LECTURE 12

OPTIMAL OPERATORS (PART I)
(or how to design linear operators to best recover signals given their linearly distorted versions)

12.1. We have seen ways to represent classes of signals using various families of orthonormal functions (or discrete-vector signals using various bases for $\mathbb{R}^n$), and we even considered the topic of determining "best bases" or best families of functions for representing signals for which the second-order statistics are given.
The aim of best bases was to minimize the error (measured by its "length" or "mean squared error criterion") in partial approximations of signals using the first $K$ coefficients instead of all of them ($N$ in the vector case, $\infty$ in the continuous case). This aim is, of course very useful for data compression purposes, and indeed the principal component analysis, the so-called Karhunen-Loève transform, is heavily involved in the design of efficient storage and transmission of signals via communication channels.

Another very important goal of signal processing is the recovery of signals that were subject to various types of deteriorations like systematic distortions and/or contamination.
by "noise". Noise is here an unwanted signal that is usually "selected" by an adversary ("nature") and is from a class of signals with properties different from the class to which the desired signals belong. Here we shall be dealing with a simple, yet very useful model of deterioration: the distortions will be modeled by some fully understood linear operator and the noise will be added to the distorted signal and will be assumed completely unrelated to it (i.e. statistically independent.) The general set-up for the problem of SIGNAL RESTORATION will therefore be the following:
\[ \epsilon_{\omega, \tilde{\omega}} = \mathcal{H}\{\epsilon_{\omega}\} + m_{\tilde{\omega}} \]

Signal Source

(\omega-genie)

Distortion Operator

\[ \mathcal{H}\{\epsilon_{\omega}\} \]

Noise Source

(\tilde{\omega}-demon)

Combining operator.
Given the signal \( \phi_{\omega,\tilde{\omega}} \) we want an operator, \( M \{ \cdot \} \) so that when it is applied to the "data" signal it will produce a good estimate of the source signal \( \phi_{\omega} \). So in general

Given:
\[
\phi_{\omega,\tilde{\omega}} = H \{ \phi_{\omega} \} + \eta_{\tilde{\omega}}
\]

Find \( M \{ \cdot \} \) so that
\[
M \{ \phi_{\omega,\tilde{\omega}} \} = \hat{\phi}_{\omega,\tilde{\omega}} \quad \text{and}
\]

Difference measure between \( \hat{\phi}_{\omega,\tilde{\omega}} \) and \( \phi_{\omega} \) is small, i.e.
\[
\sqrt{\frac{\| \phi_{\omega} - \hat{\phi}_{\omega,\tilde{\omega}} \|_2^2}{\varepsilon^2}} \quad \text{is small/possible.}
\]
In this general context we shall assume that
• the signal is selected as a realization of
  a random process with zero mean and
  given autocorrelation, i.e.
  \[ \{ \tilde{W}_w \}_{w \in \Omega} \sim \mathbb{E}\{ \tilde{W}_w \} = 0 \quad \mathbb{E}\{ \tilde{W}_w \tilde{W}_w^* \} = R_w \]
• the distortion operation \( H_{l(s)} \) is linear
  (and often also \underline{shift-invariant}): \( H_{l(s)} \)
• the noise signal is the realization of
  a random process with zero mean and
  known autocorrelation, \( \sigma_n^2 I \)
  \[ \{ \tilde{n}_w \}_{w \in \Omega} \sim \mathbb{E}\{ \tilde{n}_w \} = 0 \quad \mathbb{E}\{ \tilde{n}_w \tilde{n}_w^* \} = \sigma_n^2 I \]
• the noise and signal processes are
  independent
• the noise is additive, i.e. the operation
  is simply the sum.
In the sequel, therefore, we shall have the following model in view:

\[ \{ \phi_w \} \rightarrow \mathcal{X} \rightarrow + \rightarrow \{ \phi_{w,\tilde{w}} \} \]

\text{LINEAR DISTORTION} \quad \{ M_{w,y} \}
\text{ADDITIVE NOISE} \quad \{ \tilde{M}_{w,y} \}

where \( \{ \phi_w \} \) obeys
\[
E_w \{ \phi_w^2 \} = 0
\]
\[
E_w \{ \phi_w(t) \phi_w^*(t') \} = R_{\phi_w}(t, t') \text{ given or } R_{\phi_w}(t, t')
\]

\( \mathcal{X} \cdot f \) is described by a known matrix
\[
\mathbf{H} \quad \text{or } h(t, t')
\]

\( \tilde{M}_{\tilde{w},y} \) obeys
\[
E_{\tilde{w}} \{ \tilde{m}_{\tilde{w}}^2 \} = 0
\]
\[
E_{\tilde{w}} \{ \tilde{m}_{\tilde{w}}(t) \tilde{m}_{\tilde{w}}^*(t) \} = R_{\tilde{m}_{\tilde{w}}} = \sigma^2_{\tilde{w}} I \text{ or } \sigma^2_{\tilde{w}} \delta(t, t')
\]

and \( \{ \phi_w \} \) and \( \{ \tilde{M}_{w,y} \} \) are independent.

