MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
2.161 Signal Processing - Continuous and Discrete

MATLAB Examples of Least-Squares FIR Filter Design [1]

1 Least-Squares Filter Design

Given two sets of data (1) $\{f_n\}$, and (2) the "desired" data set $\{d_n\}$, the coefficients of the length M least-squares FIR filter are given by

$$\mathbf{b} = \mathbf{R}^{-1}\mathbf{B}$$

where $\mathbf{b}$ are the FIR filter coefficients,

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{M-1} \end{bmatrix},$$

$\mathbf{R}$ is the Toeplitz correlation matrix

$$\mathbf{R} = \begin{bmatrix} \phi_{ff}(0) & \phi_{ff}(1) & \phi_{ff}(2) & \cdots & \phi_{ff}(M-1) \\ \phi_{ff}(1) & \phi_{ff}(0) & \phi_{ff}(1) & \cdots & \phi_{ff}(M-2) \\ \phi_{ff}(2) & \phi_{ff}(1) & \phi_{ff}(0) & \cdots & \phi_{ff}(M-3) \\ \vdots & \vdots & \vdots & & \vdots \\ \phi_{ff}(M-1) & \phi_{ff}(M-2) & \phi_{ff}(M-3) & \cdots & \phi_{ff}(0) \end{bmatrix},$$

and P is a vector of cross-correlation terms

$$\mathbf{P} = \begin{bmatrix} \phi_{fd}(0) \\ \phi_{fd}(1) \\ \phi_{fdf}(2) \\ \vdots \\ \phi_{fd}(M-1) \end{bmatrix}$$

The minimum mean-squared error is

$$(\mathrm{MSE})_{\mathrm{min}} = \phi_{dd}(0) - \mathbf{P}^T\mathbf{b}$$

On the next page is a simple tutorial example of a MATLAB function LSQfilt() to compute the optimum filter coefficients:

[1] D. Rowell December 4, 2008

```
%-----------------------------------------------------------------------
function [B,MSE] = LSQFilt(f,d,M)
%LSQFilt - Demonstration routine for Least-Squares FIR filter design
%[B,MSE] = LSQFilt(f,d,M)
%       f - rowvector of data samples    - length N
%       d - row vector of desired values - length N
%       M - filter order
% Returns:
%       B -   vector of optimal filter coefficients
%       MSE - minimized value of the mean-square-error
%
% Note: This routine is for tutorial purposes only. The Levinson method for
%       toeplitz matrix inversion would be used in practical methods.
%
% Author:  D. Rowell
% Revised: 10/29/07

     N = length(f);
% Compute the correlation coefficients.
% Note that matlab defines the cross-correlaton backwards!! and
% we need to reverse the order of the subscripts.
%
     phiff=xcorr(f);
     phifd=xcorr(d,f);
%
% Extract out the central regions (low-lag values) and form
% the autocorrelation matrix.
%
     rff=phiff(N:N+M-1);
     R = toeplitz(rff);
     P=phifd(N:N+M-1);
%
% Compute the optimal filter coefficients
%
     B=inv(R)*P';
%
% and the residual mean-square-error
%
     phidd=xcorr(d);
     MSE=phidd(N) - P*B;
%
%-----------------------------------------------------------------------
```

2 Exaxmples

2.1 A one-step Linear Predictor for a Sinusoidal Input

Stearns and Hush (p. 346) solve the problem of a one-step linear predictor for an input function0f the form

$$s_n = \sin\left(\frac{2\pi n}{12}\right), \qquad n = 0, 1, 2, \ldots$$

and show that $\mathbf{b} = \left[\sqrt(3) \quad 1\right]^T$.

