To facilitate grading, please deliver each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets. You must submit the answer sheets at the class.

1. **Kalman Filter** (15, 12 Pts). Given a discrete-time system

\[ x_k = \Phi_{k-1} x_{k-1} + \Gamma_{k-1} u_{k-1} + \Lambda_{k-1} w_{k-1} \]  

(4.1)

with \( k = 1 \) to \( k_{max} \). It is assumed that \( \Phi_{k-1}, \Gamma_{k-1}, \Lambda_{k-1}, \) and \( u_{k-1} \) are known without error. However, the initial condition \( x_0 \) is a Gaussian random variable prescribed by its mean value and covariance matrix

\[ E(x_0) = m_0 \]
\[ E\left[(x_0 - m_0)(x_0 - m_0)^T\right] = P_0 \]  

(4.2)

The disturbance input is a white, zero-mean Gaussian sequence

\[ E(w_k) = 0, \quad E(w_k w_k^T) = Q_k, \quad E(w_k w_j^T) = 0, \quad (j \neq k) \]  

(4.3)

The observation vector is

\[ z_k = H_k x_k + n_k \]  

(4.4)

where \( H \) is known and the measurement error is a white, zero-mean Gaussian random sequence that is uncorrelated with the disturbance input

\[ E(n_k) = 0 \]
\[ E(n_k n_k^T) = R_k \]
\[ E(n_k n_j^T) = 0, \quad (j \neq k) \]
\[ E(n_k w_j^T) = 0 \]  

(all \( j \) and \( k \))

(4.5)

(a) The objective is to compute an estimate of the state, denoted by \( \hat{x} \), from the measurement \( z \). So, the first step is to find the system model equations which can be used to propagate the statistics of the state—without considering measurements. The results are used to compute the filter gains that weights prior knowledge of measurement error covariance with state estimate covariance.

Find the equations to propagate and update state and covariance estimates. You can follow Appendix [A] to do it.
(b) Consider the case when the disturbances that force the system are sensed directly in the observations

\[ x_k = \Phi_{k-1} x_{k-1} + \Gamma_{k-1} u_{k-1} + w_{k-1} \]
\[ z_k = H x_k + H u_k + n_k \]

(4.6)

The disturbance input and the measurement noise are modeled as a time-skewed, white joint stochastic process such that

\[ E \left\{ \begin{bmatrix} w_{k-1} \\ n_k \end{bmatrix} \begin{bmatrix} w_{k-1}^T \\ n_k^T \end{bmatrix} \right\} = \begin{bmatrix} Q_{k-1} & M_k \\ M_k^T & R_K \end{bmatrix}, \quad E \begin{bmatrix} w_{k-1} \\ n_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \]

(4.7)

Derive the propagation and update equations for this model.

2. Dynamic Programming, EM (31 Pts). Consider a discrete state dynamical system

\[ x_t = f(x_{t-1}, u_t) \]
\[ y_t = g(x_t, u_t) \]

(4.8)

where \( u_t \in \mathbb{R}^m \) is the input vector, \( y_t \in \mathbb{R}^r \) is the output vector, and \( x_t \in V = \{1, 2, ..., n\} \) is a discrete state. The system can be framed by the probabilistic graphical model depicted in figure 4.1. The model is composed by a set of states \( N_j, j = 1, ..., n \) and a set of outputs \( O_j, j = 1, ..., n \). Each one of the states and outputs is uniquely associated to one of the states in \( V \), and they are defined by mappings \( N_j(u_t; \theta_j) \) and \( O_j(u_t; \vartheta_j) \), where \( \theta_j \) and \( \vartheta_j \) are vectors of adjustable parameters. The learning algorithm can be derived from the maximum likelihood principle. The training data are a set of \( p \) pairs of input/output sequences of length \( T \) : \( D = \left\{ (u_1^{T_p}(p); y_1^{T_p}(p)) : p = 1, ..., P \right\} \).

Derive the learning algorithm from the maximum likelihood principle. The state distribution is multinomial. Comply with the variables defined in Appendix B.


(a) Consider a class of latent variable models which capture observed data as being generated by the linear mixing of hidden sources. Suppose there are \( M \) sources identified with
their probability density functions as \( p_m(s^m) \), observations, \( x_t \in \mathbb{R}^n \), are produced by instantaneous linear mixing of the sources by \( A \)

\[
x_t = A s_t \tag{4.9}
\]

The aim is to recover the latent sources \( \hat{s}_t \) by finding \( W \), the (pseudo) inverse of \( A \),

\[
\hat{s}_t = \hat{W} x_t = W A s_t.
\]

Find the log-likelihood of observations. Assume that the sources are independent.

