Understanding and Application of Kalman Filter

Kim June Mo

Dept. of Aerospace Engineering

Lab. For Navigation Control, and Applications

April 2, 2016

Contents

I Introduction

II Kalman Filter Algorithm

III Simulation

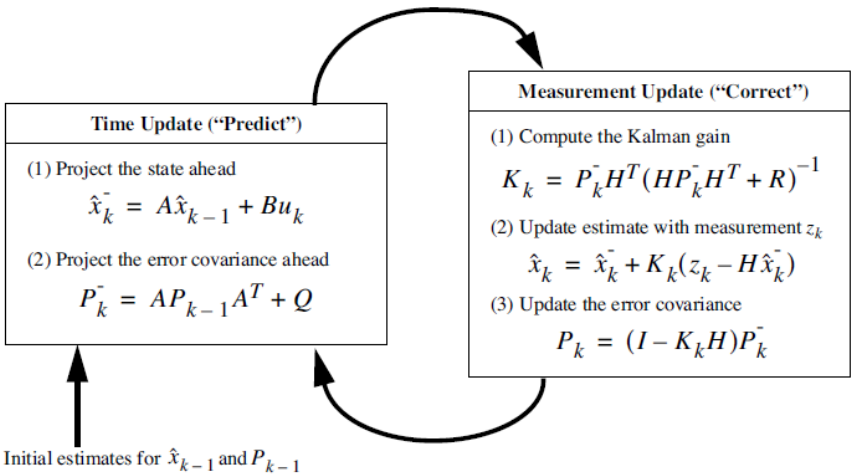
IV Future plans

Kalman Filter?

Rudolf E. Kálmán(1930∼)

- 1960년대 루돌프 칼만에 의해 개발
- Nosie가 포함된 역학 시스템 상태를 재귀필터를 이용해 참값과 가까운 값을 추적
- 신호처리,로봇 공학, 인공위성 등의 여러 분야에 사용
- 종류:
  Linear Kalman Filter
  Extended Kalman Filter(EKF)
  Uncented Kalman Filter(UKF)

Kalman Filter Recursive Algorithm



EKF(Extended Kalman Filter)

- $Ax_k \Rightarrow f(x_k)$
- $Hx_k \Rightarrow h(x_k)$

Kalman Filter Algorithm

Kalman Filter values

1)System model

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \qquad z_k = Hx_k + v_k$$

x: 상태변수(nx1) 열벡터

$A(= F, \Phi)$: 상태전이행렬(nxn)

B: Control matrix(nx1)

H:측정값과 상태 변수의 관계(mxn행렬)

Z: 측정값(mx1)행렬

Kalman Filter values

2)Linearization

Jacobian Matrix(비선형 모델 $\longrightarrow$ 선형 모델)

:n개의 변수를 가진 함수 m개를 모두 편미분 한 행렬

$$F_1(x_1,\ldots,x_n),\ldots,F_m(x_1,\ldots,x_n) \Rightarrow \mathbf{J} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

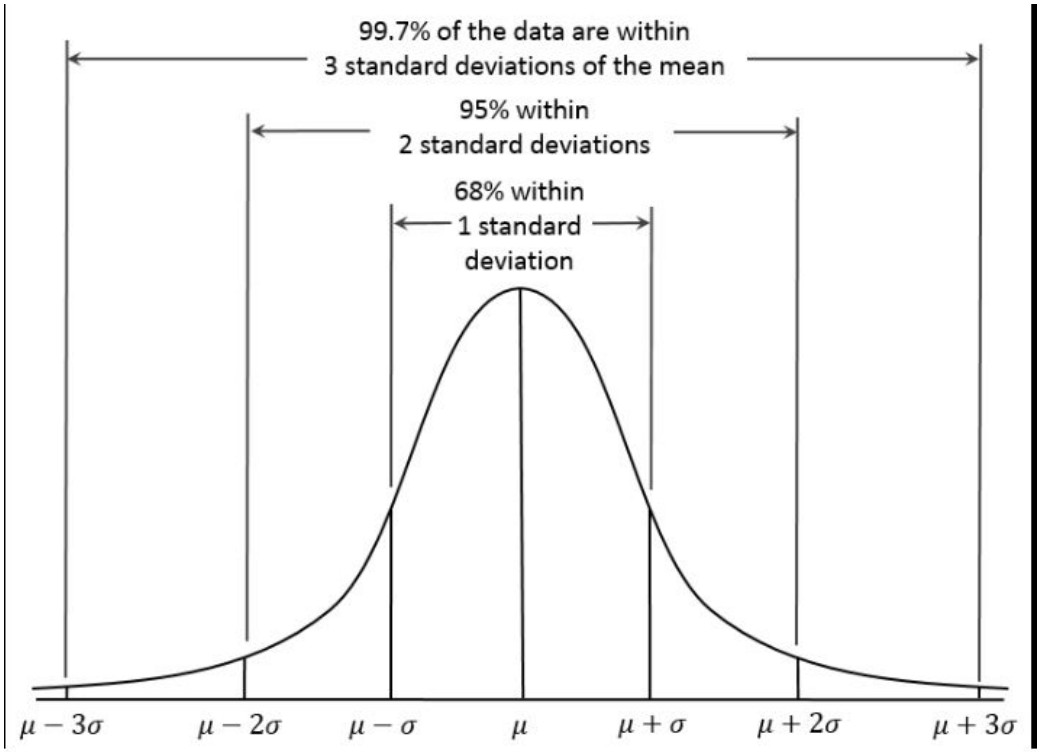
	KF	EKF
System	Ax_k	$f(x_k)$
Measurement	Hx_k	$h(x_k)$