The following MATLAB code

```
% This is the example done in Stearns and Hush, page 346.
% One-step linear predictor for a sinusoidal inpur.
% Define a time vector, and the input vector:
t = 0:199;
s = sin(2*pi*t/12);
% In this case the desired output is the input
d = s;
% To make it causal we must delay the input to the filter
f = zeros(1,200);
f(2:200) = s(1:199);
% Compute the filter coefficients for a first-order filter
[B1,MSE] = LSQFilt(f,d,2)
% Repeat with a second-order model
[B2,MSE] = LSQFilt(f,d,3)
```

produces the resuts

$$B1 = \begin{bmatrix} 1.73205080756888e+000 \\ -1.00000000000000e+000 \end{bmatrix}$$

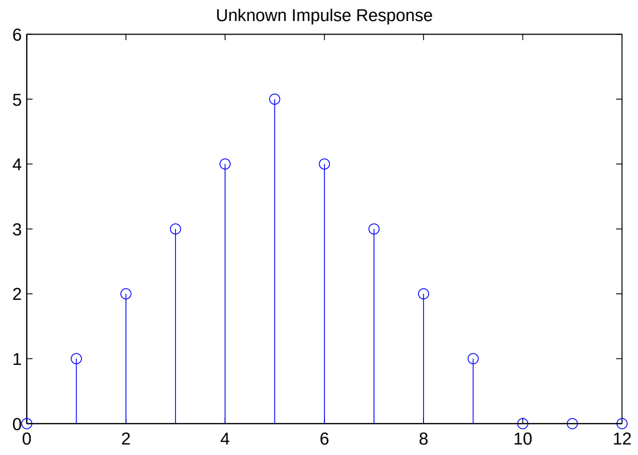
$$B2 = \begin{bmatrix} 1.73205080756897e+000 \\ -1.00000000000016e+000 \\ 112.354570092066e-015 \end{bmatrix}$$

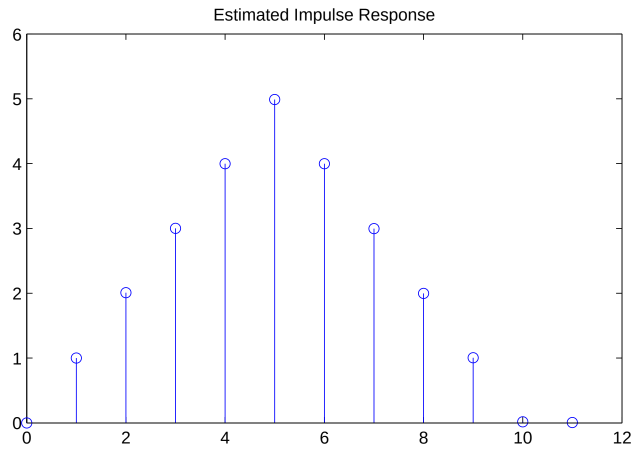
2.2 System Identification

In this example we create an "unknown" plant as a linear FIR system with a given impulse response. We then excite the model with white noise and record the input and output series, The least-squares estimator is then used to determine the coefficients of the filter that best matches the plant, and these coefficients then estimate the plant impulse response (the coefficients of an FIR filter are its impulse response).

```
% System ID Using LSQFilt
% Create a FIR filter as the "unknown" plant
h = [0 1 2 3 4 5 4 3 2 1 0 0 0];
%
f = randn(1,1000);
% create output data representing the exerimental
% measurements
y = filter(h,1,f);
%
% Estimate the impulse response from the data
[h_opt,MSE] = LSQFilt(f,y,15);
figure(1); stem(0:length(h)-1,h);         title('Unknown Impulse Response');
figure(2); stem(0:length(h)-1,h_opt(1:length(h))); title('Estimated Impulse Response');
```

giving the following plots:





2.3 Channel Compensation

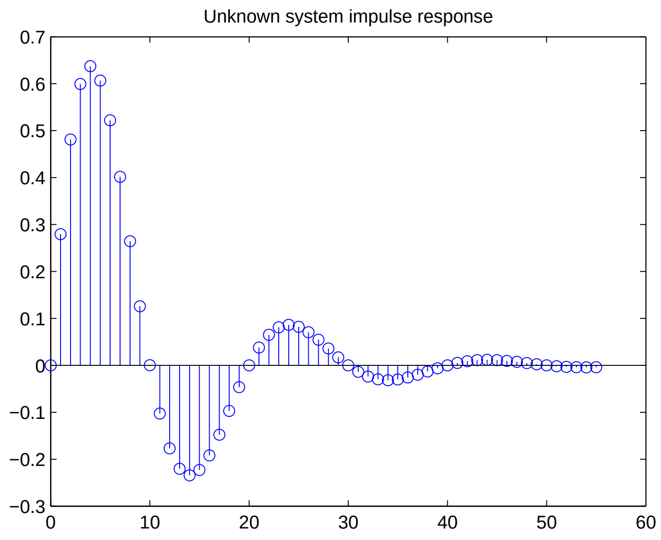
Suppose a waveform has been “corrupted” by passing through an unknown LTI filter $H(z)$, and it is desired to recover the waveform by using a cascade inverse filter $H'(z) = 1/H(z)$. The following uses `LSQFilt()` with white noise as the input to an “unknown” recursive filter

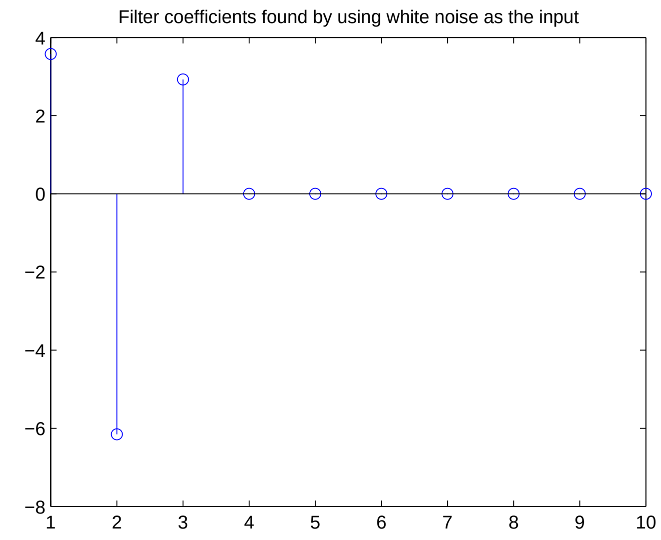
$$H(z) = \frac{0.296}{1 - 1.721z^{-1} + 0.8187Z^{-2}}$$

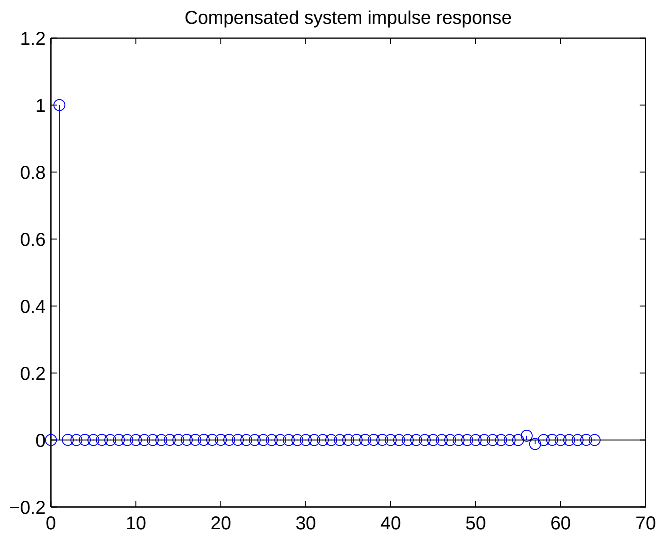
to directly estimate the coefficients of the FIR inverse filter.

```
% Channel compensation using Least-Squares filter design.
% Input the numerator B(z), and denominator A(z) of the "unknown" plant
B_u = [0.2796 0];
A_u = [1 -1.7211 0.8187];
unknown = tf(B_u, A_u ,1);
h_unknown = impulse(unknown);
figure(1), stem(0:length(h_unknown)-1, h_unknown)
title('Unknown system impulse response');
%
% Use white noise as the input signal:
s = randn(1,200);
f = lsim(unknown,s);
% Create the desired output (delay by one step so that resulting filter is causal)
d = zeros(1,200);
d(2:200) = s(1:199);
[B_noise,MSE] = LSQFilt(f,d,10);
figure(2), stem(B)
title('Filter coefficients found by using white noise as the input')
h_compensated = conv(h_unknown, B);
figure(3), stem(0:length(h_compensated)-1, h_compensated)
title('Compensated system impulse response')
```