The problem will be to design an operator \( M \{ \cdot \} \) that is linear so that
\[
E_{w,\tilde{w}} \{ \| \phi_w - M(\mathbf{R}_{w,\tilde{w}}) \|_2^2 \} \text{ is minimized.}
\]
We shall discuss in detail three particular cases:

1) \( \phi \) is a signal and \( \Phi_{\text{DATA}} \) is simply the distorted noiseless version of \( \Phi_\{\phi\} \). This problem is called INVERSE FILTERING or DECONVOLUTION and entails "undoing" as well as possible the known distortion.

2) \( \phi \) is a signal and we know that it is selected from a class of signals for which some operator \( A \) yields a very "short" vector/signal with small energy. \( M \) is a noise with given second order statistics and \( M_1 \) is a known operator. This problem is called the CONSTRAINED DECONVOLUTION problem.
3) $\xi_\omega$ is a realization of a random process with known second order statistic, $\nu_\omega$ is a realization of a noise process and $\mathcal{H}_1$ is given. This is the full problem requiring consideration of the stochastic properties of both $\xi_\omega$ and $\nu_\omega$, and it is called the optimal least squares estimation problem or the Wiener filtering problem.

In each of these cases we shall design the optimal linear operator that solves the problem. We note however that the solutions are not the best possible, only the best linear operators that recover the distorted and/or noisy signal with least mean squared error.
12.2 INVERSE FILTERING and the Pseudo-Inverse

Consider that a given signal \( p \) is distorted by a linear operator \( \mathcal{H} \) to yield \( \tilde{p} = \mathcal{H} \{ p \} \).

From \( \tilde{p} \) we want to recover \( p \). Clearly the problem would be trivial if we could find an operator \( \mathcal{H}^{-1} \) so that
\[
\mathcal{H}^{-1} \{ \mathcal{H} \{ p \} \} = p
\]
i.e. if
\[
\mathcal{H}^{-1} \{ \mathcal{H} \{ \cdot \} \} = \text{Identity} \{ \cdot \}.
\]
However distortions are rarely invertible.
Let us first consider the continuous case \( \phi(t) : [0, 1] \rightarrow [\phi_L, \phi_U] \) is distorted by a linear operator described by \( h(t, \tau) \) as follows:

\[
\phi^{\text{DATA}}(t) = \int_0^t \phi(\xi) h(t, \xi) d\xi
\]

Let us assume that \( h(t, \xi) \) is \( h_{st}(t-\xi) \), i.e. that \( H \) is a LSI operator. Then we have

\[
\phi^{\text{DATA}}(t) = \int_0^t \phi(\xi) h(t-\xi) d\xi
\]

where it is assumed that both \( \phi(\xi) \) and \( h(\cdot) \) extend beyond \([0, 1]\) by periodic extension. Then we have that
\[ \varphi(t) = \sum_{k=-\infty}^{+\infty} \langle \varphi(t) e^{i\pi k t} \rangle e^{i\pi k t} \]
\[ h(t) = \sum_{k=-\infty}^{+\infty} \langle h(t) e^{i\pi k t} \rangle e^{i\pi k t} \]

and therefore:

\[ \rho_{\text{DATA}}(+) = \sum_{k} \sum_{\ell} \rho_{k\ell} e^{i\pi k s} \delta_{k\ell} e^{i\pi (t-s)} \]
\[ = \sum_{k} \sum_{\ell} \rho_{k\ell} \int_{0}^{+\infty} e^{i\pi (k-\ell) s} \delta_{k\ell} e^{i\pi (t-s)} \]
\[ = \sum_{k} \sum_{\ell} \rho_{k\ell} \delta_{k\ell} e^{i\pi (t-s)} \]
\[ = \sum_{k} \rho_{k\ell} e^{i\pi k t} \]

Therefore we have for

\[ \rho_{\text{DATA}}(+) = \sum_{k=-\infty}^{+\infty} \rho_{k\ell} e^{i\pi k t} \]
and \( \rho_{k\ell} = \rho_{k\ell} \)
and by comparing the expansions we see that for the coefficients we have:

\[ \tilde{p}_k^{\text{DATA}} = \tilde{c}_k \cdot h_k \]

Therefore, given \( h_k \) (we know \( h \)) and looking at \( \tilde{p}_k^\text{DATA} \) we get that we could "in principle" recover \( \tilde{p}_k \) by a linear operator \( M \) that should be time-invariant and have \( m_k = \frac{1}{h_k} \).

This would yield

\[ M \{ \tilde{p}_k^\text{DATA} \} = \frac{1}{h_k} \tilde{p}_k^\text{DATA} = \frac{1}{h_k} \cdot h_k \cdot \tilde{c}_k = \tilde{c}_k \]

and would provide for us perfect recovery of \( \tilde{c}_k \). The only problem with this approach is that we could have
for some values of \( k \), \( |h_k| = 0 \).

In this case \( M_k \cdot \bar{z} = m(t) \)
cannot be
\[
m(t) = \sum_{k=e}^{\infty} \frac{1}{h_k} e^{iz \pi k t},
\]
for obvious reasons. What is to be done?

