Laboratory work 9

Development of tracking filter of a moving object

when measurements and motion model are in different coordinate systems

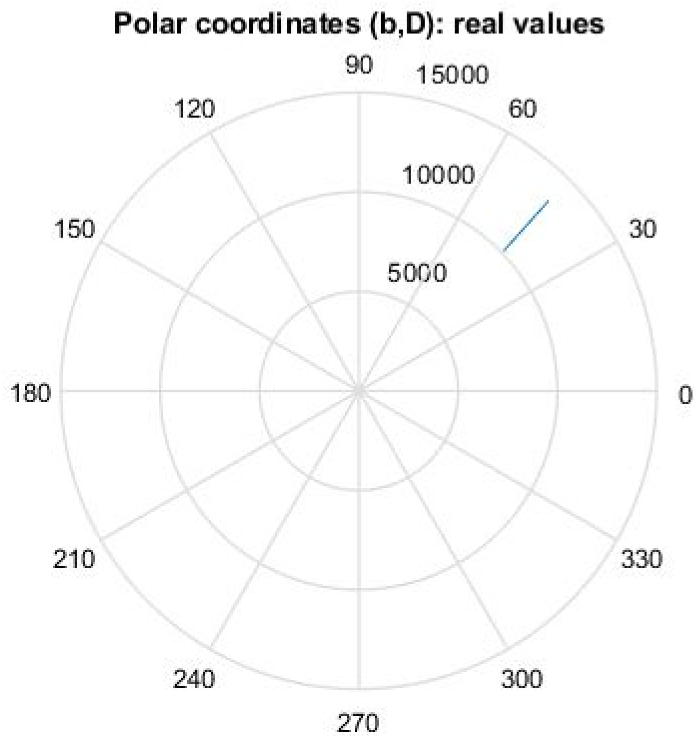
Group 5: Ruslan Agishev, Andrei Chemikhin, Valery Nevzorov

Skoltech, 2017

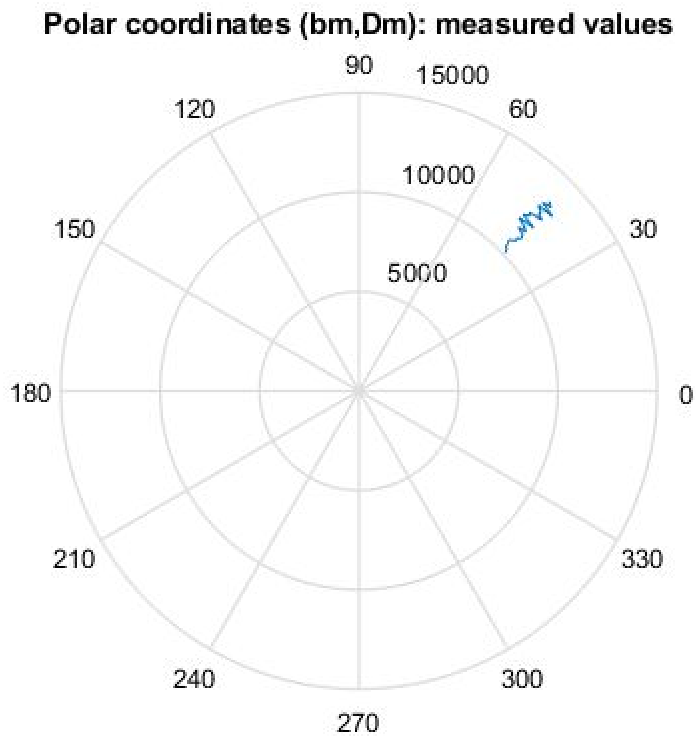
Processing with 1 trajectory

```
close all;
clear;
[N,M,T,vx1,vy1, x1,y1, sigmaD,sigmab, t, D1,b1, X0,P0, F,H] = init();
[x, y, D,b, Dm,bm] = trajgen(x1,y1, N, T, vx1, vy1, D1,b1, sigmaD,sigmab);
figure(1)
polar(b,D);
title('Polar coordinates (b,D): real values')

figure(2)
polar(bm,Dm);
title('Polar coordinates (bm,Dm): measured values')

[xm,ym] = polar2cartesian(Dm,bm);

Z = [xm; ym];
R = covarMatrix(Dm,bm, sigmaD,sigmab);

[Xpr,Ppr,Xfl,Pfl,K] = kalman_filter(X0,P0,F,H,R,Z);
xfl = Xfl(1,:);
yfl = Xfl(3,:);
vxfl = Xfl(2,:);
vyfl = Xfl(4,:);

figure(3)
plot(xm,ym, x,y, xfl(2:end),yfl(2:end));
legend('measure','real', 'filter')
% xlim([7000,11000])
% ylim([7000,11000])
grid on

[Dfl,bfl] = cartesian2polar(xfl,yfl);
figure(4)
polar(bfl(2:end),Dfl(2:end))
title('Polar coordinates (bfl,Dfl): filtered values')

% conditional values of covariance matrix for each tim moment
```

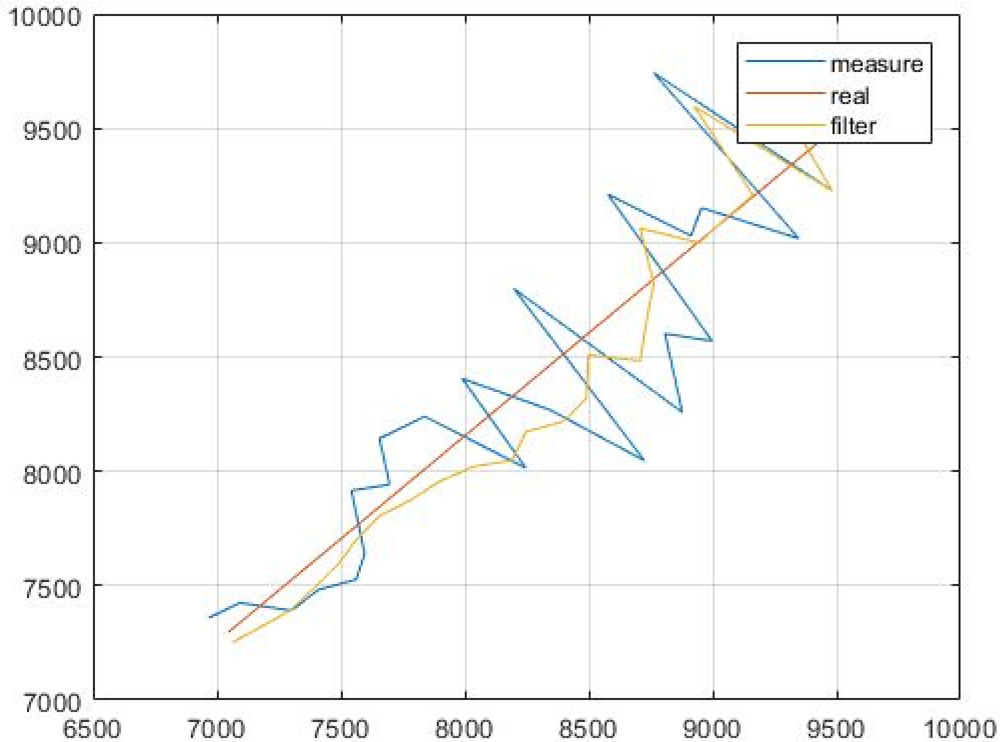
```
cv = nan(size(R));
for i=1:length(cv)
    cv(i) = cond(R{i});
end
figure(5)
plot(t,cv)
xlim([1,26])
xlabel('time')
ylabel('cond(R)')
title('Conditional number of covariance matrix')
grid on
% cond(R) is relativelly small and decreases over time. This means that
% measurements are accurate enough.

% analysis of filter gain K(1,1)
k = nan(1,N-1);
for i=1:(N-1)
    k(i) = K{i+1}(1,1);
end
figure(6)
plot(t(2:N),k,'o');
xlabel('time')
ylabel('K(1,1)')
title('Kalman filter gain over time')
grid on
% For some moments of time K>1. This is related to the fact that matrix ð?‘…
% depends on polar measurements that have errors.
```



