Indian Institute of Science
Department of Electrical Communication Engineering

# E9 211: Adaptive Signal Processing

October 2020 - January 2021

**Homework 3 (deadline 15 Jan. 2021)**

This homework consists of two parts on implementing and studying the adaptive filters: (a) source separation using an antenna array using RLS and (b) time-varying channel estimation using a Kalman filter.

Make a short report containing the required Matlab/Python files, plots, explanations, and answers, and turn it in by the deadline using Microsoft Teams under your name.

## Part A: Antenna beamforming

As in HW1, using `function X = gen_data(M,N,Delta,theta,SNR)` generate the data matrix $\mathbf{X} = \mathbf{A}_\theta \mathbf{S} + \mathbf{N}$. Recall that

$$\mathbf{A}_\theta = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \cdots, \mathbf{a}(\theta_d)] : M \times d.$$

The source symbols $\mathbf{S} = [\mathbf{s}_1 \cdots \mathbf{s}_d]^{\mathrm{T}} : d \times N$ are chosen uniformly at random from a QPSK alphabet $\{(\pm 1 \pm j)/\sqrt{2}\}$. The noise matrix $\mathbf{N} : M \times N$ is random zero-mean complex Gaussian matrix.

Consider a system with two sources and take $\boldsymbol{\theta} = [0^\circ, 5^\circ]^{\mathrm{T}}$, $M = 5$, $\Delta = 0.5$, $N = 2000$, SNR = 20 dB. Make Matlab subroutines to compute the beamformer for the first source, i.e., $\mathbf{y} = \mathbf{w}^{\mathrm{H}} \mathbf{X} : 1 \times N$ using recursive least squares (RLS) as

$$\texttt{[y,w]=rls(X,s\_ref,}\lambda\texttt{)}$$

Here, $\lambda$ is the factor in the exponential window RLS. Use $\theta_{\texttt{init}} = \mathbf{0}$, $\texttt{P}_{\texttt{init}} = 100\mathbf{I}$, and for $\texttt{s}_{\texttt{ref}}$ use the true source symbols of the source at $0^\circ$. Plot the estimated symbols in the complex plane such that you observe four clusters (use `plot(s_est,'x')`) and compare it with the symbol estimates from the LMMSE receiver from HW1 (for $\lambda = 1$) and LMS from HW2. Also, plot the learning curves to show the convergence of RLS for different values of $\lambda$ and compare it to LMS from HW2. Compare it the minimum mean-squared error obtained with the LMMSE receiver when the algorithm converges. What do you observe?

## Part B: Time-varying channel estimation

Consider a slow-fading dispersive channel. The time-varying channel is described by the following model

$$\mathbf{h}[n] = \mathbf{A}\mathbf{h}[n-1] + \mathbf{w}[n]$$

where $\mathbf{h}[n] = [h_n[0], h_n[1], \ldots, h_n[p-1]]^T$, $\mathbf{A}$ is a known $p \times p$ matrix, and $\mathbf{w}[n]$ is the Gaussian noise vector having zero mean and covariance matrix $\mathbf{Q}_w$. The received signal is given by

$$y[n] = \sum_{k=0}^{p-1} h_n[k]s[n-k] + v[n]$$

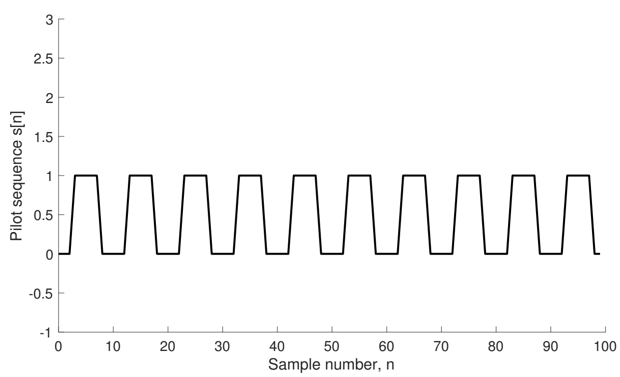
where $s[n]$ is the pilot symbol and $v[n]$ is the Gaussian observation noise having zero mean and variance $Q_v = \sigma^2$. We assume $p = 2$ with

$$\mathbf{A} = \begin{bmatrix} 0.99 & 0 \\ 0 & 0.999 \end{bmatrix}; \qquad \mathbf{Q}_w = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix},$$

and $\sigma^2 = 0.1$.

Make subroutines to

1. Generate the pilot sequence as below




   To generate 100 samples of such a sequence, you may use the following Matlab commands:

```
>> N=10; M=10; N1=M*N; n=0:N1-1;
>> s=0.5-0.5*sign(cos(2*pi*n/N)).
```

   Generate the channel with initial taps $\mathbf{h}[0] = [1, 0.9]^T$.

2. Estimate the channel using Kalman filter

```
h = kf(A,s,Q_w,Q_v)
```

   Use $\hat{\mathbf{h}}[-1|-1] = \mathbf{0}$ and $\mathbf{P}[-1|-1] = 100\mathbf{I}$. Plot the estimated channel taps and compare it to the true channel taps. Plot the Kalman gain, predicted, and estimated error covariances for each tap. What do you conclude?