Matrix and Fourier transform formulations for Wiener Filters

(c) 2007 Andy Adler, Carleton University, adler@sce.carleton.ca

Notation is used as follows. Note that since, multiplication of Fourier transforms (eg. $Z = XY$) is element-by-element, we can't directly interpret this as a matrix. Instead, we can interpret X, Y, and Z as column vectors, and write this multiplication using matrix multiplication notation as $Z = \mathrm{diag}(X)Y$.

- *vectors:* bold lowercase (eg. $\mathbf{x}$)
- *matrices:* bold uppercase (eg. $\mathbf{H}$)
- *Fourier transform:* $F\{\}$ and $F^{-1}\{\}$
- *Fourier transforms:* uppercase non-bold (eg. $X = F\{\mathbf{x}\}$)
- *convolution:* $\mathbf{x} \star \mathbf{y} = F^{-1}\{XY\}$
- *conjugate transpose:* $\mathbf{x}^*$; for Fourier transforms, interpret as conjugate only.

The term "Fourier transform" is not, strictly speaking, correct, since the signals are discrete for a finite time. We thus use "Fourier transform" and "Fourier series" interchangably.

We begin with a basic linear model in which measurements ($\mathbf{y}$) are made from an ideal image ($\mathbf{x}$) via a degradation process ($\mathbf{H}$) and independant noise, $\mathbf{n}$.

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

The linear restoration process calculates an estimate of the ideal image ($\hat{\mathbf{x}}$) from $\mathbf{y}$ via the restoration process

$$\hat{\mathbf{x}} = \mathbf{L}\mathbf{y} \tag{2}$$

The Wiener filter calculates the optimal linear restoration filter ($\mathbf{L}$) to minimize the error, ϵ:

$$\epsilon^2 = \mathrm{E}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right] \tag{3}$$

Matrix Formulation

Thus, we calculate

$$\begin{aligned}
\epsilon^2 &= \mathrm{E}\left[\|\mathbf{x}-\hat{\mathbf{x}}\|^2\right] \\
\epsilon^2 &= \mathrm{E}\left[\mathrm{trace}(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^*\right] \\
\epsilon^2 &= \mathrm{trace}\,\mathrm{E}\left[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^*\right] \\
\epsilon^2 &= \mathrm{trace}\,\mathrm{E}\left[\mathbf{x}\mathbf{x}^* - \mathbf{x}\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\mathbf{x}^* + \hat{\mathbf{x}}\hat{\mathbf{x}}^*\right] \\
\epsilon^2 &= \mathrm{trace}\left(\mathrm{E}\left[\mathbf{x}\mathbf{x}^*\right] - \mathrm{E}\left[\mathbf{x}\hat{\mathbf{x}}^*\right] - \mathrm{E}\left[\hat{\mathbf{x}}\mathbf{x}^*\right] + \mathrm{E}\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^*\right]\right)
\end{aligned}$$

$\mathrm{E}[\mathbf{x}\mathbf{x}^*] = \boldsymbol{\Sigma}_x$ is the *a priori* covariance of image pixels. It is called *a priori* because it describes images $\mathbf{x}$ in general, not the specific image of this problem. Conceptually, $\boldsymbol{\Sigma}_x$ is calculated from sampling and averaging the effect a large number of possible images *before* we arrive at the current problem. Along the diagonal, $\boldsymbol{\Sigma}_x$ it represents the power in each image pixel; the off diagonal elements represent the covariance between image pixels.

$\mathrm{E}[\hat{\mathbf{x}}\mathbf{x}^*]$ is calculated as follows

$$\begin{aligned}
\mathrm{E}[\hat{\mathbf{x}}\mathbf{x}^*] &= \mathrm{E}\left[(\mathbf{L}\mathbf{y})\mathbf{x}^*\right] \\
&= \mathrm{E}\left[\mathbf{L}(\mathbf{H}\mathbf{x}+\mathbf{n})\mathbf{x}^*\right] \\
&= \mathrm{E}\left[(\mathbf{L}\mathbf{H}\mathbf{x}+\mathbf{L}\mathbf{n})\mathbf{x}^*\right] \\
&= \mathrm{E}\left[\mathbf{L}\mathbf{H}\mathbf{x}\mathbf{x}^*\right] + \mathrm{E}\left[\mathbf{L}\mathbf{n}\mathbf{x}^*\right] \\
&= \mathbf{L}\mathbf{H}\mathrm{E}\left[\mathbf{x}\mathbf{x}^*\right] + \mathbf{L}\mathrm{E}\left[\mathbf{n}\mathbf{x}^*\right] \\
&= \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x
\end{aligned}$$

$\mathrm{E}[\mathbf{n}\mathbf{x}^*] = 0$ because the noise is statistically independant from the image (by our assumption).

