EE254, Spring 2014 Prof. Essam Marouf
Adaptive Signal Processing

Term Project: Simulation Study of Example Adaptive Systems

Due May 12, 2014 (last day of instruction)

This term project enforces through completion of computer simulations better understanding of some of the foundation concepts in EE254. You are to select 3 out of the 4 example adaptive filters applications listed below and complete the indicated computer simulation problems taken from the two textbooks for the course. The 4 application areas are:

1- Adaptive System Identification: (LMS and NLMS algorithms)
   a- Hayes C9.2
   b- Hayes C9.3
   Read also F-B Sec 6.4.1 and examine related Matlab program he posted.

2- Adaptive Linear Prediction: (Comparison of LMS and RLS algorithms)
   Hayes C9.6

3- Adaptive Noise Cancellation, or Adaptive Line Enhancement: (NLMS algorithm)
   Hayes C9.9
   Read also F-B Sec 6.4.3 and examine related Matlab program he posted.

4- Adaptive Channel Equalization: (LMS algorithm)
   Study the example in Sec 9.2.9 of Hayes. Read also F-B Sec 6.4.2 and examine related Matlab program he posted. Use F-B's Matlab program 'equalizer.m. to reproduce the learning curves shown in his Figure 6.9. If you have the time, explore larger range of the adaption parameter m, and compare with theoretical predictions of the speed of convergence and misadjustment. Comment on your results.

Results of your work should be documented in a coherent final report to be submitted by the last day of lectures (Monday 5/12/2014). Think of your report as a way of conveying useful information to another colleague wanting to learn about the subject. Start it with a brief summary of the basics of adaptive filters, summary of the algorithms used and corresponding theoretical results used in your report, then for each of the 3 simulation cases addressed, give complete informative answers to questions asked, labeled plots of the results, comments on the results, and summary of what has been learned from completing the simulation.

Reports adopting "question/answer" type of format do not earn the best grades, despite being complete. Well-structured reports conveying more complete information and including meaningful discussion of the results, while still addressing all posed questions, earns better grades. Reports are graded *relatively*.

Commented Matlab programs should be appended at the end of the report. A CD or USB stick that includes all Matlab code and an electronic copy of the report should be attached to your final report (a report without a CD or equivalent is not acceptable and is not graded). You should work individually and submit an individual report. You're welcome to discuss general ideas with other students in the class, however, you're not allowed to exchange solutions or any Matlab code with others (neither provide nor receive).

The Honor code is strictly observed. Please see the Greensheet.

You can also pursue a project of your own choice, if you wish. It could be software or hardware oriented, but should address a DSP topic relevant to the subject matter of EE254. You need to submit by Wednesday 4/1/2014 a one page proposal that includes an outline of the tasks you plan to perform. It should be of scope not less than the structured project above. A good source of project ideas are Applications Notes published by various DSP vendors (e.g., Texas Instruments, Analog Devices, Motorola, ...), usually downloadable from their websites.

For convenience, you can use the links below to download the Matlab programs referenced in the two textbooks.

1) Hayes

http://users.ece.gatech.edu/~mhayes/stat_dsp/matlab.html

or see

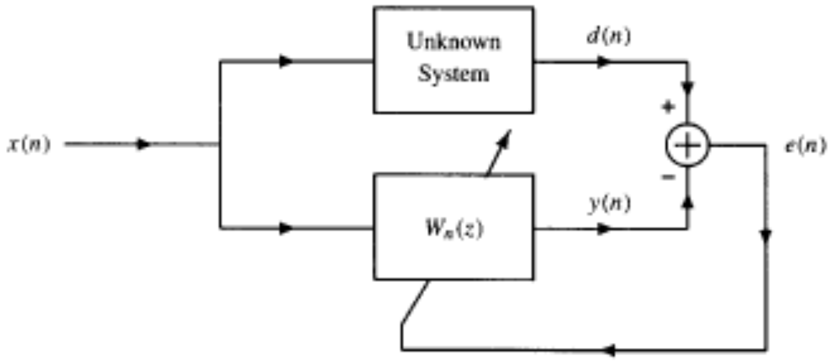
http://www.mathworks.com/matlabcentral/fileexchange/2183-statistical-digital-signal-processing-and-modeling

2) Farhang-Boroujeny (F-B)

www.wiley.com/go/adaptive_filters

Computer Simulation Problems from Hayes

C9.2. One of the many uses of adaptive filters is for system identification as shown in the figure below. In this configuration, the same input is applied to an 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. Adaptive system identification may be used to model a system with slowly-varying parameters, provided both the input and output signals are available.



Let the unknown system that is to be identified be an FIR filter with unit sample response

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

With an input $x(n)$ consisting of 1000 samples of unit variance white Gaussian noise, create the reference signal $d(n)$.

(a) Determine the range of values for the step size μ in the LMS algorithm for convergence in the mean.

