6.003 Lab 1

Nada Amin
namin@mit.edu

March 3, 2005

Basic Problems

(a)

The vector he is 1 at $n = 0$, $\alpha = 0.5$ at $n = N = 1000$ and null in between.

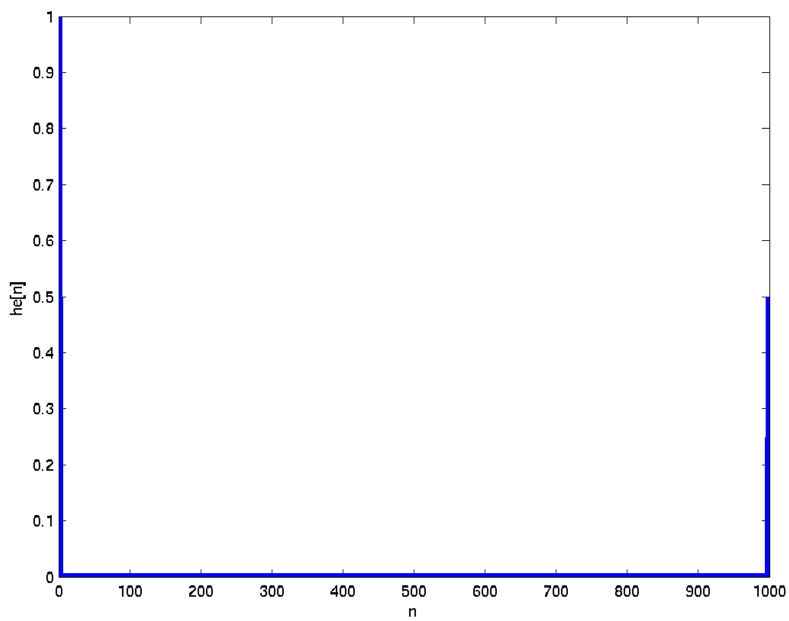

Figure 1: Plot of $he[n]$

(b)

(b.1) Impulse Response of the Echo System

$$y[n] = x[n] + \alpha x[n-N]$$

I compute the impulse response h_y, assuming the condition of initial rest.

$$\begin{aligned}
h_y[n] & && = 0 \text{ if } n < 0 \text{ (initial rest)} \\
h_y[0] &= \delta[0] + \alpha\delta[0-N] && = 1 \\
h_y[n] & && = 0 \text{ if } 0 < n < N \\
h_y[N] &= \delta[N] + \alpha\delta[N-N] && = \alpha \\
h_y[n] & && = 0 \text{ if } n > N
\end{aligned}$$

Thus, $h_y[n] = \delta[n] + \alpha\delta[n-N]$.

(b.2) Impulse Response of the Echo Removal System

$$z[n] + \alpha z[n-N] = y[n]$$
$$z[n] = y[n] - \alpha z[n-N]$$

I compute the impulse response h_z, assuming the condition of initial rest.

$$\begin{aligned}
h_z[n] & && = 0 && \text{if } n < 0 \text{ (initial rest)} \\
h_z[0] & = \delta[0] - \alpha h_z[0-N] && = 1 && \\
h_z[n] & && = 0 && \text{if } 0 < n < N \\
h_z[N] & = \delta[N] - \alpha h_z[N-N] && = -\alpha && \\
h_z[n] & && = 0 && \text{if } N < n < 2N \\
h_z[2N] & = \delta[2N] - \alpha h_z[2N-N] && = \alpha^2 && \\
h_z[n] & && = 0 && \text{if } n \neq kN \\
h_z[kN] & = \delta[kN] - \alpha h_z[kN-N] && = (-\alpha)^k &&
\end{aligned}$$

Thus, $h_z[n] = \sum_{k=0}^{\infty} (-\alpha)^k \delta[n-kN]$.