$$A = \frac{\partial f}{\partial x}\Big|_{\hat{x}_k} \quad H = \frac{\partial h}{\partial x}\Big|_{\hat{x}_k^-}$$

Kalman Filter values

3)Noise

가우시안 노이즈: 평균이 0이고 표준편차가 1인 표준정규분포를 따르는 잡음



Kalman Filter values

4)오차공분산(P)

: 필터의 추정값과 참값의 차이를 나타내는 척도

$$P_k^- = AP_{k-1}A^T + Q$$

$$\Downarrow$$

$$P_k = (I - K_k H)P_k^-$$

- $P_k = E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\}$
- 오차공분산은 시간이 지날수록 줄어든다.
- 추정값은 오차가 충분히 작아지면 거의 줄어들지 않는다.

Kalman Filter values

5)Kalman gain

추정값을 계산할때 사용되는 가중치

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R_k}$$

a priori estimate error covariance

measurement error covariance

$$\lim_{R_k \to 0} K_k = H^{-1}$$

$$\lim_{P_k^- \to 0} K_k = 0$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$\hat{x}_k = H^{-1} z_k$$ observation

$$\hat{x}_k = \hat{x}_k^-$$ prediction

Example: Linear Filter Algorithm

등가속도운동(uniformly accelerated motion)

1.필터 초기값 설정

1)시스템 모델 ($X_k = AX_{k-1} + Bu$)

$$\boxed{s = s_0 + vt + \frac{1}{2}at^2 \quad v = v_0 + at \quad a = Const}$$

$$\dot{\mathbf{X}} = \begin{pmatrix} P \\ V \\ a \end{pmatrix} \quad \mathbf{X_k} = \begin{pmatrix} 1 & dt & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} X_{k-1} + \begin{pmatrix} \frac{dt^2}{2} \\ dt \\ 0 \end{pmatrix} X(3)$$

Example: Linear Filter Algorithm

등가속도운동(uniformly accelerated motion)

1.필터 초기값 설정

2)측정 모델 ($z_k = Hx_k + v_k$)

$$z_k = \begin{pmatrix} P \\ V \\ a \end{pmatrix} = \begin{pmatrix} x(1) + x(2)dt + x(3)dt^2/2 \\ x(2) + x(3)dt \\ x(3) \end{pmatrix} + v_k \quad H = eye(3)$$

v_k = 측정잡음 , $(mx1)$ 열벡터

Example: Linear Filter Algorithm

등가속도운동(uniformly accelerated motion)

1.필터 초기값 설정

3)Noise

$$Q = \begin{bmatrix} 0.09^2 & 0 & 0 \\ 0 & 0.02^2 & 0 \\ 0 & 0 & 0.01^2 \end{bmatrix} \quad R = \begin{bmatrix} 3^2 & 0 & 0 \\ 0 & 1.5^2 & 0 \\ 0 & 0 & 0.5^2 \end{bmatrix}$$

$$w_k = \sqrt{Q} * randn(3,1)$$

$$v_k = \sqrt{R} * randn(3,1)$$

$$P = 10 * eye(3)$$

Simulation with matlab

1)Initial values

```
1  dt=0.01;                                  %step size
2  t=0:dt:10;
3  x=[10 20 2]';                             %Initial values
4  xh=[0 0 0]';
5  A=[1 dt 0; 0  1  0; 0  0  1];             %State Matrix
6  B=[dt^2/2; dt; 0];                        %Control Matrix
7  P=10*eye(3);                              %
8  Q=diag([0.09^2 0.02^2 0.01^2]);           %
9  R=diag([3^2 1.5^2 0.5^2]);                %
10 H=eye(3);                                 %
```