$\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*]$ is calculated as follows

$$\begin{aligned}
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathrm{E}\left[(\mathbf{L}\mathbf{y})(\mathbf{L}\mathbf{y})^*\right] \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathrm{E}\left[\mathbf{L}\mathbf{y}\mathbf{y}^*\mathbf{L}^*\right] \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathbf{L}\mathrm{E}\left[\mathbf{y}\mathbf{y}^*\right]\mathbf{L}^* \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathbf{L}\mathrm{E}\left[(\mathbf{H}\mathbf{x}+\mathbf{n})(\mathbf{H}\mathbf{x}+\mathbf{n})^*\right]\mathbf{L}^* \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathbf{L}\mathrm{E}\left[(\mathbf{H}\mathbf{x}+\mathbf{n})(\mathbf{x}^*\mathbf{H}^*\mathbf{x}+\mathbf{n}^*)\right]\mathbf{L}^* \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathbf{L}\mathrm{E}\left[\mathbf{H}\mathbf{x}\mathbf{x}^*\mathbf{H}^* + \mathbf{H}\mathbf{x}\mathbf{n}^* + \mathbf{n}\mathbf{x}^*\mathbf{H}^* + \mathbf{n}\mathbf{n}^*\right]\mathbf{L}^* \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] &= \mathbf{L}\mathbf{H}\mathrm{E}\left[\mathbf{x}\mathbf{x}^*\right]\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\mathbf{H}\mathrm{E}\left[\mathbf{x}\mathbf{n}^*\right]\mathbf{L}^*
\end{aligned}$$

$$
\begin{aligned}
& + \quad \mathbf{L}\mathrm{E}\left[\mathbf{n}\mathbf{x}^*\right]\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\mathrm{E}\left[\mathbf{n}\mathbf{n}^*\right]\mathbf{L}^* \\
\mathrm{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^*] & = \quad \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\boldsymbol{\Sigma}_n\mathbf{L}^*
\end{aligned}
$$

Using statistically independant noise, $\mathrm{E}[\mathbf{n}\mathbf{x}^*] = \mathrm{E}[\mathbf{x}\mathbf{n}^*] = 0$. The noise covariance is $\mathrm{E}[\mathbf{n}\mathbf{n}^*] = \boldsymbol{\Sigma}_n$. It is *a priori* because it describes noise $\mathbf{n}$ in general, not the specific noise of this problem. Conceptually, $\boldsymbol{\Sigma}_n$ is calculated from sampling and averaging the effect a large number of noise measurements *before* we arrive at the current problem. Along the diagonal, $\boldsymbol{\Sigma}_n$ it represents the power in each noise channel; the off diagonal elements represent the covariance between noise channels; these will normally be close to zero in most measurement instruments problems.

Thus, we can write an expression for ϵ,

$$
\begin{aligned}
\epsilon^2 &= \mathrm{trace}\left(\mathrm{E}\left[\mathbf{x}\mathbf{x}^*\right] - \mathrm{E}\left[\mathbf{x}\hat{\mathbf{x}}^*\right] - \mathrm{E}\left[\hat{\mathbf{x}}\mathbf{x}^*\right] + \mathrm{E}\left[\hat{\mathbf{x}}\hat{\mathbf{x}}^*\right]\right) \\
&= \mathrm{trace}\left(\boldsymbol{\Sigma}_x - \left(\mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x\right)^* - \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x + \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\boldsymbol{\Sigma}_n\mathbf{L}^*\right) \\
&= \mathrm{trace}\left(\boldsymbol{\Sigma}_x - \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x + \mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\boldsymbol{\Sigma}_n\mathbf{L}^*\right) \\
&= \mathrm{trace}\left(\boldsymbol{\Sigma}_x - 2\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{L}^* + \mathbf{L}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)\mathbf{L}^*\right)
\end{aligned}
$$

Since the trace $\mathbf{A} = \mathrm{trace}\,\mathbf{A}^*$, and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^*$. We choose $\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{L}^*$, rather than $\mathbf{L}\mathbf{H}\boldsymbol{\Sigma}_x$, because the matrix sizes are compatible in the next step when we take the derivative.

