Kalman Filter

Joshua Moskowitz

Machine Learning Seminar

May 24, 2016

Houston, we have a problem

Actual position of spacecraft???

$p = (x_a, y_a, z_a)$

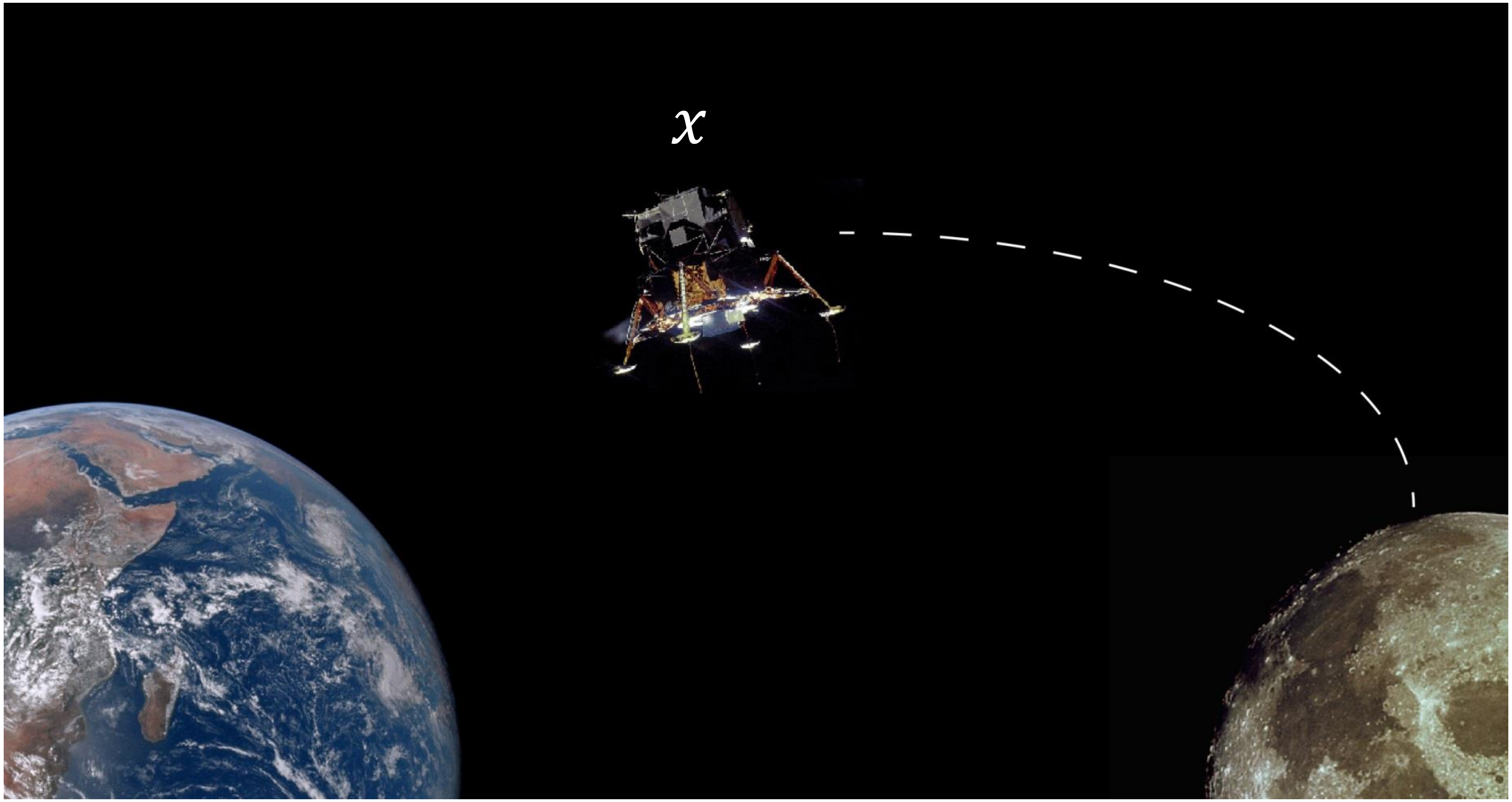
Inertial Measurement Unit in lander

$p = (x_d, y_e, z_f)$

Ground-based doppler radar

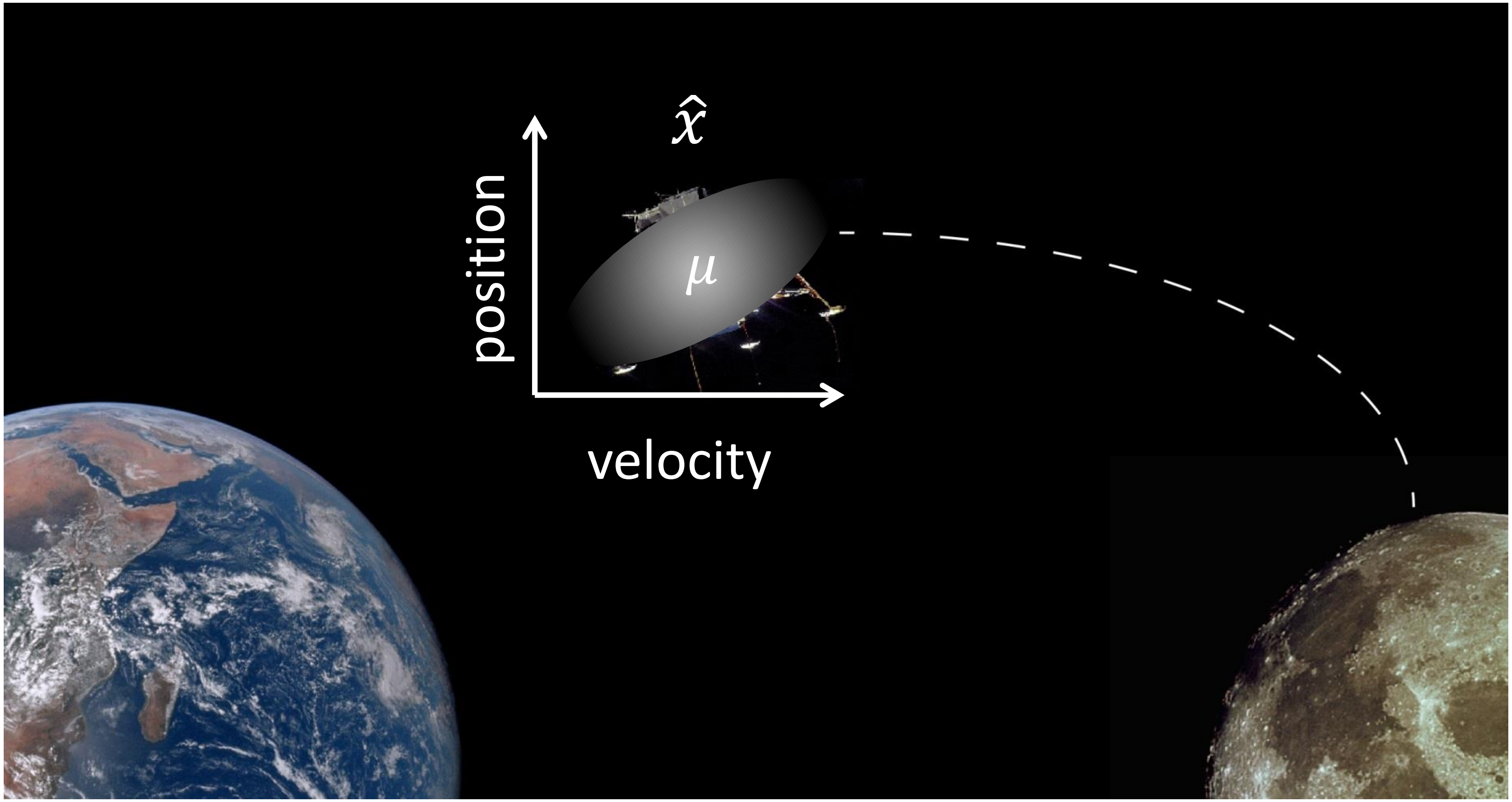
Overview

- Takes in the latest estimate of the state of a system and provides a best estimate of the current state given some measurement
- Best used when:
  - Need to **combine** multiple **noisy** sources of information
  - We have approximate knowledge of how a system changes over time ($p_k = p_{k-1} + \Delta t \times v_{k-1}$)
  - Noise is modeled by a Gaussian process



$$x = \begin{bmatrix} position \\ velocity \end{bmatrix}$$

Don’t forget, we’re working with matrices!

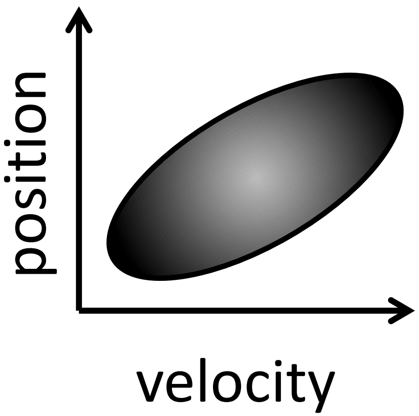


$$\hat{x} = \begin{bmatrix} position \\ velocity \end{bmatrix}$$

This is just an estimate of the state!