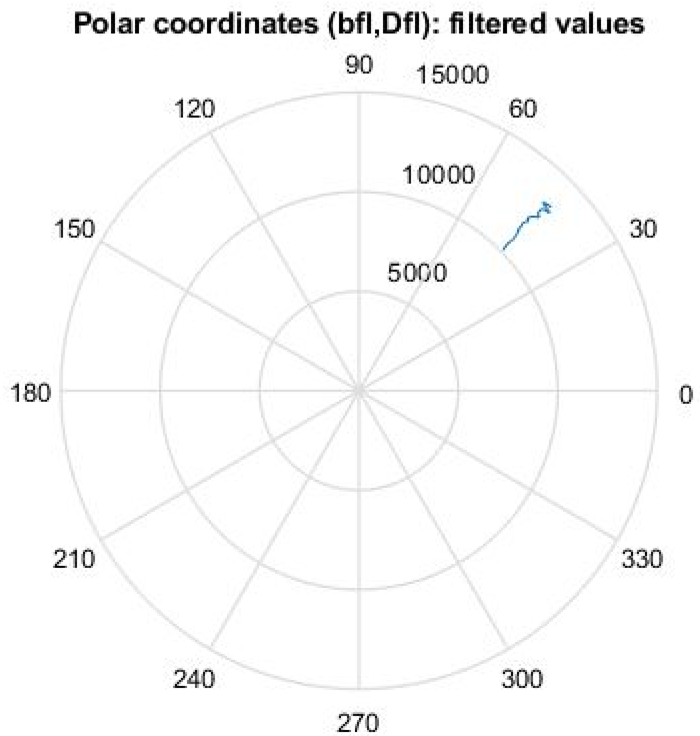
Real trajectory in polar coordinates



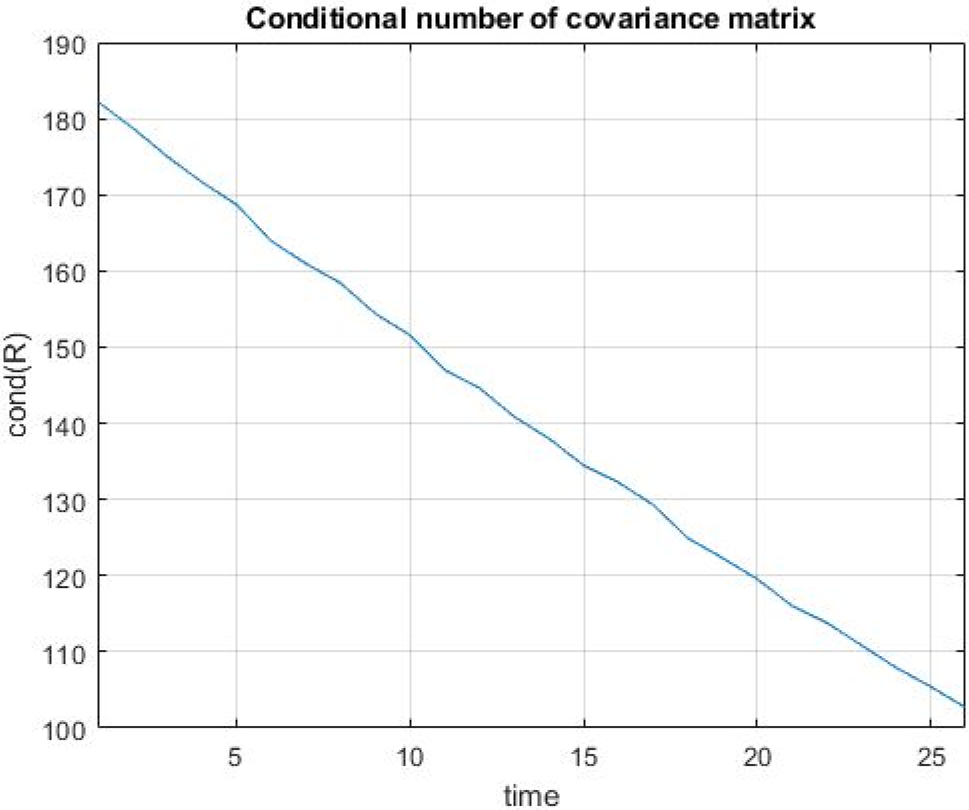
Measurements in polar coordinates



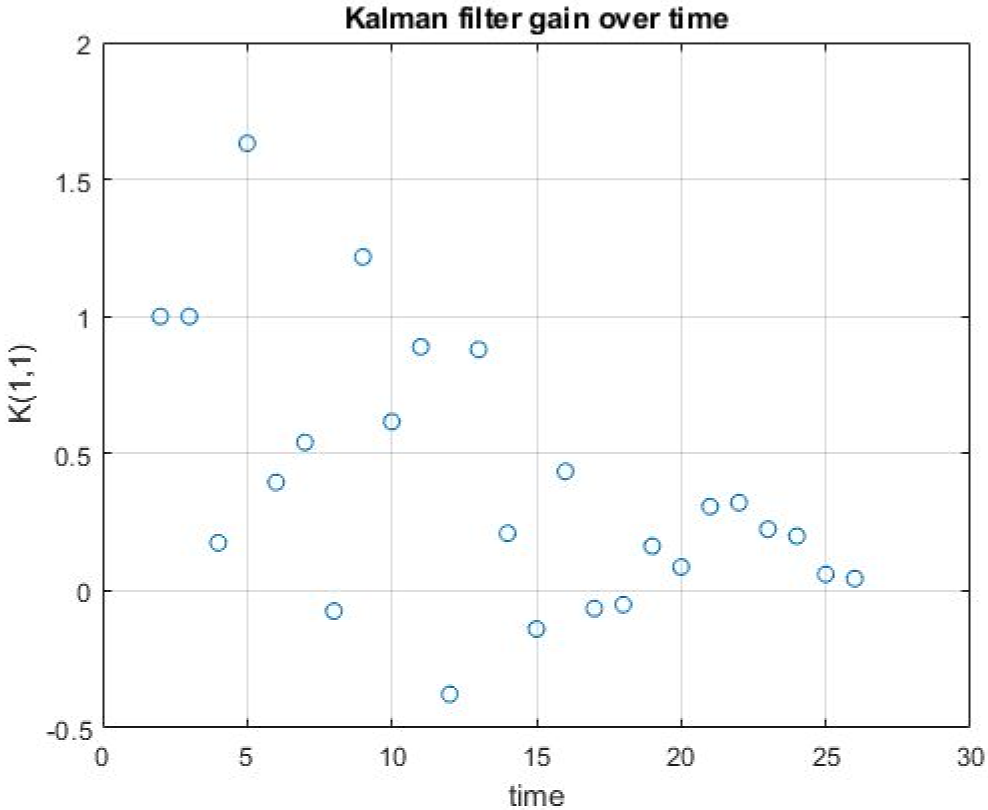
Measurements, real and filtered trajectory in cartesian coordinates



