3F3 - Detection, Estimation and Inference for Signal Processing

Orignal notes by S.J. Godsill
Edited by S.S. Singh

Signal Processing and Communications Laboratory,
Engineering Department,
Cambridge, UK

Overview of course I

Discrete-time random processes form the basis of modern communications systems, digital signal processing systems and many other application areas, including speech and audio modelling for coding/noise-reduction/recognition, radar and sonar, stochastic control systems, ...

This course extends the theory of Probability and Random processes to perform detection and estimation of signals and inference about their parameters.

The topics covered are

- Section 0 - Preliminaries, random processes (Already covered)
- Section 1 - Optimal Filtering Theory - Wiener Filters
- Section 2 - Optimal Detection - Matched Filters
- Section 3 - Estimation theory and Inference

A good textbook for sections 0-2 is:
Statistical Digital Signal Processing and Modeling - M.H. Hayes - Wiley. [This may be out of print, but there seem to be online versions freely available].
This covers all of the topics in a very understandable style, with may application examples and practice questions.
This second book is an excellent engineering introduction to more advanced ideas in probability and random processes:
An Introduction to Statistical Signal Processing - R.M. Gray and Davisson - CUP 2010 and at `http://ee.stanford.edu/ gray/sp.pdf`

Section 0: Preliminaries I

You will need the following fundamental concepts about discrete time random processes for this course:

- Autocorrelation Function
- Wide-sense stationarity
- Power spectrum
- Ergodicity

Change of notation

	Part 1	Part 2
PSD $\sum_{k=-\infty}^{\infty} R_X(k)e^{-j2\pi fk}$	$S_X(f)$	$\mathcal{S}_X(e^{j\omega})$
autocorrelation $\mathbb{E}(X_nX_m)$	$R_X(n,m)$	$r_{XX}[n,m]$
autocorrelation WSS $\mathbb{E}(X_nX_{n+k})$	$R_X(k)$	$r_{XX}[k]$
crosscorrelation $\mathbb{E}(X_nY_m)$		$r_{XY}[n,m]$

- The notation $\mathcal{S}_X(e^{j\omega})$ emphasises the fact that the PSD is similar to a z-transform: the z-transform evaluated at $z = e^{j\omega}$.
- In Part 1 we visualises the PSD as a function of frequency f whereas in Part 2 we visualises it as a function of angular frequency ω.
- Part 2 uses both capital and lower case letters for random variables, e.g. expect to see $E(x_ny_m)$ instead of $E(X_nY_m)$.

Correlation functions I

- The mean of a random process $\{X_n\}$ is defined as $\mathbb{E}[X_n]$ and the autocorrelation function as

$$\boxed{r_{XX}[n, m] = \mathbb{E}[X_n X_m]}$$

Autocorrelation function of random process

Correlation functions II

- The cross-correlation function between two processes $\{X_n\}$ and $\{Y_n\}$ is:

$$\boxed{r_{XY}[n,m] = \mathbb{E}[X_n Y_m]}$$

Cross-correlation function

Stationarity I

A stationary process has the same statistical characteristics *irrespective of shifts along the time axis*. To put it another way, an observer looking at the process from sampling time n_1 would not be able to tell the difference in the *statistical* characteristics of the process if he moved to a different time n_2. This idea is formalised by considering the Nth order density for the process:

$$\boxed{f_{X_{n_1}, X_{n_2}, \dots, X_{n_N}}(x_{n_1}, x_{n_2}, \dots, x_{n_N})}$$

Nth order density function for a discrete-time random process

Stationarity II

which is the joint probability density function for N arbitrarily chosen time indices $\{n_1, n_2, \ldots n_N\}$.
Since the probability distribution of a random vector contains all the statistical information about that random vector, we should expect the probability distribution to be unchanged if we shifted the time axis any amount to the left or the right, for a stationary signal. This is the idea behind *strict-sense stationarity* for a discrete random process.

Stationarity III

A random process is strict-sense stationary if, *for any finite* c, N *and* $\{n_1, n_2, \ldots n_N\}$:

$$\boxed{\begin{aligned} & f_{X_{n_1}, X_{n_2}, \ldots, X_{n_N}}(x_1, x_2, \ldots, x_N) \\ & \qquad = f_{X_{n_1+c}, X_{n_2+c}, \ldots, X_{n_N+c}}(x_1, x_2, \ldots, x_N) \end{aligned}}$$

Strict-sense stationarity for a random process

Strict-sense stationarity is hard to prove for most systems. In this course we will typically use a less stringent condition which is nevertheless very useful for practical analysis. This is known as *wide-sense stationarity*, which only requires first and second order moments (i.e. mean and autocorrelation function) to be invariant to time shifts.

Stationarity IV

A random process is *wide-sense stationary (WSS)* if:

1. $\mu_n = \mathbb{E}[X_n] = \mu$, (mean is constant)
2. $r_{XX}[n, m] = r_{XX}[m - n]$, (autocorrelation function depends only upon the difference between n and m).
3. The variance of the process is finite:

$$\mathbb{E}[(X_n - \mu)^2] < \infty$$

Stationarity V

Wide-sense stationarity for a random process

Note that strict-sense stationarity *plus finite variance (condition 3)* implies wide-sense stationarity, but not *vice versa*.

Example: the harmonic process I

The harmonic process is important in a number of applications, including radar, sonar, speech and audio modelling. An example of a real-valued harmonic process is the random phase sinusoid.

Here the signals we wish to describe are in the form of sine-waves with known amplitude a and frequency ω_0. The phase, however, is unknown and random, which could correspond to an unknown delay in a system, for example.

We can express this as a random process in the following form:

$$X_n = a \sin(n\omega_0 + \Phi)$$

Here a and ω_0 are fixed constants and Φ is a random variable having a uniform probability distribution over the range $-\pi$ to $+\pi$:

$$f(\phi) = \begin{cases} 1/(2\pi) & -\pi < \phi \leq +\pi \\ 0, & \text{otherwise} \end{cases}$$

Example: the harmonic process II

A selection of members of the ensemble is shown in the figure.

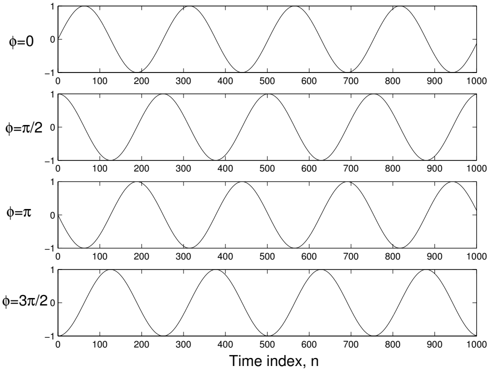


Figure 1 : A few members of the random phase sine ensemble. $a = 1$, $\omega_0 = 0.025$.

Example: random phase sine-wave I

Continuing with the same example, we can calculate the mean and autocorrelation functions and hence check for stationarity. The process was defined as:

$$X_n = a \sin(n\omega_0 + \Phi)$$

a and ω_0 are fixed constants and Φ is a random variable having a uniform probability distribution over the range $-\pi$ to $+\pi$:

$$f(\phi) = \begin{cases} 1/(2\pi) & -\pi < \phi \leq +\pi \\ 0, & \text{otherwise} \end{cases}$$

Example: random phase sine-wave II

1. **Mean:**

$$\begin{aligned}
\mathbb{E}[X_n] &= \mathbb{E}[a \sin(n\omega_0 + \Phi)] \\
&= a\,\mathbb{E}[\sin(n\omega_0 + \Phi)] \\
&= a\left\{\mathbb{E}[\sin(n\omega_0)\cos(\Phi) + \cos(n\omega_0)\sin(\Phi)]\right\} \\
&= a\left\{\sin(n\omega_0)\mathbb{E}[\cos(\Phi)] + \cos(n\omega_0)\mathbb{E}[\sin(\Phi)]\right\} \\
&= 0
\end{aligned}$$

since $\mathbb{E}[\cos(\Phi)] = \mathbb{E}[\sin(\Phi)] = 0$ under the assumed uniform pdf $f(\phi)$.

