

Matlab[©] Exercise II: Active Noise Control with an FIR filter

Ruben BIESHEUVEL[†] and Mars GEUZE[‡]

[†]Student number 4076680

[‡]Student number 4109139

March 4, 2015

Question 1

Question 1.1

Derivation of Wiener-Hopf

Given are the signals v_1 and v_2 generated as follows:

$$v_1(n) - a_1(1)v_1(n-1) = b_1(0)g(n) - b_1(1)g(n-1); \quad (1)$$

$$v_2(n) - a_2(1)v_1(n-1) = b_2(0)g(n) - b_2(1)g(n-1), \quad (2)$$

where $g(n)$ is white noise with variance σ_g^2 and with parameters as given in Table 1. By applying the z -transform, the equations can be written as systems with transfer functions:

$$H_{v_1}(z) = \frac{V_1(z)}{G(z)} = \frac{b_1(0) - b_1(1)z^{-1}}{1 - a_1(1)z^{-1}}; \quad (3)$$

$$H_{v_2}(z) = \frac{V_2(z)}{G(z)} = \frac{b_2(0) - b_2(1)z^{-1}}{1 - a_2(1)z^{-1}}. \quad (4)$$

The poles of $H_{v_1}(z)$ and $H_{v_2}(z)$ are $a_1(1)$ and $a_2(1)$ respectively, which are inside of the unit circle, and therefore these causal filters are stable. From this and since we are filtering white noise we can conclude that both v_1 and v_2 are WSS.

The signal $v_2(n)$ must be filtered by a filter $W(z)$ with order m in such a way that the mean-square error of the noise signal $v_1(n)$ and the output

Table 1: Parameters given for the generation of signals v_1 and v_2 .

$a_1(1)$	0.9
$b_1(0)$	1
$b_1(1)$	0.2
$a_2(1)$	0.95
$b_2(0)$	1
$b_2(1)$	0.3

of the filter $\hat{v}_1(n)$ is minimized. To estimate the filter coefficients \hat{w} , the following criterium is formulated:

$$\hat{w} = \arg \min_w E\{e(n)e^*(n)\}, \quad (5)$$

with

$$e(n) = v_1(n) - \hat{v}_1(n) = v_1(n) - \sum_{l=0}^{m-1} w(l)v_2(n-l), \quad (6)$$

The lower limit of the sum is 0, since the filter is causal.

To find \hat{w} for which Equation 5 holds it is sufficient to solve:

$$\frac{\delta}{\delta w^*(k)} E\{e(n)e^*(n)\} = E\{e(n) \frac{\delta e^*(n)}{\delta w^*(k)}\} = 0, \quad (7)$$

Where $k = 0, 1, \dots, m$. Using Equation 6, the differential in Equation 7 can be written as:

$$\frac{\delta e^*(n)}{\delta w^*(k)} = \frac{\delta}{\delta w^*(k)} \left(v_1^*(n) - \sum_{l=0}^{m-1} w^*(l)v_2^*(n-l) \right) = -v_2^*(n-k) \quad (8)$$

Substituting into Equation 7 and noting that the minus sign can be discarded because the expression is equal to zero, we get:

$$E\{e(n)v_2^*(n-k)\} = 0, \quad (9)$$

In this equation we identify $e(n)$ and use Equation 6 to get:

$$E\{v_1(n)v_2^*(n-k)\} - \sum_{l=0}^{m-1} w(l)E\{v_2(n-l)v_2^*(n-k)\} = 0 \quad (10)$$

Under the assumption that v_1 and v_2 are jointly WSS, $E\{v_2(n-l)v_2^*(n-k)\} = r_{v_2}(k-l)$ and $E\{v_1(n)v_2^*(n-k)\} = r_{v_1v_2}(k)$, so that

$$\sum_{l=0}^{m-1} w(l)r_{v_2}(k-l) = r_{v_1v_2}(k) \quad (11)$$

These equations are known as the *Wiener-Hopf equations*. We will now derive the expressions for $r_{v_2}(k)$ and $r_{v_1v_2}(k)$, so that \hat{w} can be determined.