Filtered trajectory of the object in polar coordinates



Decrease of conditional number of covariance matrix. It is posiible to use linear model



Filter gain K(1,1) can be greater than 1, due to conversion of coordinate system errors

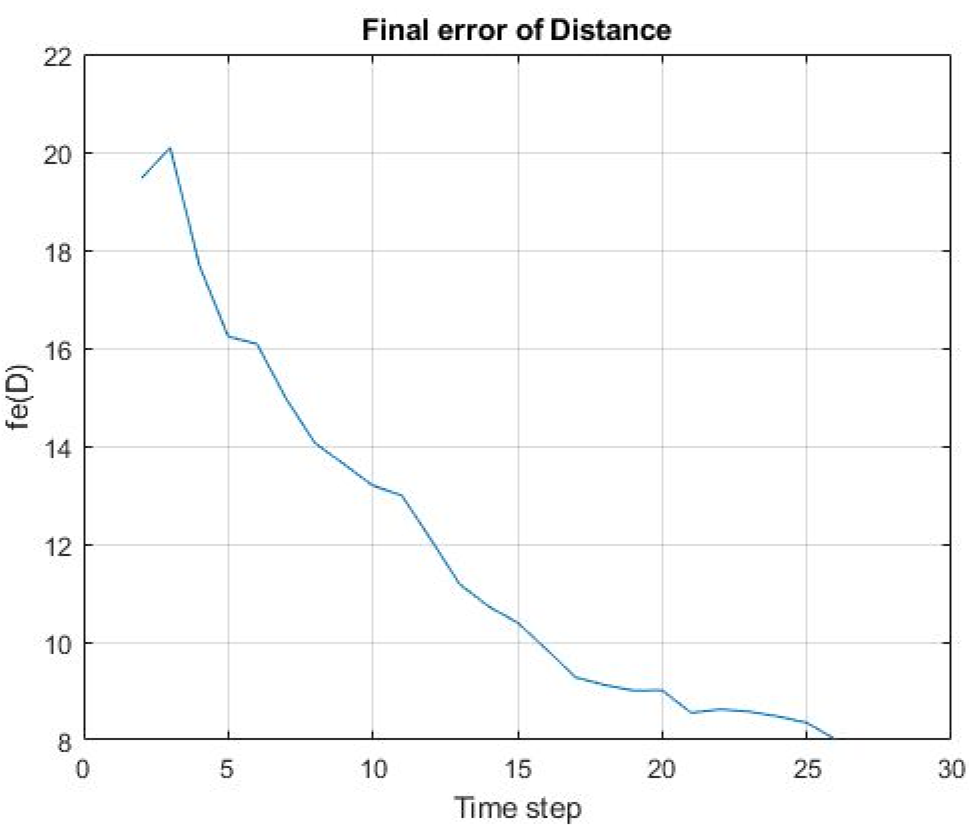
Processing over M=500 random trajectories

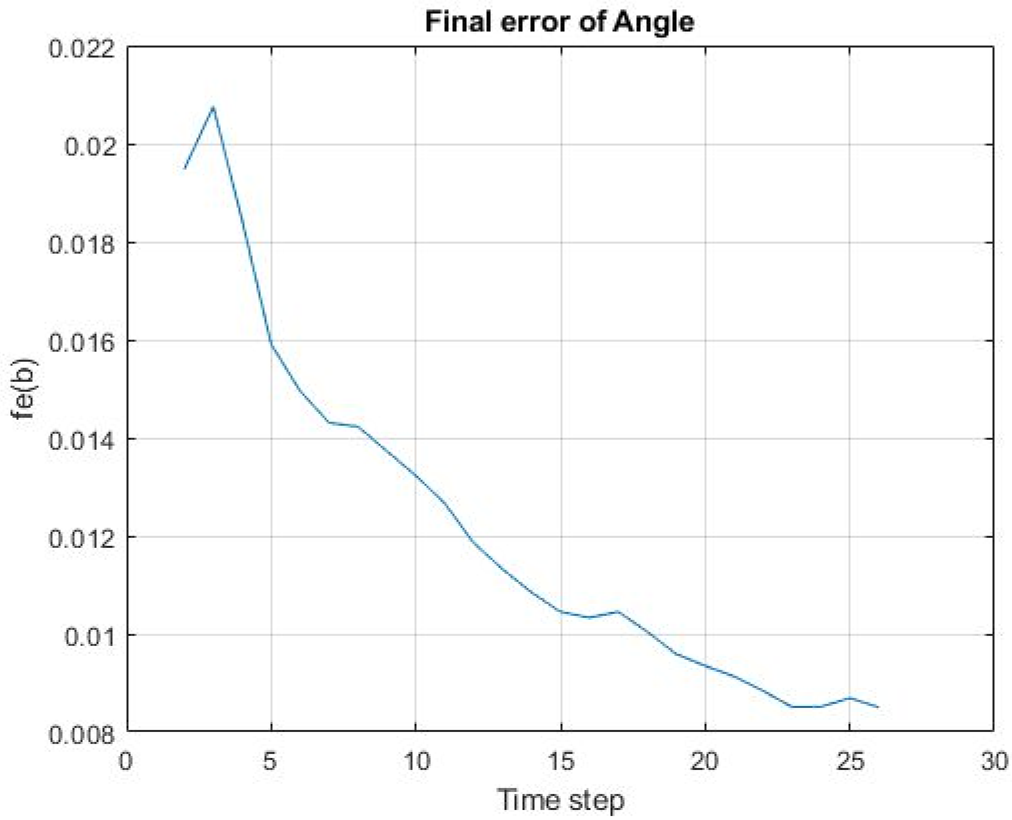
```
close all;
clear;
[N,M,T,vx1,vy1, x1,y1, sigmaD,sigmab, t, D1,b1, X0,P0, F,H] = init();

% generation of M=500 realiztions of trajectories
X = cell(1,M);
Y = cell(1,M);
Xm = cell(1,M);
Ym = cell(1,M);
D = cell(1,M);
b = cell(1,M);
Dm = cell(1,M);
bm = cell(1,M);
R = cell(1,M);
Z = cell(1,M);
for i=1:M
    [X{i},Y{i}, D{i},b{i}, Dm{i},bm{i}] = trajgen(x1,y1, N, T, vx1, vy1, D1,b1, sigmaD,sigmab);
    [Xm{i},Ym{i}] = polar2cartesian(Dm{i},bm{i});
```

