521273S Biosignal Processing I

Assignment 3. Adaptive Filtering

Deadline: Friday November 16th at 10.00

Learning outcomes

After this assignment student can

- use adaptive filter to remove interfering signal summed with the desired signal

In order to pass the exercise, all correctly executed task results must be personally presented to a course assistant during the scheduled lab consulting times.

Background

Read chapter 3.9 from the course book.
Participate in the lecture on Tuesday November 13th.

The problem is defined as follows: we would like to measure a signal of interest. However, there is an interfering signal component, that is summed with the desired signal. How could we get rid of the interfering signal component? If the desired and interfering signal components are in the same frequency band, we cannot apply simple FIR-filtering, because we would lose both signals. Instead, we can use adaptive filtering to solve the problem. To accomplish this, we measure two signals simultaneously: the actual measurement signal (*primary input*) and an additional signal (*reference signal*), which is correlated with the interfering signal component. We try to subtract the reference signal from the measurement signal, but we need to modify it, so that the filter output resembles the interference component as much as possible. Adaptive filter can do all this as it adapts automatically to the interference signal to subtract it even if it is nonstationary (characteristics change over time).

A digital adaptive filter is basically constructed by two main components, a digital FIR filter with adjustable coefficients, and an adaptation algorithm which modifies the coefficients in order to optimize the filter. We would like to find out an optimum estimate of the interfering signal component in the measurement signal in order to eventually subtract it.

Figure 1 depicts a schematic diagram of an adaptive filter, where there is a primary observed signal $x(n)$, which is the sum of interference $m(n)$ and the signal of interest $v(n)$. The reference signal is $r(n)$, which correlates with the interfering component $m(n)$.

The aim of an adaptive filter is to find $y(n)$ as close as possible to $m(n)$. So, $v(n) \simeq x(n) - y(n)$ can be extracted. Therefore, if $y(n) \simeq m(n)$, then the output $e(n)$ would be approximately the signal of interest, $v(n)$.

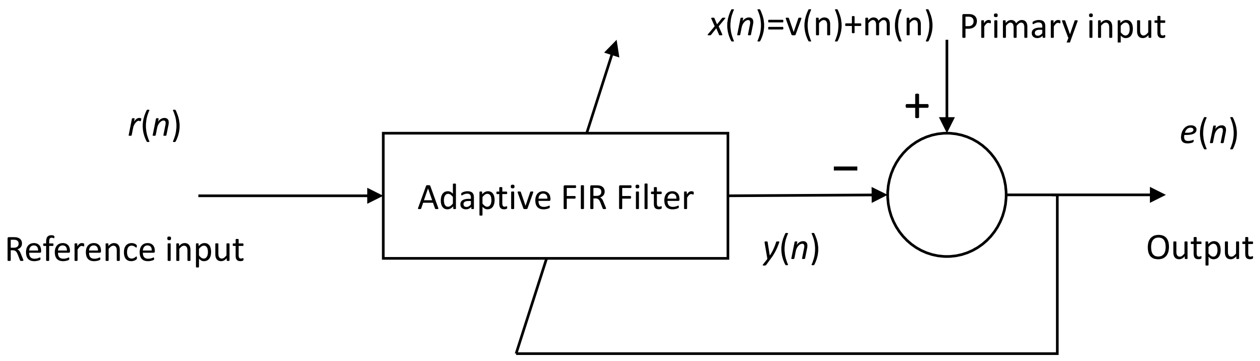

Figure 1. Schematic diagram of a generic adaptive filter

It can be proven, that the output $e(n)$ is the minimum mean square error (MMSE) estimate of the signal of interest $v(n)$, if the FIR filter coefficients are adjusted properly. By considering filter coefficients collected in vector $\boldsymbol{W}$, the filter output can be written in vector form as:

$$e(n) = x(n) - \boldsymbol{W}^{\boldsymbol{T}}(n)\boldsymbol{r}(n) \quad (1)$$

(Bold letters notate vectors.) Minimizing the energy of *e(n)* maximizes the signal-to-noise ratio (SNR) at the output. That happens when the interference component has been optimally subtracted from *x(n)*; which is exactly what we want! So the purpose is to

$$\text{Adjust tab coefficients} \longrightarrow \text{Minimize the MMSE}$$

Equation (1) can be rewritten as:

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + 2\mu e(n)\boldsymbol{r}(n) \quad (2)$$

This is the famous update rule of the least mean squares (LMS) adaptive filter: it modifies filter coefficients sample-by-sample in order to find optimal values. There is an additional parameter μ, which controls how big changes to the coefficients are allowed at each update step. It should be selected by experimentation, but to guarantee stability, the allowed range of μ is (for a FIR filter with *M* coefficients):

$$0 < \mu < \frac{1}{\sum_{n=1}^{M} r^2(n)} = \mu_{max} \qquad (3)$$

Therefore, the maximum allowed value of μ must be less than the inverse of the signal energy of the reference signal inside the FIR filter at any time. For simplicity, equations (2) and (3) can be combined as follows (with *0<c<1*, defined experimentally):

