Effect of Noise on Estimates

Prateek Garg

December 5th, 2021

Objective: To study the effect of normal distributed errors in measurements and the performance of $\boldsymbol{\alpha}-\boldsymbol{\beta}-\boldsymbol{\gamma}$ Filter on such measurements.

Implementation:

Three scripts are used : System.m, Filter.m, Experiment.m. System generates true values based on given inputs, Filter returns the estimates based on measurements and Experiment use values returned by the system and add noise which is given input to the filter which returns estimates to be plotted by experiment.m . Additionally a metric of performance is created which judges how fast the estimates reach an upper threshold(epsilon=0.1) percentage error of the true value. First 500(n) samples are taken for the experiment.

System:

```
function [X,V,A] = System(x,v,a,t,n)
X=zeros(1,n);
V=zeros(1,n);
A=zeros(1,n);
A=A+a;

X(1)=x;
V(1)=v;

for i=1:n-1
    X(i+1)=X(i)+V(i)*t+A(i)*t*t/2;
    V(i+1)=V(i)+A(i)*t;
end

end
```

Filter:

```
function [x,v,a,x_est,v_est,a_est] = Filter(alph,bet,gamm,t,z
,x_guess,v_guess,a_guess)
n=size(z,2);

alpha=zeros(1,n);
beta=zeros(1,n);
gamma=zeros(1,n);

alpha=alpha+alph;
beta=beta+bet;
gamma=gamma+gamm;

x=zeros(1,n); %range estimates of aircraft post measurement
v=zeros(1,n); %velocity estimates of aircraft post measurement
a=zeros(1,n); %acceleration estimates of aircraft post measurement

x_est=zeros(1,n); %range estimates of aircraft before measurement
v_est=zeros(1,n); %velocity estimates of aircraft before measurement
a_est=zeros(1,n); %acceleration estimates of aircraft before measurement

t_samp=t;
%Estimate initial values
x_temp=x_guess+v_guess*t_samp+a_guess*t_samp*t_samp/2;
v_temp=v_guess+a_guess*t_samp;
a_temp=a_guess;

for i = 1:n
    %State Extrapolation Equations:
    x_est(i)=x_temp;
    v_est(i)=v_temp;
    a_est(i)=a_temp;

    %Present estimate after measurement
    x(i)=x_temp+alpha(i)*(z(i)-x_temp);
    v(i)=v_temp+beta(i)*(z(i)-x_temp)/t_samp;
    a(i)=a_temp+2*gamma(i)*(z(i)-x_temp)/t_samp/t_samp;
```

```
    x_temp=x(i)+v(i)*t_samp+a(i)*t_samp*t_samp/2;
    v_temp=v(i)+a(i)*t_samp;
    a_temp=a(i);
end

end
```

Experiment:

```
%% Initialization
clear ; close all; clc
format ShortG ;

t_samp=5; %track-to-track interval
n=500; %Number of Samples
epsilon=0.01; %To check for convergence

t=linspace(t_samp,t_samp*n,n); %time values
%% ==================== Static System ====================
true_val=1000;
[X,~,~]=System(true_val,0,0,t_samp,n);
Z = X+randn(1,n);

alpha=1:n;
alpha=1./alpha;

val_guess=998;

[x,~,~,~,~,~]=Filter(alpha,0,0,t_samp,Z,val_guess,0,0);

%[t./t_samp;Z;x] %#ok<NOPTS>
figure
plot(t,X,t,Z,t,x,'linewidth',1.5);
legend("True Value","Measurements","Estimates");

E=((X-x)./X)*100;
figure
plot(t,E,'linewidth',1.5);
```

```
legend("Error Percentage");

disp(find(E>=epsilon,1,'last'));

fprintf('Program paused. Press enter to continue.\n');
pause;
close all;
%% ====================== Constant Velocity ======================
x_init=30000;
v_init=40;
[X,~,~]=System(x_init,v_init,0,t_samp,n);

Z = X+400*randn(1,n);

alpha=0.2;
beta=0.1;

x_guess=30000;
v_guess=40;
%Filter(alpha,beta,gamma,t,z ,x_guess,v_guess,a_guess)
[x,~,~,x_est,~,~]=Filter(alpha,beta,0,t_samp,Z,x_guess,v_guess,0);

%[t./t_samp;X;Z;x;v;x_est;v_est] %#ok<NOPTS>
figure
plot(t,X,t,Z,t,x,t,x_est,'linewidth',1.5);
%plot(t,X,'linewidth',1.5);

legend("True Value","Measurements","Estimates","Predictions");
E=((X-x)./X)*100;
figure
plot(t,E,'linewidth',1.5);
legend("Error Percentage");

disp(find(E>epsilon,1,'last'));
fprintf('Program paused. Press enter to continue.\n');
pause;
close all;
%% =================== Constant Acceleration ===================
x_init=30000;
```

```
v_init=40;
a_init=zeros(1,n);
n2=40;
a_init(n2:end)=80;

[X,V,A]=System(x_init,v_init,a_init,t_samp,n);

Z = X+1000*randn(1,n);

alpha=0.2;
beta=0.1;
gamma=0.01;

x_guess=60000;
v_guess=40;
a_guess=0;
%Filter(alpha,beta,gamma,t,z ,x_guess,v_guess,a_guess)
[x,v,a,x_est,v_est,a_est]=Filter(alpha,beta,gamma,t_samp,Z,x_guess,v_gue
ss,a_guess);

%[t./t_samp;X;Z;x;v;x_est;v_est] %#ok<NOPTS>
figure
plot(t,X,t,Z,t,x,t,x_est,'linewidth',1.5);
legend("True Value","Measurements","Estimates","Predictions");

figure
plot(t,V,t,v,t,v_est,'linewidth',1.5);
legend("True Value","Estimates","Predictions");

figure
plot(t,A,t,a,t,a_est,'linewidth',1.5);
legend("True Value","Estimates","Predictions");

E=((X-x)./X)*100;
figure
plot(t,E,'linewidth',1.5);
legend("Error Percentage");

disp(find(E>epsilon,1,'last'));
```