Example: random phase sine-wave III

2. **Autocorrelation:**

$$
\begin{aligned}
& r_{XX}[n, m] \\
& = \mathbb{E}[X_n X_m] \\
& = \mathbb{E}[a \sin(n\omega_0 + \Phi).a \sin(m\omega_0 + \Phi)] \\
& = 0.5a^2 \left\{\mathbb{E}\left[\cos[(n-m)\omega_0] - \cos[(n+m)\omega_0 + 2\Phi]\right]\right\} \\
& = 0.5a^2 \{\cos[(n-m)\omega_0] - \mathbb{E}\left[\cos[(n+m)\omega_0 + 2\Phi]\right]\} \\
& = 0.5a^2 \cos[(n-m)\omega_0]
\end{aligned}
$$

where the last line follows since $\mathbb{E}\left[\cos[(n+m)\omega_0 + 2\Phi\right] = 0$ - verify this yourself...
Hence the process satisfies the three criteria for wide sense stationarity (3. is easily verified from 2. with $n = m$).

A comment on notation and complex-valued processes I

Our course studies only real-valued random processes, as complex notation can clutter the presentation. However, in comms. especially, complex random processes are studied and definitions are modified in fairly obvious ways, e.g.

$$r_{XX}[k] = E[X_n X_{n+k}^*]$$

Concepts are basically unchanged but, for example, the autocorrelation function has Hermitian symmetry $(r_{XX}[k] = r_{XX}^*[-k])$ and the power spectrum does not have symmetry around zero frequency. For 3F3 though, we dispense with these fairly minor details and focus on real-valued processes.

Power spectra I

For a wide-sense stationary random process $\{X_n\}$, the power spectrum is defined as the discrete-time Fourier transform (DTFT) of the discrete autocorrelation function:

$$\boxed{\mathcal{S}_X(e^{j\omega}) = \sum_{m=-\infty}^{\infty} r_{XX}[m]\, e^{-jm\omega}} \tag{1}$$

Power spectrum for a random process

where $\omega = 2\pi f$.

Power spectra II

Note in this part of the course we explicitly write the PSD as $\mathcal{S}_X(e^{j\omega})$ which is the z-transform of the autocorrelation function evaluated as $z = e^{j\omega}$. When you want to visualise $\mathcal{S}_X(e^{j\omega})$, plot it as a function of ω.

The autocorrelation function can thus be found from the power spectrum by inverting the transform using the inverse DTFT:

$$\boxed{r_{XX}[m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{S}_X(e^{j\omega})\, e^{jm\omega}\, d\omega} \tag{2}$$

Autocorrelation function from power spectrum

Power spectra III

- The power spectrum is a real, positive, even and periodic function of frequency.
- The power spectrum can be interpreted as a density spectrum in the sense that the mean-squared signal value at the output of an ideal band-pass filter with lower and upper cut-off frequencies of ω_l and ω_u is given by

$$\frac{1}{\pi}\int_{\omega_l}^{\omega_u} \mathcal{S}_X(e^{j\omega})\, d\omega$$

Here we have assumed that the signal and the filter are real and hence we add together the powers at negative and positive frequencies.

Example: power spectrum I

The autocorrelation function for the random phase sine-wave was previously obtained as:

$$r_{XX}[m] = 0.5a^2 \cos[m\omega_0]$$

Example: power spectrum II

Hence the power spectrum is obtained as:

$$\begin{aligned}
\mathcal{S}_X(e^{j\omega}) &= \sum_{m=-\infty}^{\infty} r_{XX}[m]\, e^{-jm\omega} \\
&= \sum_{m=-\infty}^{\infty} 0.5a^2 \cos[m\omega_0]\, e^{-jm\omega} \\
&= 0.5a^2 \times \sum_{m=-\infty}^{\infty} (\exp(jm\omega_0) + \exp(-jm\omega_0))\, e^{-jm\omega} \\
&= \pi a^2 \times \sum_{m=-\infty}^{\infty} \delta(\omega - \omega_0 - 2m\pi) + \delta(\omega + \omega_0 - 2m\pi).
\end{aligned}$$

The last line above is obtained from the Fourier series of a periodic train of δ functions (check you can derive this yourself from the

Example: power spectrum III

complex Fourier series of a δ function, or look at 1B Signal and Data Analysis notes):

$$\sum_{m=-\infty}^{+\infty} \delta(t + t_0 - 2m\pi) = \frac{1}{2\pi} \sum_{m=-\infty}^{+\infty} \exp(-jmt_0) \exp(+jmt)$$

Alternatively (and equivalently) just take the inverse DTFT of the delta function to check the result:

$$\frac{1}{2\pi} \int_{-\pi}^{+\pi} \delta(\omega + \omega_0) e^{jm\omega} d\omega = \frac{1}{2\pi} e^{-jm\omega_0}$$

White noise I

White noise is defined in terms of its auto-covariance function. A wide sense stationary process is termed white noise if:

$$c_{XX}[m] = \mathbb{E}[(X_n - \mu)(X_{n+m} - \mu)] = \sigma_X^2 \delta[m]$$

where $\delta[m]$ is the discrete impulse function:

$$\delta[m] = \begin{cases} 1, & m = 0 \\ 0, & \text{otherwise} \end{cases}$$

$\sigma_X^2 = \mathbb{E}[(X_n - \mu)^2]$ is the variance of the process. If $\mu = 0$ then σ_X^2 is the *mean-squared* value of the process, which we will sometimes refer to as the 'power'.

White noise II

The power spectrum of zero mean white noise is:

$$
\begin{aligned}
\mathcal{S}_X(e^{j\omega}) &= \sum_{m=-\infty}^{\infty} r_{XX}[m]\, e^{-jm\omega} \\
&= \sigma_X^2
\end{aligned}
$$

i.e. flat across all frequencies.

Example: white Gaussian noise (WGN) I

There are many ways to generate white noise processes, all having the property

$$c_{XX}[m] = \mathbb{E}[(X_n - \mu)(X_{n+m} - \mu)] = \sigma_X^2 \delta[m]$$

The usual way is X_n are drawn *independently* from a Gaussian distribution with mean 0 and variance σ_X^2.
The N^{th} order pdf for the Gaussian white noise process is:

$$\begin{aligned} f_{X_{n_1}, X_{n_2}, \ldots, X_{n_N}}(x_1, x_2, \ldots, x_N) \\ = \prod_{i=1}^{N} \mathcal{N}(x_i | 0, \sigma_X^2) \end{aligned}$$

Example: white Gaussian noise (WGN) II

where

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

is the univariate normal pdf.

We can immediately see that the Gaussian white noise process is *Strict sense stationary*, since:

$$\begin{aligned} f_{X_{n_1}, X_{n_2}, \ldots, X_{n_N}}(x_1, x_2, \ldots, x_N) \\ = \prod_{i=1}^{N} \mathcal{N}(x_i|0, \sigma_X^2) \\ = f_{X_{n_1+c}, X_{n_2+c}, \ldots, X_{n_N+c}}(x_1, x_2, \ldots, x_N) \end{aligned}$$

Linear systems and random processes I

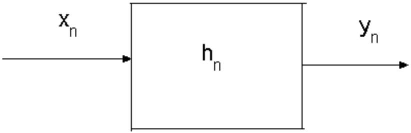


Figure 2 : Linear system

Linear systems and random processes II

When a wide-sense stationary discrete random process $\{X_n\}$ is passed through a *stable*, linear time invariant (LTI) system with digital impulse response $\{h_n\}$, the output process $\{Y_n\}$, i.e.

$$y_n = \sum_{k=-\infty}^{+\infty} h_k \, x_{n-k} = (x * h)(n)$$

is also wide-sense stationary.
The notation $(x * h)(n)$ refers to the convolution of the sequences $\{x_k\}$ and $\{h_k\}$, which results in a new sequence and this new sequence is evaluated at the time instance or index n.