We minimize ϵ^2 by setting its derivative to zero

$$
\begin{aligned}
0 &= \frac{\partial\epsilon^2}{\partial\mathbf{L}} \\
0 &= \mathrm{trace}\left(-2\boldsymbol{\Sigma}_x\mathbf{H}^* + 2\mathbf{L}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)\right)
\end{aligned}
$$

The element in the trace() will be zero if $\boldsymbol{\Sigma}_x\mathbf{H}^* = \mathbf{L}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)$. Thus, we calculate:

$$
\begin{aligned}
\mathbf{L}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right) &= \boldsymbol{\Sigma}_x\mathbf{H}^* \\
\mathbf{L}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1} &= \boldsymbol{\Sigma}_x\mathbf{H}^*\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1} \\
\mathbf{L} &= \boldsymbol{\Sigma}_x\mathbf{H}^*\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}
\end{aligned}
$$

This is the matrix formulation of the Wiener filter.

$$
\mathbf{L} = \boldsymbol{\Sigma}_x\mathbf{H}^*\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1} \tag{4}
$$

$\mathbf{L}$ is the optimal linear filter, or filter which minimizes the expected error $\mathrm{E}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right]$. Note that much modern work on signal reconstruction seeks to work either on non-linear systems, or to minimize different representations of the expected error, such as the absolute error $\mathrm{E}\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_1\right]$.

Fourier transform representation of the Wiener Filter

Expressed as a matrix, the Wiener filter applies to any linear system. For example, a camera system with the lens at an angle will have a spatial blur which differs at each point. This can be represented by the degradation matrix $\mathbf{H}$. However, an important class of linear systems are time, or space invariant, called LTI (linear time invariant) or LSI (lienar space invariant).

For these systems, the degradation process $\mathbf{H}$ can be represented as a convolution kernel $\mathbf{h}$, where

$$\mathbf{y} = \mathbf{h} \star \mathbf{x} \tag{5}$$

The degradation matrix $\mathbf{H}$ corresponding to this convolution may be represented (for a 1D signal $\mathbf{h}$ with three elements $[h_{-1}, h_0, h_1]$) as:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & 0 & 0 & h_{-1} \\ h_{-1} & h_0 & h_1 & 0 & 0 \\ 0 & h_{-1} & h_0 & h_1 & 0 \\ 0 & 0 & h_{-1} & h_0 & h_1 \\ h_1 & 0 & 0 & h_{-1} & h_0 \end{bmatrix}$$

The edge effect is shown by "circulating" h_{-1} and h_1 around to the other side of the matrix. A matrix of this form is known as a circulant matrix, and has properties for the Fourier transform. Other edge effects are possible, the most common is to place zeros to give the "zero padding" edge effect.

The Fourier transform is a linear operation and can be represented as a matrix $\mathbf{F}$, such that

$$\begin{aligned} X &= F\{\mathbf{x}\} = \mathbf{F}\mathbf{x}, \text{ and} \\ \mathbf{x} &= F^{-1}\{\mathbf{x}\} = \mathbf{F}^{-1}\mathbf{x} \end{aligned}$$

Since the Fourier transform (Fourier series in this context) is an orthonormal basis, $\mathbf{F}^{-1} = \mathbf{F}^*$.

The Fourier transform of a filter can be calculated, we begin with $\mathbf{y} = \mathbf{h} \star \mathbf{x}$ which corresponds to a matrix representation $\mathbf{y} = \mathbf{H}\mathbf{x}$ and a Fourier representation $Y = HX$.

$$\begin{aligned} \mathbf{y} &= \mathbf{H}\mathbf{x} \\ Y &= F\{\mathbf{y}\} = \mathbf{F}\mathbf{y} \end{aligned}$$

$$
\begin{aligned}
Y &= \mathbf{FHx} \\
Y &= \mathbf{FH}(\mathbf{F}^*\mathbf{F})\mathbf{x} = \mathbf{F}(\mathbf{HF}^*)(\mathbf{Fx}) \\
Y &= (\mathbf{FHF}^*)X = HX
\end{aligned}
$$

Here, we use a "trick" of inserting the identity matrix (as $\mathbf{I} = \mathbf{F}^{-1}\mathbf{F}$).

Thus the Fourier transform representation of a linear operator represented as a Matrix $\mathbf{H}$ is $\mathbf{FHF}^*$. This applies to all linear operators represented as a square matrix (ie. the size of $\mathbf{x}$ and $\mathbf{y}$ are equal).