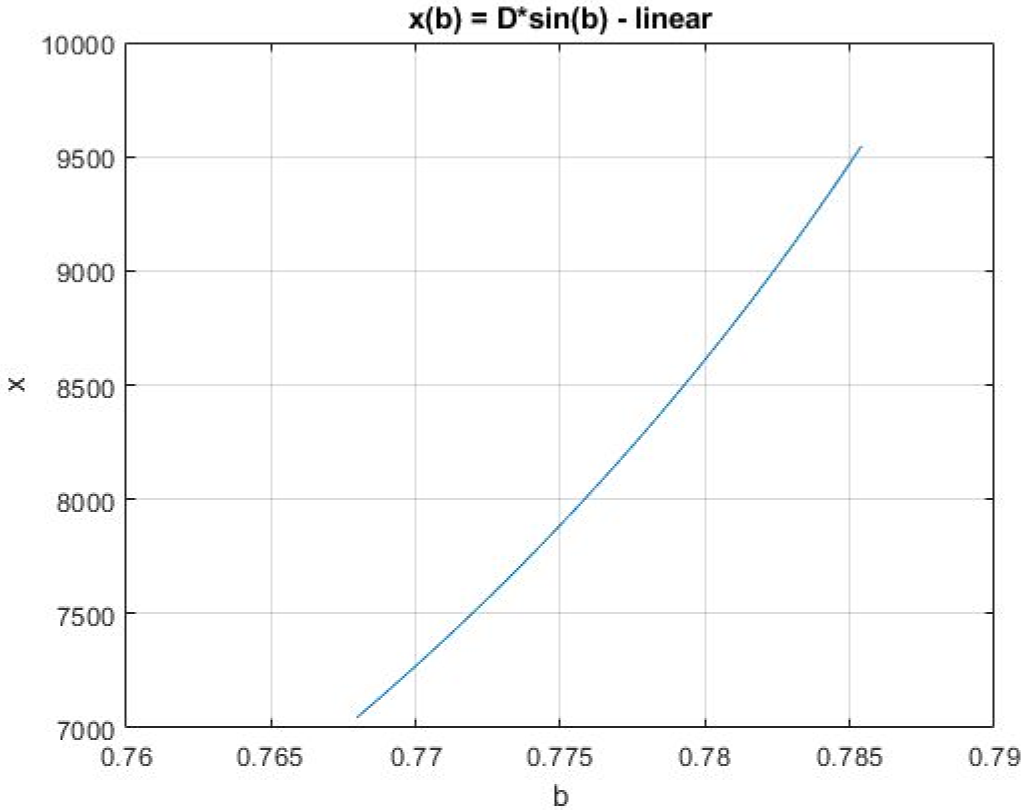
```
    R{i} = covarMatrix(Dm{i},bm{i}, sigmaD,sigmab);
    Z{i} = [Xm{i}; Ym{i}];
end

% Kalman-filtration of generated trajectories
Xfl = cell(1,M);
Dfl = cell(1,M);
bfl = cell(1,M);
for i=1:M
    [Xpr,Ppr,Xfl,Pfl,K] = kalman_filter(X0,P0,F,H,R{i},Z{i});
    [Dfl{i}, bfl{i}] = cartesian2polar(Xfl(1,:),Xfl(3,:));
end


feD = final_error(Dfl, D);
figure(1)
plot(t(2:end),feD(2:end))
ylabel('fe(D)')
xlabel('Time step')
title('Final error of Distance')
grid on;

feb = final_error(bfl, b);
figure(2)
plot(t(2:end),feb(2:end))
ylabel('fe(b)')
xlabel('Time step')
title('Final error of Angle')
grid on;

[x, y, D,b, Dm,bm] = trajgen(x1,y1, N, T, vx1, vy1, D1,b1, sigmaD,sigmab);
figure(3)
plot(b,x)
grid on
xlabel('b')
ylabel('x')
xlim([0.76, 0.79])
title('x(b) = D*sin(b) - linear')
% x = D sin(b)
% x(b)-dependence is close to linear.
% It means that linearization errors don't accumulate over time quickly.

% new, closer initial conditions
x1 = 3500/sqrt(2);
y1 = 3500/sqrt(2);
```

```
% generation of M=500 realiztions of trajectories
X = cell(1,M);
Y = cell(1,M);
Xm = cell(1,M);
Ym = cell(1,M);
D = cell(1,M);
b = cell(1,M);
Dm = cell(1,M);
bm = cell(1,M);
R = cell(1,M);
Z = cell(1,M);

for i=1:M
    [X{i},Y{i}, D{i},b{i}, Dm{i},bm{i}] = trajgen(x1,y1, N, T, vx1, vy1, D1,b1, sigmaD,sigmab);
    [Xm{i},Ym{i}] = polar2cartesian(Dm{i},bm{i});
    R{i} = covarMatrix(Dm{i},bm{i}, sigmaD,sigmab);
    Z{i} = [Xm{i}; Ym{i}];
end

% Kalman-filtration of generated trajectories
Xfl = cell(1,M);
Dfl = cell(1,M);
bfl = cell(1,M);
for i=1:M
    [Xpr,Ppr,Xfl,Pfl,K] = kalman_filter(X0,P0,F,H,R{i},Z{i});
    [Dfl{i}, bfl{i}] = cartesian2polar(Xfl(1,:),Xfl(3,:));
end


feD2 = final_error(Dfl, D);
figure(4)
plot(t(2:end),feD(2:end), t(2:end),feD2(2:end))
legend('bad i.c.', 'closer i.c.')
ylabel('fe(D)')
xlabel('Time step')
title('Final error of Distance')
grid on;

feb2 = final_error(bfl, b);
figure(5)
plot(t(2:end),feb(2:end), t(2:end),feb2(2:end))
legend('bad i.c.', 'closer i.c.')
ylabel('fe(b)')
xlabel('Time step')
```

