Second SPPA assignment

To be delivered, in a PDF document, on January 30

Problem 1

The main goal of this first problem is to identify two "unknown" systems, using a transversal filter and the NLMS algorithm. The identification of the first system, which is a time-invariant IIR system with one pole and one zero, has as main objective the verification that the student's implementation of the NLMS algorithm works and to make it easier to observe its main steady-state weight adaptation characteristics. The identification of the second system has as main objective the study of the transients in the adaptation of the weights.

With respect to the first system to be identified, implement the NLMS algorithm applied to a transversal filter of order `n` (`n` delays). Test it using the input signal `x` and the desired signal `d`, generated by the following code

```
T=10000;
sigma=0.01;
x=randn(1,T);
d=filter([1,2],[1,0.5],x)+sigma*randn(1,T);
```

In the middle of the adaptation (time `T/2`), reduce μ to $1/10$ of its initial value. Draw graphs with the evolution of the weights of the filter over time and of the evolution of the squared error

$$\xi(t) = (1-\lambda)\sum_{k=0}^{\infty} \lambda^k e^2(t-k)$$

over time. Use $\lambda = 0.95$ and try several values of `n` and of μ. Comment your results and draw conclusions.

Continuing with the first system to be identified, use now as input and desired signals the signals generated bu the following code

```
T=10000;
sigma=0.01;
phase=0.01+0.01*floor(4*(0:T-1)/T);
x0=randn(1,T);
x1=sin(2*pi*phase.*(0:T-1));
x2=cos(8*pi*(0:T-1)/T)>=0;
x=x0.*x2+x1.*(1-x2);
d=filter([1,2],[1,0.5],x)+sigma*randn(1,T);
```

Again, in the middle of the adaptation (time `T/2`), reduce μ to $1/10$ of its initial value. Draw graphs with the evolution of the weights of the filter over time and of the evolution of the squared error.

Going now to the second system to be identified, use as input and desired signals the signals

```
T=10000;
sigma=0.01;
x=randn(1,T);
d=zeros(1,T);
U=zeros(4,T);
xx=[0,0,0,0];
for t=1:T
  xx=[x(t),xx(1:3)];
  U(:,t)=[0.2+0.3*t/T;...
          0.7*cos(4*pi*t/T);...
```

```
          0.3*sign(sin(10*pi*t/T))-0.1;...
          -0.2];
    d(t)=xx*U(:,t);
  end
  d=d+sigma*randn(1,T);
```

As usual, in the middle of the adaptation (time `T/2`), reduce μ to 1/10 of its initial value. Draw graphs with the evolution of the weights of the filter over time and of the evolution of the squared error.

Problem 2

Let $v(t)$ be a Gaussian white noise stochastic process with zero mean and unit variance. Let

$$x(t) = v(t) + 3v(t-1) + 3v(t-2)$$

be the input signal of a order n transversal filter, let

$$d(t) = v(t) - 2v(t-1) + v(t-2)$$

be its desired signal, and let

$$y(t) = \sum_{k=0}^{n} w_k x(t-k)$$

be its output signal. The goal of this problem is to compute the optimal weights, and respective (least) squares error, when the adaptation is inactive (i.e., time-invariant weights), for $n = 0, 1, \ldots, 10$. In order to do that:

1. Compute the auto-correlation function of $x(t)$ and cross-correlation function of $x(t)$ and $d(t)$. In other words, for $k = 0, \pm 1, \pm 2, \cdots$, compute $r_{xx}(k) = \langle x(t+k), x(t) \rangle$ and $r_{xd}(k) = \langle x(t+k), d(t) \rangle$. Use the fact that $v(t)$ is white noise. Present your results in a table. (Suggestion: confirm your theoretical results by using, say, 1000000 samples of one realization of $x(t)$ and $d(t)$ to estimate $r_{xx}(k)$ e $r_{xd}(k)$.)

2. Solve the system of normal equations for $n = 0, 1, \ldots, 10$, and compute the respective minimum squared error. Make a graph of this error as a function of n.

Repeat this problem for $x(t) = 3v(t) + 3v(t-1) + v(t-2)$.

Why does the behavior of the minimum squared error differ when n increases in the two cases?

[Hard] When n increases, and for each of the two cases, to which filter is the transversal filter converging?

Problem 3

For the signal $d(t)$ and for each of the two signals $x(t)$ of problem 2, use the NLMS algorithm to adapt the transversal filter coefficients as time goes by. Use 10000 samples, $n = 3$, and $\mu = 0.01$. Make a graph of the time evolution of the coefficients and of the (squared) error as a function of time. Comment our results and draw conclusions. Also, compare the final values of the transversal filter coefficients with the optimal coefficients obtained in problem 2.

