Table of Contents

Receiver Impairments and compensation

Transmit and receive some symbols through an AWGN channel, taking into consideration that the IQ receiver has some DC offset and phase mismatch and crosstalk.

```
clc; clear; close all;
```

Parameters

```
mod_type = ModulationTypes.QAM;     % Modulation type.
M = 16;                             % Modulation order.
symbol_qtty = 1e4;                  % Amount of symbols to send.
L = 10;                             % Oversampling factor.
nTaps = 30;                         % Taps for pulse shaping FIR filters.
beta = 1;                           % Slope of the SRRC filter.
EsNo_dB = 20;                       % EsNo

% Change the behaviour of the IQ receiver here
gain = 0.9;                         % Gain mismatch [times]
phi = 8;                            % Phase mismatch [degree]
dc_i = 1.7;                         % DC offset for In-phase branch.
dc_q = 1.9;                         % DC offset for Quadrature branch.
```

Transmitter

```
d = randi([0, M-1], 1, symbol_qtty);
[u, constellation] = Modulator.modulate(d, mod_type, M);
v = Modulator.upsample(u, L);
[s, p, delay_tx] = Modulator.pulse_shaping_srrc(v, beta, L, nTaps);
```

Channel

```
[r, h_c] = Channel.add_awgn_noise(s, EsNo_dB, L);
```

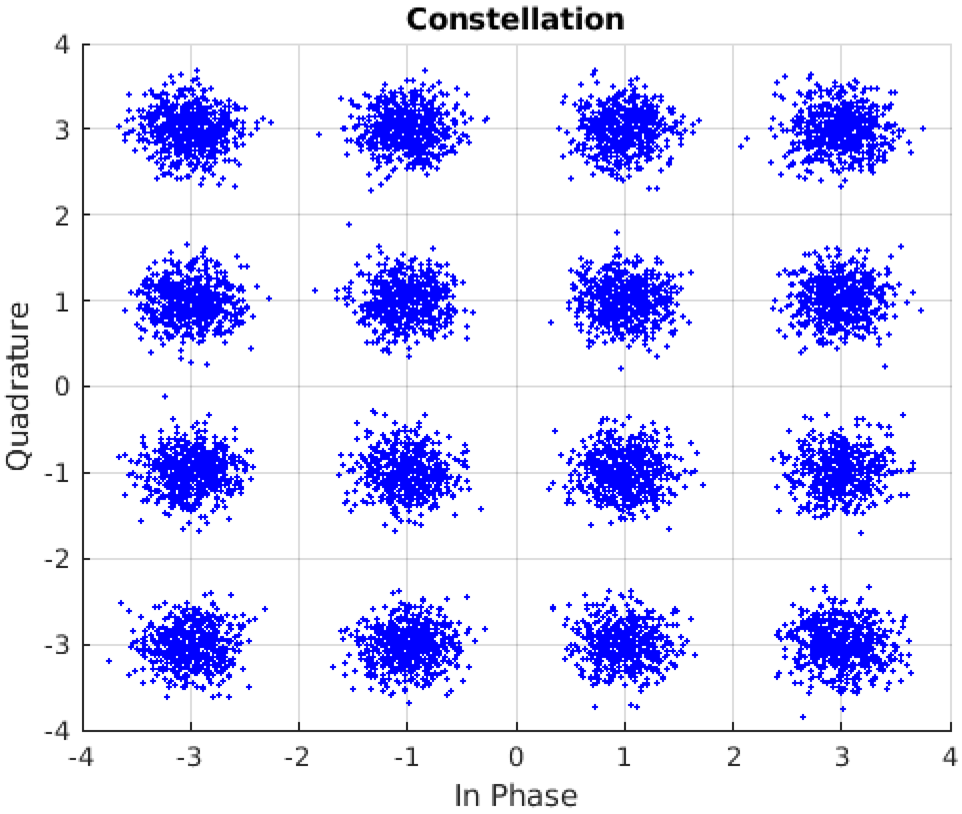
Receiver

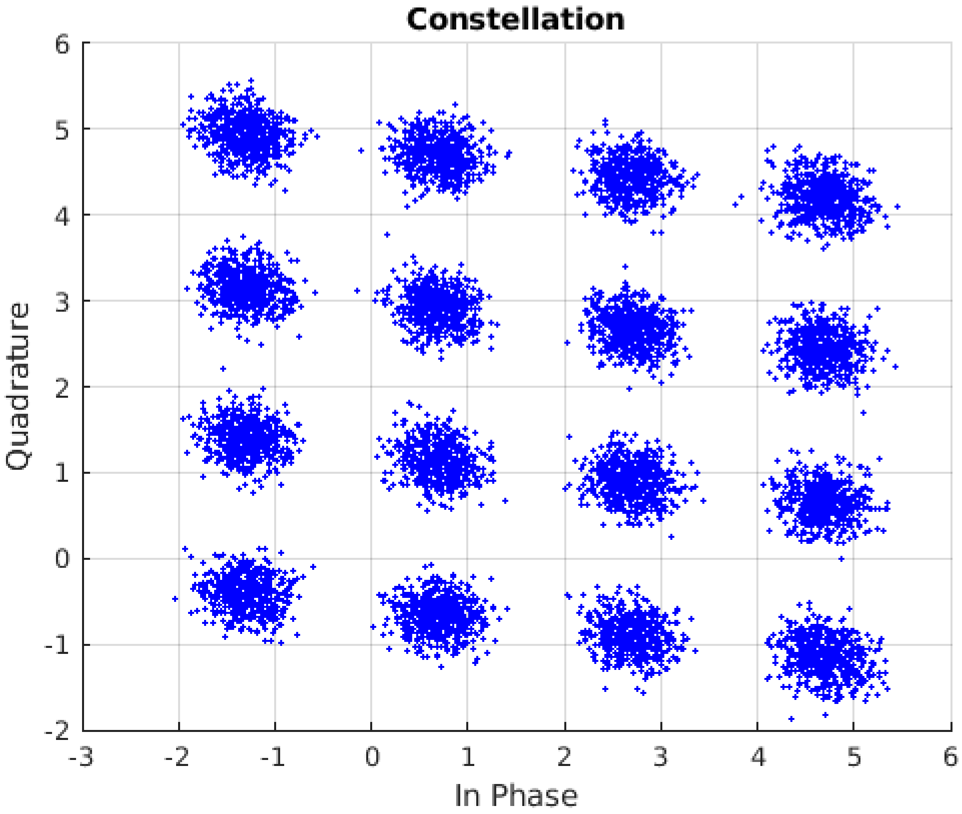
```
r = Demodulator.flat_fading_equalizer(r, h_c);
```

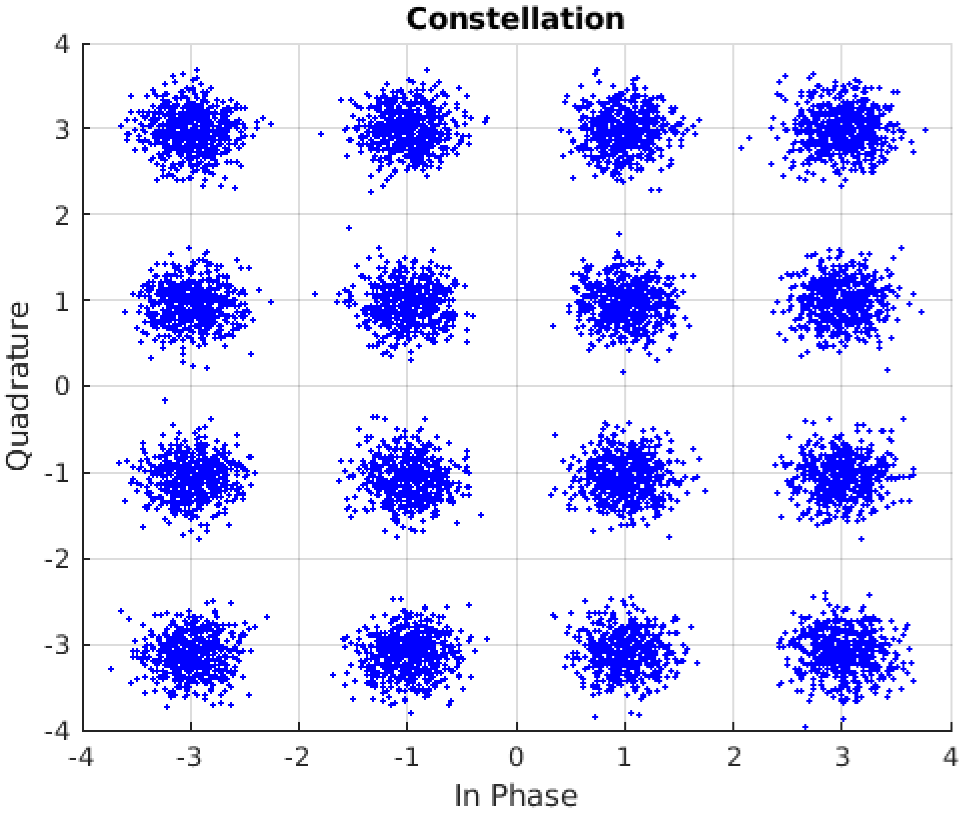
```
[v_r, g, delay_rx] = Demodulator.pulse_filter_srrc(r, beta, L, nTaps);
u_r = Demodulator.downsample(v_r, L, delay_tx + delay_rx);

z = Demodulator.receiver_impairments(u_r, gain, phi, dc_i, dc_q);
w = Demodulator.blind_iq_compensation(z, constellation);
d_r = Demodulator.demodulate(u_r, mod_type, M, constellation);
```