$$position = N(\mu, \sigma_p^2)$$

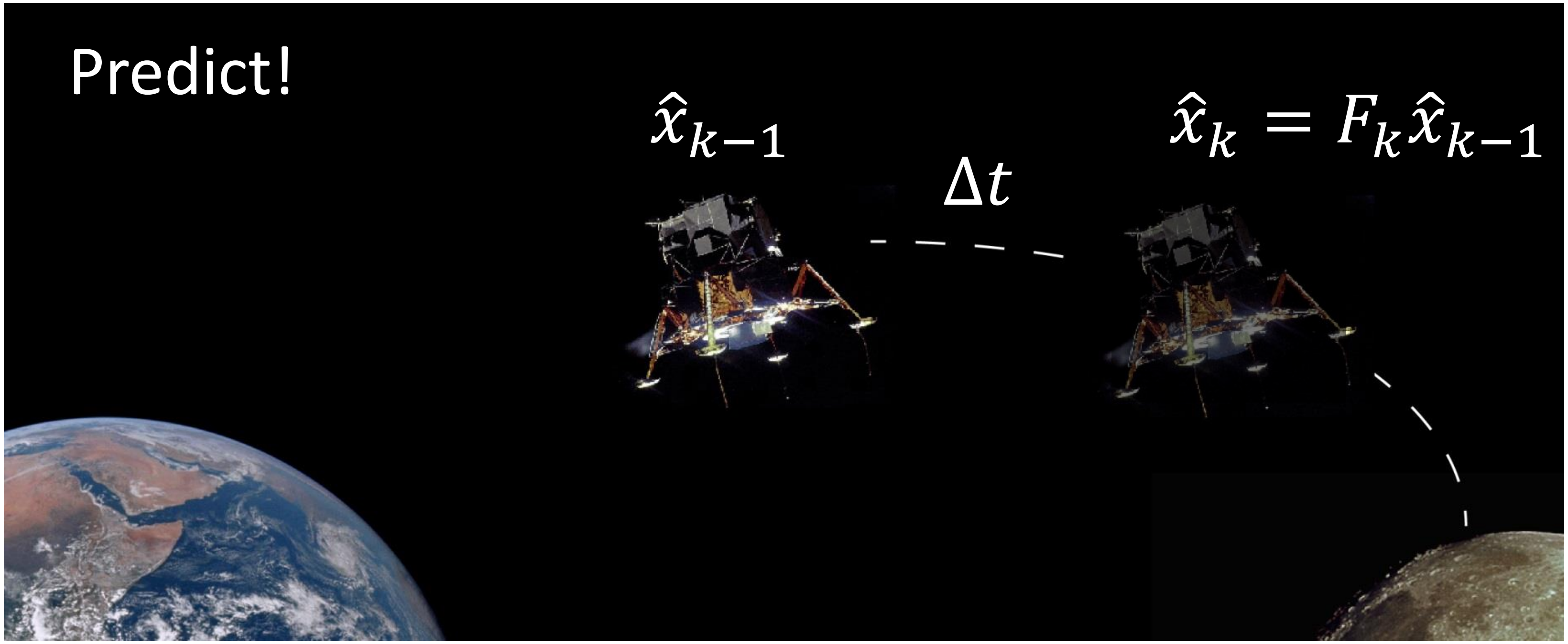
How can we use the variance?



$$P_k = \begin{bmatrix} \sum_{pp} & \sum_{pv} \\ \sum_{vp} & \sum_{vv} \end{bmatrix}$$

$$\sum_{ij} = COV(X_i, X_j) = \frac{\sum(X_i - \bar{X}_i)(X_j - \bar{X}_j)}{N - 1}$$

- We'll use the covariance to get a more accurate estimate of our current state
- $\boldsymbol{P_k}$ represents the accuracy of our state estimate

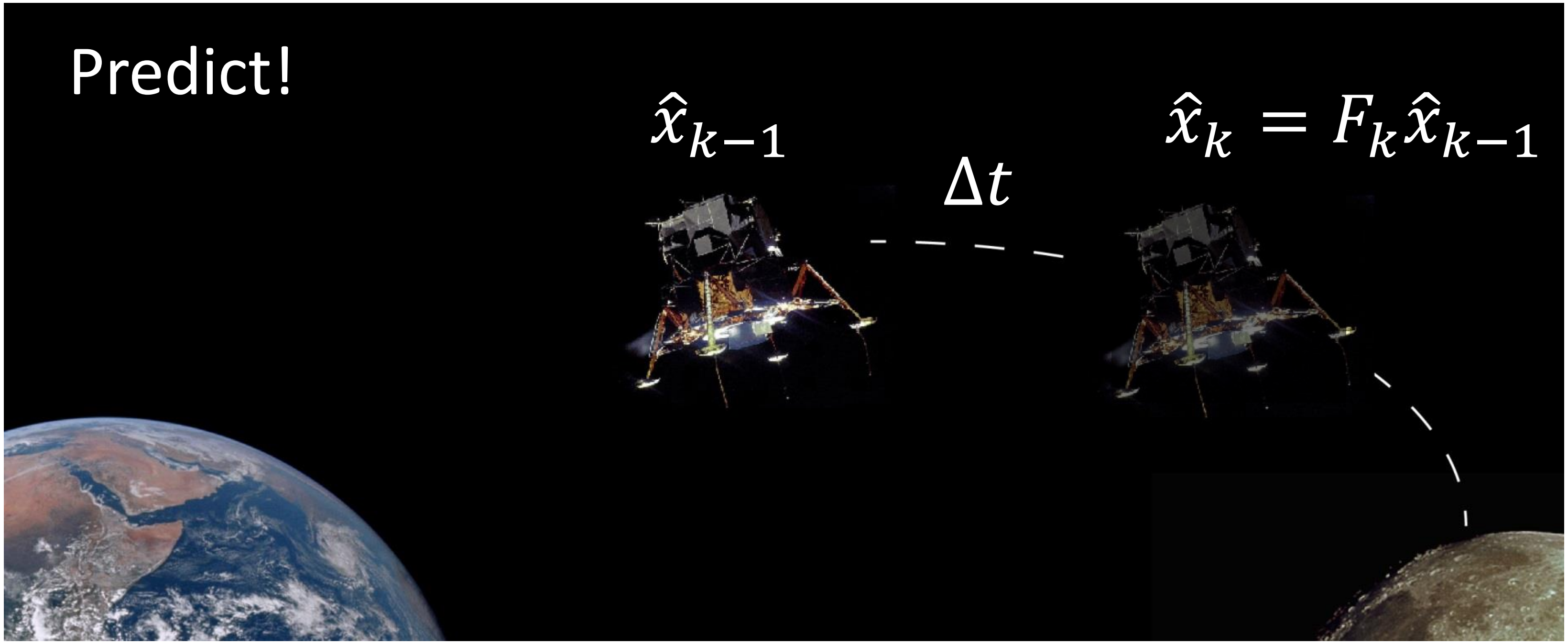


$$\hat{p}_k = \hat{p}_{k-1} + \Delta t \times \hat{v}_{k-1}$$

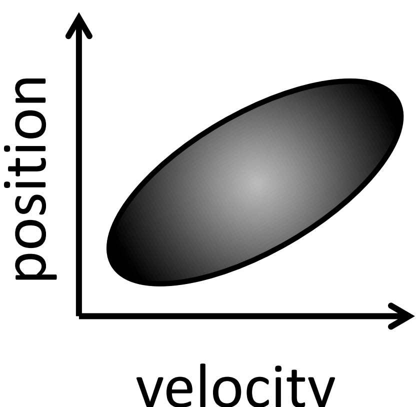
$$\hat{v}_k = \hat{v}_{k-1}$$

$$\hat{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \hat{x}_{k-1}$$

We will now apply a model $\boldsymbol{F_k}$ of how we believe our state will change by the next time time-step



Need to update the covariance estimate....



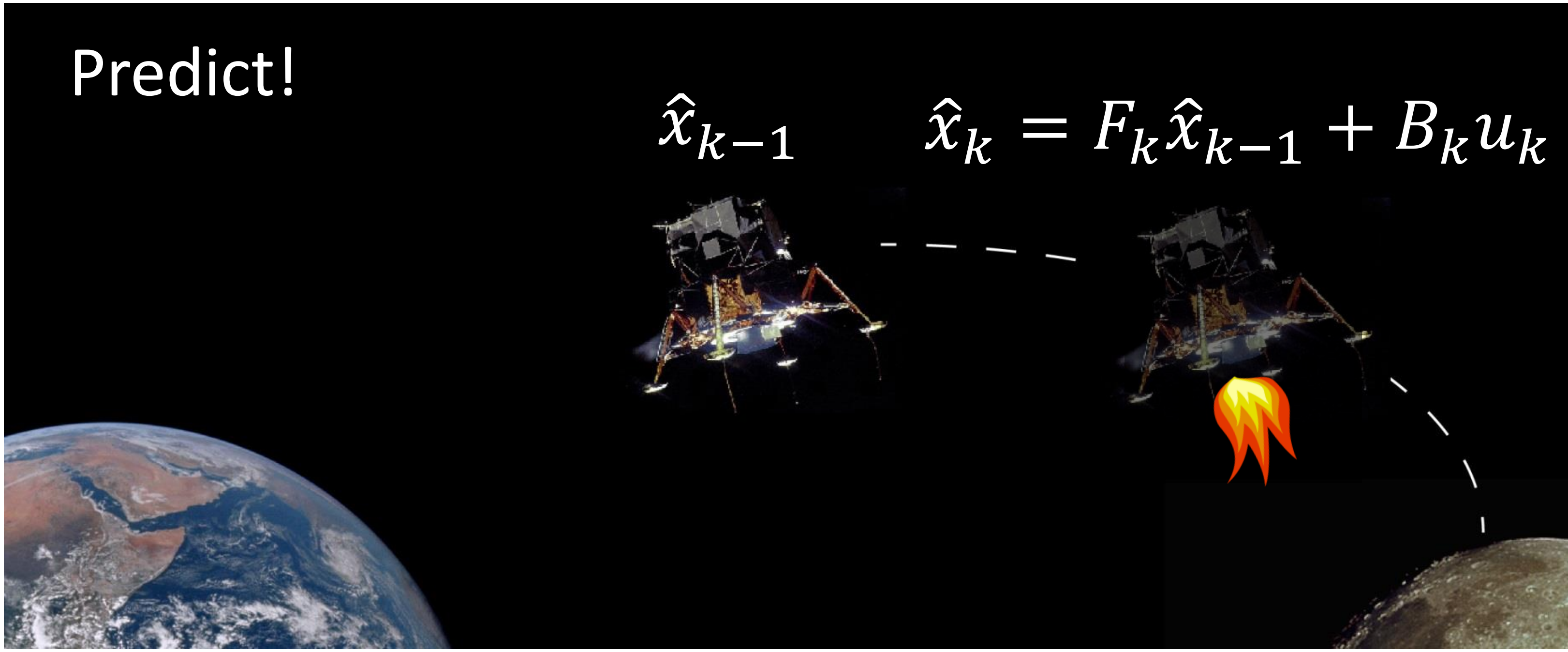
$$Cov\ Matrix(X) = Cov(X_i, X_j)$$

$$Cov\ Matrix(\boldsymbol{F_k}X) = Cov(\boldsymbol{F_k}X_i, \boldsymbol{F_k}X_j) = \boldsymbol{F_k}\mathbf{A}\mathrm{Cov}(X_i . X_j)\boldsymbol{F_k}^{\boldsymbol{T}}$$

$$P_k = F_k P_{k-1} F_k^T$$

How can we refine our prediction?

