```
% Tests for Matching Solutions method with noise

close all;
clearvars;

k = 16; % Number of sensors
m = 4; % Size of observation vectors b
n = 20; % Size of unknown vector x
s = 14;
SNR = [5 10 15 20 25]; % SNR wanted
noise_levels_sigma = (10.^(-SNR/20));
restriction_delta = 10^-10;
threshold = 10^-4;
MCexperiments = 1000;

for noise_index = 1:length(noise_levels_sigma)

    noise_sigma = noise_levels_sigma(noise_index);
    reliable_sensors = [ones(1, s) zeros(1, k-s)];

    fprintf('Considered SNR: %d. ', SNR(noise_index));

    parfor j=1:MCexperiments
        %preallocations
        bi = zeros(m, 1, k);

        % unknown vector is modeled as x0 ~ N(0, n^(-1/2)In)
        x0 = mvnrnd(zeros(1, n), n^(-1)*eye(n))';

        % Entries of matrix A are drawn independently from N(0, 1)
        Ai = randn(m, n, k);

        for i=1:s
            % reliable sensors measures
            vi = mvnrnd(zeros(1, m), (noise_sigma^2)*eye(m))';
            bi(:, :, i) = Ai(:, : ,i)*x0 + vi;
        end

        for i=s+1:k
```

```
            % unreliable sensors measures
            bi(:, : , i) = mvnrnd(zeros(1, m), (1+noise_sigma^2)↙
*eye(m))';
        end

        x_iter0 = randn(n, k) / sqrt(n);
        lambda_iter0 = mean(x_iter0, 2);

        lambda_iter1 = matching_solutions( Ai, bi, n, k,↙
restriction_delta, x_iter0, lambda_iter0);
        results_noise_ms(j, noise_index) = norm(x0-lambda_iter1)^2;
    end
end

results_mse = mean(results_noise_ms, 1);

% plot data and add pretty stuff
semilogy(results_mse, '.-', 'MarkerSize',20, 'LineWidth', 1.5)
title('MSE variation with SNR')
xlabel('SNR [dB]')
ylabel('MSE')
legend('MS(1)', 'Location', 'southwest');
ax = gca;
ax.XTick = [1 2 3 4 5];
ax.XTickLabel = SNR;
grid on;
```