**Assignment 2: Due Tuesday May 14 in class.**

1. Simulate a 2-state Markov chain $X$ with transition probabilities

$$A = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}$$

and state levels $q = [-1, 1]$. The Markov chain is observed via the observation process $y \in \{-1, 1\}$ with $P(y = 1|X = 1) = P(y = 0|X = 0) = p$, where $p$ is a parameter that we will change below.

   (a) Simulate a 1000 point sample path of $y$ when $p = 0.8$. Derive and simulate a HMM filter for estimating $X$. Compute the state estimation error probability of the HMM filter by rounding off the conditional mean estimate to the nearest state value and comparing with the actual state of the Markov chain. (You need to scale the estimate to avoid numerical underflow).

   (b) Suppose that, given observations until time $k$, i.e., $y_1, \ldots, y_k$, you wish to compute the optimal estimate of the state at time $k - \Delta$. Derive the optimal estimation algorithm for any fixed value of $\Delta$. Your algorithm should generate estimates for time $k = \Delta, \Delta + 1, \ldots, 1000$. Simulate in Matlab the state estimation error probability of the algorithm for $\Delta = 1, 2, \ldots, 5$.

   (c) Suppose now that you wish to predict the Markov chain at time $k+\Delta$, given observations $y_1, \ldots, y_k$. Derive the optimal estimation algorithm for any fixed value of $\Delta$. Your algorithm should generate estimates for time $k = 1, 2, \ldots, 1000 - \Delta$. Simulate in Matlab the state estimation error probability of the algorithm for $\Delta = 1, 2, \ldots, 5$.

   (d) Finally, simulate the Viterbi algorithm for this problem and compute the error probabilities of the Viterbi algorithm.

   (e) Redo the above steps for different values of $p$ to compare the 4 algorithms. Illustrate your results in terms of state estimation error probabilities.

2. (a) Simulate the Expectation Maximization algorithm to compute the maximum likelihood estimate of the transition probabilities and observation probabilities $p$ in the above example. Plot at each iteration the model likelihood generated by the EM algorithm.

(b) Compare your result above by optimizing the likelihood with a general purpose Matlab optimization algorithm. That is, use the HMM filter as discussed in class to evaluate the model likelihood – and then use a Matlab optimization algorithm to optimize the likelihood.

3. Suppose $\theta$ is a random variable with distribution that is Gaussian with unit variance and zero mean. Now assume that you have measurements

$$y_k = \theta + w_k, \quad k = 1, \ldots, 1000$$

where $w_k$ is Gaussian with zero mean and variance 0.25. Note that in the above model $\theta$ is only generated once (it is a random variable) and then the sequence of observations is generated for $k = 1, 2, \ldots,$.

- Derive an optimal (conditional mean) recursive estimator for $\theta$ given the sequence of observations $y_1, \ldots, y_k$, for $k = 1, 2, \ldots$. Simulate your estimator in Matlab and plot the mean square error of the state estimate versus time (averaged over several simulation runs).
- Suppose now that $w_k$ is uniformly distributed $U[-1, 1]$. Derive a simulation based approach to estimate the conditional mean of $\theta$ given the sequence of observations $y_1, \ldots, y_k$, for $k = 1, 2, \ldots$. Simulate your estimator in Matlab and plot the mean square error of the state estimate versus time (averaged over several simulation runs).
- For both of the above cases, compare your answer with a recursive least squares estimator for $\theta$ in terms of mean square error of the state estimate.