- Need to take into account *process noise*
  - forces acting on our system we don't know about (e.g. meteoroid impact)
  - Assumed to be drawn from a normal distribution
- What about input to the system?
  - Forces acting on our system that we DO know about! (e.g. thrusters being activated)
  - Need to apply a new model to convert this input to the resulting state



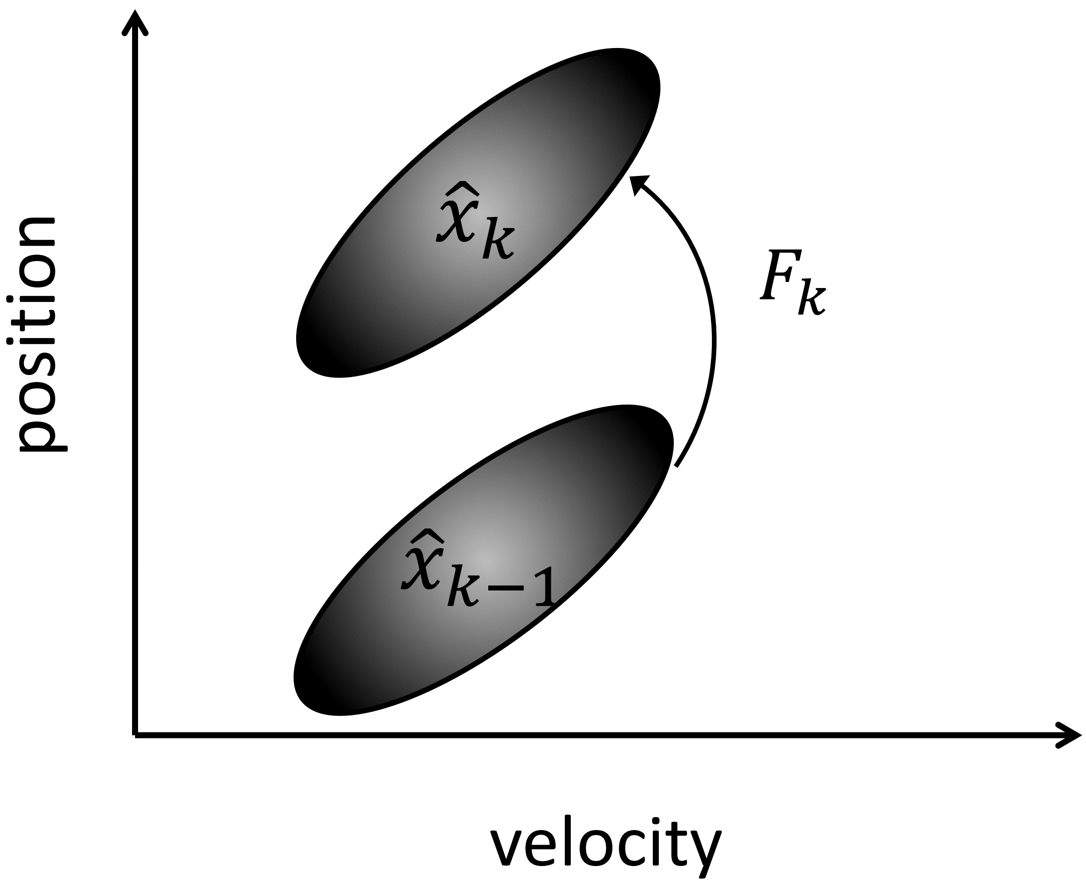
$$\hat{p}_k = \hat{p}_{k-1} + \Delta t \times \hat{v}_{k-1} + \frac{1}{2} a \Delta t^2$$

$$\hat{v}_k = \hat{v}_{k-1} + a\Delta t$$

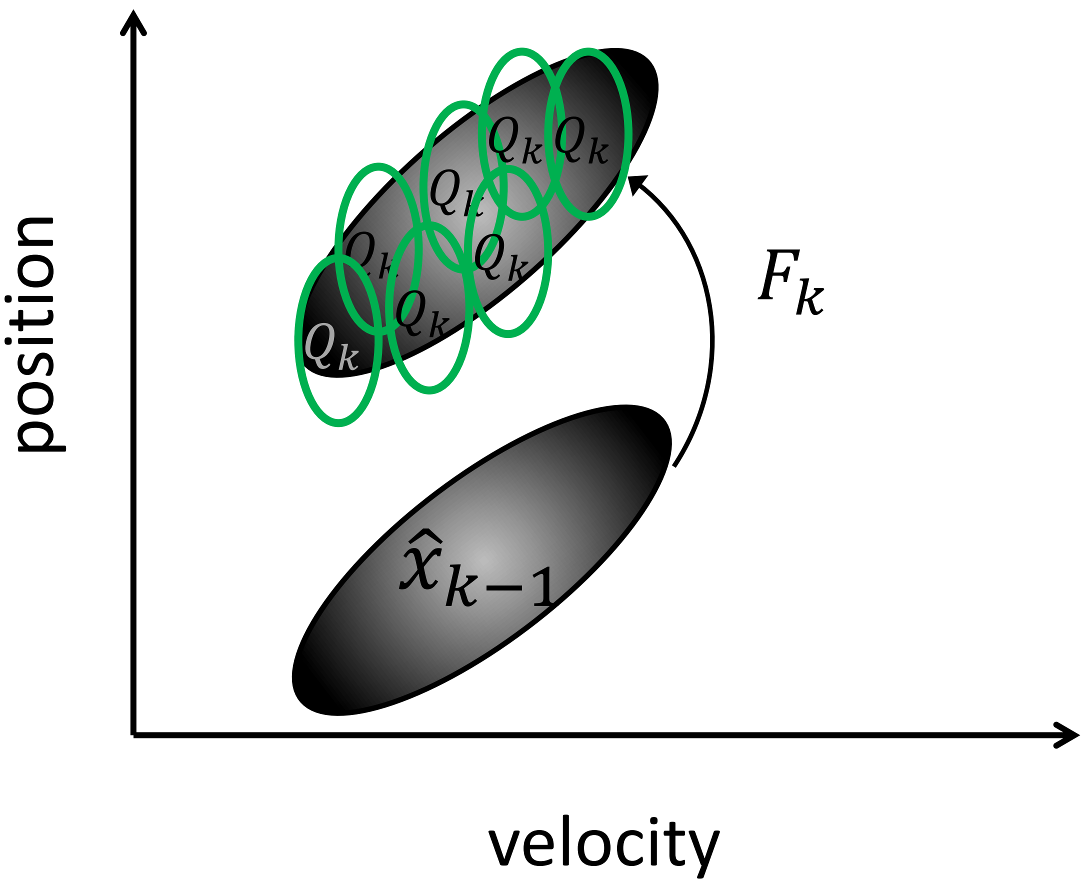
$$\hat{x}_k = F_k \hat{x}_{k-1} + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} a$$

We will now apply a model $\boldsymbol{B_k}$ of how we believe our input $\boldsymbol{u_k}$ will change our state

