EE779 : Advanced Topics in Signal Processing

Table of Contents

Navjot Singh (130110071)

```
clear all
close all
load('data/blocks_deconv.mat');
```

(a) Finding the convolution matrix A

```
function [A] = findConvolutionMatrix(h,x)
    N = length(x);
    L = length(h);
    M = N + L -1;
    A = zeros(M,N);
    for i = 1:M
        for j = max(1,i-length(h)+1):min(i,N)
                A(i,j) = h(i-j+1);
        end
    end
end
```

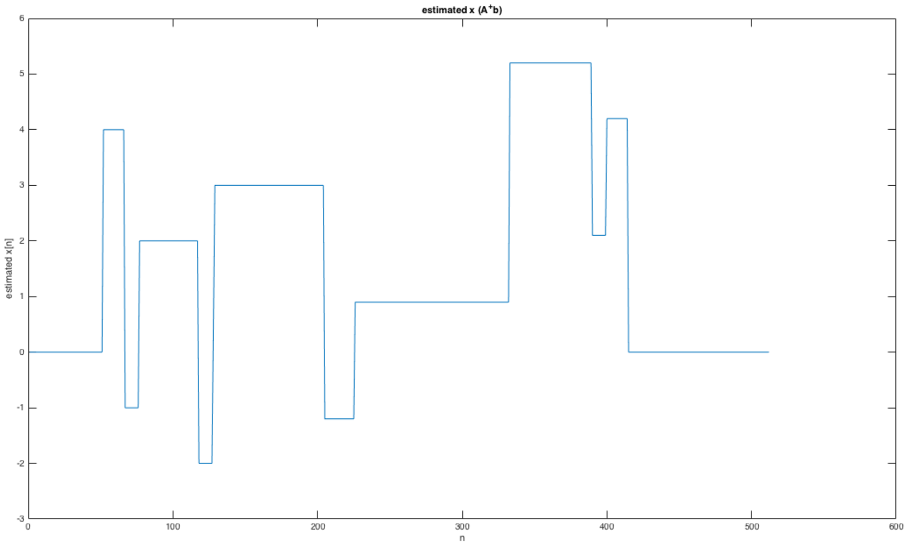
```
N = length(x);
L = length(h);
M = length(y);
A = findConvolutionMatrix(h,x);
```

(b) SVD of A

```
[U,S,V] = svd(A);
largest_singular_value = S(1,1)
smallest_singlar_value = S(rank(A),rank(A))
p = rank(A);
```

```
U_new = U(:,1:p);
S_new = S(1:p,1:p);
V_new = V(1:p,1:p);
A_dagger = V_new*(inv(S_new))*U_new';
x_est = A_dagger*y;

fig = figure;
plot(x_est);
ylabel('estimated x[n]')
title('estimated x (A^+b)');
xlabel('n')
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/q4b_x_estimated.jpg','jpg');
```

```
largest_singular_value =

    0.9987


smallest_singlar_value =

    0.0029
```



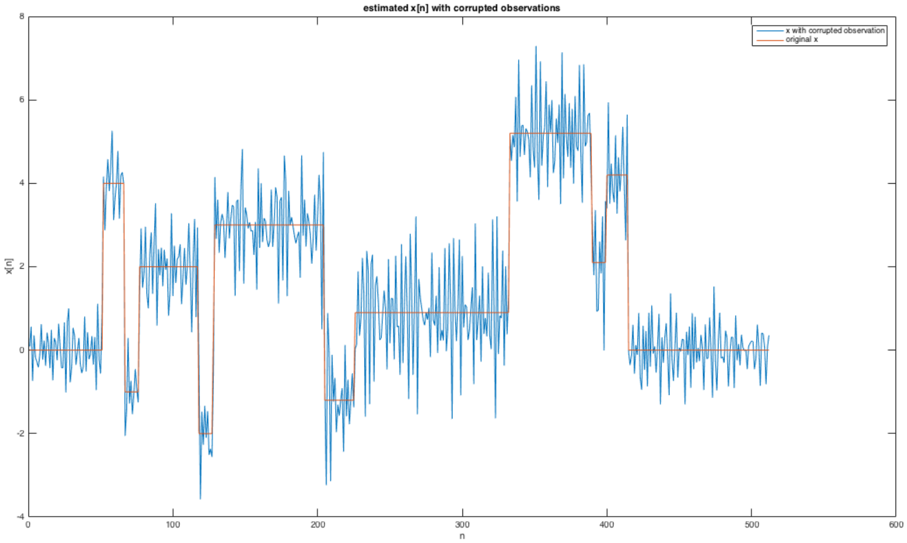
Since we are using uncorrupted version of y and all the singular values of to create A^+ (pseudo inverse of A), hence x will be perfectly reconstructed.