Simulation with matlab

2)Measurement and Kalman Filter

```
1  for i=1:length(t)
2  x=rk4(x,dt);                                  %Runge kutta method
3  v=sqrt(R)*randn(3,1);                         %Measurement Noise
4  pm=x(1)+x(2)*dt+x(3)*dt^2/2+v(1);             %Position+noise
5  vm=x(2)+x(3)*dt+v(2);                         %Velocity+noise
6  am=x(3)+v(3);                                 %Acceleration+noise
7  z=[pm vm am]';                                %Measurement values
```

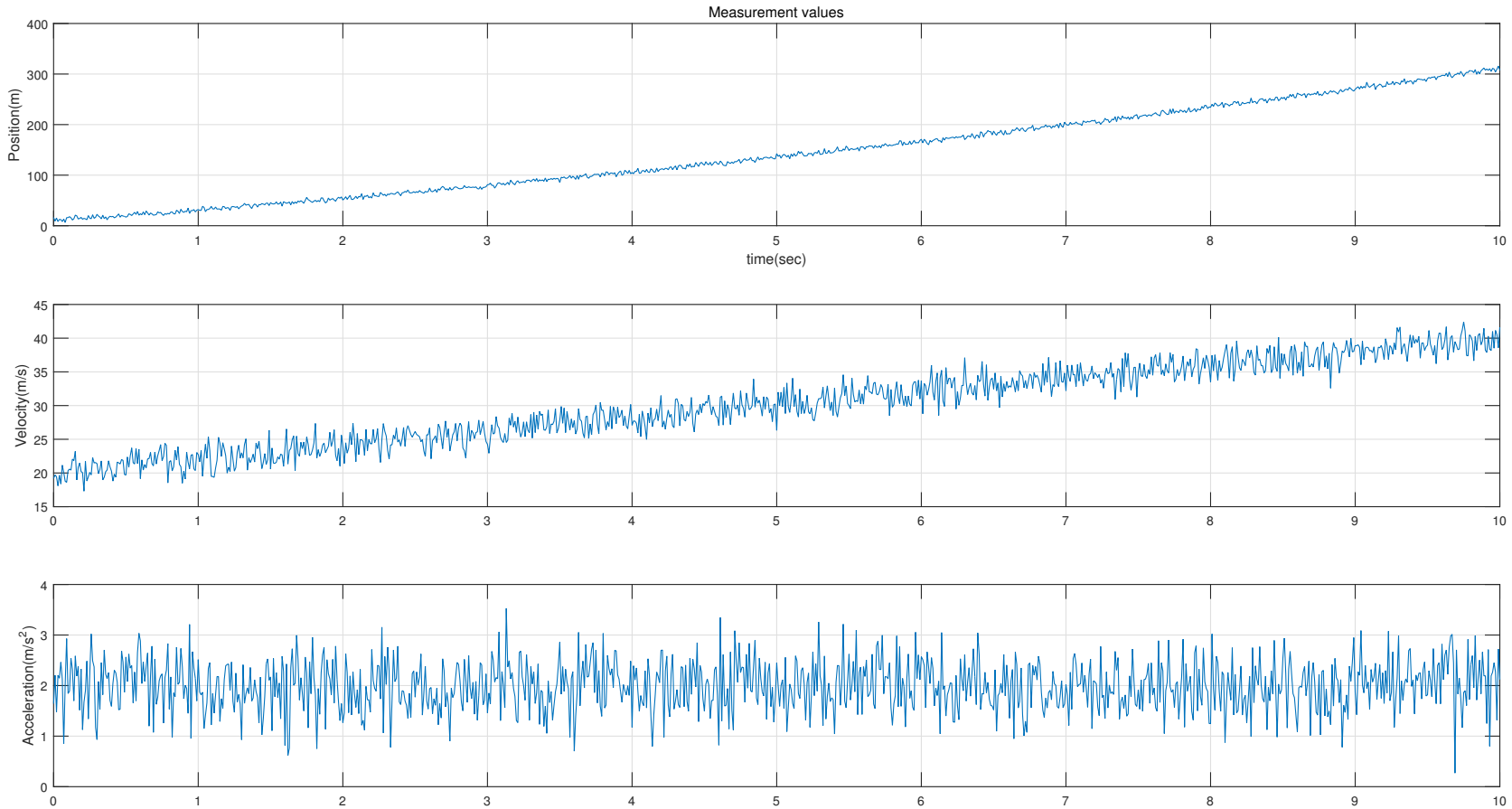
Simulation with matlab

2)Measurement and Kalman Filter

```
1  w=sqrt(Q)*randn(3,1); %System Noise
2  xh=A*xh+B*xh(3)+w;    %Project the state ahead
3  P=A*P*A' + Q;       %Project the error covariance ahead
4  K=P*H'*inv(H*P*H'+R); %Compute the Kalman Gain
5  xh=xh+K*(z-H*xh);     %Update the estimate via z
6  P=(eye(3)-K*H)*P;     %Update the error covariance
7  end
```

