```
% Michael Bentivegna, Simon Yoon, Joya Debi
% ECE302 Stochastic Processes Project 5: MMSE FIR

clear;
clc;
close all;

% Summary of Project

% The goal is to design a filter to estimate a signal (target process) and
% achieve an MMSE estimate. Using Wiener filtering formulae, we derived
% theoretical valeus for correlations to be made applicable in the simulation
% of the MMSE estimation. We solve for the impulse response values via the
% linear equations of FIR filter length N. We solve for these normal
% equations for LMMSE.


% length of filter h
N = [4 6 10];

% sigma^2
sigsq = 0.5;

% M is number of samples, process len
M = 1000;

% random vectors
s = 2*randi(2, [M 1]) - 3;
d = sqrt(sigsq) * randn([M 1]);

% need to pad c so that convolution works at Rrr,Rsr[0] index
c = [0 0 1 .2 .4];
r = conv(s, c, 'same') + d;

for m = 1:length(N)

 % normal equations

    % Rrr autocorrelation
 R_rr = zeros([N(m), 1]);
 R_rr(1:3) = [1.2+sigsq .28 .4];
    % Rrr = xcorr(r);

    % Rsr
 R_sr = zeros([N(m), 1]);
 R_sr(1:3) = [1 .2 .4].';
    % Rsr = xcorr(s,r);

 R = R_rr(abs((1:N(m)) - (1:N(m)).') + 1);
    % Ra = Rrr(abs((1:N(m)) - (1:N(m)).') + 1);

 % solve for h
```

```
    % h = inv(R)*R_sr
 h = R \ R_sr(1:N(m));
    % hA = Ra \ Rsr(1:N(m));

 % need to pad h so that it's correctly centered at middle
 h = [zeros([N(m)-1 1]); h];
    % hA = [zeros([N(m)-1 1]); hA];

 % calculate estimate with our filter
 s_hat = conv(r, h, 'same');
    % s_hatA = conv(r, hA , 'same');

 % calculate and print mse
 mse = mean((s-s_hat).^2);
    % mseA = mean((s-s_hatA).^2);
 fprintf('Theoretical: N=%d: MSE=%f\n', N(m), mse);
    % fprintf('Via xcorr: N=%d: MSE=%f\n', N(m), mseA);
end

% The MSE is low, but not much better than random guessing at 0.5. Changing
% the sigma squared significantly changes performance, but N is too small
% for minute changes to be observed.
```

```
Theoretical: N=4: MSE=0.474677
Theoretical: N=6: MSE=0.476575
Theoretical: N=10: MSE=0.476370
```

Published with MATLAB® R2022a