(c) Apply Pseudoinverse to noisy output yn

```
x_est_noisy = A_dagger*yn;

fig = figure;
```

```
plot([x_est_noisy,x]);
x_svd_all = x_est_noisy;
ylabel('x[n]')
title('estimated x[n] with corrupted observations');
xlabel('n')
legend('x with corrupted observation','original x');
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/q4c_x_estimated_corrupted_yn.jpg','jpg');

mse_x = mean((abs(x-x_est_noisy)).^2);
mse_y = mean((abs(y-yn)).^2);
mse_x_svd_all = mse_x;
mse_x
mse_y
```

```
mse_x =

    0.7369


mse_y =

   1.0680e-04
```



We have used currupted version of observations for reconstruction hence we are not able to reconstruct x accurately. Also it is observed that Mean Square Error of estimated x is much greater than that of y.

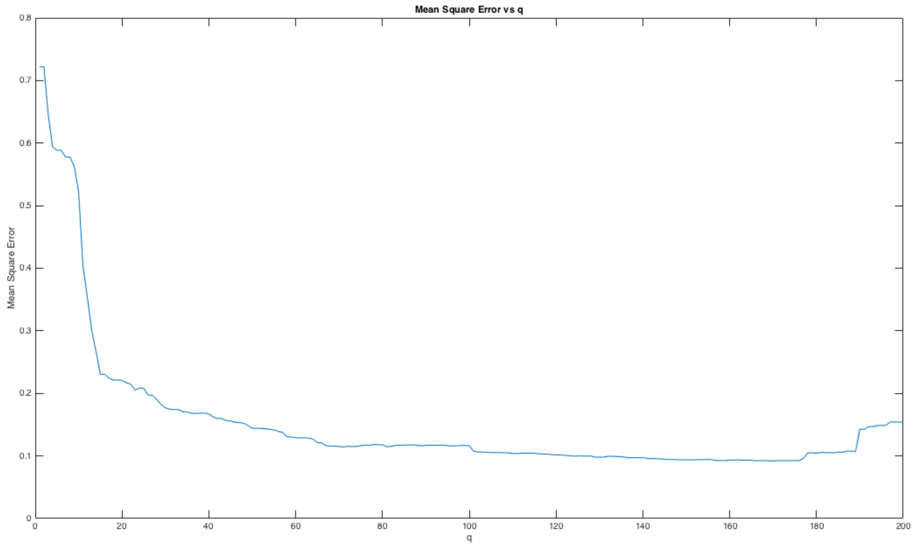
(d) Truncated SVD

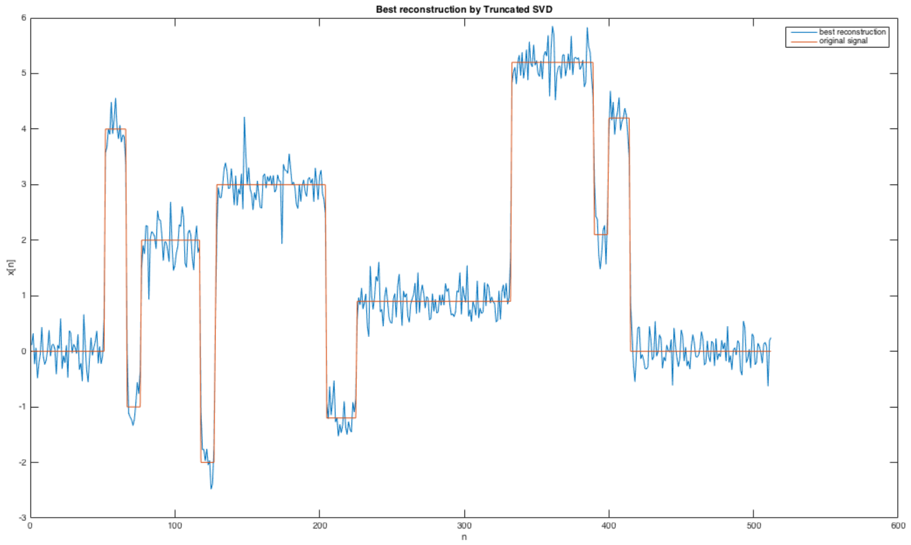
We try with different values of q i.e. we neglect last q singular values and try to reconstruct signal from remaining singular values.

```
function
 [mse_x_list,q_list,mse_x_min,x_svd_best,mse_x_svd_best,q_best] =
 q4d_truncateSVD(M,N,A,S,U,V,x,yn)

% Calculating from 200 possible values of q
q_list = (1:200);
mse_x_list = zeros(size(q_list));
mse_x_min = Inf;

for j = 1:length(q_list)
    A_trunc = zeros(M,N);
    A_dagger_trunc = zeros(N,M);
    p = rank(A);
    q = q_list(j);
    for k = 1:p-q
        A_trunc = A_trunc + S(k,k)*U(:,k)*V(:,k)';
    end
    for k = 1:p-q
        A_dagger_trunc = A_dagger_trunc + (1/S(k,k))*V(:,k)*U(:,k)';
    end
    x_est_noisy = A_dagger_trunc*yn;
    mse_x = mean((abs(x-x_est_noisy)).^2);
    mse_x_list(j) = mse_x;

    if(mse_x < mse_x_min)
        mse_x_min = mse_x;
        x_svd_best = x_est_noisy;
        q_best = q;
        mse_x_svd_best = mse_x;
    end
end


[mse_x_list,q_list,mse_x_min,x_svd_best,mse_x_svd_best,q_best] =
 q4d_truncateSVD(M,N,A,S,U,V,x,yn);

q_best

fig = figure;
plot(q_list,mse_x_list);
ylabel('Mean Square Error')
title('Mean Square Error vs q');
xlabel('q')
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/q4d_Mean_Square_Error_vs_q.jpg','jpg');

fig = figure;
plot([x_svd_best,x]);
ylabel('x[n]')
title('Best reconstruction by Truncated SVD');
xlabel('n');
legend('best reconstruction','original signal')
set(gcf, 'Position', get(0, 'Screensize'));
```

```
saveas(fig,'results/q4d_best_reconstruction_truncated_SVD.jpg','jpg');
```

```
q_best =

   170
```





