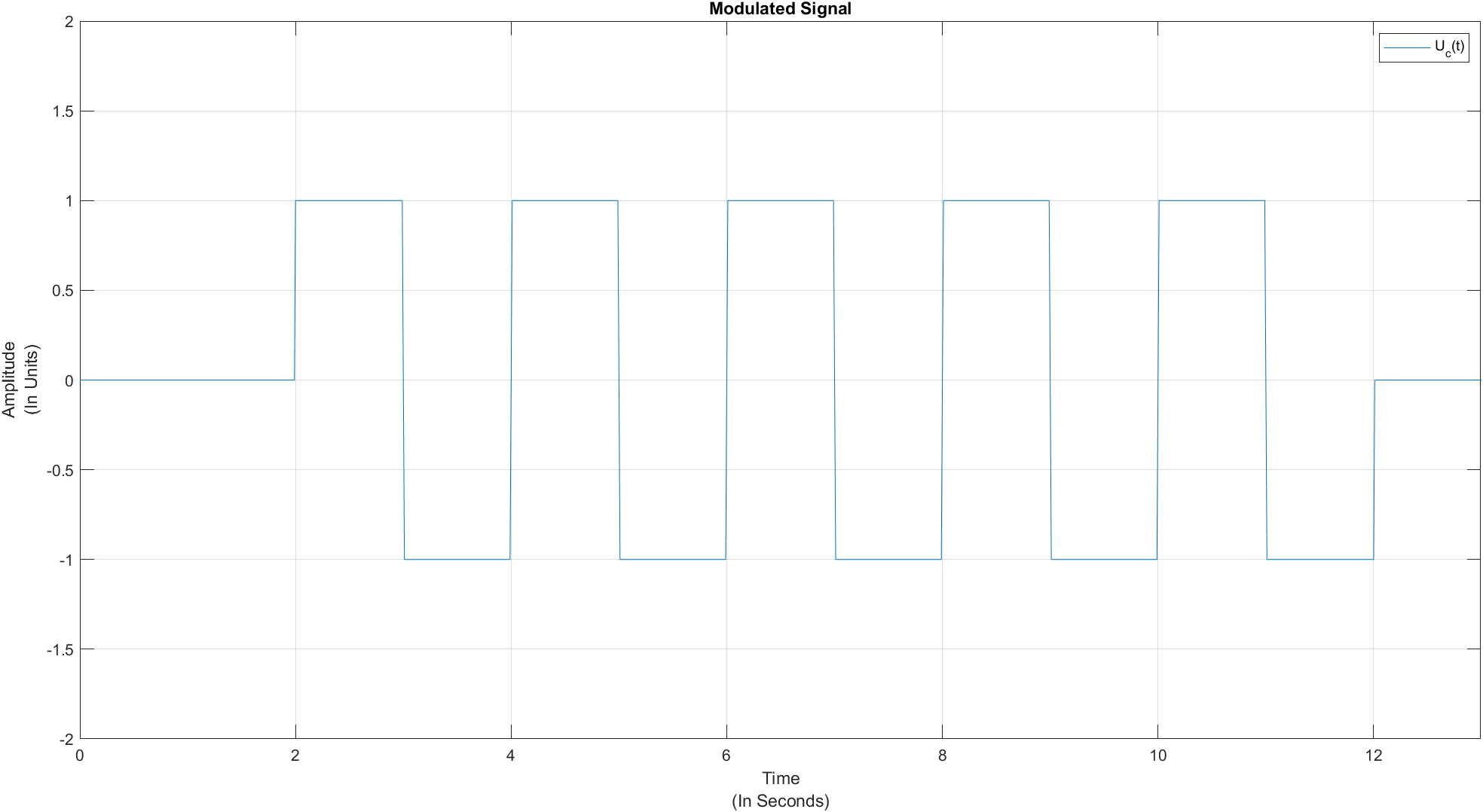
bs = [-1,1,-1,1,-1,1,-1,1,-1,1],

u​p,1​(t) = u​c​(t)cos40πt,

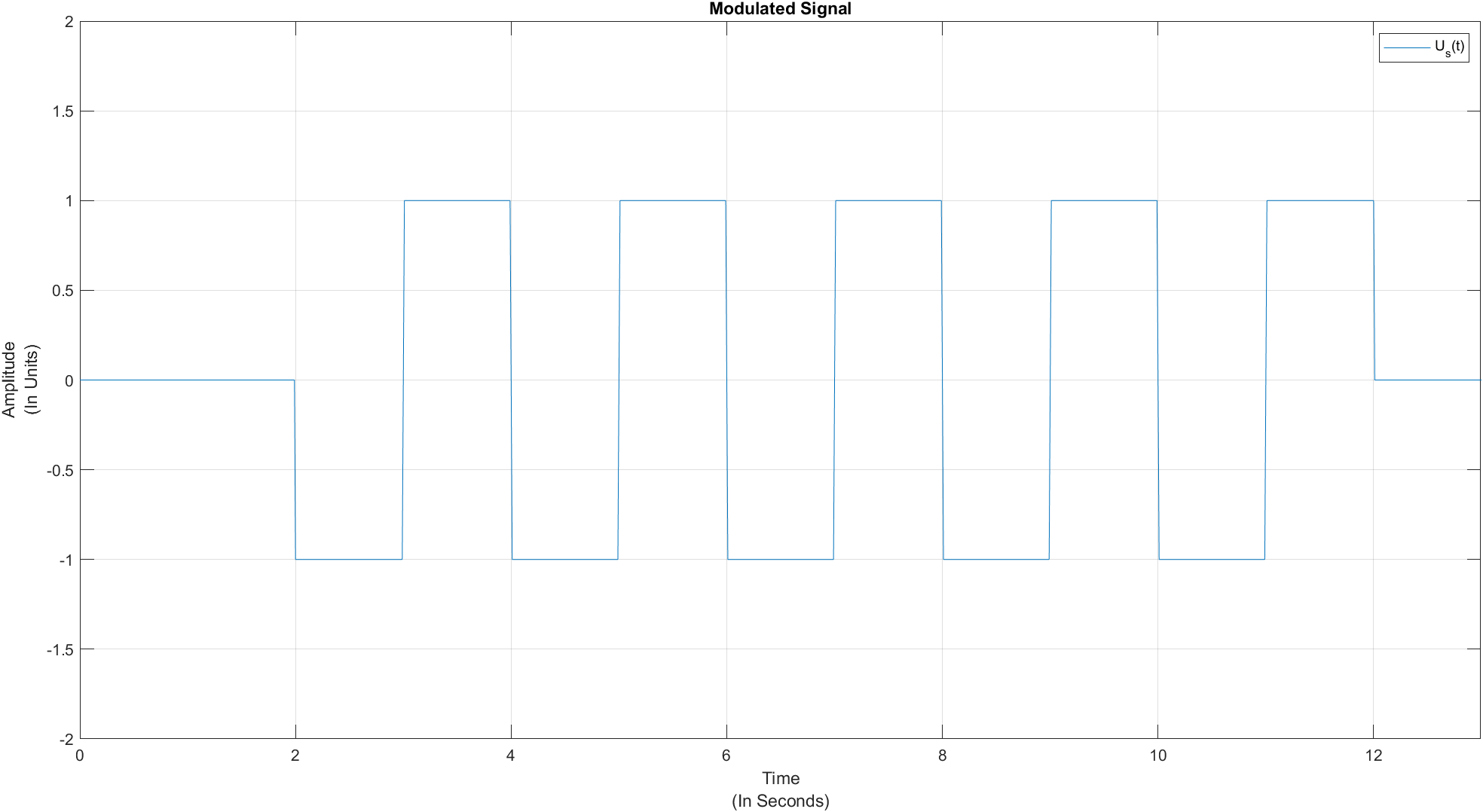