Linear systems and random processes III

We can express the output correlation functions and power spectra in terms of the input statistics and the LTI system:

$$r_{XY}[k] = \mathbb{E}[X_n Y_{n+k}] = \sum_{l=-\infty}^{\infty} h_l \, r_{XX}[k-l] = (h * r_{XX})(k) \quad (3)$$

Cross-correlation function at the output of a LTI system

$$r_{YY}[l] = \mathbb{E}[Y_n Y_{n+l}] \sum_{k=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} h_k \, h_i \, r_{XX}[l+i-k] = (h * \tilde{h} * r_{XX})(l) \quad (4)$$

where $\tilde{h}$ is the time reversal of sequence h, that is $\tilde{h}_i = h_{-i}$. *Auto-correlation function at the output of a LTI system*

Linear systems and random processes IV

Note: these are convolutions. This is easily remembered through the following figure:

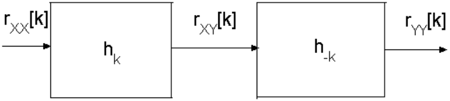


Figure 3 : Linear system - correlation functions

Power spectrum at output of LTI I

We can convert this result to the frequency domain result by taking DTFT of both sides of (4):

$$\boxed{\mathcal{S}_Y(e^{j\omega}) = |H(e^{j\omega})|^2 \mathcal{S}_X(e^{j\omega})} \tag{5}$$

Power spectrum at the output of a LTI system

Here $H(e^{j\omega T}) = \sum_{n=-\infty}^{+\infty} h_n \exp(-jn\omega)$ is the frequency response of the system.

Example: Filtering white noise I

Suppose we filter a zero mean white noise process $\{X_n\}$ with a first order *finite impulse response (FIR) filter*:

$$y_n = \sum_{m=0}^{1} b_m\, x_{n-m}, \qquad \text{or} \qquad Y(z) = (b_0 + b_1 z^{-1})X(z)$$

with $b_0 = 1$, $b_1 = 0.9$. This an example of a *moving average (MA)* process.
The impulse response of this causal filter is:

$$\{h_0, h_1, \ldots\} = \{b_0,\ b_1,\ 0,\ 0, \ldots\}$$

The autocorrelation function of $\{Y_n\}$ is obtained as:

$$r_{YY}[l] = \mathbb{E}[Y_n\, Y_{n+l}] = (h * \tilde{h} * r_{XX})[l] \tag{6}$$

Example: Filtering white noise II

This convolution can be performed directly. However, it is more straightforward in the frequency domain.
The frequency response of the filter is:

$$H(e^{j\omega}) = b_0 + b_1 e^{-j\omega}$$

The power spectrum of $\{X_n\}$ (white noise) is:

$$\mathcal{S}_X(e^{j\omega}) = \sigma_X^2$$

Hence the power spectrum of $\{Y_n\}$ is:

$$\begin{aligned}
\mathcal{S}_Y(e^{j\omega}) &= |H(e^{j\omega})|^2 \mathcal{S}_X(e^{j\omega}) \\
&= |b_0 + b_1 e^{-j\omega}|^2 \sigma_X^2 \\
&= (b_0 b_1 e^{+j\omega} + (b_0^2 + b_1^2) + b_0 b_1 e^{-j\omega}) \sigma_X^2
\end{aligned}$$

Example: Filtering white noise III

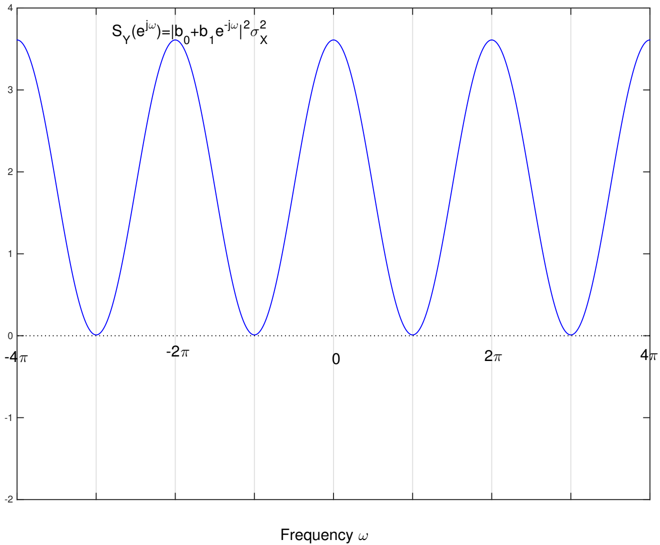
as shown in the figure overleaf. Comparing this expression with the DTFT of $r_{YY}[m]$:

$$\mathcal{S}_Y(e^{j\omega}) = \sum_{m=-\infty}^{\infty} r_{YY}[m]\, e^{-jm\omega}$$

we can identify non-zero terms in the summation only when $m = -1, 0, +1$, as follows:

$$r_{YY}[-1] = \sigma_X^2\, b_0\, b_1, \quad r_{YY}[0] = \sigma_X^2\,(b_0^2 + b_1^2)$$
$$r_{YY}[1] = \sigma_X^2\, b_0\, b_1$$

Example: Filtering white noise IV



Ergodic Random processes I

In practical signal processing systems we will often not know things like correlation functions or power spectra in advance. How could we estimate these for a wide-sense stationary process? If the process is also *ergodic* then easy estimation methods exist.

If we measure a realisation $\{x_n\}$ (i.e. one waveform from the ensemble of possible waveforms) of the process $\{X_n\}$ for some long (ideally infinite!) period of time, we can use this to estimate means and correlation functions, as follows:

- For an **Ergodic** random process we can estimate expectations by performing time-averaging on a single sample function, e.g.

$$\mu = \mathbb{E}[X_n] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x_n \quad \text{(Mean ergodic)}$$

$$r_{XX}[k] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n+k} \quad \text{(Correlation ergodic)} \qquad (7)$$

Ergodic Random processes II

- These formulae allow us to make the following estimates, for 'sufficiently' large N:

$$\mu = \mathbb{E}[X_n] \approx \frac{1}{N}\sum_{n=0}^{N-1} x_n \ \text{(Mean ergodic)}$$

$$r_{XX}[k] \approx \frac{1}{N}\sum_{n=0}^{N-1} x_n x_{n+k} \quad \text{(Correlation ergodic)} \tag{8}$$

  This is implemented with a simple computer code loop in discrete-time. Clearly the estimates get better as N gets larger.

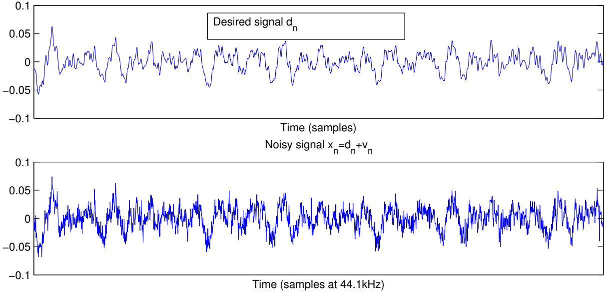
- Unless otherwise stated, we will always assume that the signals we encounter are both wide-sense stationary and ergodic. Beware though that neither of these will always be true in practice.