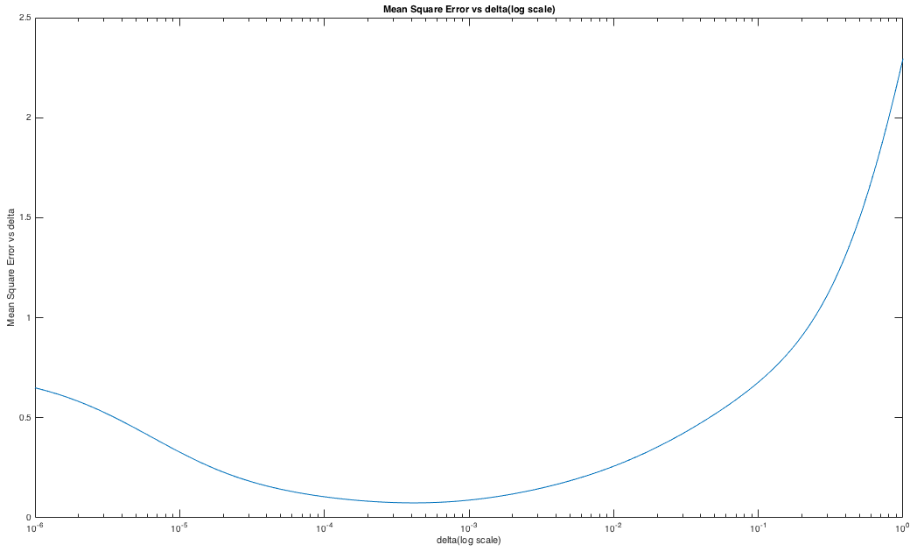
It was observed that initially reconstruction error decreases and then increases again after an optimal point. The best value of q was judged based on reconstruction error.

