Assignment #4 Steepest descent, Newton and LMS algorithm

In this assignment, you must implement the steepest descent algorithm, the Newton descent algorithm and the LMS algorithm for FIR causal adaptive filtering minimizing the mean square error.

Use the same setup as Setup A System Identification in Assignment #2:

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

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

$$e(n) = d(n) - y(n) = d(n) - \sum_{k=0}^{N-1} w_k x(n-k) = d(n) - \mathbf{w}^H \mathbf{x}(n) \text{ with } \mathbf{w} = \begin{bmatrix} w_0^* & w_1^* & \cdots & w_{N-1}^* \end{bmatrix}^T.$$

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

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

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

Use the following code to generate the signals $x(n)$ and $d(n)$:

```
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

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

- For all adaptive algorithms, use the initial conditions $\mathbf{w}(0) = \mathbf{0}_{N\times 1}$.
- 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 $\frac{|e(n)|^2}{\sigma_d^2}$ 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 $\frac{|e(n)|^2}{\sigma_d^2}$ with a low-pass FIR causal linear phase filter. Apply a proper shift to the smoothed curve so that it can be compared directly with the normalized MSE learning curve $\frac{\xi(\mathbf{w}(n))}{\sigma_d^2}$.
- 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.

1. Finding $\mathbf{R}$ and $\mathbf{p}$
   - You can re-use some material from Assignment #2:
     Using either the theoretical correlation functions $\phi_{dx}(l)$, $\phi_{xx}(l)$ or the experimentally estimated $\hat{\phi}_{dx}(l)$, $\hat{\phi}_{xx}(l)$ over 10,000,000 samples, find the $\mathbf{R}$ auto-correlation matrix and the $\mathbf{p}$ cross-correlation vector for $N = 10$ FIR filter coefficients. Also find σ_d^2 from the theoretical $\phi_{dd}(l)$ or by measuring directly $\hat{\sigma}_d^2$.
   - Find the maximum eigenvalue $\lambda_{\max}$ and the eigenvalue spread $\lambda_{\max}/\lambda_{\min}$ in the auto-correlation matrix.

2. For all algorithms, find by trial and error:
   - The range of step size μ leading to convergence (and compare it with the theoretical upper bound for convergence). No plots required here.
   - The value of the step size μ leading to the fastest initial convergence of the algorithm (and compare it with half of the theoretical upper bound for convergence). Using these step size μ values, compare the convergence of the 3 algorithms on the same figures.

3. Compare the convergence of the steepest descent algorithm and the LMS algorithm by using a common μ value: the one previously found to lead to the fastest initial convergence for the LMS algorithm.
   - Compare on the same figures the steady state MSE produced by the two algorithms with the MMSE obtained from $MMSE = \xi(\mathbf{w}_{opt}) = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p}$.

Measure the misadjustment of the LMS algorithm. Compare with the theoretical value. In case of discrepancy, provide a tentative explanation why it is different.

- Make a new simulation with the same common value of μ, but where the value of μ is reduced by a factor 10 once the algorithms reach (approximately) a steady state. Verify that for the LMS this reduces the misadjustment by a factor of 10 (perhaps approximately).