Introduction to Kalman Filtering

Paul Newman

SLAM Summer School 2006

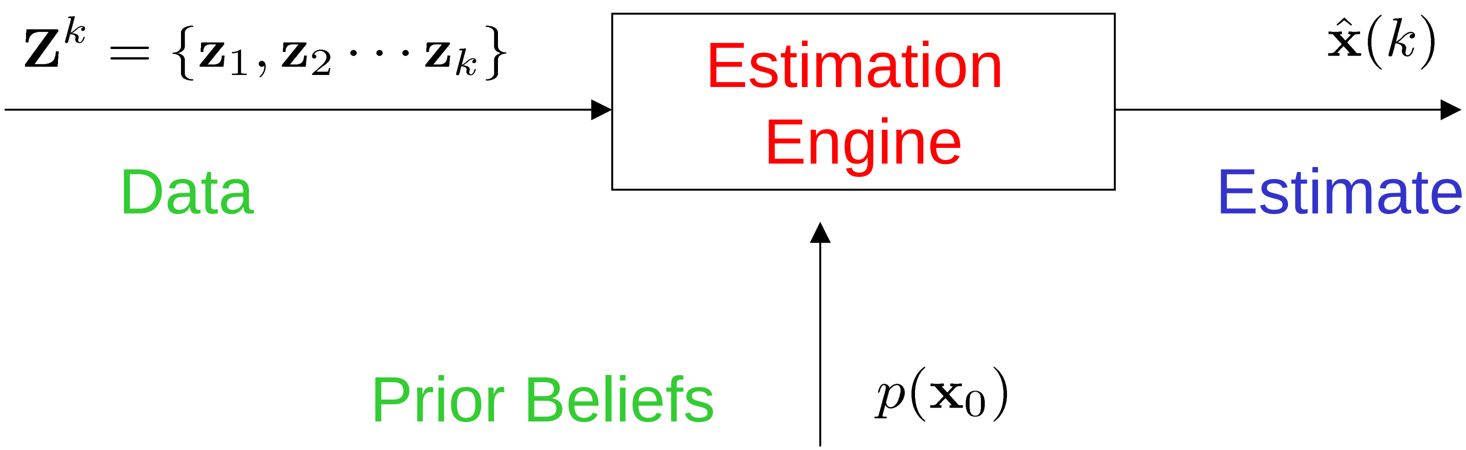
SSS 06

Why This Lecture?

- The Kalman filter is an ubiquitous estimation tool (very common in robotics)
- SLAM was first formulated using a K.F
- It lends itself to the analysis properties of the SLAM problem
- Its wholesome stuff.

Estimation is .....

"Estimation is the process by which we infer the value of a quantity of interest, $\mathbf{x}$, by processing data that is in some way dependent on $\mathbf{x}$."



Maximum Likelihood

$$\mathcal{L} \triangleq p(\mathbf{z}|\mathbf{x}) \qquad p(\mathbf{z}|\mathbf{x}) = \frac{1}{C} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T \mathbf{P}^{-1}(\mathbf{z}-\mathbf{x})}$$

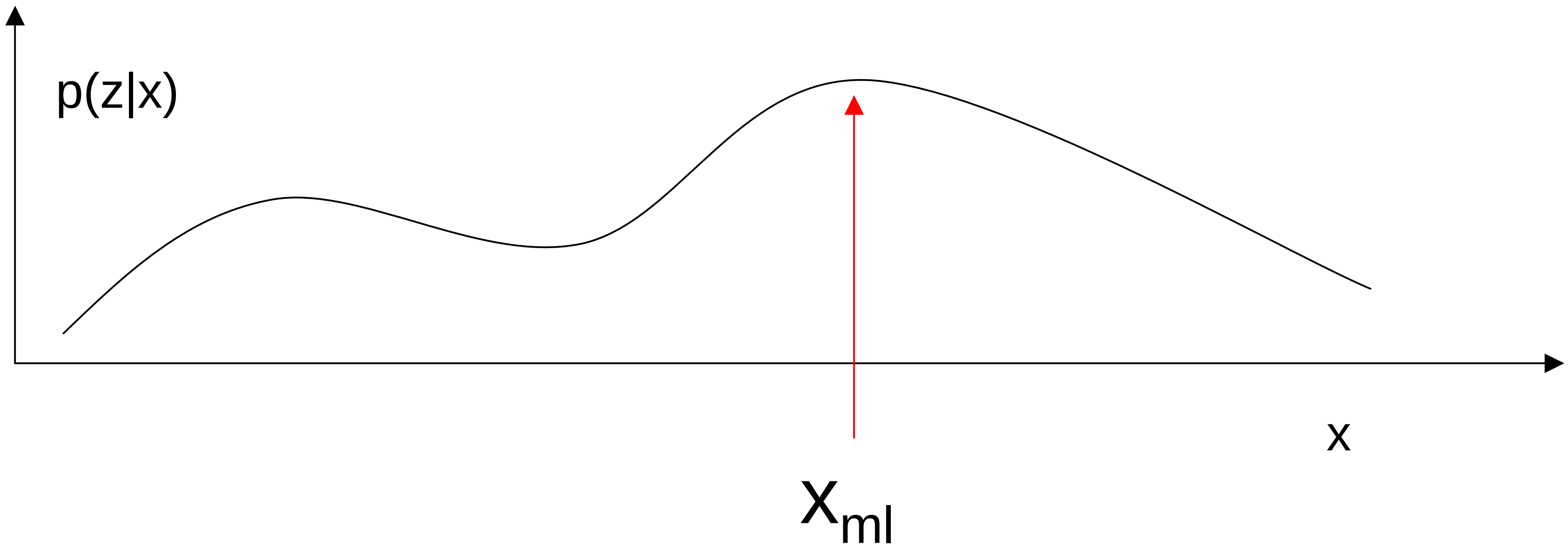
N.B Multivariate Gaussian Understood?