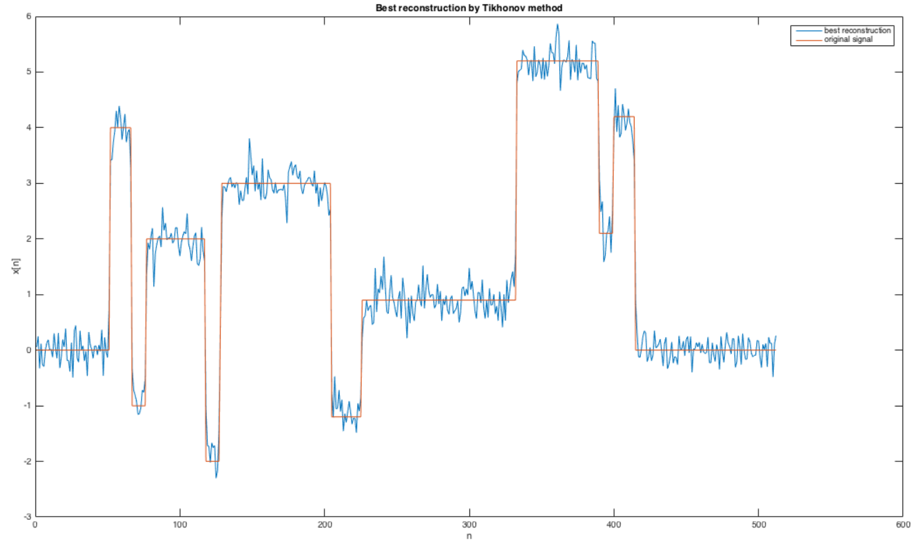
(e) Tikhonov regularization

We will vary delta in log space (because linear space will take lot of time to reach the optimum value).

```
function
 [mse_x_list,delta_list,mse_x_min,x_tikhonov_best,mse_x_tikhonov_best,delta_best]
 = q4e_tikhonov(A,S,U,V,x,yn)


I = eye(size(A'*A));
delta_list = logspace(-6,0,1000);
mse_x_list = zeros(size(delta_list));
mse_x_min = Inf;
for j = 1:length(delta_list)
    delta = delta_list(j);
    x_tikhonov_est = (inv(A'*A+delta*I))*A'*yn;
    mse_x = mean((abs(x-x_tikhonov_est)).^2);
    mse_x_list(j) = mse_x;
    if(mse_x < mse_x_min)
        mse_x_min = mse_x;
        x_tikhonov_best = x_tikhonov_est;
        delta_best = delta;
        mse_x_tikhonov_best = mse_x;
    end
end


[mse_x_list,delta_list,mse_x_min,x_tikhonov_best,mse_x_tikhonov_best,delta_best]
 = q4e_tikhonov(A,S,U,V,x,yn);

delta_best

fig = figure;
semilogx(delta_list,mse_x_list);
ylabel('Mean Square Error vs delta')
title('Mean Square Error vs delta(log scale)');
xlabel('delta(log scale)')
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/q4e_Mean_Square_Error_vs_delta(log_scale)','jpg');

fig = figure;
plot([x_tikhonov_best,x]);
ylabel('x[n]')
title('Best reconstruction by Tikhonov method');
xlabel('n');
legend('best reconstruction','original signal')
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/
q4e_best_reconstruction_tikhonov_method.jpg','jpg');
```

```
delta_best =

   4.1555e-04
```


Mean Square Error vs delta(log scale)


Best reconstruction by Tikhonov method

It is observed that initially delta reconstruction error decreases as delta increases and reaches an optimal value. The reconstruction error starts increasing again. Optimal value of delta is used.

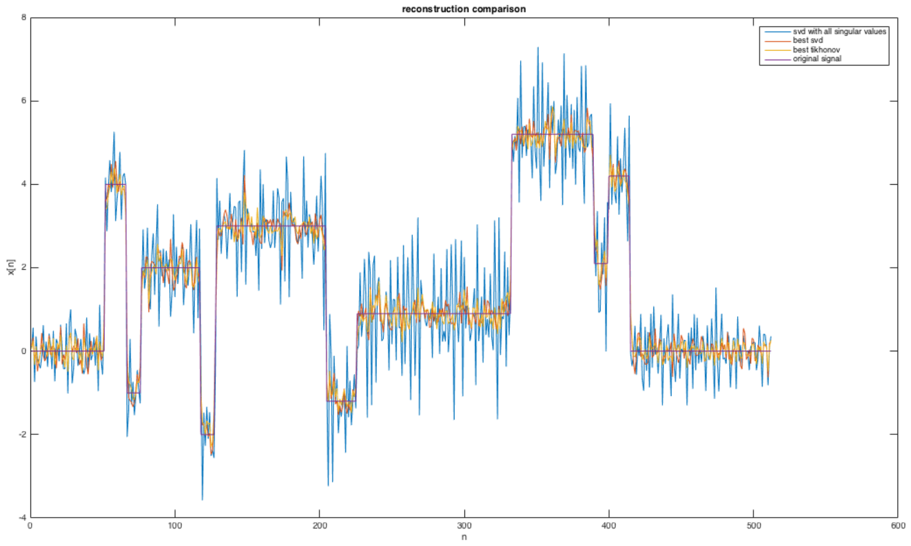
(f) Summarizing Results

- It is observed that chosing optimal q improves performance of SVD reconstruction method compared to taking all singular values blindly.
- Tichonov method performs better than optimal q svd method if optimal delta is chosen.
- Overall performance, Tichonov > Optimal q SVD > All SVD

```
fig = figure;
plot([x_svd_all,x_svd_best,x_tikhonov_best,x]);
```

```
ylabel('x[n]');
title('reconstruction comparison');
xlabel('n');
legend('svd with all singular values','best svd','best
 tikhonov','original signal');
set(gcf, 'Position', get(0, 'Screensize'));
saveas(fig,'results/q4f_comparison_reconstruction.jpg','jpg');

mse_x_svd_all
mse_x_svd_best
mse_x_tikhonov_best
```

```
mse_x_svd_all =

    0.7369


mse_x_svd_best =

    0.0922


mse_x_tikhonov_best =

    0.0746
```



Published with MATLAB® R2015b