Derivation of r_{v_2}

First we note that $r(-k)$ is $r(k)$, since v_1 and v_2 and their correlation functions are real. Since v_2 is generated by an ARMA(1,1) process (i.e. $a_2(k) = 0$ for $k > 1$), $r(-k)$ can be derived using the *Yule-Walker equations*:

$$\begin{bmatrix} r_{v_2}(0) & r_{v_2}(1) \\ r_{v_2}(1) & r_{v_2}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_2(1) \end{bmatrix} = \sigma_g^2 \begin{bmatrix} c_q(1) \\ c_q(2) \end{bmatrix}, \quad (12)$$

where $c_q(k)$ is given by:

$$c_q(k) = \sum_{l=0}^1 b_2(l+k)h(l) \quad (13)$$

By substitution of the second equation of Equation 12 into the first one, we derive the expression for $r_{v_2}(0)$ and $r_{v_2}(1)$:

$$r_{v_2}(0) = \frac{\sigma_g^2(c(1) - a_2(1)c(2))}{1 - a_2(1)^2}; \quad (14)$$

$$r_{v_2}(1) = \sigma_g^2 c(2) - a_2(1)r_{v_2}(0). \quad (15)$$

All subsequent values of $r_{v_2}(k)$ can be evaluated by using the following recursion formula:

$$r_{v_2}(k) = -a(1)r_{v_2}(k-1). \quad (16)$$

Derivation of $r_{v_1v_2}$

The power spectrum $P_{v_1v_2}$ can be written in terms of filter functions H_{v_i} :

$$P_{v_1v_2}(z) = H_{v_1}(z)H_{v_2}^{-1}(z)P_{v_2}(z) \quad (17)$$

The expressions for $H_{v_1}(z)$ and $H_{v_2}^{-1}(z)$ are given in Equation 3, and the product can be written as:

$$H_{v_1}(z)H_{v_2}^{-1}(z) = \frac{b_1(0) - (b_1(0)a_2(1) + b_1(1))z^{-1} + b_1(1)a_2(1)z^{-2}}{b_2(0) - (a_1(1)b_2(0) + b_2(1))z^{-1} + a_1(1)b_2(1)z^{-2}} \quad (18)$$

We can now take the inverse z -transform of Equation 17, to get an expression for $r_{v_1v_2}$:

$$r_{v_1v_2}(n) = b_1(0)r_{v_2}(n) - (b_1(0)a_2(1) + b_1(1))r_{v_2}(n-1) + b_1(1)a_2(1)r_{v_2}(n-2) \\ + (a_1(1)b_2(0) + b_2(1))r_{v_1v_2}(n-1) - a_1(1)b_2(1)r_{v_1v_2}(n-2)$$

Question 1.2

```

1 %% Initialization
2
3 close all
4 clear all
5
6 % Load uncorrupted signal d
7 % N is the number of samples (signal length)
8 load gong.mat;
9 [N,k]=size(y);
10 d = y;
11
12 % Generate zero-mean white noise sequence g with standard deviation 0.35
13 sg = 0.35;
14 g = sg*randn(N,1);
15 g = g - mean(g);
16
17 % Generate noise sequences v1 and v2
18 a1 = [1 -0.90]; b1 = [1 -.2];
19 a2 = [1 -0.95]; b2 = [1 -.3];
20 v1 = filter(b1,a1,g);
21 v2 = filter(b2,a2,g);
22
23 % Generate the corrupted signal x
24 x = d + v1;
25
26 %% Exercise 1: Determining the optimal FIR Wiener Filter
27 % Goal: reconstruct d from x and v2 by estimating v1 from v2
28
29 % Let n vary between the desired filter orders
30 n = [1 2 4 6];
31 Stdd = zeros(length(n),1);
32 W_tot = zeros(max(n),length(n));
33 for k = 1:length(n)
34 % First we determine Rv2 and Rv1v2 needed to set up the Wiener-Hopf
35
36 % Calculate the first two values of rv2 (i.e. rv2(0) and rv2(1))
37 % using eq (3.116) from Hayes
38 h = dimpulse(b2, a2, 20);
39 c(1,1) = b2(1)*conj(h(1)) + b2(2)*conj(h(2));
40 c(2,1) = b2(2)*conj(h(1));
41
42 rv2 = zeros(200,1);
43 rv2(1) = (sg^2*c(1,1) - a2(2)*sg^2*c(2,1))/(1-a2(2)^2);
44 rv2(2) = sg^2*c(2,1) - a2(2)*rv2(1);
45
46 % Calculate the rest of rv2 until it becomes (almost) zero
47 % We only determine one side of the auto-correlation function and then
48 % mirror, to get the double sided ACF (centered at index 200)
49 for i=3:200,
50     rv2(i) = -1*a2(2)*rv2(i-1);

