ECE 2713 Programming Module #3

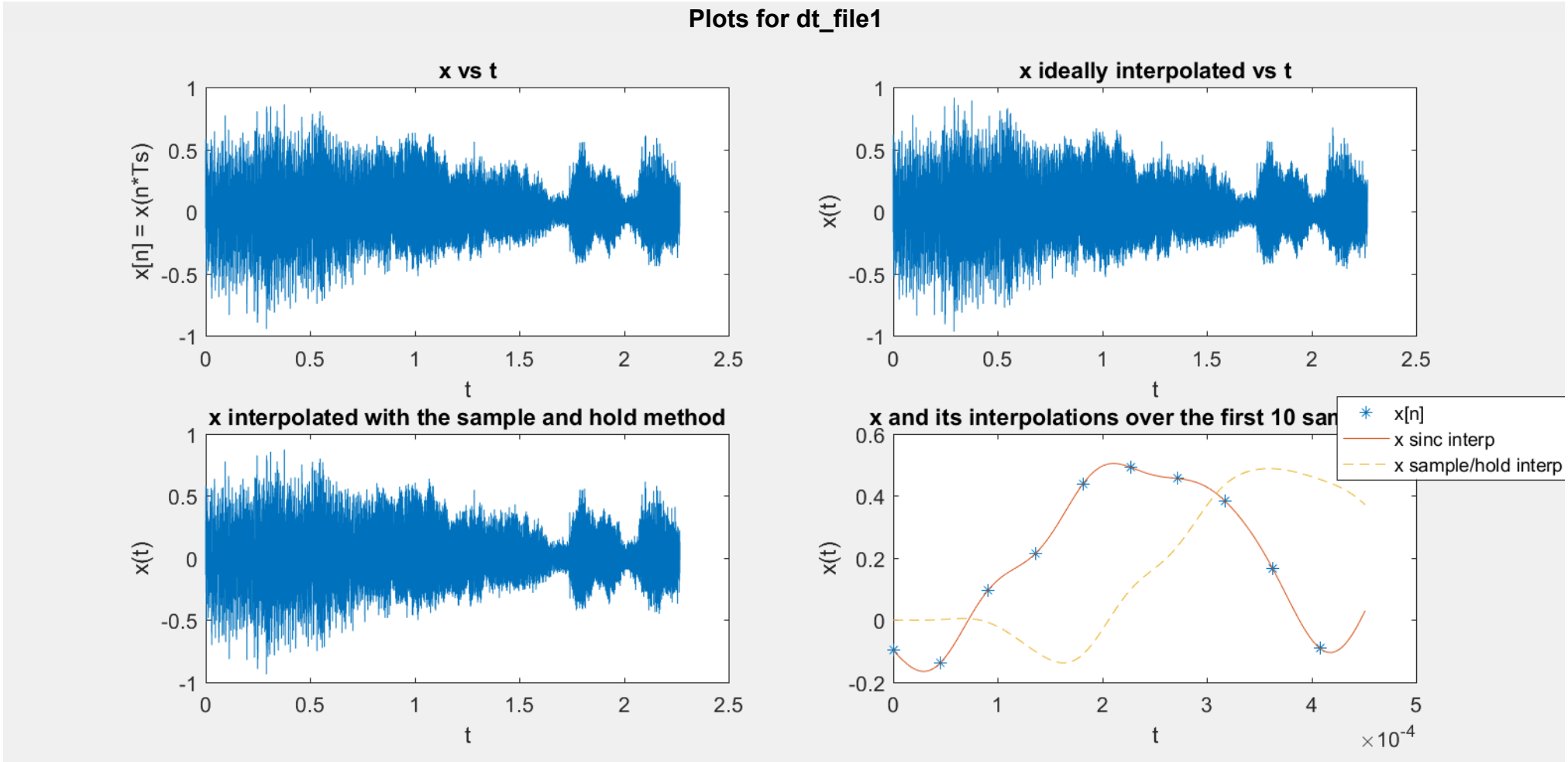
Randall Summers (113386678) – Electrical

