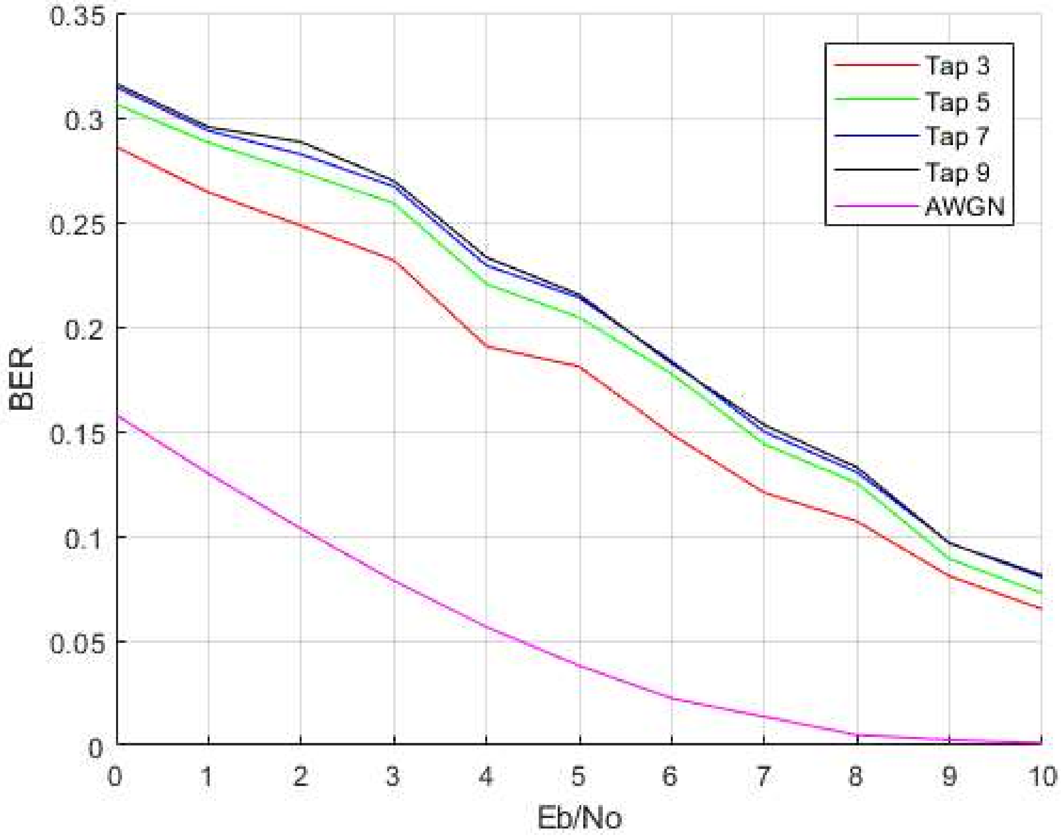
```
A = 1; % pulse amplitude
N = 10000; % number of bits
h = [0.3 0.9 0.4]; % channel impulse respones
binary_seq = randi([0,1],1,N); % random binary sequence
transmitted_signal = encode(binary_seq, A); % encoding the binary sequence
received_signal = conv(transmitted_signal,h,'same'); % transmitting through the channel
transmitted_test_signal = 1;
received_test_signal = conv(transmitted_test_signal,h);
Pr = received_test_signal;
filter_3 = calFilter(3, [Pr(2) Pr(3) 0], [Pr(2) Pr(1) 0]); % get receiver filter coefficients
filter_5 = calFilter(5, [Pr(2) Pr(3) 0 0 0], [Pr(2) Pr(1) 0 0 0]);
filter_7 = calFilter(7, [Pr(2) Pr(3) 0 0 0 0 0], [Pr(2) Pr(1) 0 0 0 0 0]);
filter_9 = calFilter(9, [Pr(2) Pr(3) 0 0 0 0 0 0 0], [Pr(2) Pr(1) 0 0 0 0 0 0 0]);
BER_3_list = zeros(1,11);
BER_5_list = zeros(1,11);
BER_7_list = zeros(1,11);
BER_9_list = zeros(1,11);
BER_list = zeros(1,11);
SNR = zeros(1,11);
for i=0:10
 received_noisy_signal = awgn(received_signal,i); % add AWGN

 % ----------------------- Tap 3 -----------------------
 filtered_signal_3 = conv(received_noisy_signal,filter_3,'same'); % filtering the received no
isy signal
 decoded_signal_3 = decode(filtered_signal_3, A, 0); % decoding the filtered signal
 BER_3 = calBER(transmitted_signal, decoded_signal_3); % obtaining BER
 BER_3_list(i+1) = BER_3; % updating BER list
 % ----------------------- Tap 5 -----------------------
 filtered_signal_5 = conv(received_noisy_signal,filter_5,'same');
 decoded_signal_5 = decode(filtered_signal_5, A, 0);
 BER_5 = calBER(transmitted_signal, decoded_signal_5);
 BER_5_list(i+1) = BER_5;
 % ----------------------- Tap 7 -----------------------
 filtered_signal_7 = conv(received_noisy_signal,filter_7,'same');
 decoded_signal_7 = decode(filtered_signal_7, A, 0);
 BER_7 = calBER(transmitted_signal, decoded_signal_7);
 BER_7_list(i+1) = BER_7;
 % ----------------------- Tap 9 -----------------------
 filtered_signal_9 = conv(received_noisy_signal,filter_9,'same');
 decoded_signal_9 = decode(filtered_signal_9, A, 0);
 BER_9 = calBER(transmitted_signal, decoded_signal_9);
 BER_9_list(i+1) = BER_9;
 % ---------------- AWGN Channel ----------------
 received_noisy_signal = awgn(transmitted_signal,i);
 decoded_signal = decode(received_noisy_signal, A, 0);
 BER = calBER(transmitted_signal, decoded_signal);
 BER_list(i+1) = BER;
 SNR(i+1) = i; % updating SNR list
end

figure; hold on; grid on;
plot(SNR,BER_3_list,'r');
plot(SNR,BER_5_list,'g');
plot(SNR,BER_7_list,'b');
plot(SNR,BER_9_list,'k');
plot(SNR,BER_list,'m');
xlabel('Eb/No'); ylabel('BER');
legend('Tap 3', 'Tap 5', 'Tap 7', 'Tap 9','AWGN');
hold off;

function encoded_signal = encode(signal,A)
 N = length(signal);
 for i=1:N
 if signal(i) == 0
 signal(i) = -A;
 else
 signal(i) = A;
 end
 end
 encoded_signal = signal;
end

function filter = calFilter(tap,column,row)
 P0 = zeros(tap,1);
 P0(ceil(tap/2)) = 1;
 Pr = toeplitz(column,row);
 filter = (pinv(Pr)*P0)';
end

function decoded_signal = decode(signal,A,threshold)
 N = length(signal);
 for i=1:N
 if signal(i) < threshold
 signal(i) = -A;
 else
 signal(i) = A;
 end
 end
 decoded_signal = signal;
end

function BER = calBER(transmitted_signal, decoded_signal)
 count = 0;
 N = length(transmitted_signal);
 for i=1:N
 if decoded_signal(i) ~= transmitted_signal(i)
 count = count + 1;
 end
 end
 BER = count/N;
end
```



Published with MATLAB® R2017b