Given an observation $\mathbf{z}$ and a likelihood function $p(\mathbf{z}|\mathbf{x})$, the **maximum likelihood estimator - ML** finds the value of $\mathbf{x}$ which maximises the likelihood function $\mathcal{L} \triangleq p(\mathbf{z}|\mathbf{x})$.

$$\hat{\mathbf{x}}_{m.l} = \arg\max_{\mathbf{x}} p(\mathbf{z}|\mathbf{x}) \tag{1}$$

<u>Find a value of x(state) that best explains z (data)</u>

ML-II



ML does not incorporate prior knowledge

Maximum A Posteriori Estimation

$$\underbrace{p(\mathbf{x}|\mathbf{z})}_{\text{posterior}} = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}$$

$$\propto \underbrace{p(\mathbf{z}|\mathbf{x})}_{\text{Likelihood}} \times \underbrace{p(\mathbf{x})}_{\text{prior}}$$

Given an observation $\mathbf{z}$,a likelihood function $p(\mathbf{z}|\mathbf{x})$ and a prior distribution on $\mathbf{x}$, $p(\mathbf{x})$, the **maximum a posteriori estimator - MAP** finds the value of $\mathbf{x}$ which maximises the posterior distribution $p(\mathbf{x}|\mathbf{z})$

$$\hat{\mathbf{x}}_{map} = \arg\max_{\mathbf{x}} p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \tag{1}$$

MAP does incorporate prior knowledge

Example: Normal Prior and Likelihoods

$$p(\mathbf{x}) = C_1 \exp\{-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2}\}$$

$$p(\mathbf{z}|\mathbf{x}) = C_2 \exp\{-\frac{(\mathbf{z}-\mathbf{x})^2}{2\sigma_z^2}\}$$

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) &= \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \\ &= C(\mathbf{z}) \times p(\mathbf{z}|\mathbf{x}) \times p(\mathbf{x}) \\ &= C(\mathbf{z}) \underbrace{\exp\{-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} - \frac{(\mathbf{z}-\mathbf{x})^2}{2\sigma_z^2}\}}_{\text{Maximise this}} \end{aligned}$$

Example Cont…

$$\underbrace{\exp\{-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} - \frac{(\mathbf{z}-\mathbf{x})^2}{2\sigma_z^2}\}}_{\text{Maximised when:}}$$

Mean

$$\frac{(x-\alpha)^2}{\beta^2} = -\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} - \frac{(\mathbf{z}-\mathbf{x})^2}{2\sigma_z^2} = 0$$

Variance

$$\alpha = \frac{\sigma_z^2\mu_p + \sigma_p^2\mathbf{z}}{\sigma_z^2+\sigma_p^2} \quad \beta^2 = \frac{\sigma_z^2\sigma_p^2}{\sigma_z^2+\sigma_p^2}$$

decreases

How does the mean change?

$$\hat{\mathbf{x}}_{map} = \mu_p + \frac{\sigma_p^2}{\sigma_p^2 + \sigma_z^2} \times (\mathbf{z} - \mu_p)$$

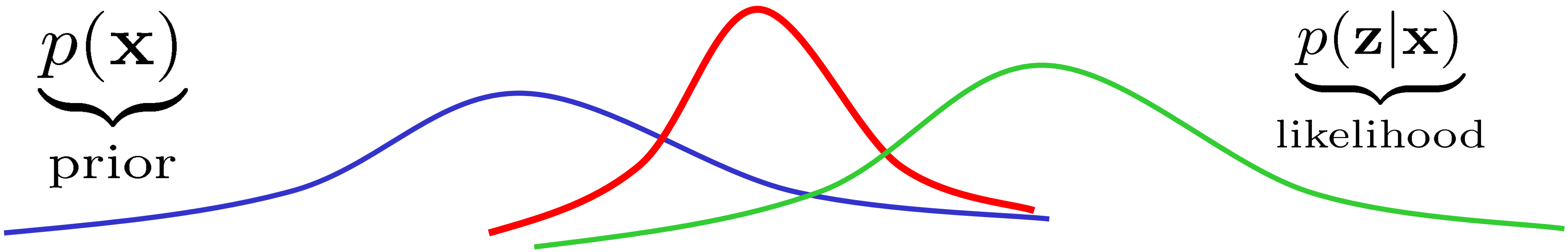
Old (prior) mean

Difference between measurement and prior

Visually...

$$\overbrace{p(\mathbf{x}|\mathbf{z})}^{\text{posterior (estimate)}}$$

$$\underbrace{p(\mathbf{x})}_{\text{prior}} \qquad \underbrace{p(\mathbf{z}|\mathbf{x})}_{\text{likelihood}}$$



$$\hat{\mathbf{x}}_{map} = \mu_p + \frac{\sigma_p^2}{\sigma_p^2 + \sigma_z^2} \times (\mathbf{z} - \mu_p) \qquad \sigma^2_{map} = \frac{\sigma_z^2 \sigma_p^2}{\sigma_z^2 + \sigma_p^2}$$

Minimum Mean Squared Error Estimation

$$\hat{\mathbf{x}}_{mmse} = \arg\min_{\hat{\mathbf{x}}} \mathcal{E}\{\underbrace{(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{Z}^{\mathbf{k}}}_{\text{Cost Function}}\}$$

Choose x so argument is minimised

Expectation operator ("average")

$$\hat{\mathbf{x}} \text{ is estimate } \mathbf{x} \text{ is truth}$$

Evaluating….

$$\mathcal{E}\{g(x)|y\} = \int_{-\infty}^{\infty} g(\mathbf{x})p(\mathbf{x}|\mathbf{y})dx$$

From probability theory

$$J(\hat{\mathbf{x}}, \mathbf{x}) = \mathcal{E}\{(\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{Z^k}\} = \int_{-\infty}^{\infty} (\hat{\mathbf{x}} - \mathbf{x})^T(\hat{\mathbf{x}} - \mathbf{x})p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x}$$

$$\frac{\partial J(\hat{\mathbf{x}}, \mathbf{x})}{\partial \hat{\mathbf{x}}} = 2\int_{-\infty}^{\infty} (\hat{\mathbf{x}} - \mathbf{x})p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x} = 0$$