u​p​(t) = u​c​(t)cos40πt − u​s​(t) sin40πt

Unit time is in seconds and sampled at 1KHz frequency.

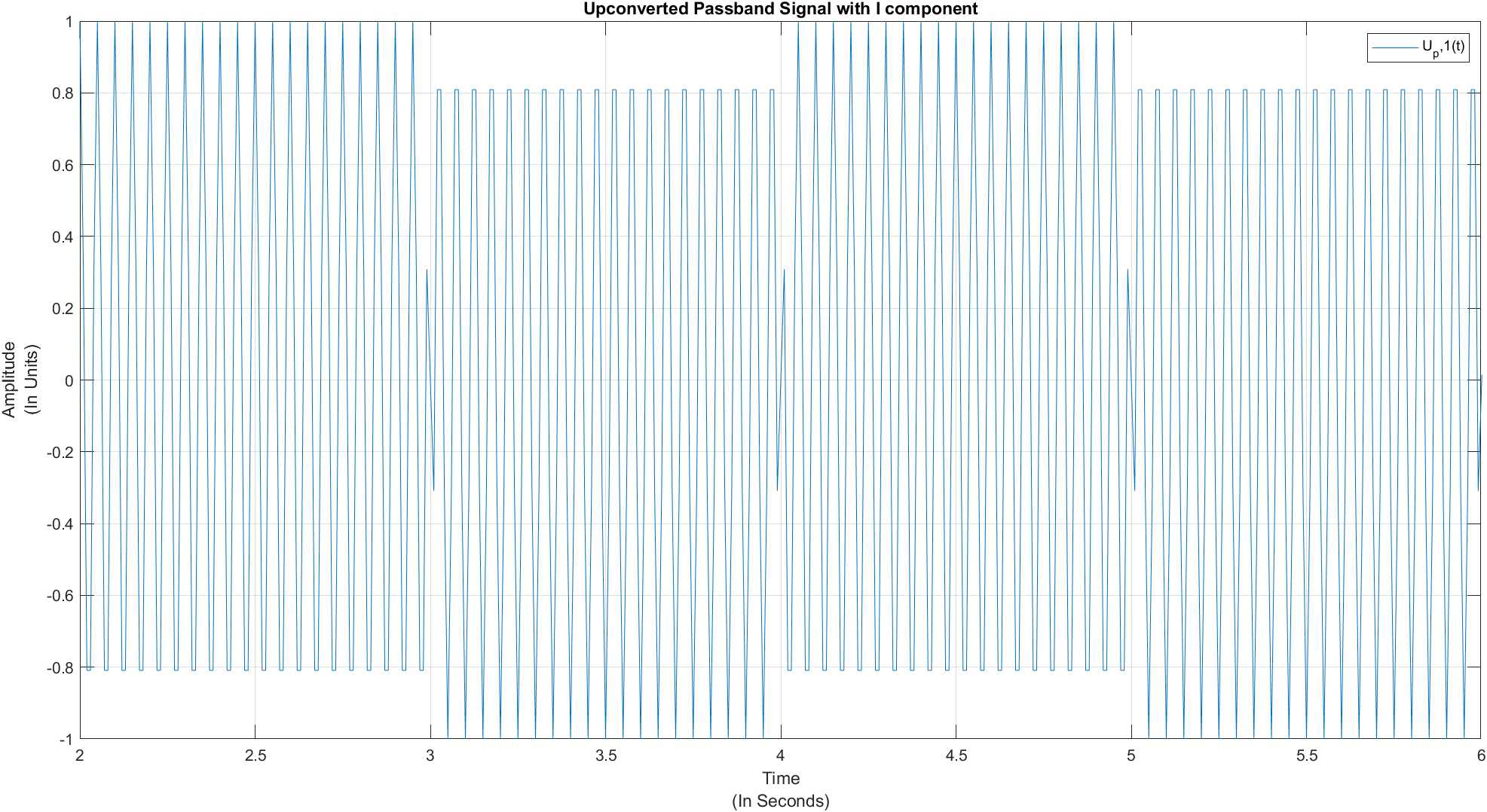
1.1.1:



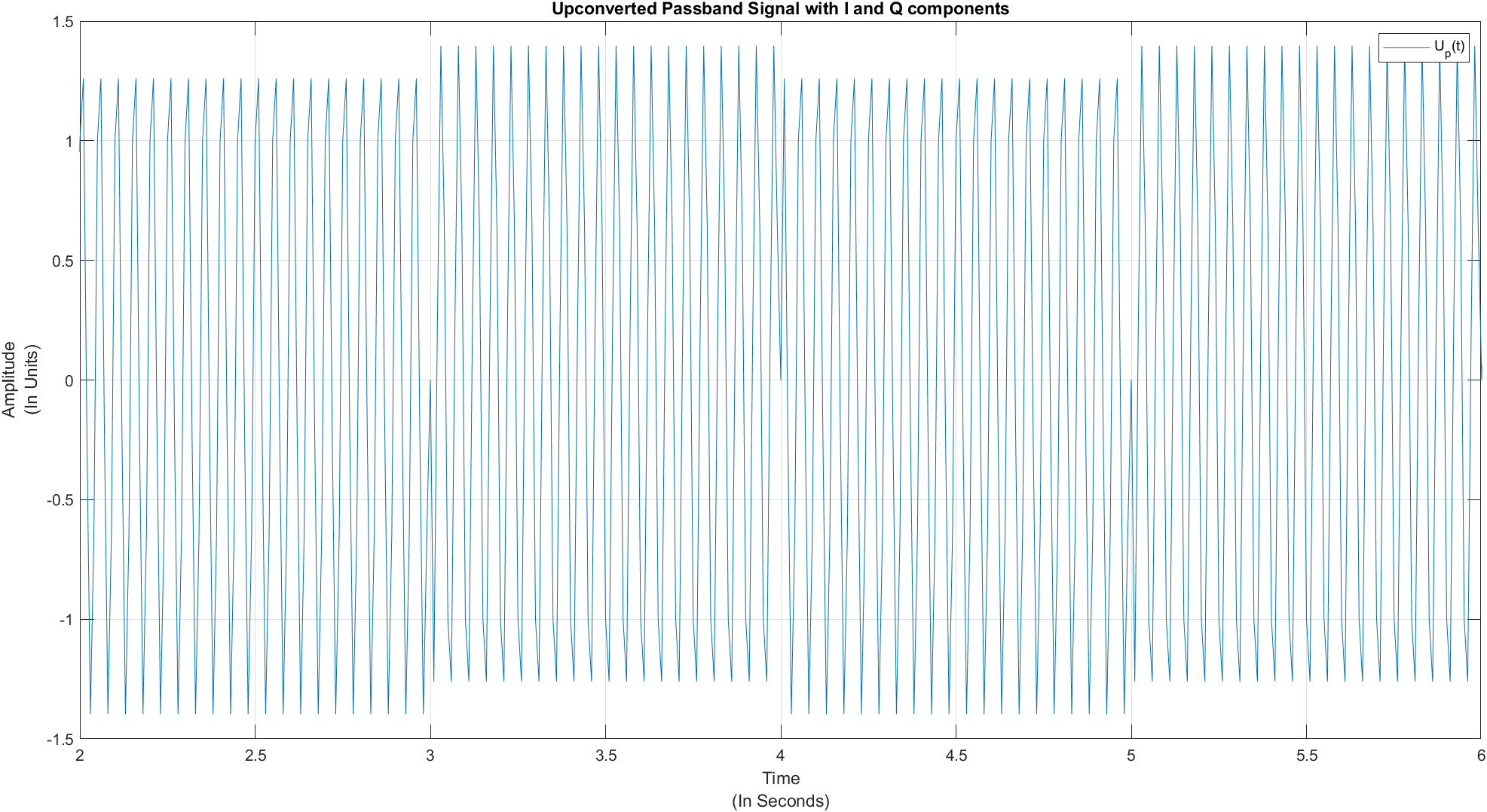
1.1.2:



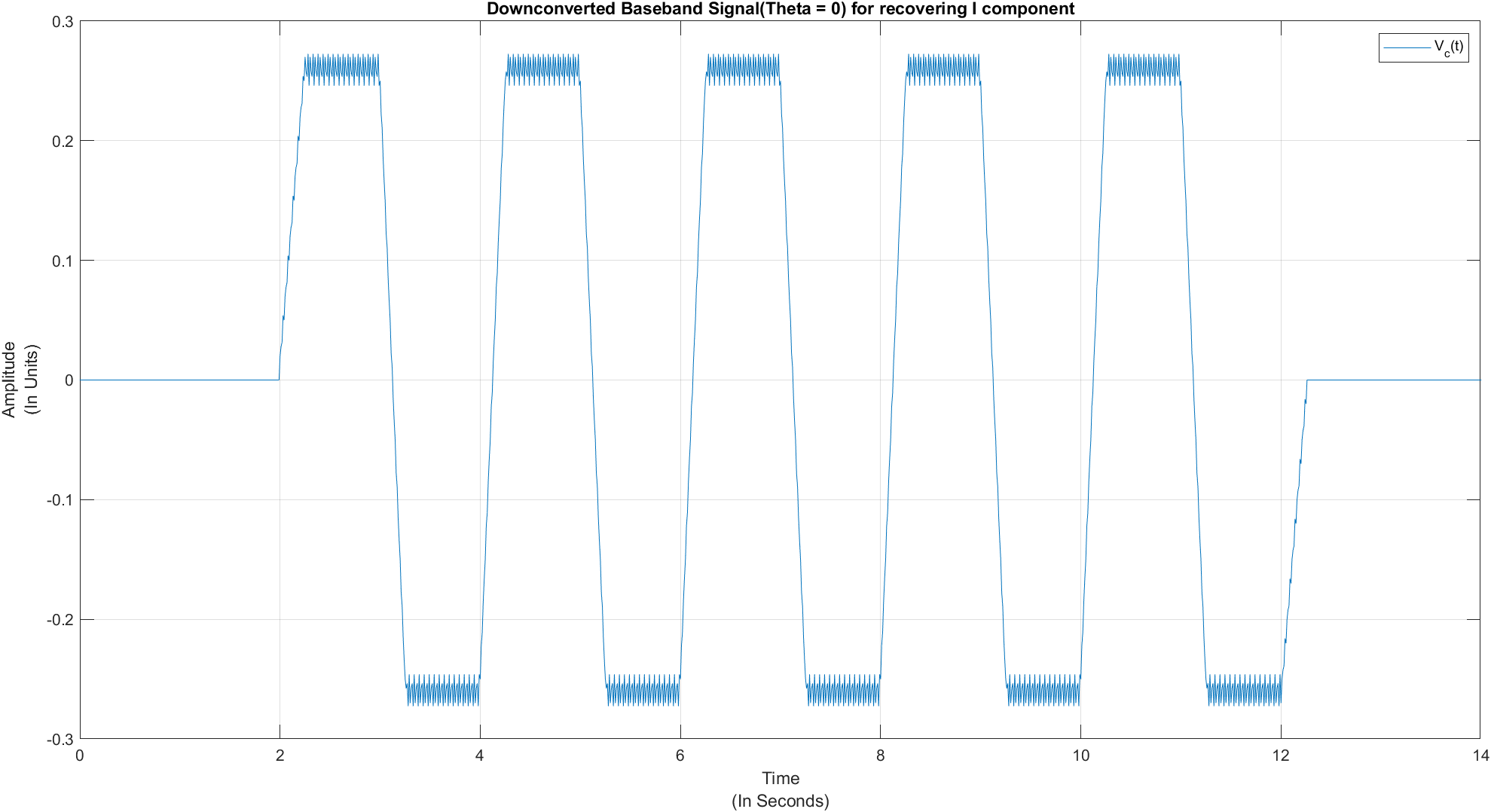
1.2:



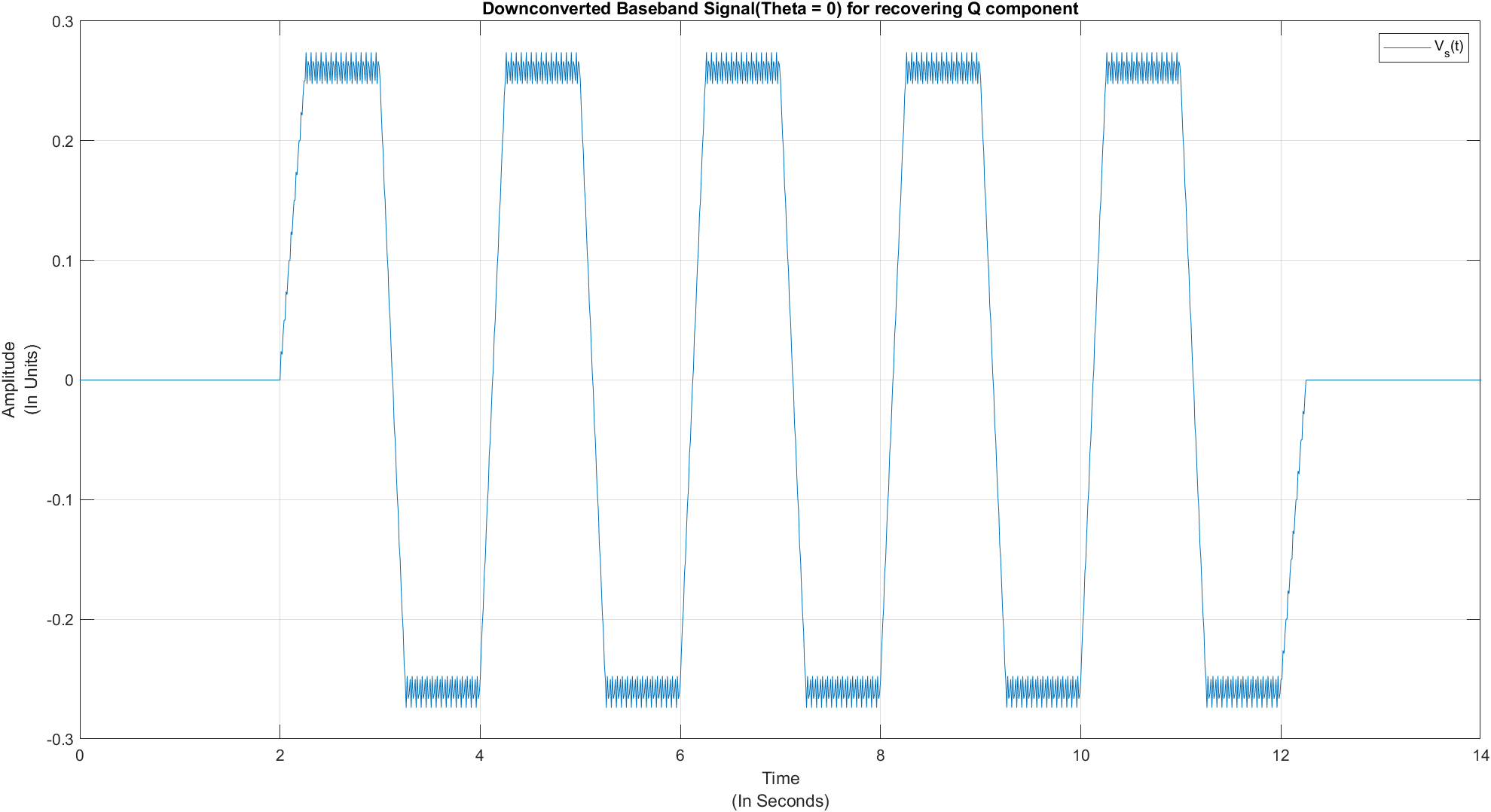
1.3:



1.4.1:

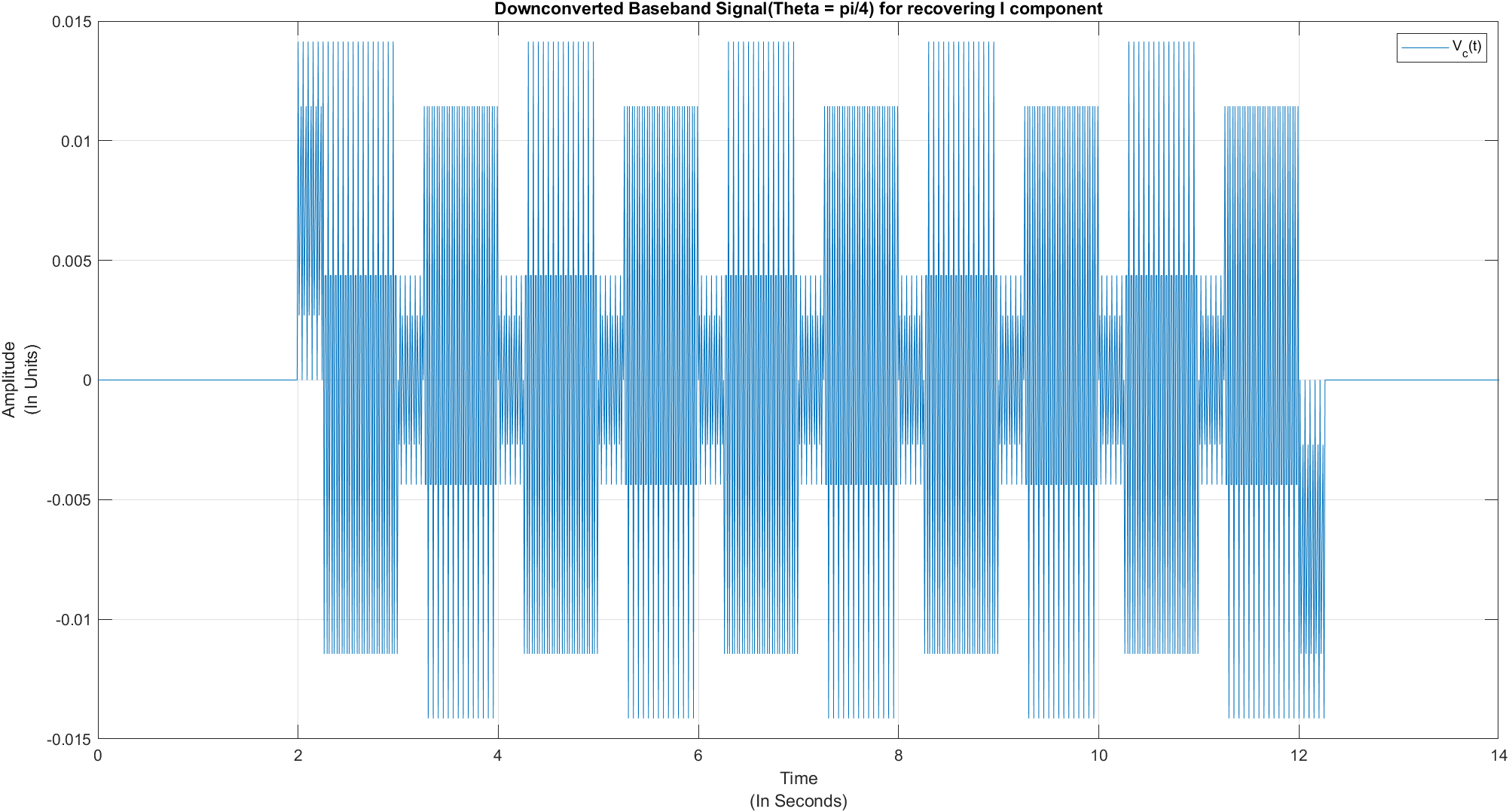


1.4.2:

