Table of Contents

c1_modulation.m

```
%{
Description:
  This script implements the modulation for a dual-band system. It
 uses as
  input the data collected in Cadence sources. The transmitted signal
  is saved in a .pwl file.

Input:
  - source_signal_1 (f1_source_data)

Output:
  - transmitted_signal.pwl
%}
clear; clc; close all;
tic
```

Parameters

```
freq_carrier_1 = 1.8e9;
freq_carrier_2 = 5.4e9;
gain = 0.04;
```

Importing functions

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

Importing data

```
source_folder = fullfile(root_folder, 'f1_source_data');
source_file = fullfile(source_folder, 'source_signal_1.mat')
```

```
load(source_file)
```

```
source_file =

    'C:\Users\Shoit\Desktop\pa_db_1p8_5p4\f1_source_data
\source_signal_1.mat'
```

Resample to RF

Creating oversampled time vector

```
freq_oversampling = 7 * max(freq_carrier_1, freq_carrier_2)
duration = time_baseband(end) - time_baseband(1)
time_oversampled = (0: freq_oversampling*duration).' /
 freq_oversampling;

% Computing interpolation
s1_oversampled = interp1(time_baseband, s1_baseband,
 time_oversampled);
s2_oversampled = interp1(time_baseband, s2_baseband,
 time_oversampled);
```

```
freq_oversampling =

   3.7800e+10
```

```
duration =

   4.0642e-05
```

Normalization

```
s1_baseband = s1_baseband / max(abs(s1_baseband));
s2_baseband = s2_baseband / max(abs(s2_baseband));

max_val_1 = max(abs(s1_baseband))
max_val_2 = max(abs(s2_baseband))
```

```
max_val_1 =

    1.0000
```

```
max_val_2 =

     1
```

Shift

Computing carrier

```
positive_carrier_1 = exp(1i*2*pi*freq_carrier_1*time_oversampled);
positive_carrier_2 = exp(1i*2*pi*freq_carrier_2*time_oversampled);

% Computing shift
s1_passband = s1_oversampled .* positive_carrier_1;
s2_passband = s2_oversampled .* positive_carrier_2;
```

Transmitted signal

Combine signals

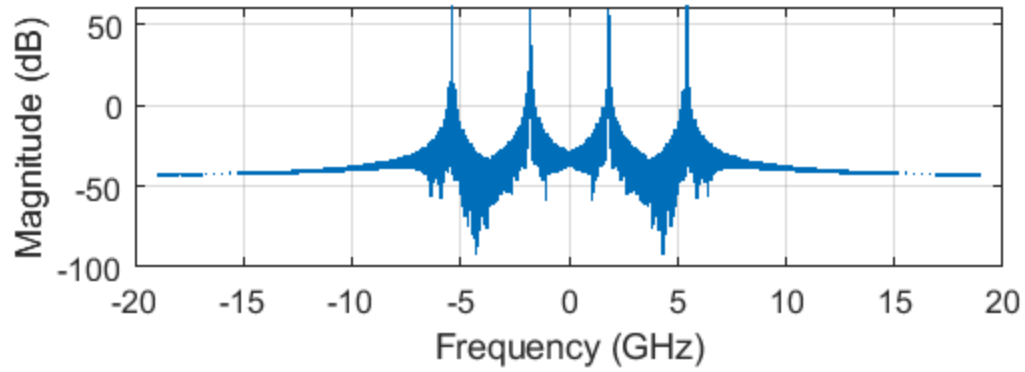
```
transmitted_signal = real(s1_passband + s2_passband);

% Normalize signal
max_val = max(abs(transmitted_signal));
transmitted_signal = transmitted_signal / max_val;
transmitted_signal = transmitted_signal * gain;

% Optional: check
check_max_value = max(abs(transmitted_signal))
```