Section 1: Optimal Filtering and the Wiener Filter I

- Optimal filtering is an area in which we design filters that are optimally adapted to the statistical characteristics of a random process. As such the area can be seen as a combination of standard filter design for deterministic signals with the random process theory of the previous section.
- This remarkable area was pioneered in the 1940's by Norbert Wiener, who designed methods for optimal estimation of a signal measured in noise. Specifically, consider the system in the figure below.

- A desired signal d_n is observed in noise v_n:

$$x_n = d_n + v_n$$



A segment of an acoustic waveform (music, voice) d_n that is corrupted by additive noise v_n

- Wiener showed how to design a linear filter which would optimally estimate d_n given just the noisy observations x_n and some assumptions about the statistics of the random signal and noise processes. This class of filters, the *Wiener filter*, forms the basis of many fundamental signal processing applications.
- Typical applications include:
  - Noise reduction e.g. for speech and music signals
  - Prediction of future values of a signal, e.g. in finance
  - Noise cancellation, e.g. for aircraft cockpit noise
  - Deconvolution, e.g. removal of room acoustics (dereverberation) or echo cancellation in telephony.

- The Wiener filter is a very powerful tool. However, it is only the optimal *linear* estimator for stationary signals. The *Kalman filter* offers an extension for non-stationary signals via *state space models*. In cases where a linear filter is still not good enough, non-linear filtering techniques can be adopted. See 4th year Signal Processing and Control modules for more advanced topics in these areas.

The Discrete-time Wiener Filter I

In a minor abuse of notation, and following standard conventions, we will refer to both random variables and their possible values in lower-case symbols, as this should cause no ambiguity for this section of work.

- In the most general case, we can filter the observed signal x_n with an infinite dimensional filter, having a non-causal impulse response

$$\{\dots, h_{-1}, h_0, h_1, \dots\} \tag{9}$$

- We filter the observed noisy signal using the filter $\{h_p\}$ to obtain an estimate $\hat{d}_n$ of the desired signal:

$$\hat{d}_n = \sum_{p=-\infty}^{\infty} h_p \, x_{n-p} \tag{10}$$

- Since both d_n and x_n are drawn from random processes $\{d_n\}$ and $\{x_n\}$, we can only measure performance of the filter in terms of *expectations*. The criterion adopted for Wiener filtering is the *mean-squared error (MSE)* criterion. First, form the error signal ϵ_n:

$$\epsilon_n = d_n - \hat{d}_n = d_n - \sum_{p=-\infty}^{\infty} h_p \, x_{n-p} \tag{11}$$

- The *mean-squared error (MSE)* is then defined as:

$$\boxed{J = \mathbb{E}[\epsilon_n^2]} \tag{12}$$

where the expectation is with respect to the random signal d and the random noise v.

- The Wiener filter minimises J with respect to the filter coefficients $\{h_p\}$.

$$\{h_p^*\} = \arg\min_{\{h_p\}} \mathbb{E}[\epsilon_n^2] \tag{13}$$

Solving for the Wiener filter I

- The Wiener filter assumes that $\{x_n\}$ and $\{d_n\}$ are *jointly wide-sense stationary*.

This means that the means of both processes are constant, and all autocorrelation functions/cross-correlation functions (e.g. $r_{xd}[n, m]$) depend only on the time difference $m - n$ between data points:

$$
\begin{aligned}
r_{xx}[n, m] &\to r_{xx}[m-n] \\
r_{dd}[n, m] &\to r_{dd}[m-n] \\
r_{xd}[n, m] &\to r_{xd}[m-n]
\end{aligned} \tag{14}
$$

Solving for the Wiener filter II

In fact we will assume that $\{d_n\}$ and $\{v_n\}$ have zero mean as well:

$$\mathbb{E}[d_n] = 0, \;\; \mathbb{E}[v_n] = 0 \tag{15}$$

Non-zero mean processes can be dealt with by first subtracting the mean before filtering.

- The expected error (12) may be minimised with respect to the impulse response values h_q. A sufficient condition for a minimum is:

$$\frac{\partial J}{\partial h_q} = \frac{\partial \mathbb{E}[\epsilon_n^2]}{\partial h_q} = \mathbb{E}\left[\frac{\partial \epsilon_n^2}{\partial h_q}\right] = \mathbb{E}\left[2\epsilon_n \frac{\partial \epsilon_n}{\partial h_q}\right] = 0 \tag{16}$$

simultaneously for all $q \in \{\ldots, -1, 0, 1, \ldots\}$.

Solving for the Wiener filter III

- The term $\frac{\partial \epsilon_n}{\partial h_q}$ is then calculated as:

$$\frac{\partial \epsilon_n}{\partial h_q} = \frac{\partial}{\partial h_q}\left\{ d_n - \sum_{p=-\infty}^{\infty} h_p\, x_{n-p} \right\} = -x_{n-q} \qquad (17)$$

This is because the random terms d_n and x_{n-p} do not depend on h_q and are hence treated as constant in the partial derivative.

Hence the coefficients must satisfy, for all q:

$$\mathbb{E}\left[\epsilon_n \frac{\partial \epsilon_n}{\partial h_q}\right] = -\mathbb{E}\left[\epsilon_n\, x_{n-q}\right] = 0$$

i.e.

$$\boxed{\mathbb{E}\left[\epsilon_n\, x_{n-q}\right] = 0; \qquad q = \ldots, -1, 0, 1 \ldots} \qquad (18)$$

Solving for the Wiener filter IV

Remember this! When you have found the optimal weight vectors,

$$\cdots = \mathbb{E}\left[\epsilon_n\, x_{-1}\right] = \mathbb{E}\left[\epsilon_n\, x_0\right] = \mathbb{E}\left[\epsilon_n\, x_1\right] = \cdots = 0$$

for all n.

Solving for the Wiener filter V

Now, substituting for ϵ_n in (18) gives:

$$\begin{aligned}
\mathbb{E}\left[\epsilon_n \, x_{n-q}\right] &= \mathbb{E}\left[\left(d_n - \sum_{p=-\infty}^{\infty} h_p \, x_{n-p}\right) x_{n-q}\right] \\
&= \mathbb{E}\left[d_n \, x_{n-q}\right] - \sum_{p=-\infty}^{\infty} h_p \; \mathbb{E}\left[x_{n-q} \, x_{n-p}\right] \\
&= r_{xd}[q] - \sum_{p=-\infty}^{\infty} h_p \, r_{xx}[q-p] \\
&= 0
\end{aligned}$$

Hence, rearranging, the solution must satisfy

$$\boxed{\sum_{p=-\infty}^{\infty} h_p \, r_{xx}[q-p] = r_{xd}[q], \qquad -\infty < q < +\infty} \qquad (19)$$

Solving for the Wiener filter VI

This is known as the *Wiener-Hopf* equations.

The *Wiener-Hopf* equations involve an infinite number of unknowns h_q. The simplest way to solve this is in the frequency domain. First note that the *Wiener-Hopf* equations can be rewritten as a discrete-time convolution:

$$(h * r_{xx})(q) = r_{xd}[q], \qquad -\infty < q < +\infty \tag{20}$$

Taking *discrete-time Fourier transforms (DTFT)* of both sides:

$$H(e^{j\omega})\mathcal{S}_x(e^{j\omega}) = \mathcal{S}_{xd}(e^{j\omega})$$

Here the term $\mathcal{S}_{xd}(e^{j\omega})$, is *defined* as the the DTFT of $r_{xd}[q]$ and is termed the *cross-power spectrum* of d and x.