Process noise



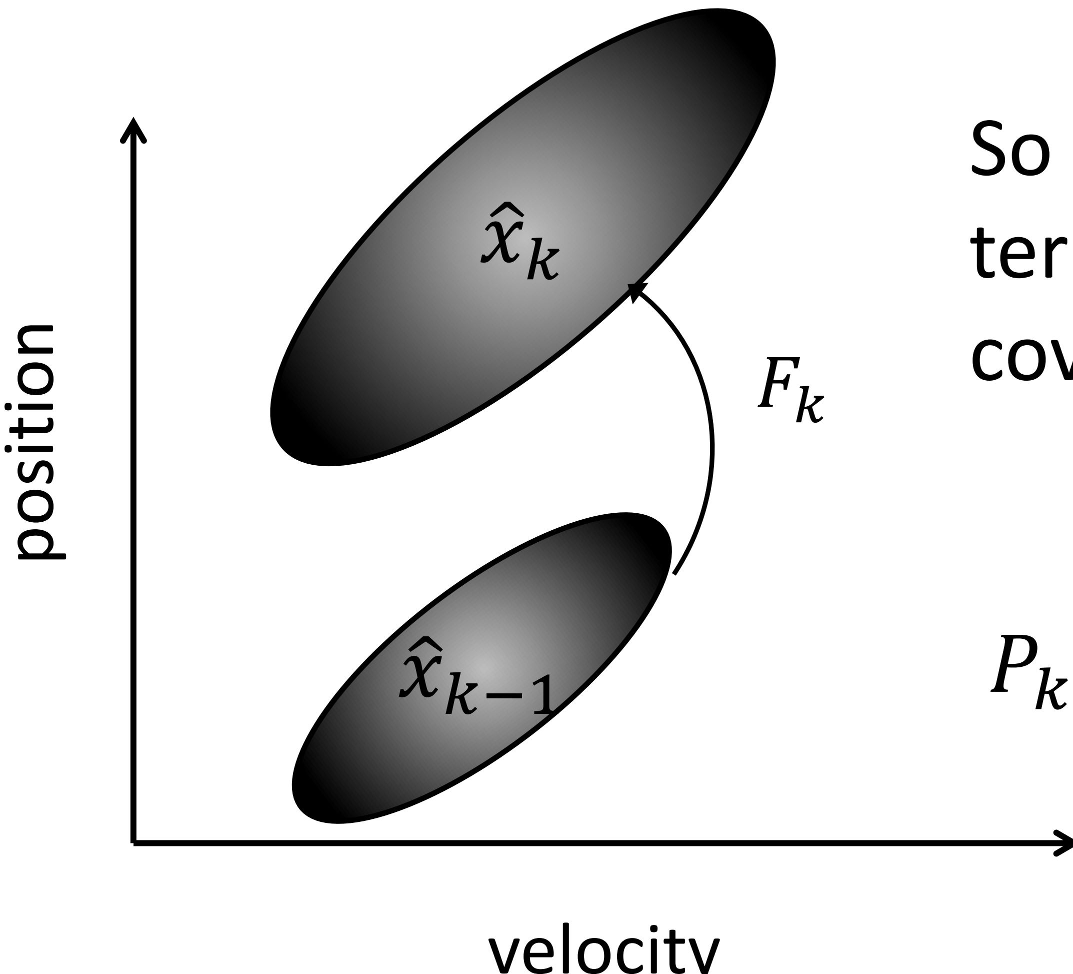
Process noise



Don't know what the noise is, assume it's drawn from a normal distribution

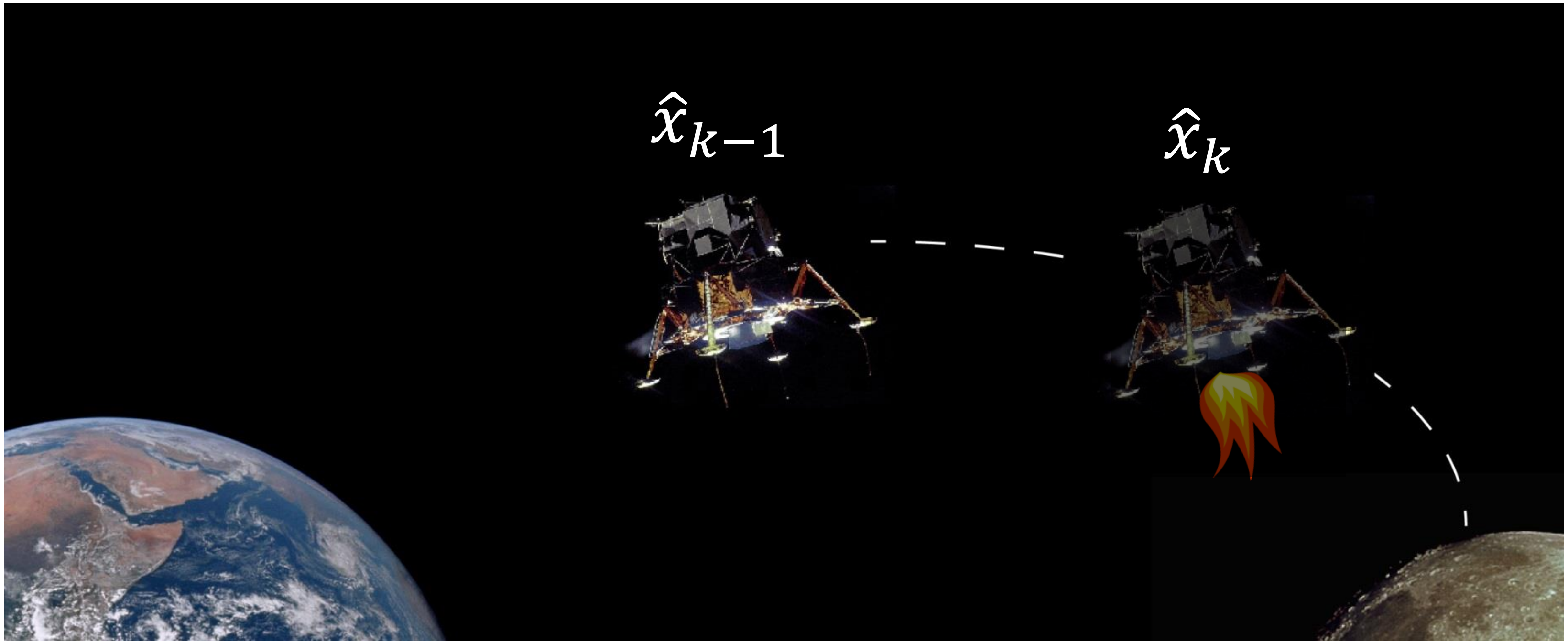
$$N(0, Q_k)$$

Process noise



So we'll just add this term to our existing covariance:

$$P_k = F_k P_{k-1} F_k^T + Q_k$$



Predict Phase:

Predicted (*a priori*) state estimate: $\hat{x}_k = F_k \hat{x}_{k-1} + B_k u_k$

Predicted (*a priori)* estimate covariance: $P_k = F_k P_{k-1} F_k^T + Q_k$

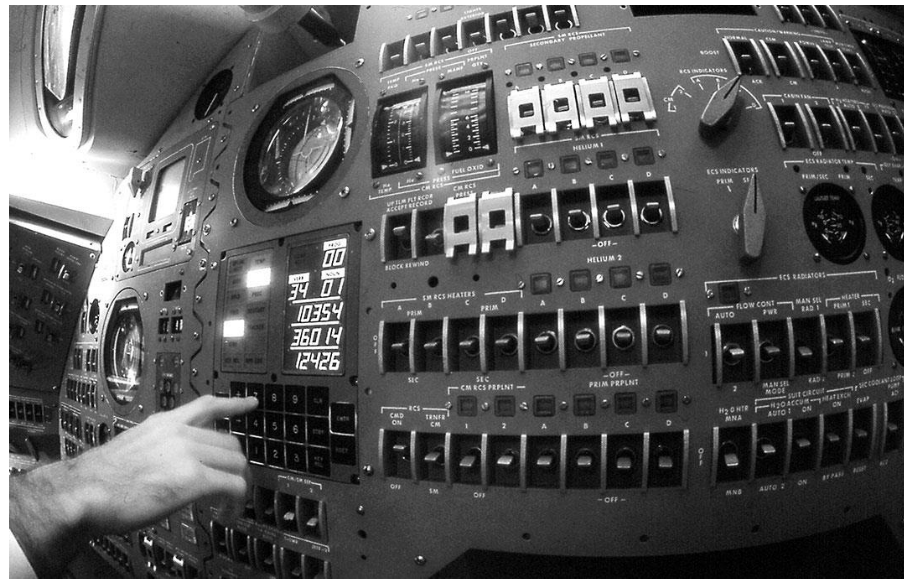
Now to update our estimate

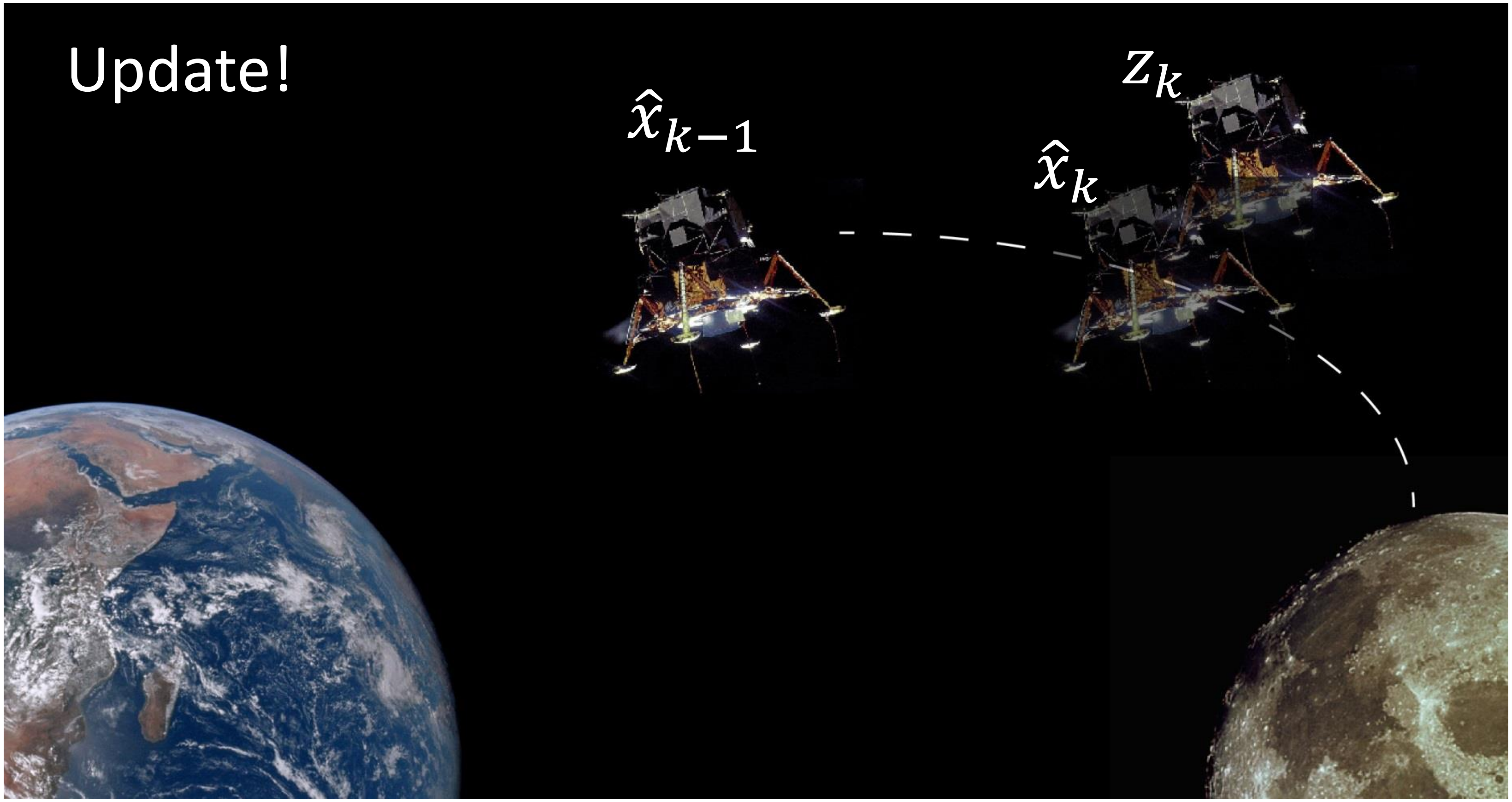
- We'll make an observation z_k of our state x_k

