Channel Equalization in Digital Communication (Zero-Forcing Equalizer)

Title of the Project:

To perform Channel Equalization in BPSK Systems Using Zero-Forcing Equalizer.

Software Required: Matrix Laboratories

Abstract:

Channel equalization is a critical process in digital communication systems to mitigate the effects of intersymbol interference (ISI) introduced by the communication channel. This project implements a Zero-Forcing (ZF) Equalizer for a Binary Phase Shift Keying (BPSK) modulation scheme to reduce ISI and improve signal recovery at the receiver. The transmitted signal is generated using random binary data, which is then modulated using BPSK. A multi-tap channel with defined impulse response introduces ISI into the transmitted signal. Additionally, additive white Gaussian noise (AWGN) is added to simulate realistic channel conditions. The Zero-Forcing Equalizer is designed based on the known channel impulse response to eliminate ISI by inverting the channel effect. The equalized signal is subsequently demodulated using a decision threshold, and the performance is evaluated in terms of the Bit Error Rate (BER). The results demonstrate that the ZF equalizer effectively reduces the BER compared to the case without equalization, though noise amplification remains a limitation in low signal-to-noise ratio (SNR) scenarios. This study highlights the trade-offs involved in using Zero-Forcing Equalizers and suggests the need for noise-resilient equalization techniques like Minimum Mean Square Error (MMSE) equalizers for enhanced performance.

Code:

```
% Channel Equalization in Digital Communication (Zero-Forcing Equalizer)


% Step 1: Generate transmitted signal (random BPSK data)

num_symbols = 1000 ;  % Number of symbols

data_bits = randi([0 1], num_symbols, 1);  % Generate random binary data (0s and 1s)

tx_signal = 2*data_bits - 1;  % BPSK modulation (0 -> -1, 1 -> 1)


%% Step 2: Create Message Signal and Carrier Signal

fs = 100;  % Sampling frequency

t = (0:num_symbols-1)/fs; % Time vector

f_c = 5;  % Carrier frequency
```

```
% Create message signal (BPSK modulated)
message_signal = tx_signal;

% Create carrier signal
carrier_signal = cos(2 * pi * f_c * t)';

% Modulate the message signal onto the carrier
modulated_signal = message_signal .* carrier_signal;

%% Step 3: Simulate the Channel (Add ISI)
channel = [0.8, 0.2, 0.1];  % Channel impulse response (includes ISI)
tx_signal_channel = conv(modulated_signal, channel, 'same');

%% Step 4: Add Noise (AWGN)
SNR = 10;  % Signal-to-noise ratio in dB
rx_signal = awgn(tx_signal_channel, SNR, 'measured');

%% Step 5: Compute ISI Ratio
% ISI = Convolved signal - Direct signal contribution
direct_signal = modulated_signal * channel(1);  % Direct path contribution
isi_component = tx_signal_channel - direct_signal;

% Compute ISI power and total signal power
isi_power = mean(abs(isi_component).^2);
total_power = mean(abs(tx_signal_channel).^2);

isi_ratio = isi_power / total_power;

%% Step 6: Channel Equalization (Zero-Forcing Equalizer)
```

```
equalizer_taps = length(channel);
rx_signal_padded = [zeros(equalizer_taps-1, 1); rx_signal];  % Zero-padding

% Create the ZF equalizer using the channel response
equalized_signal = zeros(size(rx_signal));
for n = equalizer_taps:num_symbols
   rx_segment = rx_signal_padded(n-equalizer_taps+1:n);
   rx_segment = rx_segment(:);
   channel_vector = channel(:);
   if length(rx_segment) == length(channel_vector)
      % Zero-forcing: Remove ISI by subtracting the contribution of past symbols
      equalized_signal(n) = rx_signal_padded(n) - sum(channel_vector .* rx_segment);
   else
      error('Dimension mismatch between channel and received signal segment');
   end
end

%% Step 7: Demodulate Signals
% Coherent demodulation using the carrier signal
received_signal = rx_signal .* carrier_signal;
equalized_signal = equalized_signal .* carrier_signal;

% Decision rule for BPSK
equalized_data_bits = equalized_signal > 0;
demodulated_data_bits = received_signal > 0;

%% Step 8: Compute the Bit Error Rate (BER)
ber_equalized = sum(data_bits ~= equalized_data_bits) / num_symbols;
ber_no_equalization = sum(data_bits ~= demodulated_data_bits) / num_symbols;
```

```
%% Step 9: Display Results
disp(['ISI Ratio: ', num2str(isi_ratio)]);
disp(['Bit Error Rate (with Equalization): ', num2str(ber_equalized)]);
disp(['Bit Error Rate (without Equalization): ', num2str(ber_no_equalization)]);

%% Step 10: Plot Signals
figure;

% Plot message signal
subplot(5, 1, 1);
plot(t, message_signal);
title('Message Signal (BPSK)');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot carrier signal
subplot(5, 1, 2);
plot(t, carrier_signal);
title('Carrier Signal');
xlabel('Time (s)');
ylabel('Amplitude');

% Plot transmitted signal with ISI
subplot(5, 1, 3);
plot(tx_signal_channel);
title('Transmitted Signal with ISI');
xlabel('Symbol Index');
ylabel('Amplitude');
```

```
% Plot received signal with ISI and noise (NEW)

subplot(5, 1, 4);

plot(rx_signal);

title('Received Signal with ISI and Noise');

xlabel('Symbol Index');

ylabel('Amplitude');


% Plot equalized signal

subplot(5, 1, 5);

plot(equalized_signal);

title('Equalized Signal (Zero-Forcing)');

xlabel('Symbol Index');

ylabel('Amplitude');

% end of the code
```