Well, we realize that in case some \( h_k \)'s are zero we have, from
\[
P_k^{\text{DATA}} = P_k \cdot h_k
\]
that \( P_k^{\text{DATA}} \) will be zero for all those \( k \)'s where \( h_k \) is zero. Hence the information
on those particular \( h_k \)'s of the signal
\( \phi(t) \), is irrecoverably lost! At those
places where \( h_k \neq 0 \) we can get
\[
P_k = \frac{P_k^{\text{DATA}}}{h_k}
\]
but at places where \( h_k \) is zero we have
no clue whatsoever about what the corresponding coefficient \( \hat{q}_k \) was! Therefore at the places where \( \hat{q}_k = 0 \) we may set \( \hat{q}_k = 0 \). This yields the definition of \( m_k \):

\[
m_k \triangleq \begin{cases} \frac{1}{\ell_k} & \text{if } \ell_k \neq 0 \\ 0 & \text{if } \ell_k = 0 \end{cases}
\]

and then we shall have

\[
\hat{q}_k = \begin{cases} q_k & \text{if } \ell_k \neq 0 \\ 0 & \text{if } \ell_k = 0 \end{cases}
\]

and this is the "best" we can do. What will the error be with this \( M \{ \cdot \} \)? Well we'll have

\[
\| \hat{q}(t) - q(t) \| ^2 = \frac{1}{1} \int_0^1 \left( \sum_k (\hat{q}_k - q_k) e^{i \omega_k t} \right)^2 \, dt
\]

\[
= \sum_k (\hat{q}_k - q_k)^2 \int_0^1 e^{i \omega_k t} e^{-i \omega_k t} \, dt = \sum_k (\hat{q}_k - q_k)^2
\]
Without having any further information on \( \chi(t) \), i.e. on \( \left\{ \chi_k \right\}_{k \in \mathbb{Z}} \), we cannot do better. Note that \( \hat{\chi}(t) \) has the energy
\[
\int_0^1 \left( \sum_k |\hat{\chi}_k e^{i\eta k t}|^2 \right) dt = \sum_k \hat{\chi}_k^2 + \sum_{k \mid \eta k \neq 0} \left| \hat{\chi}_k \right|^2
\]
any selecting any other values than zero for \( \chi_k \) at \( k \)'s for which \( \eta k = 0 \) will increase the energy ("size of") \( \hat{\chi}(t) \). So the solution we obtained solves the problem

\[
\begin{align*}
\text{Minimize} & \quad \int_0^1 \left( \hat{\chi}(t) \right)^2 dt \\
\text{subject to} & \quad \int_0^1 \chi(\tau) h(t-\tau) d\tau = \chi(t) \\
\end{align*}
\]
Of course selecting any set of values for \( \chi_k \) at \( \left\{ k \mid \eta k = 0 \right\} \) will yield \( \hat{\chi}(t) \) so that \( \chi \) \( \text{LSE} \) 

\( \chi \) \( \text{DATA} \)
12.4 The discrete inverse filtering problem.

Given \( \overline{\varphi} \) we have

\[
\overline{\varphi}_{\text{DATA}} = \mathbf{H} \overline{\varphi}
\]

and from \( \overline{\varphi}_{\text{DATA}} \) we want to recover \( \overline{\varphi} \).

The matrix \( \mathbf{H} \) is an \( N \times N \) matrix that might not be invertible. Of it is we clearly have that

\[
\overline{\varphi} = \mathbf{H}^{-1} \overline{\varphi}_{\text{DATA}}
\]

and we fully recover \( \overline{\varphi} \). What should we do when \( \mathbf{H} \) is not invertible. In this case we rely on the **Singular Value Decomposition** of the matrix \( \mathbf{H} \).
SINGULAR VALUE DECOMPOSITION

Any matrix $M$ has a singular value decomposition of the form:

$$M_{(m \times n)} = U \Sigma V^*$$

where $M_{m \times n}$ is the given (C or R) matrix of size $m \times n$, and $U$ is a unitary $m \times m$ matrix (i.e. $U^*U = UU^* = I_{m \times m}$), $V$ is a unitary $n \times n$ matrix (i.e. $V^*V = VV^* = I_{n \times n}$), and $\Sigma$ is a $m \times n$ matrix of the form

$$\Sigma_{m \times n} = \begin{bmatrix}
\sigma_1 & 0 & & \\
0 & \sigma_2 & & \\
& & \ddots & \\
0 & & & \sigma_p
\end{bmatrix}$$

and the real numbers $\sigma_1, \sigma_2, \ldots, \sigma_p$ are positive $\sigma_1 \geq \sigma_2 \geq \sigma_2 \ldots \geq \sigma_p$, with $p = \text{rank}(M_{m \times n})$. 
The numbers $\sigma_1, \sigma_2, \ldots, \sigma_p$ are called the singular values of $M$. The matrices $U$ and $V$ are the results of the spectral decompositions of $MM^*$ and $MM^*M$.

$MM^*$ is a nonnegative matrix of rank $p$ and it is symmetric, hence it can be written as follows:

$$MM^* = U \Lambda U^*$$

for some $U = [\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m]^T$, unitary matrix, and

$$\Lambda_{MM^*} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_p 0000]_{m \times m}$$

a diagonal $m \times m$ matrix with $\lambda_1 > \lambda_2 \ldots > \lambda_p > 0$.