Splitting apart the integral, noting that $\hat{\mathbf{x}}$ is a constant:

$$\int_{-\infty}^{\infty} \hat{\mathbf{x}}p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x} = \int_{-\infty}^{\infty} \mathbf{x}p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x}$$

$$\hat{\mathbf{x}}\int_{-\infty}^{\infty} p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x} = \int_{-\infty}^{\infty} \mathbf{x}p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x}$$

$$\hat{\mathbf{x}} = \int_{-\infty}^{\infty} \mathbf{x}p(\mathbf{x}|\mathbf{Z^k})d\mathbf{x}$$

Very Important Thing → $\hat{\mathbf{x}}_{mmse} = \mathcal{E}\{\mathbf{x}|\mathbf{Z^k}\}$

Recursive Bayesian Estimation

Key idea: “one mans posterior is another’s prior” ;-)

$\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2 \cdots \mathbf{z}_k\}$ Sequence of data (measurements)

We want conditional mean (mmse) of x given Z^k

Can we iteratively calculate this – ie every time
a new measurement comes in, update our estimate?

$$p(\mathbf{x}|\mathbf{Z^k}) = f\left(p(\mathbf{x}|\mathbf{Z}^{k-1}), p(\mathbf{z}_k|\mathbf{x})\right)$$

Yes...

$$p(\mathbf{x}|\mathbf{Z^k}) = \frac{p(\mathbf{z}_k|\mathbf{x})p(\mathbf{x}|\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})}$$

$$\underbrace{p(\mathbf{x}|\mathbf{Z^k})}_{\text{Estimate}} \propto \underbrace{p(\mathbf{z}_k|\mathbf{x})}_{\text{Likelihood}} \underbrace{p(\mathbf{x}|\mathbf{Z}^{k-1})}_{\text{Last Estimate}}$$

At time k

Explains data at time k
as function of x at time k

At time k-1

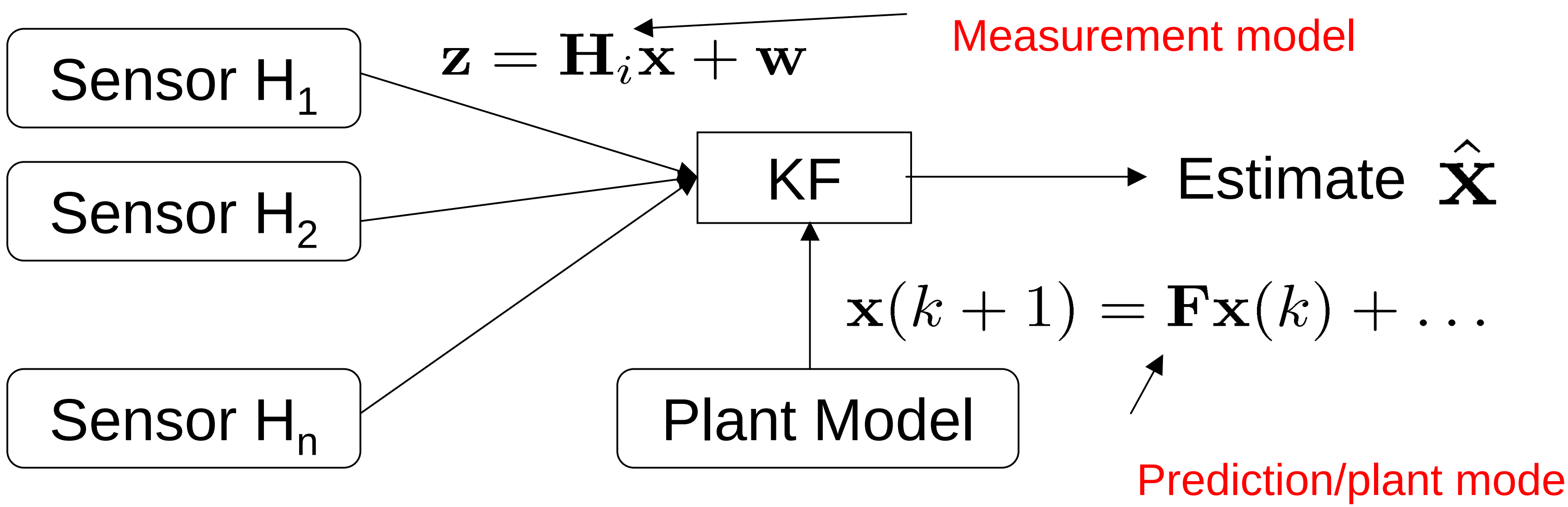
Kalman Filtering

- Ubiquitous estimation tool
- Simple to implement
- Closely related to Bayes estimation and MMSE
- Immensely Popular in robotics
  - Real time
  - Recursive (can add data sequentially)

It is not that complicated!

Overall Goal

To come up with a recursive algorithm that produces an estimate of state by processing data from a set of explainable measurements and incorporating some kind of plant model



True underlying state x

Part 1 – Data Fusion

True state (never known)

$$p(\mathbf{w}) = \frac{1}{(2\pi)^{n/2} \mid \mathbf{R} \mid^{1/2}} \exp\{-\frac{1}{2}\mathbf{w}^T\mathbf{R}^{-1}\mathbf{w}\}$$

$$\mathcal{E}\{\mathbf{w}(i)\mathbf{w}(j)^T\} = \begin{cases} \mathbf{R} & i = j \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{w}(k)$$

Noise vector – zero mean

Model – explains data
in terms of state. A matrix

Data/measurement/observation from sensor
described by H. Generally a vector

k is time index

Likelihood

If Gaussian noise process **w** has zero mean and Covariance R

$$\mathcal{E}\{\mathbf{z}(k)\} = \mathcal{E}\{\mathbf{H}\mathbf{x}(k) + \mathbf{w}(k)\} = \mathbf{H}\mathbf{x}$$

Likelihood can be written as

$$p(\mathbf{z}|\mathbf{x}) = \frac{1}{(2\pi)^{nz/2} \mid \mathbf{R} \mid^{1/2}} \exp\{-\frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x})\}$$

.

Assume we have a prior belief….