(b) In the linear generative model of \( x_t = A s_t \), the variances of components cannot be recovered. The idea is to explicitly model the variances of the components, and then integrate them out in a Bayesian framework; So, consider the variances of the components as separate parameters. Let \( \sigma_j^2 \) to be the variance of the \( j \)th independent component. Denote the vector collecting the \( \sigma_j \) as \( \sigma \), and denote \( W = (w_1, ..., w_n)^T = A^{-1} \), we have

\[
p(x|A, \sigma) = |\det W| \prod_j \frac{1}{\sigma_j} p_j\left(\frac{w_j^T x}{\sigma_j}\right) \tag{4.10}
\]

Consider a special form of the \( p_i \) as called generalized Laplacian distribution

\[
p_j(s|\alpha_j) = \frac{1}{z(\alpha_j)} \exp(-|s|^{\alpha_j} c(\alpha_j)) \tag{4.11}
\]

Use a noninformative prior for the \( \sigma_j \)

\[
p(\sigma) = \prod_j \frac{1}{\sigma_j} \tag{4.12}
\]

Consider an i.i.d. sample of \( x \) as a whole in matrix form as \( X = (x(1), ..., x(T)) \).

i. Write \( p(x, \sigma|A, \alpha) \) and then integrate out \( \sigma \) to achieve the log-likelihood function. Interpret this likelihood with choosing various values for \( \alpha \).

ii. What differences are there between Independent Component Analysis(ICA), Eq. \ref{eq:ica}, Factor analysis(FA), and Principal Component analysis(PCA). Your answers must include why you choose Gaussian distributions in FA framework(does kind of distributions defined over latent and observed variables influence the optimization procedure, i.e., the form of likelihood), what subspace or curve these models tend to learn, what directions they move to learn. An example available at \url{https://martinos.org/mne/stable/manual/preprocessing/ica.html} could be useful; Run it and conclude.

(c) We want to design a non-stationary framework for the generative model of \( x_t = A s_t \), so that the mixing matrix \( A_t \) is allowed to vary with time. We also assume that the observation \( x_t \) is contaminated by normally distributed noise \( \omega_t \sim N(0, R) \); Thus

\[
x_t = A_t s_t + \omega_t \tag{4.13}
\]
The assumptions of stationary and independence of sources permit to write likelihood as
\[ p(x_t|A_t) = \int p(x_t|A_t, s) \prod_m p_m(s^m) ds \]  
(4.14)

Let \( a_t = \text{vec}(A_t) \) be the \( N \times M \)-dimensional vector obtained by stacking the columns of \( A_t \), then \( a_t \) evolves according to the state equation
\[ a_{t+1} = Fa_t + v_t \]  
(4.15)

where \( v_t \) is zero-mean Gaussian noise with covariance \( Q \), and \( F \) is the state transition matrix. The problem is now to track \( A_t \) and to learn \( \theta \) as new observation \( x_t \) become available. Let \( X_t \) denotes the collection of observations \( x_1, \ldots, x_t \), then the goal is to deduce the probability density function of the state \( p(a_t|x_t) \).

i. Draw a probabilistic graphical model representing the problem.

ii. Outline a propagate-update procedure to compute \( p(a_t|x_t) \).

(d) **Bonus** Conditional dependency of latent variables. Modify the model \( x_t = As_t \) to include the conditional dependence of \( s^m_t \) on \( s^m_{t-1} \).

i. Trace the probabilistic graphical model of the modified model.

To capture the probability model, consider the energy of a sequence of \( T \) states and observations \( \{(s^t, x^t)\}_{t=1}^T \) as
\[
H (\{S_t, X_t\}) = \frac{1}{2} \sum_{t=1}^{T} \left( X_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right)' C^{-1} \left( X_t - \sum_{m=1}^{M} W^{(m)} S_t^{(m)} \right) \\
- \sum_{m=1}^{M} S_1^{(m)' \log \pi^{(m)}} - \sum_{t=2}^{T} \sum_{m=1}^{M} S_t^{(m)' \log P^{(m)}} S_{t-1}^{(m)}
\]  
(4.16)

The probability model is defined from the energy by the Boltzmann distribution
\[
p (\{S, X\}) = \frac{1}{Z} \exp \{-H (\{S, X\})\}
\]  
(4.17)

