```
% Digital down coversion copied from https://www.youtube.com/watch?
v=r43mirdfUAk&list=PLAH7p4h3HetCBskQ_SGC_xh-qcoEcK1j_&index=2
close all; clear;
```

```
% Paramaters
Fs = 500e6; % 500 MHz
T = 20e-6;
N = round(Fs*T);
t = (0:N-1)' / Fs;
```

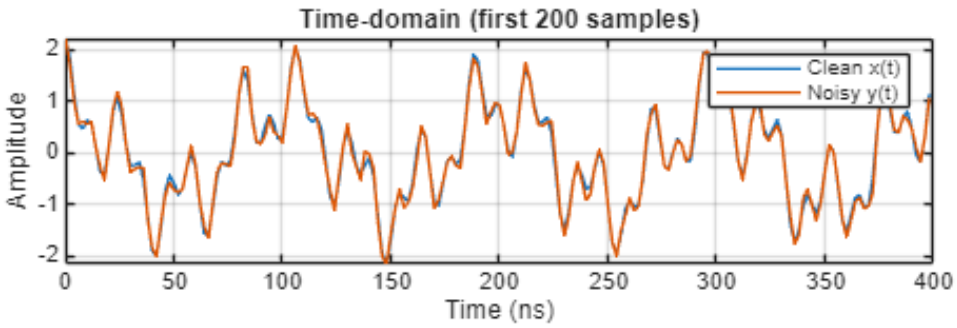
```
f = [10e6, 37e6, 85e6]; %tone frequencies 10 MHz, 37 MHz, 85 MHz
A = [1.0,  0.7,  0.5 ]; % tone amplitudes
phi = 2*pi*rand(size(f)); % random phases
```

```
x = sum(A .* cos(2*pi.*f.*t),2); % sum along rows of each signal
```

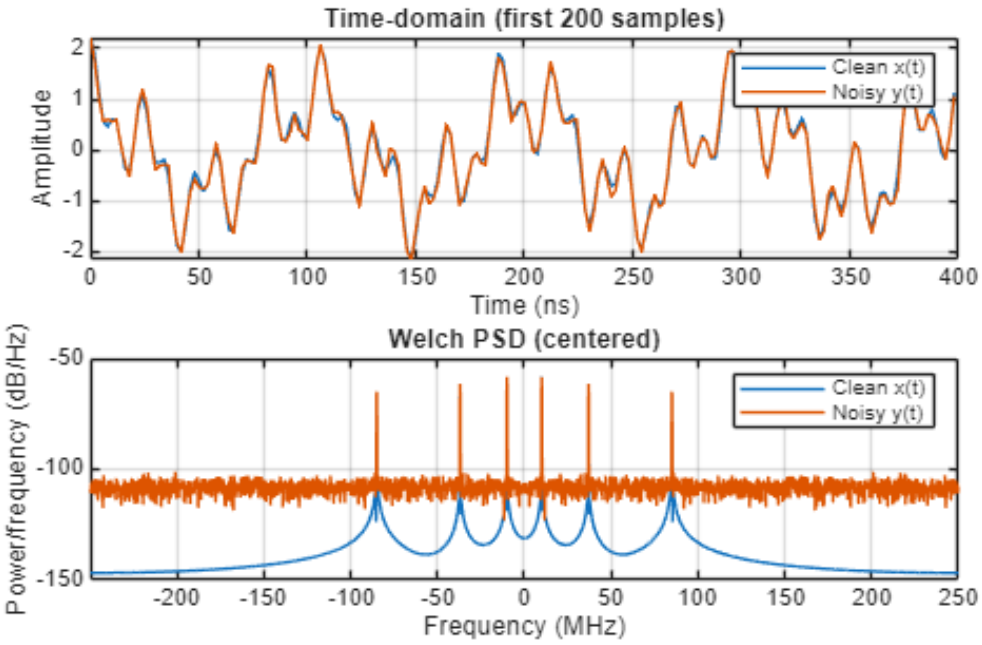
```
% add normally distributed noise
snr_DB = 20;
Px = mean(x.^2); % The power of x (Watts?)
Pn = Px / 10^(snr_DB/10); % power of noise (Watts)
noise = sqrt(Pn) * randn(size(x)); % additive white gaussian noise (AWGN)
(amplitude)
```

```
y = x + noise;
```

```
figure;
subplot(2,1,1);
plot(t(1:200)/1e-9, x(1:200), 'linewidth', 1); hold on;
plot(t(1:200)/1e-9, y(1:200), 'linewidth', 1); hold off;
xlabel('Time (ns)'); ylabel('Amplitude');
legend('Clean x(t)', 'Noisy y(t)'); grid on; title('Time-domain (first 200
samples)');
```

