Pure sum of signals

521273S - Biosignal Processing I - Online Labs - Autumn 2024 > Assignment 3 - Adaptive filtering >

Pure sum of signals

0 solutions submitted (max: Unlimited)

Open Problem in MATLAB Online

Background

Consider that we are observing a mixture of two signals

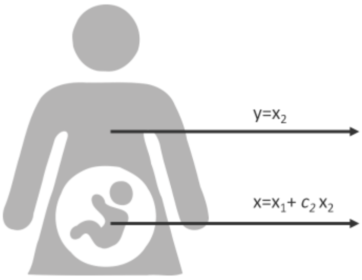
$x(t) = c_1x_1(t) + c_2x_2(t) = \sum_{i=1}^{2} c_ix_i(t)$. (1)

where $c_1 \neq 0$ and $c_2 \neq 0$ are mixing coefficients. In addition, we can measure the second signal directly as $y(t) = x_2(t)$ simultaneously.

Our task is to estimate $x_1(t)$ from $x(t)$ and $y(t)$. Since it is impossible to make a distinction between different values for the coefficients c_1 and signal $x_1(t)$ amplitudes, we can assume that without loss of generality that $c_1 = 1$. Thus, using the assumpltion that $c_1 = 1$ and $y(t) = x_2(t)$, we can rewrite (1) as

$x(t) - c_2y(t) = x_1(t)$. (2)

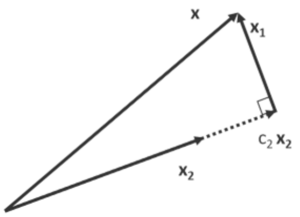
In this case, these signals are ECGs coming from a pregnant mother and the fetus. In the measurements, we can obtain the mother's ECG from the chest separately as a reference, but the abdominal measurement aimed to measure fetus' ECG is contaminated with that of the mother's as depicted in the Figure below.



For the numerical solution, we consider that we have N samples with sampling time interval T of the signals collected into corresponding vectors, i.e. $\mathbf{x} = [x(0)x(T)x(2T) \cdots x((N-1)T)]^T$, $\mathbf{x_1} = [x_1(0)x_1(T)x_1(2T) \cdots x_1((N-1)T)]^T$, and $\mathbf{y} = [y(0)y(T)y(2T) \cdots y(N-1)T)]^T$. Moreover, we are looking for optimal estimation in the l2-norm sense. Hence, we choose the value of c_2 that makes the vectors $\mathbf{x}$ and $\mathbf{y}$ most similar with each other, i.e. we minimize

$\min_{c_2} ||\mathbf{x} - c_2\mathbf{y}||_2$. (3)

According to (2), then also $||\mathbf{x_1}||_2$ is minimized. Geometrically, due to the triangle inequality, this means that $\mathbf{x_1}$ and $\mathbf{x_2}$ are orthogonal, i.e. $\mathbf{x_1}^T\mathbf{x_2} = 0$. See the Figure below.



Please also note that orthogonality means that $\sum_{i=0}^{N-1} x_1(iT)x_2(iT) = 0$. Thus, x_1 and x_2 are also uncorrelated if either (or both) of their means are zero.

We can compute c_2 via the scalar projection as

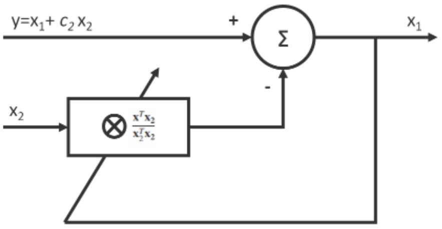
$c_2 = \frac{\mathbf{x}^T\mathbf{x_2}}{\mathbf{x_2}^T\mathbf{x_2}}$, (4)

and, $\mathbf{x_1}$ as the difference (vector rejection)

$\mathbf{x_1} = \mathbf{x} - c_2\mathbf{x_2} = \mathbf{x} - \frac{\mathbf{x}^T\mathbf{x_2}}{\mathbf{x_2}^T\mathbf{x_2}}\mathbf{x_2}$. (5)

Relation to adaptive filtering

Now, note that the solution (5) corresponds to the signal processing filter structure shown in the Figure below.