(b) Implement an adaptive filter of order $p = 4$ using the LMS algorithm. Set the initial weight vector equal to zero, and use a step size of $\mu = 0.1\mu_{\max}$, where $\mu_{\max}$ is the largest step size allowed for convergence in the mean. Let the adaptive filter adapt and record the final set of coefficients.

(c) Repeat part (b) using the normalized LMS algorithm with $\beta = 0.1$, and compare your results.

(d) As described in Example 9.2.2, make a plot of the learning curve by repeating the experiment described in part (b) for 100 different realizations of $d(n)$, and plotting the average of the plots of $e^2(n)$ versus n. How many iterations are necessary for the mean-square error to fall to 10% of its peak value? Calculate the theoretical value for the excess mean-square error and compare it to what you observe in your plot of the learning curve.

(e) Repeat parts (b) and (d) for $\mu = 0.01\mu_{\max}$ and $\mu = 0.2\mu_{\max}$.

(f) Repeat parts (b) and (d) for $p = 3$ and $p = 2$.

C9.3. In this exercise, we consider the system identification experiment described in Exercise C9.2, but this time, we look at what happens if the reference signal is corrupted by noise. Instead of $d(n)$, suppose that we observe

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

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

(a) With an adaptive filter of order $p = 4$ and $\gamma = 0.1$, use the normalized LMS algorithm to model the system defined in Exercise C9.2. Use a step size of $\beta = 0.1$ and set the initial weight vector equal to zero. Let the filter adapt and record the final set of coefficients. Repeat your experiment using $p = 5$ and discuss your results.

(b) Repeat part (a) with $\gamma = 1$ and comment on how the accuracy of the model varies with the variance of the noise. Does the accuracy of your model depend on the step size?

C9.6. In this problem we will compare LMS and RLS for adaptive linear prediction. As in Example 9.2.1, let $x(n)$ be a process that is generated according to the difference equation

$$x(n) = 1.2728x(n-1) - 0.81x(n-2) + v(n)$$

where $v(n)$ is unit variance white Gaussian noise. The adaptive linear predictor will be of the form

$$\hat{x}(n) = w_n(1)x(n-1) + w_n(2)x(n-2)$$

(a) Implement an RLS adaptive predictor with $\lambda = 1$ (growing window RLS) and plot $w_n(k)$ versus n for $k = 1, 2$. Compare the convergence of the coefficients $w_n(k)$ to those that are obtained using the LMS algorithm for several different values of the step size μ.

(b) Make a plot of the learning curve for RLS and compare it to the LMS learning curve (See Example 9.2.2 on how to plot learning curves). Comment on the excess mean-square error for RLS and discuss how it compares to that for LMS.

(c) Repeat part (b) for exponential weighting factors of $\lambda = 0.99, 0.95, 0.92, 0.90$ and discuss the trade-offs involved in the choice of λ.

(d) Modify the m-file for the RLS algorithm to implement a sliding window RLS algorithm.

(e) Let $x(n)$ be generated by the time-varying difference equation

$$x(n) = a_n(1)x(n-1) - 0.81x(n-2) + v(n)$$

where $v(n)$ is unit variance white noise and $a_n(1)$ is a time-varying coefficient given by

$$a_n(1) = \begin{cases} 1.2728 & ; \quad 0 \le n < 50 \\ 0 & ; \quad 50 \le n < 100 \\ 1.2728 & ; \quad 100 \le n \le 200 \end{cases}$$

Compare the effectiveness of the LMS, growing window RLS, exponentially weighted RLS, and sliding window RLS algorithms for adaptive linear prediction. What approach would you propose to use for linear prediction when the process has step changes in its parameters?

C9.9. In this problem we look at the problem of adaptive noise cancellation without a reference (see Fig. 9.2*b*). Let

$$x(n) = d(n) + v(n)$$

where $d(n)$ is a unit amplitude sinusoid of frequency $\omega_0 = 0.01\pi$. Suppose that $v(n)$ is a moving average process that is generated as follows

$$v(n) = g(n) + 0.5g(n-2)$$

where $g(n)$ is unit variance white noise.

(a) What is the minimum value for the delay k_0 that may be used in the adaptive noise canceller?

(b) Write a MATLAB program for adaptive noise cancellation using the normalized LMS algorithm with $x(n-k_0)$ as the reference signal.

(c) Generate 1000 samples of $v(n)$ and $x(n)$ and use your MATLAB program to estimate $d(n)$ for filter orders $p = 5, 10, 15, 20$. Use values for k_0 that range from the minimum value determined in part (a) to $k_0 = 25$. What dependence on k_0, if any, do you observe? Explain. What is the effect of the filter order p? Given that the computational costs go up with p, what order do you think would be the best to use?

(d) Find the coefficients of an FIR Wiener filter to estimate and compare them with the steady-state values of the coefficients of the adaptive noise canceller.
