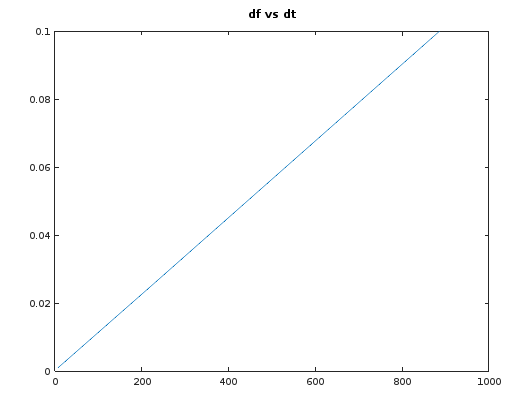
1. The max frequency is the sampling rate divided by two. In this case, the max frequency is 5000 Hz.

If you Fourier transform the entire one second sample, f is approximately 1. The exact value came out to be about 0.9.

3. df vs dt does show a straight line. This was obtained by running creating the envelope function several times with several different widths. Each envelope was applied to the sine wave and the df was recorded. dt was changed, then the loop ran again. df was determined to be the amplitude of the spike in the Fourier transformed pulse.



4. The message says "My cat likes to eat the dog's food."

The unwanted frequencies were at about 25500 and 42160 Hz.

The message was recovered by Fourier transforming the message and plotting it. The frequencies that were unwanted were identified. A filter was created essentially zeroed out the any frequencies above a certain range. The filter was applied to the Fourier transform messaged, inverse Fourier transformed, then played.

% clear variables from previous runs

clear;

t = 0.1:0.0001:1;

dt = [0.1 0.07525 0.0505 0.02575 0.001]

for k=1:5

y = sin(2\*pi\*200\*t); % Max frequency is 5000 Hz

env = exp(-(t-0.5).^2/(dt(k))^2);

pulse = env.\*y;

ftpulse = fft(pulse);

df(k) = (max(abs(ftpulse)));

plot(abs(ftpulse))

title('ftpulse')

end

figure

plot(t, pulse)

title('t vs pulse')

figure

plot(df, dt)

title('df vs dt')

%sound(pulse, 10000);

[y, fs] = audioread('SignalNoise.wav'); % audioread for use in octave

%sound(y, fs)

fmessage = fft(y);

plot(abs(fmessage))

filt = ones(size(fmessage), 1);

filt(1000:1400) = 0;

filt(3600:4400) = 0;

filt(169000:170400) = 0;

filt(172600: 173200) = 0;

filtered = fmessage .\*filt;

message = ifft(filtered);

sound(message, fs)