$$z_k = H_k x_k + v_k$$

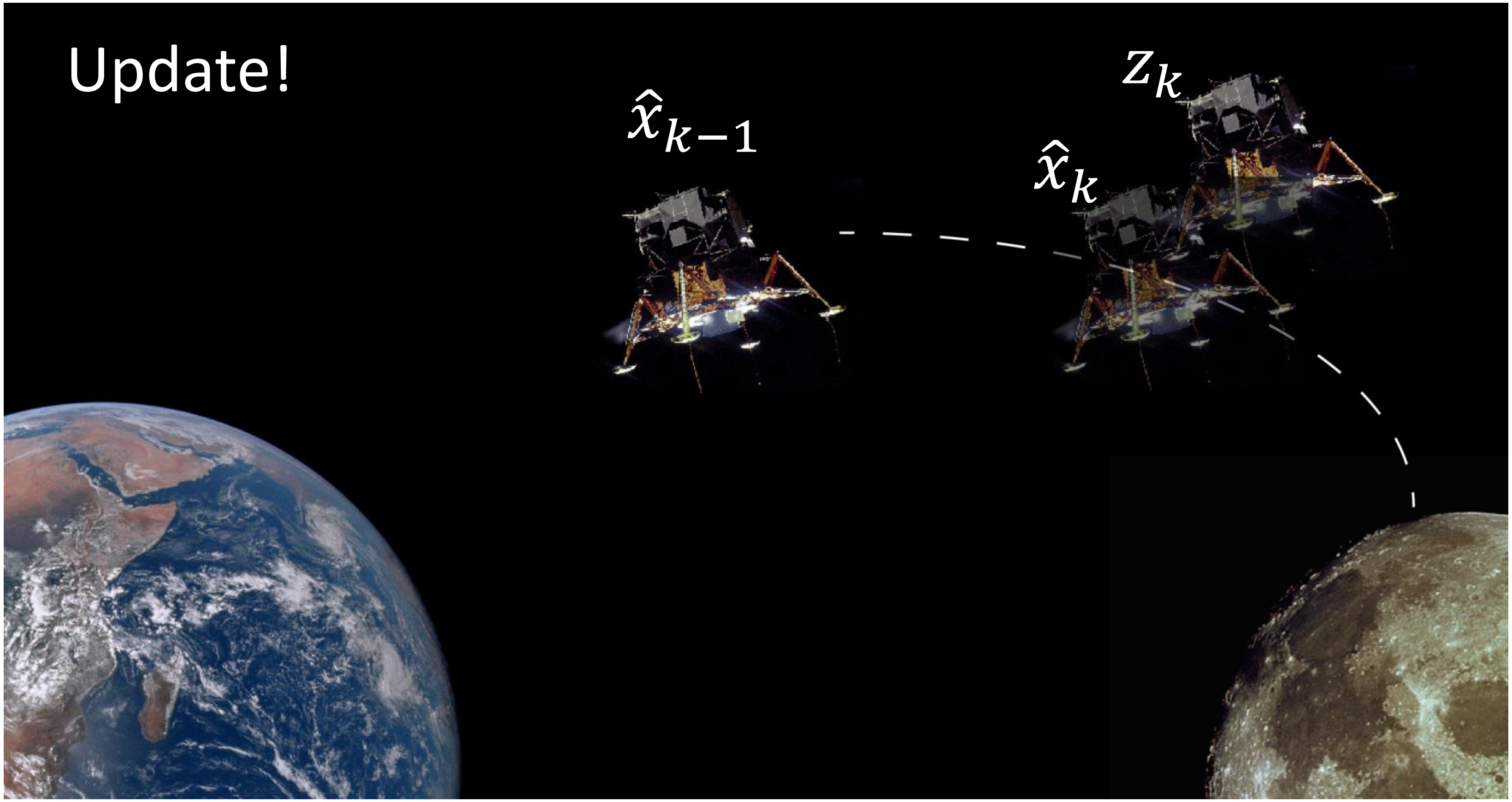
Where H_k is a model of how to transform our observation into the state (e.g. sensor has different units/scale than state)

Where v_k is the observation noise, drawn from a Gaussian $N(0, R_k)$





How can we reconcile our prediction of the state with our sensor readings?



We'll apply a term to the measurement residual: $\boldsymbol{K_k}$, our Kalman gain

Updated (*a posteriori) state estimate:*

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

Kalman Gain

- Minimum mean-square estimator, $E[|x_k - \hat{x}_k|^2]$
- When the gain is zero we keep our prediction:

$$\hat{x}_k = \hat{x}_k + 0(z_k - \hat{x}_k)$$

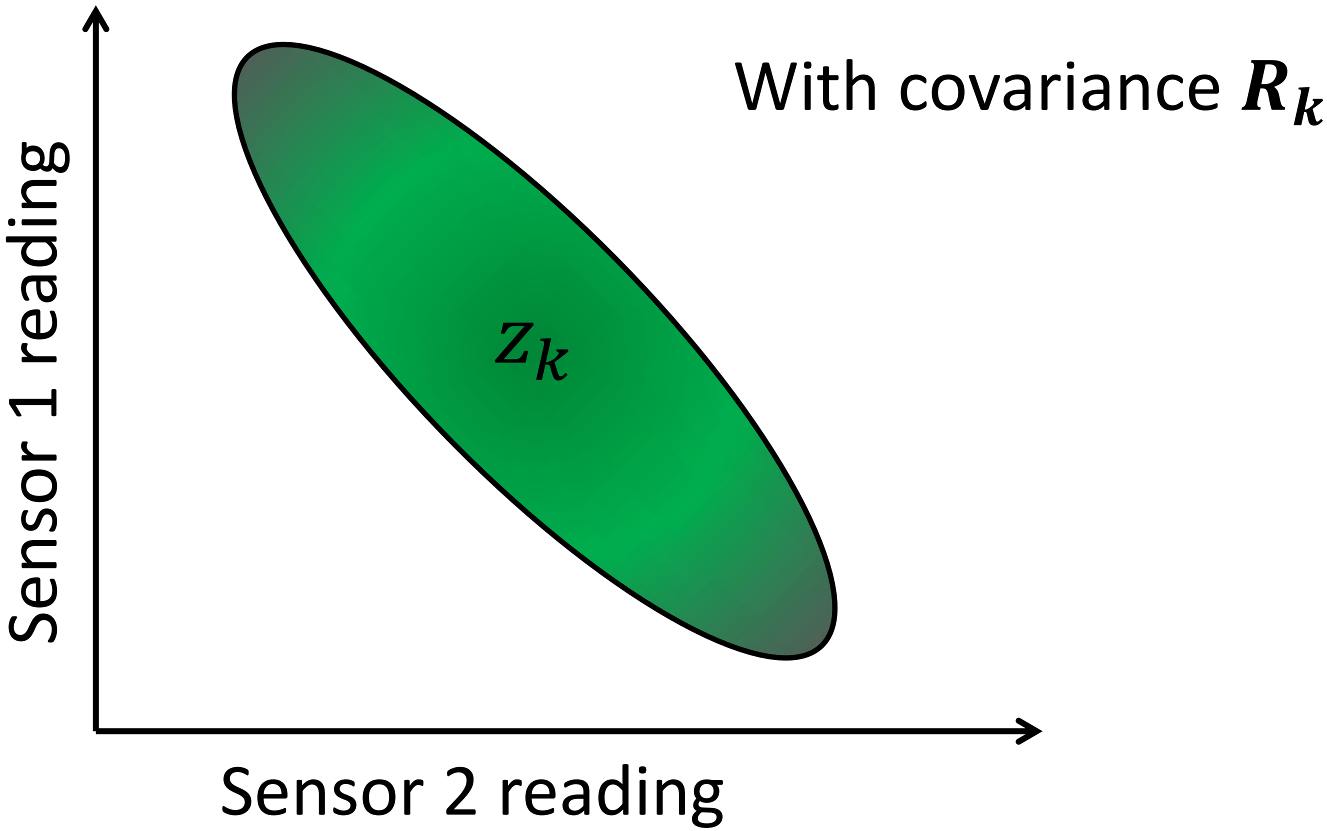
$$\hat{x}_k = \hat{x}_k$$

- When the gain is one we ignore the prediction:

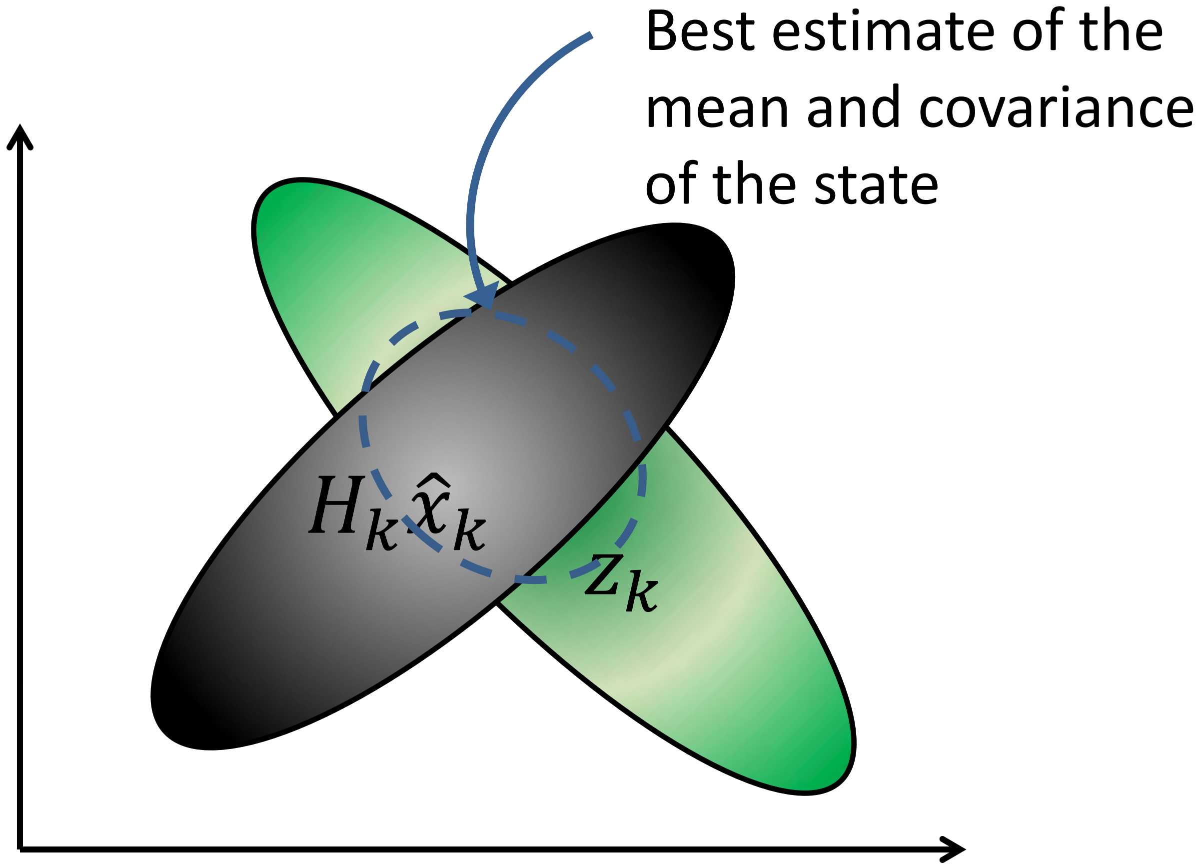
$$\hat{x}_k = \hat{x}_k + 1(z_k - \hat{x}_k)$$