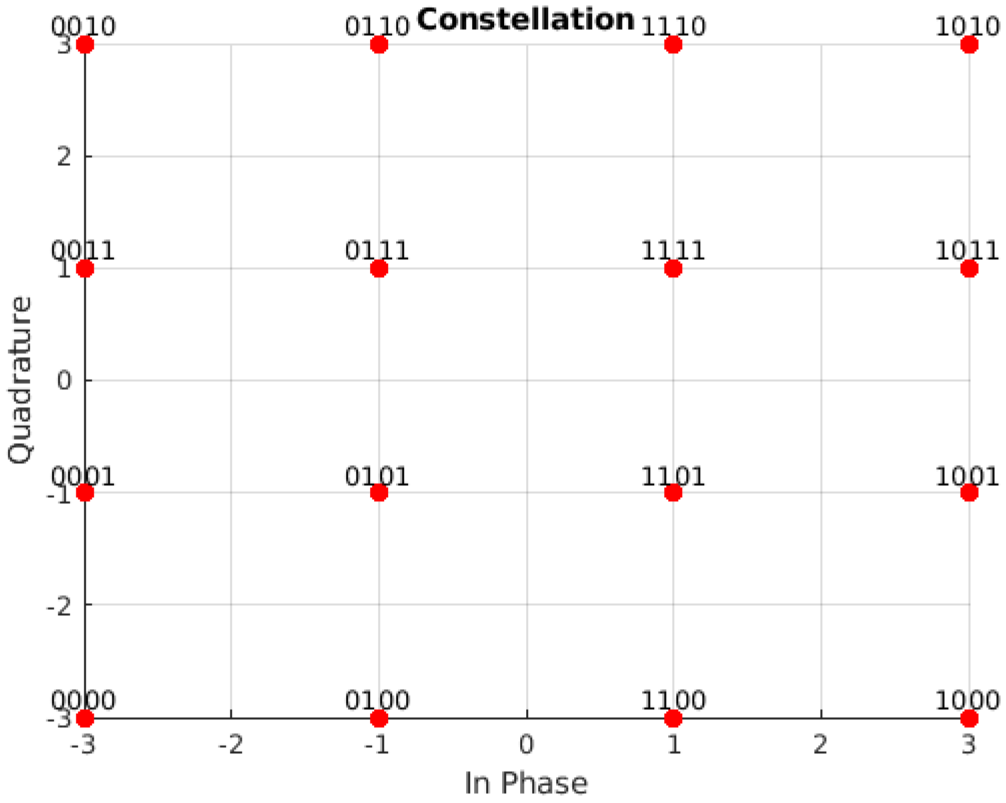
Plotting

```
Scope.plot_IQ(u_r);
Scope.plot_IQ(z);
Scope.plot_IQ(w);
Scope.plot_constellation(constellation);
```




Constellation
Quadrature
In Phase
6
5
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
4
5
6


Constellation
Quadrature
In Phase
4
3
2
1
0
-1
-2
-3
-4
-4
-3
-2
-1
0
1
2
3
4



Published with MATLAB® R2022b