Name: Aditya Rajesh

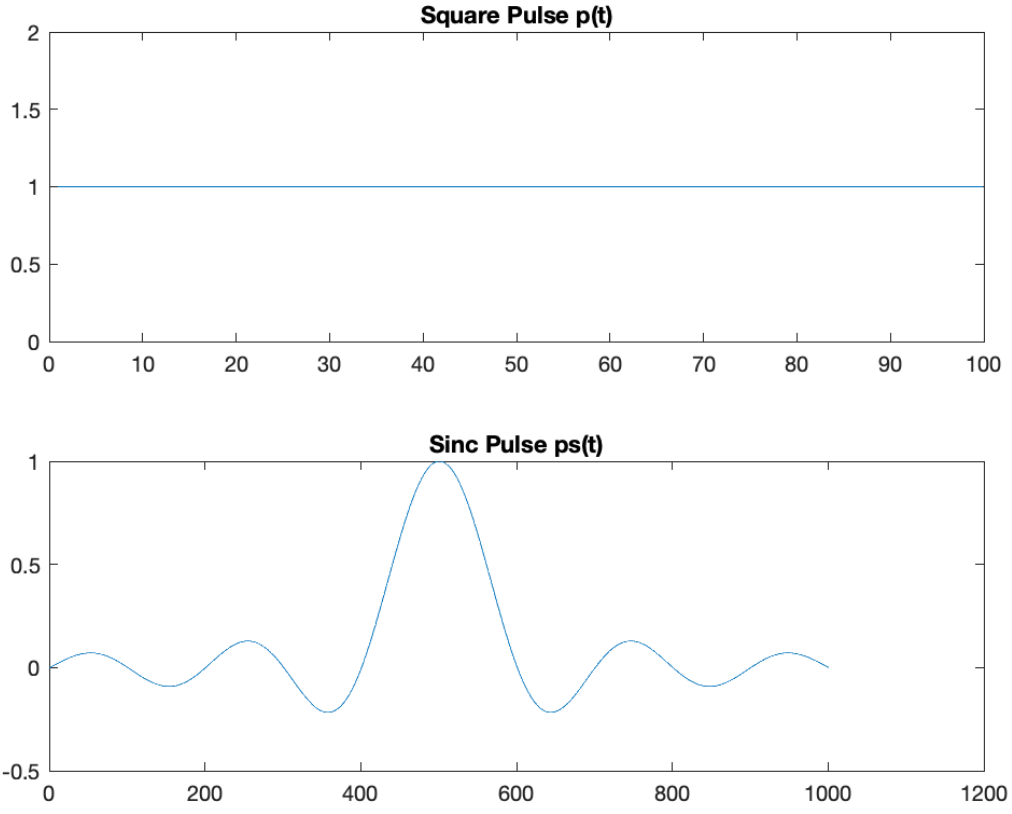
RUID: 208001821

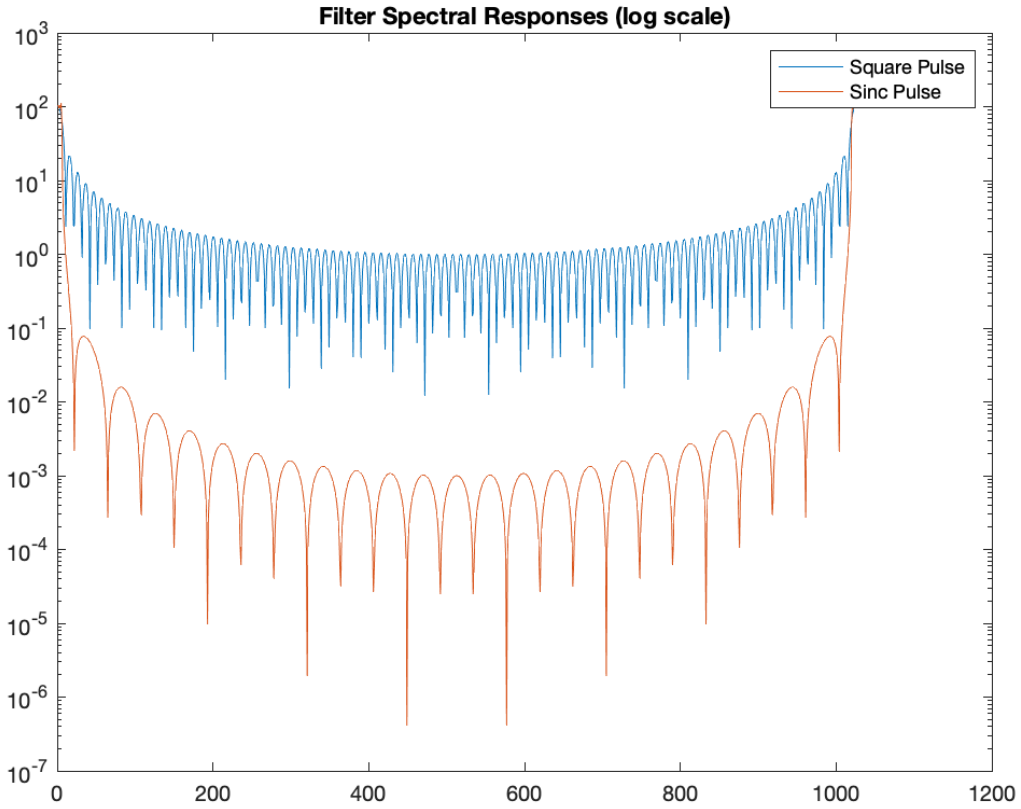
Date: 4/21/2025

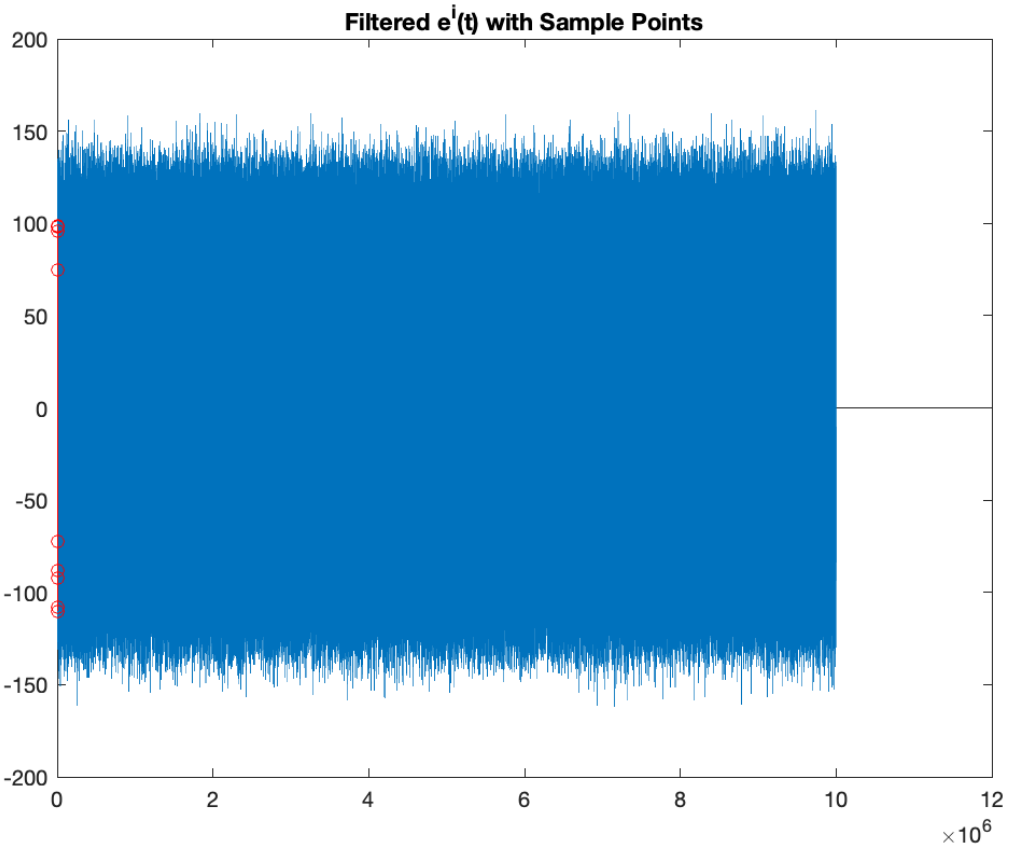
Report Format: Report should include the following four sections in this order and each starting in a new page. All pages should be submitted together in a single pdf file. If this format is not followed up to a 2 point (out of 10 points) penalty will be assessed.