The matrix $M^*M$ is also of rank $p$, symmetric and $M^*M$ has the same eigenvalues as $MM^*$. Indeed if:

$$MM^*x = \lambda x$$

then

$$M^*M(M^*x) = \lambda(M^*x)$$.
Therefore we can write
\[ M^* M = V \Lambda V^* \]
where \( V = [v_1, v_2, \ldots, v_n] \), a unitary matrix and
\[ \Lambda_{MM} = \text{diag} [\lambda_1, \lambda_2, \ldots, \lambda_p, 0, \ldots, 0]_{m \times n} \]
with \( \lambda_1 > \lambda_2 > \lambda_3 \ldots \geq \lambda_p > 0 \).

From \( M_{m \times n} = U \Sigma V^* \) we see that
\[
M M^* = U \Sigma V^* V \Sigma U^* = U \Sigma^2 U^* \\
M^* M = V \Sigma U^* U \Sigma V^* = V \Sigma^2 V^* \\
\]
hence we have \( \sigma_1^2 = \lambda_1, \sigma_2^2 = \lambda_2, \ldots, \sigma_p^2 = \lambda_p \).

We also see that the last \( m - p \) eigenvectors
in \( U_{m \times m} = [\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_p, \bar{u}_{p+1}, \ldots, \bar{u}_m] \) are an
arbitrary set of on vectors spanning the
nullspace of \( M M^* \) and the last \( n - p \) eigenvectors
in \( V_{m \times n} = [\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_p, \bar{v}_{p+1}, \ldots, \bar{v}_n] \) are an
arbitrary set of on vectors spanning the
nullspace of \( M^* M \).
The matrix $M$ can be written as follows:

$$M = \begin{bmatrix} \frac{1}{\sigma_1} & \frac{1}{\sigma_2} & \cdots & \frac{1}{\sigma_p} \\ \frac{1}{\sigma_1} & \frac{1}{\sigma_2} & \cdots & \frac{1}{\sigma_p} \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \cdots & u_p \\ \sigma_1 & \sigma_2 & \cdots & \sigma_p \end{bmatrix} \begin{bmatrix} -\frac{1}{\nu_1} & -\frac{1}{\nu_2} & \cdots & -\frac{1}{\nu_p} \\ -\frac{1}{\nu_1} & -\frac{1}{\nu_2} & \cdots & -\frac{1}{\nu_p} \end{bmatrix} =$$

$$= \sum_{k=1}^{p} \sigma_k u_k \bar{v}_k = \sum_{k=1}^{p} \frac{1}{\nu_k} u_k \bar{v}_k$$

Returning to the matrix $\mathcal{H}$, our linear operator acting on $\Phi$ we can write that

$$\mathcal{H}_{N\times N} = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \\ \lambda_1 & \lambda_2 & \cdots & \lambda_{100} \end{bmatrix} \begin{bmatrix} -\frac{1}{\nu_1} & -\frac{1}{\nu_2} & \cdots & -\frac{1}{\nu_N} \\ -\frac{1}{\nu_1} & -\frac{1}{\nu_2} & \cdots & -\frac{1}{\nu_N} \end{bmatrix} =$$

$$= \sum_{k=1}^{N} \frac{1}{\nu_k} u_k \bar{v}_k$$
The action of $\mathcal{H}$ on $\bar{\varphi}$ can be made explicit in terms of the "spectral decomposition of $\mathcal{H}$", i.e. we have

$$\bar{\varphi}^{\text{DATA}} = \mathcal{H} \bar{\varphi} =$$

$$= \mathcal{U} \Lambda^{1/2} \mathcal{V}^* \bar{\varphi} =$$

$$= \sum_{k=1}^{p} \lambda_k^{1/2} \bar{\varphi} \bar{v}_k \bar{v}_k^* \bar{\varphi} =$$

$$= \sum_{k=1}^{p} \lambda_k^{1/2} \langle \bar{\varphi} \bar{v}_k \bar{v}_k^* \bar{\varphi} \rangle \bar{u}_k = \bar{\varphi}^{\text{DATA}}$$

From $\bar{\varphi}^{\text{DATA}}$ we want to "recover" $\bar{\varphi}$. Clearly if $p < N$ the matrix $\mathcal{H}$ is not invertible. If $p = N$, we have an easy inversion

$$\bar{\varphi} = \mathcal{H}^{-1} \bar{\varphi}^{\text{DATA}} = (\mathcal{U} \Lambda^{1/2} \mathcal{V}^*)^{-1} \bar{\varphi}^{\text{DATA}} =$$

$$= \mathcal{V} \Lambda^{-1/2} \mathcal{U}^* \bar{\varphi}^{\text{DATA}} = \sum_{k=1}^{N} \lambda_k^{-1/2} \mathcal{V} \bar{\varphi} \bar{u}_k \bar{v}_k^* \bar{\varphi} \bar{u}_k =$$

$$= \sum_{k=1}^{N} \lambda_k^{-1/2} \bar{v}_k \langle \bar{\varphi}^{\text{DATA}} \bar{u}_k \rangle = \sum_{k=1}^{N} \lambda_k^{-1/2} \langle \bar{\varphi}^{\text{DATA}} \bar{u}_k \rangle \bar{v}_k$$
If $p < N$ we do not see immediately what to do. Let us therefore consider the modified $\mathcal{H}$,

$$\mathcal{H}^2(\varepsilon) = U \left[ \begin{array}{c|c} \chi^2_1 & \chi^2_0' \\ \hline \chi^2_0' & \varepsilon \\ \hline \chi^2_0 & \varepsilon \\ \hline \varepsilon & \varepsilon \end{array} \right] \sqrt{\varepsilon}.$$ 