(b.3) Verifying Echo Removal System = Inverse of Echo System

I now verify that the echo removal system is the inverse of the echo system by showing that $h_y[n] * h_z[n] = \delta[n]$.

$$
\begin{aligned}
& h_y[n] * h_z[n] \\
=& (\delta[n] + \alpha\delta[n-N]) * (\sum_{k=0}^{\infty}(-\alpha)^k \delta[n-kN]) \\
=& (\delta[n] * \sum_{k=0}^{\infty}(-\alpha)^k \delta[n-kN]) + (\alpha\delta[n-N] * \sum_{k=0}^{\infty}(-\alpha)^k \delta[n-kN]) \\
=& \sum_{k=0}^{\infty}(-\alpha)^k \delta[n-kN] - \sum_{k=0}^{\infty}(-\alpha)^{k+1} \delta[n-(k+1)N] \\
=& \delta[n] + \sum_{k=1}^{\infty}(-\alpha)^k \delta[n-kN] - \sum_{k=1}^{\infty}(-\alpha)^k \delta[n-kN] \\
=& \delta[n]
\end{aligned}
$$

Intermediate Problems

(c)

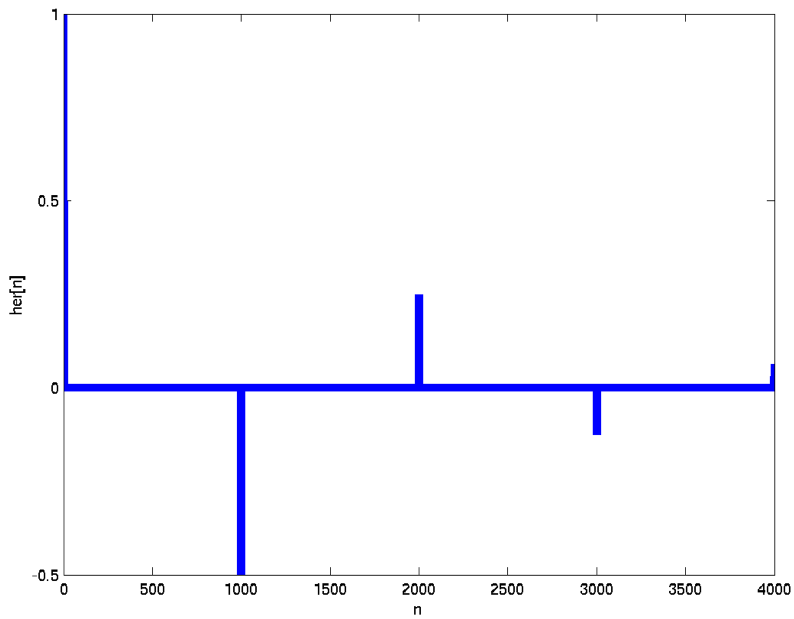

Figure 2: Plot of $her[n]$

(d)

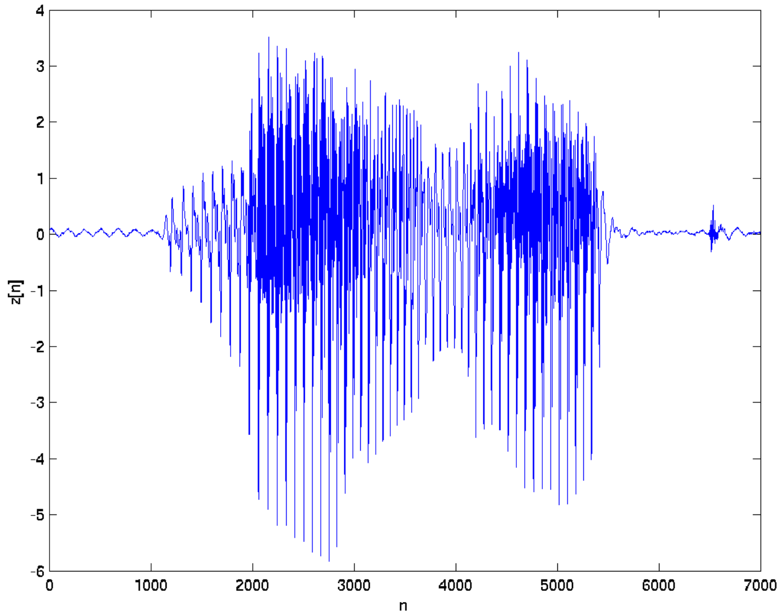

Figure 3: Plot of $z[n]$, the filter output of y

(e)

