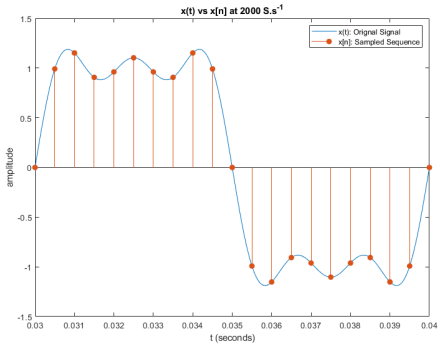
ECEN 220
Lab Report 5 : Upsampling

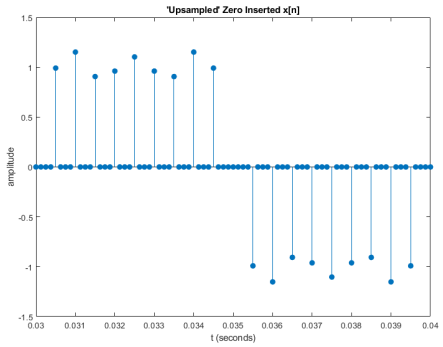
Daniel Eisen : 300447549

October 10, 2019

1 Sampling and Zero Padding



(a) signal vs sampled



(b) Zero Padded

Step one is to sample a CT signal, in this case x(t) is shown in figure 1 to be sampled at 2000 samples a second. To begin up sampling, the sequence x[n] is zero padded to create a sequence at 8000 samples a second, shown figure 2.

2 Filter Response Generation

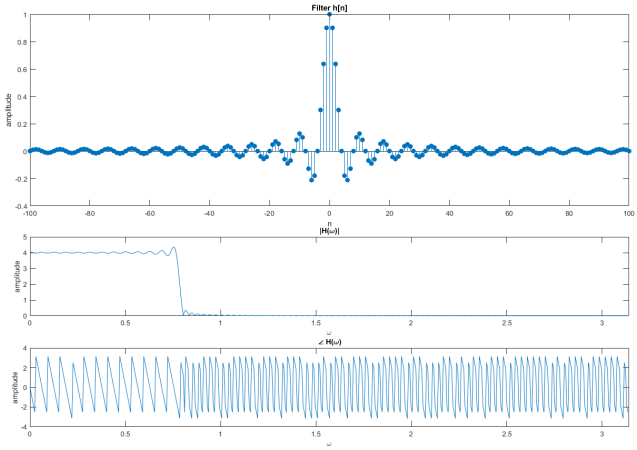


Figure 2: Filter and frequency components

A sinc is generated for use in low=pass filtering the zero padded sequence for up sampling. The magnitude plot shows the band of allowed frequencies.

3 Low Pass Filtering

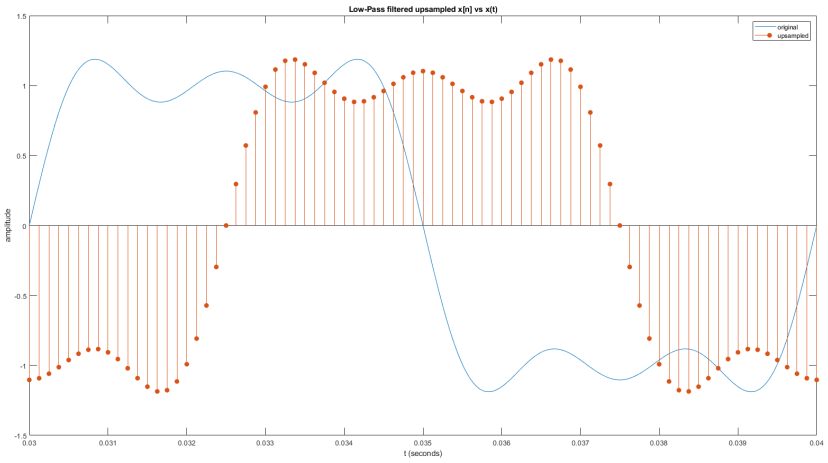


Figure 3: Filtered and Reconstructed signal and original signal

Upon filtering, figure 3 shows an up sampled sequence that appears to be accurate in form to the original signal, it is however delayed by the length of the filter, so does not match in phase.

4 Delay Reconstruction

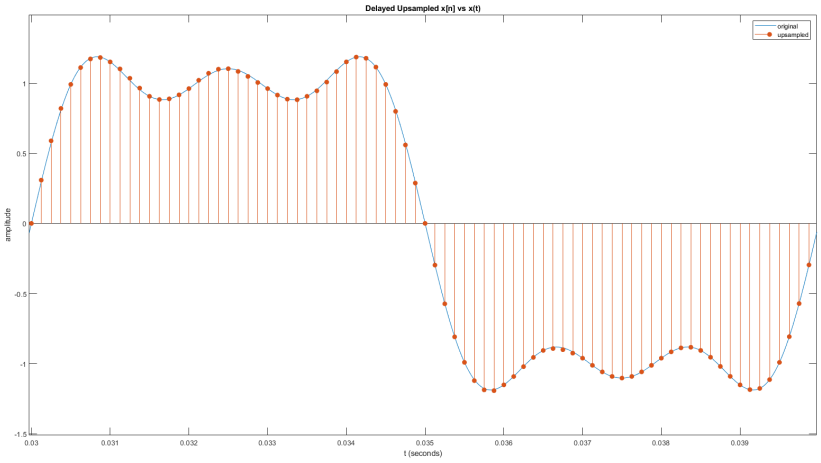


Figure 4: Filtered and Reconstructed signal and original signal

Figure 5 shows, that upon delaying the sequence by a further filter length, the overlaid signal and sequence are shown to be very close in form, so the original samples have been successfully up sampled.

