```
N = 200;
t = 1:N;
F = [0.5, 0.5, 0; 0, 0.5, 0.5; 0, 0, 0.5];     %Transition matrix for state model
H = [1, 1, 1];                                              %Measurement matrix
Q = [1, 0, 0; 0, 1, 0; 0, 0, 1];                    %Covariance Matrix for state-model random↙
noise w
R = 1;                                                           %Covariance Matrix for↙
measurement noise/error
xtrue = zeros(3,N);                                    %true states of x
xtrue = [0; 0; 0];                                         %Our initial guess/estimate for x

xtrue = zeros(3, N);
xtrue = [0; 0; 0];
w = randn([3 N], 'like', N);

for k = 2:N
xtrue(:, k) = F*xtrue(:, k-1) + w(:, k-1);
end

v = randn(1,N);
z = H*xtrue + v;
xhat = zeros(3,N);
xhat(:, 1) = [0; 0; 0];
P = [1, 0, 0; 0, 1, 0; 0, 0, 1];                                    %Error Covariance at time 0
I3 = eye(3);

for k = 2:N
xhat(:, k) = F*xhat(:, k-1);                                     %One step prediction
P = F*P*F' + Q;                                                        %Error Covariance↙
update
K = P*H'/(H*P*H' + R);                                           %Gain matrix
xhat(:, k) = xhat(:, k) + K*(z(k) - H*xhat(:,k));      %update for estimate at time k
P = (I3 - K*H)*P;
end
>>
>> figure(1);
plot(t, xhat(1,:), '-', 'LineWidth', 2);
hold on;
plot(t, xtrue(1,:), '-', 'LineWidth', 1.5)
xlabel('time'); ylabel('x1 (First Component)');
grid on;
legend('x1 Estimated', 'x1 True');

figure(2);
plot(t, xhat(2,:), '-', 'LineWidth', 2);
hold on;
plot(t, xtrue(2,:), '-', 'LineWidth', 1.5)
xlabel('time'); ylabel('x2 (Second Component)');
grid on;
legend('x2 Estimate', 'x2 True');

figure(3);
plot(t, xhat(3,:), '-', 'LineWidth', 2);
hold on;
plot(t, xtrue(3, :), '-', 'LineWidth', 1.5)
xlabel('time'); ylabel('x3 (Third Component');
grid on;
legend('x3 Estimate', 'x3 True');
```