I convolve *her* with *he* and store the result in *hoa*. Though I computed *her* to be the inverse of *he*, the convolution of the two, *hoa*, is not a unit impulse because the actual impulse response of the inverse of *he* is infinite and *her* is only an approximation. More precisely, $h_z[n] = \sum_{k=0}^{\infty}(-\alpha)^k\delta[n-kN]$ but $her[n] = \sum_{k=0}^{4}(-\alpha)^k\delta[n-kN]$.

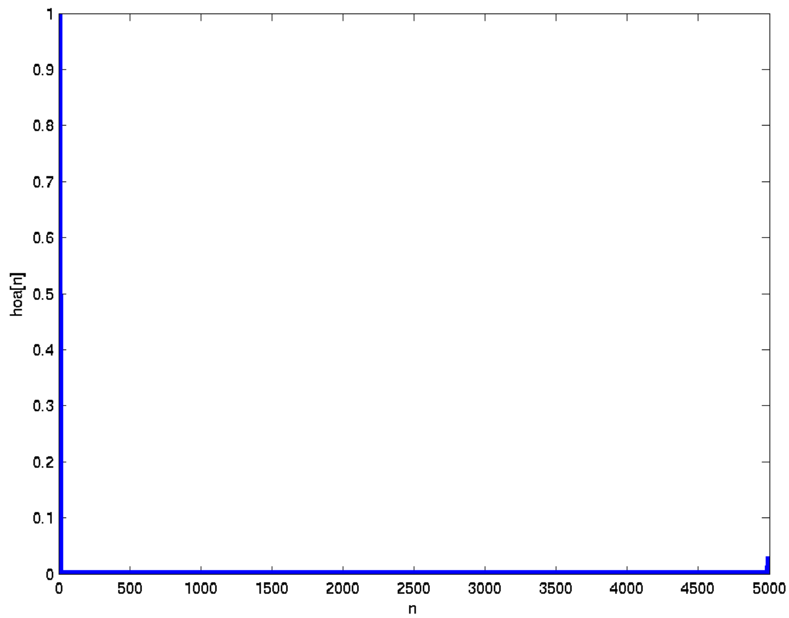

Figure 4: Plot of $hoa[n]$

Advanced Problems

(f)

The autocorrelation function $R_{ww}[n]$ of a signal $w[n]$ is defined as follows:

$$R_{ww}[n] = w[n] * w[-n]$$

I now write $R_{yy}[n]$ in terms of $R_{xx}[n]$:

$$\begin{aligned} R_{yy}[n] &= y[n] * y[-n] \\ &= (x[n] * h[n]) * (x[-n] * h[-n]) \\ &= (x[n] * x[-n]) * (h[n] * h[-n]) \\ &= R_{xx}[n] * R_{hh}[n] \end{aligned}$$

where $R_{hh}[n]$ is:

$$\begin{aligned} R_{hh}[n] &= h[n] * h[-n] \\ &= (\delta[n] + \alpha\delta[n-N]) * (\delta[-n] + \alpha\delta[-n-N]) \\ &= (\delta[n] + \alpha\delta[n-N]) * (\delta[n] + \alpha\delta[n+N]) \\ &= (1+\alpha)\delta[n] + \alpha(\delta[n-N] + \delta[n+N]) \end{aligned}$$

$$R_{yy}[n] = (1+\alpha)R_{xx}[n] + \alpha(R_{xx}[n-N] + R_{xx}[n+N])$$

Since I know that $R_{xx}[0]$ is a global maximum, i infer that $R_{yy}[n]$ has a big peak at $n = 0$ and two smaller peaks at $n = N$ and $n = -N$. By detecting these smaller peaks, I can estimate N. I use the difference equation $y[n] = x[n] - x[n+1] + x[n] - x[n+1]$ to detect those peaks.