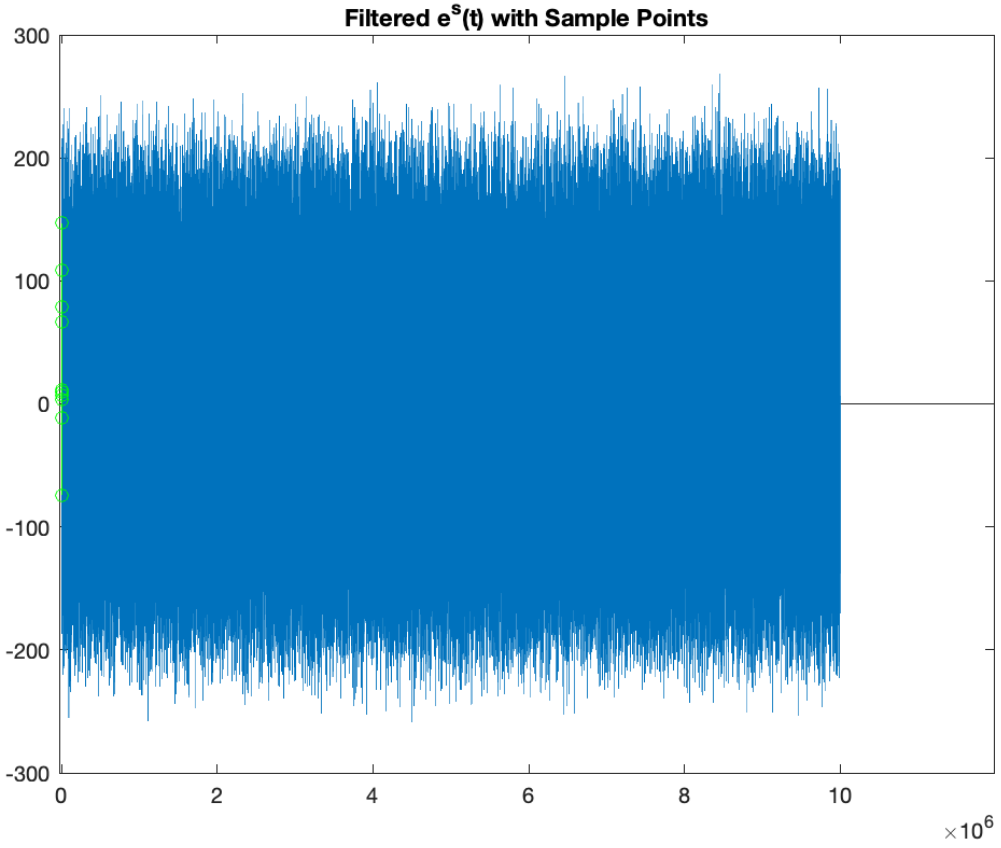
Report Sections:

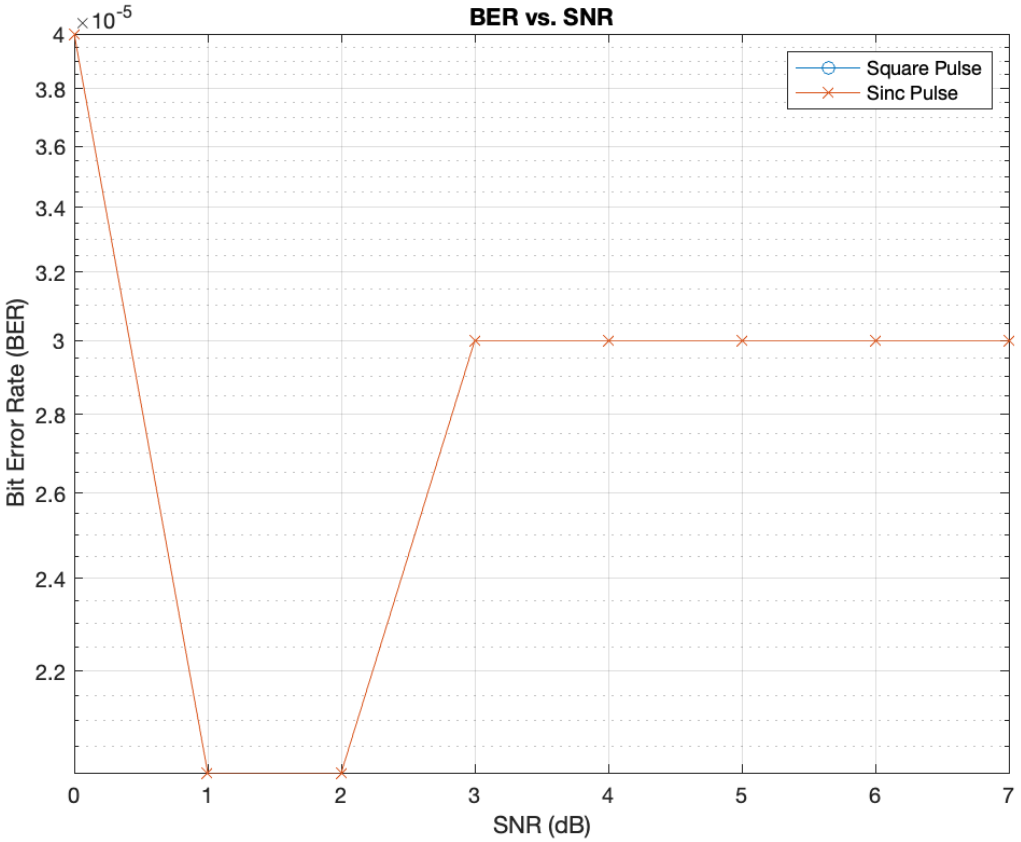
Results:




Filter Spectral Responses (log scale)
Square Pulse
Sinc Pulse
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
0
200
400
600
800
1000
1200


Filtered e^i(t) with Sample Points
200
150
100
50
0
-50
-100
-150
-200
0
2
4
6
8
10
12
×10^6


Filtered e^s(t) with Sample Points
300
200
100
0
-100
-200
-300
0
2
4
6
8
10
×10^6


BER vs. SNR
×10^-5
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
Bit Error Rate (BER)
Square Pulse
Sinc Pulse
0
1
2
3
4
5
6
7
SNR (dB)

assignment6.m

```
RUID = 208001821;
rng(RUID);

T = 2;
A = 1;
Ts = 0.02;
R = 1 / T;
N = 100000;
bit_samples = round(T / Ts);
L = N * bit_samples;

t_sinc = -5*T:Ts:5*T;
ps = A * sinc(t_sinc / T);
ps_delay = floor(length(ps)/2);

bb = randi([0, 1], 1, N);
disp('First 10 bits:');
disp(bb(1:10));

p = A * ones(1, bit_samples);
Ep = sum(p.^2) * Ts;
Eps = sum(ps.^2) * Ts;

s = zeros(1, L + length(ps));
ss = zeros(1, L + 2*ps_delay);
valid = false(1, N);

for i = 1:N
    idx = (i-1)*bit_samples + 1;
    center_idx = idx + bit_samples/2 - ps_delay;
    polarity = 2 * bb(i) - 1;

    s(idx:idx+bit_samples-1) =
s(idx:idx+bit_samples-1) + polarity * p;
```

```
    if center_idx > 0 && (center_idx + length(ps) -
1) <= length(ss)
        ss(center_idx:center_idx+length(ps)-1) = ...
            ss(center_idx:center_idx+length(ps)-1) +
polarity * ps;
        valid(i) = true;
    end
end

figure;
subplot(2,1,1); plot(p); title('Square Pulse p(t)');
subplot(2,1,2); plot(ps); title('Sinc Pulse ps(t)');

fprintf('Energy of p(t): %.4f, Energy of ps(t):
%.4f\n', Ep, Eps);
fprintf('Sinc modulated %d out of %d bits (%.2f%%)
\n', sum(valid), N, 100*sum(valid)/N);

SNR_dB = 0:1:7;
SNR = 10.^(SNR_dB / 10);
ni = cell(1, 8); nis = cell(1, 8);
for j = 1:8
    vi = Ep / SNR(j);
    vis = Eps / SNR(j);
    ni{j} = sqrt(vi) * randn(1, length(s));
    nis{j} = sqrt(vis) * randn(1, length(ss));
end

d = cell(1, 8); ds = cell(1, 8);
for j = 1:8
    d{j} = s + ni{j};
    ds{j} = ss + nis{j};
end

ft = fliplr(p);
fs = fliplr(ps);

Fft = abs(fft(ft, 1024));
Ffs = abs(fft(fs, 1024));
figure;
```

```
semilogy(Fft); hold on; semilogy(Ffs);
legend('Square Pulse', 'Sinc Pulse');
title('Filter Spectral Responses (log scale)');

e = cell(1,8); es = cell(1,8);
for j = 1:8
    e{j} = conv(d{j}, ft, 'same');
    es{j} = conv(ds{j}, fs, 'same');
end

sample_points_i = bit_samples/2 : bit_samples : N *
bit_samples - bit_samples/2;
sample_points_s = sample_points_i - ps_delay;
sample_points_s = sample_points_s(sample_points_s >
0);

figure;
plot(e{1}); hold on;
stem(sample_points_i(1:10), e{1}
(sample_points_i(1:10)), 'r');
title('Filtered e^i(t) with Sample Points');

figure;
plot(es{1}); hold on;
stem(sample_points_s(1:10), es{1}
(sample_points_s(1:10)), 'g');
title('Filtered e^s(t) with Sample Points');

BER_i = zeros(1,8);
BER_s = zeros(1,8);

for j = 1:8
    ri = e{j}(sample_points_i);
    rj = es{j}(sample_points_s);
    decoded_i = ri > 0;
    decoded_s = rj > 0;

    BER_i(j) = sum(decoded_i ~=
bb(1:length(decoded_i))) / length(decoded_i);
    BER_s(j) = sum(decoded_s ~=
bb(1:length(decoded_s))) / length(decoded_s);
```

```
end

figure;
semilogy(SNR_dB, BER_i, '-o'); hold on;
semilogy(SNR_dB, BER_s, '-x');
xlabel('SNR (dB)'); ylabel('Bit Error Rate (BER)');
legend('Square Pulse', 'Sinc Pulse');
title('BER vs. SNR');
grid on;
```