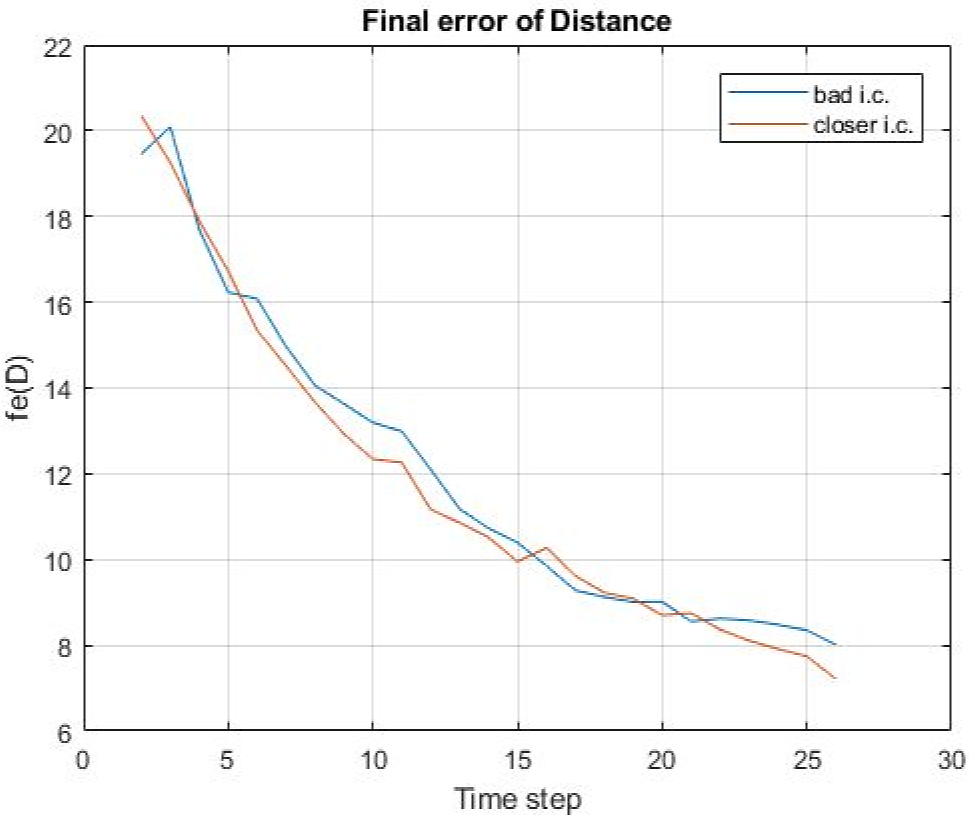
```
title('Final error of Angle')
grid on;
% Linear approximation leads to error accumulation in azimuth at the end of
% time period: x(b) = D sin(b) - non-linear function.

[x, y, D,b, Dm,bm] = trajgen(x1,y1, N, T, vx1, vy1, D1,b1, sigmaD,sigmab);
figure(6)
plot(b,x)
grid on
xlabel('b')
ylabel('x')
title('x(b) = D*sin(b) - linear')
% New dependence (with closer initial conditions) becomes not close to
% linear at the right part of time period.
% It means that linearization errors increase over time rapidly.
```

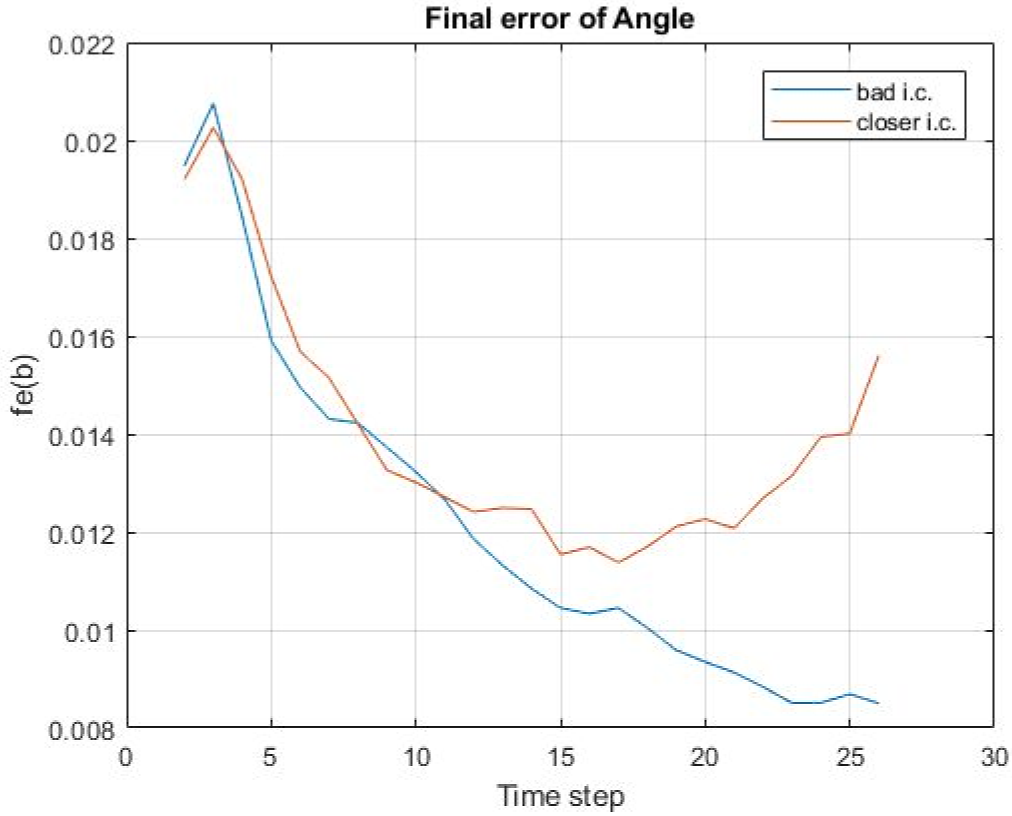



Final error of Angle
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
fe(b)
0
5
10
15
20
25
30
Time step

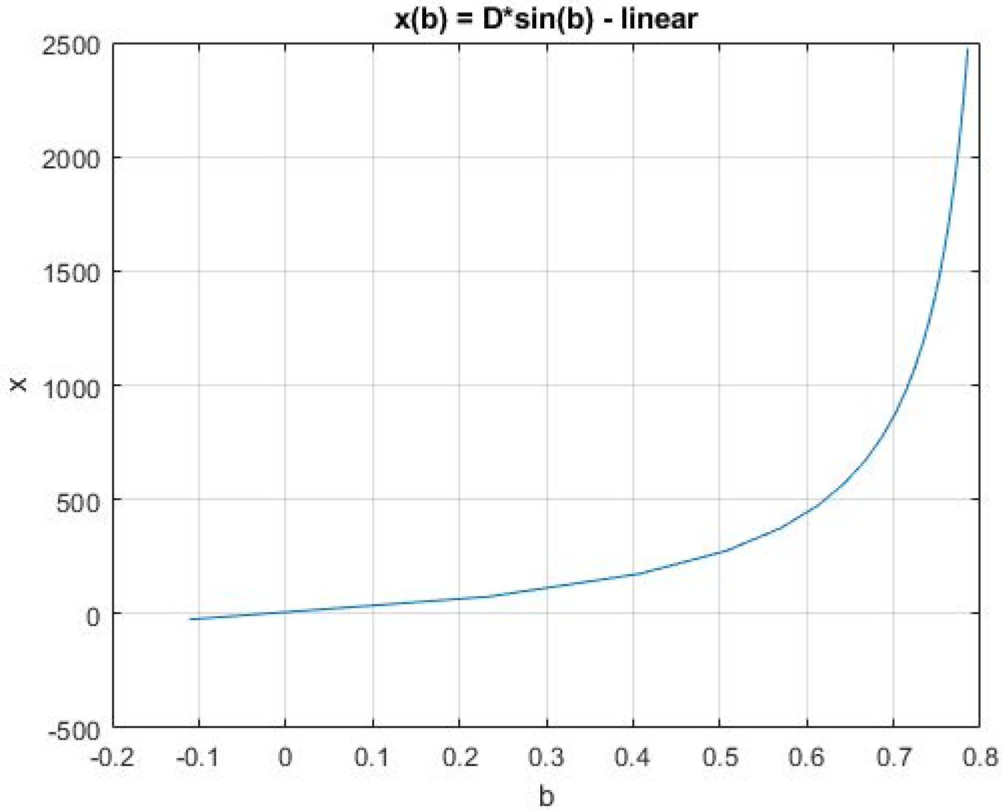

x(b) = D*sin(b) - linear
10000
9500
9000
8500
8000
7500
7000
x
0.76
0.765
0.77
0.775
0.78
0.785
0.79
b