$$\hat{x}_k = z_k$$

Optimal Kalman gain



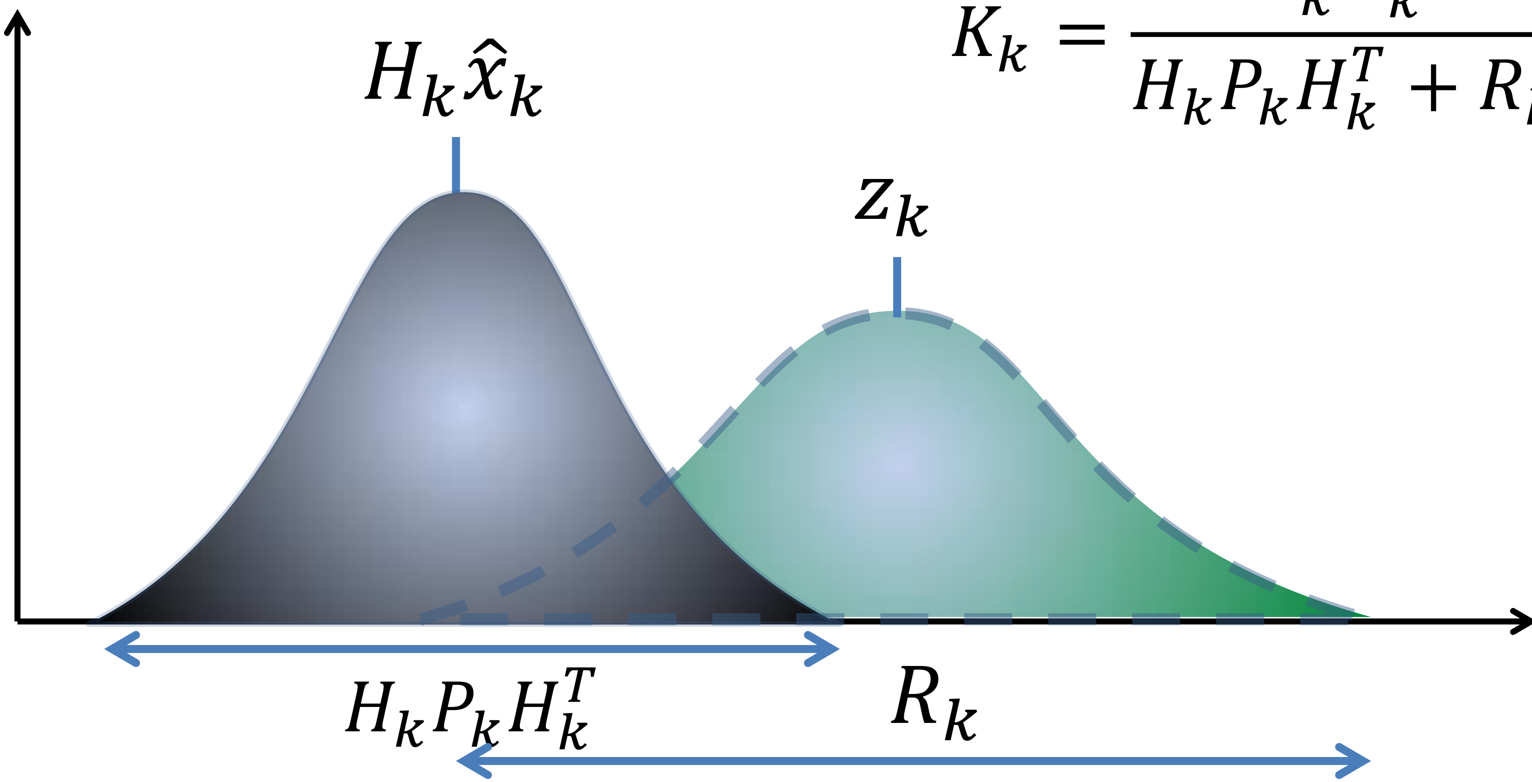
Optimal Kalman gain

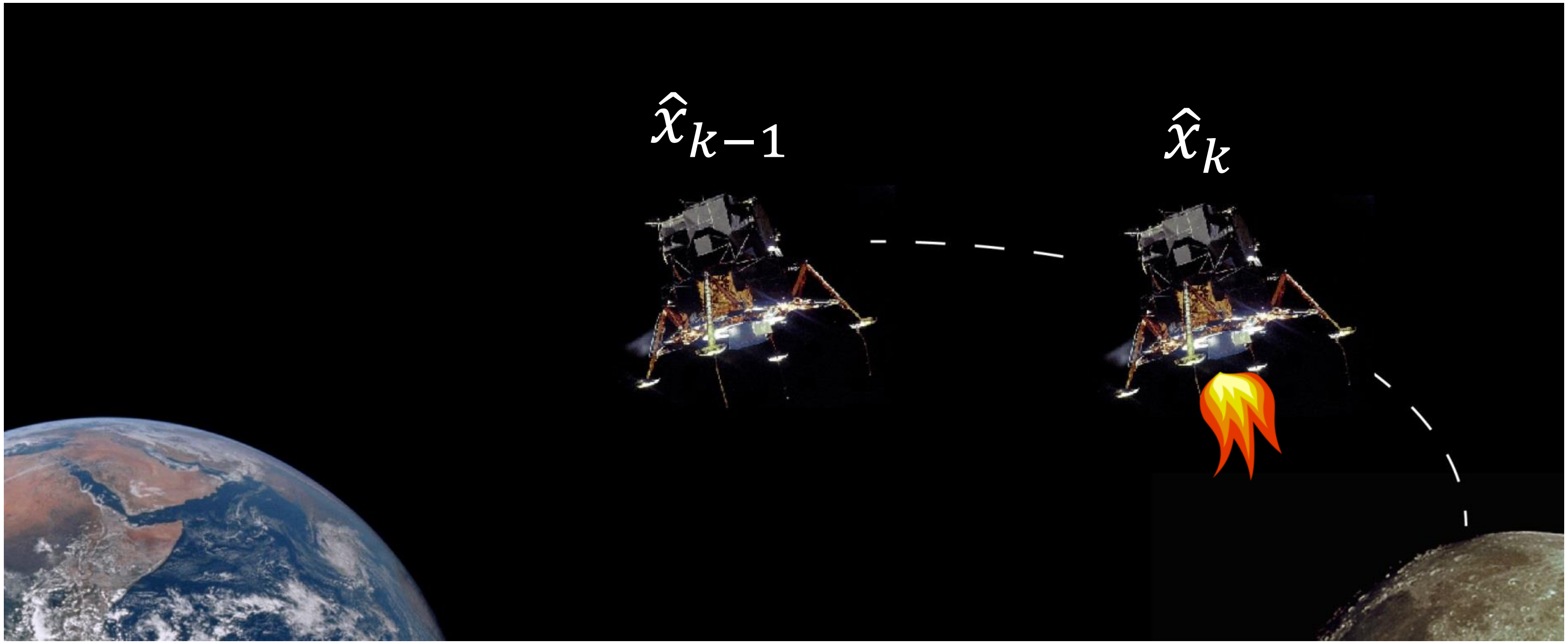


Optimal Kalman Gain

- Probably easier to conceptualize in 1-dimension

$$K_k = \frac{P_k H_k^T}{H_k P_k H_k^T + R_k}$$





Update Phase:

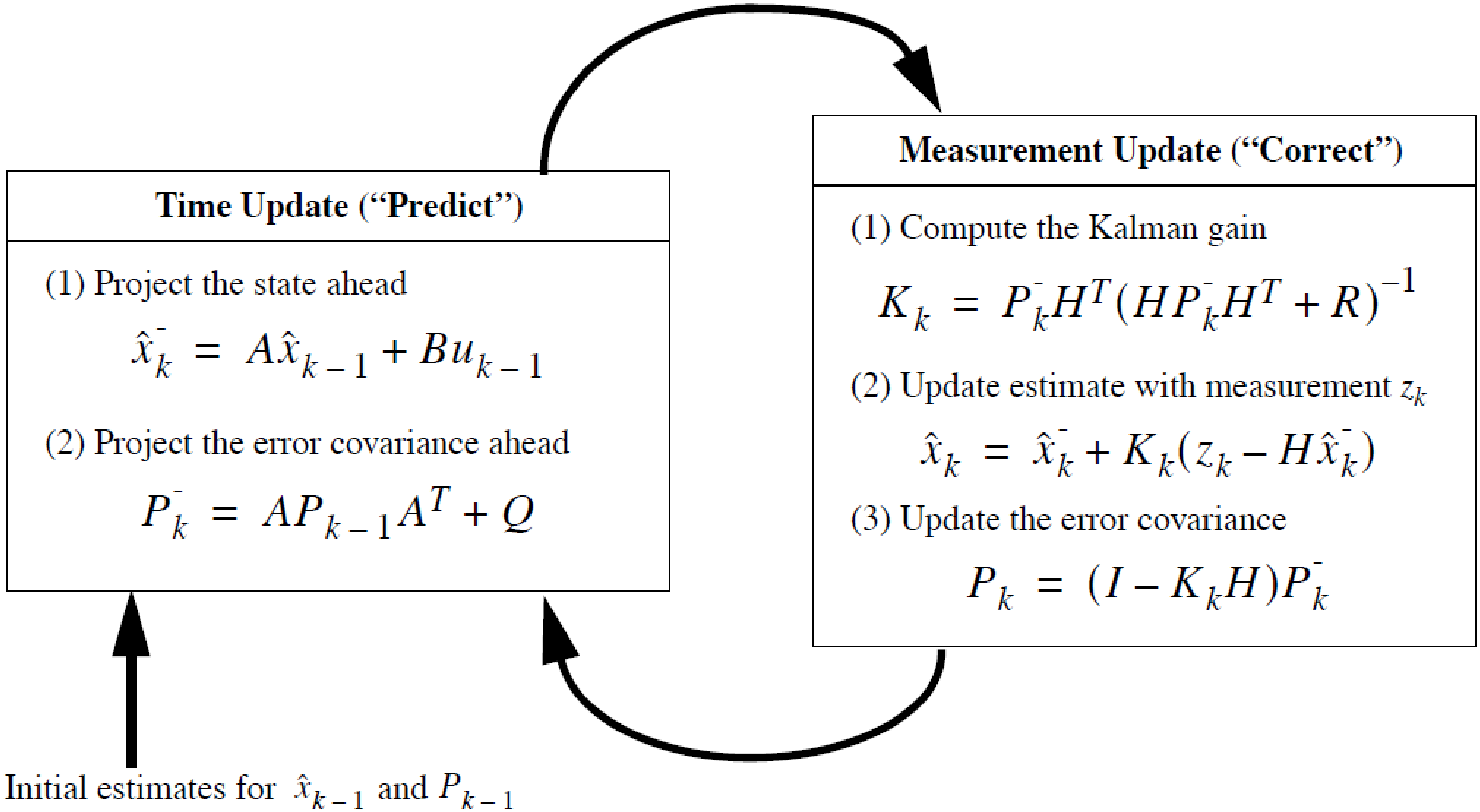
Covariance residual

Measurement residual

Optimal Kalman Gain: $K_k = P_k H_k^T \overbrace{(H_k P_k H_k^T + R_k)^{-1}}$

Updated (*a posteriori*) state estimate: $\hat{x}_k = \hat{x}_k + K_k \overbrace{(z_k - H_k \hat{x}_k)}$

