تکلیف سری پنجم فیلترهای وفقی

پیاده سازی آلگوریتم LMS

Problem 1: Computer Experiment on Adaptive System Identification
One of the many uses of adaptive filters is for system identification as shown in Figure 1. In this configuration, the same input is applied to the adaptive filter and to an unknown system, and the coefficients of the adaptive filter are adjusted until the difference between the outputs of the two systems is as small as possible. After adaptation, the system function of the unknown system can be approximated by the system function of the adaptive filter. Adaptive system identification can be used to model a system who's parameters are slowly varying when the input and output signals are both available, for example in vibration studies of mechanical systems. In realistic situations, however, the output of the unknown system will generally be distorted by additive noise. In the first exercise, we look at how adaptive system

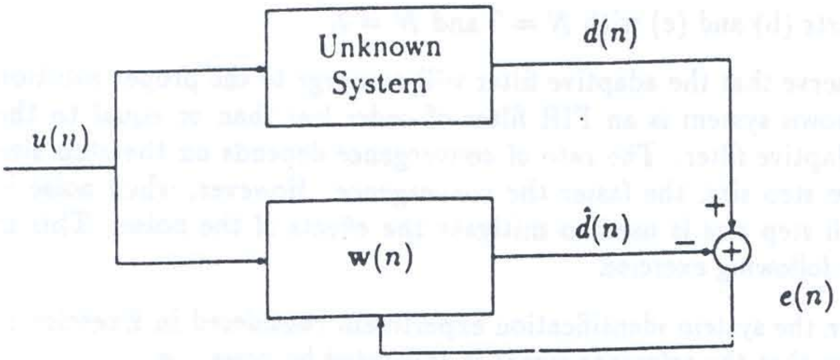


Figure 1: Adaptive system identification for modeling the output $d(n)$ of the unknown system.

identification works in an ideal setting. More realistic applications are then considered in Exercises 2 and 3.

Exercise 1 Let an unknown system be an FIR filter with the following unit sample response,

$$h(n) = \delta(n) + 1.8\delta(n-1) + 0.81\delta(n-2)$$

Using an input signal $u(n)$ consisting of at least 100 samples of unit variance white Gaussian (normal) noise, create the reference signal $d(n)$ by passing $u(n)$ through the filter. The white noise may be generated in MatLab using the rand command and the convolution may be performed using `conv`. Make sure to type `rand('normal')` first, in order to generate white noise. (You may type `help` command to get help on a command).

این روش قدیمی است. اکنون بهتر است از دستور randn استفاده کنید.

(a) Determine the range of values for the step size, μ, so that the adaptive filter will be convergent in the mean.

(b) Use the MatLab m-file `llms` (see attached listing) to implement the adaptive filter. Set the initial values for the filter coefficients to zero, set the step size to $\mu = 0.5$, and use an adaptive filter of order $N = 4$. Let the adaptive filter adapt and record the final set of coefficients.

(c) If the experiment in (b) is performed K times and the error on the kth trial is $\xi_k(n)$ after n iterations, then the mean squared error may be approximated using

$$E\{\xi(n)\} \approx \frac{1}{K}\sum_{k=1}^{K} \xi_k(n)$$

With $K = 5$, make a plot of the learning curve as a function of n using the approximation defined above. How many iterations are necessary for the mean square error to fall to 10% of its peak value?

(d) Calculate the excess mean square error and compare it to what you observe in your plot of the learning curve.

(e) Repeat parts (b) and (c) for $\mu = 0.1$ and $\mu = 0.05$.

(f) Repeat parts (b) and (c) with $N = 3$ and $N = 2$.

You should observe that the adaptive filter will converge to the proper solution when the unknown system is an FIR filter of order less than or equal to the order of the adaptive filter. The rate of convergence depends on the step size – the larger the step size, the faster the convergence. However, when noise is present, a small step size is used to mitigate the effects of the noise. This is explored in the following exercise.

Exercise 2 Consider the system identification experiment considered in Exercise 1, but now suppose that the reference signal is corrupted by noise, i.e.,

$$\tilde{d}(n) = d(n) + \gamma v(n)$$

where $v(n)$ is unit variance white Gaussian noise.

(a) With $\gamma = 1$, use the LMS algorithm to model the system. Set the initial values for the filter coefficients to zero, and use a step size of $\mu = 0.5$, and an adaptive filter of order 4. Let the filter adapt and record the final set of coefficients. Repeat your experiment using $N = 5$ and comment on your results.

(b) Repeat part (a) with $\gamma = 0.1$ and comment on how the accuracy of the model varies with the noise amplitude. Conduct some simple experiments to determine whether the modeling accuracy depends on the step size.

Exercise 3 An adaptive FIR filter may be used to model any unknown system. In this problem, we examine how well such a system can model an IIR system. The arrangement for the filter is the same as the one in Figure 1 that was used in the previous two exercises. Let the unknown system be an IIR filter with a system function

$$H(z) = \frac{1 + 0.5z^{-1}}{1 - 0.9z^{-1}}$$

Generate a 300-point Gaussian random sequence with unit amplitude using the **rand** function. Use this signal for the input to both the unknown system and to the adaptive filter.

(a) Use the MatLab m-file **llms** to implement the adaptive filter with $\mu = 0.3$ and $N = 4$. Record the final values of the filter coefficients. Plot the error function in the manner described in Exercise 1. How many iterations are necessary for the filter to converge?