The cross-power spectrum is in general complex valued. It has the property [check for yourself and in examples paper]:

Solving for the Wiener filter VII

$$\mathcal{S}_{xd}(e^{j\omega}) = \mathcal{S}_{dx}(e^{j\omega})^*$$

Hence, rearranging:

$$\boxed{H(e^{j\omega}) = \frac{\mathcal{S}_{xd}(e^{j\omega})}{\mathcal{S}_{x}(e^{j\omega})}} \tag{21}$$

Frequency domain Wiener filter

This very general result tells us that we can compute the frequency response of the optimal filter for estimating d, given knowledge of just the power spectrum (or equivalently the autocorrelation function) of the noisy signal x, and the cross power spectrum (or equivalently the cross-correlation function) between the noisy signal x and the desired signal d.

Solving for the Wiener filter VIII

Depending on the scenario, it may be possible to estimate these quantities directly from data, and/or from physical modelling considerations about the system. Estimation from data will typically require the process to be ergodic, hence time averages converge to ensemble averages of correlation functions.

Of course, our result in general yields a non-causal filter that is not implementable in practice. The practical approach is either to approximate the filtering in the frequency domain using DFTs, or derive sub-classes of Wiener filter that are causal and implementable, as considered shortly.

Important Special Case: Uncorrelated Signal and Noise Processes I

An important sub-class of the Wiener filter, which also gives considerable insight into filter behaviour, can be gained by considering the case where the desired signal process $\{d_n\}$ is uncorrelated with the noise process $\{v_n\}$, i.e.

$$r_{dv}[k] = \mathbb{E}[d_n v_{n+k}] = 0, \qquad -\infty < k < +\infty \tag{22}$$

This is a very typical scenario in which v might be environmental noise that is independent of the desired signal, and hence uncorrelated.

[Important side concept: random variables that are independent and zero mean are uncorrelated; however, uncorrelated random zero mean random variables are *not* assured to be independent.]

Important Special Case: Uncorrelated Signal and Noise Processes II

Consider the implications of this fact on the correlation functions required in the *Wiener-Hopf* equations:

$$\sum_{p=-\infty}^{\infty} h_p \, r_{xx}[q-p] = r_{xd}[q], \qquad -\infty < q < +\infty$$

1. r_{xd} is

$$r_{xd}[q] = \mathbb{E}[x_n d_{n+q}] = \mathbb{E}[(d_n + v_n) d_{n+q}] \tag{23}$$

$$= \mathbb{E}[d_n d_{n+q}] + \mathbb{E}[v_n d_{n+q}] = r_{dd}[q] \tag{24}$$

since $\{d_n\}$ and $\{v_n\}$ are uncorrelated.
Hence, taking DTFT we have:

$$\mathcal{S}_{xd}(e^{j\omega}) = \mathcal{S}_d(e^{j\omega}) \tag{25}$$

Important Special Case: Uncorrelated Signal and Noise Processes III

2. r_{xx} is

$$
\begin{aligned}
r_{xx}[q] &= \mathbb{E}[x_n x_{n+q}] = \mathbb{E}[(d_n + v_n)(d_{n+q} + v_{n+q})] \\
&= \mathbb{E}[d_n d_{n+q}] + \mathbb{E}[v_n v_{n+q}] + \mathbb{E}[d_n v_{n+q}] + \mathbb{E}[v_n d_{n+q}] \\
&= \mathbb{E}[d_n d_{n+q}] + \mathbb{E}[v_n v_{n+q}]
\end{aligned}
$$

$$r_{xx}[q] = r_{dd}[q] + r_{vv}[q] \tag{26}$$

Hence

$$\mathcal{S}_x(e^{j\omega}) = \mathcal{S}_d(e^{j\omega}) + \mathcal{S}_v(e^{j\omega}) \tag{27}$$

Important Special Case: Uncorrelated Signal and Noise Processes IV

Thus the Wiener filter becomes

$$H(e^{j\omega}) = \frac{\mathcal{S}_d(e^{j\omega})}{\mathcal{S}_d(e^{j\omega}) + \mathcal{S}_v(e^{j\omega})} = \frac{1}{1 + 1/\rho(\omega)} \tag{28}$$

where $\rho(\omega) = \mathcal{S}_d(e^{j\omega})/\mathcal{S}_v(e^{j\omega})$ is the (frequency-dependent) signal-to-noise (SNR) power ratio.

The interpretation of this result is:

- The gain is always non-negative, and ranges between 0 and 1.
- The filter never boosts a particular frequency component but is an optimal attenuation rule.

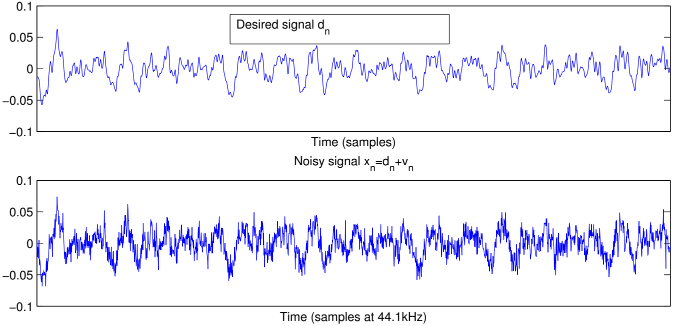
Important Special Case: Uncorrelated Signal and Noise Processes V

- At ω where the SNR is large, the gain tends to unity, whereas the gain tends to a small value at those frequencies where the SNR is small. (When the signal is very noisy, the best estimate $\hat{d}_n$ the filter can make is zero.)

Case Study: Audio Noise Reduction I

Consider a section of acoustic waveform (music, voice, ...) d_n that is corrupted by additive noise v_n

$$x_n = d_n + v_n$$



Case Study: Audio Noise Reduction II

- Assume that the section of data is wide-sense stationary and ergodic (approx. true for a short segment around 1/40 s).
- Assume also that the noise is white with variance σ_v^2:

$$r_{vv}[k] = \sigma_v^2 \delta[k]$$

and uncorrelated with the audio signal: for all n and k

$$\mathbb{E}[d_n v_k] = 0$$

- The Wiener filter in this case needs $r_{xx}[k]$ and

$$r_{xd}[k] = \mathbb{E}[x_n d_{n+k}] = \mathbb{E}[d_n d_{n+k}] + \mathbb{E}[v_n d_{n+k}] = r_{dd}[k].$$

Case Study: Audio Noise Reduction III

We can implement the Wiener filter if we have noisy acoustic data x_n and we know σ_v^2.

- We will estimate these quantities with sample averages:

$$r_{xx}[k] \approx \frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n+k}$$

$$r_{dd}[k] = r_{xx}[k] - r_{vv}[k] = \begin{cases} r_{xx}[k], & k \neq 0 \\ r_{xx}[0] - \sigma_v^2, & k = 0 \end{cases}$$

- Now calculated the PSDs and use equation (21) to solve for the Wiener filter.