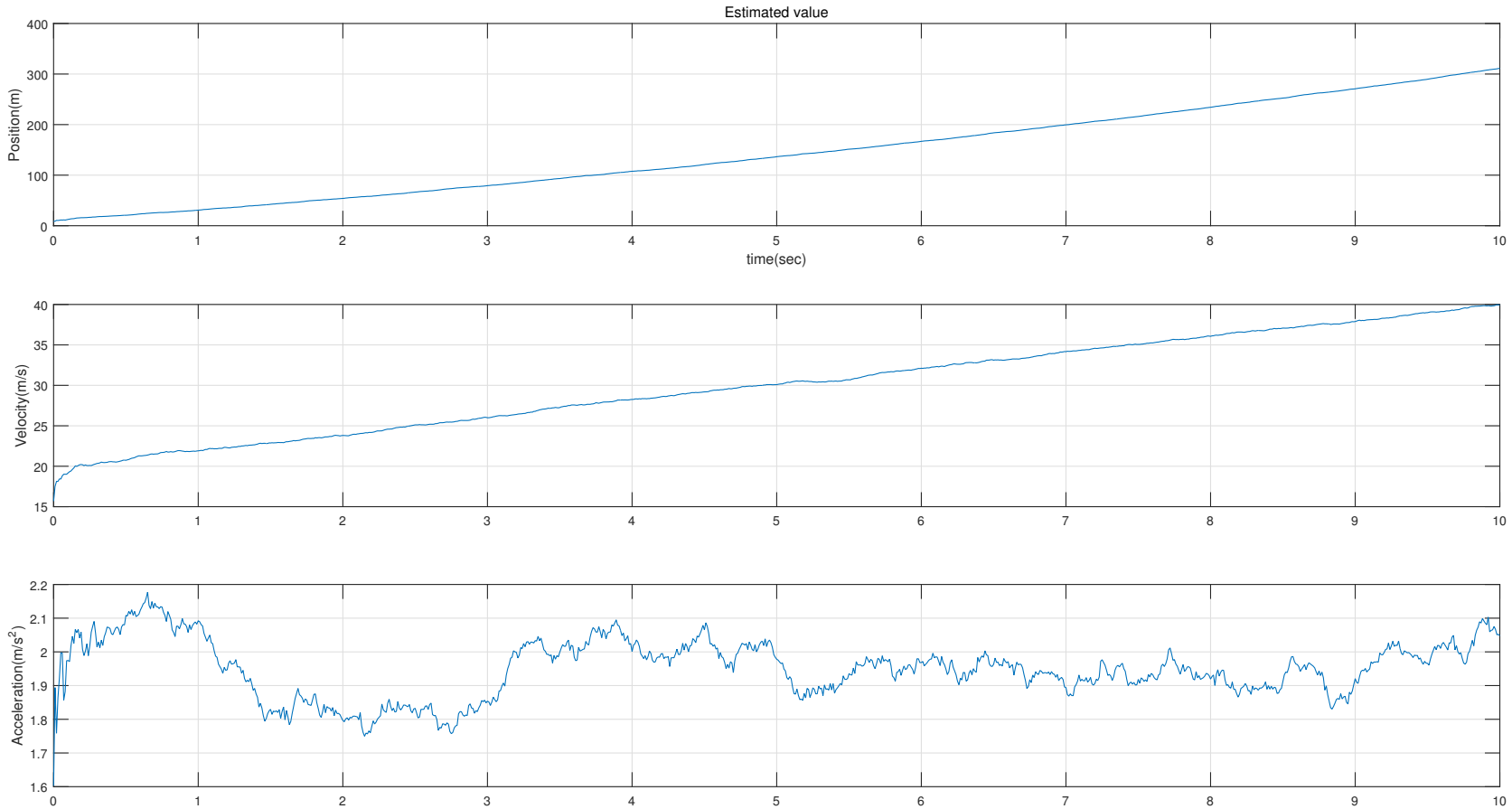
Simulation with matlab

3)Reasult



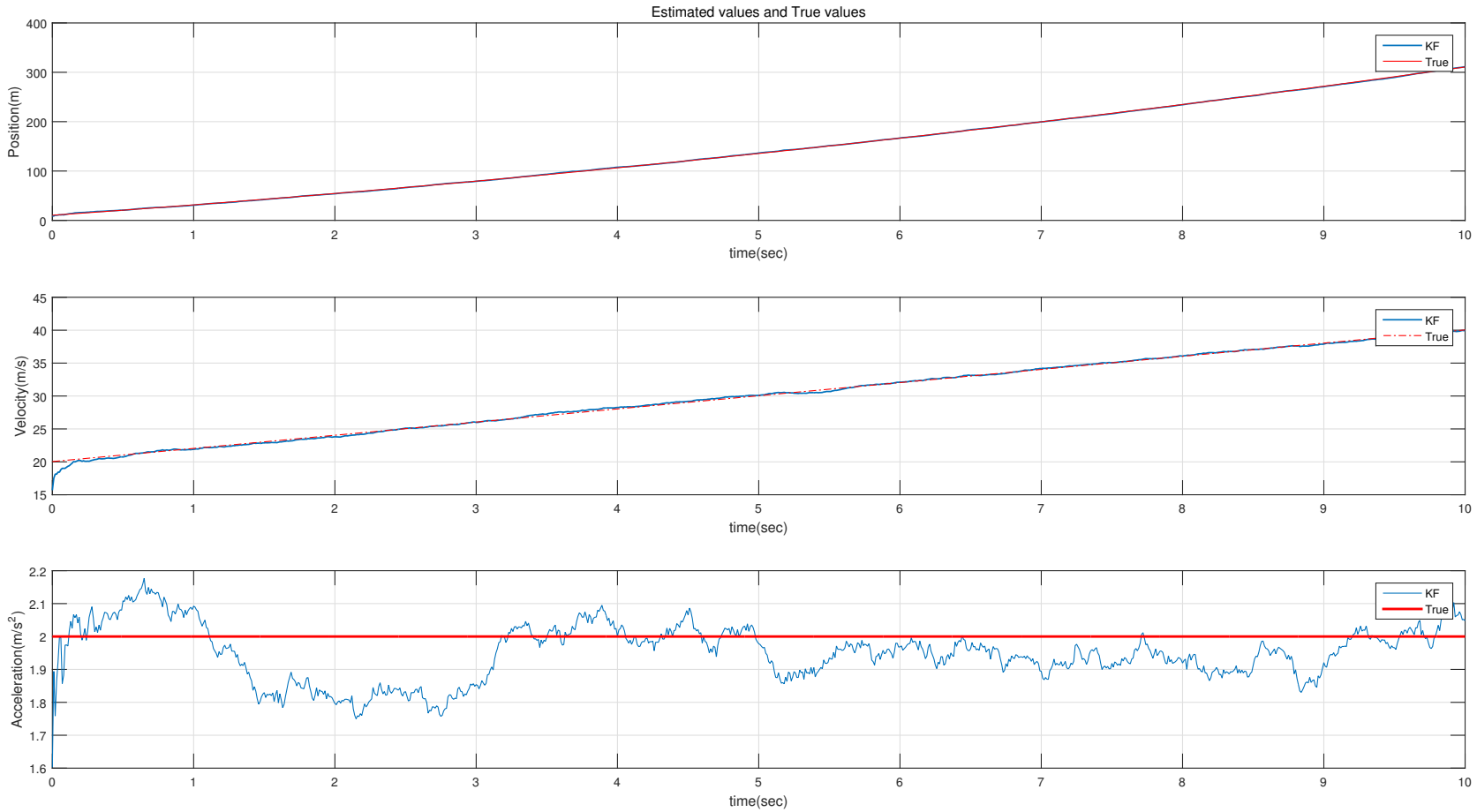
Simulation with matlab

