

---

## Table of Contents

Kalman Filter .....	1
Import numbers from excel .....	1
A priori statistics .....	1
State variables and Equations .....	1
FOR LOOP .....	2
Smoothing .....	3
Plot .....	3

## Kalman Filter

```
clear all;clc;close all
addpath([' /Users/kevin/SkyDrive/KTH Work/ ',...
        'Period 4 2014/GNSS/Labs/L4 ',...
        '- Kalman filtering/']);
```

## Import numbers from excel

Measured values from sheet 1

```
data.s1 = xlsread('Measurement.xlsx');
meas.time = data.s1(1:25,1);
meas.east = data.s1(1:25,2);
meas.north = data.s1(1:25,3);
meas.speed = data.s1(1:25,4);
% True values from sheet 2
data.s2 = xlsread('Measurement.xlsx','True values');
true.time = data.s2(1:25,1);
true.east = data.s2(1:25,2);
true.north = data.s2(1:25,3);
true.vel_east = data.s2(1:25,4);
true.vel_north = data.s2(1:25,5);
```

## A priori statistics

```
PSD = 0.01; % - PSD (power spectral density) of the
% random acceleration
meas.sd_coord = 3; % m - Standard error of measured coordinates
meas.sd_abs_vel = 0.5; % m/s - Standard error of measured
% abs. velocity
sd_ini_vel = 3; % m/s - Standard error of initial velocity
sd_ini_coord = 10; % m - Standard error of initial coordinates
ve = 3.53; % m/s
vn = 0.86; % m/s
dt = 2; % time difference -> 2 sec between measurements
```

## State variables and Equations

```
xk = [meas.east(1) meas.north(1) ve vn];
```

---

```

xk = padarray(xk,24,0,'post');
xk = xk';
% Equation 4
F = zeros(4);
F(1,3) = 1;
F(2,4) = 1;
G = zeros(4,2);
G(3,1) = 1;
G(4,2) = 1;
% Equation 5
Tk = eye(length(F)) + dt * F;
% Equation 9
Q = [ PSD 0 ; 0 PSD];
% Equation 11
QG = G*Q*G';
% Equation 12
Qk = QG * dt + (F*QG + QG*F')*dt^2/2 + F*QG*F'*dt^3/3;
% Equation 15
% covariance matrix of initial state
Qx(:, :, 1) = diag([sd_ini_coord^2 ...
    sd_ini_coord^2 sd_ini_vel^2 sd_ini_vel^2]);
Rk = diag([meas.sd_coord meas.sd_coord meas.sd_abs_vel]);
Qxm_predicted = zeros(4,4,25);

```

## FOR LOOP

```

for i=1:25
    % Equation 16 Time propagation
    xkm_predicted(:,i) = Tk * xk(:,i);
    % Qx = cov(xk(:,i));
    Qxm_predicted(:, :, i) = Tk * Qx(:, :, i) * Tk' + Qk;
    vm_predicted = sqrt(xkm_predicted(3,i)^2 + ...
        xkm_predicted(4,i)^2); % should be equal to speed_meas(1)
    Hk(:, :, i) = [1,0, 0, 0;...
        0, 1, 0, 0;...
        0, 0, xkm_predicted(3,i)/vm_predicted, ...
        xkm_predicted(4,i) /vm_predicted];
    % Equation 17 Gain
    Kk(:, :, i) = Qxm_predicted(:, :, i) * Hk(:, :, i)'*inv([Rk + ...
        Hk(:, :, i) * Qxm_predicted(:, :, i) * Hk(:, :, i)']);
    Lk(:, i) = [meas.east(i) meas.north(i)...
        meas.speed(i)]';
    hkm_predicted = [xkm_predicted(1,i) xkm_predicted(2,i) ...
        sqrt(xkm_predicted(3,i)^2 + xkm_predicted(4,i)^2)]';

    xk(:,i+1) = xkm_predicted(:,i) + Kk(:, :, i)*[ Lk(:, i) ...
        - hkm_predicted ]; % Equation18
    % Measurement update
    % Equation 19
    Qx(:, :, i+1) = [eye(length(Kk(:, :, i))*...
        Hk(:, :, i)) -Kk(:, :, i)*Hk(:, :, i)]*Qxm_predicted(:, :, i);
    % Equation 22
    % Lk = [meas.east(i) meas.north(i)...

```

---

```

%           sqrt((xk(3,i))^2 + (xk(4,i))^2)]';

%       Hk = inv(xk(:,i))*Lk;
final.xplot(:,i+1) = xk(:,i);
end

```

## Smoothing

```

xkhat(:,25) = xk(:,end);
nStep = 25;
count = nStep;
Qxkhat(:, :, 25) = Qx(:, :, end);
for i = 1:(nStep-1)
    Dk = Qx(:, :, count+1)*Tk'*inv(Qxm_predicted(:, :, count));
    xkhat(:, count-1) = xk(:, count) + Dk*[xkhat(:, count) ...
        - xkm_predicted(:, count)];
    Qxkhat(:, :, count-1) = Qx(:, :, count+1) ...
        + Dk*[Qxkhat(:, :, count) - Qxm_predicted(:, :, count)]*Dk';
    count = count - 1;
end

```

## Plot

```

final.x1 = xk(1,:); % final x values
final.y1 = xk(2,:); % final y values
meas.x2 = meas.east; % original x
meas.y2 = meas.north; % original y
true.x3 = true.east; % true x
true.y3 = true.north; % true y
figure('Units', 'pixels', ...
    'Position', [100 100 500 375]);
hold on;
y1_plot = plot(final.x1, final.y1, ... % Before Smoothing
    xkhat(1,:), xkhat(2,:), ... % Smoothing
    meas.x2, meas.y2, ... % Original
    true.x3, true.y3) % True Plot
hTitle = title('Kalman filtering');
hXLabel = xlabel('Easting [m]');
hYLabel = ylabel('Northing [m]');
hLegend = legend(...
    'Before Smoothing', ...
    'Smoothing', ...
    'Original', ...
    'True Plot', ...
    'location', 'best');
set(gca, ...
    'FontName', 'Helvetica');
set([hTitle, hXLabel, hYLabel], ...
    'FontName', 'AvantGarde');
set([hLegend, gca], ...
    'FontSize', 10);
set([hXLabel, hYLabel], ...
    'FontSize', 11);

```

---

```
set( hTitle
    'FontSize'    , 13
    'FontWeight'  , 'bold'
    );
set(gca, ...
    'Box'         , 'off'
    'TickDir'     , 'out'
    'TickLength'  , [.02 .02]
    'XMinorTick'  , 'on'
    'YMinorTick'  , 'on'
    'YGrid'       , 'on'
    'XColor'      , [.3 .3 .3]
    'YColor'      , [.3 .3 .3]
    'LineWidth'   , 1
    );
hold off;
```

*Published with MATLAB® R2013a*