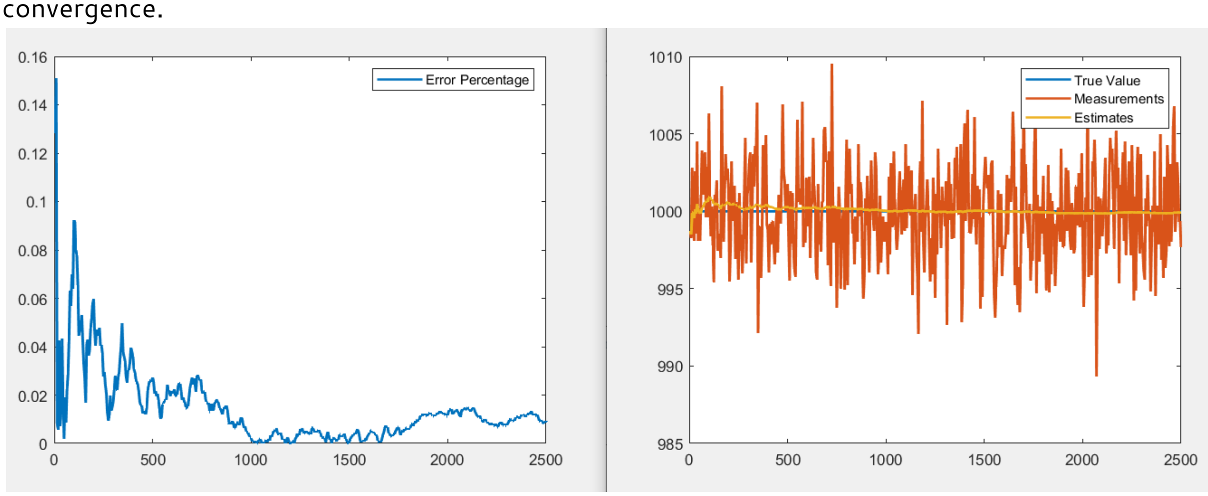
```
fprintf('Program paused. Press enter to continue.\n');
pause;
close all;
```

Observations:

Static System:

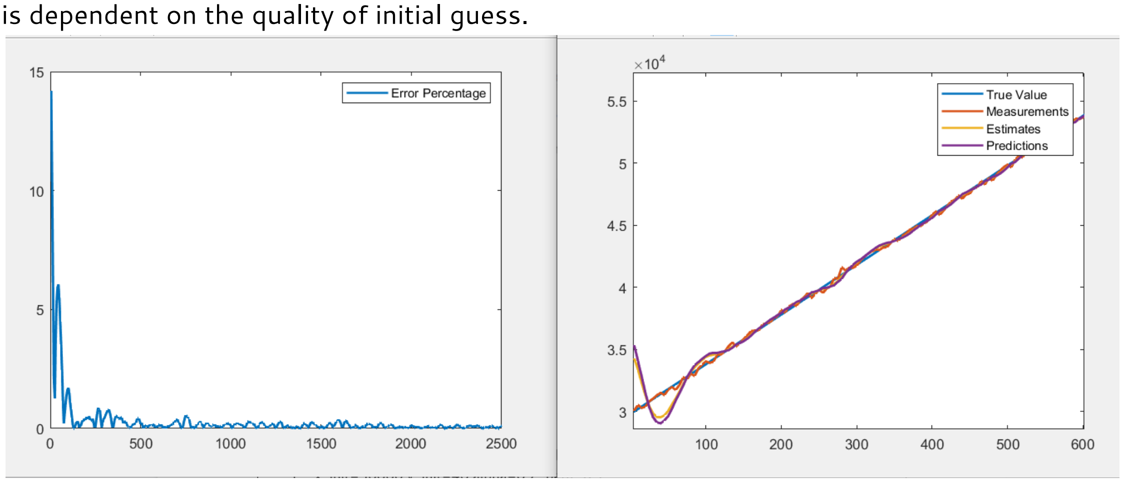
1.The value of initial guess has no effect in convergence because it is eliminated in the calculation of first estimate.