With $\mathcal{H}^2(\varepsilon)$ we know what to do. We have

$$\overline{\xi}(\varepsilon) = \mathcal{H}^{-1} \overline{\phi} = \sum_{k=1}^{p} \frac{1}{\lambda^k} \langle \phi^\text{DATA}, u^*_k \rangle v^*_k + \sum_{k=p+1}^{N} \frac{1}{\varepsilon} \langle \phi^\text{DATA}, u^*_k \rangle v^*_k.$$

Note however that we have

$$\overline{\phi}^\text{DATA} = \mathcal{H^2} \overline{\phi} = \left[ \begin{array}{c|c|c} u_1 & u_2 & \ldots & u_N \end{array} \right] \left[ \begin{array}{c} \chi^2_1 \\ \hline \chi^2_0' \\ \hline \chi^2_0' \\ \hline \varepsilon \\ \hline \varepsilon \end{array} \right] \overline{\xi} = \sum_{k=1}^{p} \lambda^k \langle \phi, v^*_k \rangle u^*_k \in \text{span}\{u_1, u_2, \ldots, u_N\}.$$

\[\text{hence for every } k \in \{p+1, \ldots, N\}, \text{ we have } \langle \overline{\phi}^\text{DATA}, u^*_k \rangle = 0.\]
We therefore have that:

\[
\overline{\varphi}(\varepsilon) = \Pi(\varepsilon) \overline{\varphi}^{\text{DATA}} =
\]

\[
= \sum_{k=1}^{p} \frac{1}{\chi_k^{1/2}} \langle \overline{\varphi}^{\text{DATA}}, \overline{u}_k \rangle \overline{v}_k + \\
+ \sum_{k=p+1}^{m} \frac{1}{\varepsilon} \cdot 0 \cdot \overline{v}_k =
\]

\[
= \sum_{k=1}^{p} \frac{1}{\chi_k^{1/2}} \langle \overline{\varphi}^{\text{DATA}}, \overline{u}_k \rangle \overline{v}_k = \text{independent of } \varepsilon ! ! !
\]

Therefore for any \( \varepsilon \) we choose

\[
\overline{\varphi}(\varepsilon) = \overline{\varphi} = \Pi(\varepsilon) \overline{\varphi}^{\text{DATA}} \]

will be the same! Therefore we simply select:

\[
\Pi^{-1} \triangleq V \begin{bmatrix}
\chi_1^{1/2} & \chi_2^{1/2} & \ldots & \chi_p^{1/2} \\
\chi_1^{-1/2} & \chi_2^{-1/2} & \ldots & \chi_p^{-1/2} \\
\cdot & \cdot & \ldots & \cdot \\
\chi_1^{-1/2} & \chi_2^{-1/2} & \ldots & \chi_p^{-1/2} 
\end{bmatrix} U^*
\]

to be the "inverse" operator to be applied to \( \overline{\varphi}^{\text{DATA}} \) to "recover" an estimate of \( \overline{\varphi} \), i.e. \( \overline{\varphi} \).
(This selection is as if we take $\varepsilon \to \infty$).

Note that we obtained
\[
\hat{\phi} = \mathbb{H}^{-1} \mathcal{P} \mathbb{H} = V \left[ \begin{array}{cccc}
\lambda_{p_1}^{1/2} & & \\
& \ddots & \\
& & \lambda_{p_M}^{1/2}
\end{array} \right] \mathbb{H} \bar{\phi}_{\text{DATA}}
\]
\[
= \sum_{k=1}^{\text{span} \{V_1, V_2, \ldots, V_p\}} \frac{1}{\lambda_{p_k}^{1/2}} \langle \bar{\phi}_{\text{DATA}}, \vec{u}_{p_k} \rangle \vec{v}_k
\]

What is the (mean) squared error in recovering $\hat{\phi}$ instead of $\bar{\phi}$? Well, we know that the vector $\bar{\phi}$ can be represented in the basis $[\vec{V}_1, \vec{V}_2, \ldots, \vec{V}_N] = V$, i.e.
\[
\bar{\phi} = V V^* \bar{\phi} = [V_1 \ V_2 \ \ldots \ V_N] \begin{bmatrix}
\langle \bar{\phi}_{\text{DATA}}, \vec{v}_1 \rangle \\
\langle \bar{\phi}_{\text{DATA}}, \vec{v}_2 \rangle \\
\vdots \\
\langle \bar{\phi}_{\text{DATA}}, \vec{v}_N \rangle
\end{bmatrix}
\]