Updated (*a posteriori*) estimate covariance: $P_k = (I - K_k H_k) P_k$

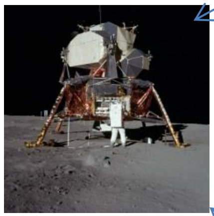


Welch & Bishop (2006)

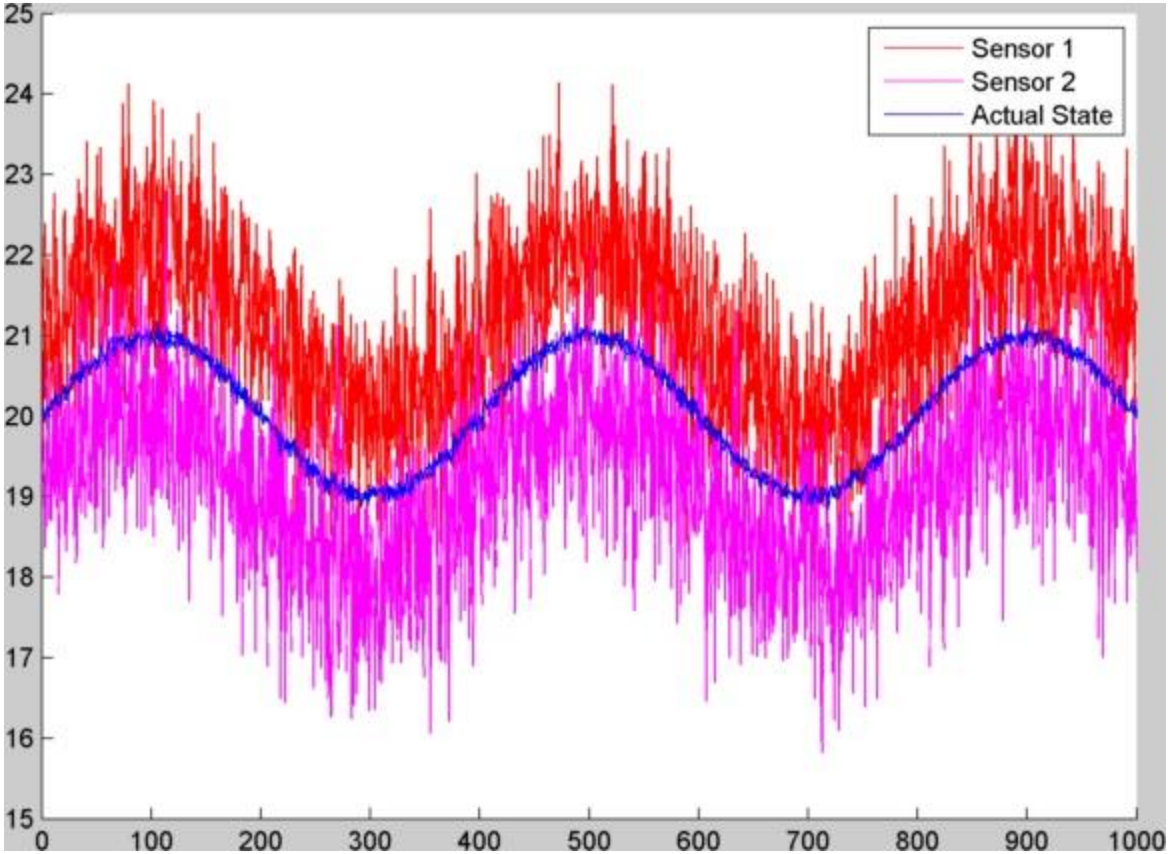
Sensor Fusion Example

- How can we integrate the information from multiple sensors?

$$\begin{bmatrix} gyroscope_k \\ gps_k \\ accelerometer_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} position_{k-1} \\ velocity_{k-1} \end{bmatrix}$$

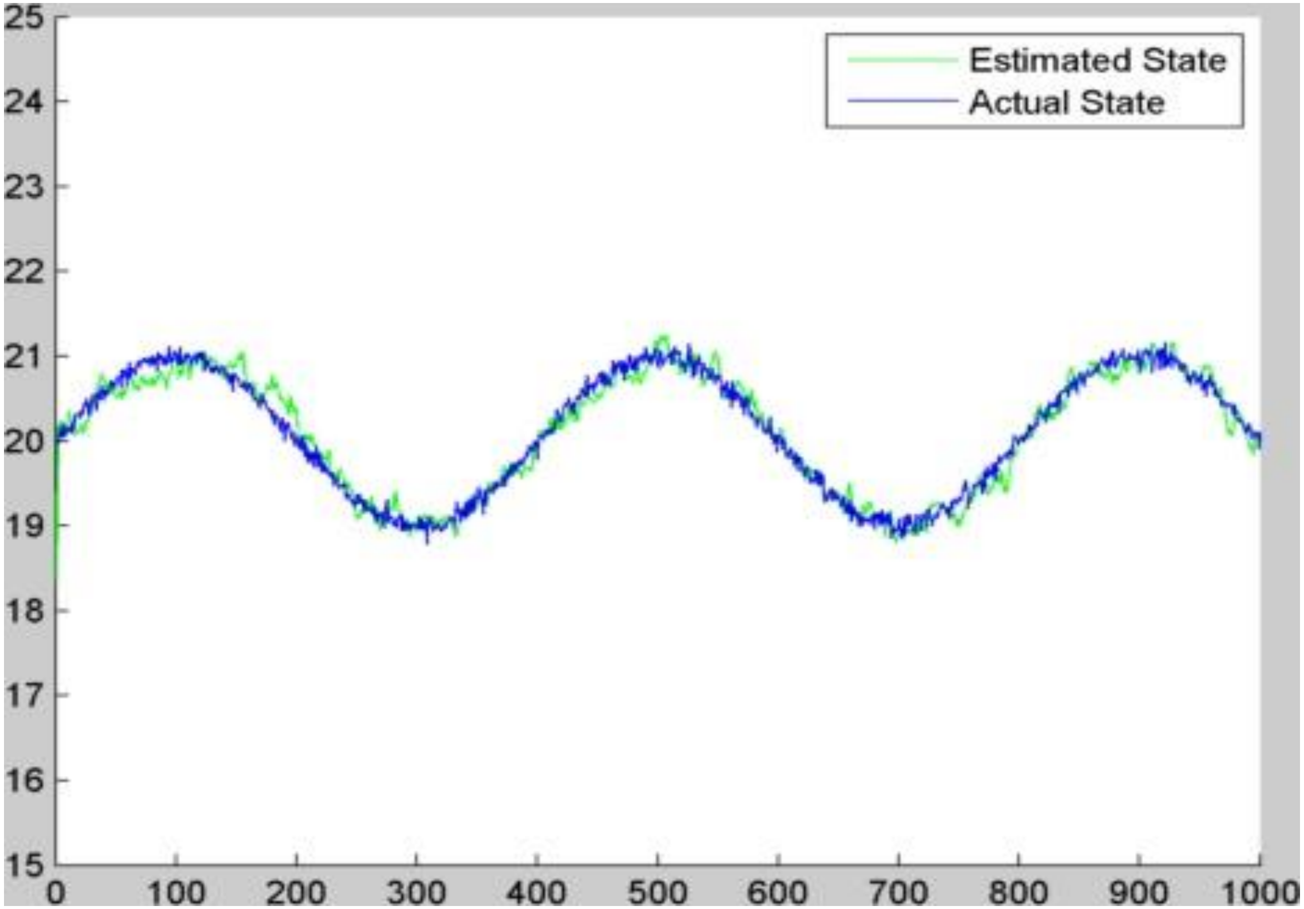


Matlab demo



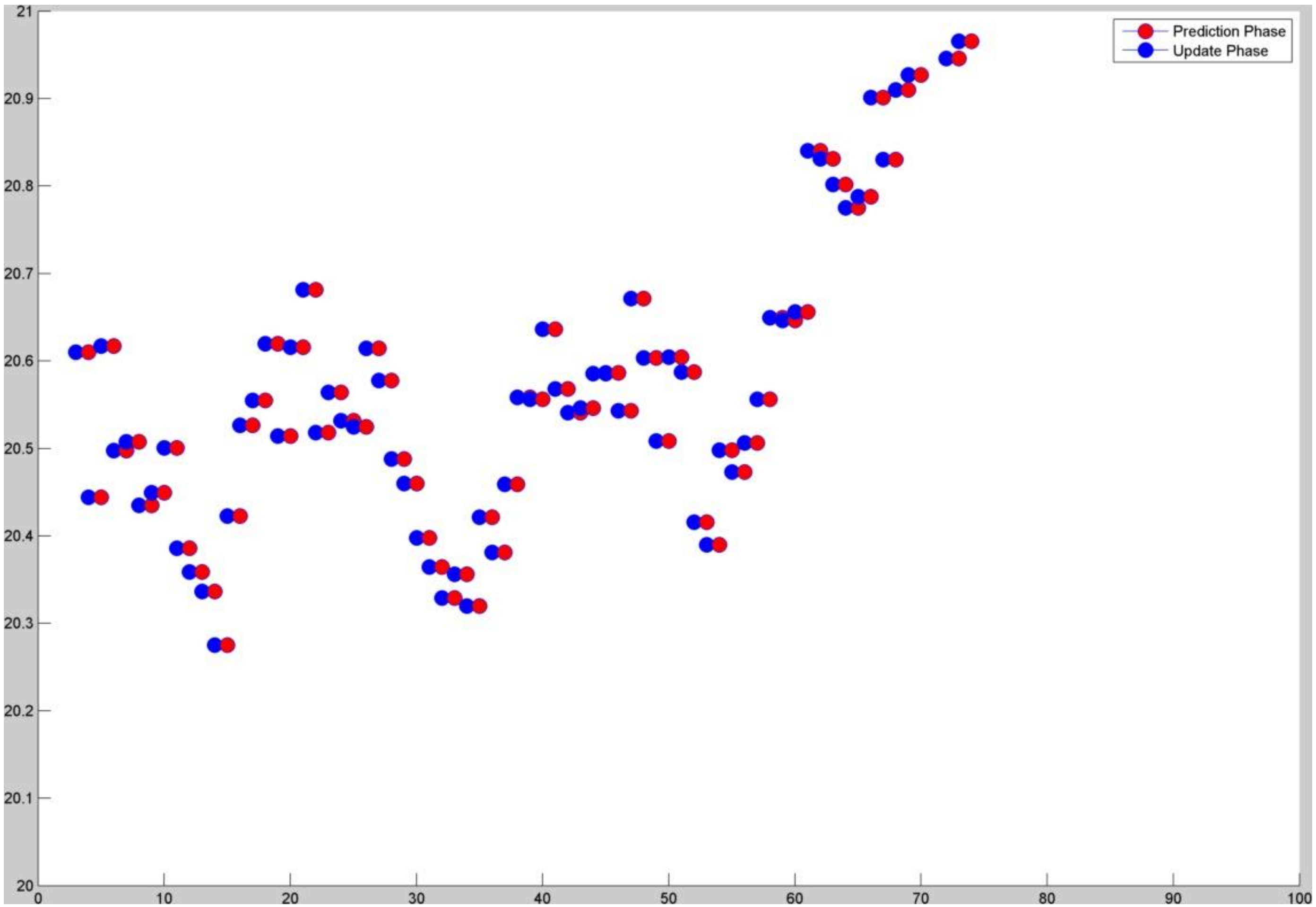
Sensor 1
Sensor 2
Actual State
25
24
23
22
21
20
19
18
17
16
15
0
100
200
300
400
500
600
700
800
900
1000

