Contents

Scenario 2

```
% Raymond/Lei Chi and Arav Sharma
```

Setup

```
clc
close all
clear

samples = 10000;
% sets the number of random samples for Y and R

uY = 1;
N = 7;
% Number of observations

variancesY = [10, 50, 500, 2500];
variancesR = [5, 10, 20, 40];

SNR = variancesY./variancesR;
% compute signal-to-noise ratios

disp('Y Variances:')
disp(variancesY)
disp('R Variances:')
disp(variancesR)
disp('SNRs:')
disp(SNR)

sigmasY = sqrt(variancesY);
sigmasR = sqrt(variancesR);
% compute std deviations from variances

theoVals = zeros(size(variancesY));
expVals = zeros(size(variancesY));
% preallocate memory for simulation
```

```
Y Variances:
          10          50         500        2500

R Variances:
     5    10    20    40

SNRs:
```

```
    2.0000    5.0000   25.0000   62.5000
```

Simulation

```
for i = 1:length(variancesY)

    varY = variancesY(i);
    varR = variancesR(i);
    % choose corresponding variances from list

    sigY = sigmasY(i);
    sigR = sigmasR(i);
    % choose corresponding std deviations from list

    theoVals(i) = getMse(varY, varR, N);
    % calculate theoretical values for given variances

    a = computeA(varY, varR, N);
    % calculate coefficient column vector

    a0 = uY * (1 - sum(a));
    %calculate a0 (pg 159 eq 8.74)

    Y = repmat(normrnd(uY, sigY, [1, samples]), N, 1);
    % generates random vals for Y for given sigma
    % duplicates these rows to feed to each individual
    % noisy channel

    R = normrnd(0, sigR, [N, samples]);
    % generates independent noise for each individual
    % channel for given sigma (each row is R_n)

    X = Y + R;
    % pg 159 (8.75)

    Xsum = sum(X);
    c2 = 1/(N*varY + varR);
    Yhat = c2*((varR*uY) + (varY*Xsum));
    % calculate Yhat in 3 steps
    % pg 160 (8.79)

    expVals(i) = mean((Y(1,:) - Yhat).^2);
    % save error from simulation
    % E[(Y - Yhat)^2]

end
```

Output Results

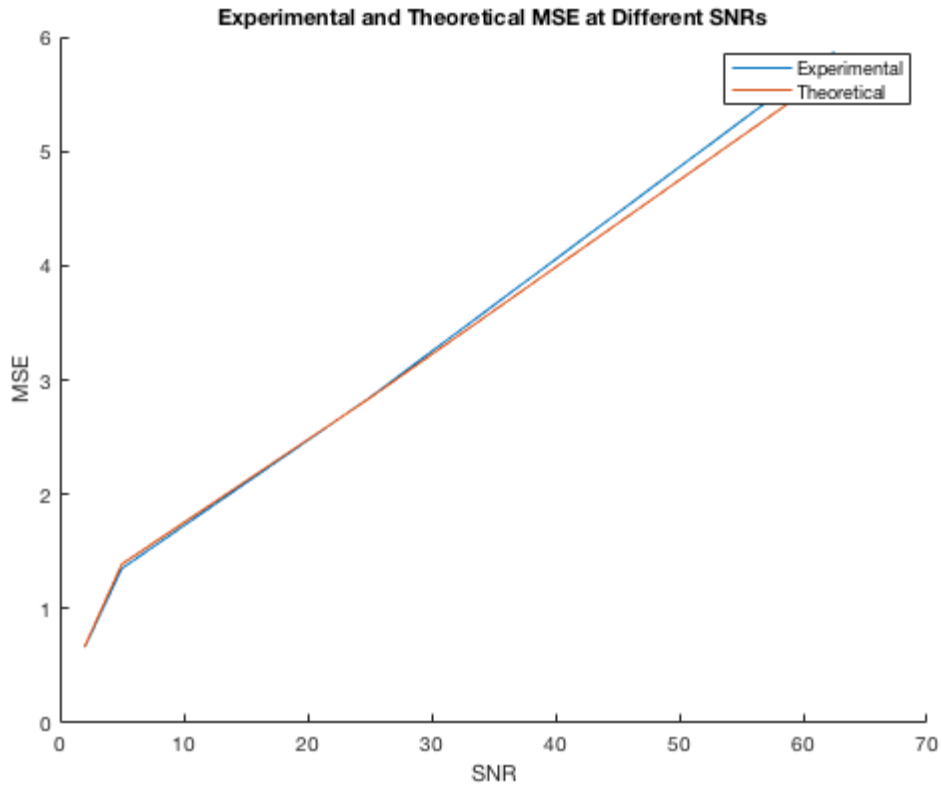
```
disp('Theoretical Mean Square Errors:')
disp(theoVals)
disp('Experimental Mean Square Errors:')
disp(expVals)

% plot values
figure;
hold on;
```

```
plot(SNR, expVals);
plot(SNR, theoVals);
legend('Experimental', 'Theoretical');
title('Experimental and Theoretical MSE at Different SNRs');
xlabel('SNR');
ylabel('MSE')
hold off;
```

```
Theoretical Mean Square Errors:
    0.6667    1.3889    2.8409    5.7013

Experimental Mean Square Errors:
    0.6614    1.3513    2.8453    5.8701
```



Functions

```
function mse = getMse(varY, varR, N)

    mse = (varY * varR)/(N*varY + varR);
    % Theoretical MSE from textbook page 160
end

function a = computeA(varY, varR, N)

    c = varY/(N*varY + varR);
    a = ones(N, 1) * c;
    % coefficient column vector pg 160 (8.78)
    % from textbook page 160 (8.80)
end
```

Published with MATLAB® R2020a