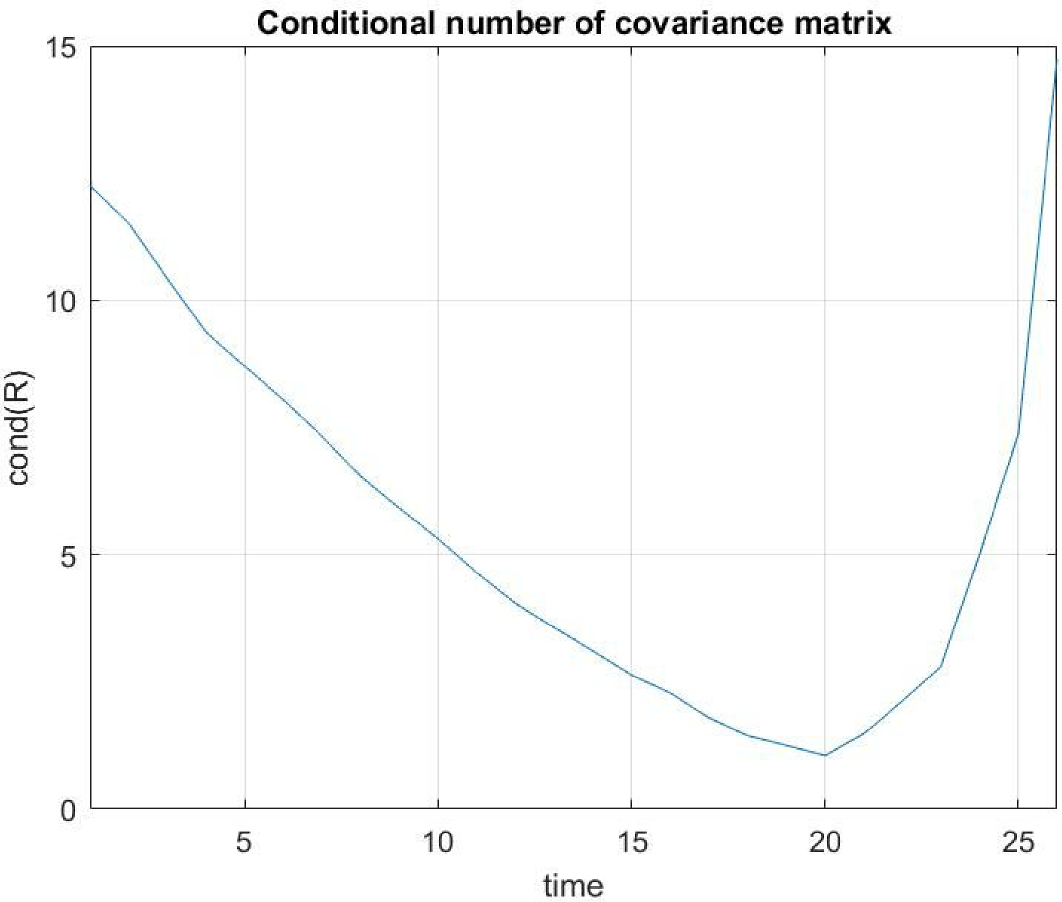
Closer initial conditions: (x0, y0) = (3500/sqrt(2),3500/sqrt(2))





Final error of Angle
bad i.c.
closer i.c.
fe(b)
Time step
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0
5
10
15
20
25
30

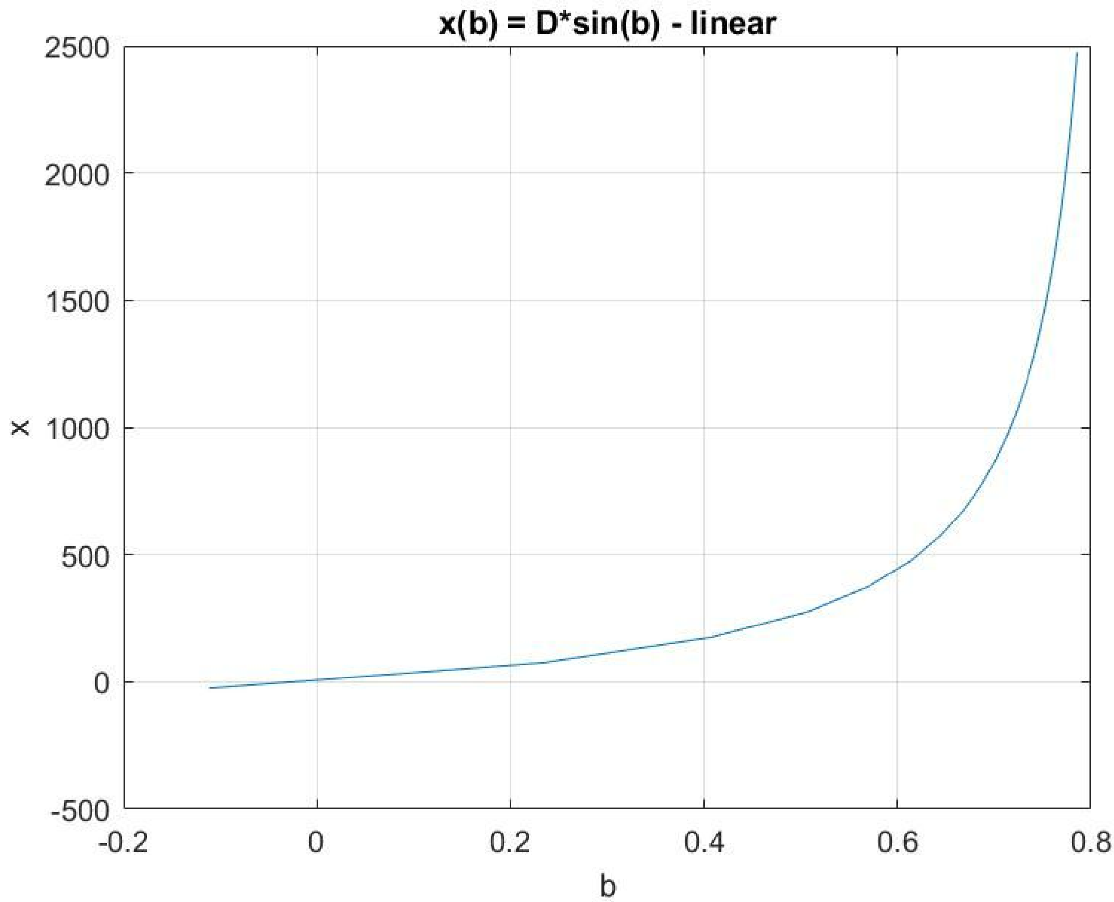

x(b) = D*sin(b) - linear
x
b
2500
2000
1500
1000
500
0
-500
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8