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + 2\frac{c}{\sum_{n=1}^{m} r^2(n)} e(n)\boldsymbol{r}(n) \qquad (4)$$

The simplicity of implementation of LMS algorithm is an advantage compared to other adaptive filters. **In this Labwork, we use Equation (2) and the schematic diagram (Fig. 1) of the filter in the figure for implementation of LMS adaptive filters.**

Data

Download the material from Noppa. Load the *signals.mat* file into MATLAB workspace. The data pack contains all the signals of the assignment. The given signals and variables are the following:

- Fetus signal (***`fhb`***): pure fetus signal that is used to assess analysis results (F_S = 1 kHz)
- Mother's chest signal (***`mhb`***): mother's signal, used to assess analysis results (F_S = 1 kHz)
- Abdomen signals (***`abd_sig1`***, ***`abd_sig2`*** and ***`abd_sig3`***): mixed signals, containing both fetus and mother signals (F_S = 1 kHz), used in cases **A**, **B**, and **C**.
- A real respiration movement (***`RespReference`***) signal and R-to-R interval sequence from ECG signal (***`RRiInput`***), both sampled at 4 Hz, used in case **D**.

Design all the related vectors and functions with the respective sampling rate.

We have four different study cases: three of them (Case **A-C**) are synthesized (artificially mixed from two pure signals) and one (Case **D**) consists of real signals recorded in a clinic at the hospital.

Cases **A-C**: We consider mother's ECG to be the interference signal that sums with the fetus ECG signal after going through some unknown linear transformation. Due to the linearity of the transformation, the LMS is able to estimate the transform (the FIR filter) so that the interference estimate at the FIR output can be subtracted from the abdominal signal successfully. As a result, we obtain the fetus ECG signal, which is normally not measurable separately with simple equipment.

Case **D**: We consider respiration signal to be the interference signal that sums with the ECG signal. See lecture slides!

Useful MATLAB commands

`filter, for, plot, hold, dsp.LMSFilter, xlim, immse, corr`

Tasks

Write your solution as a MATLAB script (m-file)!

In every case (**A-D**), plot all asked signals in a single figure with 3x1 subplot matrix.

In cases **A-C**: Limit the *x-axis* of all the subfigures to 10 seconds using time interval from 0.0 to 10.0 seconds of the signals. In other words, *only plot the first 10 seconds*!

Note: Ensure that the axes of the plots are labeled in suitable units such as *seconds* and ECG amplitude in *mV.* Also remember to put titles on all plots!

1. Make a suitable time vector and plot the ***`mhb`*** (chest) signal with respect to time in the first subfigure. Notice, that we are aiming to find the ***`fhb`*** signal from the abdomen signals. For cases **A-C**, the mother's chest signal (interference) is the same.

- ***Case A (`abd_sig1`):***

2. Visualize the first abdomen signal (***`abd_sig1`***) in the second subfigure. According to the given schematic diagram in figure 1, abdomen signal acts as $x(n)$ and the chest recording of the mother as $r(n)$. In the ***case A***, the abdomen signal is synthesized **simply by adding the maternal chest signal to the fetus signal**.

   Notice the extra peaks in the abdomen ECG signal of the mother. They are from the fetus ECG signal.

   Now familiarize yourself with the LMS adaptive filtering function of Matlab, ***`dsp.LMSFilter`***. We are going to use the `'Method', 'LMS'` parameter pair, and adjust the `'Length'`, and `'StepSize'` parameters of the filtering to find the best results.

   ```
   my_lms_filter = dsp.LMSFilter('Method', 'LMS' ...
       , 'Length', filter_length, 'StepSize', mu_value);
   ```
   constructs your filter, and

   `[y, e] = my_lms_filter(r, x);` will give you *y* and *e* from *r* and *x* as noted in Figure 1.

3. Find experimentally the best *m* (filter length) and *c*.
   - Consider different adaptive filter lengths (order of FIR filter or the number of coefficients) to find an appropriate length for the filter. The range of values [1, 5, 11, 15, 21] should be tested.
   - In the function you should define LMS step size (μ). Set the value of μ as *c/Energy*, where *Energy* is 186 in *Case **A*** and 0<*c*<1 (to be experimented).
   - After defining the LMS adaptive filter parameters, apply it to *Case **A*** signals.