Description of Code:

This above code simulates a digital communication system with Binary Phase Shift Keying (BPSK) modulation. It includes steps to simulate channel effects such as Inter-Symbol Interference (ISI), add noise, apply a Zero-Forcing (ZF) equalizer to mitigate these effects, and evaluate the performance by calculating the Bit Error Rate (BER).

These are the steps involved;

1: Generate Transmitted Signal

- **BPSK Modulation**: The code starts by generating a random binary sequence (data_bits) and then applies Binary Phase Shift Keying (BPSK) modulation to it. In BPSK, binary 0 is mapped to -1 and binary 1 is mapped to 1 (tx_signal = 2*data_bits - 1).

2: Simulate the Channel

- **Channel Impulse Response**: A simple channel impulse response is defined with three taps (channel = [0.8, 0.2, 0.1]). This simulates a channel that introduces Inter-Symbol Interference (ISI).

- **Convolution**: The transmitted signal is convolved with the channel impulse response to simulate the effect of the channel on the signal (tx_signal_channel = conv(tx_signal, channel, 'same')).

3: Add Noise

- **AWGN**: Additive White Gaussian Noise (AWGN) is added to the signal to simulate real-world communication conditions. The signal-to-noise ratio (SNR) is set to 10 dB (rx_signal = awgn(tx_signal_channel, SNR, 'measured')).

4: Channel Equalization (Zero-Forcing Equalizer)

- **Equalizer Design**: The zero-forcing equalizer is designed based on the channel impulse response. It aims to remove ISI by applying an inverse filter.
- **Implementation**: The code attempts to implement a simple form of equalization but contains errors in logic. A correct implementation would typically involve finding the inverse of the channel response and using it to filter the received signal.

5: Demodulate Signals

- **Decision Rule**: The equalized and received signals are demodulated using a simple decision rule for BPSK (equalized_data_bits = equalized_signal > 0 and demodulated_data_bits = rx_signal > 0).

6: Compute Bit Error Rate (BER)

- **BER Calculation**: The bit error rates are calculated for both the equalized and non-equalized signals (ber_equalized and ber_no_equalization).