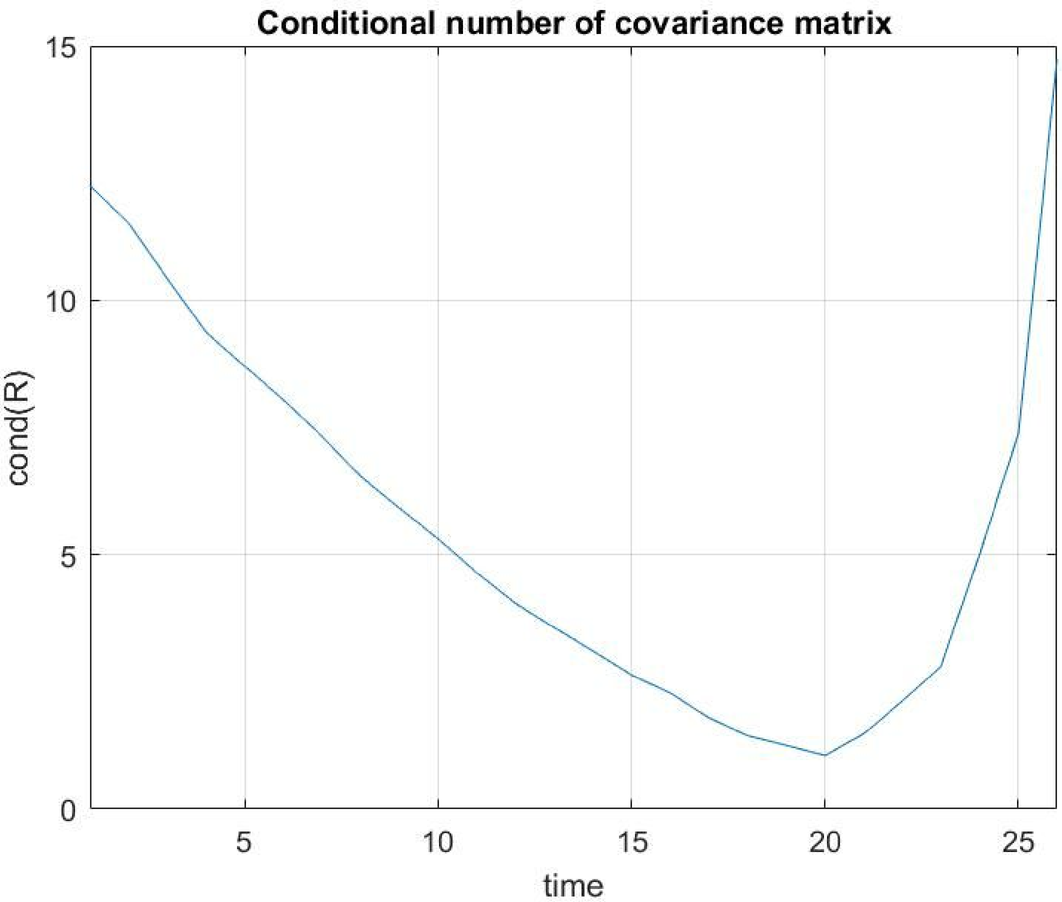
For sufficiently close initial values over time, the linearization error increases exponentially, which can not be said for less accurate initial values.

We can make conclusion that if we use the linearization we need take less accurate initial values.

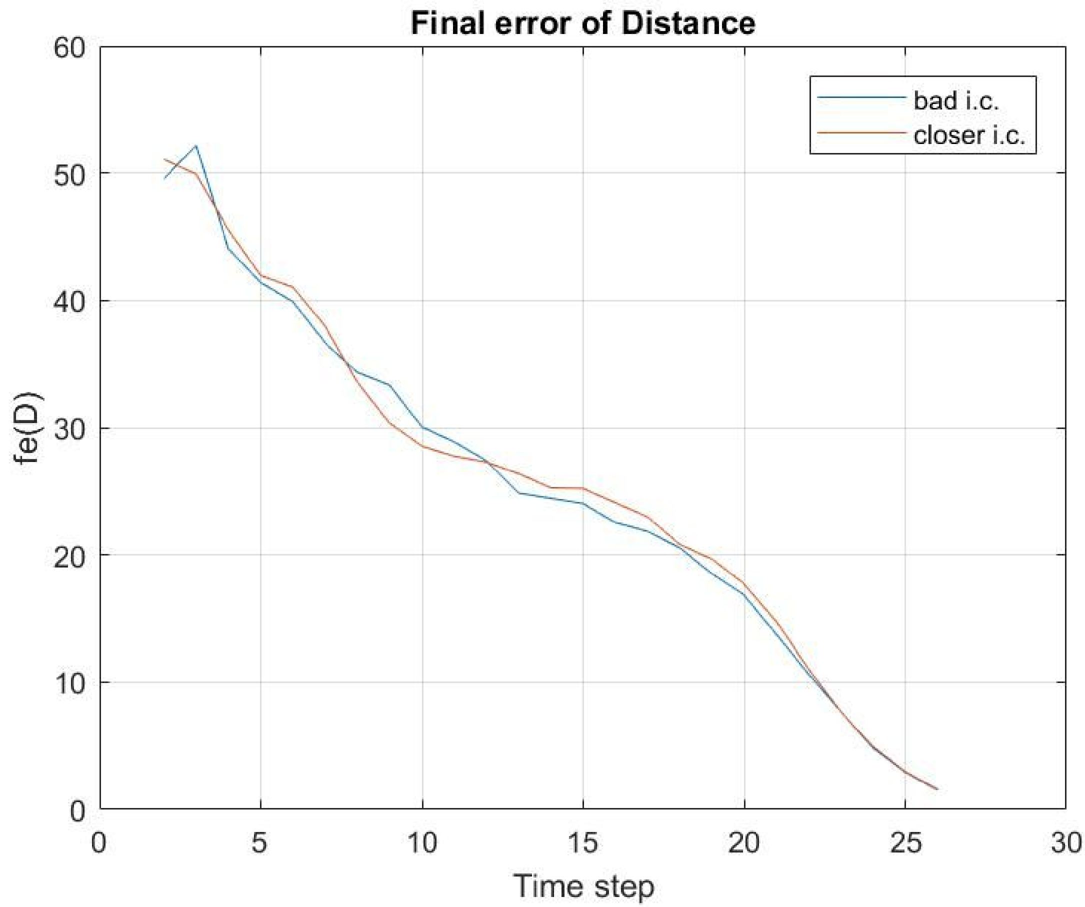
Setting new values of variances: $\sigma_D = 50$, $\sigma_\beta = 0.0015$