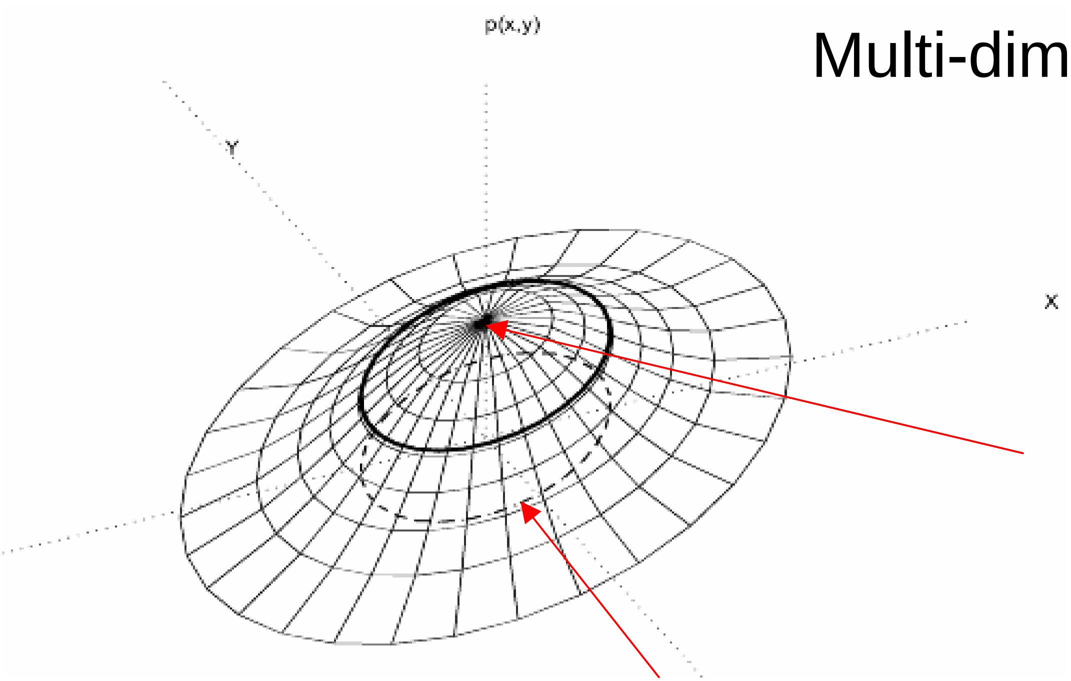
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{nx/2} \mid \mathbf{P}_{\ominus} \mid^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_{\ominus})^T \mathbf{P}_{\ominus}^{-1} (\mathbf{x} - \mathbf{x}_{\ominus})\}$$

Prior mean

Prior Covariance

Covariance is…..

Multi-dimensional analogy of variance



mean

P is a symmetric matrix that describes a 1-standard deviation contour (ellipsoid in 3D+) of the pdf

Getting The Posterior

Now we can use Bayes rule to figure out an expression for the posterior $p(\mathbf{x}|\mathbf{z})$.:

$$\begin{aligned}
p(\mathbf{x}|\mathbf{z}) &= \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \\
&= \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{\int_{-\infty}^{\infty} p(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}} \\
&= \frac{\frac{1}{(2\pi)^{nz/2}|\mathbf{R}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{z}-\mathbf{H}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{z}-\mathbf{H}\mathbf{x})\} \frac{1}{(2\pi)^{nx/2}|\mathbf{P}_\ominus|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_\ominus)^T\mathbf{P}_\ominus^{-1}(\mathbf{x}-\mathbf{x}_\ominus)\}}{\mathbf{C}(\mathbf{z})} \\
&\propto \exp\{-\frac{1}{2}(\mathbf{z}-\mathbf{H}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{z}-\mathbf{H}\mathbf{x})\} \exp\{-\frac{1}{2}(\mathbf{x}-\mathbf{x}_\ominus)^T\mathbf{P}_\ominus^{-1}(\mathbf{x}-\mathbf{x}_\ominus)\} \\
&= \exp\{-1/2 \underbrace{\left((\mathbf{z}-\mathbf{H}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{z}-\mathbf{H}\mathbf{x}) + (\mathbf{x}-\mathbf{x}_\ominus)\mathbf{P}_\ominus^{-1}(\mathbf{x}-\mathbf{x}_\ominus)^T\right)}_{\text{express as } (\mathbf{x}-\mathbf{x}_\oplus)^T\mathbf{P}_\oplus^{-1}(\mathbf{x}-\mathbf{x}_\oplus)}\}
\end{aligned}$$

Which leads to…

Comparing terms allows the following to be derived:

Updated covariance $\mathbf{P}_{\oplus} = (\mathbf{P}_{\ominus}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$

Updated mean $\mathbf{x}_{\oplus} = (\mathbf{P}_{\ominus}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{P}_{\ominus}^{-1}\mathbf{x}_{\ominus} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}).$

$$p(\mathbf{x}|\mathbf{z}) = \frac{1}{(2\pi)^{nx/2} \mid \mathbf{P}_{\oplus} \mid^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{x}_{\oplus})^T\mathbf{P}_{\oplus}^{-1}(\mathbf{x} - \mathbf{x}_{\oplus})\}$$

$$\mathcal{E}\{p(\mathbf{x}|\mathbf{z})\} = \mathbf{x}_{\oplus} = \text{MMSE estimate}$$

Quick Review…

1. Derived a closed for solution for MAP estimator with Gaussian sensor noise model and Gaussian Prior

2. Mean of p(x|z) shown to be MMSE estimator

3. Kalman filter simply takes in observations (z) and old estimates $p(x|z)_{old}$ and produces and new estimated distribution $p(x|z)_{new}$

4. If pdfs are Gaussian we only need talk about mean and Covariance and from (2) the mean is the best estimate

Inverting the Inverse

$$\mathbf{P}_{\oplus} = (\mathbf{P}_{\ominus}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$$
$$\mathbf{x}_{\oplus} = (\mathbf{P}_{\ominus}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}(\mathbf{P}_{\ominus}^{-1}\mathbf{x}_{\ominus} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}).$$

