
Table of Contents

c3_check_modulation.m	1
Paths	1
Importing input data after modulation	1
Importing input data before modulation	1
Resample	2
Plot signal 1 (real & imag)	2
Plot signal 2 (real & imag)	3

c3_check_modulation.m

```
%{  
Description:  
    This script verifies the consistency between the original baseband  
    signals (before modulation) and the signals recovered after  
    modulation  
    and resampling from PA simulation data.  
  
Input:  
    - lte_real.csv (f1_source_data)  
    - lte_imag.csv (f1_source_data)  
    - wlan11n_real.csv (f1_source_data)  
    - wlan11n_imag.csv (f1_source_data)  
    - pa_data.mat  
}  
clear; clc; close all;  
tic
```

Paths

Functions

```
current_folder = fileparts(mfilename('fullpath'));  
root_folder = fileparts(current_folder);  
functions_folder = fullfile(root_folder, 'f0_functions');  
addpath(functions_folder);  
  
% Source  
source_folder = fullfile(root_folder, 'f1_source_data');
```

Importing input data after modulation

```
pa_data_file = fullfile(current_folder, 'pa_data.mat');  
load(pa_data_file);
```

Importing input data before modulation

```
[s1_time, s1_signal] = read_complex_csv( ...
```

```

fullfile(source_folder, 'lte_real.csv'), ...
(fullfile(source_folder, 'lte_imag.csv')));

[s2_time, s2_signal] = read_complex_csv( ...
    fullfile(source_folder, 'wlan1n_real.csv'), ...
    fullfile(source_folder, 'wlan1n_imag.csv'));

```

Resample

```

s1_baseband = interp1(s1_time, s1_signal,
    time_baseband, 'linear', 'extrap');
s2_baseband = interp1(s2_time, s2_signal,
    time_baseband, 'linear', 'extrap');

% Adjust amplitudes so that both baseband signals (before modulation)
% match
% the maximum values of the recovered signals (after demodulation).
% This adjustment is necessary because during modulation a gain was
% applied
% to push the PA input signal close to its limit, introducing a slight
% nonlinear behavior for analysis.

max_s1_recovered = max(abs(signal_1_in));
max_s1_original = max(abs(s1_baseband));
ratio = max_s1_recovered / max_s1_original;
s1_baseband = s1_baseband * ratio;

max_s2_recovered = max(abs(signal_2_in));
max_s2_original = max(abs(s2_baseband));
ratio = max_s2_recovered / max_s2_original;
s2_baseband = s2_baseband * ratio;

```

Plot signal 1 (real & imag)

```

figure('Name','Signal 1 Baseband Comparison','Color','w');

% --- Real part ---
ax1 = subplot(2,1,1);
plot(time_baseband*1e6, real(signal_1_in), 'b', 'LineWidth', 1.2);
hold on;
plot(time_baseband*1e6, real(s1_baseband), 'r--', 'LineWidth', 1.2);
xlabel('Time (\mu s)');
ylabel('Amplitude (real)');
legend('Recovered Signal 1','Original Signal 1');
grid on;

% --- Imag part ---
ax2 = subplot(2,1,2);
plot(time_baseband*1e6, imag(signal_1_in), 'b', 'LineWidth', 1.2);
hold on;
plot(time_baseband*1e6, imag(s1_baseband), 'r--', 'LineWidth', 1.2);
xlabel('Time (\mu s)');
ylabel('Amplitude (imag)');

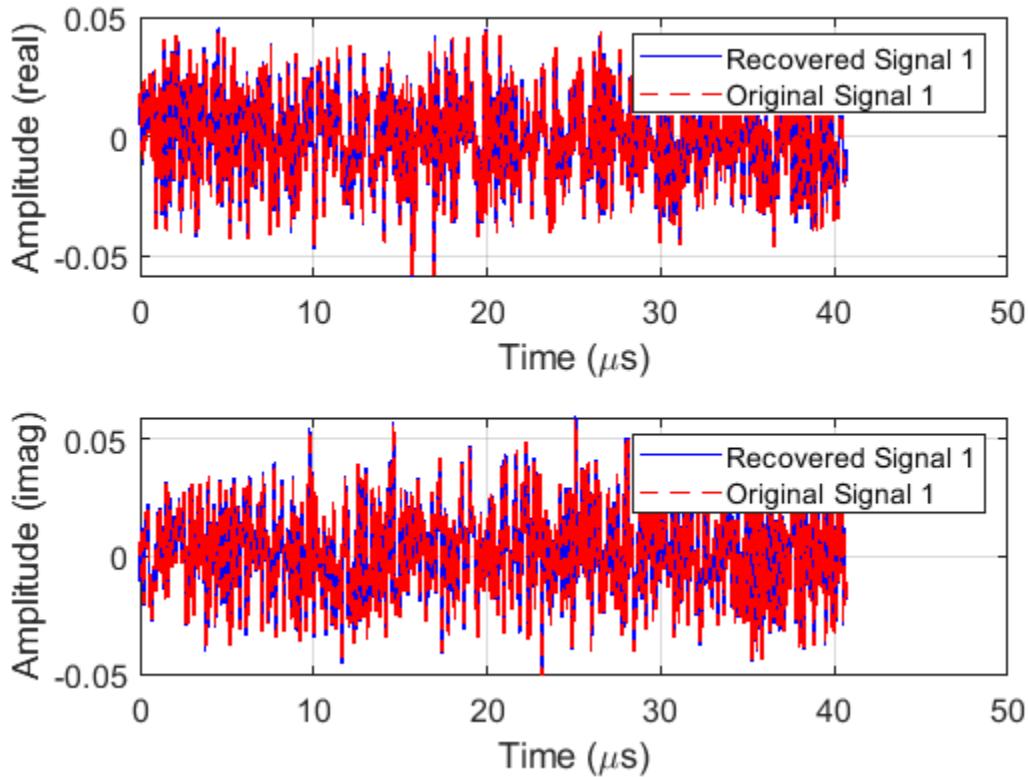
```

```

legend('Recovered Signal 1','Original Signal 1');
grid on;

% Paper-style formatting
set([ax1, ax2], 'FontSize',12,'LineWidth',1);