Mean-squared error for the optimal filter I

The derivations thus far show how to calculate the optimal filter for a given problem. They don't, however, tell us how well that optimal filter performs. This can be assessed from the mean-squared error value of the optimal filter:

$$
\begin{aligned}
J = \mathbb{E}[\epsilon_n^2] &= \mathbb{E}[\epsilon_n (d_n - \sum_{p=-\infty}^{\infty} h_p \, x_{n-p})] \\
&= \mathbb{E}[\epsilon_n \, d_n] - \sum_{p=-\infty}^{\infty} h_p \, \mathbb{E}[\epsilon_n \, x_{n-p}] \qquad (29)
\end{aligned}
$$

Mean-squared error for the optimal filter II

When the filter is the Wiener filter, the sum term on the right-hand side is zero, seecondition (18). The minimum error is thus

$$\begin{aligned} J_{\text{min}} &= \mathbb{E}[\epsilon_n \, d_n] \\ &= \mathbb{E}[(d_n - \sum_{p=-\infty}^{\infty} h_p \, x_{n-p}) \, d_n] \\ &= r_{dd}[0] - \sum_{p=-\infty}^{\infty} h_p \, r_{dx}[-p] \\ &= r_{dd}[0] - (h * r_{dx})[0] \end{aligned} \tag{30}$$

In the frequency domain the result is

$$J_{\text{min}} = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \mathcal{S}_d(e^{j\omega}) - H(e^{j\omega}) \mathcal{S}_{xd}^*(e^{j\omega}) d\omega \tag{31}$$

Example: AR Process I

An autoregressive process $\{d_n\}$ of order 1 is generated as:

$$d_n = a_1 d_{n-1} + e_n$$

with e_n as zero mean white noise having variance σ_e^2.
We can write in z-transform domain and zero initial conditions:

$$D(z) = a_1 z^{-1} D(z) + E(z)$$

Hence

$$D(z) = \frac{1}{1 - a_1 z^{-1}} E(z) = H(z) E(z)$$

where $H(z) = \frac{1}{1 - a_1 z^{-1}}$ is a *transfer function* between e and d.
The frequency response is thus:

$$H(\exp(j\omega)) = \frac{1}{1 - a_1 \exp(-j\omega)}$$

Example: AR Process II

We then obtain the power spectrum from the Linear Systems result as:

$$\mathcal{S}_d(e^{j\omega}) = |H(\exp(j\omega))|^2 \mathcal{S}_e(e^{j\omega}) = \frac{\sigma_e^2}{(1 - a_1 e^{-j\omega})(1 - a_1 e^{j\omega})}$$

Suppose the process is observed in zero mean white noise with variance σ_v^2, which is uncorrelated with $\{d_n\}$:

$$x_n = d_n + v_n$$

The Wiener filter for estimation of d_n: since the noise and desired signal are uncorrelated, we can use the form of Wiener filter from the

Example: AR Process III

previous page. Substituting in the various terms and rearranging, its frequency response is:

$$H^{\text{opt}}(e^{j\omega}) = \frac{\sigma_e^2}{\sigma_e^2 + \sigma_v^2(1 - a_1 e^{-j\omega})(1 - a_1 e^{j\omega})}$$

The impulse response of the filter can be found by inverse (discrete-time) Fourier transforming the frequency response. This is studied in the examples paper.

The FIR Wiener filter I

We note that, in general, the Wiener filter given by equation (21) is an IIR filter and thus non-causal and cannot be implemented online.

Here we consider a practical alternative in which a causal Pth order Finite Impulse Response (FIR) Wiener filter is developed. In the FIR case the signal estimate is formed as

$$\hat{d}_n = \sum_{p=0}^{P} h_p \, x_{n-p} \tag{32}$$

We minimise, as before, the objective function

$$J = \mathbb{E}[(d_n - \hat{d}_n)^2]$$

The FIR Wiener filter II

The filter derivation proceeds much as before, and we need to solve

$$\frac{\partial J}{\partial h_q} = \mathbb{E}\left[2\epsilon_n \frac{\partial \epsilon_n}{\partial h_q}\right] = 0, \qquad q = 0, 1, \ldots, P$$

Then, as before:

$$\frac{\partial \epsilon_n}{\partial h_q} = \frac{\partial}{\partial h_q}\left(d_n - \sum_{p=0}^{P} h_p \, x_{n-p}\right) = -x_{n-q}$$

leading to the *optimality* equations

$$\boxed{\mathbb{E}\left[\epsilon_n \, x_{n-q}\right] = 0; \qquad q = 0, \ldots, P} \tag{33}$$

The FIR Wiener filter III

Now, substituting for ϵ_n in (33) gives:

$$\begin{aligned}
\mathbb{E}\left[\epsilon_n\, x_{n-q}\right] &= \mathbb{E}\left[\left(d_n - \sum_{p=0}^{P} h_p\, x_{n-p}\right)\, x_{n-q}\right] \\
&= \mathbb{E}\left[d_n\, x_{n-q}\right] - \sum_{p=0}^{P} h_p\; \mathbb{E}\left[x_{n-q}\, x_{n-p}\right] \\
&= r_{xd}[q] - \sum_{p=0}^{P} h_p\, r_{xx}[q-p] = 0
\end{aligned}$$

The *Wiener-Hopf* equations are

$$\boxed{\sum_{p=0}^{P} h_p\, r_{xx}[q-p] = r_{xd}[q], \qquad q = 0, 1, \ldots, P} \tag{34}$$

The FIR Wiener filter IV

These equations may be written in matrix form as

$$\mathbf{R}_x \mathbf{h} = \mathbf{r}_{xd} \quad \text{or} \quad \mathbf{h} = \mathbf{R}_x{}^{-1}\mathbf{r}_{xd}$$

where:

$$\mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_P \end{bmatrix} \qquad \mathbf{r}_{xd} = \begin{bmatrix} r_{xd}[0] \\ r_{xd}[1] \\ \vdots \\ r_{xd}[P] \end{bmatrix}$$

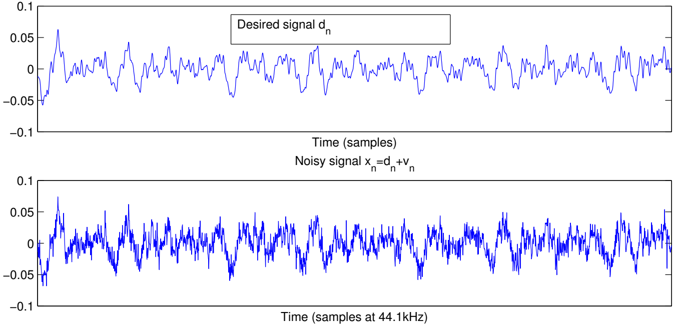
and

$$\mathbf{R}_x = \begin{bmatrix} r_{xx}[0] & r_{xx}[1] & \cdots & r_{xx}[P] \\ r_{xx}[1] & r_{xx}[0] & \dots & r_{xx}[P-1] \\ \vdots & \vdots & & \vdots \\ r_{xx}[P] & r_{xx}[P-1] & \cdots & r_{xx}[0] \end{bmatrix}$$

Case Study: Audio Noise Reduction I

- Consider a section of acoustic waveform (music, voice, ...) d_n that is corrupted by additive noise v_n

$$x_n = d_n + v_n$$



Case Study: Audio Noise Reduction II

- We could try and noise reduce the signal using the FIR Wiener filter.
- Assume that the section of data is wide-sense stationary and ergodic (approx. true for a short segment around 1/40 s). Assume also that the noise is white and uncorrelated with the audio signal - with variance σ_v^2, i.e.

$$r_{vv}[k] = \sigma_v^2 \delta[k]$$

- The Wiener filter in this case needs (see eq. (35)):

$$r_{xx}[k], \quad \text{Autocorrelation of noisy signal}$$

$$r_{xd}[k] = r_{dd}[k] \quad \text{Autocorrelation of desired signal}$$

[since noise uncorrelated with signal, as in eq. (24)]

Case Study: Audio Noise Reduction III

- Since signal is assumed ergodic, we can estimate these quantities:

$$r_{xx}[k] \approx \frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n+k}$$

$$r_{dd}[k] = r_{xx}[k] - r_{vv}[k] = \begin{cases} r_{xx}[k], & k \neq 0 \\ r_{xx}[0] - \sigma_v^2, & k = 0 \end{cases}$$

- Note have to be careful that $r_{dd}[k]$ is still a valid autocorrelation sequence since r_{xx} is just an estimate.