Can be re-written as

$$\begin{aligned}\mathbf{P}_{\oplus} &= \mathbf{P}_{\ominus} - \mathbf{P}_{\ominus}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{P}_{\ominus}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{P}_{\ominus} \\ &= \mathbf{P}_{\ominus} - \mathbf{W}\mathbf{S}\mathbf{W}^T \\ &= (\mathbf{I} - \mathbf{W}\mathbf{H})\mathbf{P}_{\ominus}\end{aligned}$$

where

$$\mathbf{S} = \mathbf{H}\mathbf{P}_{\ominus}\mathbf{H}^T + \mathbf{R}$$
$$\mathbf{W} = \mathbf{P}_{\ominus}\mathbf{H}^T\mathbf{S}^{-1}$$

Covariance decreases when sensor data is fused with prior

And a new state update Eqn:

$$\mathbf{x}_{\oplus} = \mathbf{x}_{\ominus} + \mathbf{W}(\mathbf{z} - \mathbf{H}\mathbf{x}_{\ominus})$$

New estimate
(mean of pdf)

Real observation

Best previous
estimate

Predicted observation

Kalman Update Equations:

Given an observation $\mathbf{z}$ with uncertainty (covariance) $\mathbf{R}$ and a prior estimate $\mathbf{x}_\ominus$ with covariance $\mathbf{P}_\ominus$ the new estimate and covariance are calculated as:

$$\mathbf{x}_\oplus = \mathbf{x}_\ominus + \mathbf{W}\nu$$

$$\mathbf{P}_\oplus = \mathbf{P}_\ominus - \mathbf{W}\mathbf{S}\mathbf{W}^T$$

where the **"Innovation"** ν is

$$\nu = \mathbf{z} - \mathbf{H}\mathbf{x}_\ominus$$

the **"Innovation Covariance"** $\mathbf{S}$ is given by

$$\mathbf{S} = \mathbf{H}\mathbf{P}_\ominus\mathbf{H}^T + \mathbf{R}$$

and the **"Kalman Gain"** $\mathbf{W}$ is given by

$$\mathbf{W} = \mathbf{P}_\ominus\mathbf{H}^T\mathbf{S}^{-1}$$

We can use the "recursive Bayesian" result to allow us to use one iteration's estimate (posterior) as the next iteration's prior

The i|j notation

$\hat{\mathbf{x}}(i|j)$ is the estimate of $\mathbf{x}$ at time i given measurements up until time j. Commonly you will see the following combinations:

- $\hat{\mathbf{x}}(k|k)$ estimate at time k given all available measurements. Often simply called the **estimate**
- $\hat{\mathbf{x}}(k|k-1)$ estimate at time k given first $k-1$ measurements. This is often called the **prediction**

$$\mathbf{P}(i|j) = \mathcal{E}\{(\mathbf{x}(i) - \hat{\mathbf{x}}(i|j))(\mathbf{x}(i) - \hat{\mathbf{x}}(i|j))^T|\mathbf{Z}^j\}$$

true

estimated

Data up to t=j

This is useful for derivations but we can never use it in a calc as x is unknown truth!

Incorporating Plant Models

We have modelled the noisy sensor as a z(k) = Hx(k) +w(k)

We may also have a (noisy) understanding of how the state evolves with time and other control inputs

$$\mathbf{x}(k) = \mathbf{F}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{G}\mathbf{v}(k)$$

new state

Previous state

control

noise

Note this is a truth model not an estimation equation – no i|j

Predicting the mean:

From the "MMSE" result

$$
\begin{aligned}
\hat{\mathbf{x}}(k|k-1) &= \mathcal{E}\{\mathbf{x}(k|\mathbf{Z}^{k-1})\} \\
&= \mathcal{E}\{\mathbf{F}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{G}\mathbf{v}(k)|\mathbf{Z}^{k-1}\} \\
&= \mathbf{F}\mathcal{E}\{\mathbf{x}(k-1)|\mathbf{Z}^{k-1}\} + \mathbf{B}\mathbf{u}(k) + \mathbf{G}\mathcal{E}\{\mathbf{v}(k)|\mathbf{Z}^{k-1}\} \\
&= \mathbf{F}\hat{\mathbf{x}}(k-1|k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{0}
\end{aligned}
$$

Noise term is unknown
but zero mean doesn't
affect prediction

Last best estimate

Control at time k e.g. steering

Predicting the Covariance

$$\mathbf{P}(k|k-1) = \mathcal{E}\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1))(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1))^T|\mathbf{Z}^{k-1}\}$$

Substitution and cancellation (zero mean) gives

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T$$

This result should also seem familiar to you - remember that if $\mathbf{x} \sim N(\mu, \mathbf{P})$ and $\mathbf{y} = \mathbf{F}\mathbf{x}$ then $\mathbf{y} \sim N(\mathbf{F}\mu, \mathbf{F}\mathbf{P}\mathbf{F}^T)$?.

Q is the strength of the process noise, **v,** - its covariance

Prediction + Update:

Rewriting update equations using k|k notation:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k)\nu(k)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}\mathbf{W}(k)^T$$

$$\nu(k) = \mathbf{z}(k) - \mathbf{H}\mathbf{x}(k|k-1)$$

$$\mathbf{S} = \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{R}$$

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T\mathbf{S}^{-1}$$

Combining with Prediction

Linear Kalman Filter Equations

prediction:

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{F}\hat{\mathbf{x}}(k-1|k-1) + \mathbf{B}\mathbf{u}(k) \tag{1}$$

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{2}$$

update:

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k)\nu(k) \tag{3}$$

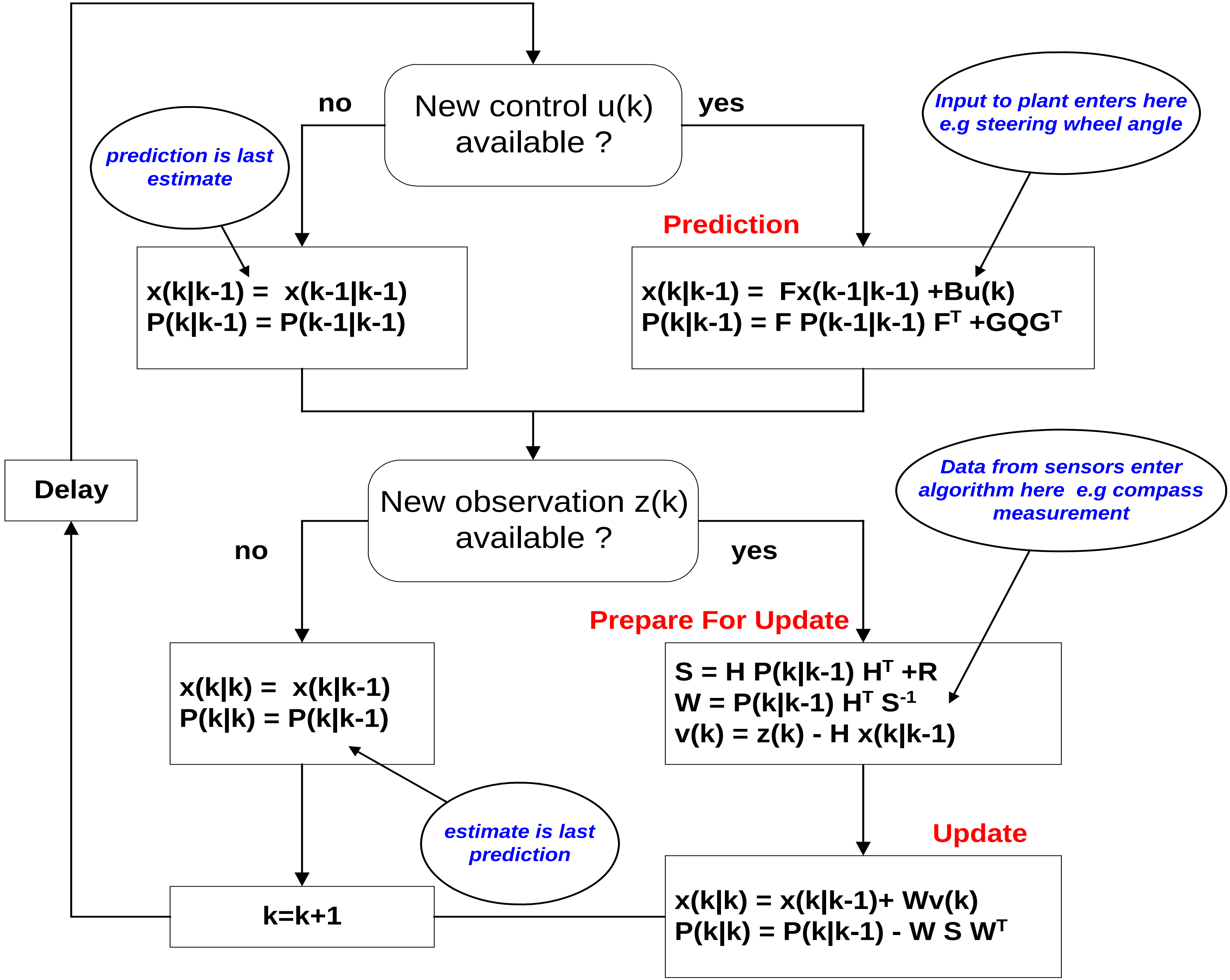
$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k)\mathbf{S}\mathbf{W}(k)^T \tag{4}$$

where

$$\nu(k) = \mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1) \tag{5}$$

$$\mathbf{S} = \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{R} \tag{6}$$

$$\mathbf{W}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T\mathbf{S}^{-1} \tag{7}$$


New control u(k) available ?
no
yes
prediction is last estimate
Input to plant enters here e.g steering wheel angle
Prediction
x(k|k-1) = x(k-1|k-1)
P(k|k-1) = P(k-1|k-1)
x(k|k-1) = Fx(k-1|k-1) +Bu(k)
P(k|k-1) = F P(k-1|k-1) F^T +GQG^T
Delay
New observation z(k) available ?
no
yes
Data from sensors enter algorithm here e.g compass measurement
Prepare For Update
x(k|k) = x(k|k-1)
P(k|k) = P(k|k-1)
S = H P(k|k-1) H^T +R
W = P(k|k-1) H^T S^-1
v(k) = z(k) - H x(k|k-1)
estimate is last prediction
Update
k=k+1
x(k|k) = x(k|k-1)+ Wv(k)
P(k|k) = P(k|k-1) - W S W^T

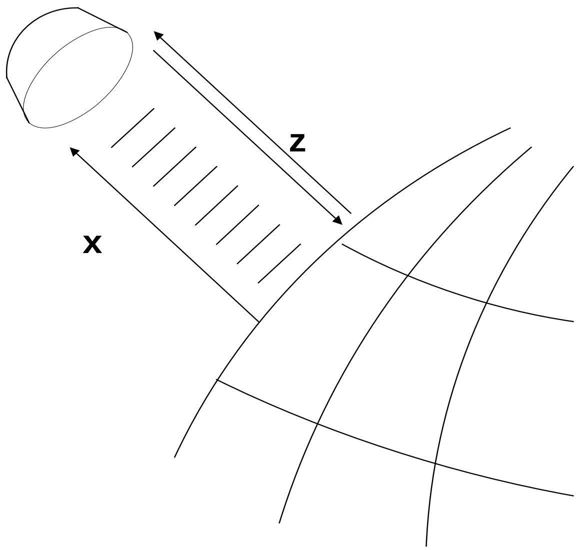
Crucial Characteristics

- Asynchronisity
- Prediction Covariance Inflation
- Update Covariance Deflation
- Observability
- Correlations

