```
%Refresh
clc;
close all;
clear;

% Channel is currently modeled as the 10 cycle impulse from:
% Noah's Laptop -> Sound Card -> Radio x2 -> Sound Card -> Laptop
load('impulse_resp.mat');

% Number of Bits to send (1406*8 is a cat)
Nbits = 1000;
% Set a random data sequence
data = round(rand(1,Nbits));
% Sample Frequency, from MSK
FS = 44100;

% % Assemble total data vector
x = [];
x = MSKmod((data));

% Set noise powers based on x and given SNR
avg_pow_x = mean(x.^2);
SNR = 13.5;
noise_pow = 10^(-SNR/10)*avg_pow_x

% Remove DC offset from channel (wierd)
impulse_rx = impulse_rx - mean(impulse_rx);

% "Send" X by convolving with impulse response
y = conv(x,impulse_rx);
y = y + sqrt(noise_pow)*randn(size(y));
% Normalize as the actual radios through audacity do this
y = y/max(y);

% figure(2);
% subplot(2,1,1);
% plot((1:length(x))/40e3,x);
% title('Output from TX');
% ylabel('Amplitude');
% xlabel('Time');
% ylim([-1.5 1.5])
%
%
% subplot(2,1,2);
% plot((1:length(y))/40e3,y);
% title(sprintf('Input to RX (output + gaussian noise, SNR=%.3f dB)',
 SNR));
% ylabel('Amplitude');
% xlabel('Time');

% Decode recieved y, lots going on here
```

```
[dout, STARTWORD_INDICES, ENDWORD_INDICES,CLOCKLOCK_INDICES] =
 MSKdecode(y);


% Overlay where the sync words, end words and clocking adjustments
 take
% place
STARTWORD_Y = -10*ones(1,length(y));
STARTWORD_Y(STARTWORD_INDICES) = 1;
ENDWORD_Y = -10*ones(1,length(y));
ENDWORD_Y(ENDWORD_INDICES) = 1;
CLOCKLOCK_Y = -10*ones(1,length(y));
CLOCKLOCK_Y(CLOCKLOCK_INDICES) = 1;

% Print data success statistics
min_length = min([length(data) length(dout)]);
agrees = (data(1:min_length) == dout(1:min_length));
length_eq = length(dout) == length(x)
BitErrors = sum(~agrees);
```

Plot data transmission, key timer markers

```
h=plot(1:length(y), y,                          ...
    1:length(y),    STARTWORD_Y,    'g.',    ...
    1:length(y),    ENDWORD_Y,      'r.',    ...
    1:length(y),    CLOCKLOCK_Y,    'k.');

set(h,{'markers'},{1;15;15;15})
xlabel 'Sample #'
ylabel 'Amplitude'
title('Data Transmission')
ylim([-9 9]);
BER = BitErrors/length(data)
```

Published with MATLAB® R2015b