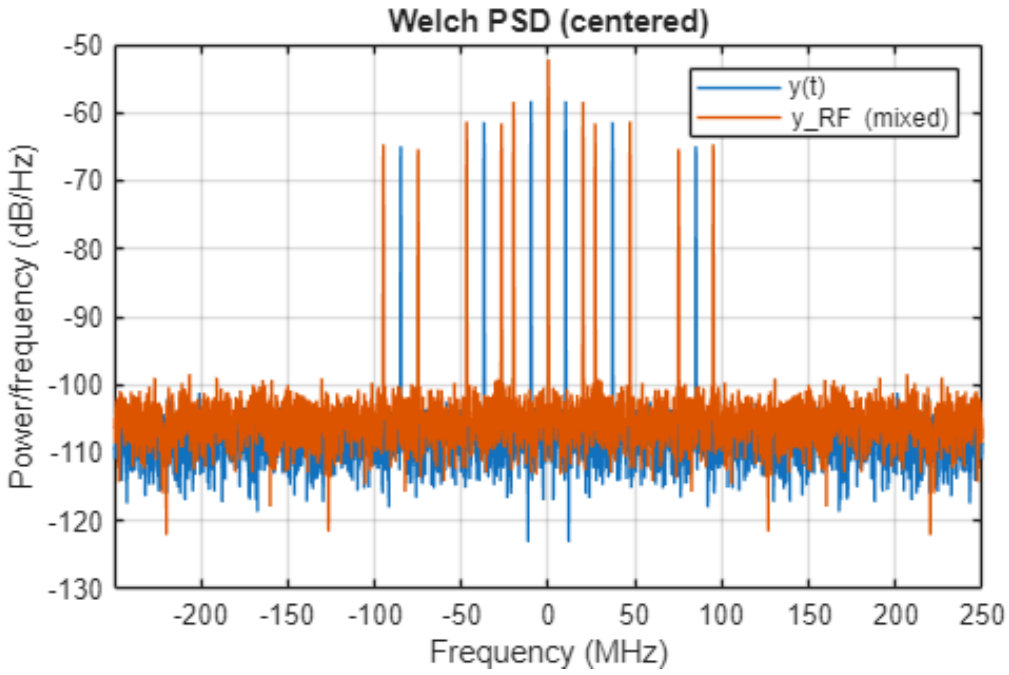



```
% Find power spectral density using pwelch
% power distribution across frequency
subplot(2,1,2);
pwelch(x, 4096, 2048, 4096, Fs, 'centered'); hold on;
pwelch(y, 4096, 2048, 4096, Fs, 'centered'); hold off;
legend('Clean x(t)', 'Noisy y(t)'); title('Welch PSD (centered)'); grid on;
```



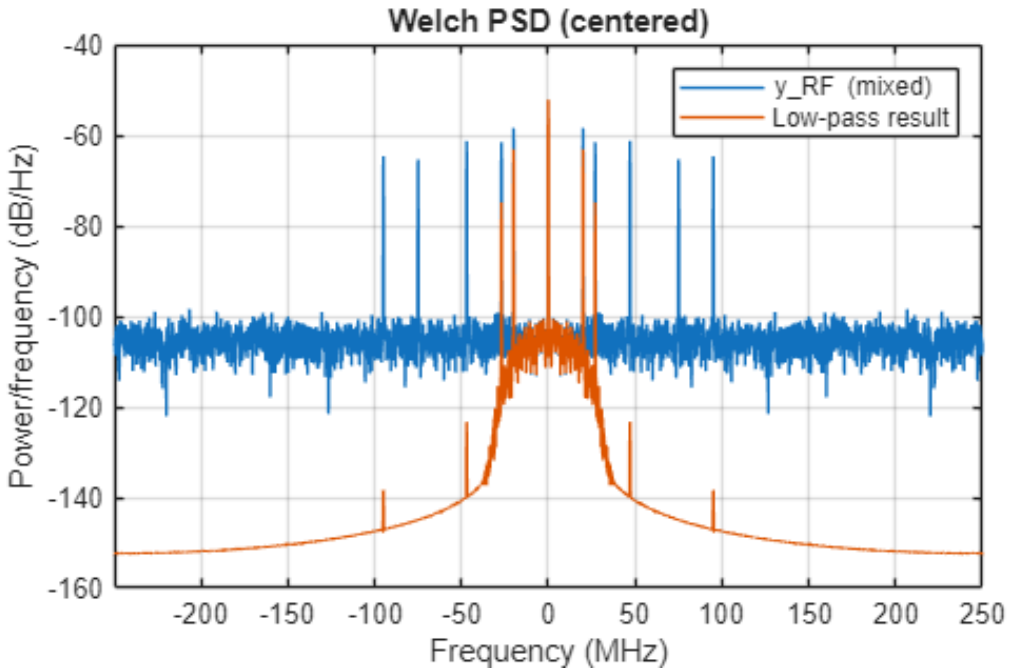

```
% downshift time :)
f_LO = 10e6; % 10 MHz local osc
```

```
% real mixing (dobule-sideband): y_RF has components at f +/- f_LO
y_RF = 2*y .* cos(2*pi*f_LO*t);  % 2x for ~0 dB conversion gain

% If we didn't multiply by 2 this would be more realistic because we would
% not be magically increasing the power of the signal at DC by 6 dB

figure;
pwelch(y, 4096, 2048, 4096, Fs, 'centered'); hold on;
pwelch(y_RF, 4096, 2048, 4096, Fs, 'centered'); hold off;
legend('y(t)', 'y\_RF (mixed)'); title('Welch PSD (centered)'); grid on;
```