Example - Landing on Mars

A fine Movie

Mars Lander Example



$$\mathbf{x}(k) = \begin{bmatrix} h \\ \dot{h} \end{bmatrix}$$

state

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{w}(k)$$

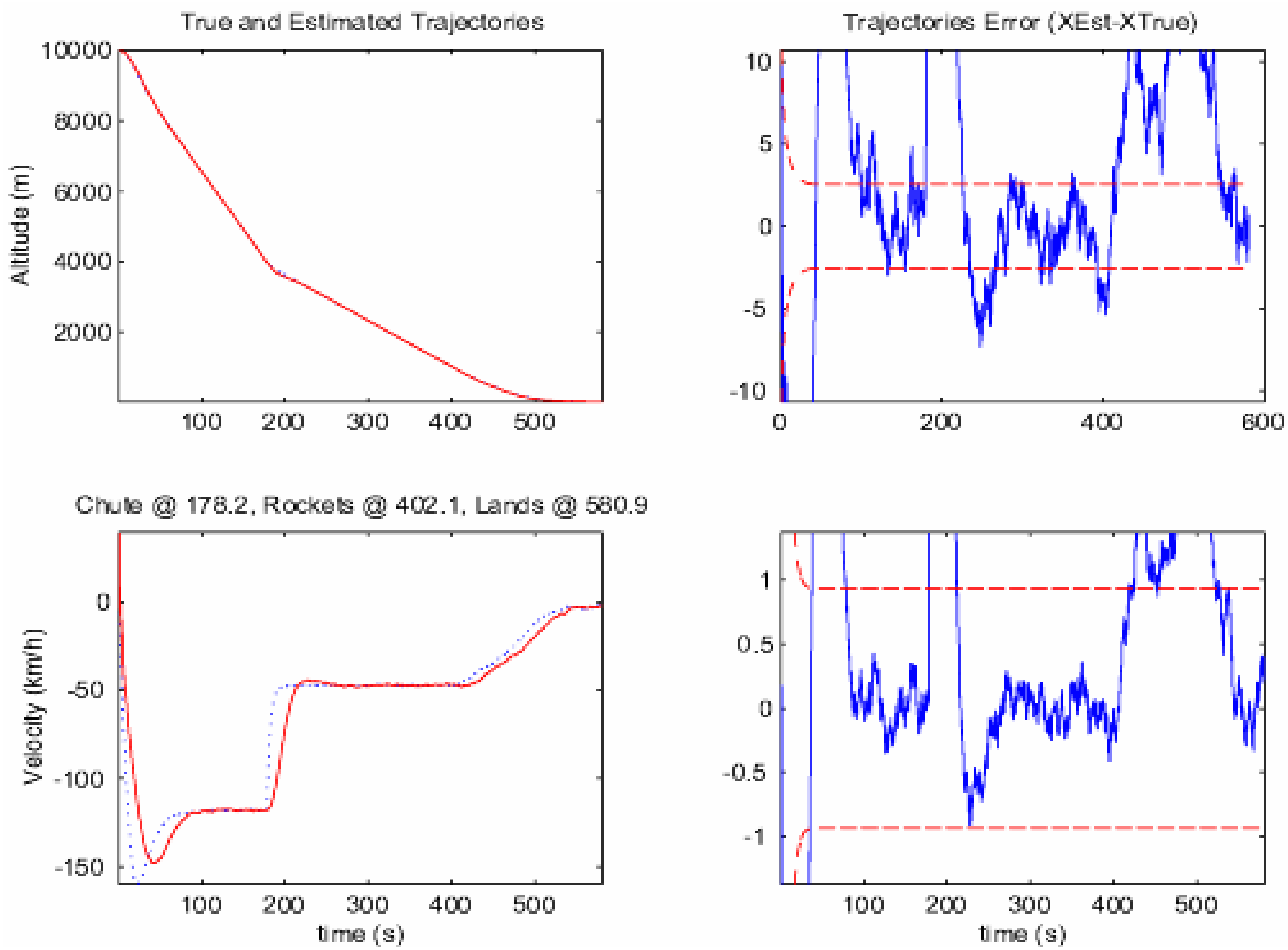
$$\mathbf{z}(k) = \begin{bmatrix} \frac{2}{c} & 0 \end{bmatrix} \begin{bmatrix} h \\ \dot{h} \end{bmatrix} + \mathbf{w}(k)$$

obs model

$$\mathbf{x}(k) = \underbrace{\begin{bmatrix} 1 & \delta T \\ 0 & 1 \end{bmatrix}}_{\mathbf{F}} \mathbf{x}(k-1) + \underbrace{\begin{bmatrix} \frac{\delta T^2}{2} \\ \delta T \end{bmatrix}}_{\mathbf{G}} \mathbf{v}(k)$$

