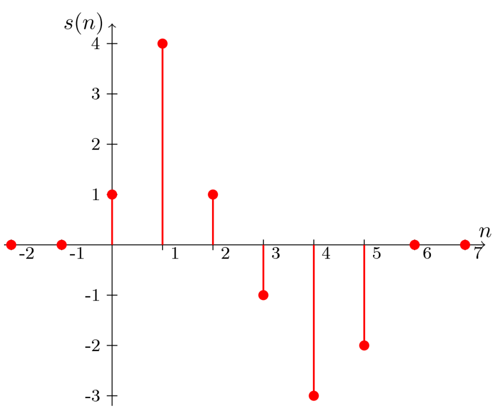
Homework 9

Table of Contents

ADSI Problem 6.1: Matched filters

Consider a deterministic signal that is only non-zero in the vicinity of *n*=0 as shown below and zero for all other values of n.



The signal is disturbed by additive noise from an AR(3) process given by

$$v(n) = -0.5v(n-1) - 0.5v(n-2) - 0.25v(n-3) + w(n)$$
$$\text{where } w(n) \sim WGN(0, 4).$$

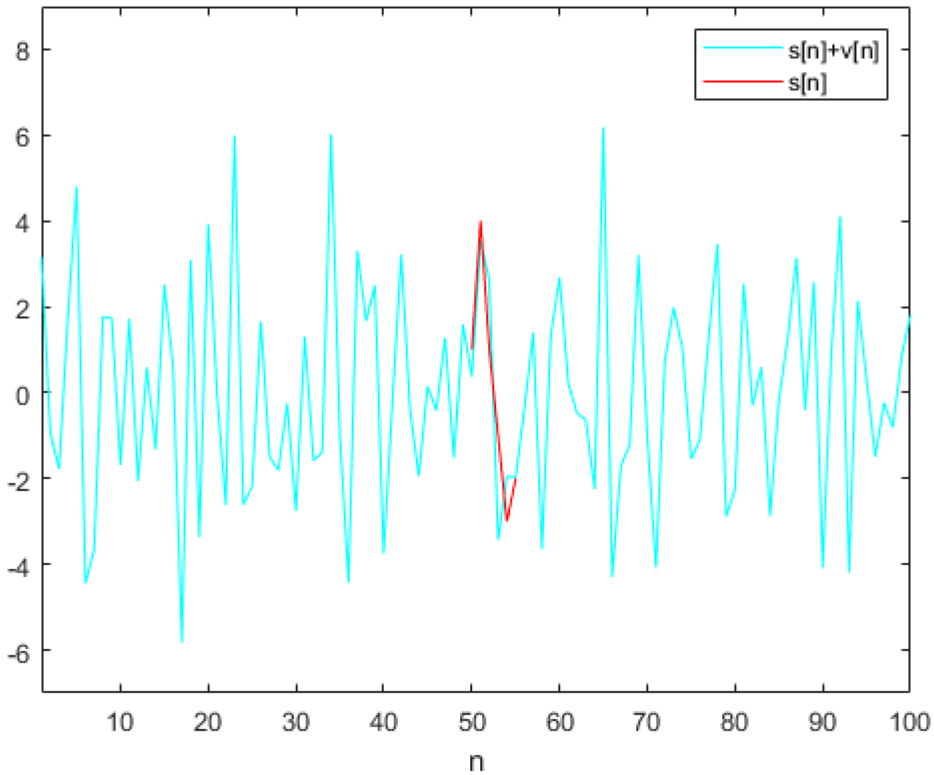
The goal of this problem is to construct a matched filter that can help us distinguish between the presence or absence of the signal in the noise.

```
clear variables;
```

1) Plot a realisation of the signal with the added noise

Create a realization of the noise and add the signal somewhere in the noise. Plot the result and comment on whether the signal is detectable.

```
D = 50; % The place to embed signal into the noise


% Generate the deterministic signal s[n]
s = [1, 4, 1, -1, -3, -2];

% Generate white noise w[n] with zero mean and variance 4
w_var = 4;
w = sqrt(w_var) * randn(1, 10000);

% Generate noise from an AR(3) process
b = 1;
a = [1, 0.5, 0.5, 0.25];
v = filter(b, a, w);

% Extract 100 samples for plotting
x = v(900:1000);

% Embed the signal s[n] in v[n] from D
idx_start = D;
idx_end =   D + length(s) - 1;
x(idx_start:idx_end) = x(idx_start:idx_end) + s;

plot(1:length(x),       x, 'c', ...
     idx_start:idx_end, s, 'r') % the signal is completely invisible
legend('s[n]+v[n]', 's[n]')
xlabel('n')
xlim([1,100])
ylim([-7, 9])
```



The signal is cannot be distinguished from the noise.