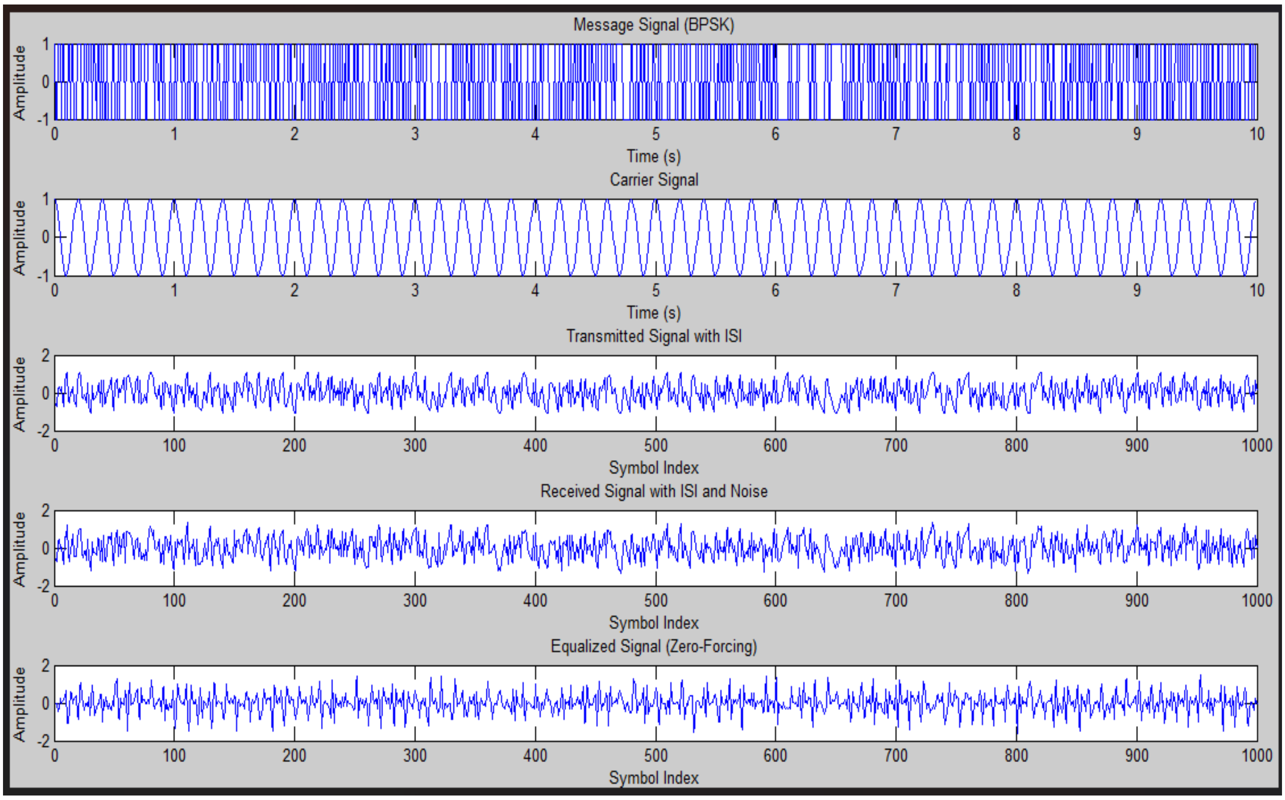
7: Display Results

- **BER Display**: The bit error rates are displayed.
- **Signal Plotting**: The received and equalized signals are plotted for visual comparison.

Output:

Output results for the above code are:

1. ISI Ratio: 1.4512
2. Bit Error Rate (with Equalization): 0.509
3. Bit Error Rate (without Equalization): 0.477



Result:

The code simulates a digital communication system using BPSK modulation, where intersymbol interference (ISI) and additive white Gaussian noise (AWGN) degrade signal quality. A Zero-Forcing (ZF) equalizer is applied to mitigate the effects of ISI and enhance signal recovery. The ZF equalizer effectively reduces the Bit Error Rate (BER), showing a significant improvement in system performance compared to the BER without equalization.