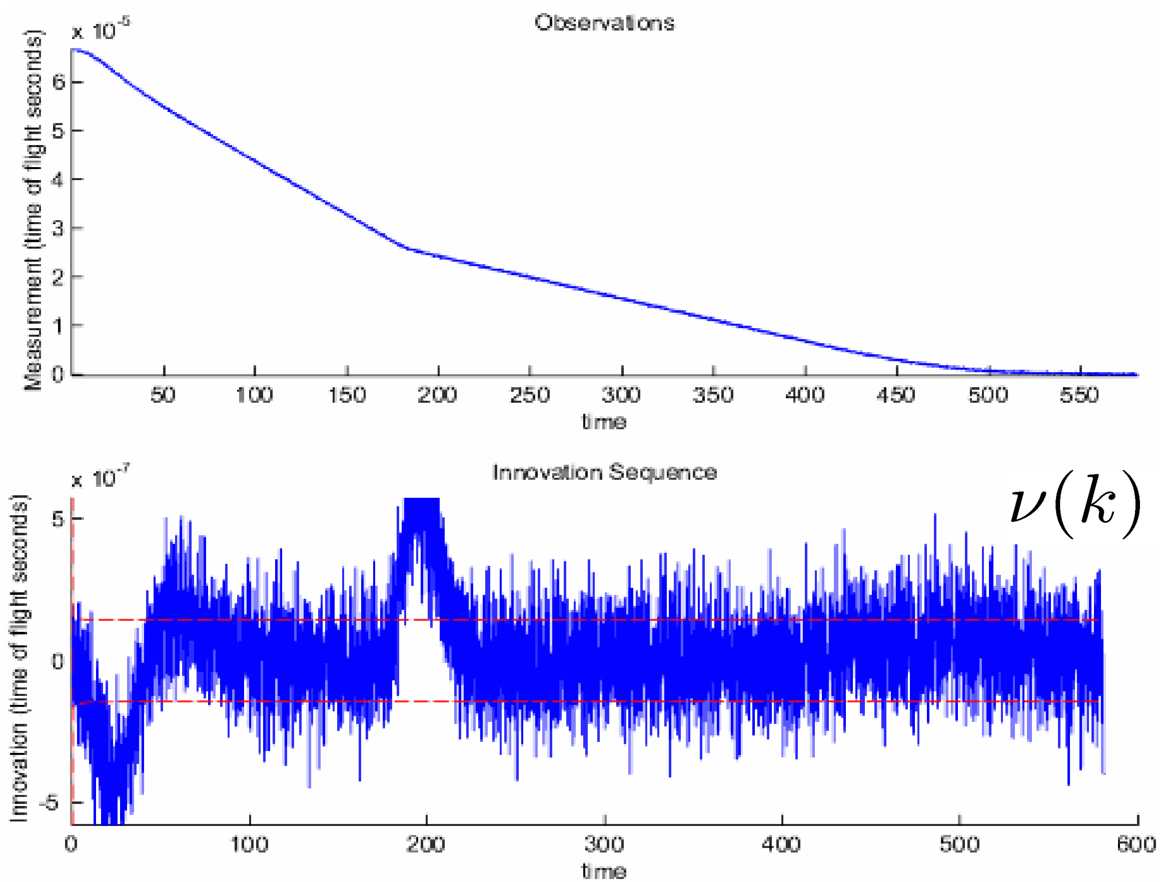
plant model

Trajectory



True and Estimated Trajectories
Altitude (m)
10000
8000
6000
4000
2000
100
200
300
400
500
Trajectories Error (XEst-XTrue)
10
5
0
-5
-10
0
200
400
600
Chute @ 178.2, Rockets @ 402.1, Lands @ 560.9
Velocity (km/h)
0
-50
-100
-150
time (s)
1
0.5
0
-0.5
-1
time (s)

Innovation



$$\nu(k) = \mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)$$

Non–Linear Kalman Filtering

Same trick as in Non-linear Least Squares:

- Linearise around a current estimate using jacobian
- Problem becomes linear again

Complete derivation is in the notes but…

To convert the linear Kalman Filter to the Extended Kalman Filter simply replace $\mathbf{F}$ with $\nabla\mathbf{F}_{\mathbf{x}}$ and $\mathbf{H}$ with $\nabla\mathbf{H}_{\mathbf{x}}$ **in the covariance and gain calculations only**. The jacobians are always evaluated at the best available estimate ($\hat{\mathbf{x}}(k-1|k-1)$ for $\nabla\mathbf{F}_{\mathbf{x}}$ and $\hat{\mathbf{x}}(k|k-1)$ for $\nabla\mathbf{H}_{\mathbf{x}}$

Prediction:

$$\underbrace{\hat{\mathbf{x}}(k|k-1)}_{\text{predicted state}} = \overbrace{\mathbf{f}(\underbrace{\hat{\mathbf{x}}(k-1|k-1)}_{\text{old state est}},\ \underbrace{\mathbf{u}(k)}_{\text{control}}, k)}^{\text{plant model}}$$

$$\underbrace{\mathbf{P}(k|k-1)}_{\text{predicted covariance}} = \nabla\mathbf{F_x}\underbrace{\mathbf{P}(k-1|k-1)}_{\text{old est covariance}}\nabla\mathbf{F_x}^T + \underbrace{\nabla\mathbf{G_v}\mathbf{Q}\nabla\mathbf{G_v}^T}_{\text{process noise}}$$

$$\underbrace{\mathbf{z}(k|k-1)}_{\text{predicted obs}} = \overbrace{\mathbf{h}(\hat{\mathbf{x}}(k|k-1))}^{\text{observation model}}$$

Update:

$$\underbrace{\hat{\mathbf{x}}(k|k)}_{\text{new state estimate}} = \overbrace{\hat{\mathbf{x}}(k|k-1) + \mathbf{W}\ \underbrace{\nu(k)}_{\text{innovation}}}^{\text{prediction and correction}}$$

$$\underbrace{\mathbf{P}(k|k)}_{\text{new covariance estimate}} = \underbrace{\mathbf{P}(k|k-1) - \mathbf{W}\mathbf{S}\mathbf{W}^T}_{\text{update decreases uncertainty}} \quad \nu(k) = \overbrace{\mathbf{z}(k)}^{\text{measurement}} - \mathbf{z}(k|k-1)$$

$$\mathbf{W} = \underbrace{\mathbf{P}(k|k-1)\nabla\mathbf{H_x}^T\mathbf{S}^{-1}}_{\text{kalman gain}}$$

$$\mathbf{S} = \underbrace{\nabla\mathbf{H_x}\mathbf{P}(k|k-1)\nabla\mathbf{H_x}^T + \mathbf{R}}_{\text{Innovation Covariance}}$$

Recalculate Jacs at each iteration

Lets be Clear about Jacobians..

Consider a function $\mathbf{f}$ which maps a vector $\mathbf{x}$ $(m \times 1)$ to a vector $\mathbf{y}$ $(n \times 1)$. We also allow the transformationto be a function of another vector $\mathbf{u}$.

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Lets think of this as a stacked vector of functions (one for each element of $\mathbf{y}$) and lets write out $\mathbf{x}$ element by element:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f_1(\overbrace{x_1, \cdots x_m}^{\mathbf{x}}, \mathbf{u}) \\ \vdots \\ f_n(x_1, \cdots x_m, \mathbf{u}) \end{bmatrix}$$

The term $\nabla \mathbf{F_x}$ is understood to be the jacobian of $(\mathbf{f})$ with respect to $\mathbf{x}$ evaluated at a specified value of $\mathbf{x}$:

$$\nabla \mathbf{F_x} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_m} \\ \vdots & & \vdots \\ \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_m} \end{bmatrix}$$

This is a matrix of functions. We'll always be told what value of x to use to turn it into a matrix of numbers.

Had we be talking about $\nabla \mathbf{F_u}$ then we would have differentiated w.r.t $\mathbf{u}$ and ended up with a $n \times dim(\mathbf{u})$ matrix

And now the fun part…

Using a EKF to do some robotics….

But you'll have to wait until the next lecture…