EECS225A Course Notes

Disclaimer: These notes reflect EECS225A when I took the course. They may not accurately reflect current course content, so use at your own risk.

Contents

1 Hilbert Space Theory

Complex random variables form a Hilbert space with inner product $\langle X, Y \rangle = \mathbb{E}[XY^*]$. If we have a random complex vector, then we can use Hilbert Theory in a more efficient manner by looking at the matrix of inner products. For simplicity, we will call this the "inner product" of two complex vectors.

Definition 1 *Let the inner product between two random, complex vectors $\boldsymbol{Z_1}, \boldsymbol{Z_2}$ be defined as*

$$\langle \boldsymbol{Z_1}, \boldsymbol{Z_2} \rangle = \mathbb{E}[\boldsymbol{Z_1}\boldsymbol{Z_2}^*]$$

The ij-th entry of the matrix is simply the scalar inner product $\mathbb{E}\left[X_i Y_j^*\right]$ where X_i and Y_j are the ith and jth entries of $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. This means the matrix is equivalent to the cross correlation R_{XY} between the two vectors. We can also specify the auto-correlation $R_X = \langle \boldsymbol{X}, \boldsymbol{X} \rangle$ and auto-covariance $\Sigma_X = \langle \boldsymbol{X} - \mathbb{E}[\boldsymbol{X}], \boldsymbol{X} - \mathbb{E}[\boldsymbol{X}] \rangle$. One reason why we can think of this matrix as the inner product is because it also satisfies the properties of inner products. In particular, it is

1. Linear: $\langle \alpha_1 \boldsymbol{V_1} + \alpha_2 \boldsymbol{V_2}, \boldsymbol{u} \rangle = \alpha_1 \langle \boldsymbol{V_1}, \boldsymbol{u} \rangle + \alpha_2 \langle \boldsymbol{V_2}, \boldsymbol{u} \rangle$.
2. Reflexive: $\langle \boldsymbol{U}, \boldsymbol{V} \rangle = \langle \boldsymbol{V}, \boldsymbol{U} \rangle^*$.
3. Non-degeneracy: $\langle \boldsymbol{V}, \boldsymbol{V} \rangle = \boldsymbol{0} \Leftrightarrow \boldsymbol{V} = \boldsymbol{0}$.

Since we are thinking of the matrix as an inner product, we can also think of the norm as a matrix.

Definition 2 *The norm of a complex random vector is given by $\|\boldsymbol{Z}\|^2 = \langle \boldsymbol{Z}, \boldsymbol{Z} \rangle$.*

When thinking of inner products as matrices instead of scalars, we must rewrite the Hilbert Projection Theorem to use matrices instead.

Theorem 1 (Hilbert Projection Theorem) *The minimization problem $\min_{\hat{\boldsymbol{X}}(\boldsymbol{Y})} \|\hat{\boldsymbol{X}}(\boldsymbol{Y}) - \boldsymbol{X}\|^2$ has a unique solution which is a linear function of $\boldsymbol{Y}$. The error is orthogonal to the linear subspace of $\boldsymbol{Y}$ (i.e $\langle \boldsymbol{X} - \hat{\boldsymbol{X}}, \boldsymbol{Y} \rangle = \boldsymbol{0}$)*

When we do a minimization over a matrix, we are minimizing it in a PSD sense, so for any other linear function $\boldsymbol{X}'$,

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|^2 \preceq \|\boldsymbol{X} - \boldsymbol{X}'\|^2.$$

1.1 Innovations

Suppose we have jointly distributed random variables $Y_0, Y_1, \cdots, Y_n$. Ideally, we would be able to "de-correlate" them so each new vector E_0 captures the new information which is orthogonal to previous random vectors in the sequence. Since vectors of a Hilbert Space operate like vectors in $\mathbb{R}^n$, we can simply do Gram-Schmidt on the $\{Y_i\}_{i=0}^n$.

Definition 3 *Given jointly distributed random vectors $\{Y_i\}_{i=0}^n$ with $\mathcal{L}_i = span\{Y_j\}_{j=0}^i$, the ith innovation E_i is given by*

$$E_i = Y_i - proj(Y_i|\mathcal{L}_{i-1}) = Y_i - \sum_{j=0}^{i-1} \frac{\langle Y_i, E_j \rangle}{\|E_j\|^2} E_j$$

Innovations have two key properties.

1. $\forall i \neq j,\ \langle E_i, E_j \rangle = 0$
2. $\forall i,\ \text{span}\{Y_j\}_{j=0}^i = \text{span}\{E_j\}_{j=0}^i$

We can also write innovations in terms of a matrix where $\boldsymbol{\varepsilon} = A\boldsymbol{Y}$ where $\boldsymbol{\varepsilon} = \begin{bmatrix} E_0 & E_1 & \cdots & E_n \end{bmatrix}^T$ and $\boldsymbol{Y} = \begin{bmatrix} Y_0 & Y_1 & \cdots & Y_n \end{bmatrix}^T$. Since each E_i only depends on the previous Y_i, then A must be lower triangular, and because we need each E_i to be mutually orthogonal, R_ε should be diagonal. $R_\varepsilon = AR_YA^*$, so if $R_Y \succ 0$, then we can use its unique LDL decomposition $R_Y = LDL^*$ and let $A = L^{-1}$.

2 Linear Estimation

In Linear Estimation, we are trying to estimate a random variable $\boldsymbol{X}$ using an observation $\boldsymbol{Y}$ with a linear function of $\boldsymbol{Y}$. If $\boldsymbol{Y}$ is finite dimensional, then we can say $\hat{\boldsymbol{X}}(\boldsymbol{Y}) = W\boldsymbol{Y}$ where W is some matrix. Using theorem 1 and the orthogonality principle, we know that

$$\langle \boldsymbol{X} - W\boldsymbol{Y}, \boldsymbol{Y} \rangle = \boldsymbol{0} \Leftrightarrow R_{XY} = W\boldsymbol{R}_Y$$

This is known as the **Normal Equation**. If R_Y is invertible, then we can apply the inverse to find W. Otherwise, we can apply the pseudoinverse $R_Y^\dagger$ to find W, which may not be unique. If we want to measure the quality of the estimation, since $\boldsymbol{X} = \boldsymbol{X} + (\boldsymbol{X} - \hat{\boldsymbol{X}})$,

$$\|\boldsymbol{X}\|^2 = \|\hat{\boldsymbol{X}}\|^2 + \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|^2 \implies$$
$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|^2 = \|\boldsymbol{X}\|^2 - \|\hat{\boldsymbol{X}}\|^2 = R_X - R_{XY}R_Y^{-1}R_{YX}$$

2.1 Affine Estimation

If we allow ourselves to consider an affine function for estimation $\hat{\boldsymbol{X}}(\boldsymbol{Y}) = W\boldsymbol{Y}+b$, then this is equivalent to instead finding an estimator

$$\hat{\boldsymbol{X}}(\boldsymbol{Y}') = W\boldsymbol{Y}' \qquad \text{where } \boldsymbol{Y}' = \begin{bmatrix} \boldsymbol{Y} \\ 1 \end{bmatrix}$$

This is equivalent to the following orthogonality conditions:

1. $\langle \boldsymbol{X} - \hat{\boldsymbol{X}}, \boldsymbol{Y} \rangle$
2. $\langle \boldsymbol{X} - \hat{\boldsymbol{X}}, 1 \rangle$

Solving gives us

$$\hat{\boldsymbol{X}}(\boldsymbol{Y}) = W(\boldsymbol{Y} - \boldsymbol{\mu}_Y) + \mu_x \qquad \text{where } W\Sigma_Y = \Sigma_{XY}.$$

Σ_Y and Σ_{XY} are the auto-covariance and cross-covariance respectively. Recall that if

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_Y \end{bmatrix}, \begin{bmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{bmatrix} \right)$$

then

$$\boldsymbol{X}|\boldsymbol{Y} \sim \mathcal{N}\left(\boldsymbol{\mu}_X + \Sigma_{XY}\Sigma_Y^{-1}(\boldsymbol{Y} - \boldsymbol{\mu}_Y), \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{YX} \right)$$

Thus in the Joint Gaussian case, the mean of the conditional distribution is the best affine estimator of $\boldsymbol{X}$ using $\boldsymbol{Y}$, and the covariance is the estimation error. This has two interpretations.