Matlab demo

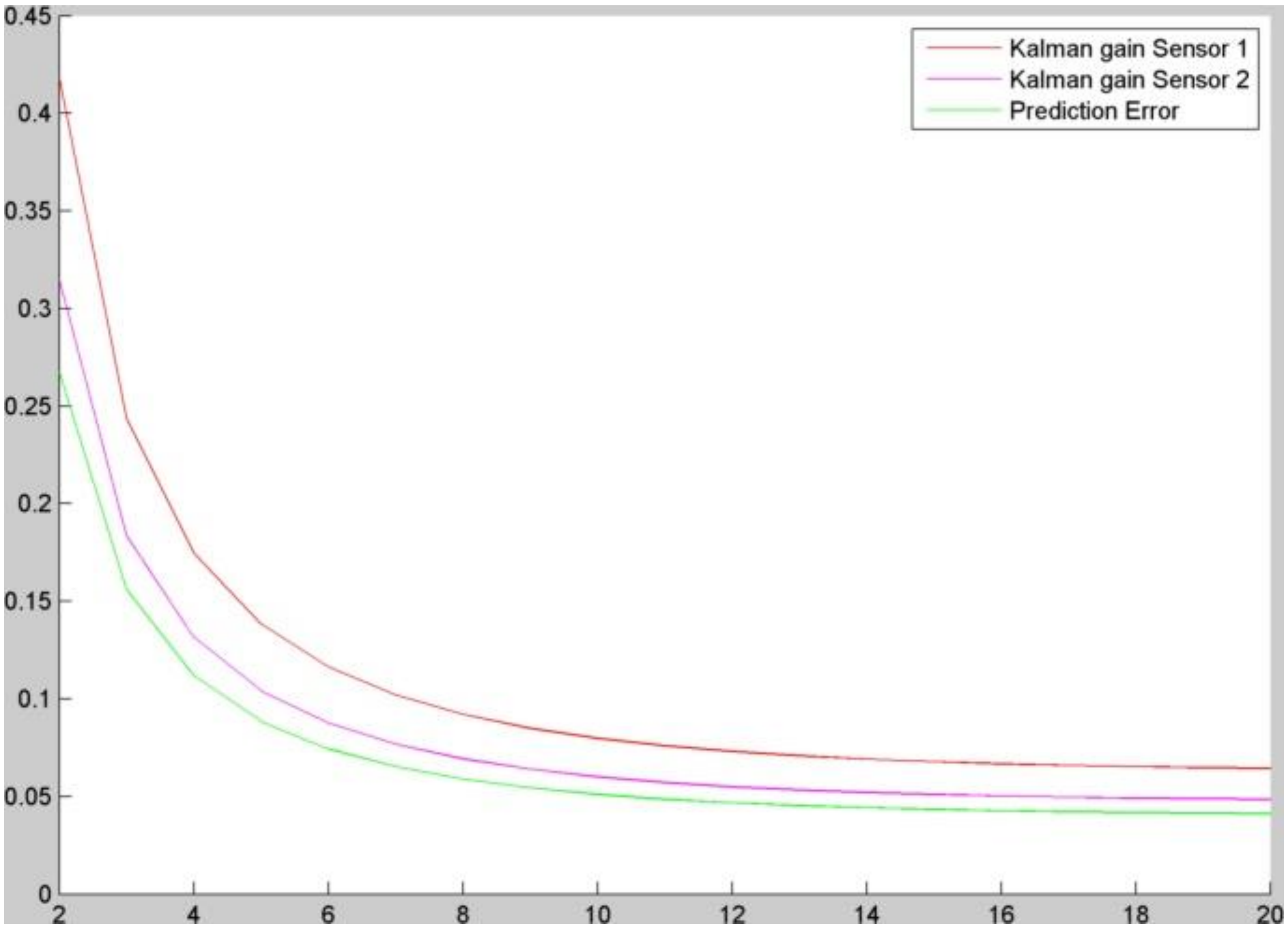


Estimated State
Actual State
25
24
23
22
21
20
19
18
17
16
15
0
100
200
300
400
500
600
700
800
900
1000

Matlab demo

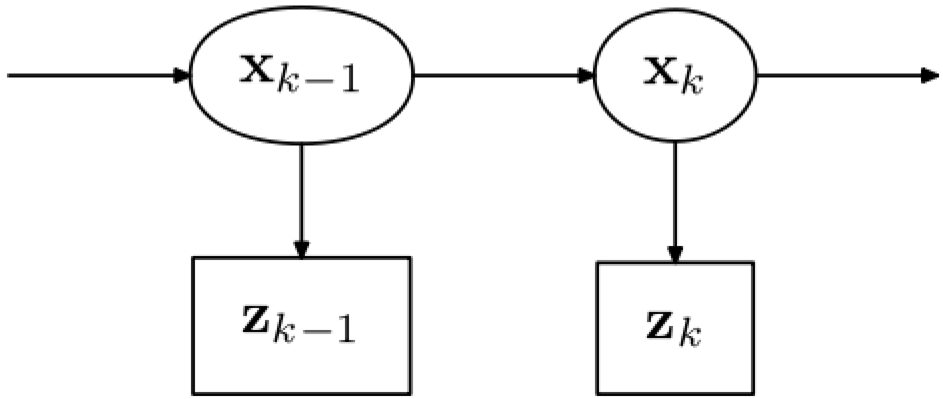

Prediction Phase
Update Phase
21
20.9
20.8
20.7
20.6
20.5
20.4
20.3
20.2
20.1
20
0
10
20
30
40
50
60
70
80
90
100

Matlab demo



Kalman gain Sensor 1
Kalman gain Sensor 2
Prediction Error
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2
4
6
8
10
12
14
16
18
20

Recursive bayesian estimation



- Assume that our true state is a Markov process

$$p(x_k|x_0 \dots x_{k-1}) = p(x_k|x_{k-1})$$
$$p(z_k|x_0 \dots x_k) = p(z_k|x_k)$$
$$p(x_k|Z_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z_{k-1})\,dx_{k-1}$$

Extended Kalman Filter

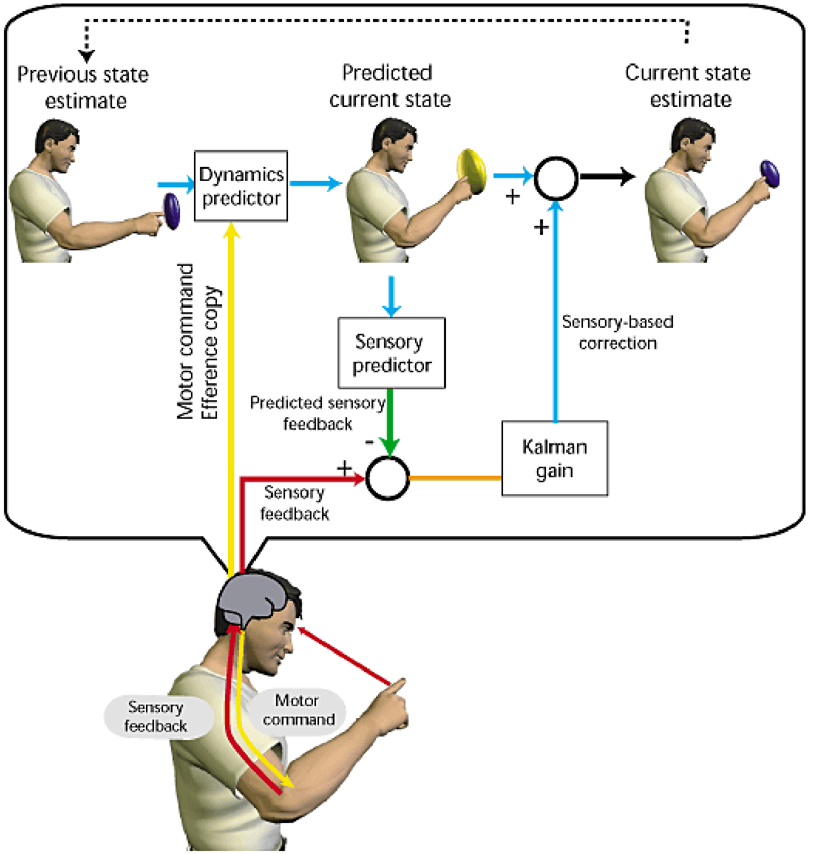
- Allows us to model nonlinear processes (e.g. exponential, quadratic, periodic)

$$x_k = f(x_{k-1}, u_k) + w_k$$
$$z_k = h(x_k) + v_k$$

- Our H_k will be a matrix with the first derivative of each sensor value with respect to each state value, known as the Jacobian

$$z_k = H_k x_k \qquad z_k = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \end{bmatrix}$$

Estimating current state of motor system



Thank You!