Basically, we have here a one tap (adaptive) filter whose coefficient is matched by comparing the input signals in the l2-sense minimizing $\mathbf{x_1}$ which leads to minimizing the difference between $\mathbf{x_1}$ and $c_2\mathbf{x_2}$.

Finally, please also note that (3) is the least squares formulation of the matrix equation

$\mathbf{x} = c_2\mathbf{y} = c_2\mathbf{x_2}$, (6)

where the vectors $\mathbf{x}$ and $\mathbf{y} = \mathbf{x_2}$ are Nx1 matrices. Hence, we can also obtain c_2 via the Moore-Penrose pseudoinverse $c_2 = \mathbf{x_2}^+\mathbf{x}$.

[Delayed sum of signals]

Delayed sum of signals

0 solutions submitted (max: Unlimited)

Background

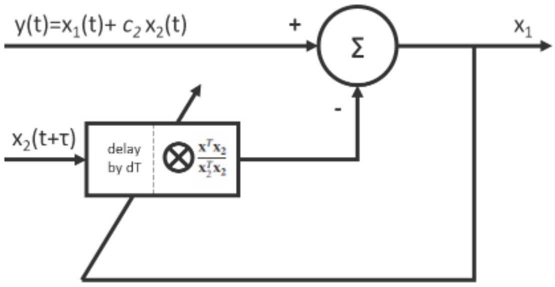
Consider now the same basic setting as in the previous problem.

This time, however, let us suppose that there is a small time delay between the signals so that the mothers signal in the abdomen -- that is x_2 in $x(t) = x_1(t) + c_2 x_2(t)$ -- is delayed compared to the chest by d samples (**integer!**). We denote this as $y(t) = x_2(t+\tau) = x_2(t+dT)$ where $\tau = dT$ is the amount how much a head of time the chest signal is.

Clearly, if we know the time difference, we time shift (delay) the chest signal, and then use the l2-norm based optimal rejection/cancellation method of the previous problem. For example, cross-correlation can be used to find the time delay at which the signals correlate with each other the most.

Relation to adaptive filtering

Now, note that the solution of the problem corresponds to the signal processing filter structure shown in the Figure below.



Basically, we have here a multitap (adaptive) filter whose all other coefficients are zero, but the one matching the delay is found optimally as in the first problem. The delay can be estimated using cross correlation.

Delayed sum of signals - subsample accuracy

Delayed sum of signals - subsample accuracy

1 solution submitted (max: Unlimited) | View my solutions

Open Problem in MATLAB Online

Background

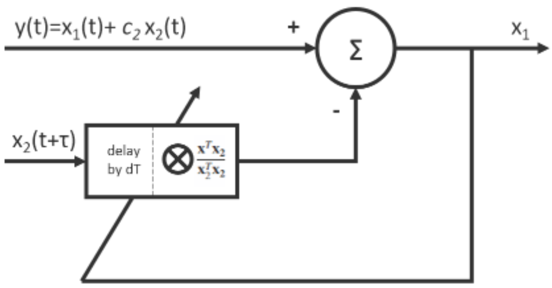
Consider now the same basic setting as in the previous problem. This time, however, let us suppose that there is a small time delay between the signals so that the mothers signal in the abdomen -- that is x_2 in $x(t) = x_1(t) + c_2x_2(t)$ -- is delayed compared to the chest by d samples (**non-integer!**). We denote this as $y(t) = x_2(t+\tau) = x_2(t+dT)$ where $\tau = dT$ is the amount how much a head of time the chest signal is.

Clearly, if we know the time difference, we can time shift (delay) the chest signal, and then use the l2-norm based optimal rejection/cancellation method of the first problem. The cross-correlation can be used to find the time delay at which the signals correlate with each other the most.

This time, however, we cannot go ahead and just shift the samples due to non-integer delay. To do the shifting, we must use interpolation. In this exercise, we use spline interpolation in which the values between the samples are approximated using piece-wise polynomials fitted to the neighbouring data points. If the polynomial to be fitted is of the first order, we have linear interpolation wherein the value in-between two given samples can be computed as a weighted average of the the closest neighboring samples on the both sides without information about samples further away. For example, the value directly in between two samples is just the average of the said samples. However, to better represent band limited sampled signals, we use third order polynomials and hence cubic spline interpolation.

