Table of Contents

EQUALIZER EXAMPLE

Plots channel and equalizer frequency responses, for an arbitrary channel response defined in the time domain.

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

Parameters

```
nTaps = 14;                     % Length of the equalizer filter
Fs = 100;                       % [Hz] Sampling frequency
L = 5;                          % Oversampling factor
```

Calculations

```
Ts = 1 / Fs;                    % Sampling time
Tsym = L * Ts;                  % Symbol time
t = -6*Tsym : Ts : 6*Tsym;      % Time vector for channel response

h_t = 1 ./ (1 + (t/Tsym).^2);   % Channel temporal response

% Add noise to the channel
N0 = 0.001;
h_t = h_t + N0*randn(1, length(h_t));

% Sample channel at n*Tsym samples.
h_n = h_t(1:L:end);

% Get equalizer filter (the transmitted signal will be a Dirac's delta
% which, after passing through the channel, returns the impulse response
% "h_n").
[h_sys, w, delay, error] = Demodulator.zf_equalizer(h_n, h_n, nTaps);
```

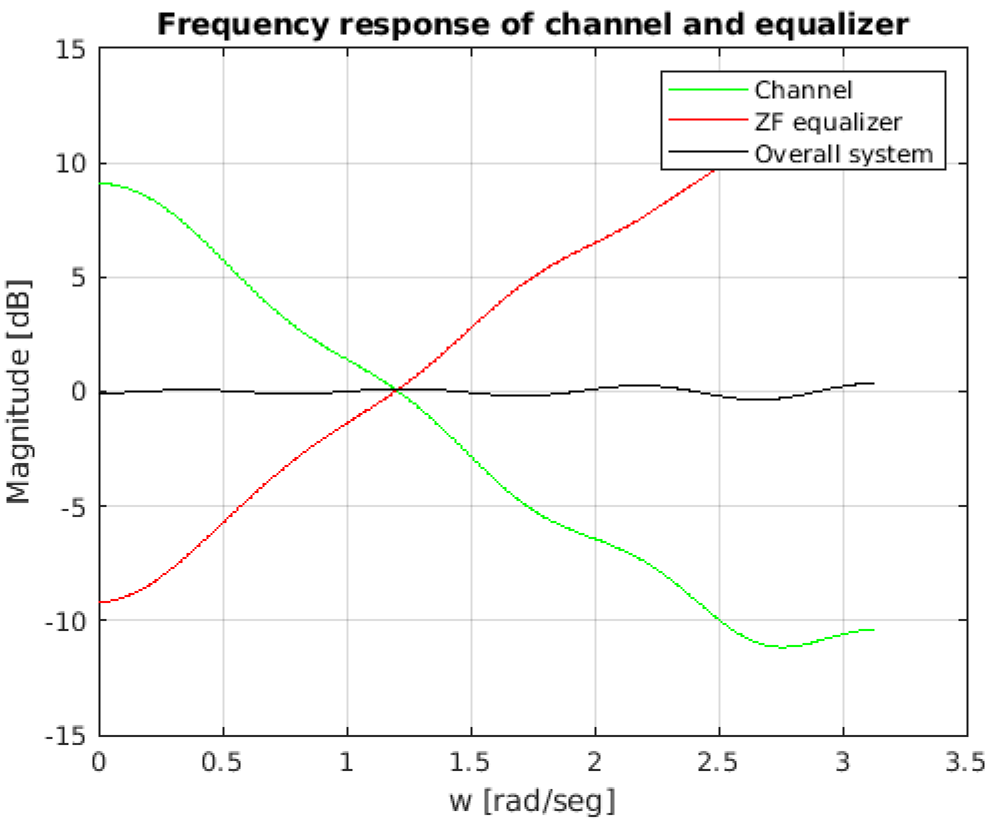
Plotting

Plot frequency response of the channel, of the equalizer, and both

```
[H_F, omega_hf] = freqz(h_n);
[W, omega_w] = freqz(w);
[H_SYS, omega_hsys] = freqz(h_sys);

figure;
plot(omega_hf, 20*log10(abs(H_F)), "g"); hold on;
```

```
plot(omega_w, 20*log10(abs(W)), "r");
plot(omega_hsys, 20*log10(abs(H_SYS)), "k");
legend("Channel", "ZF equalizer", "Overall system");
title("Frequency response of channel and equalizer");
ylabel("Magnitude [dB]");
xlabel("w [rad/seg]");
grid on;
```



Published with MATLAB® R2022b