```
% Tests for LS-GA, LS, l1, P1, P2(1) methods with noise

% clear workspace and close all figures
close all;
clearvars;

% Setting parameters:
k = 16; % Number of sensors
m = 4; % Size of observation vectors b
n = 20; % Size of unknown vector x
s = 14; % Number of consistent sensors
delta = 1e-6; % Concave approximation related constant
methods = 4; % Methods being studied (LS, l1, P1, P2(1) )
SNR = [5 10 15 20 25]; % SNR wanted
noise_levels_sigma = (10.^(-SNR/20));
MCexperiments = 1000;

% save all MSE values for all methods
results_mse = zeros(length(noise_levels_sigma), methods);

for noise_index = 1:length(noise_levels_sigma)

    noise_sigma = noise_levels_sigma(noise_index);

    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

        % Rearrange arrays and matrices
        b = bi(:);
        C = permute(Ai, [1 3 2]);
        A = reshape(C, [], size(Ai, 2), 1);

        b_ga = b(1:m*s);
        A_ga = A(1:m*s,:)

        % LS-GA method
        x_ls_ga = ls_method(A_ga, b_ga, n);
        results_noise_ls_ga(j, noise_index) = norm(x0-x_ls_ga)^2;

        % LS method
        x_ls = ls_method(A, b, n);
        results_noise_ls(j, noise_index) = norm(x0-x_ls)^2;

        % l1 method
        x_l1 = l1_method(A, b, n);
        results_noise_l1(j, noise_index) = norm(x0-x_l1)^2;

        % P1 method
        x_p1 = p1_method(Ai, bi, n, k);
        results_noise_p1(j, noise_index) = norm(x0-x_p1)^2;

        % P2(1) method
        x_p2_1 = p2_1_method(Ai, bi, n, k, x_p1, delta);
        results_noise_p2_1(j, noise_index) = norm(x0-x_p2_1)^2;
    end
end
```

```
results_mse(:,1) = mean(results_noise_ls_ga, 1);
results_mse(:,2) = mean(results_noise_ls, 1);
results_mse(:,3) = mean(results_noise_l1, 1);
results_mse(:,4) = mean(results_noise_p1, 1);
results_mse(:,5) = mean(results_noise_p2_1, 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('LS-GA', 'LS', 'L_1', 'P_1', 'P_2(1)', 'Location',↙
'southwest');
ax = gca;
ax.XTick = [1 2 3 4 5];
ax.XTickLabel = SNR;
grid on;
```