2. On the errors with standard deviation=3 The errors converge at n=2. Which indicates a very fast convergence.



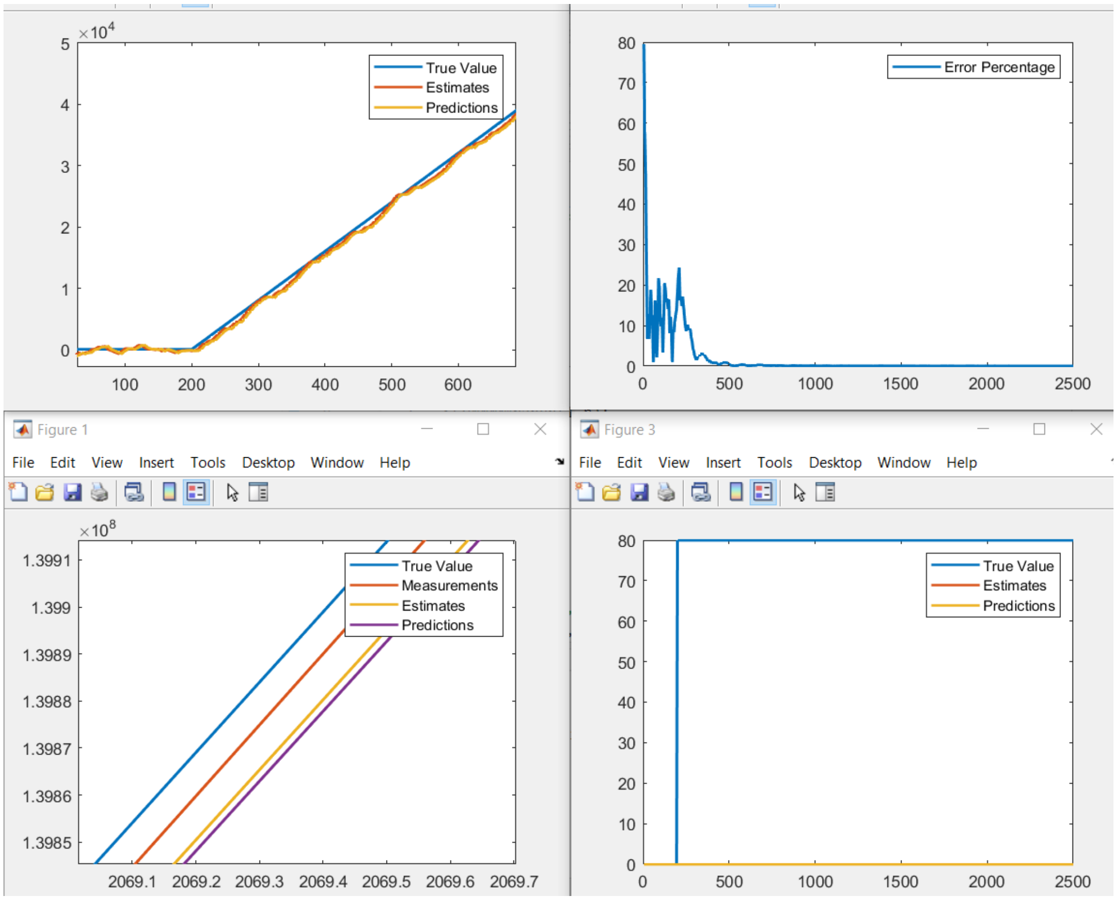
Dynamic System with constant rate of change:

1. x_init=30000;v_init=40;alpha=0.2; beta=0.1; x_guess=35000;v_guess=60;Variance=40000;
2. Despite a very poor initial guess, the range estimates converge to the true value but fail to converge within the threshold in the first 500 measurements. The rate of this convergence is dependent on the quality of initial guess.