2) Plot the power density spectrum of the noise and spectrum of the signal

Plot the power density spectrum of the noise and spectrum of the signal, see Eq. (14.91).

The noise is generated by an AR(3) process plus some white Gaussian noise:

$$v(n) = -0.5v(n-1) - 0.5v(n-2) - 0.25v(n-3) + w(n)$$
$$\text{where } w(n) \sim WGN(0, 4).$$

We can determine the Power Density Spectrum of an ARMA(p,q) process is given by

$$S_{yy}(\omega) = \sigma_x^2 \left|H(\mathrm{e}^{\mathrm{j}\omega})\right|^2 = \sigma_x^2 \left| \frac{\sum_{k=0}^{q} b_k \mathrm{e}^{-\mathrm{j}\omega k}}{1 + \sum_{k=1}^{p} a_k \mathrm{e}^{-\mathrm{j}\omega k}} \right|^2. \qquad (13.133)$$

The power spectrum of an AR(p) process is given by:

$$S_{yy}(\omega) = \sigma_x^2 \left| \frac{1}{1 + \sum_{k=1}^{p} a_k e^{-j\omega k}} \right|^2$$

For this problem, we have:

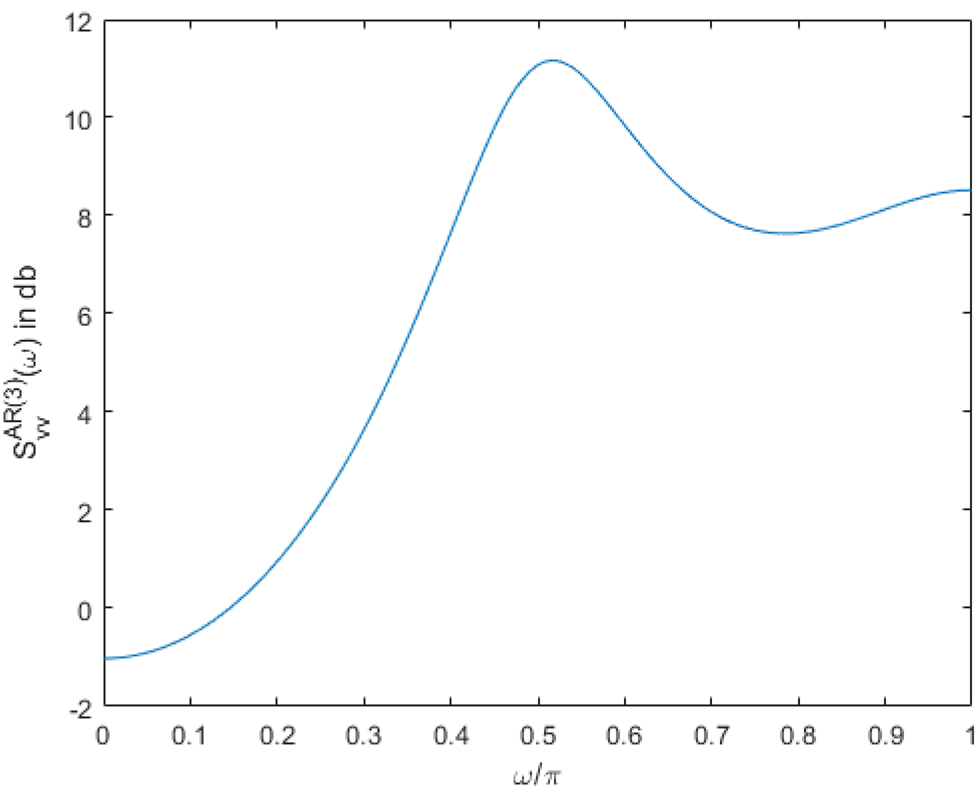
$$S_{\mathrm{vv}}(\omega) = 4\left|\frac{1}{1 + 0.5e^{-j\omega} + 0.5e^{-\mathrm{j}2\omega} + 0.25e^{-\mathrm{j}3\omega}}\right|^2$$

The algorithm is as follows:

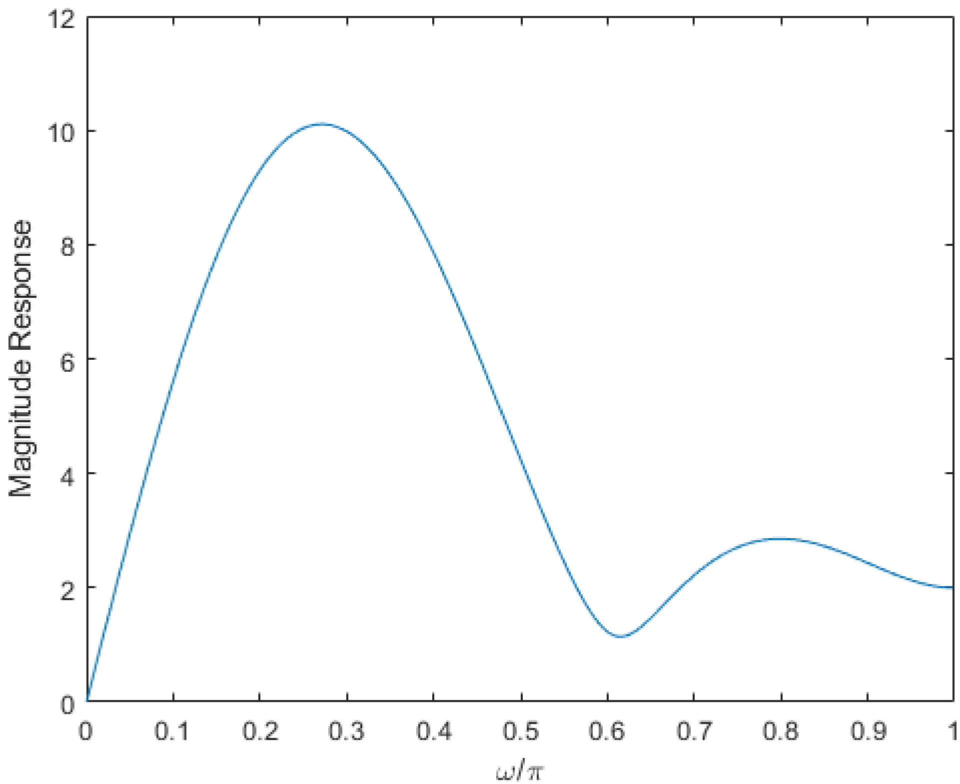
1. Use the coefficents $\{a_1, a_2, \cdots, a_p\}$ for the $\mathrm{AR}(p)$ model,
2. Compute the transfer for the $\mathrm{AR}(p)$ by computing the sum and finding its reciprocal
3. Compute the conjugate of the transfer function: $|H(e^{j\omega})|^2$
4. Multiply it with the variance σ_x^2

The algorithm is implemented in the functions `ar2psd()` function:

```
N = 256;

a = [0.5, 0.5, 0.25]; % The coefficients of the AR(3) model
w_var = 4; % The variance of white noise
[S_vv, w] = ar2psd(a, w_var, N); % Compute the PDS of AR(3) model

plot(w/pi, pow2db(S_vv))
xlabel('\omega/\pi')
ylabel('S_{vv}^{AR(3)}(\omega) in db')
```



```
[H,w2] = freqz(s, 1);
plot(w2/pi, abs(H))
xlabel('\omega/\pi')
ylabel('Magnitude Response')
```



3) Calculate the matched filter

Calculate the matched filter, the frequency response of the matched filter and the optimum signal to noise ratio. You can use `xcorr` *on the noise realization to find* $\boldsymbol{R}_v$.

4) Does the frequency response of the matched filter make sense?

Does the frequency response of the matched filter make sense seen in relation with the spectra from question 2?

5) How efficient is the matched filter?

Plot the signal before and after the matched filter as well as the square of the output of the matched filter on the same graph and comment on the effiency of the matched filter in our quest to detect the presence of the signal.