1. Under the Gaussian assumption, the best nonlinear estimator $\mathbb{E}\left[\boldsymbol{X}|\boldsymbol{Y}\right]$ is affine
2. Gaussian random variables are the hardest predict because nonlinearity should improve our error, but it does not in the Gaussian case. This means if affine estimation works well, we shouldn't try and find better non-linear estimators.

2.2 Least Squares

The theory of linear estimation is very closely connected with the theory behind least squares in linear algebra. In least squares, we have a deterministic $\boldsymbol{x}$ and assume nothing else about it, meaning we are looking for an unbiased estimator. theorem 2 tells us how to find the best linear unbiased estimator in a linear setting.

Theorem 2 (Gauss Markov Theorem) *Suppose that* $\boldsymbol{Y} = H\boldsymbol{x} + \boldsymbol{Z}$ *and* Z *is zero-mean with* $\langle \boldsymbol{Z}, \boldsymbol{Z} \rangle = \boldsymbol{I}$*,* H *is full-column rank, then* $\hat{\boldsymbol{x}}_b = (H^*H)^{-1}H^*\boldsymbol{Y}$ *is the best linear unbiased estimator.*

2.2.1 Recursive Least Squares

Suppose we extend the least squares setup to allow a stochastic, but fixed, $\boldsymbol{X}$ where $\langle \boldsymbol{X}, \boldsymbol{X} \rangle = \Pi_0$. At each timestep, we receive observations of $\boldsymbol{X}$ such that $\boldsymbol{Y}_i = h_i^* \boldsymbol{X} + \boldsymbol{V}_i$ where $\langle \boldsymbol{V}_i, \boldsymbol{V}_j \rangle = \delta[i, j]$ and $\langle \boldsymbol{X}, \boldsymbol{V} \rangle$. Define

$$\boldsymbol{Y}^i = \begin{bmatrix} \boldsymbol{Y}_0 \\ \boldsymbol{Y}_1 \\ \cdots \\ \boldsymbol{Y}_i \end{bmatrix} \qquad H_i = \begin{bmatrix} h_0^* \\ h_1^* \\ \vdots \\ h_i^* \end{bmatrix} \qquad \boldsymbol{V}^i = \begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \cdots \\ \boldsymbol{V}_i \end{bmatrix}$$

Then our setup becomes $\boldsymbol{Y}^i = H_i \boldsymbol{X} + \boldsymbol{V}^i$.

$$R_{XY^i} = \Pi_0 H_i^* \qquad R_{Y^i} = (H_i \Pi_0 H_i^* + I)$$

Applying theorem 1 and solving the normal equation, we see

$$\begin{aligned} W &= \Pi_0 H_i^* (H_i \Pi_0 H_i^* + I)^{-1} = \Pi_0 H_i^* (I - H_i (\Pi_0^{-1} + H_i^* H_i)^{-1} H_i^*) \\ &= \Pi_0 (I - H_i^* H_i (\Pi_0^{-1} + H_i^* H_i)^{-1}) H_i^* \\ &= \Pi_0 ((\Pi_0^{-1} + H_i^* H_i)(H_i^* H_i)^{-1} (H_i^* H_i)(\Pi_0^{-1} + H_i^* H_i)^{-1} - H_i^* H_i (\Pi_0^{-1} + H_i^* H_i)^{-1}) H_i^* \\ &= \Pi_0 \Pi_0^{-1} (H_i^* H_i)^{-1} H_i^* H_i (\Pi_0^{-1} + H_i^* H_i)^{-1} H_i^* \\ &= (\Pi_0^{-1} + H_i^* H_i)^{-1} H_i^* \end{aligned}$$

Suppose we want to do this in an online fashion where at each timestep i, we only use the current $h_i, \boldsymbol{Y}_i$ and our previous estimate $\boldsymbol{X}_{i-1}$. Let $P_i = (\Pi_0^{-1} + H_i^* H_i)^{-1}$. Then

$$P_i^{-1} = \Pi_0 + \sum_{k=0}^{i} h_k h_k^* = P_{i-1}^{-1} + h_i h_i^*.$$

By applying the Sherman-Morrison-Woodbury identity, we can see that

$$P_i = P_{i-1} = P_{i-1} \frac{h_i h_i^*}{1 + h_i^* P_{-1} h_i} P_{i-1}$$

Theorem 3 (Recursive Least Squares Update) *The best least squares estimate using $i+1$ data points can be found by updating the best least squares estimate using i data points using*

$$\hat{\boldsymbol{X}}_i = \hat{\boldsymbol{X}}_{i-1} + \frac{P_{i-1} h_i}{1 + h_i^* P_{i-1} h_i} (\boldsymbol{Y}_i - h_i^* \hat{\boldsymbol{X}}_{i-1})$$

Notice that this formula scales an innovation in order to improve the current estimate of $\boldsymbol{X}$.

Just as we could compute a recursive update, we can also compute a "downdate" where we forget a particular observation. More concretely, we want to use $\hat{\boldsymbol{X}}_i$ to find $\hat{\boldsymbol{X}}_{i|k}$, the best linear estimator of $\boldsymbol{X}$ using $\boldsymbol{Y}_0, \boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_{k-1}, \boldsymbol{Y}_{k+1}, \cdots, \boldsymbol{Y}_i$. Defining $P_{i|k} = (\Pi_0^{-1} + H_{i|k}^* H_{i|k})^{-1}$,

$$P_{i|k}^{-1} = \Pi_0^{-1} + \sum_{j=0, j\neq k}^{i} h_j h_j^* = P_i^{-1} - h_k h_k^{-1}.$$

Applying the Sherman-Morrison-Woodbury identity,

$$P_{i|k} = P_i + P_i \frac{h_k h_k^*}{h_k^* P_i h_k - 1} P_i$$

Theorem 4 (Recursive Least Squares Downdate) *The best least squares estimate using all but the kth observation can be found by updating the best least squares estimate using all data points using*

$$\hat{\boldsymbol{X}}_{i|k} = \hat{X}_i + \frac{P_i h_k}{h_k^* P_i h_k - 1}(Y_k - h_k^* \hat{\boldsymbol{X}}_i)$$

3 Discrete Time Random Processes

Definition 4 *A Discrete-Time Random Process is a countably infinite collection of random variables on the same probability space $\{X_n : n \in \mathbb{Z}\}$.*

Discrete Time Random Processes have a mean function $\mu_n = \mathbb{E}\left[X_n\right]$ and an autocorrelation function $R_X(n_1, n_2) = \mathbb{E}\left[X_{n_1} X_{n_2}^*\right]$

3.1 Wide-Sense Stationary Random Processes

Definition 5 *A Wide-Sense Stationary Random Process is a disrete-time random process with constant mean, finite variance, and an autocorrelation function that can be re-written to only depend on $n_1 - n_2$.*

We call this wide-sense stationary because the mean and covariance do not change as the process evolves. In a strict-sense stationary process, the distribution of each random variable in the process would not change.

Definition 6 *A WSS process $Z \sim \mathcal{WN}(0, \sigma^2)$ is a white noise process with variance σ^2 if and only if $\mathbb{E}\left[Z_n\right] = 0$ and $\mathbb{E}\left[Z_n Z_m^*\right] = \sigma^2 \delta[n, m]$.*

3.1.1 Spectral Density

Recall that the Discrete Time Fourier Transform is given by

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}.$$

The Inverse Discrete Time Fourier Transform is given by

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}d\omega.$$

Since the DTFT is an infinite summation, it may or may not converge.

Definition 7 *A signal $x[n]$ belongs to the l^1 class of signals if the series converges absolutely. In other words,*

$$\sum_{k=-\infty}^{\infty} |x[k]| < \infty.$$

This class covers most real-world signals.