If we look at this representation of $\bar{\phi}$ and consider what $\mathbb{H} \bar{\phi}$ is, we see the following:
\[ \overline{\phi^{DATA}} = \overline{H} \overline{\phi} = \overline{H} \overline{V} \cdot \overline{V} \overline{\phi} = \]

\[ = \overline{U} \left[ \overline{\lambda_1} \overline{\lambda_2} \ldots \overline{\lambda_{p+\ldots+0}} \right] \overline{V} \overline{V} \overline{\phi} = \]

\[ = \overline{U} \left[ \overline{\lambda_1} \overline{\lambda_2} \ldots \overline{\lambda_{p+\ldots+0}} \right] \left[ \begin{array}{c} <\overline{\phi}, \overline{v}_1> \\ <\overline{\phi}, \overline{v}_2> \\ \vdots \\ <\overline{\phi}, \overline{v}_N> \end{array} \right] = \]

\[ = \sum_{k=1}^{p} \lambda_k <\overline{\phi}, \overline{v}_k> \overline{u}_k \]

Therefore, if we write

\[ \overline{\phi} = \sum_{k=1}^{p} \underbrace{<\overline{\phi}, \overline{v}_k> \overline{v}_k}_{\text{Span } \{v_1, v_2 \ldots v_p\}} + \sum_{k=p+1}^{N} \underbrace{<\overline{\phi}, \overline{v}_k> \overline{v}_k}_{\text{Span } \{v_{p+1}, v_{p+2} \ldots v_N\}} \]

we see that \( \overline{\phi} \) is simply the part of \( \overline{\phi} \) which is in the subspace spanned by \( \{v_1, v_2, \ldots v_p\} \). Indeed writing

\[ \overline{\phi} = \sum_{k=1}^{p} \frac{1}{\lambda_k} <\overline{\phi^{DATA}}, \overline{u}_k> \overline{v}_k \]

and
expanding, we get:

\[ \hat{\phi} = \sum_{k=1}^{p} \frac{1}{\lambda_k^{1/2}} \left( \sum_{e=1}^{p} \lambda_e^{1/2} \langle \bar{\varphi}, \bar{v}_e \rangle \bar{u}_e, \bar{u}_k \right) V_k = \]

\[ = \sum_{k=1}^{p} \frac{1}{\lambda_k^{1/2}} \left( \sum_{e=1}^{p} \lambda_e^{1/2} \langle \bar{\varphi}, \bar{v}_e \rangle \langle \bar{u}_e, \bar{u}_k \rangle \right) V_k = \]

\[ = \sum_{k=1}^{p} \frac{1}{\lambda_k^{1/2}} \lambda_k^{1/2} \langle \bar{\varphi}, \bar{v}_k \rangle \cdot 1 \cdot V_k = \]

\[ = \sum_{k=1}^{p} \langle \bar{\varphi}, \bar{v}_k \rangle V_k \text{ as claimed.} \]

(This can easily be seen also from)

\[ \hat{\phi} = H^* \hat{\phi}^{\text{DATA}} = V \left[ \begin{array}{c} \lambda_k^{1/2} \lambda_e^{1/2} \vdots \lambda_{p-1}^{1/2} \end{array} \right] u^*_k \right] v^*_p = \]

\[ = V \left[ \begin{array}{c} \vdots \vdots \vdots \vdots \vdots \vdots \end{array} \right] v^*_p \]
Concluding, we see that the pseudo inverse recovers the part of \( \mathcal{P} \) which is in the span of \( V_1, V_2, \ldots, V_p \). The part which is in the span of \( V_{p+1}, \ldots, V_N \) cannot be recovered since any vector in this subspace is mapped to zero by \( \mathcal{H} \). The span of \( \{ V_{p+1}, \ldots, V_N \} \) is the null space of \( \mathcal{H} \), or the kernel of \( \mathcal{H} \), defined as follows:

\[
\text{Null Space } \mathcal{H} = \{ \bar{x} \mid \mathcal{H}\bar{x} = \bar{0} \}
\]

Hence for a given signal \( \mathcal{P} \) the pseudo-inverse of \( \mathcal{H} \) recovers for us \( \hat{\mathcal{P}} \), the part of \( \mathcal{P} \) which is not in the null space of \( \mathcal{H} \).
The error in recovering $\bar{\phi}$ is therefore

$$\|\bar{\phi} - \bar{\phi}\| = \|\sum_{k=p+1}^{N} \langle \bar{\phi}, V_k \rangle V_k \| =$$

$$= \sum_{k=p+1}^{N} \langle \bar{\phi}, V_k \rangle^2$$

and

$$\bar{\phi} = \sum_{k=1}^{p} \langle \bar{\phi}, V_k \rangle V_k$$

We see that we have

$$\overline{\phi}_{\text{DATA}} = \mathcal{H} \bar{\phi} = \mathcal{H} \bar{\phi}$$

and in fact we could add to $\bar{\phi}$ any vector in the null-space of $\mathcal{H}$ and have the same $\overline{\phi}_{\text{DATA}}$. Indeed:

$$\mathcal{H} \left( \bar{\phi} + \sum_{k=p+1}^{N} \alpha_k V_k \right) = \mathcal{H} \bar{\phi} = \mathcal{H} \bar{\phi}$$

Note that:

$$\| \bar{\phi} + \sum_{k=p+1}^{N} \alpha_k V_k \|^2 = \| \bar{\phi} \|^2 + \sum_{k=p+1}^{N} \alpha_k^2$$

therefore $\bar{\phi}$ is seen as the shortest
vector obeying $\hat{\rho}^T H = \bar{\rho}_{DATA}$.