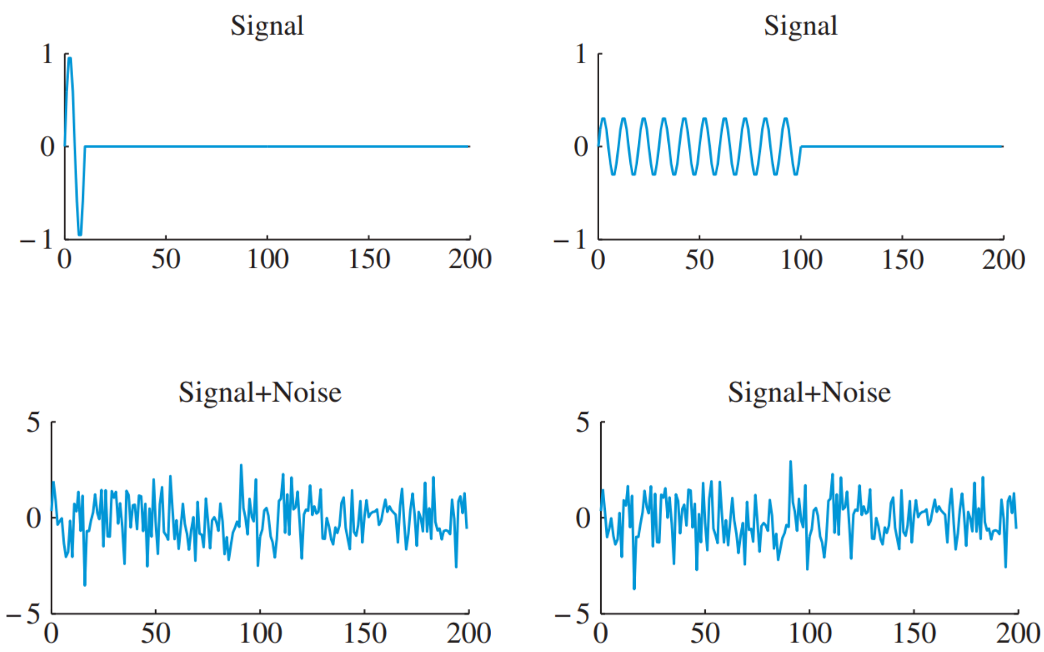
6) What happens to the optimum SNR when the noise is twice as powerful?

Is the optimum signal to noise ratio halved if the AR(3) is instead driven by twice as powerful white noise, i.e. $w(n) \sim WGN(0,8)$? Why or why not?

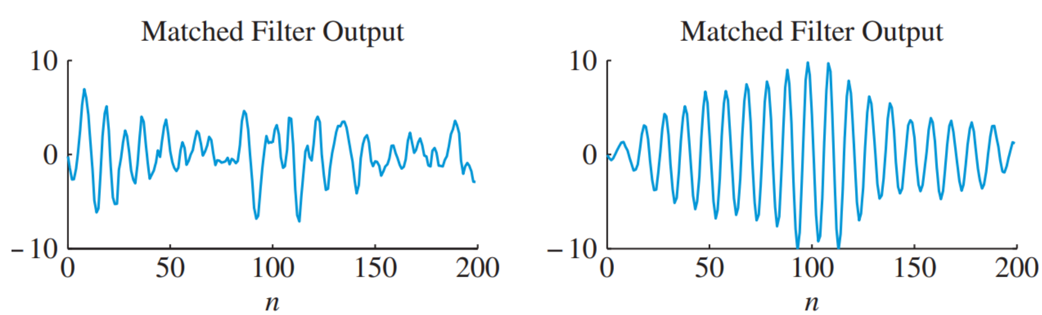
Problem 14.14: Show identity relations hold for auto- and cross-covariance sequences

In this problem we discuss in detail the matched filtering problem illustrated in Figure 14.18.

Figure 14.18 The operation of a matched filter in white noise. The two signals have different length (p = 10 and p = 100, respectively) but the same energy.



We note that the peak response of the matched filter occurs at $n0 = p - 1$ because there is no time delay (D = 0).



1) Determine the impulse response of the matched filter and the output SNR.

Determine the impulse response of the matched filter and the output SNR. Explain why, in the absence of noise, the output is the ACRS of the desired signal.

2) Determine the matched filter of a short cosine signal

Suppose that $p = 10$ and $s_i[n] = \cos\left(2\pi \frac{n}{10}\right)$.

Generate *N* = 200 samples of the noisy signal *x*[*n*] and process it through the matched filter designed for $s_i[n]$.

Plot the desired, input, and filtered signals and determine when the matched filter output is output.

2) Determine the matched filter of a long cosine signal

4) Which signal can be detected more easily by visual inspection?

Which signal can be detected more easily by visual inspection of the matched filter output? Is this justified by comparing the output SNR in each case?

Problem 14.35:

Consider the two 10-point signals *x*0[*n*] and *x*1[*n*] given below:

```
clear variables;
```

Problem 14.37:

```
clear variables;
```

Exam 2013, Problem 3

```
clear variables;
```

Functions

```
function [S, w] = ar2psd(a, v, N)
% AR2PSD Compute the Power Spectral Density from AR(p) coefficients
%   [S, w] = ar2psd(a, v, N)
% a: AR(p) coefficients
% v: the variance
% N: number of points in the range [1, pi]
% S: the estimated power spectrum
% w: frequencies
    w = linspace(0, 1, N) * pi;

    % Compute the transfer function
    % Used Eq. (13.133) in the book
    H = ones(N, 1);
    for k=1:numel(a)
        H = H + a(k)*exp(-1j * w' * k);
    end
    H = 1./H;

    % Finally compute the PSD
    S = v * H.*conj(H);
end
```