Theorem 5 *If $x[n]$ is a l^1 signal, then the DTFT $X(e^{j\omega})$ converges uniformly and is well-defined for every ω. $X(e^{j\omega})$ is also a continuous function.*

Definition 8 *A signal $x[n]$ belongs to the l^2 class of signals if it is square summable. In other words,*

$$\sum_{k=-\infty}^{\infty} |x[k]|^2 < \infty.$$

The l^2 class contains important functions such as sinc.

Theorem 6 *If $x[n]$ is a l^2 signal, then the DTFT $X(e^{j\omega})$ is defined almost everywhere and only converges in the mean-squared sense:*

$$\lim_{N\to\infty}\int_{-\pi}^{\pi}\left|\left(\sum_{k=-N}^{N} x[k]e^{-j\omega n}\right) - X(\omega)\right|^2 d\omega = 0$$

Tempered distributions like the Dirac Delta function are other functions which are important for computing the DTFT, and they arise from the theory of generalized functions.

Suppose we want to characterize the signal using its DTFT.

Definition 9 *The energy of a deterministic, discrete-time signal $x[n]$ is given by*

$$\sum_{n\in\mathbb{Z}} |x[n]|^2.$$

The autocorrelation of $x[n]$, given by $a[n] = x[n] * x^*[-n]$, is closely related to the energy of the signal since $a[0] = \sum_{n\in\mathbb{Z}} |x(n)|^2$.

Definition 10 *The Energy Spectral Density $x[n]$ with auto-correlation $a[n]$ is given by*

$$A(e^{j\omega}) = \sum_{n\in\mathbb{Z}} a[n]e^{-j\omega n}$$

We call the DTFT of the autocorrelation the energy spectral density because, by the Inverse DTFT,

$$a[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(e^{j\omega})d\omega.$$

Since summing over each frequency gives us the energy, we can think of $A(e^{j\omega})$ as storing the energy density of each spectral component of the signal. We can apply this same idea to wide-sense stationary stochastic processes.

Definition 11 *The Power Spectral Density of a Wide-Sense Stationary random process is given by*

$$S_X(e^{j\omega}) = \sum_{k\in\mathbb{Z}} R_X(k)e^{-j\omega k}.$$

Note that when considering stochastic signals, the metric changes from energy to power. This is because if X_n is Wide-Sense Stationary, then

$$\mathbb{E}\left[\sum_{n\in\mathbb{Z}} |X_n|^2\right] = \infty,$$

so energy doesn't even make sense. To build our notion of power, let $A_T(\omega)$ be a truncated DTFT of the auto-correlation of a wide-sense stationary process, then

$$\begin{aligned}
\lim_{T\to\infty} \frac{\mathbb{E}\left[A_T(e^{j\omega})\right]}{2T+1} &= \lim_{T\to\infty} \frac{1}{2T+1} \left(\sum_{n=-T}^{T} x[n]e^{-j\omega n}\right) \left(\sum_{m=-T}^{T} x^*[m]e^{j\omega m}\right) \\
&= \lim_{T\to\infty} \frac{1}{2T+1} \sum_{n,m\in[-T,T]} \mathbb{E}\left[x[n]x^*[m]\right] e^{-j\omega(n-m)} \\
&= \lim_{T\to\infty} \frac{1}{2T+1} \sum_{n,m\in[-T,T]} R_x(n-m)e^{-j\omega(n-m)} \\
&= \lim_{T\to\infty} \sum_{k=-2T}^{2T} R_X(k)e^{-j\omega k} \left(1 - \frac{|k|}{2T+1}\right) \\
&= \sum_{k=-\infty}^{\infty} R_X(k)e^{-j\omega k}
\end{aligned}$$

The DTFT of the auto-correlation function naturally arises out of taking the energy spectral density and normalizing it by time (the truncated sequence is made of $2T+1$ points). In practice, this means to measure the PSD, we need to either use the distribution of the signal to compute R_X, or estimate the PSD by averaging multiple realizations of the signal.

The inverse DTFT formula tells us that we can represent a deterministic, discrete-time signal $x[n]$ as a sum of complex exponentials weighted by $\frac{X(e^{j\omega})d\omega}{2\pi}$. This representation has an analog for stochastic signals as well.

Theorem 7 (Cramer-Khinchin) *For a complex-valued WSS stochastic process X_n with power spectral density $S_X(\omega)$, there exists a unique right-continuous stochastic process $F(\omega), \omega \in (-\pi, \pi]$ with square-integrable, orthogonal increments such that*

$$X_n = \int_{-\pi}^{\pi} e^{j\omega n} dF(\omega)$$

where for any interval $[\omega_1, \omega_2], [\omega_3, \omega_4] \subset [-\pi, \pi]$,

$$\mathbb{E}\left[(F(\omega_2) - F(\omega_1))(F(\omega_4) - F(\omega_3))^*\right] = f((\omega_1, \omega_2] \cap (\omega_3, \omega_4])$$

where f is the structural measure of the stochastic process and has Radon-Nikodym derivative $\frac{S_X(e^{j\omega})}{2\pi}$.

Besides giving us a decomposition of a WSS random process, theorem 7 tells a few important facts.

1. $\omega_1 \neq \omega_2 \implies \langle dF(\omega_1), dF(\omega_2)\rangle = 0$ (i.e different frequencies are uncorrelated).
2. $\mathbb{E}\left[|dF(\omega)|^2\right] = \frac{S_X(e^{j\omega})d\omega}{2\pi}$

3.1.2 Z-Spectrum

Recall that the Z-transform converts a discrete-time signal into a complex representation. It is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}.$$

It is a special type of series called a **Laurent Series**.

Theorem 8 *A Laurent Series will converge absolutely on an open annulus*

$$A = \{z | r < |z| < R\}$$

for some r and R.

We can compute r and R using the signal $x[n]$.

$$r = \limsup_{n\to\infty} |x[n]|^{\frac{1}{n}}, \qquad \frac{1}{R} = \limsup_{n\to\infty} |x[-n]|^{\frac{1}{n}}.$$

In some cases, it can be useful to only compute the Z-transform of the right side of the signal.

Definition 12 *The unilateral Z-transform of a sequence $x[n]$ is given by*

$$[X(z)]_+ = \sum_{n=0}^{\infty} x[n]z^{-n}$$

If the Z-transform of the sequence is a rational function, then we can quickly compute what the unilateral Z-transform will be by leveraging its partial fraction decomposition.

Theorem 9 *Any arbitrary rational function $H(z)$ with region of convergence including the unit circle corresponds with the unilateral Z-transform*

$$[H(z)]_+ = r_0 + \sum_{i=1}^{m}\sum_{k=1}^{l_i} \frac{r_{ik}}{(z+\alpha_i)^k} + \sum_{i=m+1}^{n}\sum_{k=1}^{l_i} \frac{r_{ik}}{\beta_i^k}$$

where $|\alpha_i| < 1 < |\beta_i|$.

Definition 13 *For two jointly WSS processes X_n, Y_n, the z-cross spectrum is the Z-Transform of the correlation function $R_{YX}(k) = \mathbb{E}\left[Y_n X_{n-k}^*\right]$.*

$$S_{YX}(z) = \sum_{k\in\mathbb{Z}} R_{YX}(k)z^{-k}$$

Using this definition, we can see that

$$S_{XY}(z) = S^*_{YX}(z^{-*}).$$

We can also look at the Z-transform of the auto-correlation function of a WSS process X to obtain $S_X(z)$.