Overall, you can also consider this configuration; The probability for a \( D \times 1 \) observation vector \( x_t \) is
\[
p (x_t|S_t) = |C|^{-1/2} (2\pi)^{-D/2} \exp \left\{ -\frac{1}{2} (x_t - \mu_t)' C^{-1} (x_t - \mu_t) \right\}
\]  
(4.18)

where \( \mu_t = \sum_{m=1}^{M} W^{(m)} S_t^m \). The state representation is \( S_t = S_t^{(1)}, \ldots, S_t^{(M)} \) and state transition as follows
\[
p(S_t|S_{t-1}) = \prod_m p(S_t^{(m)}|S_{t-1}^{(m)})
\]  
(4.19)

ii. Establish exactly the steps of EM algorithm.
Appendix A. We begin with the linear observation equation

$$z = Hx + n$$
$$= y + n$$

(4.20)

Our objective is to compute an estimate of the state, $\hat{x}$. An estimate of the output, given $\hat{x}$,

$$\hat{y} = H\hat{x}$$

(4.21)

Define the state and measurement residual errors respectively as

$$\epsilon_x \triangleq x - \hat{x}$$
$$\epsilon_z \triangleq z - \hat{y}$$

(4.22)

A quadratic cost function of the state residual is formed as

$$J(z) = \frac{1}{2} \epsilon_z^T \epsilon_z$$

(4.23)

Take the derivative of $J$ and obtain $\hat{x}$.

Regarding the measurement error, it may vary from one point to the next; Hence, it is desirable to weight the good data more heavily than the poor data in the estimator. We have

$$\epsilon_{zk} = (z_k - \hat{y}_k) = H_k \epsilon_{x_k} + n_k$$

(4.24)

If $\epsilon_{x_k}$ and $n_k$ were known, a normalized scalar residual $\epsilon'_{zk}$ could be formed,

$$\epsilon'_{zk} = \frac{\epsilon_{zk}}{H_k \epsilon_{x_k} + n_k} = \frac{\epsilon_{zk}}{v_k}$$

$$\epsilon'_{z} = N^{-1} \epsilon_{z} = N^{-1}(z - H\hat{x})$$

(4.25)

Found $J'(z') = \frac{1}{2} \epsilon_z'^T \epsilon_z'$ and then take its derivative. For integrity let $S = N^T N$.

At this point, you obtained $\hat{x} = H_{WL} Z$, where $H_{WL}$ is the weighted left pseudoinverse of $H$. Trouble is, $\epsilon_x$ and $n$ normally are not known deterministically. When they are random variables, a cost function based on the expected value of the error can be formed

$$E(\epsilon_x) = E(x - \hat{x}) = E(x) - E(\hat{x}) = m_x$$

(4.26)

If there is known bias in the measurement,

$$E(n) = m_n$$
$$E(x) = m_x + H_{WL}(z - m_n)$$

(4.27)

Here, two definitions of $S$ are available: the expected measurement error covariance and the expected measurement residual covariance as follow

$$S_1 = E \left[ (z - y) (z - y)^T \right]$$
$$S_2 = E \left[ (z - H\hat{x}) (z - H\hat{x})^T \right]$$

(4.28)
For integrity, take $P = E[(x - \hat{x})(x - \hat{x})^T]$, $M = E[(x - \hat{x})n^T]$. We have $\hat{x} = x + (H^T S^{-1} H)^{-1} H^T S^{-1} n$. Obtain $P$ and $M$ with $S = S_1 = R$.

The previous estimators are batch processing algorithm. The prior estimate can be used as the starting point for a sequential estimation algorithm. Given $K_1$ measurements, $Z_1$, the corresponding output error matrices, $H_1$ and $S = R_1$, and the resulting estimate $\hat{x}_1$

\begin{align*}
  z_1 &= H_1 x + n_1 \\
  \hat{x}_1 &= (H_1^T R_1^{-1} H_1)^{-1} H_1^T R_1^{-1} z_1
\end{align*}

(4.29)

The new measurement $z_2$ with dimension $k_2$, is

\begin{equation}
  z_2 = H_2 x + n_2
\end{equation}

(4.30)