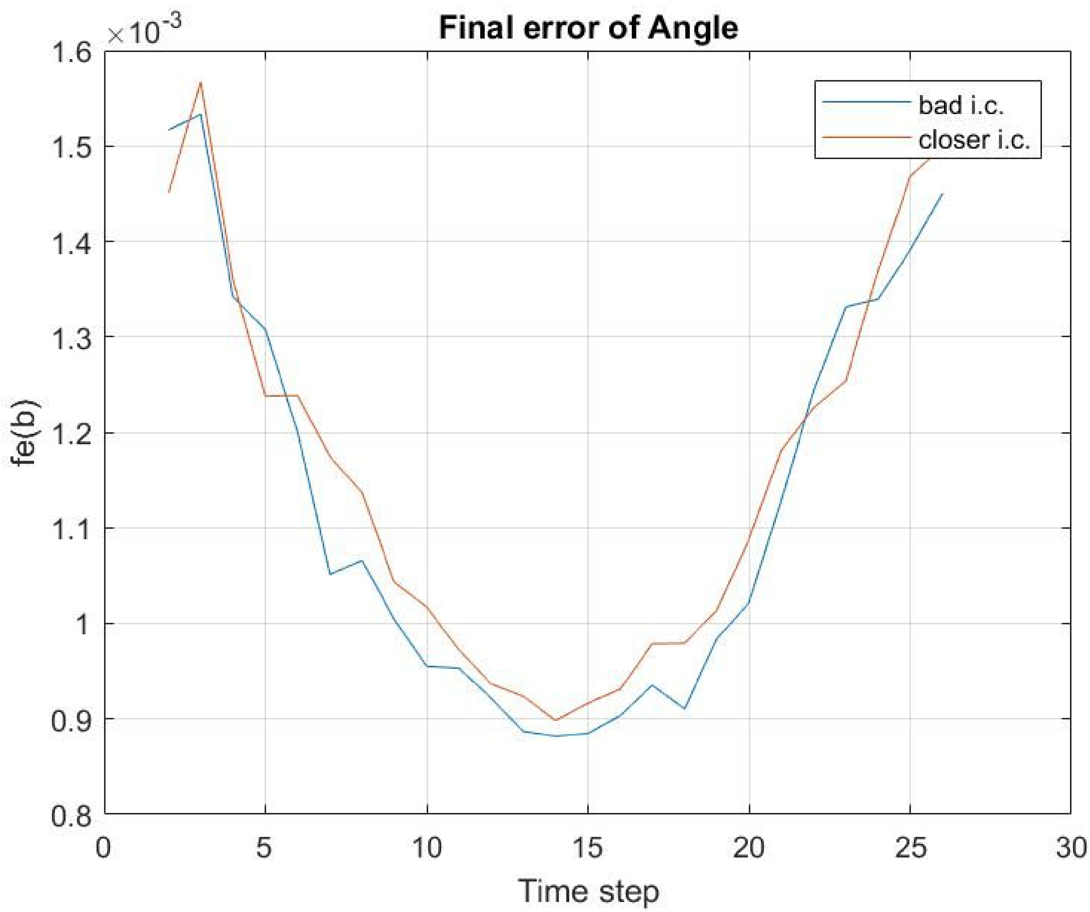


x(b) = D*sin(b) - linear
2500
2000
1500
1000
500
0
-500
x
-0.2
0
0.2
0.4
0.6
0.8
b



Conditional number of covariance matrix
cond(R)
time
0
5
10
15
5
10
15
20
25


Final error of Distance
60
50
40
30
20
10
0
fe(D)
0
5
10
15
20
25
30
Time step
bad i.c.
closer i.c.

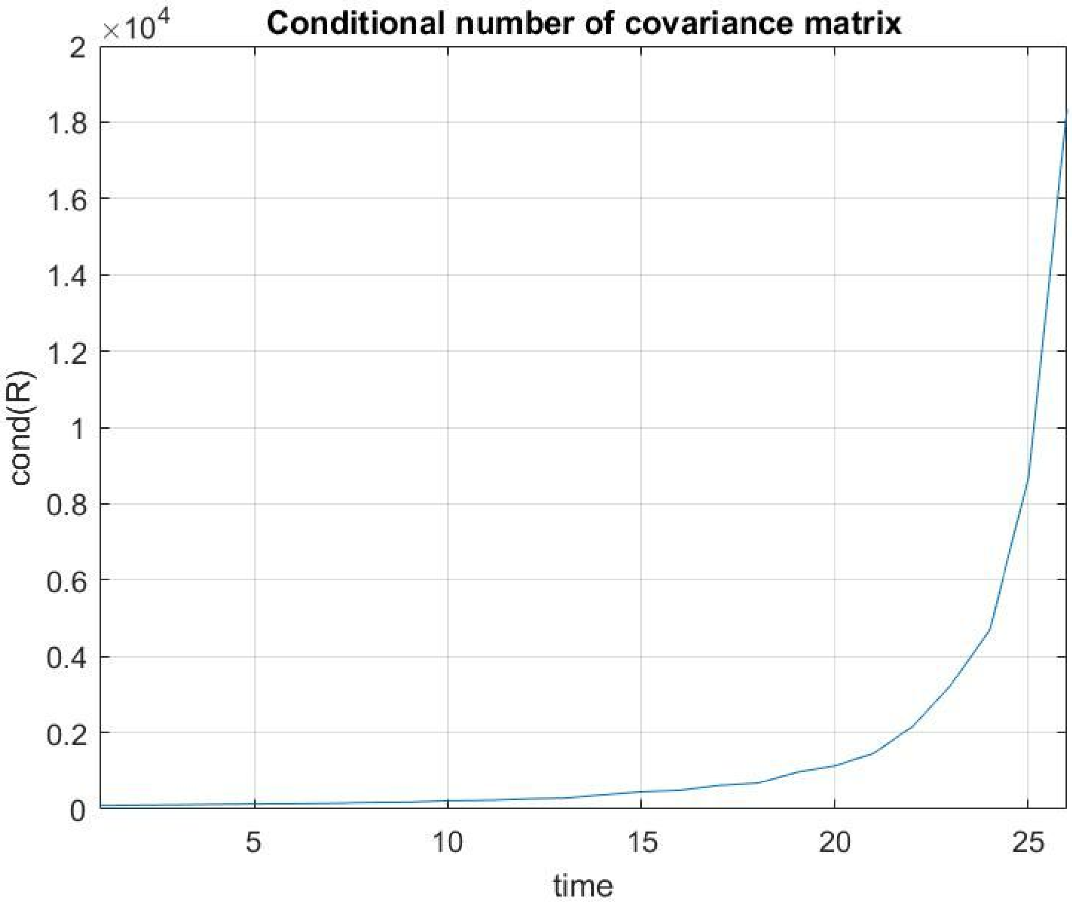


System may become blind and filter may diverge if:

1. too close initial conditions and linear model of estimation;
2. too small variance of angle for linear model.

Linearization errors have greater influence on the problem, as systems diverges in this case under any initial conditions. Decrease of variance of angle doesn't help to estimate angle better. In order to improve results we should decrease accuracy of angle measurements. This phenomenon is illustrated at the graphs. Conditional number of covariance matrix rapidly increases over time, this means that we have great accumulation of error.

In order to solve this problem it is possible to move measurement station far from the target object or even use device with lower accuracy of measurements.



Published with MATLAB® R2017b