Definition 14 *For a rational function $S_X(z)$ with finite power $\left(\int_{-\pi}^{\pi} S_X(e^{j\omega})d\omega < \infty\right)$ and is strictly positive on the unit circle, the canonical spectral factorization decomposes $S_X(z)$ into a product of a $r_e > 0$ and the transfer function of a minimum phase system $L(z)$ with $L(\infty) = 1$*

$$S_X(z) = L(z)r_e L^*(z^{-*})$$

Because $L(z)$ is minimum phase and $L(\infty) = 1$, it must take the form

$$L(z) = 1 + \sum_{i=1}^{\infty} l[i]z^{-i}$$

since minimum phase systems are causal. Using definition 14, we can express $S_X(z)$ as the product of a right-sided and left-sided process.

$$S_X(z) = (\sqrt{r_e}L(z))(\sqrt{r_e}L^*(z^{-*})) = S_X^+(z)S_X^-(z)$$

Note that $S_X^-(e^{j\omega}) = \left(S_X^+(e^{j\omega})\right)^*$. Using the assumptions built into definition 14, we can find a general form for $L(z)$ since we know $S_Y(z)$ takes the following form

$$S_Y(z) = r_e \frac{\prod_{i=1}^m (z-\alpha_i)(z^{-1}-\alpha_i^*)}{\prod_{i=1}^n (z-\beta_i)(z^{-1}-\beta_i^*)} \quad |\alpha_i| < 1, |\beta_i| < 1, r_e > 0.$$

If we let the $z - \alpha_i$ and $z - \beta_i$ terms be part of $L(z)$, then

$$L(z) = z^{n-m}\frac{\prod_{i=1}^m (z-\alpha_i)}{\prod_{i=1}^n (z-\beta_i)}.$$

3.2 Markov Processes

Definition 15 *We say that random variables X, Y, Z form a Markov Triplet X—Y—Z if and only if X and Z are conditionally independent on Y*

Mathematically, Markov triplets satisfy three properties.

1. $p(x, z|y) = p(x|y)p(z|y)$

2. $p(z|x, y) = p(z|y)$

3. $p(x|y, z) = p(x|y)$

Because of these rules, the joint distribution can be written as $p(x, y, z) = p(x)p(y|x)p(z|y)$.

Theorem 10 *Random variables X, Y, Z form a Markov triplet if and only if there exist ϕ_1, ϕ_2 such that $p(x, y, z) = \phi_1(x, y)\phi_2(y, z)$.*

To simplify notation, we can define $X_m^n = (X_m, X_{m+1}, \cdots, X_n)$ and $X^n = X_1^n$.

Definition 16 *A Markov Process is a Discrete Time Random Process $\{X_n\}_{n\geq 1}$ where X_n—X_{n-1}—X^{n-2} for all $n \geq 2$*

Because of the conditional independence property, we can write the joint distribution of all states in the Markov process as

$$p(x^n) = \prod_{t=1}^{n} p(x_t|x^{t-1}) = \prod_{t=1}^{n} p(x_t|x_{t-1}).$$

The requirement for X—Y—Z to satisfy $p(x, y, z) = p(x)p(y|x)p(z|y)$ is a very strict requirement. If we wanted to create a "wider" requirement of Markovity, then we could settle for $\hat{X}(Y) = \hat{X}(Y, Z)$ where $\hat{X}$ is the best linear estimator of X since this property is satisfied by all Markov triplets, but does not imply a Markov Triplet.

Definition 17 *Random variables X, Y, Z form a Wide Sense Markov Triplet X—Y—Z if and only if the best linear estimator of X given Y is identical to the best linear estimator of X given Y and Z.*

$$\hat{X}(Y) = \hat{X}(Y, Z)$$

Definition 18 *A stochastic process $\{Y_i\}_{i=0}^n$ is a Wide-Sense Markov Process if and only if for any $1 \leq i \leq n-1$, Y_{i+1}—Y_i—Y^{i-1} forms a Wide-Sense Markov Triplet.*

All Wide-Sense Markov models have a very succint representation.

Theorem 11 *A process $\boldsymbol{X}$ is Wide-Sense Markov if and only if $\boldsymbol{X}_{i+1} = F_i\boldsymbol{X}_i + G_i\boldsymbol{U}_i$ and*

$$\langle \begin{bmatrix} U_i \\ \boldsymbol{X}_0 \end{bmatrix}, \begin{bmatrix} U_j \\ \boldsymbol{X}_0 \end{bmatrix} \rangle = \begin{bmatrix} Q_i\delta[i-j] & 0 \\ 0 & \Pi_0 \end{bmatrix}$$

3.2.1 Hidden Markov Processes

Definition 19 *If* $\{X_n\}_{n\geq 1}$ *is a Markov Process, then* $\{Y_n\}_{n\geq 1}$ *is a Hidden Markov Process if we can factorize the conditional probability density*

$$p(y^n, x^n) = \prod_{i=1}^{n} p(y_i|x_i)$$

We can think of Y as a noisy observation of an underlying Markov Process. The joint distribution of $\{X_n\}_{n\geq 1}$ and $\{Y_n\}_{n\geq 1}$ can be written as

$$p(x^n, y^n) = p(x^n)p(y^n|x^n) = \prod_{t=1}^{n} p(x_t|x_{t-1}) \prod_{i=1}^{n} p(y_i|x_i).$$

Hidden Markov Models can be represented by undirected graphical models. To create an undirected graphical model,

1. Create a node for each random variable.
2. Draw an edge between two nodes if a factor of the joint distribution contains both nodes.

Undirected graphical models of Hidden Markov Processes are useful because they let us derive additional Markov dependedepencies between groups of variables.

Theorem 12 *For 3 disjoint sets* S_1, S_2, S_3 *of notes in a graphical model, if any path from* S_1 *to* S_3 *passes through a node in* S_2*, then* S_1—S_2—S_3.

3.2.2 State-Space Models

Suppose we have a discrete-time random process which evolves in a recursive fashion, meaning the current state depends in some way on the previous state. We can express this recursion with a set of equations.

Definition 20 *The standard state space model describes random processes which describe the evolution of state vectors* $\boldsymbol{X}_i$ *and observation vectors* $\boldsymbol{Y}_i$ *according to the equations*

$$\begin{cases} \boldsymbol{X}_{i+1} = F_i \boldsymbol{X}_i + G_i \boldsymbol{U}_i \\ \boldsymbol{Y}_i = H_i \boldsymbol{X}_i + \boldsymbol{V}_i \end{cases}$$

with initial condition

$$\langle \begin{bmatrix} \boldsymbol{X}_0 \\ \boldsymbol{U}_i \\ \boldsymbol{V}_i \end{bmatrix}, \begin{bmatrix} \boldsymbol{X}_0 \\ \boldsymbol{U}_j \\ \boldsymbol{V}_j \end{bmatrix} \rangle = \begin{bmatrix} \Pi_0 & 0 & 0 \\ 0 & Q_i\delta[i-j] & S_i\delta[i-j] \\ 0 & S_i^*\delta[i-j] & R_i\delta[i-j] \end{bmatrix}$$

From theorem 11, we can easily see that state space models are Wide-Sense Markov. Note that U_i and V_i are white noise, and that the dynamics of the system can change at every time step. From these equations, we can derive six different properties. Let $\Pi_i = \langle \boldsymbol{X}_i, \boldsymbol{X}_i \rangle$ and $\Phi_{i,j} = \prod_{k=j}^{i-1} F_k$ and $\Phi_{i,u} = I$.

1. $\forall i \geq j,\ \langle \boldsymbol{U}_i, \boldsymbol{X}_j \rangle = 0,\ \langle \boldsymbol{V}_i, \boldsymbol{X}_j \rangle = 0$
2. $\forall i > j,\ \langle \boldsymbol{U}_i, \boldsymbol{Y}_j \rangle = 0,\ \langle \boldsymbol{V}_i, \boldsymbol{Y}_j \rangle = 0$
3. $\forall i,\ \langle \boldsymbol{U}_i, \boldsymbol{Y}_i \rangle = S_i,\ \langle \boldsymbol{V}_i, \boldsymbol{Y}_i \rangle = R_i$
4. $\Pi_{i+1} = F_i \Pi_i F_i^* + G_i Q_i G_i^*$
5. $$\langle \boldsymbol{X}_i, \boldsymbol{X}_j \rangle = \begin{cases} \Phi_{i,j}\Pi_j & i \geq j \\ \Pi_i \Phi_{j,i}^* & i \leq j \end{cases}$$
6. $$\langle \boldsymbol{Y}_i, \boldsymbol{Y}_j \rangle = \begin{cases} H_i \Phi_{i,j+1} N_j & i > j \\ R_i + H_i \Pi_i H_i^* & i = j \\ N_i^* \Phi_{j,i+1}^* H_j^* & i < j \end{cases} \text{ where } N_i = F_i \Pi_i H_i^* + G_i S_i$$

4 Filtering

If we think of our signal as a discrete time random process, then like a normal deterministic signal, we can try filtering our random process. Filtering can either

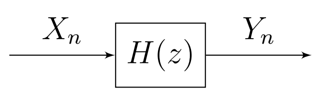

Figure 1: Filtering a Disrete Time Random Process with an LTI system with transfer function $H(z)$

be accomplished with an LTI system or some other non-linear/non-time-invariant system just like with deterministic signals.

