Table of Contents

INIT

```
t=0.001*(0:.001:2); % Given. time values.
x=@(t) 2*sin(2*pi*1000*t)+2*sin(2*pi*4000*t)+2*sin(2*pi*6000*t);
% Don't fully understand what the anonymous function does. It seems to
% create a 'new' function that takes arguments? This seems doable with the
% typical way to define a function.
Samplerate=[5000, 10000, 20000]; % Array of sample rates. Index with a for
loop.
```

Plot

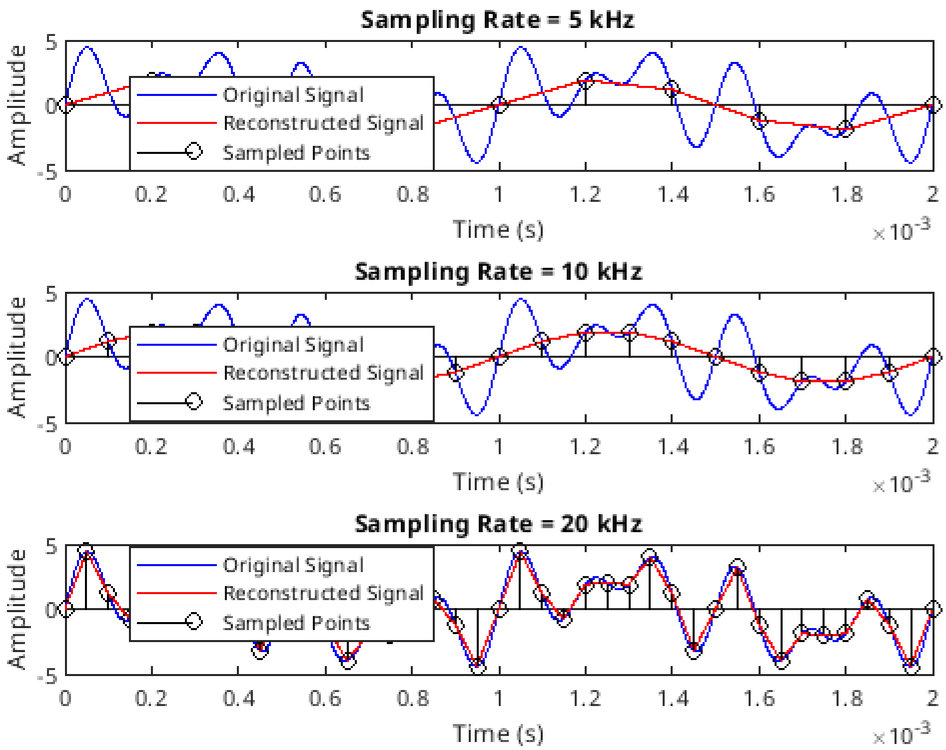
```
for i=1:length(Samplerate) % Choose each sample rate iteratively
    Ts=1/Samplerate(i); % Sampling period - inverse of sampling rate
    n=(0:Ts:2e-3); % discrete time values. in intervals of sampling period.

    % Sampled Signal
    x_discrete=x(n); % Plug in discrete time values to 'x' function.
    % This should return our discrete valued plot which we will get stems
    % from.

    % Reconstruct
    t_new=0:1e-6:2e-3; % The only way i could think of to reconstruct an
analog signal from a discrete was to make the time steps a lot smaller, so
that's basically what I did.
    x_new=interp1(n, x_discrete, t_new, 'linear');
    % I use linear interpolation. Found this online. Seems to fill in
    % missing data points.

    %Plot
    subplot(length(Samplerate), 1, i);
    plot(t, x(t), 'b', t_new, x_new, 'r');
    hold on;
    stem(n, x_discrete, 'k');
    hold off;
    % This took some fiddling to make it look right and display on one
    % graph.

    xlabel('Time (s)');
    ylabel('Amplitude');
    title(['Sampling Rate = ' num2str(Samplerate(i)/1000) ' kHz']);
    legend('Original Signal', 'Reconstructed Signal', 'Sampled Points',
'Location', 'best');
end
```



Writeup

```
%{
Aliasing occurs because of improper sampling. In these examples, we can see
that the reconstructed signal does not properly represent the original;
this is because the sampling rate was too low and failed to capture enough
data points to produce a close approximation, instead representing an 'alias'
to the original signal.
When we increase the signal to 10 or 15 khz, we see a much more faithful
representation of the original
signal, since the faster sampling rate captures more data. This is an
example of aliasing - when a digital signal is reconstructed into an
'alias' that does not match the original analog signal.

%}
```

Published with MATLAB® R2023b