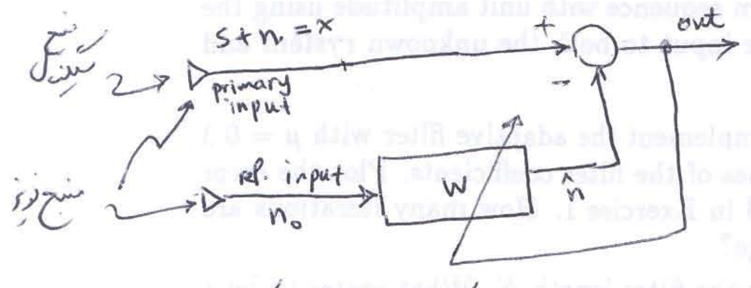
(b) Repeat (a) for different values for the filter length N. What seams to be a "reasonable" value to use?

(c) Add noise of amplitude A to the output of the unknown system and repeat (a) for $A = 0.1, 0.3, 1.0$.

```
function [A,E] = llms(x,d,alpha,nord,A_in)
%LMS                                                   [mhh3 3/92]
%--- (Widrow-Hoff LMS adaptive filtering algorithm)
% Adapted from lms.m written by j.mcclellan.
%
%      [A,E] = llms(x,d,alpha,nord,a0)
%            x     : input data to the adaptive filter.
%            d     : desired output vector
%            alpha: adaptive filtering update (step-size) parameter
%            nord : number of filter coefficients
%            a0    : (optional) initial guess for FIR filter coeffs
%                    (row vector). If a0 is omitted, a0=0 is assumed.
%      [A,E] = llms(X,D,ALPHA)  uses A_in=0
%
%      The output matrix A contains filter coefficients.
%         - The n'th row contains the filter coefficients at time n
%         - The m'th column contains the m'th filter coeff vs. time.
%         - The output vector E contains the error sequence versus time.
%
X=convolm(x,nord,'<');
[M,N] = size(X);
if nargin < 5,    A_in = zeros(1,N);    end
A_in = A_in(:).';
E(1) = d(1) - A_in*X(1,:).';
A(1,:) = A_in + alpha*E(1)*conj(X(1,:));
if M>1
   for k=2:M;
E(k) = d(k) - A(k-1,:)*X(k,:).';
A(k,:) = A(k-1,:) + ...
         alpha*E(k)*conj(X(k,:));
   end
end
```

مسئله ۳: در این مسئله کاربرد فیلترهای وفقی (الگوریتم LMS) در حذف نویز بررسی می‌شود (ANC).

سیستم زیر را در نظر بگیرید



ورودی primary در محلی قرار گرفته که هم سیگنال و هم نویز را دریافت می‌کند (مثل یک میکروفون درون اتاق یک اتومبیل). ورودی ref. input در محلی قرار گرفته که تنها نویز را دریافت می‌کند (مثل یک میکروفون که تنها صدای موتور اتومبیل را دریافت می‌کند). چون تابع تبدیل از محل نویز تا این دو سنسور متفاوت است، نویز در pr. in. با نویز در ref. in. یکی نیست، ولی تحت یک تابع تبدیل مجهول یکی می‌شود. پس هدف تخمین W است به نحوی که $\hat{n}$ و n یکی شده و در نتیجه اثر نویز در output حذف شود. در درس دیدیم که حداقل کردن انرژی out معادل حذف شدن نویز است. در نتیجه مدل یک Adaptive Filter معمولی در اینجا برقرار است که در آن $d(n)=s(n)+n(n)=x(n)$ و $u(n)=n_0(n)$ و $e(n)=out(n)$.

الف) یک نویز تصادفی با واریانس 1 تولید کنید (با استفاده از randn) و آن را n_0 بنامید، که همان ورودی ref. input است.

ب) فرض کنید $H(z)$ تابع تبدیل از محل نویز (n_0) تا ورودی primary باشد (که مجهول است). با استفاده از دستور rand یک FIR $H(z)$ مجهول با طول $N=5$ با ضرایب random تولید کنید و سیگنال $n(n)=[H(z)]n_0(n)$ (نویز در محل primary input) را محاسبه کنید.

ج) یک سیگنال صحبت 8KHz را به عنوان s برداشته و با دستورات زیر میانگین آن را صفر و واریانس آن را 1 کنید.

s = s - mean(s);

s = s/std(s);

د) $x=s+n$ ورودی میکروفون primary را محاسبه کرده و آن را گوش کنید.

ه) با استفاده از الگوریتم LMS برای یک W با طول $N=10$، نویز را حذف کنید. پس از همگرایی، سیگنال out را گوش کنید و اثر الگوریتم را مشاهده نمایید. همچنین $\underline{W}$ تخمین زده شده را با $H(z)$ بکار رفته مقایسه کنید [توجه: چون در اینجا توان نهایی out کوچک نمی‌شود (چون سیگنال صحبت اصلی است) μ بکار رفته باید خیلی کوچک باشد، وگرنه الگوریتم واگرا می‌شود].

و) واریانس نویز تولید شده (n_0) را تا 10 برابر افزایش دهید و قسمت‌های د و ه را تکرار کنید.

ز) برای یک $H(z)$، IIR مثل $H(z)=\frac{1+0.5z^{-1}}{1-0.9z^{-1}}$ اثر الگوریتم را بررسی کنید.