Once I found N, α is trivial to find. I can use any value $n \geq N$ in the original difference equation for the system.

$$\begin{aligned} y[n] &= x[n] + \alpha x[n-N] \\ \alpha &= \frac{y[n] - x[n]}{x[n-N]} \qquad \text{for } n \geq N \end{aligned}$$

For $y3$, I needed to be a bit more ingenious because it contains two echoes, not just one. I find N_1, the delay of the echo with the largest α, by looking at how far the largest uncentered peak is from the centered peak in the auto-correlation signal $R_{y3y3}[n]$. I then estimate α_1 to be the approximated ratio of the found peak with the center peak of the auto-correlation signal $R_{xx}[n]$. Having found N_1 and α_1, I filter the largest echo from the output signal. I auto-correlate this filtered signal, and again look for the largest peak away from the center, which gives me N_2. I then use the original difference equation for the double echo system to find α_2.

$$\begin{aligned} y[n] &= x[n] + \alpha_1 x[n-N_1] + \alpha_2 x[n-N_2] \\ \alpha_2 &= \frac{y[n] - x[n] - \alpha_1 x[n-N_1]}{x[n-N_2]} \qquad \text{for } n \geq \max(N_1, N_2) \end{aligned}$$

For y, I find that $N = 1000$ and $\alpha = 0.5$, for $y2$, $N = 250$ and $\alpha = 0.7$, and for $y3$, $N_1 = 1500$, $\alpha_1 = 0.6$, $N_2 = 2500$, $\alpha_2 = 0.3$.

