Assignment #5: NLMS, AP, LS and RLS algorithms

Use the same system and initial code as in assignment #4:

$$x(n) = u(n) * t(n)$$

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

$$e(n) = d(n) - y(n) = d(n) - \sum_{k=0}^{N-1} w_k(n) x(n-k) = d(n) - \mathbf{w}^H(n)\mathbf{x}(n)$$

with $\mathbf{w}(n) = \begin{bmatrix} w_0^*(n) & w_1^*(n) & \cdots & w_{N-1}^*(n) \end{bmatrix}^T$

$$T(z) = \frac{1}{1 - 0.98z^{-1}} \quad |z| > 0.98 \;\; (\, t(n) = 0.98^n u(n)\,),$$

$$H(z) = \frac{1}{1 - 0.95z^{-1}} \quad |z| > 0.95 \;\; (\, h(n) = 0.95^n u(n)\,),$$

$u(n), v(n)$ uncorrelated zero-mean Gaussian white noise stochastic processes with power (and variance) $\sigma^2 = 1$.

```
randn('seed',13); % to use same random sequence between different simulations
L=10000000;
u=randn(L+200,1); % white Gaussian, variance of one
x=filter(1,[1 -0.98],u); % t(n) filter
x=x(101:end);      % discard first 100 samples to remove transients
                   % in simulated signals
v=randn(L+100,1); % white Gaussian, variance of one

d=filter(1,[1 -0.95],x) + v;  % h(n) filter plus additive noise
d=d(101:end);      % discard first 100 samples to remove transients
                   % in simulated signals
x=x(101:end);      % discard first 100 samples to remove transients
                   % in simulated signals
```

For this assignment, you must implement the NLMS, AP, LS and RLS algorithms.

Some statements from the previous assignment can also be repeated here:

- For all adaptive algorithms, use the initial conditions $\mathbf{w}(0) = \mathbf{0}_{N\times 1}$ and $N = 100$ coefficients for the filter.

- To observe in the error signal $e(n)$ the fast convergence of some algorithms, make sure that the input signal vector $\mathbf{x}(n) = \left[x(n) \cdots x(n-N+1)\right]^T$ is filled with non-zero values at the beginning, i.e., start the first algorithm iteration when the vector is filled. This ensures that $e(n) = d(n) - y(n) = d(n) - \mathbf{w}^H\mathbf{x}(n)$ will quickly become small.
- Even though 10,000,000 samples are generated for $x(n)$ and $d(n)$ (and they can be used for estimation of $\mathbf{R}$ and $\mathbf{p}$), you don't need use 10,000,000 samples or to perform 10,000,000 iterations for the adaptive algorithms. Just use the (much smaller) number of samples and iterations that are needed to show convergence and a part of the steady state behaviour.
- For all adaptive algorithms, when it is required to plot the performance, provide:
  - a plot of the normalized MSE learning curve $\dfrac{\xi(\mathbf{w}(n))}{\sigma_d^2} = \dfrac{\sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{w}^T(n)\mathbf{p}^* + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n)}{\sigma_d^2}$, converted to dB scale.
  - a plot of the normalized smoothed squared magnitude $\dfrac{|e(n)|^2}{\sigma_d^2}$ converted to dB scale, where the error signal is $e(n) = d(n) - \mathbf{w}^H\mathbf{x}(n)$. The smoothed curve is to be obtained by filtering $\dfrac{|e(n)|^2}{\sigma_d^2}$ with a low-pass FIR causal linear phase filter. Apply a proper shift (group delay compensation) to the smoothed curve so that it can be compared directly with the normalized MSE learning curve $\dfrac{\xi(\mathbf{w}(n))}{\sigma_d^2}$.
  - If the difference between the curves in the steady state region is not clear on a plot, produce an extra figure with a zoom around the steady state y-axis values.

Some additional comments:

- Use your code from assignment #4 to compute $\mathbf{R}$, $\mathbf{p}$ and the MMSE.
- When comparing the performance of the LS and RLS algorithms in the different plots, you may find (or should find) that there is no difference at all between them, so it may be problematic to see both curves in the plots. A trick is to change slightly the functions $\dfrac{|e(n)|^2}{\sigma_d^2}$ and $\dfrac{\xi(\mathbf{w}(n))}{\sigma_d^2}$ for one of the two algorithms, by multiplying the functions by a factor 1.1. Then when they are

plotted in dB there will be a small constant offset between the two curves. But don't use this trick when measuring the misadjustment (zoom on steady state performance), as this will corrupt your measure!

- For the AP algorithm, use the affine projection order $M = 5$ in this assignment.

Question 1

Find by trial and error the largest value of μ that leads to convergence for the NLMS and AP algorithms. Compare with the theoretical values.

Question 2

Use $\mu = 1.0$ for the NLMS. During the convergence period, you will observe a large discrepancy between the $\frac{|e(n)|^2}{\sigma_d^2}$ (or smoothed $\frac{|e(n)|^2}{\sigma_d^2}$) compared to the MSE $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$, i.e., $\frac{|e(n)|^2}{\sigma_d^2}$ is quite better than $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$. Show the resulting $\frac{|e(n)|^2}{\sigma_d^2}$, smoothed $\frac{|e(n)|^2}{\sigma_d^2}$ and $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$ plots.

Why is there a discrepancy? To answer this, remember which cost function is minimized by the NLMS algorithm.

If during the convergence period (observed in MSE curve), the adaptation of the coefficients $\mathbf{w}(n)$ is stopped and the coefficients $\mathbf{w}(n)$ have fixed values, which function is the most accurate description of the resulting the performance of the algorithm: the previous values of $\frac{|e(n)|^2}{\sigma_d^2}$ or the previous values of $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$?

Question 3

For the NLMS and AP algorithm, find the approximate value of μ which leads to the fastest convergence in the MSE plot (i.e., before the steady state performance). Compare with the LS and RLS algorithms using $\lambda = 0.999$. Show the resulting $\frac{|e(n)|^2}{\sigma_d^2}$, smoothed $\frac{|e(n)|^2}{\sigma_d^2}$ and $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$ plots.

Question 4

For both the NLMS and AP algorithms, use the value of μ found for the NLMS in Question 3, and compare the resulting misadjustment in the MSE plot. Is this as expected? Show the resulting $\frac{|e(n)|^2}{\sigma_d^2}$, smoothed $\frac{|e(n)|^2}{\sigma_d^2}$ and $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$ plots.

Note: when focusing on the misadjustment, you may need to use more iterations than in the previous question, to make sure that all algorithms have fully converged.

Question 5

Keep the same μ for the NLMS as in Questions 3 and 4, and adjust μ in the AP algorithm, as well as λ in the LS and RLS algorithm, so that all the algorithms have approximately the same misadjustment in the MSE plot (measured here by the minimum values in the different curves).

Show the resulting $\frac{|e(n)|^2}{\sigma_d^2}$, smoothed $\frac{|e(n)|^2}{\sigma_d^2}$ and $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$ plots.

For the LS and RLS algorithms, compare the value of the misadjustment with the theoretical value.

With a common misadjustment for the different algorithms, compare the convergence speed of the algorithms. Note that using a common misadjustment is often a requirement in practice when comparing convergence speed of algorithms.