Therefore we see that the pseudo-inverse solves for us the following optimization problem:

\[
\begin{align*}
\text{Minimize} & \quad \|\bar{\rho}\|^2 = \bar{\rho}^* \bar{\rho} \\
\text{subject to} & \quad H \hat{\rho} = \bar{\rho}_{DATA}
\end{align*}
\]

This interpretation of the solution we obtained is of crucial importance since, as we shall see, many signal restoration processes are posed as some natural optimization problems!
12.5 DISCRETE INVERSE FILTERING (THE CIRCULANT CASE)

Suppose that we have $H$-a circulant matrix, i.e. $X^{(N)}$ is a linear shift-invariant operator (LSI). Then we know that $H$ has a (diagonalized) representation in terms of the DFT matrix as follows:

$$H_{LSI} = [DFT]^* \begin{bmatrix} \lambda_0^* & \lambda_1^* & 0 \\ 0 & \lambda_2^* & \ddots \\ \vdots & \ddots & \ddots \\ 0 & \cdots & \lambda_N^* \end{bmatrix} [DFT]$$

where

$$\begin{bmatrix} \lambda_0^* \\ \lambda_1^* \\ \vdots \\ \lambda_N^* \end{bmatrix} = [DFT] \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_N \end{bmatrix}$$

(Note that this is not the same as the SVD of $H_{LSI}$ which is based on $H_{LSI}^*H_{LSI} = [DFT]^* \begin{bmatrix} \lambda & 0 \\ 0 & \text{Identity} \end{bmatrix} [DFT]$)

and

$$H_{LSI}^*H_{LSI} = [DFT]^* \begin{bmatrix} \lambda & 0 \\ 0 & \text{Identity} \end{bmatrix} [DFT] [DFT]^* \begin{bmatrix} \lambda & 0 \\ 0 & \text{Identity} \end{bmatrix} [DFT]= [DFT]^* \lambda \lambda^* [DFT]$$
and the matrix $\Lambda^H \Lambda^* = \Lambda^H \Lambda = \begin{bmatrix} \lambda_0^* \lambda_0 & \lambda_1^* \lambda_1 & 0 \\ \lambda_1^* \lambda_0 & \lambda_1^2 & 0 \\ 0 & 0 & \lambda_{n-1}^* \lambda_{n-1}^2 \end{bmatrix}$

contains only positive values!

Here we can write:

$$
\mathbf{H}_{\text{LSI}} = \begin{bmatrix} DFT^* \end{bmatrix} \mathbf{P} \mathbf{H} \begin{bmatrix} \lambda_0^* & 0 & 0 \\ \lambda_1^* & \lambda_1^2 & 0 \\ 0 & 0 & \lambda_{n-1}^* \lambda_{n-1}^2 \end{bmatrix} \begin{bmatrix} DFT \end{bmatrix}
$$

where

$$
\mathbf{P} \mathbf{H} = \begin{bmatrix} e^{i\alpha_0} \\ e^{i\alpha_1} \\ e^{i\alpha_{n-1}} \end{bmatrix}
$$

where $\lambda_k^* = \lambda_k e^{i\alpha_k}$

or

$$
\mathbf{H}_{\text{LSI}} = \begin{bmatrix} DFT^* \end{bmatrix} \begin{bmatrix} \lambda_0^* & \lambda_1^* & 0 \\ 0 & \lambda_1^2 & \lambda_{n-1}^* \lambda_{n-1}^2 \\ 0 & 0 & \lambda_{n-1}^* \lambda_{n-1}^2 \end{bmatrix} \begin{bmatrix} DFT \end{bmatrix}
$$

are the possible SVD's of $\mathbf{H}_{\text{LSI}}$.

Clearly

$$
(DFT^* \mathbf{P} \mathbf{H})(DFT \mathbf{P} \mathbf{H})^* = [DFT]^* \mathbf{P} \mathbf{H} \mathbf{P}^* \mathbf{H}^* DFT^* = I
$$

and also

$$
(\mathbf{P} \mathbf{H} DFT)(\mathbf{P} \mathbf{H} DFT)^* = \mathbf{P} \mathbf{H} DFT DFT^* \mathbf{P}^* = I
$$
(Note that circulants commute hence we have that $HH^* = H^*H$ (such matrices are called normal matrices, but this is not important for us right here!) and the SVD is always unique and the matrices $U$ and $V$ in

$$M_{NxN} = U \Lambda^2 V^*$$

can always be modified to $U \cdot \text{Ph}(\alpha_1, \alpha_2, \ldots, \alpha_N)$ and $V \cdot \text{Ph}(\alpha_1, \alpha_2, \ldots, \alpha_N)$ to get the same result

$$M_{NxN} = U \frac{\text{Ph} \Lambda^2 \text{Ph}^* V^*}{\Lambda^{1/2}} \equiv \Lambda^{1/2}$$