4.1 LTI Filtering on WSS Processes

If we use an LTI filter on a WSS process, then we can easily compute how the filter impacts the spectrum of the signal.

Theorem 13 *When $Y(n)$ is formed by passing a WSS process X_n through a stable LTI system with impulse response $h[n]$ and transfer function $H(z)$, then $S_Y(z) = H(z)S_X(z)H^*(z^{-*})$ and $S_{YX}(z) = H(z)S_X(z)$. If we have a third process Z_n that is jointly WSS with (Y_n, X_n), then $S_{ZY}(z) = S_{ZX}(z)H^*(z^{-*})$.*

This gives us an interesting interpretation of the spectral factorization (definition 14) since it essentially passing a WSS process with auto-correlation $R_W(k) = r_e\delta[n]$ through a minimum-phase filter with transfer function $L(z)$.

4.2 Wiener Filter

Suppose we have a stochastic WSS process Y_n that is jointly WSS with X_n and that we want to find the best linear estimator of X_n using Y_n. The best linear estimator of X_n given the observations Y_n can be written as

$$\hat{X}_n = \sum_{m\in\mathbb{Z}} h(m)Y_{n-m} = h[n] * Y_n.$$

This is identical to passing Y_n through an LTI filter. If we restrict ourselves to using $\{Y_i\}_{i=-\infty}^{n}$ to estimate X_n, then the best linear estimator can be written as

$$\hat{X}_n = \sum_{m=0}^{\infty} h(m)Y_{n-m} = h[n] * Y_n.$$

It is identical to passing Y_n through a causal LTI filter. Since we are trying to find a best linear estimator, it would be nice if each of the random variables we are using for estimating were uncorrelated with each other. In other words, instead of using Y directly, we want to transform Y into a new process W where $R_W(k) = \delta[k]$. This transformation is known as whitening. From the spectral factorization of Y, we know if we use the filter $G(z) = \frac{1}{S_Y^+(z)}$ then

$$S_W(z) = \frac{S_Y(z)}{S_Y^+(z)S_Y^{+*}(z^{-*})} = \frac{S_Y(z)}{S_Y^+(z)S_Y^-(z)} = 1.$$

Now we want to find the best linear estimator of X using our new process W by designing an LTI filter $Q(z)$.

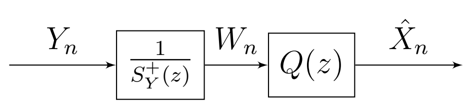

Figure 2: Finding the best linear estimator of X using W with a two-stage filter that first whitens the input.

4.2.1 Non-Causal Case

Starting with noncausal case, we can apply the orthogonality principle,

$$\mathbb{E}\left[(X_n - \hat{X}_n)W_{n-k}^*\right] = 0 \implies \mathbb{E}\left[X_n W_{n-k}^*\right] = \sum_{m\in\mathbb{Z}} q(m)\mathbb{E}\left[W_{n-m}W_{n-k}^*\right]$$

$$\therefore R_{XW}(k) = \sum_{m\in\mathbb{Z}} q(m)R_W(k-m) \implies S_{XW}(z) = Q(z)S_W(z)$$

$$\therefore Q(z) = \frac{S_{XW}(z)}{S_W(z)} = S_{XW}(z) = S_{XY}(z)(S_Y^+(z^{-*}))^{-*} = \frac{S_{XY}(z)}{S_Y^-(z)}$$

When we cascade these filters,

$$H(z) = Q(z)G(z) = \frac{S_{XY}(z)}{S_Y^-(z)}\frac{1}{S_Y^+(z)} = \frac{S_{XY}(z)}{S_Y(z)}.$$

Definition 21 *The best linear estimator of X_n using Y_n where (X_n, Y_n) is jointly WSS is given by the non-causal Wiener filter.*

$$H(z) = \frac{S_{XY}(z)}{S_Y(z)}$$

If we interpret definition 21 in the frequency domain, for a specific ω, we can understand $H(e^{j\omega})$ as an optimal linear estimator for $F_X(\omega)$ where $F_X(\omega)$ is the the stochastic process given by the Cramer-Khinchin decomposition (theorem 7). More specifically, we can use the Cramer-Khinchin decomposition of Y_n.

$$\begin{aligned}
\hat{X}_n &= \sum_{i\in\mathbb{Z}} h[i]\int_{-\pi}^{\pi} e^{j\omega(n-i)}dF_Y(\omega) \\
&= \int_{-\pi}^{\pi}\left(\sum_{i\in\mathbb{Z}} h[i]e^{-j\omega i}\right)e^{j\omega n}dF_Y(\omega) \\
&= \int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}dF_Y(\omega)
\end{aligned}$$

Since F_X and F_Y have jointly orthogonal increments, this tells us that $H(e^{j\omega})$ is just the optimal linear estimator of $dF_X(\omega)$ using $dF_Y(\omega)$. $dF_X(\omega)$ and $dF_Y(\omega)$ exist on a Hilbert space, meaning we are essentially projecting each frequency component of X_n onto the corresponding frequency component of Y_n.

4.2.2 Causal Case

First, note that in the causal case, whitening doesn't break causality because $\frac{1}{S_Y^+(z)}$ is causal. When we apply the orthogonality principle,

$$\mathbb{E}\left[(X_n - \hat{X}_n)W_{n-k}^*\right] = 0 \implies \mathbb{E}\left[X_n W_{n-k}^*\right] = \sum_{m=0}^{\infty} q(m)\mathbb{E}\left[W_{n-m}W_{n-k}^*\right]$$

$$\therefore R_{XW}(k) = \sum_{m=0}^{\infty} q[m]R_W(k-m) \qquad k \geq 0$$

We can't take the Z-transform of both sides because the equation is not necessarily true for $k < 0$. Instead, we can look at the function

$$f(k) = R_{XW}(k) - \sum_{m=0}^{\infty} R_W(k-m)q[m] = \begin{cases} 0 & k \geq 0, \\ ? & \text{else.} \end{cases}$$

Taking the unilateral Z-transform of both sides,

$$\begin{aligned} [F(z)]_+ &= [S_{XW}(z) - S_W(z)Q(z)]_+ = [S_{XW}(z)]_+ - Q(z) = 0 \\ Q(z) &= [S_{XW}(z)]_+ = \left[\frac{S_{XY}(z)}{S_Y^-(z)}\right]_+ \end{aligned}$$

Thus the filter H which gives the causal best linear estimator of X using Y is

$$H(z) = Q(z)G(z) = \left[\frac{S_{XY}(z)}{S_Y^-(z)}\right]_+ \frac{1}{S_Y^+(z)}.$$

Definition 22 *The best linear estimator of X_n using $\{Y_i\}_{i=-\infty}^{n}$ is given by the causal Wiener filter.*

$$H(z) = Q(z)G(z) = \left[\frac{S_{XY}(z)}{S_Y^-(z)}\right]_+ \frac{1}{S_Y^+(z)}.$$

Intuitively, this should make sense because we are using the same W process as in the non-causal case, but only the ones which we are allowed to use, hence use the unilateral Z-transform of the non-causal Wiener filter, which amounts to truncated the noncausal filter to make it causal.

Theorem 14 *If $\hat{X}_{NC}(n)$ is the non-causal Wiener filter of X, then the causal wiener filter of X given Y is the same as the causal wiener filter of $\hat{X}_{NC}$ given Y, and if Y is white noise, then*

$$\hat{X}_C(n) = \sum_{i=0}^{\infty} h[i]Y_{n-i}$$

4.2.3 Vector Case

Suppose that instead of a Wide-Sense Stationary process, we an N length signal $\boldsymbol{X}$ which we want to estimate with another N length signal $\boldsymbol{Y}$. We can represent both $\boldsymbol{X}$ and $\boldsymbol{Y}$ as vectors in $\mathbb{C}^N$. If we are allowed to use all entries of $\boldsymbol{Y}$ to estimate $\boldsymbol{X}$, this is identical to linear estimation.