5 Delaying Effect of Filtering

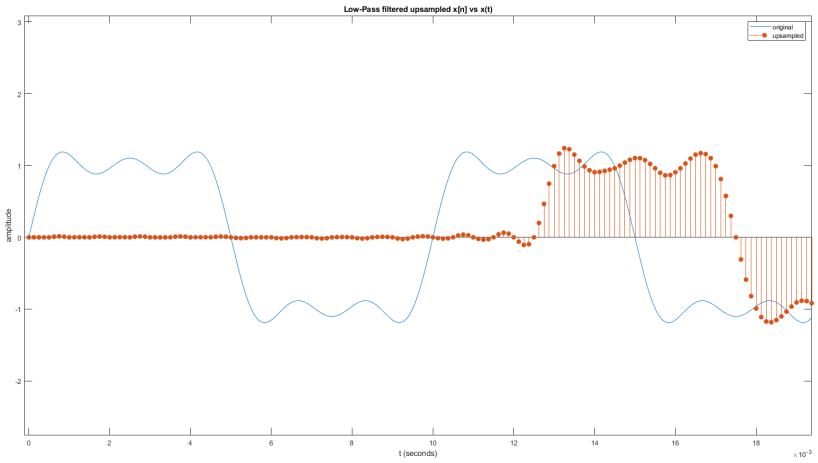


Figure 5: Introduced delay

The previously shown figure focused on 30ms - 40ms of the signal. This is due to what figure 6 shows above; the induced filtering delay and warping at the beginning of the sequence.

Appendix

```
clear
clc

f0 = 100;
Q = 2;
M = 4;
L = 100;
fs1 = 2000;
fs2 = 8000;
tmin = 30/1000;
tmax = 40/1000;

t = 0:0.00001:1;
t1 = 0:1/fs1:1;
t2 = 0:1/fs2:1;
n = -L:L;

x = zeros(Q+1, length(t)); %orignial signal
xs1 = zeros(Q+1, length(t1)); %x sampled at 2000Hz

% populate matrix
for q=0:Q
    x(q+1 , :) = (sin(2*pi*(2*q+1)*f0.*t))...
                        ./(2*q+1);

    xs1(q+1 , :) = (sin(2*pi*(2*q+1)*f0.*t1))...
                        ./(2*q+1);
end

%% 1

% define sums
x = 4./pi.*sum(x);
xs1 = 4./pi.*sum(xs1);

figure(1); % x vs x samples
plot(t,x)
hold on;
stem(t1,xs1, "filled")
xlim([tmin tmax]);
hold off;
title("x(t) vs x[n] at 2000 S.s^{-1}")
legend("x(t): Orignal Signal", "x[n]: Sampled Sequence");
xlabel("t (seconds)")
ylabel("amplitude")


%% 2
figure(2);
xs2 = upsample(xs1, M);
xs2 = xs2(1:8001);
stem(t2, xs2, "filled")
xlim([tmin tmax]);
title("'Upsampled' Zero Inserted x[n]")
xlabel("t (seconds)")
ylabel("amplitude")

%% 3
figure(3)
subplot(4,1,[1 2]);
```

```
h = M.*((sin(pi.*n./M))./(pi.*n));
h(101) = 1;
stem(n,h, "filled")
title("Filter h[n]")
xlabel("n")
ylabel("amplitude")
subplot(4,1,3);
[H, W] = freqz(h, 1, 500);
plot(W,abs(H))
xlim([0 pi]);
title("|H(\omega)|")
xlabel("\omega")
ylabel("amplitude")
subplot(4,1,4);
plot(W,angle(H))
xlim([0 pi]);
title("\angle H(\omega)");
xlabel("\omega")
ylabel("amplitude")



%% 4
figure(4)
y = filter(h,1,xs2);
plot(t,x)
hold on;
stem(t2, y, "filled");
hold off;
xlim([tmin tmax]);
title("Low-Pass filtered upsampled x[n] vs x(t)")
xlabel("t (seconds)")
ylabel("amplitude")
legend("original", "upsampled")

%% 5
figure(5)
n = 0:L;
del = zeros(1, L+1);
del(L+1) = -1;
yshift = filter(del, 1, y);
plot(t,x)
hold on;
stem(t2, yshift, "filled")
hold off;
xlim([tmin tmax]);
title("Delayed Upsampled x[n] vs x(t)")
xlabel("t (seconds)")
ylabel("amplitude")
legend("original", "upsampled")
```