5-th practical exercise
Matched Filter Receiver for Binary Shift Keying Signals

<u>The purpose</u> of the work is to create to perform reception of the signals modulated by BASK, BFSK and BPSK. This exercise adds matched filter receiver to programs already created during the 3-rd practical exercise.

The report should be prepared and sent to email address: elans.grabs@rtu.lv. The report must include the following:

1. The objective of practical exercise;
2. The full source code of simulation programs (3 programs);
3. The plots obtained during simulation (for each modulation, with and without noise).

<u>The tasks to be solved</u>:

1. Add optimal matched filter receiver to each of the 3 previously created programs (in 3-rd practical exercise).
2. In each plot (as subplot of 2 or 3 figures), You must show:
   a. Received noisy signal;
   b. Matched filter (or 2 matched filters) signal.
3. Perform detection and output the received data in command line. Compare result with originally selected data vector sB.
4. Analyze influence of noise power (SNR ratio) on accuracy of detection.

<u>The guidelines for practical exercise</u>

1. The first step is to specify the impulse response for each Matched filter. There is 1 matched filter for BASK and BPSK signals, and two matched filters for BFSK modulation type. The impulse response must be set to “mirrored” carrier signal s0. In Matlab, this can be achieved by using fliplr( ) (for vector-row) or flipud( ) (for vector-column) function:

```
hMF1 = fliplr(s0);
```

2. Then the matched filtering procedure must be performed. Note, that here You have 2 options, both can be achieved by Octave built-in functions, and produce the <u>same result</u> in the end.

1. Use “***filter***( )” function. For this function, the filter transfer function must be specified. In this task, FIR filter is designed, so numerator coefficients are the same as impulse response, and denumerator is just 1. So, the command would be:

```
sMF1 = filter(hMF1, 1, sBASK);
```

2. Alternatively, You can use “conv( )” function, which calculates convolution for two discrete sequences. This will produce longer signal, than necessary, so sometimes it may be necessary to truncate output signal. The convolution for digital filter is calculated between input signal and filter impulse response as follows:

```
sMF1 = conv(hMF1, sBASK);
sMF1 = sMF1(1:length(sBASK));     % Truncating length
```

3. Please, note that You need to pick only 1 method of two proposed above, and the result will be the same for both cases. The filtered signal needs to be sampled by the length of impulse response L to read signal energy levels.

```
Es = sMF1(L:L:end);          % Reading 1 sample per L values (at end)
```

These energy levels are then converted into binary symbols, depending on modulation type.

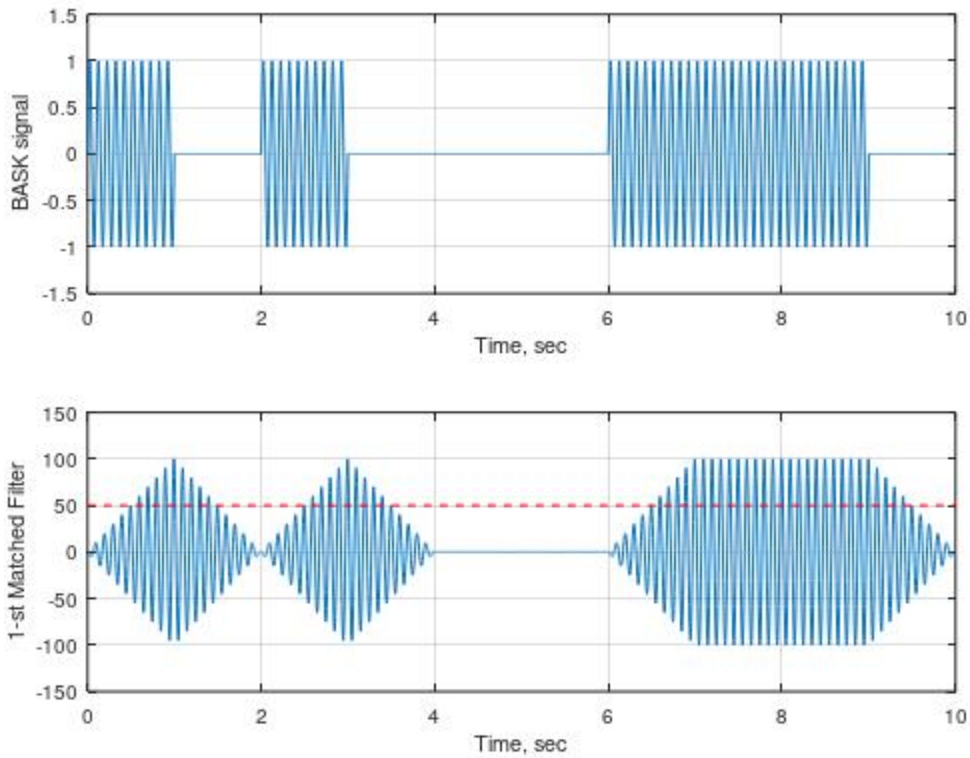
4. Use subplot(211) and subplot(212) to draw plots for modulated signal sBASK and matched filter output signal sMF1.

5. On the second subplot, You must draw threshold value (if there is any). For BASK case, the threshold is $E/2$, where E is the carrier signal energy (calculated before for noise power estimation).

6. Don't forget, You can draw multiple plots simultaneously in a plot( ) command:

```
plot(x1, y1, x2, y2, 'r');
```

As a result, plots similar to what is shown below must be obtained:



7. Use *if* function to perform comparison with threshold and set received symbol values.

8. Obtain results in a similar way for BFSK and BPSK modulation types. Note, that in case of BFSK You must calculate 2 matched filter output signals!