Definition 23 *The non-causal Wiener filter of a finite length N signal $\boldsymbol{Y}$ is given by*

$$K_s = R_{\boldsymbol{XY}} R_{\boldsymbol{Y}}^{-1}.$$

Note that this requires $R_{\boldsymbol{Y}} \succ 0$. Suppose that we wanted to design a causal filter for the vector case, so $\hat{X}_i$ only depends on $\{Y_j\}_{j=1}^i$. By the orthogonality principle,

$$\forall 1 \leq l \leq i,\ \mathbb{E}\left[X_i - \sum_{j=1}^{i} K_{f,ij} Y_j Y_l^*\right] = 0 \implies R_{\boldsymbol{XY}}(i,l) = \sum_{j=1}^{i} K_{f,ij} R_{\boldsymbol{Y}}(j,l)$$

In matrix form, this means

$$R_{\boldsymbol{XY}} - K_f R_{\boldsymbol{Y}} = U^+$$

where U^+ is strictly upper triangular.

Theorem 15 *If matrix $H \succ 0$, then there exists a unique lower-diagonal upper triangular factorization of $H = LDL^*$ where L is lower diagonal and invertible with unit diagonal entries and D is diagonal with positive entries.*

Applying the LDL decomposition, we see that

$$\begin{aligned} R_{\boldsymbol{XY}} - K_f LDL^* = U^+ \implies R_{\boldsymbol{XY}} L^{-*} D^{-1} - K_f L &= U^+ L^{-*} D^{-1} \\ \therefore [R_{\boldsymbol{XY}} L^{-*} D^{-1}]_L - K_f L &= 0 \end{aligned}$$

where $[\cdot]_L$ represent the lower triangular part of a matrix.

Definition 24 *The causal Wiener filter of a finite length N signal $\boldsymbol{Y}$ is given by*

$$K_f = [R_{\boldsymbol{XY}} L^{-*} D^{-1}]_L L^{-1}$$

4.3 Hidden Markov Model State Estimation

Suppose we have a Hidden Markov Process $\{Y_n\}_{n\geq 1}$. We can think of determining the state $\{X_n\}_{n\geq 1}$ as filtering $\{Y_n\}_{n\geq 1}$.

4.3.1 Causal Distribution Estimation

Suppose we want to know the distribution of X_t after we have observered Y^t.

$$\begin{aligned}
p(x_t|y^t) &= \frac{p(x_t, y^t)}{p(y^t)} = \frac{p(x_t)p(y_t, y^{t-1}|x_t)}{\sum_x p(y_t, y^{t-1}|x_t = x)p(x_t = x)} \\
&= \frac{p(x_t)p(y_t|x_t)p(y^{t-1}|x_t)}{\sum_x p(y_t|x_t = x)p(y^{t-1}|x_t = x)p(x_t = x)} = \frac{p(y_t|x_t)p(y^{t-1})p(x_t|y^{t-1})}{\sum_x p(y_t|x_t = x)p(y^{t-1})p(x_t = x|y^{t-1})} \\
&= \frac{p(y_t|x_t)p(x_t|y^{t-1})}{\sum_x p(y_t|x_t = x)p(x_t = x|y^{t-1})}
\end{aligned}$$

Now if we know $p(x_t|y^{t-1})$, then we are set.

$$\begin{aligned}
p(x_t|y^{t-1}) &= \sum_x p(x_t, x_{t-1} = x|y^{t-1}) = \sum_x p(x_{t-1} = x|y^{t-1})p(x_t|x_{t-1} = x, y^{t-1}) \\
&= \sum_x p(x_{t-1} = x|y^t)p(x_t|x_{t-1} = x)
\end{aligned}$$

Now we have a recursive algorithm for computing the distribution of x_t.

Algorithm 1: Forward Recursion

$\beta_1(x_1) = p(x_1)$;
for $t \geq 1$ **do**
  $\alpha_t(x_t) = p(x_t|y^t) = \frac{\beta_t(x_t)p(y_t|x_t)}{\sum_x \beta_t(x)p(y_t|x_t=x)}$ (Measurement Update);
  $\beta_{t+1}(x_{t+1}) = p(x_{t+1}|y^t) = \sum_x \alpha_t(x)p(x_{t+1}|x_t = x)$ (Time Update);
end

4.3.2 Non-Causal Distribution Estimation

Suppose we are allowed to non-causally filter our signal and we care about the distribution of X_t after we have observed Y^n. In other words, for $t \geq n$, we want to find $\gamma_t(x_t) = p(x_t|y^n)$. When $t = n$, $\gamma_n(x_n) = \alpha_n(x_n)$. If we continue expanding backwards, then

$$\begin{aligned}
p(x_t|y^n) &= \sum_x p(x_t, x_{t+1} = x|y^n) = \sum_x p(x_{t+1} = x|y^n)p(x_t|x_{t+1} = x, y^t, y_{t+1}^n) \\
&= \sum_x p(x_{t+1} = x|y^n)p(x_t|x_{t+1}, y^t) = \sum_x p(x_{t+1} = x|y^n)\frac{p(x_t|y^t)p(x_{t+1} = x|x_t, y^t)}{p(x_{t+1} = x|y^t)} \\
&= \sum_x \gamma_{t+1}(x)\frac{\alpha_t(x_t)p(x_{t+1} = x|x_t)}{\beta_{t+1}(x)}
\end{aligned}$$

This gives us a clear algorithm for non-causally computing the distribution of x_t.

Algorithm 2: Backward Recursion

Run Forward Recursion;
$\gamma_n(x_n) = \alpha_n(x_n)$;
for $t = n-1$ **to** *1* **do**
  $\gamma_t(x_t) = \sum_x \gamma_{t+1}(x) \frac{\alpha_t(x_t) p(x_{t+1}=x|x_t)}{\beta_{t+1}(x)}$;
end

4.3.3 State Sequence Estimation

Suppose we want to find the most likely sequence of states given our observations. This means we should compute

$$\hat{X}^n = \operatorname*{argmax}_{X^n} p(x^n|y^n)$$

$$\begin{aligned} p(x^t, y^t) &= p(x^{t-1}, y^{t-1}) p(x_t, y_t | x^{t-1}, y^{t-1}) \\ &= p(x^{t-1}, y^{t-1}) p(x_t | x^{t-1}, y^{t-1}) p(y_t | x_t, x^{t-1}, y^{t-1}) \\ &= p(x^{t-1}, y^{t-1}) p(x_t | x_{t-1}) p(y_t | x_t) \end{aligned}$$

We see that there is a recursion in the joint distribution, so if we let $V_t(x_t) = \max_{x^{t-1}} p(x^t, y^t)$, then

$$\begin{aligned} V_t(x_t) = \max_{x^{t-1}} p(x^t, y^t) &= p(y_t|x_t) \max_{x^{t-1}} p(x^{t-1}, y^{t-1}) p(x_t|x_{t-1}) \\ &= p(y_t|x_t) \max_{x^{t-1}} \left[p(x_t|x_{t-1}) \max_{x^{t-2}} p(x^{t-1}, y^{t-1}) \right] \\ &= p(y_t|x_t) \max_{x^{t-1}} p(x_t|x_{t-1}) V_{t-1}(x_{t-1}) \end{aligned}$$

The base case is that $V_1(x_1) = p(x_1)p(y_1|x_1)$. V_t is useful because $\hat{x}_n = \operatorname{argmax}_{x_n} V_n(x_n)$. This is because we can first maximize over $\hat{X}^{n-1}$ and Y^n, so the only thing left to maximize is $\hat{x}_n$. Once we have $\hat{x}_t$, then we can comptue $\hat{x}_{t-1}$ by

$$\hat{x}_{t-1} = \operatorname*{argmax}_{x_{t-1}} p(\hat{x}_t|x_{t-1}) V_{t-1}(x_{t-1}).$$

Putting these equations gives us the Viterbi algorithm.