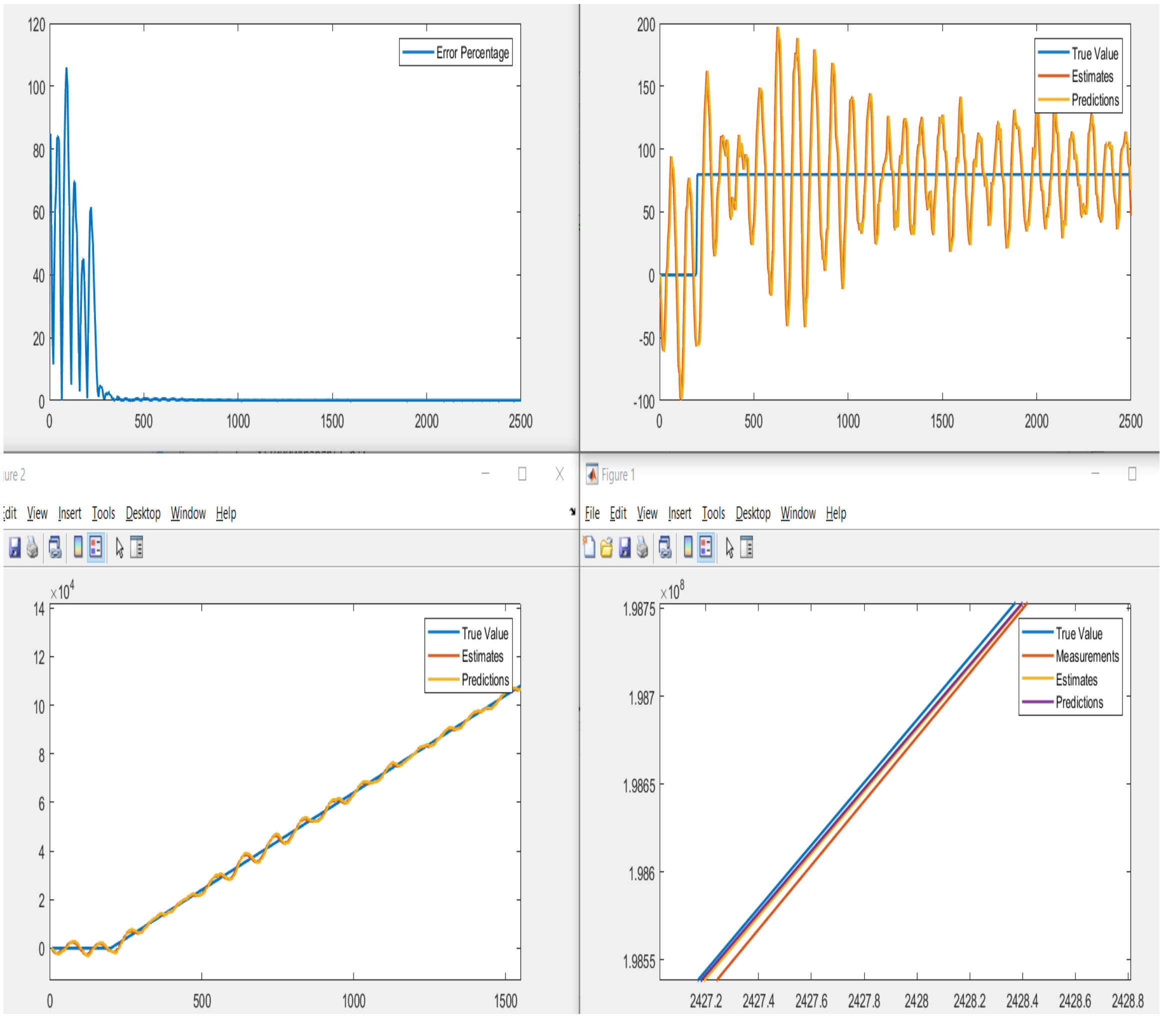
Dynamic System with rate of change changing with a constant rate of change:

1. On alpha-beta filter(gamma=0):Variance=100000000; alpha=0.2;beta=0.1;x_guess=60000;v_guess=40;a_guess=0;



   Despite percentage error decreasing there is a bias in the range estimates against true values because the acceleration estimates are zero. To eliminate this bias, alpha-beta-gamma filter is used.

2. On alpha-beta-gamma filter:Variance=100000000; alpha=0.2;beta=0.1;gamma=0.01;x_guess=60000;v_guess=40;a_guess=0;



The estimates of acceleration doesn't converge since it largely follows the distribution in error but velocity and range estimates converge very fast with no bias.