```

```

51 end
52 rv2_ds = [rv2(end:-1:1); rv2(2:end)];
53
54 % Next we determine the the cross-correlation function between v1 and v2
55 bb = conv(b1,a2);
56 aa = conv(b2,a1);
57 rv1v2_ds = filter(bb,aa,rv2_ds);
58
59 % Put rv2 and rv1v2 into matrix form for the Wiener-Hopf equations
60 Rv2 = zeros(n(k),n(k));
61 for i = 1:n(k)
62     for j = 1:n(k)
63         Rv2(i,j) = rv2_ds(200+j-i);
64     end
65 end
66 Rv1v2 = zeros(n(k),1);
67 for i=1:n(k),
68     Rv1v2(i,1) = rv1v2_ds(200+i-1);
69 end
70
71 % Solve for the optimal filter
72 W = Rv2\Rv1v2;
73 vle = filter(W,1,v2);
74 de = x - vle;
75
76 % Save the necessary data necessary to evaluate the sound
77 W_tot(1:length(W),k) = W;
78 Std(k) = std(d-de);
79 end

```

Table 2: Output coefficients $w(j)$ for the optimal FIR Wiener filter

Coefficient $w(j)$	Filter order m			
	1	2	4	6
$w(0)$	0.7759	0.9209	0.9935	0.9995
$w(1)$	0	-0.1623	0.0327	0.0487
$w(2)$	0	0	-0.0765	-0.0242
$w(3)$	0	0	-0.2255	-0.0514
$w(4)$	0	0	0	-0.0859
$w(5)$	0	0	0	-0.1916

Question 2

Table 3: Standard deviation σ_W between the sound $d(n)$ and the estimated sound $x(n) - \hat{v}_1(n)$

Filter order m	σ_W
1	0.2075
2	0.1996
4	0.1674
6	0.1362

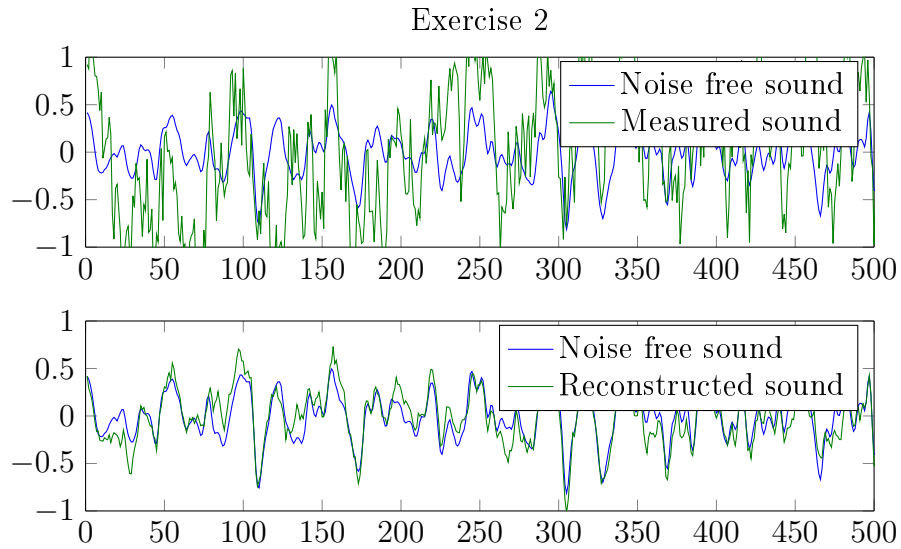


Figure 1: Subplot 1 shows the original sound and the noise corrupted signal. Subplot 2 shows the original sound and the filtered sound with optimal Wiener filter $W(z)$ of order 6