   To evaluate the quality of your estimation

   - Plot the fetus signal (***`fhb`***) in blue color in the third subfigure. Plot on top the output of LMS adaptive filter (fetus interference estimate) in red color in the same subfigure.
   - Compute the correlation coefficient and MSE of the filter output with the ideal fetus signal (***`fhb`***). You can use ***`immse`*** and ***`corr`*** for computing these measures or implement them yourself. Notice, that the filter takes many iterations (depending on the μ parameter) to get to the optimal configuration, hence the initial samples of the output signal are not properly filtered. Evaluate the quality of the estimation with the first ***2 seconds*** removed from the fetus interference estimate and the pure fetus signal.
   - **As final result,** you should report the best correlation coefficient and MSE. Also, the third subfigure should now contain the best output of the filter.

- ***Case B (abd_sig2):***

4. As in step 1 above, make a new figure (***mhb*** is the same for cases **A-C**). Visualize the second abdomen signal ***(abd_sig2)*** in the second subfigure. Notice that in this abdomen signal, the maternal ECG is reversed. You can now observe how adaptive filtering can tackle also such cases with ease.

5. As in step 3 above, find the optimal parameters with computing MSE and the correlation coefficient, and filter the signal. Set the value of μ as *c/Energy*, where *Energy* is 186 in *Case **B*** and 0<*c*<1 (to be experimented). Plot the fetus signal (***fhb***) in blue color in the third subfigure. Also, plot the output of LMS adaptive filter in red color in the third subfigure.

- ***Case C (abd_sig3):***

6. As in step 1 above, make a new figure. Visualize the third abdomen signal ***(abd_sig3)*** in the second subfigure. Notice, that in this abdomen signal the maternal ECG signal is low pass filtered.

7. As in step 3 above, find the optimal parameters with computing MSE and the correlation coefficient, and filter the signal. Set the value of μ as *c/Energy*, where *Energy* is 186 in *Case **C*** and 0<*c*<1 (to be experimented). Plot the fetus signal (***fhb***) in blue color in the third subfigure. Also, plot the output of LMS adaptive filter in red color in the third subfigure.

- ***Case D (real signals):***

Now, conduct experiments with real signals. You should do this part when you already understood and successfully completed tasks 1-7.

8. Plot the ***RRiInput*** and ***RespReference*** with respect to time in the first and second subfigures, respectively.

9. As in step 3 above, find the optimal parameters with computing MSE and the correlation coefficient, and filter the signal. Evaluate the quality of the estimation with the first ***25 seconds*** removed from the reference signal and respiratory component. Set the value of μ as *c/Energy*, where *Energy* is 93 in *Case **D*** and 0<*c*<1 (to be experimented). Plot the ***RespReference*** in blue color in the third subfigure. Also, plot the estimated respiratory component from LMS adaptive filter in red color in the third subfigure. ***Hint:*** the estimated respiratory component is the interference (*y*), and not the filter output (*e*)!

Congratulations! You are done with the compulsory task assignment! But, you can continue and learn more by proceeding to the following additional tasks.

<u>Additional tasks (if you wish to learn more):</u>

Implement an adaptive filter from scratch, using only bare MATLAB operations (not built-in functions for adaptive filtering).

As explained in the introduction, adaptive filter learns to subtract an interference signal from a measurement signal by adapting to the interference signal sample-by-sample using the update rule. However, adaptive filter coefficients need be initialized properly in order to yield the best performance. There are two basic types of initialization that are often used:

- <u>On-line processing</u>: Initialize the coefficients with zero values or set them to small random values in the range `[-1,1]`. Filter coefficients are then updated with the update rule when new signal samples are received. In this case, the filter takes many iterations (depending on the μ parameter) to get to the optimal configuration, and hence the initial samples of the output signal are not properly filtered.
- <u>Off-line/batch processing</u>: Initialize the coefficients with zero values or set them to small random values in the range `[-1,1]`. Take a time reverse of the signals $x(n)$ and $r(n)$, so the first sample becomes the last one and vice versa. Run the LMS as usual. Now the FIR filter has learned the interference, and it is ready to be used in actual (forward) filtering. So, run the LMS filtering with the input signals in proper time order. This procedure can only be used, when taking the signals from memory, as you need to time reverse the signals. The advantage is, that the filter is optimally initialized already at the beginning of filtering! Note: the time reversal is a good idea when the interference is expected to be nonstationary, as with the adaptation the FIR filter remembers the signal behavior at the end, which turns out to be the beginning of the signal. However, if the interference is stationary, there is no need to time reverse the signals.

10. Implement the adaptive LMS filter in MATLAB.
   - Hint:

$$\boldsymbol{W^T(n)r(n)} = \sum_{\boldsymbol{k=0}}^{\boldsymbol{M-1}} \boldsymbol{w_k\, r(n-k)}$$

11. Initialize the FIR coefficients using the on-line approach. Repeat the above filtering for Cases **A-D** by using the best parameter values you identified earlier. Compute the correlation coefficient and MSE values.

12. Initialize the FIR coefficients using the off-line approach. Repeat the above filtering for Cases **A-D** by using the best parameter values you identified earlier. Compute the correlation coefficient and MSE values.