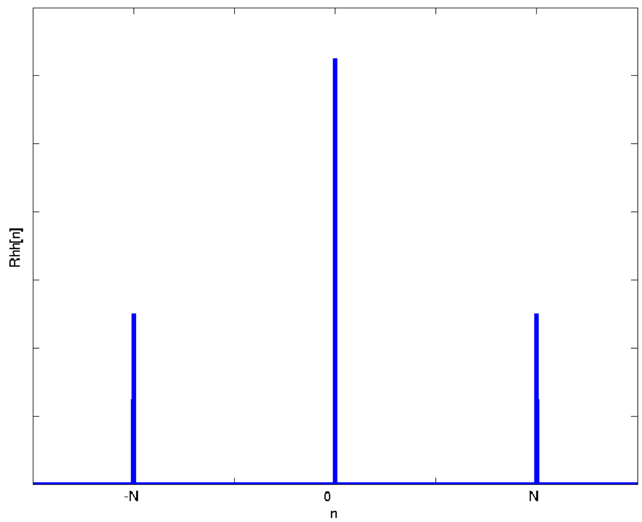

Figure 5: Plot of $R_{hh}[n]$

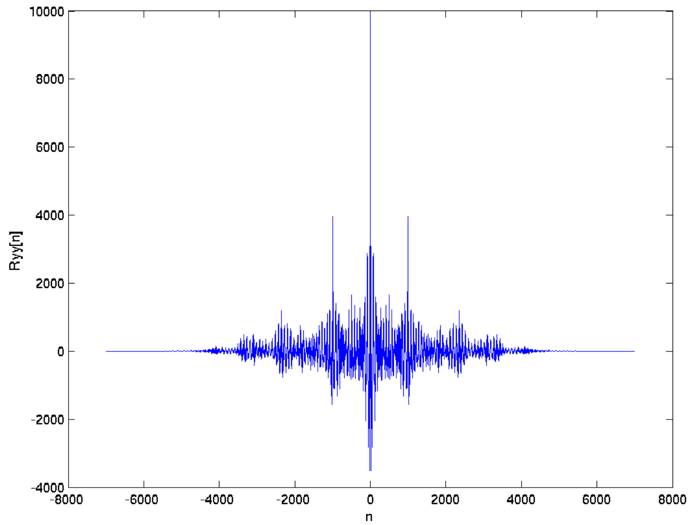

Figure 6: Plot of $R_{yy}[n]$

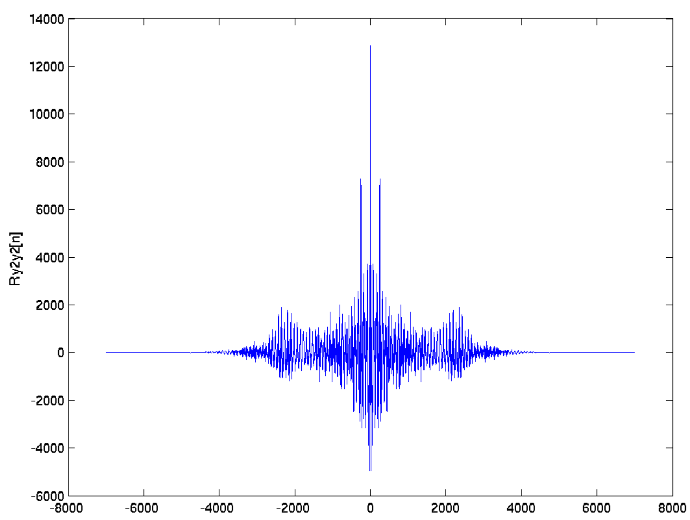

Figure 7: Plot of $R_{y2y2}[n]$

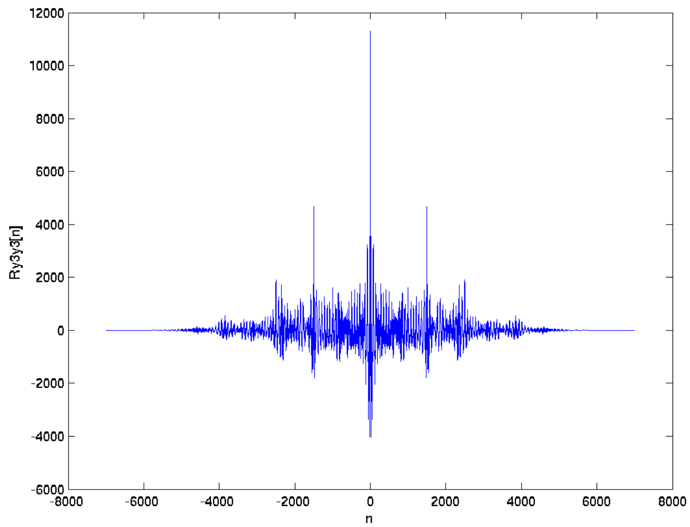

Figure 8: Plot of $R_{y3y3}[n]$

MATLAB Code

`lab1.m`

```
% Nada Amin (namin@mit.edu)
% 6.003 Lab 1
% Matlab Code
load lineup;
sound(y, 8192);
N = 1000;
alpha = 0.5;
% (a)
a = [1 zeros(1, N)];
b = [1 zeros(1, N-1) alpha];
d = [1 zeros(1, N)];
he = filter(b, a, d);
figure;
plot([0:N], he);
xlabel('n'); ylabel('he[n]');
% (b) analytical only
% (c)
her = filter(a, b, [1 zeros(1,4000)]);
figure;
plot([0:4000], her);
```

```
xlabel('n'); ylabel('her[n]');
% (d)
z = filter(a, b, y);
figure;
plot([0:length(z)-1],z);
xlabel('n'); ylabel('z[n]');
sound(z, 8192);
% (e)
hoa = conv(he, her);
figure;
plot([0:length(hoa)-1],hoa);
xlabel('n'); ylabel('hoa[n]');
% (f)
[Ny1, Ryy] = detect1(y);
half = (length(Ryy)-1)/2;
figure;
plot([-half:half], Ryy);
xlabel('n'); ylabel('Ryy[n]');
ay1 = alpha1(z, y, Ny1);

[Ny2, Ry2y2] = detect1(y2);
half = (length(Ry2y2)-1)/2;
figure;
plot([-half:half], Ry2y2);
xlabel('n'); ylabel('Ry2y2[n]');
ay2 = alpha1(z, y2, Ny2);

% y3 is composed of two echoes
% will process it in two passes
% **pass 1**
[N1, Ry3y3] = detect1(y3);
half = (length(Ry3y3)-1)/2;
figure;
plot([-half:half], Ry3y3);
xlabel('n'); ylabel('Ry3y3[n]');
% approximate alpha to be the ratio of the current little peak with the
% original center peak
ci = half+1;
Rxx = conv(z, flipud(z));
a1 = Ry3y3(ci-N1)/Rxx(ci);
% round alpha to the nearest decimal
a1 = round(a1*10)/10;
% z1 is y3 filtered for this echo
z1 = filter([1 zeros(1, N1)], [1 zeros(1, N1-1) a1], y3);