The output is shown on the next page.


Unknown system impulse response
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0
10
20
30
40
50
60

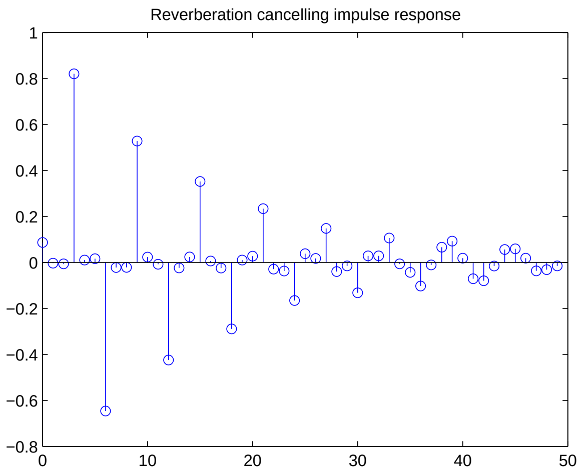

Filter coefficients found by using white noise as the input
4
2
0
−2
−4
−6
−8
1
2
3
4
5
6
7
8
9
10

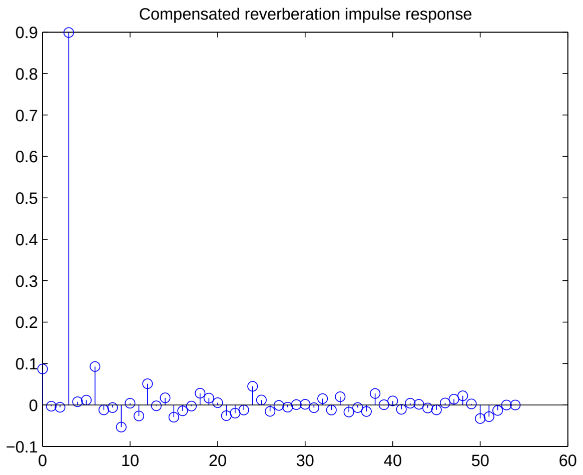

Compensated system impulse response
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
0
10
20
30
40
50
60
70

2.4 Reverberation suppression

The following is another example of channel compensation, this time where the channel is corrupted by an echo. For clarity we use a simple "strong" echo, so that the waveform is $g_n = f_n + 0.9 f_{n-3}$. White noise is used as the excitation. Notice that complete suppression is not possible with a finite length FIR filter.

```
% Model the environment as a nonrecursive filter:
b = [1 0 0 0.9 0 0];
% Use white noise as the excitation
s = randn(1,200);
f = filter(b,1,s);
% The desired output is a delayed version of the input:
d = zeros(1,200); d(4:200) = s(1:197);
% Design the filter
[B,MSE] = LSQFilt(f,d,50);
figure(1), stem(0:length(B)-1, B)
title('Reverberation cancelling impulse response')
% Find the overall compensated system impulse response
h_comp = conv(B,b);
figure(2), stem(0:length(h_comp)-1, h_comp)
title('Compensated reverberation impulse response')
```


Reverberation cancelling impulse response


Compensated reverberation impulse response