For circulant matrices $\mathbf{H}$, the Fourtier transform has an additional important property — it is diagonal: $\mathbf{FHF}^* = \mathrm{diag}(H)$. This means that the effect of the filter on each frequency component in the operator acts alone on each frequency component in the signal $\mathbf{x}$. This is what we expect of the Fourier transform – and it applies only to LTI (LSI) systems. For other systems, the appropriate transform is obtained from the singular value decomposition.

Based on this, we calculate the Fourier transform representation of the Wiener filter.

$$
\begin{aligned}
\hat{\mathbf{x}} &= \mathbf{Ly} \\
\hat{X} &= F\{\hat{\mathbf{x}}\} = \mathbf{F}\hat{\mathbf{x}} \\
&= \mathbf{FLy} = \mathbf{FL}(\mathbf{F}^{-1}Y) \\
&= \mathbf{F}\boldsymbol{\Sigma}_x\mathbf{H}^* \left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^*Y = LY
\end{aligned}
$$

Thus, $L = \mathbf{FLF}^*$, and

$$
\begin{aligned}
L &= \mathbf{F}\boldsymbol{\Sigma}_x\mathbf{H}^* \left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^* \\
&= \mathbf{F}\boldsymbol{\Sigma}_x(\mathbf{F}^*\mathbf{F})\mathbf{H}^*(\mathbf{F}^*\mathbf{F}) \left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^* \\
&= (\mathbf{F}\boldsymbol{\Sigma}_x\mathbf{F}^*)(\mathbf{FH}^*\mathbf{F}^*)\mathbf{F} \left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^* \\
&= S_x H^*\mathbf{F} \left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^*
\end{aligned}
$$

where $H^* = (\mathbf{FHF}^*)^* = (FB^*)^*\mathbf{H}^*(\mathbf{F})^* = \mathbf{FH}^*\mathbf{F}^*$, and $S_x = \mathbf{F}\boldsymbol{\Sigma}_x\mathbf{F}^* = F\{\boldsymbol{\Sigma}_x\}$. S_x thus represents the power spectrial density in the signal $\mathbf{x}$.

Using the identity $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\mathbf{B}^{-1}\mathbf{A}^{-1}$,

$$
\begin{aligned}
& \mathbf{F}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}\mathbf{F}^* \\
={}& (\mathbf{F}^*)^{-1}\left(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n\right)^{-1}(\mathbf{F})^{-1} \\
={}& ((\mathbf{F})(\mathbf{H}\boldsymbol{\Sigma}_x\mathbf{H}^* + \boldsymbol{\Sigma}_n)(\mathbf{F}^*))^{-1} \\
={}& (\mathbf{FH}\boldsymbol{\Sigma}_x\mathbf{H}^*\mathbf{F}^* + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^*)^{-1}
\end{aligned}
$$

$$
\begin{aligned}
&= \left(\mathbf{FH}(\mathbf{F}^*\mathbf{F})\boldsymbol{\Sigma}_x(\mathbf{F}^*\mathbf{F})\mathbf{H}^*\mathbf{F}^* + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^*\right)^{-1} \\
&= \left((\mathbf{FHF}^*)(\mathbf{F}\boldsymbol{\Sigma}_x\mathbf{F}^*)(\mathbf{FH}^*\mathbf{F}^*) + \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^*\right)^{-1} \\
&= \left(HS_xH^* + S_n\right)^{-1}
\end{aligned}
$$

where $S_n = \mathbf{F}\boldsymbol{\Sigma}_n\mathbf{F}^* = F\{\boldsymbol{\Sigma}_n\}$. S_n thus represents the power spectrial density in the noise $\mathbf{n}$.

Thus, the Wiener filter in the Fourier transform domain is $L = S_xH^*\left(HS_xH^* + S_n\right)^{-1}$. However, we are able to use the property that matrices are diagonal to exchange the order of multiplication.

$$
\begin{aligned}
L &= S_xH^*\left(HS_xH^* + S_n\right)^{-1} \\
L &= H^*(S_x^{-1})^{-1}\left(HH^*S_x + S_n\right)^{-1} \\
L &= H^*\left(S_x^{-1}S_xHH^* + S_x^{-1}S_n\right)^{-1} \\
L &= H^*\left(HH^* + S_nS_x^{-1}\right)^{-1}
\end{aligned}
$$

This is equivalent to the classic Wiener representation in the Fourier domain:

$$
L = \frac{H^*}{HH^* + \frac{S_n}{S_x}} \tag{6}
$$