Question 3

```

1 %% Exercise 3: Filter by approximating the correlation functions
2 % Goal: Approximate the auto- and cross correlation functions used in
3 %       the Wiener-Hopf equations from the signals
4
5 % Let n vary between the desired filter orders
6 n = [1 2 4 6];
7 Stdd2 = zeros(length(n),1);
8 W_tot2 = zeros(max(n),length(n));
9 for k = 1:length(n)
10 % Construct an n by (N-n+1) matrix V2 containing shifted versions of v2
11 V2 = zeros(n(k), N-n(k)+1);
12 for i=1:n(k)
13     V2(i,:) = v2(n(k)+1-i:N+1-i);
14 end
15
16 % Use V2 and v1 to estimate Rv2 and Rv1v2
17 rv2e = zeros(n(k),1);
18 for i = 1:n(k)
19     rv2e(i) = sum(V2(1,:).*V2(i,:));
20 end
21
22 % Put rv2e in a Toeplitz matrix like in Exercise 2
23 rv2e_ds = [rv2e(end:-1:1);rv2e(2:end)];
24 Rv2e = zeros(n(k),n(k));
25 for i = 1:n(k)
26     for j = 1:n(k)
27         Rv2e(i,j) = rv2e_ds(n(k)+j-i);
28     end
29 end
30
31 Rv1v2e = zeros(n(k),1);
32 for i=1:n(k),
33     Rv1v2e(i,1) = sum(x(n(k):end).*(V2(i,:).'));
34 end
35
36 % Calculate filter using Wiener-Hopf equations and reconstruct the signal
37 w = Rv2e\Rv1v2e;
38 v1e = filter(w,1,v2);
39 de = x - v1e;
40
41 % Save all the variables for the different values of n
42 std(d-de);
43 Stdd2(k) = std(d-de);
44 W_tot2(1:length(w),k) = w;
45 end

```

Table 4: Output coefficients $w(j)$ for the estimated FIR Wiener filter

Coefficient $w(j)$	Filter order m			
	1	2	4	6
$w(0)$	0.7679	0.9200	0.9960	0.9995
$w(1)$	0	-0.1689	0.0339	0.0513
$w(2)$	0	0	-0.0783	-0.0243
$w(3)$	0	0	-0.2342	-0.0548
$w(4)$	0	0	0	-0.0885
$w(5)$	0	0	0	-0.1956

Table 5: Standard deviation σ_w between the sound $d(n)$ and the estimated sound $x(n) - \hat{v}_1(n)$

Filter order m	σ_w
1	0.2177
2	0.2089
4	0.1750
6	0.1423

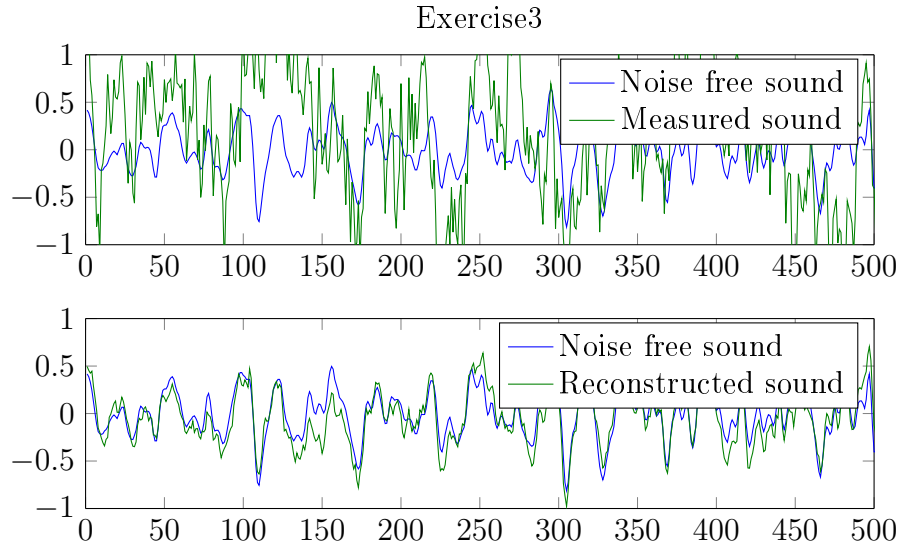


Figure 2: Subplot 1 shows the original sound and the noise corrupted signal. Subplot 2 shows the original sound and the filtered sound with estimated Wiener filter $w(z)$ of order 6

Question 4

Kristalhelder!