3)Reasult



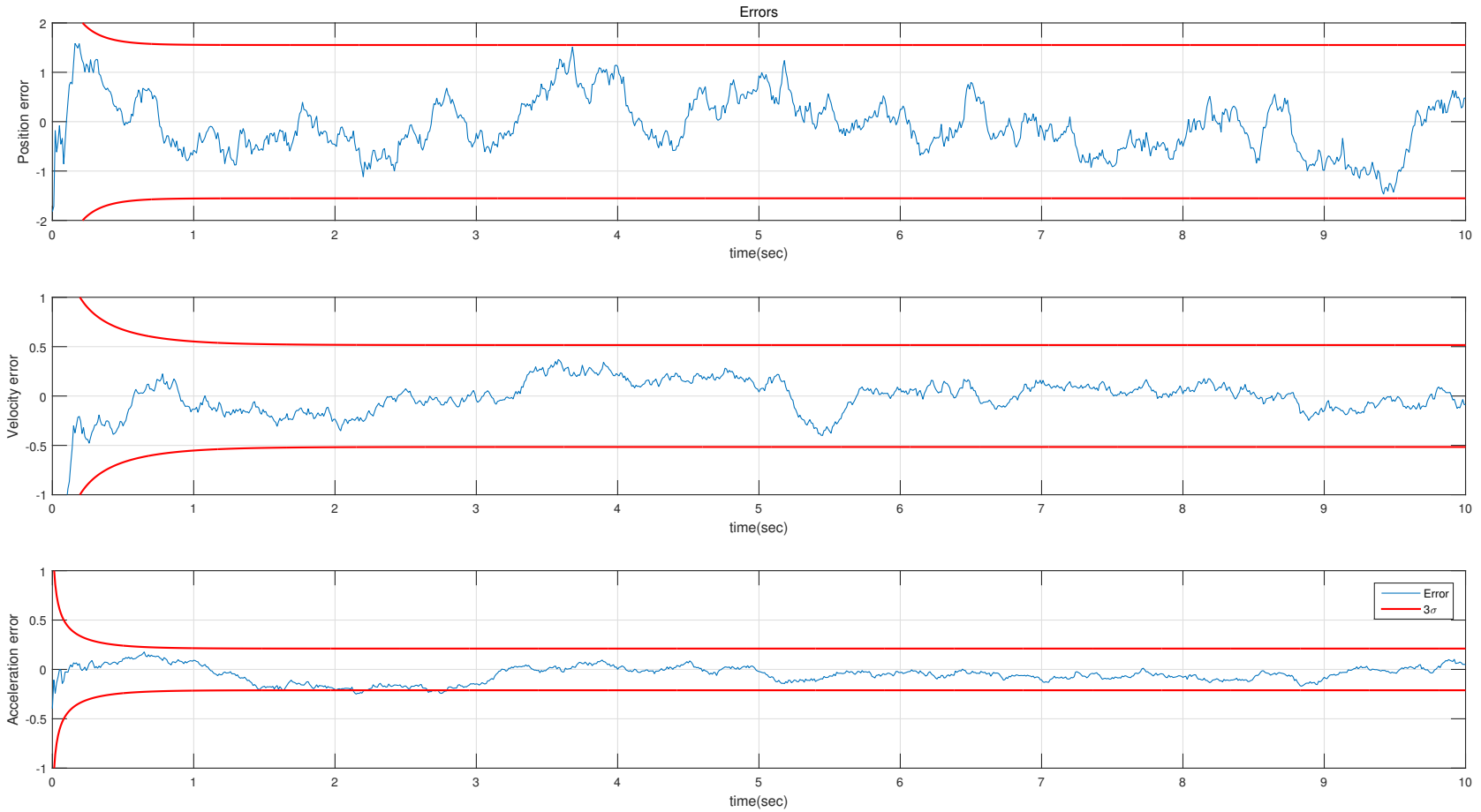
Simulation with matlab

3)Reasult



Simulation with matlab

3)Reasult



EKF Example (Van Der Pol's equation)

Van Der Pol's equation:

$$m\ddot{x} + 2c(x^2 - 1)\dot{x} + kx = 0$$

state variable:

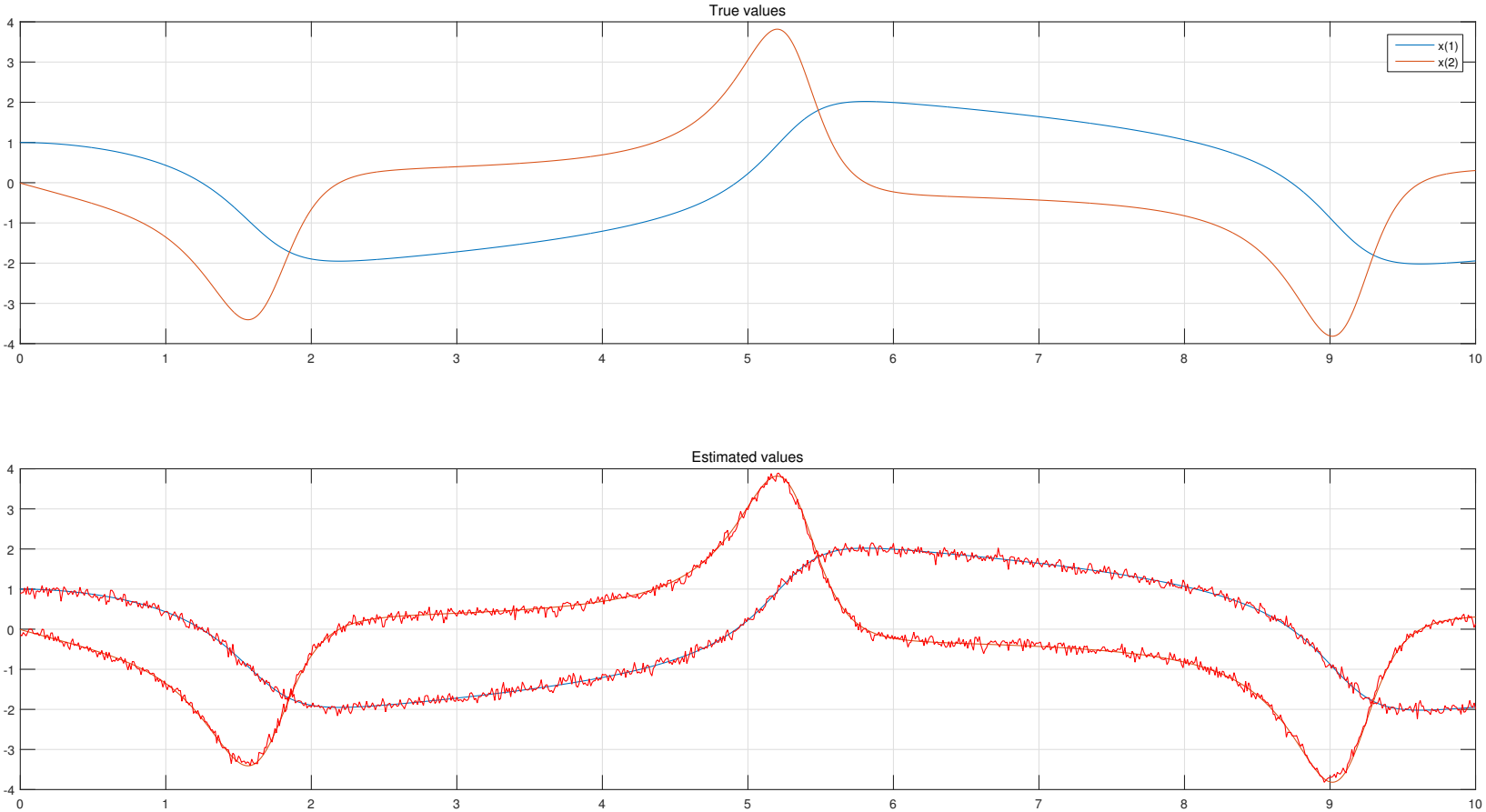
$$\mathbf{x} = [x\ \dot{x}]^T \rightarrow \dot{\mathbf{x}} = [\dot{x}\ \ddot{x}]^T$$