Algorithm 3: Viterbi Algorithm

$V_1(x_1) = p(x_1)p(y_1|x_1)$;
for $t = 2$ **to** n **do**
  $V_t(x_t) = p(y_t|x_t) \max_{x_{t-1}} p(x_t|x_{t-1})V_{t-1}(x_{t-1})$;
end
$\hat{x}_n = \mathrm{argmax}_{x_n} V_n(x_n)$;
for $t = n$ **to** 2 **do**
  $\hat{x}_{t-1} = \mathrm{argmax}_x\, p(\hat{x}_t|x_{t-1} = x)V_{t-1}(x)$;
end

4.4 Kalman Filtering

In the Kalman Filter setup, we assume that the signal we would like to filter can be represented by a state-space model. We want to predict the state vectors $\hat{\boldsymbol{X}}_i$ using some linear combination of the observations $\boldsymbol{Y}_i$.

4.4.1 Kalman Prediction Filter

Suppose that we want to compute the one-step prediction. In other words, given $\boldsymbol{Y}^i$, we want to predict $\hat{\boldsymbol{X}}_{i+1}$. Our observations $\boldsymbol{Y}$ are the only thing which give us information about the state, so it would be nice if we could de-correlate all of the $\boldsymbol{Y}$. To do this, we can define the innovation process

$$\boldsymbol{e}_i = \boldsymbol{Y}_i - \hat{\boldsymbol{Y}}_{\boldsymbol{i|i-1}} = \boldsymbol{Y}_i - H_i \hat{\boldsymbol{X}}_{i|i-1}$$

The last equality follows from the state-space modela and that past observation noises are uncorrelated with the current one. Now, to compute the one-step prediction, we just need to project $\hat{\boldsymbol{X}}_i$ onto the innovations.

$$\begin{aligned}
\hat{\boldsymbol{X}}_{i+1|i} &= \sum_{j=0}^{i} \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_j \rangle R_{\boldsymbol{e},j}^{-1} \boldsymbol{e}_j \\
&= \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i + \sum_{j=0}^{i-1} \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_j \rangle R_{\boldsymbol{e},j}^{-1} \boldsymbol{e}_j \\
&= \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i + \hat{\boldsymbol{X}}_{i+1|i-1} = \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i + \hat{\boldsymbol{X}}_{i+1|i} \\
&= \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i + F_i \hat{\boldsymbol{X}}_{i|i-1}
\end{aligned}$$

The second to last equality follows from the Wide-Sense Markovity of state space models, and the last equality is due to the state evolution noises being uncorrelated. If we let $K_{p,i} = \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1}$ (called the prediction gain), then we have a recursive estimate of the optimal one-step predictor.

$$\hat{\boldsymbol{X}}_{i+1|i} = F_i\hat{\boldsymbol{X}}_{i|i-1} + K_{p,i}\boldsymbol{e}_i.$$

Now, we just need to find a recursive formulation for $K_{p,i}$ and $R_{\boldsymbol{e},i}$. Starting with $R_{\boldsymbol{e},i}$, notice that we can write $\boldsymbol{e}_i = \boldsymbol{Y}_i - H_i\hat{\boldsymbol{X}}_{i|i-1} = H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V}_i$.

$$\begin{aligned} R_{\boldsymbol{e},i} &= \langle H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V_i}, H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V_i}\rangle \\ &= H_i\langle \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}\rangle H_i^* + R_i \end{aligned}$$

To find $K_{p,i}$, we should first find $\langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i\rangle$.

$$\begin{aligned} \langle \boldsymbol{X}_{i+1}, \boldsymbol{e}_i\rangle &= F_i\langle \boldsymbol{X}_i, \boldsymbol{e}_i\rangle + G_i\langle \boldsymbol{U}_i, \boldsymbol{e}_i\rangle \\ &= F_i\langle \boldsymbol{X}_i, H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V}_i\rangle + \langle \boldsymbol{U}_i, H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V}_i\rangle \\ &= F_i\langle \boldsymbol{X}_i, \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}\rangle H_i^* + G_iS_i \\ &= F_i\langle (\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}\rangle H_i^* + G_iS_i \\ &= F_i\langle \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}\rangle H_i^* + G_iS_i \end{aligned}$$

Notice that the matrix $P_i = \langle \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}\rangle$ is the auto-correlation of the estimation error, and it shows up in both $K_{p,i}$ and $R_{\boldsymbol{e}_i}$. It would be useful to have a recursive solution for this matrix as well.

$$\begin{aligned} P_{i+1} &= \Pi_{i+1} - \langle \hat{\boldsymbol{X}}_{i+1|i}, \hat{\boldsymbol{X}}_{i+1|i}\rangle \\ &= F_i\Pi_iF_i^* + G_iQ_iG_i^* - \langle F_i\hat{\boldsymbol{X}}_{i|i-1} + K_{p,i}\boldsymbol{e}_i, F_i\hat{\boldsymbol{X}}_{i|i-1} + K_{p,i}\boldsymbol{e}_i\rangle \\ &= F_i\Pi_iF_i^* + G_iQ_iG_i^* - F_i\langle \hat{\boldsymbol{X}}_{i|i-1}, \hat{\boldsymbol{X}}_{i|i-1}\rangle F^* + K_{p,i}R_{\boldsymbol{e},i}K_{p,i}^* \\ &= F_iP_iF_i^* + G_iQ_iG_i^* - K_{p,i}R_{\boldsymbol{e},j}K_{p,i}^* \end{aligned}$$

Putting this into a concrete algorithm, we get the Kalman Prediction Filter.

Algorithm 4: Kalman Prediction Filter

$P_0 = \Pi_0$;
$\hat{\boldsymbol{X}}_{0|-1} = 0$;
for $i \geq 0$ **do**
  $\boldsymbol{e}_i = \boldsymbol{Y}_i - H_i\hat{\boldsymbol{X}}_{i|i-1}$;
  $R_{\boldsymbol{e},i} = H_iP_iH_i^* + R_i$;
  $K_{p,i} = (F_iP_iH_i^* + G_iS_i)R_{\boldsymbol{e},i}^{-1}$;
  $P_{i+1} = F_iP_iF_i^* + G_iQ_iG_i^* - K_{p,i}R_{\boldsymbol{e},i}K_{p,i}^*$;
  $\hat{\boldsymbol{X}}_{i+1|i} = F_i\hat{\boldsymbol{X}}_{i|i-1} + K_{p,i}\boldsymbol{e}_i$;
end

4.4.2 Schmidt's Modification of the Kalman Filter

The predictive Kalman filter goes directly from $\hat{\boldsymbol{X}}_{i|i-1}$ to $\hat{\boldsymbol{X}}_{i+1|i}$ without ever determining $\hat{\boldsymbol{X}}_{i|i}$. The Schmidt Modification of the Kalman filter separates the predictive kalman filter into two steps, allowing us to estimate the current state.

1. Measurement Update: Find $\hat{\boldsymbol{X}}_{i|i}$ given the latest observation $\boldsymbol{Y}_i$ and $\hat{\boldsymbol{X}}_{i|i-1}$.
2. State Evolution (Time) Update: Find $\hat{\boldsymbol{X}}_{i+1|i}$ using what we know about the state evolution.