Relation to adaptive filtering

Now, note that the solution of the problem corresponds to the signal processing filter structure shown in the Figure below.



Basically, in the linear interpolation case, we have here a multitap (adaptive) filter where only two of the coefficients are non-zero, and those two contain the optimal scale c_2 divided between them in the ratio how close the non-interger part of the time shift is to those lags. For example, if the shift were 1.5 samples ($d = 1.5$), the 2nd and 3rd coefficients of the filter would both be $\frac{c_2}{2}$ all the others being zeros. For the more generic spline interpolation, there will be more non-zero coefficients depeding on the order of the spline.

[Full adaptive filtering]

521273S - Biosignal Processing I - Online Labs - Autumn 2024 > Assignment 3 - Adaptive filtering >

Full adaptive filtering

0 solutions submitted (max: Unlimited)

Open Problem in MATLAB Online

Background

We have already seen that adaptive filter constructions can handle signal amplitude differences and time delays. This time, let us suppose that, in addition there is also some power line interference in the maternal ECG recorded from the chest.

We use the LMS-filter to adapt to the signal, and to cancel out the maternal ECG from the fetal ECG. Chapters 3.6.1 and 3.6.2 of the course book show the derivation of the filter and its learning rule.

Adaptive LMS-filter

A digital adaptive filter is basically constructed by two main components including a digital FIR filter with adjustable coefficients and an adaptation algorithm which modifies the coefficients in order to optimize the filter. In MATLAB, the DSP System Toolbox includes the dsp.LMSFilter object that implements the classic LMS-filter among others.

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

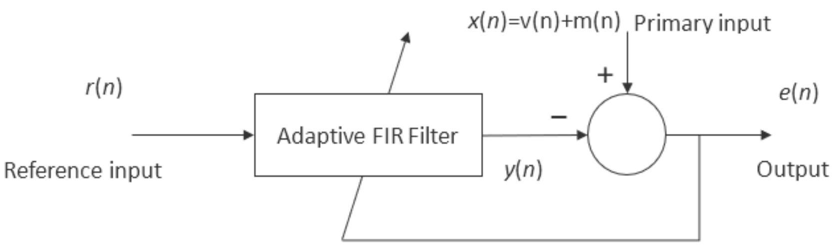


Figure 1. Schematic diagram of a generic adaptive filter

The aim of an adaptive filter is to find $y(n)$ as close as possible to $m(n)$. So, $v(n) \approx x(n) - y(n)$. Since $y(n) \approx m(n)$, the output would approximately be the signal of interest $v(n)$.

It can be proved 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 according to the famous update rule of the LMS filter

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n) r(n) \quad (1)$$

where μ is *the step size* of the algorithm controlling how big changes are allowed at each update step. The value of μ should be selected by experimentation, but to guarantee stability the allowed range of μ is (for a FIR filter with m coefficients)

$$0 < \mu < \frac{1}{\sum_{n=1}^{m} r^2(n)} =: \mu_{max}. \quad (2)$$

It is often practical to select μ as a fraction $(0 < c < 1)$ of μ_{max}, i.e. $\mu = c\mu_{max}$.

In contrast to the course book, the step size of MATLAB dsp.LMSFilter includes the factor 2 in (1). Taking this into account, the step size can be set as

$$step = 2\mu = 2c\mu_{max} = \frac{2c}{\left(\frac{1}{m}\sum_{n=1}^{m} r^2(n)\right)m} \quad (3)$$

showing that the selected fraction, signal power, and filter length are the factors needed to determine the step size.

The maxstep function can be used to estimate the maximum step size, but for this exercise, we use $\mu_{max} m = \left(\frac{1}{m}\sum_{n=1}^{m} r^2(n)\right)^{-1} = 0.05$ for all m. To get the step size for each parameter combination, multiply it by $2c$ and divide by m.