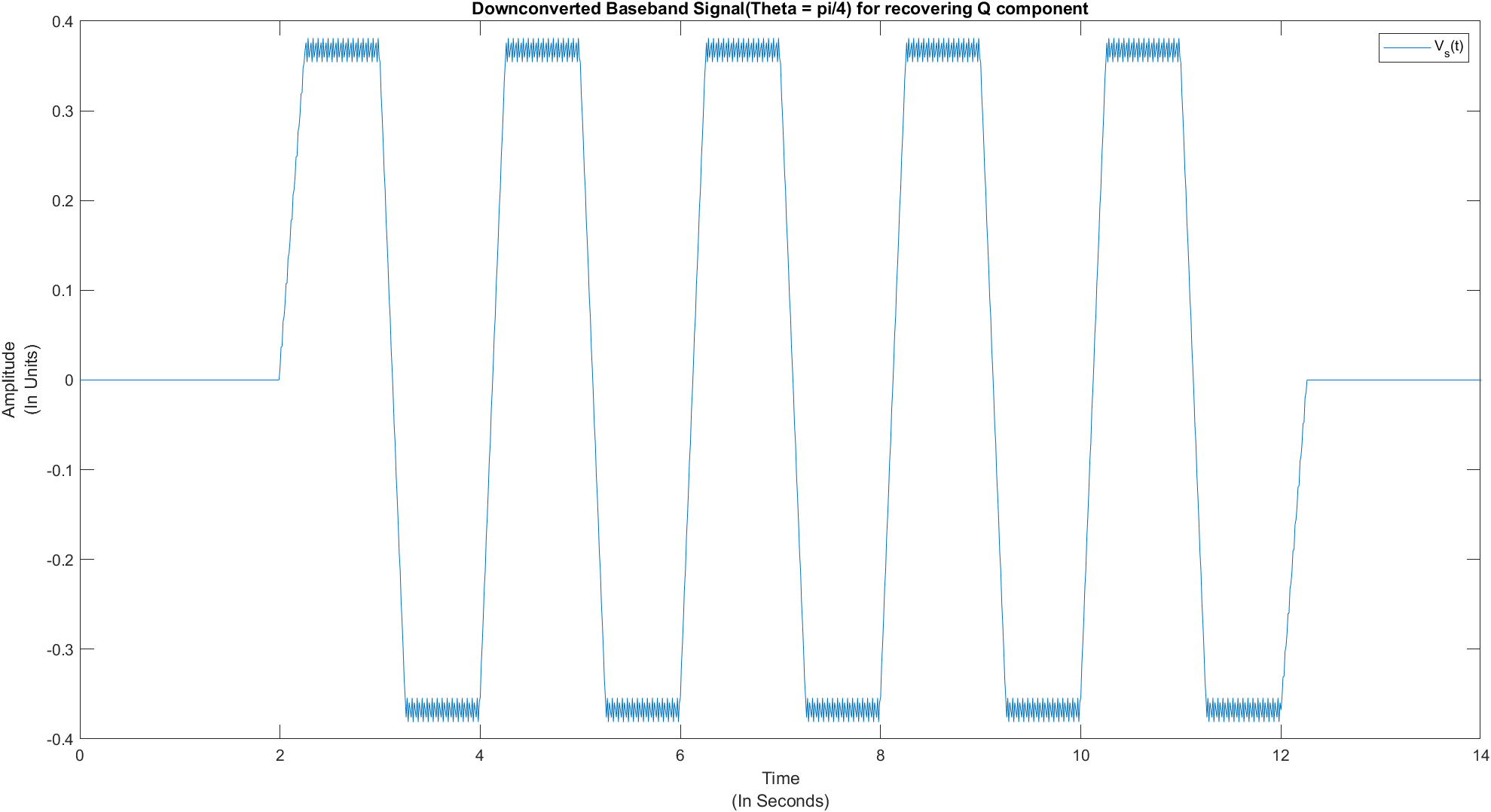


Ans: It is possible to read the graph by eyeballing. The graph shown in 1.4.2 is actually Us\*(-cos(0)) and the graph shown in 1.4.1 is actually Uc(cos(0)).