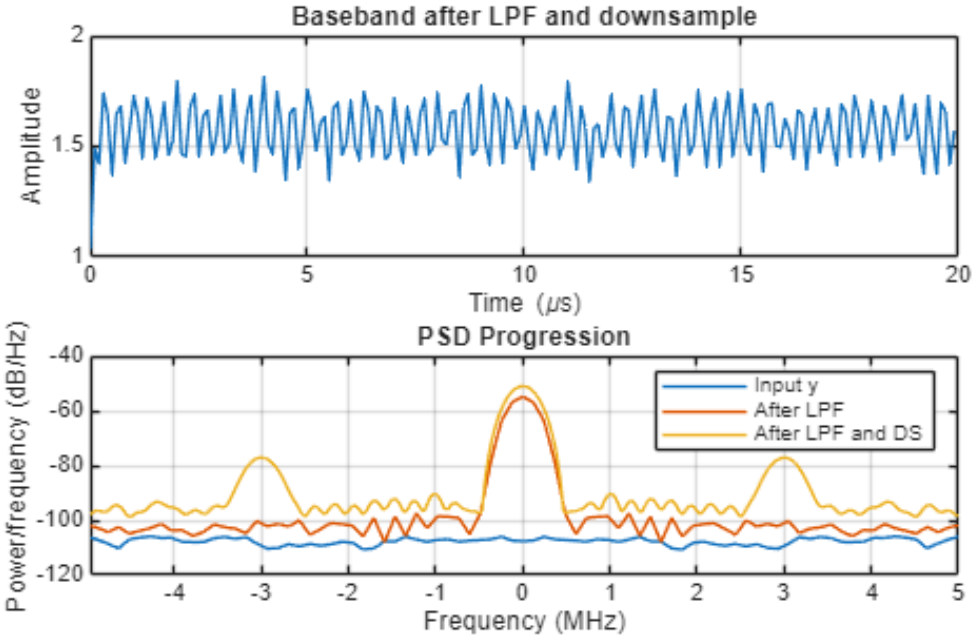

```
% low pass filter
F_c = 2e6; % cutoff of 2 MHz
z_lp = lowpass(y_RF, F_c, Fs);
```

```
figure;
pwelch(y_RF, 4096, 2048, 4096, Fs, 'centered'); hold on;
pwelch(z_lp, 4096, 2048, 4096, Fs, 'centered'); hold off;
legend('y\_RF (mixed)', 'Low-pass result'); title('Welch PSD (centered)'); grid on;
```




```
% downsample
M = 50; % decimation of 50
Fs_ds = Fs / M;
z_ds = downsample(z_lp, M);
t_ds = (0:numel(z_ds)-1)' / Fs_ds;
```

```
figure;
subplot(2,1,1);
ns = min(5000, numel(z_ds));
plot(t_ds(1:ns)/1e-6, z_ds(1:ns), 'linewidth', 1); hold on;
xlabel('Time (\mus)'); ylabel('Amplitude'); grid on;
title('Baseband after LPF and downsample');

subplot(2,1,2);
pwelch(y, [], [], [], Fs, 'centered'); hold on;
pwelch(z_lp, [], [], [], Fs, 'centered');
pwelch(z_ds, [], [], [], Fs_ds, 'centered'); hold off;
legend('Input y', 'After LPF', 'After LPF and DS'); title('PSD Progression'); grid
on;
```




```
% why change the paramaters of pwelch here? It looks like it affects the
% amplitudes slightly. IDK what value it uses by default for window size
% etc.
```

Small bumps at -3 and 3 MHz are aliasing or filter leakage