E5ADSB Exercise 1 – System Identification

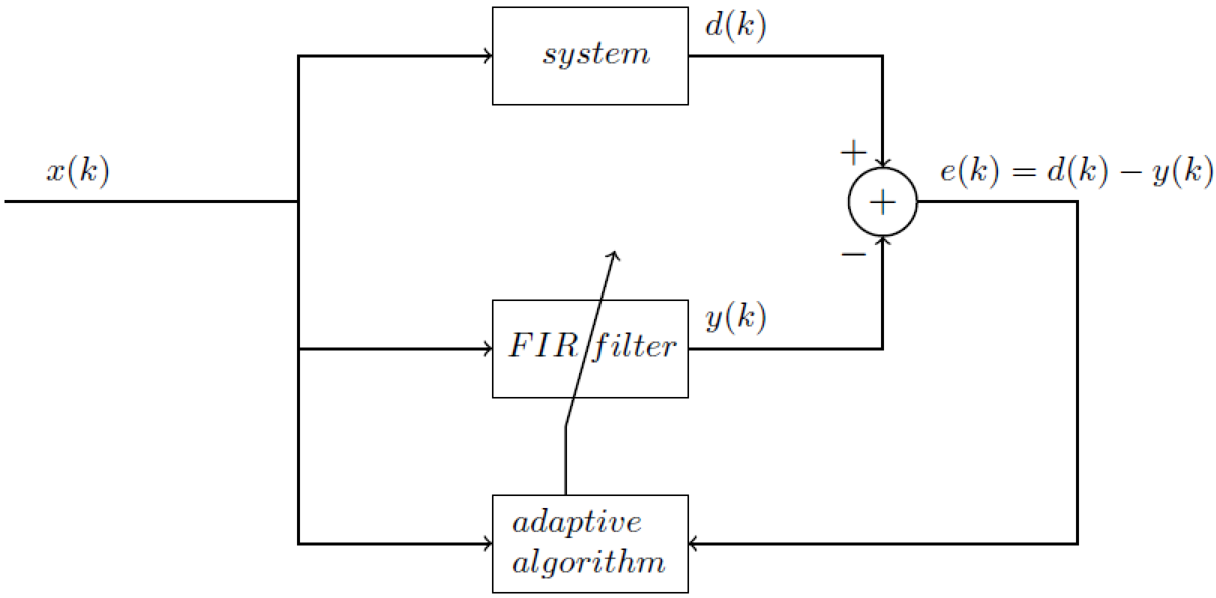
– introduction to the LMS filter

KPL, 2017-08-02

This first exercise is meant as an appetizer to adaptive filtering. The purpose is to implement an adaptive algorithm (the LMS algorithm) in Matlab® and see how it performs. If you chose to work with adaptive filters in the course, you'll have to look into the underlying theory.

In system identification, which is sometimes called system modelling, unknown parameters of a system are to be found. To keep it "simple", we'll assume that the system can be modelled as an FIR filter.

Consider the following setup, where the parameters in "*system*" have to be identified.



The idea behind system identification is, that both the unknown system and the FIR filter (the model) are excited with the same input signal (or sequence, since we are talking discrete-time signals), $x(k)$. The output of the system is called the *desired signal,* $d(k)$, and the output of the FIR filter is called $y(k)$. If the FIR filter represents a good model of the system, the *error signal,* $e(k) = d(k) - y(k)$, will be small (close to zero), but if the model is bad, $e(k)$ will be large. The error signal is send back to an *adaptive algorithm*, which tries to minimize the error by adjusting the filter coefficients in the FIR filter. After a number of *iterations*, the FIR filter coefficients will converge to the set resulting in the smallest possible error given some fixed length of the FIR filter.

Let the system be given by the following system function (transfer function)

$$H(z) = 0.67 + 0.21z^{-1}$$

With corresponding difference equation

$$d(k) = 0.67x(k) + 0.21x(k-1)$$

1. Generate the signal $x(k)$ in Matlab using the function `randn`.
2. Generate $d(k)$ using Matlab's `filter` command.

The length of the FIR filter must be decided before starting the adaptive algorithm. In this case we already know the length, M=2, but very often in practice, that is not the case. It is usually a good idea to choose some small value for M, e.g. 2 or 3 coefficients, and then increase the number, if the model doesn't become very good. The initial values of the coefficients are normally chosen to be zero, unless you have a better guess.

The adaptive algorithm that we'll use is the LMS algorithm [Diniz, Algorithm 3.1, p. 80]

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\mu e(k)\boldsymbol{x}(k)$$

In the equation $\boldsymbol{w}(k)$ is a vector that holds the FIR filter coefficients at time k, $e(k)$ is the error at time k, and $\boldsymbol{x}(k)$ is a vector that holds the current and the previous M-1 input samples – that is: the FIR filter's delay line. The length of $\boldsymbol{x}(k)$ equals the length of $\boldsymbol{w}(k)$ and the length is denoted by M (filter length).

$$\boldsymbol{w}(k) = \begin{bmatrix} w_0(k) \\ w_1(k) \\ \vdots \\ w_{M-1}(k) \end{bmatrix}$$

$$\boldsymbol{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-(M-1)) \end{bmatrix}$$

Note that $\boldsymbol{x}(k)$ and $\boldsymbol{w}(k)$ are column vectors of size M×1. The factor μ is called the step-size (or sometimes the convergence factor), and it controls the step size of each adjustment of the coefficients. μ should initially be set to a number much smaller than 1, e.g. $\mu = 10^{-3}$.

3. Implement the LMS algorithm. That is, write a loop that adapts the filter coefficients using the LMS algorithm. Store $e(k)$ and $\boldsymbol{w}(k)$ in arrays, so you can see how they have evolved after the code finishes. The tricky part in this implementation is the ordering of data in the arrays. $x(k)$ is the newest sample in the vector $\boldsymbol{x}(k)$, and it should be in the top entry. Matlab's function `flipud` (flip up-down) might come in handy here.
   Note: the error samples $e(k)$ can be calculated as
   $$e(k) = d(k) - y(k) = d(k) - \sum_{m=0}^{M-1} w_m x(k-m) = d(k) - \boldsymbol{w}^T\boldsymbol{x}(k)$$
4. Plot $e^2(k)$ and $\boldsymbol{w}(k)$ and discuss whether the adaptive filter works as intended.
5. Try different values of the step-size μ and the filter length M, and see how good a system identification you can obtain.

6. ☺ BONUS PART! ☺ Let's try modelling an IIR system by an FIR filter ☺
   Now let the system be given by the following system function (transfer function)

$$H(z) = \frac{1}{1 + 0.25z^{-1} - 0.5z^{-2}}$$

   With corresponding difference equation

$$d(k) = x(k) - 0.25d(k-1) + 0.5d(k-2)$$

   Repeat tasks 1-5 using this new "unknown" system (you should still try to model the system as an FIR system).

Good luck!

/Kristian Lomholdt