Problem 4

The input signal of an adaptive transversal filter is given by

$$x(t) = \cos(\omega t) + v(t), \qquad t = 0, 1, \ldots, 10000,$$

where $v(t)$ is white Gaussian noise with zero mean and a standard deviation of σ. The desired signal of this filter is given either by

$$d(t) = x(t+1)$$

or by

$$d(t) = A(t)\,\sin(\omega t), \qquad A(t) = \begin{cases} +2 & \text{se } t < 5000, \\ -2 & \text{se } t \geqslant 5000. \end{cases}$$

With respect to the first desired signal, the goal is to try to predict the next sample of the input signal. With respect to the second desired signal, the goal is to try to remove the noise and to try to change the amplitude and phase of the sinusoidal signal. Use $\omega = 0.2\pi$ and $\sigma = 0.01$.

Using the NLMS algorithm to adjust the coefficients of adaptive transversal filters with $n = 1$, $n = 2$, and $n = 3$. Make graphs of the time evolution of the weights and squared error of the adaptive filters. Make your experiments for $\mu = 0.1$ and for $\mu = 0.8$. Comment our results and draw conclusions.

Finally, determine the optimal filter coefficients for the two desired signals when $n = 1$ and $n = 2$. It is known that

$$\big\langle \cos(\omega t + \theta_1), \cos(\omega t + \theta_2) \big\rangle = \frac{\cos(\theta_1 - \theta_2)}{2}$$

and that the white noise is decorrelated with all signals of the form $\cos(\omega t + \theta)$.

Problem 5

The blind equalization of a slowly time-varying transmission channel is one of the most interesting applications of an adaptive filter. One of the problems that has to be solved in this type of applications is the choice of the desired signal of the adaptive filter. Indeed, as in this type of applications one is trying to recover the digital information sent by the transmitter, which is spread over time by the transmission channel, it does not make any practical sense to use the original transmitted signal as the desired signal, because that is precisely the signal one is trying to recover.

At the receiver, one has to filter the received signal and then one has to make a decision about the signal that was transmitted. This decision is done based on the known characteristics of the signal that was transmitted (the so-called signal constellation). So, one possible way to they to solve the blind equalization problem is to make the desired signal be the decision that was made. If this is done, the adaptive filter will try to make the decisions that were make "more perfect" as time goes by. For example, for a bipolar transmitted signal (each sent symbol is either $+1$ or -1), when the filtered received signal is $+0.1$ the decision is $+1$, and the coefficients of the adaptive filter are changed so that in the future a received signal similar to the current one would produce a filtered signal closer to $+1$.

In this problem we will be transmitting 8-PSK symbols — signal $u(t)$ — to the transmission channel, which can be generated by the following code

```
T=50000;
s=exp(2i*pi*(1:2:15)/16);       % 8-PSK symbols
u=s(ceil(length(s)*rand(1,T))); % transmitted data symbols
```

The transmission channel will distort these symbols in the following way

```
x=filter([1,-1,2,-2,7,2,2,2,-1],1,u); % received data,
x=x+0.01*randn(1,T)+0.01i*randn(1,T); %   with some noise
```

and so the received signal, which will be the input signal of the adaptive transversal filter, will have a significant inter-symbol interference. Out goal here is to try to recover the original signal u. To that end, use the NLMS algorithm, **modified to work with complex signals and coefficients**, and the desired signal will be the 8-PSK constellation symbol that is closest to the filtered signal $y(t)$ (this is called "decision-directed" adaptation).

The coefficients of the transversal adaptive filter should all be initially set to zero, with the exception of the middle one, which should be initialized to 1. Make experiments with filters with orders 10, 20 e 30. Use $\mu = 0.1$.

The following aspects of the problem and interesting and should be explored experimentally:

1. Does the adaptive filter converge always to the same "optimal" filter?
2. When it converges, on average how much time does it take to converge?
3. When it converges, what is the relationship between $d(t)$ and $u(t)$?
4. What μ and be used?

Annex

Possible implementation of the NLMS algorithm, for real signals and weights, for a transversal filter:

```
% NLMS algorithm for a transversal filter
%   inputs:
%     x in the input signal
%     d is the desired signal
%     n is the transversal filter order
%     mu is the convergence factor
%   outputs:
%     W are the weights
%     y is the output
%     e is the error
T=length(x);
W=zeros(n+1,T+1);
xx=zeros(n+1,1);
y=zeros(size(x));
%d=zeros(size(x)); % uncomment if d(t) is conputed inside the for loop
for t=1:T
  xx=[x(t);xx(1:n)];
  y(t)=W(:,t)'*xx;
  e(t)=d(t)-y(t);
  W(:,t+1)=W(:,t)+2*mu/(1e-20+xx'*xx)*e(t)*xx;
end
```