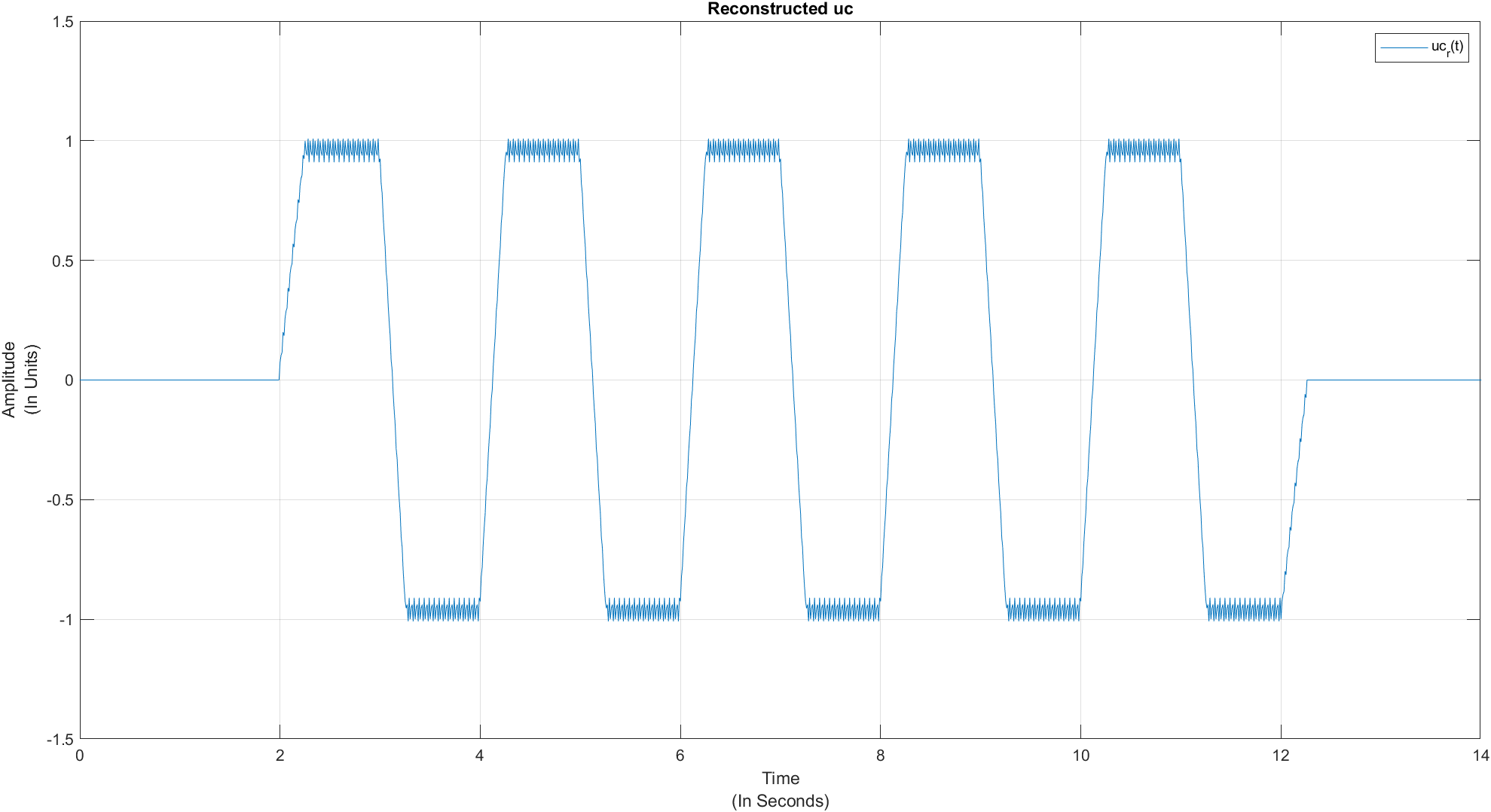
1.5.1:



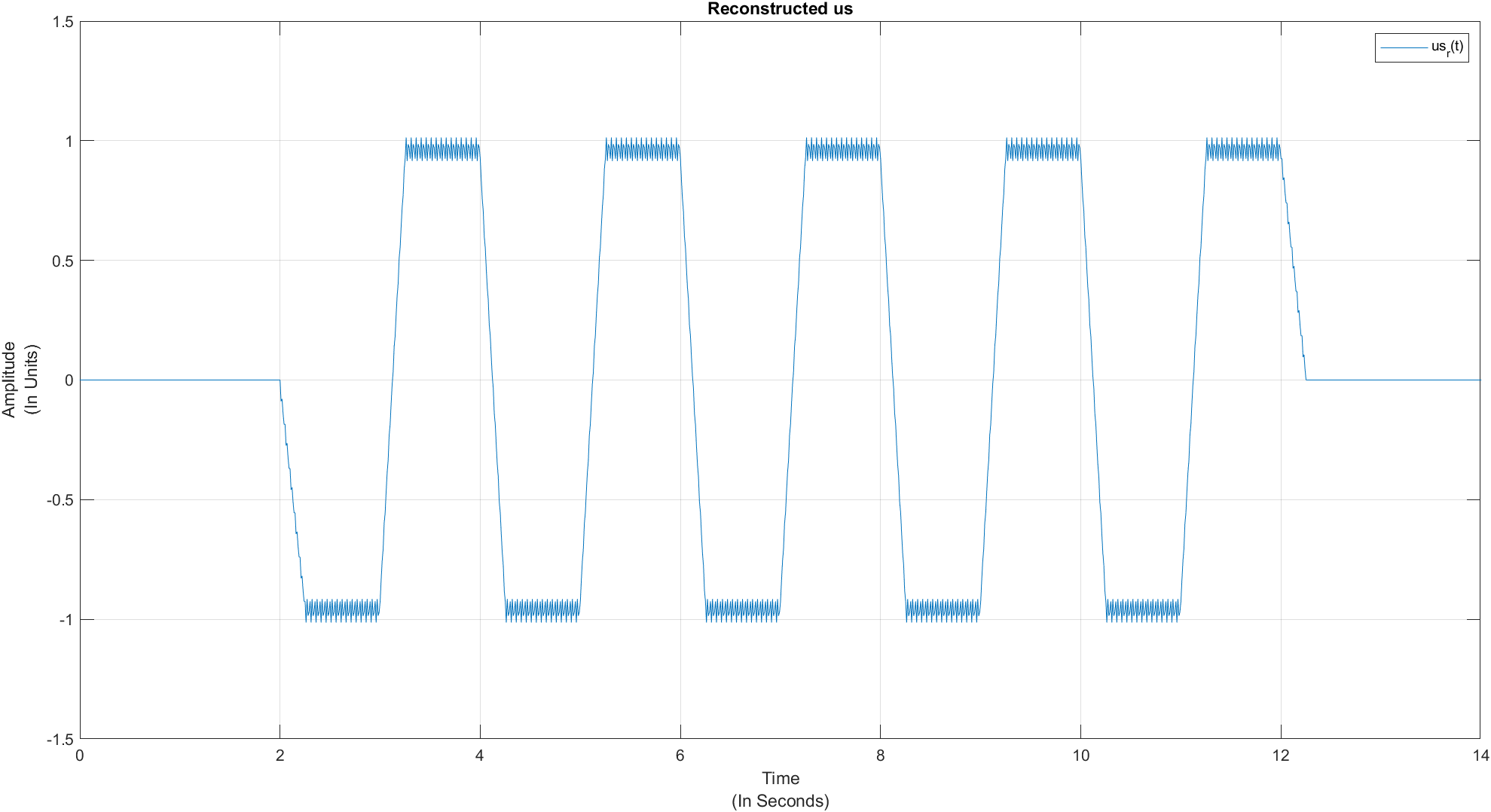
1.5.2:



1.6.1:



1.6.2:



Ans: 1.6.1 and 1.6.2 graphs represent the reconstructed version of the original Uc and Us signals.

To recunstruct the signal we use

Uc(bar) = Vc\*cos(theta= pi/4) + Vs\*sin(theta=pi/4)

Us(bar) = Vc\*cos(theta= pi/4) - Vs\*sin(theta=pi/4).

Appendix:

function upsampling

theta = pi/4;

m = 100;

N = 100;

dt = 1/m;

t = -2:dt:2;

p = signalx(t);

bc = [1,-1,1,-1,1,-1,1,-1,1,-1];

bs = [-1,1,-1,1,-1,1,-1,1,-1,1];

[uc,tc] = upsampler(p,bc,m,10);

[us,ts] = upsampler(p,bs,m,10);

figure(1);

plot(tc,uc);

xlim([0 13]);

ylim([-2 2]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Modulated Signal');

legend('U\_c(t)');

grid on;

figure(2);

plot(ts,us);

xlim([0 13]);

ylim([-2 2]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Modulated Signal');

legend('U\_s(t)');

grid on;

[uc1,tc1] = upsampler(p,bc,m,4);

[us1,ts1] = upsampler(p,bs,m,4);

up1 = uc1.\*cos(tc1.\*40\*pi);

figure(3);

plot(tc1,up1);

xlim([2 6]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Upconverted Passband Signal with I component');

legend('U\_p,1(t)');

grid on;

up\_1 = uc1.\*cos(tc1.\*40\*pi) - us1.\*sin(ts1.\*40\*pi);

up = uc.\*cos(tc.\*40\*pi) - us.\*sin(ts.\*40\*pi);

figure(4);

plot(tc1,up\_1);

xlim([2 6]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Upconverted Passband Signal with I and Q components');

legend('U\_p(t)');

grid on;

h = lowpassfilter(t);

[udc,tdc] = contconv(double(2\*up.\*(cos(tc.\*40\*pi))),double(h),tc(1),t(1),dt);

figure(5);

plot(tdc,udc);

xlim([0 14]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Downconverted Baseband Signal(Theta = 0) for recovering I component');

legend('V\_c(t)');

grid on;

[uds,tds] = contconv(double(2\*up.\*(sin(ts.\*40\*pi))),double(h),ts(1),t(1),dt);

figure(6);

plot(tds,uds);

xlim([0 14]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Downconverted Baseband Signal(Theta = 0) for recovering Q component');

legend('V\_s(t)');

grid on;

[udco,tdco] = contconv(double(2\*up.\*(cos(tc.\*40\*pi+theta))),double(h),tc(1),t(1),dt);

figure(7);

plot(tdco,udco);

xlim([0 14]);

grid on;

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Downconverted Baseband Signal(Theta = pi/4) for recovering I component');

legend('V\_c(t)');

[udso,tdso] = contconv(double(2\*up.\*(sin(ts.\*40\*pi+theta))),double(h),ts(1),t(1),dt);

figure(8);

grid on;

plot(tdso,udso);

xlim([0 14]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Downconverted Baseband Signal(Theta = pi/4) for recovering Q component');

legend('V\_s(t)');

uc\_ = (udco\*cos(pi/4) + udso\*sin(pi/4))\*3.7;

us\_ = (udco\*sin(pi/4) - udso\*sin(pi/4))\*3.7;

figure(9);

plot(tdco,uc\_);

xlim([0 14]);

grid on;

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Reconstructed uc');

legend('uc\_r(t)');

figure(10);

plot(tdso,us\_);

grid on;

xlim([0 14]);

xlabel({'Time','(In Seconds)'});ylabel({'Amplitude','(In Units)'});

title('Reconstructed us');

legend('us\_r(t)');

end

function [y,t] = contconv(x1,x2,s1,s2,dt)

y = conv(x1,x2)\*dt;

s1\_2 = s1 + (length(x1)-1)\*dt;

s2\_2 = s2 + (length(x2)-1)\*dt;

t1 = s1+ s2;

t2 = s2\_2 + s1\_2;

t = t1:dt:t2;

end

function [s,t] = upsampler(p,bc,m,N)

nsymbols = length(bc);

nsymbols\_upsampled = 1 + (nsymbols - 1) \* m;

symbols\_upsampled = zeros(nsymbols\_upsampled, 1);

symbols\_upsampled(1:m:nsymbols\_upsampled) = bc;

s1 = conv(symbols\_upsampled,double(p));

s = s1.';

t = 0:1/m:(length(s) - 1)/m;

end

function u = signalx(t)

syms x;

y = piecewise(0 <= x <= 1, 1, 0);

u = subs(y,x,t);

end

function u = lowpassfilter(t)

syms x;

y = piecewise(0 <= x <= 0.25, 1, 0);

u = subs(y,x,t);

end