Returning to the basic problem we are considering we have

$$\overline{\phi_{\text{DATA}}} = HH \overline{\phi} = [\text{DFT}^* \Lambda^2 \text{DFT}] \overline{\phi}$$

This yields

$$[\text{DFT}] \overline{\phi_{\text{DATA}}} = \Lambda^2 [\text{DFT}] \overline{\phi}$$
Therefore in the "transform domain" we have that in order to recover $\tilde{\phi}$ we need to invert a diagonal matrix with elements on the diagonal given by $\lambda_0^*, \lambda_1^*, \ldots, \lambda_{N-1}^*$, the DFT applied to the vector $[h_0, h_1, \ldots, h_{N-1}]^*$, the (discrete) impulse response of $\mathcal{L}_{1st}$.

The problem is that we might have some $k$'s where $|\lambda_k^*| = 0$, hence the (deconvolution) inversion process may be impossible to carry out. However, the places where $|\lambda_k^*| = 0$ are the places where $[\text{DFT}] \tilde{\phi}^{\text{DATA}}$ will also be zero. Denoting by

$$
\begin{bmatrix}
\tilde{\phi}^{\text{DATA}} \\
\tilde{\phi}^{\text{DATA}} \\
\vdots \\
\tilde{\phi}^{\text{DATA}}
\end{bmatrix}
= [\text{DFT}] \begin{bmatrix}
\tilde{\phi}^{\text{DATA}} \\
\tilde{\phi}^{\text{DATA}} \\
\vdots \\
\tilde{\phi}^{\text{DATA}}
\end{bmatrix}
$$
and by \[
\begin{bmatrix}
\tilde{\phi}_0 \\
\vdots \\
\tilde{\phi}_{N-1}
\end{bmatrix} = [\text{DFT}] \tilde{\phi}
\] we have

in the transform domain

\[
\tilde{\phi}_{\text{DATA}}^k = \lambda_k^+ \tilde{\phi}_k^k
\]

If \(|\lambda_k^+| = 0\) we know that \(\tilde{\phi}_{\text{DATA}}^k = 0\) too. Otherwise we have for \(k\) such that \(|\lambda_k^+| \neq 0\)

\[
\tilde{\phi}_k^k = \frac{1}{\lambda_k^+} \tilde{\phi}_{\text{DATA}}^k
\]

\[
\tilde{\phi}_k^k = \frac{1}{|\lambda_k^+|} e^{i \text{phase}(\lambda_k^+)} \tilde{\phi}_{\text{DATA}}^k
\]

(Where we used \(\lambda_k^+ = |\lambda_k^+| e^{i \text{phase}(\lambda_k^+)}\)). Hence the result is

\[
\tilde{\phi}_k = \begin{cases} 
\frac{1}{|\lambda_k^+|} e^{i \text{phase}(\lambda_k^+)} \tilde{\phi}_{\text{DATA}}^k & \text{where } |\lambda_k^+| \neq 0 \\
0 & \text{where } |\lambda_k^+| = 0
\end{cases}
\]

This is the Inverse Filter Solution for DECONVOLUTION in the Fourier Domain.
Example:

**Sliding Window Smoothing Operator**

\[ \mathbf{C}_{DATA} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \end{bmatrix} \]

hence:

\[ \mathbf{C}_{DATA} = \begin{bmatrix} 100 & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{N-1} \end{bmatrix} \]

We have the following filter:

\[
\begin{bmatrix}
\lambda_0^H \\
\lambda_1^H \\
\vdots \\
\lambda_{N-1}^H
\end{bmatrix}
= \begin{bmatrix} \mathbf{DFT}^* \end{bmatrix} \begin{bmatrix}
1 \\
1 \\
\vdots \\
1
\end{bmatrix} = 
\begin{bmatrix}
W^0 + W^1 & + W^0(m-1) \\
W^1(0) + W^1(1) & + W^1(m-1) \\
W^2 & + W^2(m-1) \\
\vdots \\
W^{(N-1)(0)} + W^{(N-1)(1)} & + W^{(N-1)(m-1)}
\end{bmatrix}
\begin{bmatrix}
\frac{W^0}{W^0} \\
\frac{W^1}{W^1} \\
\vdots \\
\frac{W^{(N-1)}}{W^{(N-1)}}
\end{bmatrix}
\]

\[
W = e^{i \frac{2\pi}{N}} \Rightarrow e^{i \frac{2\pi}{N} (km)} = 1 \text{ if } (km) \text{ is integer } N
\]
If for example $N = 2^n$ and $m = 2$ we have that $e^{i \frac{2\pi}{2^n} \cdot k \cdot 2} = 1$ if $2k = 2^n \implies k = 2^{m-1}$.
Therefore $\lambda_{2^{m-1}} = \lambda_{m/2} = 0$ and the matrix $\mathbf{H}_{1st}$ will not be invertible.

In general if $k = \left( \frac{\text{integer}}{m} \right) N$ we'll have $\lambda_k = 0$.

In all these cases we'll have to apply the "pseudo-inverse" filtering process.