% **pass 2**
[N2, Rz1z1] = detect1(z1);
```

```
half = (length(Rz1z1)-1)/2;
figure;
plot([-half:half], Rz1z1);
xlabel('n'); ylabel('Rz1z1[n]');
L = length(z);
if mod(L,2)==0
    c = L/2;
else
    c = (L+1)/2;
end
n = c+max(N2,N1)+1;
a2 = (y3(n)-z(n)-a1*z(n-N1))/z(n-N2);

% sanity checks
zf2 = filter([1 zeros(1, Ny2)], [1 zeros(1, Ny2-1) ay2], y2);
dzf2 = sum(abs(z-zf2)); % must be very small

if N1 < N2
    minN = N1;
    maxN = N2;
    mina = a1;
    maxa = a2;
else
    minN = N2;
    maxN = N1;
    mina = a2;
    maxa = a1;
end

as = [1 zeros(1, maxN)];
bs = [1 zeros(1, minN-1) mina zeros(1, maxN-minN-1) maxa];
zf3 = filter(as, bs, y3);
dzf3 = sum(abs(z-zf3)); % must be very small
```

`detect1.m`

```
function [ N, Ryy ] = detect1( y )
%DETECT1 Given the output signal y of the echo system returns the delay N
%in time of the echo and the auto-correlation function for y.
%@requires y is a column-vector
Ryy = conv(y, flipud(y));
L = length(Ryy);
% y[n] = x[n]-x[n-1] + x[n]-x[n+1] = -x[n+1] + 2x[n] - x[n-1]
% y[n-1] = -x[n] + 2x[n-1] - x[n-2]
dRs = filter([-1 2 -1], 1, Ryy);
dR = dRs([2:L]);
```

```
[cm, ci] = max(dR);
% the greatest peak should be at the center
if ci ~= (L+1)/2
    ci
    error('DETECT1: the greatest peak is not at the center');
end
% find the next greatest peak, once we discard values close to the center peak
[nm, ni] = max(dR([1:ci-200]));
N = ci-ni;
```

alpha1.m

```
function [ alpha ] = alpha1( x, y, N, n )
%ALPHA1 Given an input signal x, an output signal y and the delay of the
%echo N, returns alpha, the magnitude of the echo.
L = length(x);
if mod(L,2)==0
    c = L/2;
else
    c = (L+1)/2;
end
n = c+N+1;
alpha = (y(n)-x(n))/x(n-N);
```