```



Plot signal 2 (real & imag)

```

figure('Name','Signal 2 Baseband Comparison','Color','w');

% --- Real part ---
ax1 = subplot(2,1,1);
plot(time_baseband*1e6, real(signal_2_in), 'b', 'LineWidth', 1.2);
hold on;
plot(time_baseband*1e6, real(s2_baseband), 'r--', 'LineWidth', 1.2);
xlabel('Time (\mu s)');
ylabel('Amplitude (real)');
legend('Recovered Signal 2','Original Signal 2');
grid on;

% --- Imag part ---
ax2 = subplot(2,1,2);
plot(time_baseband*1e6, imag(signal_2_in), 'b', 'LineWidth', 1.2);
hold on;
plot(time_baseband*1e6, imag(s2_baseband), 'r--', 'LineWidth', 1.2);
xlabel('Time (\mu s)');

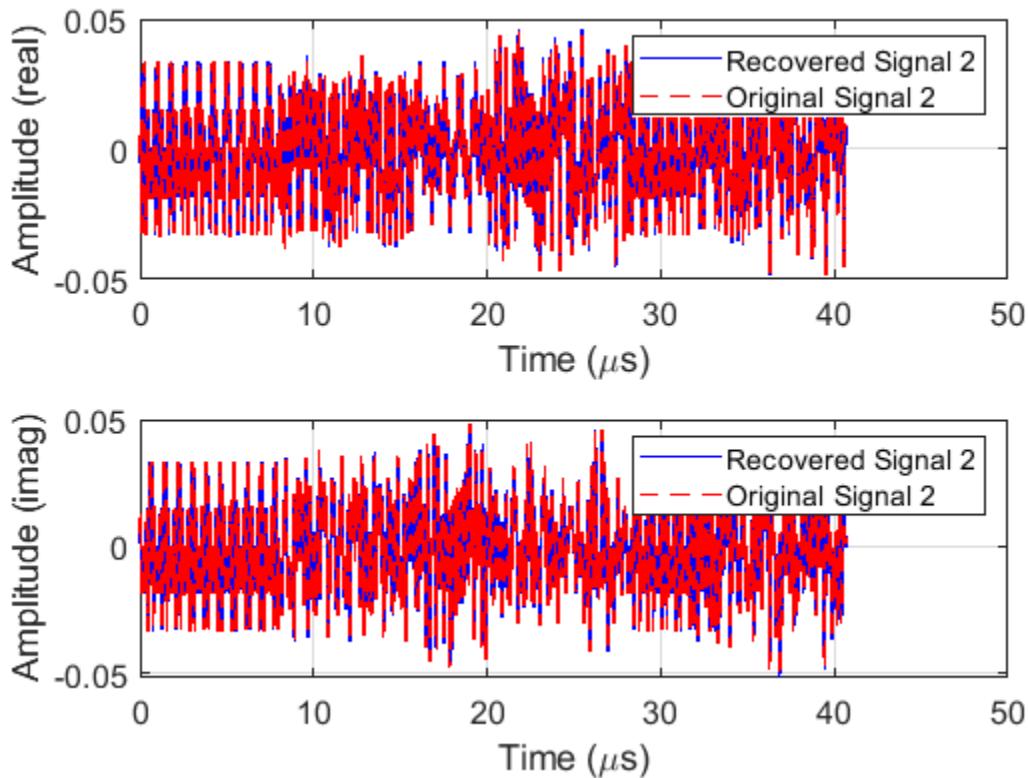
```

```
ylabel('Amplitude (imag)');
legend('Recovered Signal 2','Original Signal 2');
grid on;
```

```
% Paper-style formatting
set([ax1, ax2], 'FontSize',12,'LineWidth',1);
```

```
toc
```

Elapsed time is 2.882315 seconds.



Published with MATLAB® R2019a