$$\dot{x_1} = x_2 \qquad \dot{x_2} = -2(c/m)(x_1^2 - 1)x_2 - (k/m)x_1$$

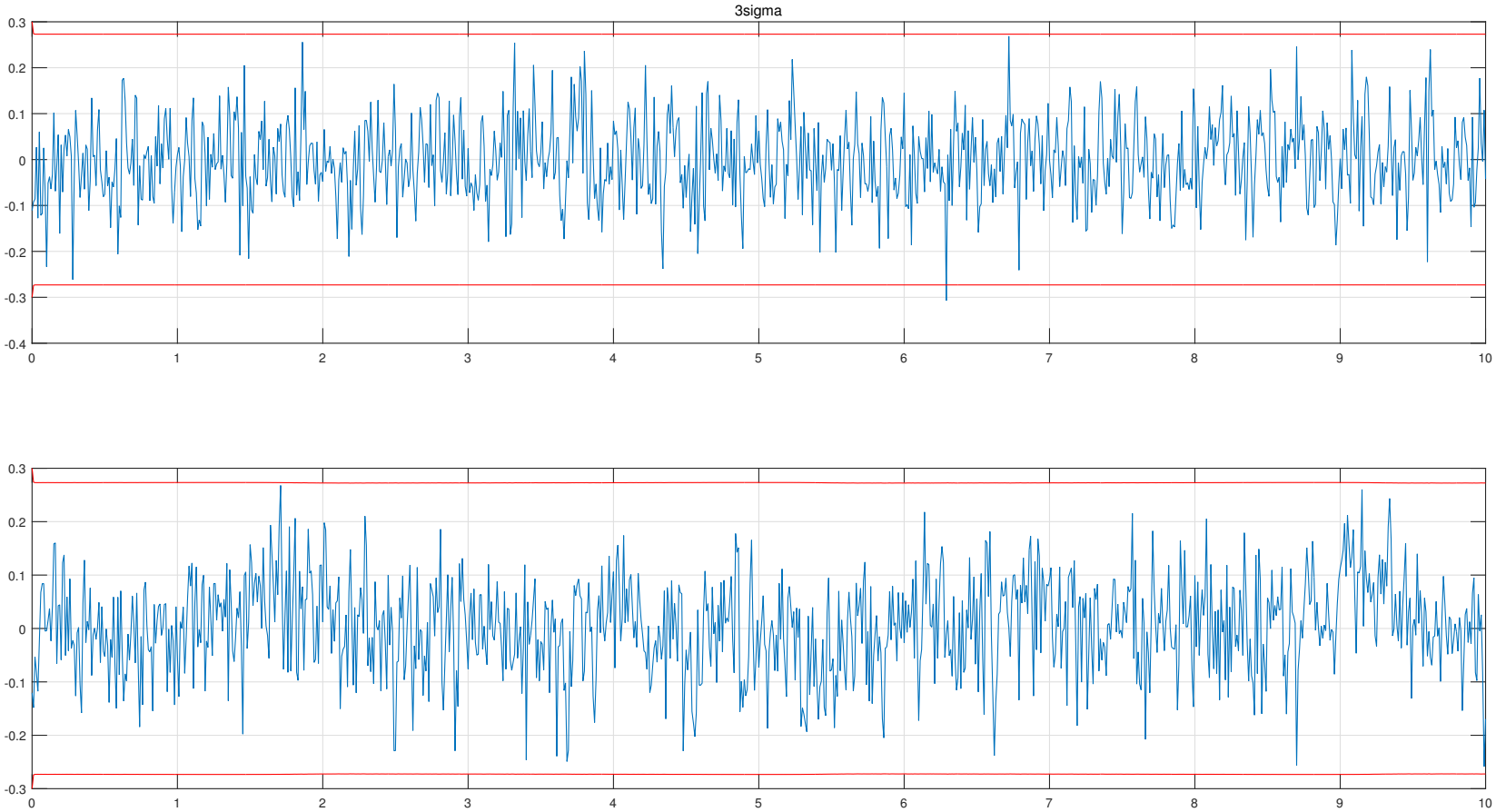
Thus, The linearized model is given by

$$F = \begin{bmatrix} 0 & 1 \\ -4(c/m)\hat{x}_1\hat{x}_2 - (k/m) & -2(c/m)(\hat{x}_1^2 - 1) \end{bmatrix}$$

EKF Example (Van Der Pol's equation)



EKF Example (Van Der Pol's equation)



EKF Example (Van Der Pol's equation)

문제점

```
1   F = eye(2)+[0 1;
2  -4*(c/m)*xh(1)*xh(2)-(k/m) ...
       -2*(c/m)*(xh(1)^2-1)]*dt;
3   xh = F*xh;
4  %xh=fx(xh,dt);
5  P = F*P*F'+ G*Q*G';
6  K=P*H'*(H*P*H'+R)^(-1);
7  xh=xh+K*(z-H*xh);
8  P=(eye(2)-K*H)*P;
```

- 이산시스템에 대한 개념 부족
- Discrete-Time, Continuous-Discrete 알고리즘 구현방법의 차이
- 코드 간결화 문제

Future plan

- EKF 알고리즘 및 정확한 시스템 모델을 이해, 동역학적인 문제 적용
- UKF 알고리즘 구현

Thank you for your attention!