HA2 Analysis

Table of Contents

1 A first Kalman filter and its properties

1a

```
clear; clc; close all;
rng(970926)
% Process model
A = 1;
Q = 1.5;

% Measurement model
H = 1;
R = 2.5;

% Initial Prior p(x0) = N(x0; 2; 6)
mu = 2;
P_0 = 6;

N = 20;

X = genLinearStateSequence(mu, P_0, A, Q, N);
Y = genLinearMeasurementSequence(X, H, R);

x_0 = X(1);

figure(1); clf;
plot(0:N,X)
hold on
plot(1:N, Y, '*')
grid on

title('State and measurement sequence')
legend('State', 'Measurement')
```

```
xlabel('Sample (k)');
ylabel('State value');
```

1b

```
[X_kalman, P] = kalmanFilter(Y, x_0, P_0, A, Q, H, R);

figure(1); clf;hold on;
plot(0:N,X)
plot(1:N,Y, '*');
plot(0:N, [x_0 X_kalman], 'g')
plot(0:N, [x_0 X_kalman] + 3*sqrt([P_0 P(:)']), '--b');
plot(0:N, [x_0 X_kalman] - 3*sqrt([P_0 P(:)']), '--b');

grid on

title('Kalman filter')
legend('State', 'Measurement', 'Kalman filter', '+-3\sigma level')
xlabel('Sample (k)');
ylabel('State value');

% Plotposterior density with true density
n = 2;
for k = [5, 10, 15]
    figure(n); clf; hold on;
    space = 3*sqrt(P(k));
    x = (X_kalman(k)-space):0.01:(X_kalman(k)+space);
    px = normpdf(x, X_kalman(k), P(k));
    plot(x, px)
    xline(X(k+1), 'r');
    xline(Y(k), 'y');

    legend('Posterior density', 'True state', 'Measurement')
    title("Posterior density, N = " + k)
    xlabel('State value')
    n = n + 1;
end
```

1c

```
% Plot PDF of P(xk-1 given y1k-1 ) etc instead
clc
figure(2); clf; hold on;grid on;

sample = 14;
%Perform Kalman manually
% Plot previous kalman estimation and next
%plot(start:start+1, X_kalman(start:start+1), 'b--o')

% p(x_k-1|y_1:k-1) - Previous kalman
x = -10:0.01:4;
p = normpdf(x,X_kalman(sample), P(sample));
plot(x,p)
```

```
% p(x_k|y_1:k-1) - Prediction
[X_predict, P_predict] = linearPrediction(X_kalman(sample), P(sample),
 A, Q);
p = normpdf(x, X_predict, P_predict);
plot(x, p)

% y_k - Measurement
xline(Y(sample+1))

% p(x_k|y_1:k) - Measurement update
p = normpdf(x, X_kalman(sample+1), P(sample+1));
plot(x,p)



% Plot prediction
%[X_predict, P_predict] = linearPrediction(X_kalman(start), P(start),
 A, Q);
%plot(start+1, X_predict, '+');

% Plot measurement point.
%plot(start+1, Y(start+1), 'r*')

legend("p(x_{k-1}|y_{1:k-1})", "p(x_k|y_{1:k-1})", "y_k", 'p(x_k|
y_{1:k})')
xlabel('State value')
title('Prev. kalman, prediction, measurement and kalman comparison')
```

1d

```
clc
%Histogram part
N = 5000;

X = genLinearStateSequence(mu, P_0, A, Q, N);
Y = genLinearMeasurementSequence(X, H, R);
x_0 = X(1);

[X_kalman, P, V] = kalmanFilter(Y, x_0, P_0, A, Q, H, R);

est_error = X(2:end) - X_kalman;
fprintf('True mean: %.4f\n', mean(X));
fprintf('Estimated mean: %.4f\n', mean(X_kalman));
fprintf('Esttimation error mean: %.4f\n', mean(est_error));
figure(1); clf; hold on; grid on;
histogram(est_error,'Normalization','pdf')

x = -5:0.01:5;
px = normpdf(x, 0, P(end));
plot(x, px, 'LineWidth', 2)

title('Histogram of estimation error and N(x;0,P_{N|N})')
```

```
legend('$x_k - \hat{x}_{k|k}$', '$N(x;0,P_{N|
N})$', 'Interpreter','latex', 'FontSize', 14)
xlabel('Error')

% Auto correlation part
figure(2); clf; hold on; grid on;
V_mean = mean(V);
fprintf('Inovation mean (V): %.4f\n', V_mean);
autocorr(V);
title('Autocorrelation of inovation mean (V)')
```

1f

```
N = 20;

X = genLinearStateSequence(mu, P_0, A, Q, N);
Y = genLinearMeasurementSequence(X, H, R);

x_0 = X(1);

[X_kalman, P]       = kalmanFilter(Y, x_0, P_0, A, Q, H, R);

% Generate kalman with wrong initial assumption
x_0_wrong = 10;
P_0_wrong = 6;
[X_kalman_wrong, P_wrong] = kalmanFilter(Y, x_0_wrong, P_0_wrong, A,
 Q, H, R);

figure(1); clf; hold on; grid on;
plot(0:N,X)
plot(1:N,Y, '*');
plot(0:N, [x_0 X_kalman], 'g')
plot(0:N, [x_0_wrong X_kalman_wrong], 'r--')
legend('True state', 'Measurement', 'Kalman', 'Kalman_{wrong}')
title('Investigation of initial state impact')
xlabel('Sample (k)');
ylabel('State value');
```