This mimics the approach of the forward algorithm for Hidden Markov Models, which separated updates to the distribution using a time update and a measurement update. Using our innovation process,

$$\begin{aligned}\hat{\boldsymbol{X}}_{i|i} &= \sum_{j=0}^{i} \langle \boldsymbol{X}_i, \boldsymbol{e}_j \rangle R_{\boldsymbol{e},j}^{-1} \boldsymbol{e}_j \\ &= \hat{\boldsymbol{X}}_{i|i-1} + \langle \boldsymbol{X}_i, \boldsymbol{e}_j \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}^j \\ &= \hat{\boldsymbol{X}}_{i|i-1} + \langle (\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \hat{\boldsymbol{X}}_{i|i-1}, H_i(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}) + \boldsymbol{V}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}^j \\ &= \hat{\boldsymbol{X}}_{i|i-1} + P_i H_i^* R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i\end{aligned}$$

The gain on the coefficient of the innovation $K_{f,i} = P_i H_i^* R_{\boldsymbol{e},i}$ is called the Kalman Filter Gain. The error of our estimator $P_{i|i} = \langle \hat{\boldsymbol{X}}_{i|i}, \hat{\boldsymbol{X}}_{i|i} \rangle$ is given by

$$P_{i|i} = P_i - P_i H_i^* R_{\boldsymbol{e},i}^{-1} H_i P_i.$$

For the time update,

$$\begin{aligned}\hat{\boldsymbol{X}}_{i+1|i} &= F_i \hat{\boldsymbol{X}}_{i|i} + G_i \hat{\boldsymbol{U}}_{i|i} \\ &= F_i \hat{\boldsymbol{X}}_{i|i} + G_i \langle \boldsymbol{U}_i, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i = F_i \hat{\boldsymbol{X}}_{i|i} + G_i \langle \boldsymbol{U}_i, \boldsymbol{e}_i \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i \\ &= F_i \hat{\boldsymbol{X}}_{i|i} + G_i \langle \boldsymbol{U}_i, H\boldsymbol{X}_i + \boldsymbol{V}_i - H_i \hat{\boldsymbol{X}}_{i|i-1} \rangle R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i \\ &= F_i \hat{\boldsymbol{X}}_{i|i} + G_i S_i R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i\end{aligned}$$

We can re-write the error if this estimator P_{i+1} as

$$P_{i+1} = F_i P_{i|i} F_i^* + G_i(Q_i - S_i R_{\boldsymbol{e},i}^{-1} S_i^*) G_i^* - F_i K_{f,i} S_i^* G_i^* - G_i S_i K_{f,i}^* F_i^*$$

Writing this as an algorithm,

Algorithm 5: Kalman Prediction Filter

$P_0 = \Pi_0$;
$\hat{\boldsymbol{X}}_{0|-1} = 0$;
for $i \geq 0$ **do**
  $\boldsymbol{e}_i = \boldsymbol{Y}_i - H_i \hat{\boldsymbol{X}}_{i|i-1}$;
  $R_{\boldsymbol{e},i} = H_i P_i H_i^* + R_i$;
  $K_{f,i} = P_i H_i^* R_{\boldsymbol{e},i}^{-1}$;
  $\hat{\boldsymbol{X}}_{i|i} = \hat{\boldsymbol{X}}_{i|i-1} + K_{f,i} \boldsymbol{e}_i$;
  $P_{i|i} = P_i - P_i H_i^* R_{\boldsymbol{e},i}^{-1} H_i P_i$;
  $\hat{\boldsymbol{X}}_{i+1|i} = F_i \hat{\boldsymbol{X}}_{i|i} + G_i S_i R_{\boldsymbol{e},i}^{-1} \boldsymbol{e}_i$;
  $P_{i+1} = F_i P_{i|i} F_i^* + G_i(Q_i - S_i R_{\boldsymbol{e},i}^{-1} S_i^*) G_i^* - F_i K_{f,i} S_i^* G_i^* - G_i S_i K_{f,i}^* F_i^*$;
end

4.4.3 Kalman Smoother

The Kalman Prediction Filter and Schmidt's modification of the Kalman filter are both causal filters. The Kalman Smoother provides a non-causal estimate in the same way that the Backward Recursion algorithm does for Hidden Markov Processes. In other words, the Kalman Smoother predicts $\hat{\boldsymbol{X}}_{i|n}$, the best linear estimator of $\hat{\boldsymbol{X}}_i$ from $\{\boldsymbol{Y}_0, \cdots, \boldsymbol{Y}_N\}$. As before, we can start with the innovation process $\boldsymbol{e}_j = \boldsymbol{Y}_j - H_j\hat{\boldsymbol{X}}_{j|j-1}$.

$$\hat{\boldsymbol{X}}_{i|N} = \sum_{j=0}^{N}\langle \boldsymbol{X}_i, \boldsymbol{e}_j\rangle R_{e,j}^{-1}\boldsymbol{e}_j = \hat{\boldsymbol{X}}_{i|i-1} + \sum_{j=i}^{N}\langle X_i, \boldsymbol{e}_j\rangle R_{e,j}^{-1}\boldsymbol{e}_j.$$

We can compute $\hat{\boldsymbol{X}}_{i|i-1}$ using the Predictive Kalman filter, so we just need to compute the $\langle \boldsymbol{X}_i, \boldsymbol{e}_j\rangle R_{e,j}^{-1}\boldsymbol{e}_j$.

$$\begin{aligned}\langle \boldsymbol{X}_i, \boldsymbol{e}_j\rangle &= \langle \boldsymbol{X}_i, H_j(\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{j|j-1}) + \boldsymbol{V}_j\rangle = \langle \boldsymbol{X}_i, H_j(\boldsymbol{X}_j - \hat{\boldsymbol{X}}_{j|j-1})\rangle \\ &= \langle \hat{\boldsymbol{X}}_{i|i-1} + (\boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}), \boldsymbol{X}_j - \hat{\boldsymbol{X}}_{j|j-1}\rangle H_j^*\end{aligned}$$

$\boldsymbol{X}_j - \hat{\boldsymbol{X}}_{j|j-1}$ is orthgonal to any linear function of $\{\boldsymbol{Y}_0, \cdots, \boldsymbol{Y}_{j-1}\}$, so when $j \geq i$, it must be orthgonal to $\hat{\boldsymbol{X}}_{i|i-1}$ since it is a function of $\{\boldsymbol{Y}_k\}_0^{i-1}$. Thus, for $j \geq i$,

$$\langle \boldsymbol{X}_i, \boldsymbol{e}_j\rangle = \langle \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_j - \hat{\boldsymbol{X}}_{j|j-1}\rangle H_j^*$$

If we denote $P_{ij} = \langle \boldsymbol{X}_i - \hat{\boldsymbol{X}}_{i|i-1}, \boldsymbol{X}_j - \hat{\boldsymbol{X}}_{j|j-1}\rangle$, then

$$P_{ij} = P_i\Phi_p^*(j,i) = P_i\begin{cases} I & \text{if } j = i \\ \prod_{k=i}^{j-1} F_{p,k} & \text{if } j > i\end{cases}$$

where $F_{p,k} = (F_k - K_{p,k}H_k)$. This gives us the expression

$$\hat{\boldsymbol{X}}_{i|N} = \hat{\boldsymbol{X}}_{i|i-1} + P_i\sum_{j=i}^{N}\Phi_p^*(j,i)H_i^*R_{e,j}^{-1}\boldsymbol{e}_j.$$

If we let $\lambda_{i|N} = \sum_{j=i}^{N}\Phi_p^*(j,i)H_i^*R_{e,j}^{-1}\boldsymbol{e}_j$, then we get the recursion

$$\lambda i|N = F_{p,i}^*\lambda_{i+1|N} + H_i^*R_{e,i}^{-1}\boldsymbol{e}_i.$$

If we want to look at the error of this estimator, we see that

$$P_{i|N} = P_i - P_i\Lambda_{i|N}P_i \text{ where } \Lambda_{i|N} = \langle \lambda_{i|N}, \lambda_{i|N}\rangle = F_{p,i}^*\Lambda_{i+1|N}F_{p,i} + H_i^*R_{e,i}^{-1}H_i.$$

Writing all of this as an algorithm,

Algorithm 6: Kalman Smoother Filter

Run Kalman Prediction Filter;

$\lambda_{N+1|N} = 0$;

$\Lambda_{N+1|N} = 0$;

for $i = N$ **to** *0* **do**

  $\lambda_{i|N} = F_{p,i}^{*}\lambda_{i+1|N} + H_i^{*}R_{e,i}^{-1}\boldsymbol{e}_i$;

  $\hat{\boldsymbol{X}}_{i|N} = \hat{\boldsymbol{X}}_{i|i-1} + P_i\lambda_{i|N}$;

  $\Lambda_i = F_{p,i}^{*}\Lambda_{i+1|N}F_{p,i} + H_i^{*}R_{e,i}^{-1}H_i$;

  $P_{i|N} = P_i - P_i\Lambda_{i|N}P_i$;

end