The cost function for all $(k_1 + k_2)$ measurements $z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ can be partitioned as

\begin{equation}
  J(z_1, z_2) = \begin{bmatrix} (z_1 - H_1 \hat{x}_2)^T (z_2 - H_2 \hat{x}_2)^T \end{bmatrix} \begin{bmatrix} R_1^{-1} & 0 \\ 0 & R_2^{-1} \end{bmatrix} \begin{bmatrix} (z_1 - H_1 \hat{x}_2) \\ (z_2 - H_2 \hat{x}_2) \end{bmatrix}
\end{equation}

(4.31)

Taking the derivative of $J(z_1, z_2)$ and setting it equal to zero provides the least-squares estimate $\hat{x}_2$

\begin{equation}
  \hat{x}_2 = (H_1^T R_1^{-1} H_1 + H_2^T R_2^{-1} H_2)^{-1} (H_1^T R_1^{-1} z_1 + H_2^T R_2^{-1} z_2)
\end{equation}

(4.32)

Using the matrix inversion lemma, we have

\begin{equation}
  \hat{x}_2 = \hat{x}_1 + P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1} (z_2 - H_2 \hat{x}_1) \\
  \quad = \hat{x}_1 + K_2(z_2 - H_2 \hat{x}_1)
\end{equation}

(4.33)

where $K_2 = P_1 H_2^T (H_2 P_1 H_2^T + R_2)^{-1}$ is the recursive weighted least-squares estimator gain matrix.

Now, you are given with $\hat{x}_k, K_k, P_k$ by redefining $k$ as a time index. To cope with the Eq. \ref{eq:4.34}, you could compute $E(x_k) = m_k$ and $E[(x_k - m_k)(x_k - m_k)^T] = P_k$ as the state and covariance estimate propagations respectively.

A recursive optimal filter propagates the statistics of one sampling instant to the next, taking into account dynamics and inputs, and it incorporates measurements and measurement error statistics in the estimate. So, we must distinguish between estimates made before and after the updates occur; $\hat{x}_k(-)$ is the state estimate that results from the propagation equation alone (i.e., before the measurements are considered), and $\hat{x}_k(+) \text{ is the corrected state estimate that accounts for the measurements.}$ $P_k(-)$ and $P_k(\text{+})$ are defined similarly. Now, you are able to write the Kalman equations.

**Appendix B.** To derive the learning algorithm for problem 2, we could start by likelihood function

\begin{equation}
  L_c(\Theta) = p(y, x|u) = \prod_{p=1}^{P} p(y_1^{T_p}(p), x_1^{T_p}(p)|u_1^{T_p}(p); \Theta)
\end{equation}

(4.34)
Break the \( p(y^T_1, x^T_1 | u^T_1) \) as \( p(y_T, x_T | x_{T-1}, u_T)p(y^T_{T-1}, x^T_{T-1} | u^T_1) \) and substitute it in the likelihood. Let \( z_{i,t} = 1 \) if \( x_t = i \) and zero otherwise. Consequently, \( E[z_{i,t} | u^T_1] = \xi_{i,t} \) and the log likelihood would become

\[
l_c = \sum_p \sum_t \sum_i z_{i,t} \log p(y_t, x_t = i | u_t) + \sum_j z_{i,t} \log p(x_t = i | x_{t-1} = j, u_t) \tag{4.35}
\]

In the E step, you must resolve \( E_{X} [z_{i,t} | y^T_1, u^T_1] \) and \( E_{X} [z_{i,t}, z_{j,t-1} | u^T_1, y^T_1] \). For unity consider the following

\[
E[z_{i,t}] = p(x_t = i | y^T_1, u^T_1) \triangleq g_{i,t},
\]

\[
E[z_{i,t}, z_{j,t-1}] = p(x_t = i, x_{t-1} = j | y^T_1, u^T_1) \triangleq h_{ij,t},
\]

\[
\alpha_{i,t} \triangleq p(x_t = i, y^T_{t+1} | u^T_1),
\]

\[
\beta_{i,t} \triangleq p(y^T_{t+1} | x_t = i, u^T_t)
\]

\[
\xi_{i,t} \triangleq \log p(x_t = i | u^T_1)
\]

\[
\varphi_{i,t} \triangleq p(y^T_{t} | x_t = i, u^T_{t-1}) \triangleq \varphi_{i,t}
\]

\[
L \triangleq E [y_t | x_t = i, u_t] = \eta_{i,t}
\]

After computing the expectations \( (h_{ij,t}, g_{i,t}) \) according to \( \alpha \) and \( \beta \), the maximization step is to be calculated. There, however, is not any direct analytic maximization, a simple way of producing an increase in likelihood is to use gradient ascent. You can just write maximization step generally.