2 Kalman filter and its tuning

2a

```
T = 0.01;

% Process model
A = [1 T; 0 1];
Q = [0 0; 0 1.5]; %Process noise covariance


% Measurement model
C = [1 0];
R = 2;      % Measurement noise variance
```

```
% Initial Prior p(x0) = N(x0; [1 3]'; 4*I)
mu = [1; 3];
P_0 = 4*eye(2);

N = 0.5/T; % 2s long data

X = genLinearStateSequence(mu, P_0, A, Q, N);
Y = genLinearMeasurementSequence(X, C, R);

x_0 = X(:,1);

% Position plot
figure(1); clf; hold on; grid on;
plot((0:N).*T ,X(1,:), 'b') % Position
plot((1:N).*T,Y, 'r.')      % Measurements
legend('Position ', 'Measurements')
title('Position')

xlabel('Time [s]')
ylabel('1D Position [m]');

% Velocity plot
figure(2); clf; hold on; grid on;
plot((0:N).*T,X(2,:), 'b') % Speed
xlabel('Time [s]')
ylabel('1D Speed [m]');
legend('Speed')
title('Speed')
```

2b

```
[X_kalman, P] = kalmanFilter(Y, x_0, P_0, A, Q, C, R);

% Position plot
figure(1); clf; hold on; grid on;
plot((0:N).*T ,X(1,:), 'b') % Position
plot((1:N).*T,Y, 'r.')      % Measurements
plot((0:N).*T, [x_0(1) X_kalman(1,:)], 'g') % Kalman filter Position

pos_var = P(1,1,:);
plot((0:N).*T, [x_0(1) X_kalman(1,:)] + 3*sqrt([P_0(1,1)
 pos_var(:)']),'-', 'color', [0 0.5 0 0.5]);
plot((0:N).*T, [x_0(1) X_kalman(1,:)] - 3*sqrt([P_0(1,1)
 pos_var(:)']), '-', 'color', [0 0.5 0 0.5]);


title('Kalman filter on position')
xlabel('Time [s]');
ylabel('1D Pos [m]');
legend('Position ', 'Measurements', 'Kalman of pos', '$\pm 3\sigma
$', 'Interpreter','latex')
```

```
% Velocity plot
figure(2); clf; hold on; grid on;

speed_var = P(2,2,:);
plot((0:N).*T,  X(2,:), 'b') % Speed
plot((0:N).*T, [x_0(2) X_kalman(2,:)], 'g') % Kalman filter Speed
plot((0:N).*T, [x_0(2) X_kalman(2,:)] + 3*sqrt([P_0(2,2)
 speed_var(:)']),'-', 'color', [0 0.5 0 0.5]);
plot((0:N).*T, [x_0(2) X_kalman(2,:)] - 3*sqrt([P_0(2,2)
 speed_var(:)']), '-', 'color', [0 0.5 0 0.5]);



title('Kalman filter on speed')
xlabel('Time [s]')
ylabel('1D Speed [m]');
legend('Speed', 'Kalman of speed', '$\pm 3\sigma
$', 'Interpreter','latex')
```

2c

```
q = [0.1 1 10 1.5];
for i = 1:length(q)
    Q = [0 0; 0 q(i)];

    figure(i); clf; grid on;
    [X_kalman, P] = kalmanFilter(Y, x_0, P_0, A, Q, C, R);

    % Position
    subplot(2,1,1)
    hold on
    plot((0:N).*T ,X(1,:), 'b') % Position
    plot((1:N).*T,Y, 'r.')      % Measurements
    plot((0:N).*T, [x_0(1) X_kalman(1,:)], 'g') % Kalman filter
 Position

    pos_var = P(1,1,:);
    plot((0:N).*T, [x_0(1) X_kalman(1,:)] + 3*sqrt([P_0(1,1)
 pos_var(:)']),'-', 'color', [0 0.5 0 0.5]);
    plot((0:N).*T, [x_0(1) X_kalman(1,:)] - 3*sqrt([P_0(1,1)
 pos_var(:)']), '-', 'color', [0 0.5 0 0.5]);


    title("Motion noise Q = " + Q(2,2))
    xlabel('Time [s]');
    ylabel('1D Pos [m]');
    legend('Position ', 'Measurements', 'Kalman of pos', '$\pm 3\sigma
$', 'Interpreter','latex')

    % Velocity
    subplot(2,1,2)
    hold on
```

```
    speed_var = P(2,2,:);
    plot((0:N).*T,  X(2,:), 'b') % Speed
    plot((0:N).*T, [x_0(2) X_kalman(2,:)], 'g') % Kalman filter Speed
    plot((0:N).*T, [x_0(2) X_kalman(2,:)] + 3*sqrt([P_0(2,2)
 speed_var(:)']),'-', 'color', [0 0.5 0 0.5]);
    plot((0:N).*T, [x_0(2) X_kalman(2,:)] - 3*sqrt([P_0(2,2)
 speed_var(:)']), '-', 'color', [0 0.5 0 0.5]);


    title('Kalman filter on speed')
    xlabel('Time [s]')
    ylabel('1D Speed [m]');
    legend('Speed', 'Kalman of speed', '$\pm 3\sigma
$', 'Interpreter','latex')

end
```

Published with MATLAB® R2019b