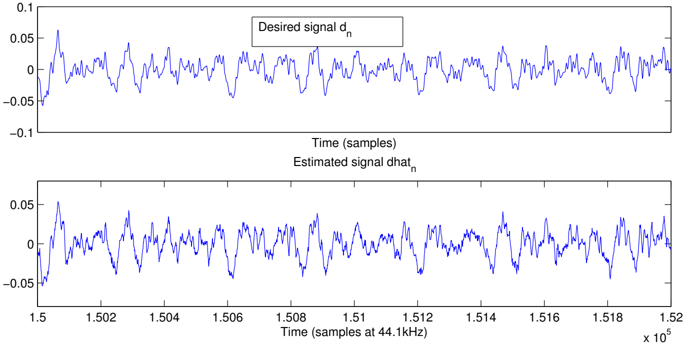
Case Study: Audio Noise Reduction IV

- Under what conditions would $r_{xx}[k]$ form a valid autocorrelation sequence for construction of $\mathbf{R}_x$?
  The argument below applies both to the autocorrelation sequences $r_{dd}[k]$ and $r_{xx}[k]$, though it is posed in terms of just x.
  It turns out that a necessary condition is that the resulting $\mathbf{R}_x$ matrix is positive definite.
- Choose the filter length P, form the autocorrelation matrix and cross-correlation vector and solve in e.g. Matlab:

$$\mathbf{h} = \mathbf{R}_x{}^{-1}\mathbf{r}_{xd}$$

Case Study: Audio Noise Reduction V

- The output looks like this, with $P = 350$:



Case Study: Audio Noise Reduction VI

- The minimum error is

$$
\begin{aligned}
J_{\text{min}} &= \mathbb{E}[\epsilon_n(d_n - \hat{d}_n)] \\
&= \mathbb{E}[\epsilon_n d_n] \qquad \text{using (33)} \\
&= \mathbb{E}\Big[\Big(d_n - \sum_{p=0}^{P} h_p\, x_{n-p}\Big)\, d_n\Big] \\
&= r_{dd}[0] - \sum_{p=0}^{P} h_p\, r_{xd}[p] \\
&= r_{dd}[0] - \mathbf{r}_{xd}^T \mathbf{h}
\end{aligned}
$$

- To find the best filter length, we can compute this for various filter lengths.
- If you can estimate $\mathbf{h}$ from the data, then J_{min} can also be estimated.

Extending the Wiener filter I

The Wiener Filter can readily be extended to deal with cases outside the regular noise reduction case. You simply replace the desired signal d with whatever one wants to predict or estimate and then rederive the new version of the filter.
More generally, for the FIR case the formula

$$\mathbf{h} = \mathbf{R}_x{}^{-1}\mathbf{r}_{xd}$$

applies whatever the form of the desired signal and the 'noisy' signal, provided you can calculate the necessary correlation functions.
You will be expected to be able to work out the theory for non-standard cases in the exam. Some standard examples are:

Extending the Wiener filter II

- The aim is to design a predictor of a signal $\{u_n\}$ from noisy measurements of it:

$$x_n = u_n + v_n$$

where v_n is noise. As before denote the output of the Wiener filter as

$$\hat{d}_n = \sum_{p=0}^{P} h_p \, x_{n-p}$$

- Choosing the filter coefficients h_p to minimise $\mathbb{E}\left[\epsilon_n^2\right]$ where

$$\epsilon_n = u_{n+1} - \hat{d}_n = d_n - \hat{d}_n$$

gives the best one-step predictor.

- Set $d_n = u_{n+L}$ if L is the desired prediction horizon.

Extending the Wiener filter III

- Smoothing of a noisy signal. In this case we use the current samples to get an even better estimate of the signal at some point in the past. The setup is exactly the same as for prediction, except that $d_n = u_{n-L}$ where L is the amount of 'lookahead' that we can allow. This would be appropriate in systems where a certain time-lag or latency is allowable before the signal estimate needs to be obtained.
- Deconvolution. A somewhat more complex example is to extract a signal u_n from a noisy convolved version of itself:

$$x_n = \sum_{q=0}^{Q} h'_q u_{n-q} + v_n$$

The setup is the same as for the regular filter, setting $d_n = u_n$, but we will have a more complex expression for the

Extending the Wiener filter IV

autocorrelation function of x_n and the cross-correlation between d and x.
A good application of this would be removal of room acoustics from a voice signal u_n.

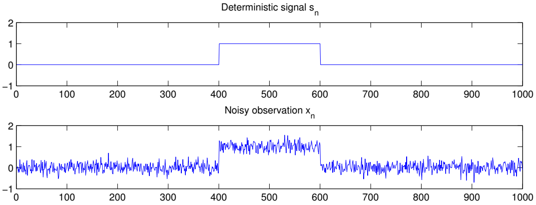
Section 2: Signal Detection: Matched Filters I

- The Wiener filter shows how to extract a random signal from a random noise environment.
- How about the (apparently) simpler task of detecting a known deterministic signal

$$s_0, s_1, \ldots, s_{N-1}$$

of known length buried in random noise v_n:

$$x_n = s_n + v_n$$



Section 2: Signal Detection: Matched Filters II

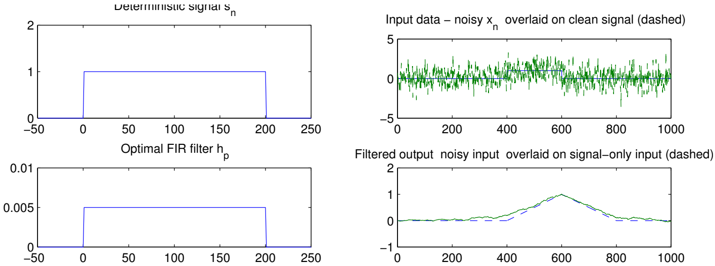
- Note that the segment of x_n that contains the signal of interest is unknown. (We do not know when the transmission of s_0 begins.)
- The technical method for finding the signal of interest in x_n is called *matched* filtering.
- It finds extensive application in detection of pulses in communications data, radar and sonar data.
- The matched filter is the optimal FIR filter with coefficients $\{h_0, h_1, \ldots, h_{N-1}\}$ for detecting s_n:

$$y_n = \sum_{m=0}^{N-1} h_m x_{n-m}$$

- Note the matched filter length is equal to length of the signal of interest.

Section 2: Signal Detection: Matched Filters III

- Monitor output y_n of the matched filter until a maxima is detected:



- Once we have found the maxima, we say that is the time point the signal of interest was completely located. In the above figure, time 600 is correctly identified.

- To formulate the problem, assume that the signal of interest is transmitted at time 0. (In practice we would now run this filter over a much longer length of data x which contains s at some unknown position and find the time at which maximum energy occurs.)
- 'Vectorise' all signals

$$\mathbf{x} = \mathbf{s} + \mathbf{v}$$

$$\mathbf{s} = [s_0,\ s_1, ..., s_{N-1}]^T,\ \mathbf{x} = [x_0,\ x_1, ..., x_{N-1}]^T$$

- The filter has coefficients h_m, for $m = 0, 1, \ldots, N-1$ and the output of the filter at time $N-1$ is:

$$\begin{aligned} y_{N-1} &= \sum_{m=0}^{N-1} h_m x_{N-1-m} = \mathbf{h}^T \tilde{\mathbf{x}} = \mathbf{h}^T (\tilde{\mathbf{s}} + \tilde{\mathbf{v}}) = \mathbf{h}^T \tilde{\mathbf{s}} + \mathbf{h}^T \tilde{\mathbf{v}} \\ &= y^s_{N-1} + y^n_{N-1} \end{aligned}$$

where

$$\tilde{\mathbf{x}} = \left[x_{N-1}, x_{N-2}, ..., x_0\right]^T$$

is the 'time-reversed' vector, and $\tilde{\mathbf{s}}$ defined similarly. y^s_{N-1} is defined as the output from the signal-only part and y^s_{N-1} is the output from just the noise going through the filter.