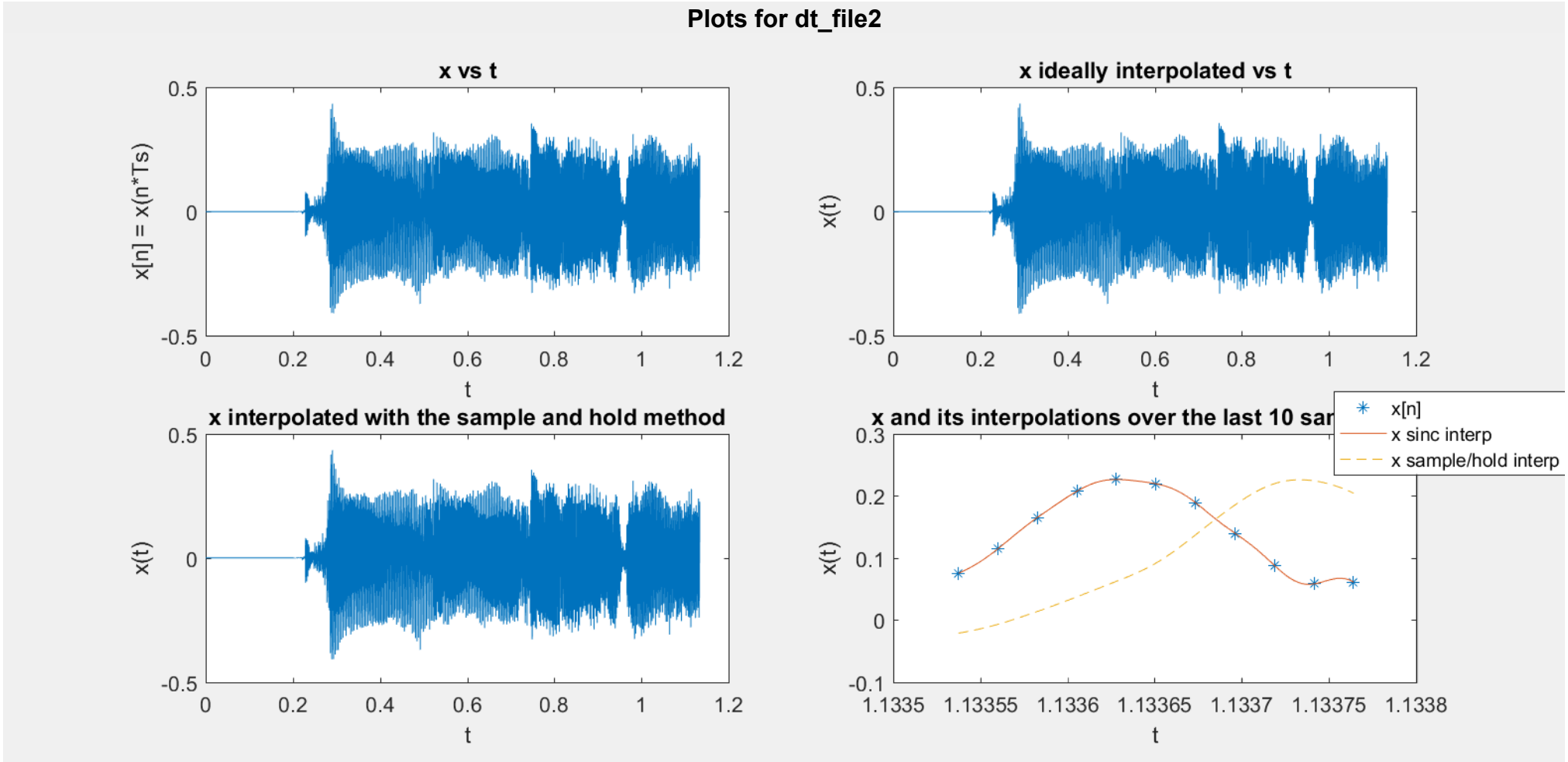
Tim Rose (113355567) – Computer and Electrical