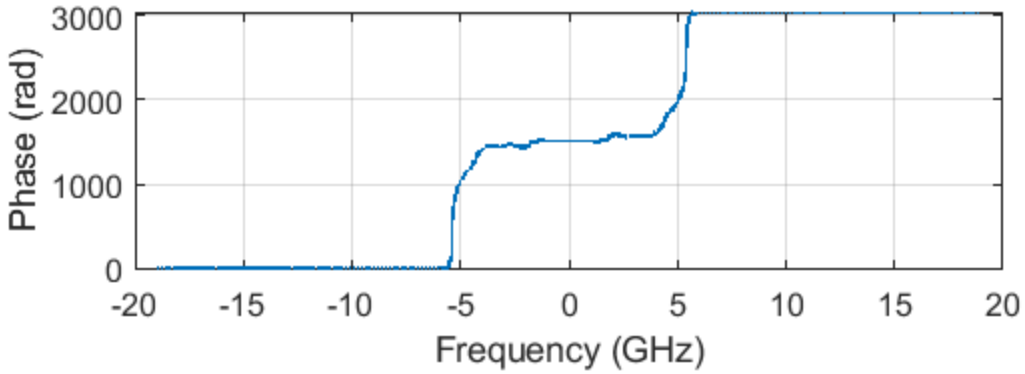
```
check_max_value =

    0.0400
```

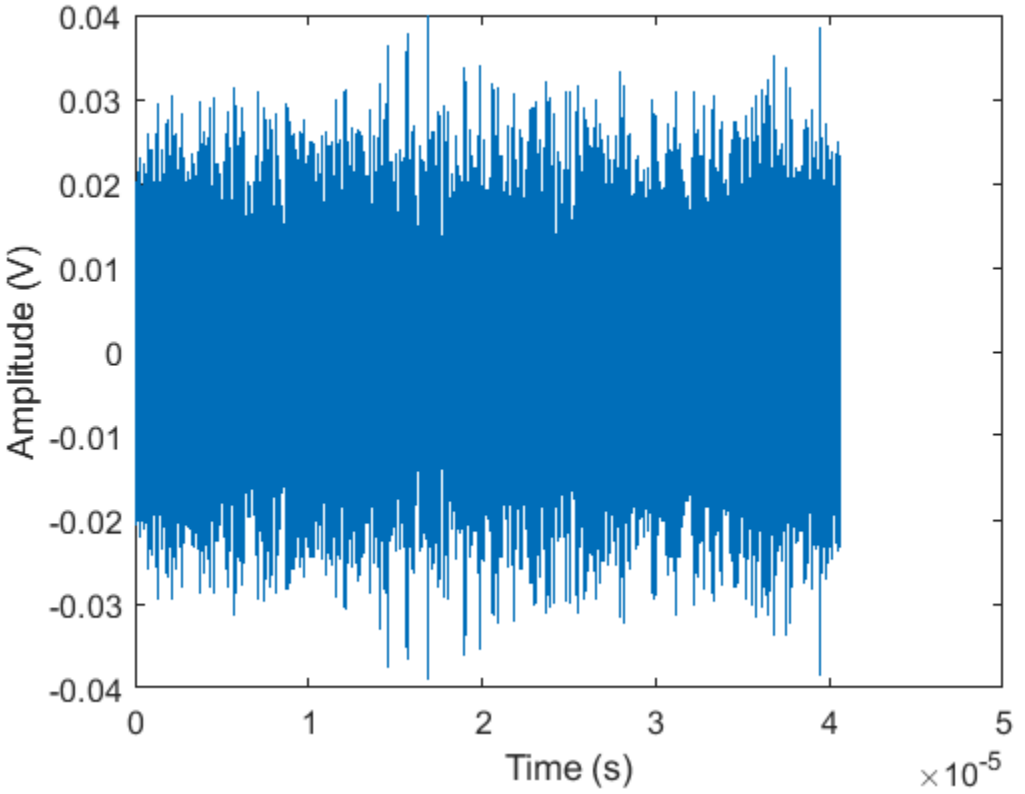
Plot

Frequency domain

```
plot_spectrum(transmitted_signal, freq_oversampling, 'Transmitted
 Signal')

% Time domain
figure();
plot(time_oversampled, transmitted_signal);
xlabel('Time (s)');
ylabel('Amplitude (V)');
ax = gca;
set(ax,'FontSize',12,'LineWidth',1);
```


Magnitude (dB)
50
0
-50
-100
-20
-15
-10
-5
0
5
10
15
20
Frequency (GHz)


Phase (rad)
3000
2000
1000
0
-20
-15
-10
-5
0
5
10
15
20
Frequency (GHz)


Amplitude (V)
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
0
1
2
3
4
5
Time (s)
×10-5

Save in .pwl file

Get folder of current script

```
script_folder = fileparts(mfilename('fullpath'));
pwl_filename = fullfile(script_folder, 'transmitted_signal.pwl');

% Combine time and signal into two columns
data_to_save = [time_oversampled, transmitted_signal];

% Open file
fid = fopen(pwl_filename, 'w');
if fid == -1
    error('Could not open file %s for writing.', pwl_filename);
end

% Write data as two columns: time, signal
% note the transpose for column-wise fprintf
fprintf(fid, '%.16e,%.16e\n', data_to_save.');
fclose(fid);

disp(['Data saved as: ', pwl_filename]);

toc
```

```
Data saved as: C:\Users\Shoit\Desktop\pa_db_1p8_5p4\f2_pa_mod
\transmitted_signal.pwl
Elapsed time is 9.147451 seconds.
```

Published with MATLAB® R2019a