- Define output SNR as:

$$\mathrm{SNR}(\mathbf{h}) = \frac{\mathbb{E}[|y^s_{N-1}|^2]}{\mathbb{E}[|y^n_{N-1}|^2]} = \frac{\mathbb{E}[|\mathbf{h}^T\tilde{\mathbf{s}}|^2]}{\mathbb{E}[|\mathbf{h}^T\tilde{\mathbf{v}}|^2]} = \frac{|\mathbf{h}^T\tilde{\mathbf{s}}|^2}{\mathbb{E}[|\mathbf{h}^T\tilde{\mathbf{v}}|^2]}$$

since numerator is not a random quantity.

- Find the filter $\mathbf{h}$ that maximises the SNR. Since

$$\mathrm{SNR}(\mathbf{h}) = \mathrm{SNR}(c\mathbf{h})$$

where c is any real number, make the solution unique by finding the best unit norm $\mathbf{h}$.

Noise output energy I

- Consider the expected noise output energy, which may be simplified as follows:

$$\mathbb{E}[|\mathbf{h}^T\tilde{\mathbf{v}}|^2] = \mathbb{E}[\mathbf{h}^T\tilde{\mathbf{v}}\tilde{\mathbf{v}}^T\mathbf{h}] = \mathbf{h}^T\mathbb{E}[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^T]\mathbf{h}$$

- We will here consider the case where the noise is white and zero mean with variance σ_v^2. Then, for any time indexes $i = 0, \ldots, N-1$ and $j = 0, \ldots, N-1$:

$$\mathbb{E}[v_i v_j] = \begin{cases} \sigma_v^2, & i = j \\ 0, & i \neq j \end{cases}$$

and hence

$$\mathbb{E}[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^T] = \sigma_v^2\mathbf{I}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix
[diagonal elements correspond to $i = j$ terms and off-diagonal to $i \neq j$ terms].

Noise output energy II

- So, we have finally the expression for the noise output energy:

$$\mathbb{E}[|\mathbf{h}^T\tilde{\mathbf{v}}|^2] = \mathbf{h}^T\sigma_v^2\mathbf{I}\mathbf{h} = \sigma_v^2\mathbf{h}^T\mathbf{h} = \sigma_v^2$$

  since $\mathbf{h}^T\mathbf{h} = 1$.
- For any unit norm $\mathbf{h}$, the denominator of $\mathrm{SNR}(\mathbf{h})$ is always σ_v^2.

Signal output energy I

- The signal component at the output is $y^s_{N-1} = \mathbf{h}^T \tilde{\mathbf{s}}$, with energy
$$|\mathbf{h}^T \tilde{\mathbf{s}}|^2 = (\mathbf{h}^T \tilde{\mathbf{s}})(\tilde{\mathbf{s}}^T \mathbf{h}) = \mathbf{h}^T (\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T)\mathbf{h}$$
- To analyse this, consider the matrix $\mathbf{M} = \tilde{\mathbf{s}}\tilde{\mathbf{s}}^T$. What are its eigenvectors/eigenvalues ?
- From 1A Maths, finding the unit norm vector $\mathbf{h}$ that maximises $\mathbf{h}^T(\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T)\mathbf{h}$ is equivalent to finding the largest eigenvalue of $\mathbf{M} = \tilde{\mathbf{s}}\tilde{\mathbf{s}}^T$.
- Recall the definition of eigenvectors ($\mathbf{e}$) and eigenvalues (λ):

$$\mathbf{M}\mathbf{e} = \lambda \mathbf{e}$$

Signal output energy II

- Try $\mathbf{e} = \tilde{\mathbf{s}}/|\tilde{\mathbf{s}}|$:

$$\mathbf{M}\frac{\tilde{\mathbf{s}}}{|\tilde{\mathbf{s}}|} = \tilde{\mathbf{s}}|\tilde{\mathbf{s}}|$$

  Hence $\tilde{\mathbf{s}}/|\tilde{\mathbf{s}}|$ is an eigenvector with eigenvalue $\lambda = (\tilde{\mathbf{s}}^T\tilde{\mathbf{s}})$.

- All other eigenvalues of $\mathbf{M}$ are zero.
- We have the solution as:

$$\mathbf{h}^{\mathrm{opt}} = \frac{\tilde{\mathbf{s}}}{|\tilde{\mathbf{s}}|}$$

  i.e. the optimal filter coefficients are just the (normalised) time-reversed signal!

Signal output energy III

- The SNR at the optimal filter setting is given by

$$\mathrm{SNR}^{\mathrm{opt}} = \frac{\tilde{\mathbf{s}}^T \tilde{\mathbf{s}}}{\sigma_v^2}$$

  and clearly the performance depends (as expected) very much on the energy of the signal s and the noise v.

Practical Implementation of the matched filter I

- We chose a batch of data of same length as the signal **s** and optimised a filter **h** of the same length.
- In practice we would now run this filter over a much longer length of data **x** which contains **s** at some unknown position and find the time at which maximum energy occurs. This is the point at which **s** can be detected, and optimal thresholds can be devised to make the decision on whether a detection of **s** should be declared at that time.
- In fact, we have only proved that the signal to noise ratio is maximised at one single filter output time, $n = N - 1$. However, given a stationary noise process, it is straightforward to show that the signal output power term for the optimal filter is always less than that computed for $n = N - 1$ since the deterministic correlation function of s_n always has its maximum at lag zero.

Practical Implementation of the matched filter II

- Example (like a simple square pulse radar detection problem):

$$s_n = \text{Rectangle pulse} = \begin{cases} 1, & n = 0, 1, ..., T-1 \\ 0, & \text{otherwise} \end{cases}$$

- Optimal filter is the (normalised) time reversed version of s_n:

$$h_n^{\text{opt}} = \begin{cases} 1/\sqrt{T}, & n = 0, 1, ..., T-1 \\ 0, & \text{otherwise} \end{cases}$$

Practical Implementation of the matched filter III

- SNR achievable at detection point:

$$\text{SNR}^{\text{opt}} = \frac{\tilde{\mathbf{s}}^T \tilde{\mathbf{s}}}{\sigma_v^2} = \frac{T}{\sigma_v^2}$$

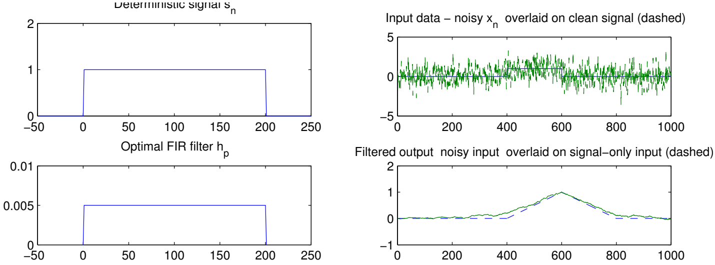
Compare with best SNR attainable before matched filtering:

$$\text{SNR} = \frac{\text{Max signal value}^2}{\text{Average noise energy}} = \frac{1}{\sigma_v^2}$$

i.e. a factor of T improvemment, which could be substantial for long pulses $T >> 1$.

Practical Implementation of the matched filter IV

- The figures below shows the input signal, which is the signal we are trying to detect buried in noise, being filtered (or convolved) with the optimal FIR filter. Note how the output is maximised at the first instance the filter "sees" the entire signal to be detected.



Practical Implementation of the matched filter V

- See below for a different case where the signal is a saw-tooth pulse:

$$s_n = \text{Sawtooth pulse} = \begin{cases} n+1, & n = 0, 1, ..., T-1 \\ 0, & \text{otherwise} \end{cases}$$