Cathrine Bartlett (112934887) – Electrical

```
% Load data file and define important parameters
load('dt_file2.mat');
x = x(1:50000);
Ts = 1/Fs;
n = 0:(length(x)-1);
times = n*Ts;
% Create an array of 'continuous' times
Fsc = 441000;
Tsc = 1/Fsc;
L = Fsc/Fs;
t = linspace(0,times(end),length(x)*L);
% Create an array of ideally interpolated x-values
xinterp = zeros(1,length(t));
for N = 1:length(t)
    xinterp(N) = sum(x.*sinc((t(N)-n*Ts)/Ts));
end
% Create an array of sample-and-hold values
xheld = zeros(1,length(t));
Ntemp = 1;
for ntemp = 1:length(x)
    while Ntemp*Tsc < ntemp*Ts
        xheld(Ntemp) = x(ntemp);
        Ntemp = Ntemp + 1;
    end
end
filterCoeff = fir1(100,1/L);
xpractical = conv(xheld,filterCoeff);
% Plot everything in one gorgeous graphic
f = figure;
p = uipanel('Parent',f,'BorderType','none');
p.Title = 'Plots for dt_file2';
p.TitlePosition = 'centertop';
p.FontSize = 12;
p.FontWeight = 'bold';

ax1 = subplot(2,2,1,'Parent',p);
plot(ax1,times,x);
title(ax1,'x vs t');
ylabel(ax1,'x[n] = x(n*Ts)');
xlabel(ax1,'t');

ax2 = subplot(2,2,2,'Parent',p);
plot(ax2,t,xinterp);
title(ax2,'x ideally interpolated vs t');
ylabel(ax2,'x(t)');
xlabel(ax2,'t');

ax3 = subplot(2,2,3,'Parent',p);
plot(ax3,t,xpractical(1:length(t)));
title(ax3,'x interpolated with the sample and hold method');
```

```
ylabel(ax3,'x(t)');
xlabel(ax3,'t');

ax4 = subplot(2,2,4,'Parent',p);
plot(ax4,times((length(x)-10):end),x((length(x)-10):end), '*',t((length(t)-(10*L)):↙
length(t)),xinterp((length(t)-(10*L)):length(t)), '-',t((length(t)-(10*L)):length(t)), ↙
xpractical((length(t)-(10*L)):length(t)), '--');
title(ax4,'x and its interpolations over the last 10 samples of x' );
ylabel(ax4,'x(t)');
xlabel(ax4,'t');
legend(ax4,'x[n]','x sinc interp','x sample/hold interp');
```


Plots for dt_file1
x vs t
x[n] = x(n*Ts)
t
x ideally interpolated vs t
x(t)
t
x interpolated with the sample and hold method
x(t)
t
x and its interpolations over the first 10 sam
x(t)
t
×10^-4
x[n]
x sinc interp
x sample/hold interp


Plots for dt_file2
x vs t
x[n] = x(n*Ts)
t
x ideally interpolated vs t
x(t)
t
x interpolated with the sample and hold method
x(t)
t
x and its interpolations over the last 10 sar
x(t)
t
x[n]
x sinc interp
x sample/hold interp

Comments:

- The sinc interpolation over a finite time period seems to produce a highly accurate interpolation of the input signal.
- The sample and hold method produces increasingly smooth but delayed interpolations as you increase the filter's order, but the increased order also increases the interpolated signal's delay.
- Overall, the sinc interpolation is better if you have infinite computing power, but in practical scenarios, the sample and hold method is likely the more ideal choice assuming it is okay for the interpolated signal to be out of phase with the original.