ME/CS 132:
State Estimation & Localization

Lecture 1/6:
Introduction and Batch Filtering

Nicolas Hudson

2/15/2011

State Estimation & Localization Overview

- (1/6) TODAY: Introduction
  - What is estimation?
  - Review of Probability
  - Least Squares
  - Non-Linear Least Squares

- (2/6) Linear Kalman Filter
- (3/6) Extended Kalman Filter
- (4/6) Particle Filters
- (5/6) Simultaneous Localization and Mapping (SLAM)
- (6/6) Issues in SLAM

What is Estimation?

- *We need to infer the STATE of a robot or the world (x), based upon measurements of a quantity (z) dependant on x.*

- We will view estimation as analyzing the conditional probability density function:

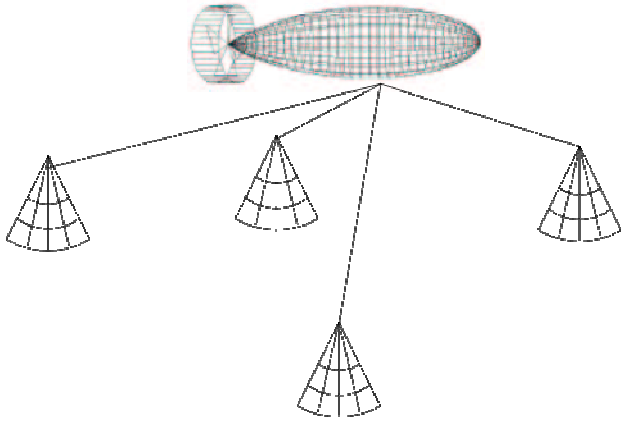
$$p(x \mid z)$$

"state" "measurements"

JPL's AeroBot

- Localization (Homework 1)
  - We have a KNOWN map, and 'see' multiple KNOWN landmarks
  - Estimator STATE = robot pose = [x, y, z, roll, pitch, yaw]

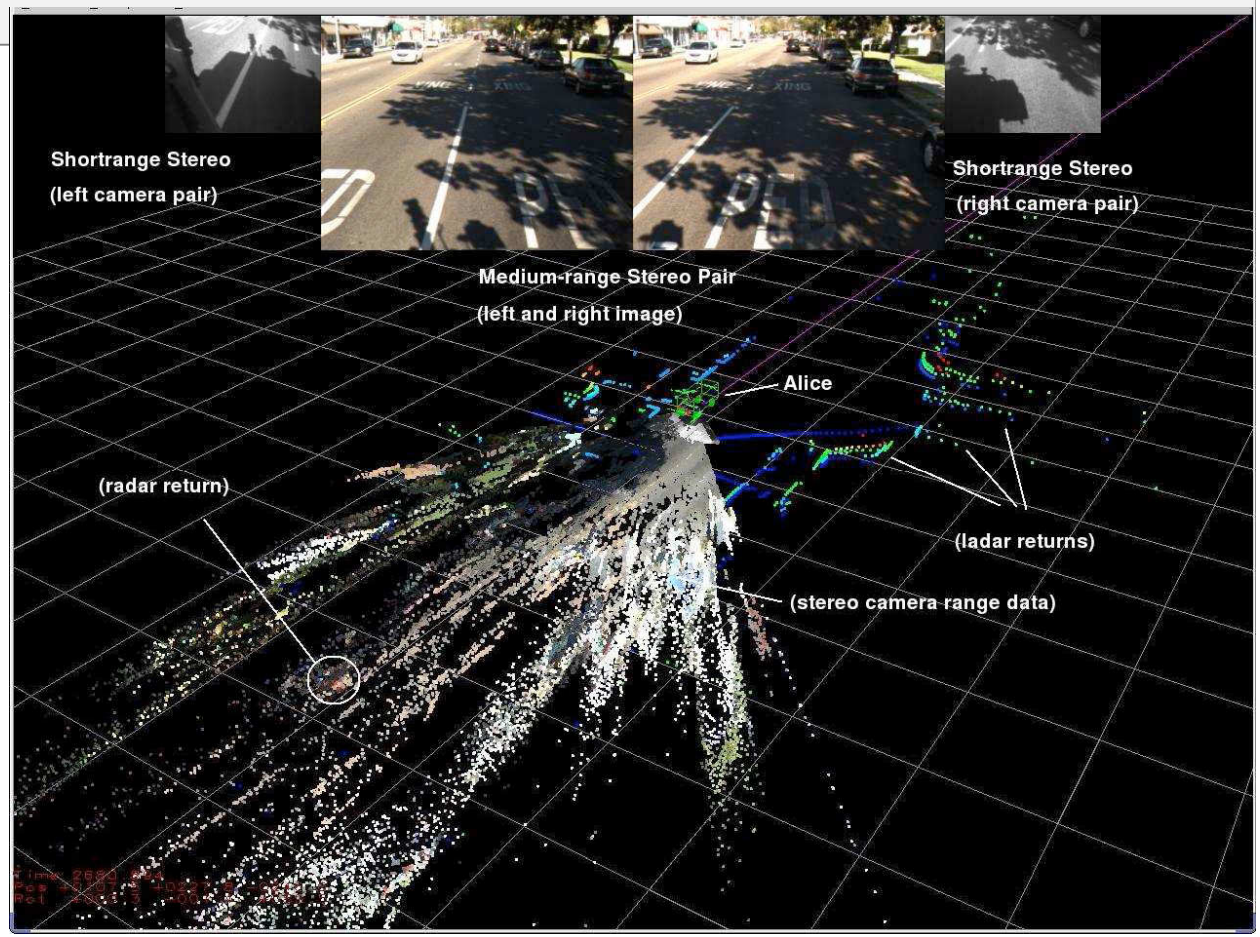
JPL Aerobot



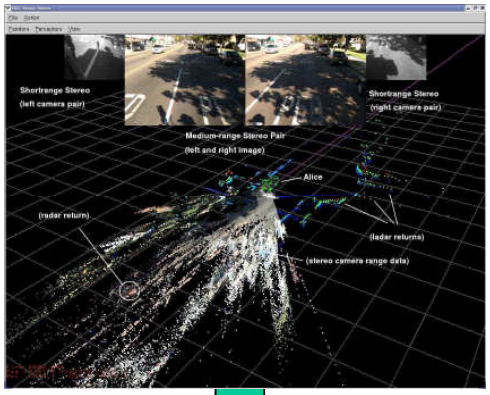
Camera can get 'bearing'
To landmarks

The DARPA Urban Challenge: An Example

- Simultaneous Localization and Mapping: (Lecture 5)
  - localize yourself AND build the map at the same time
  - Estimator STATE = robot pose + world object locations



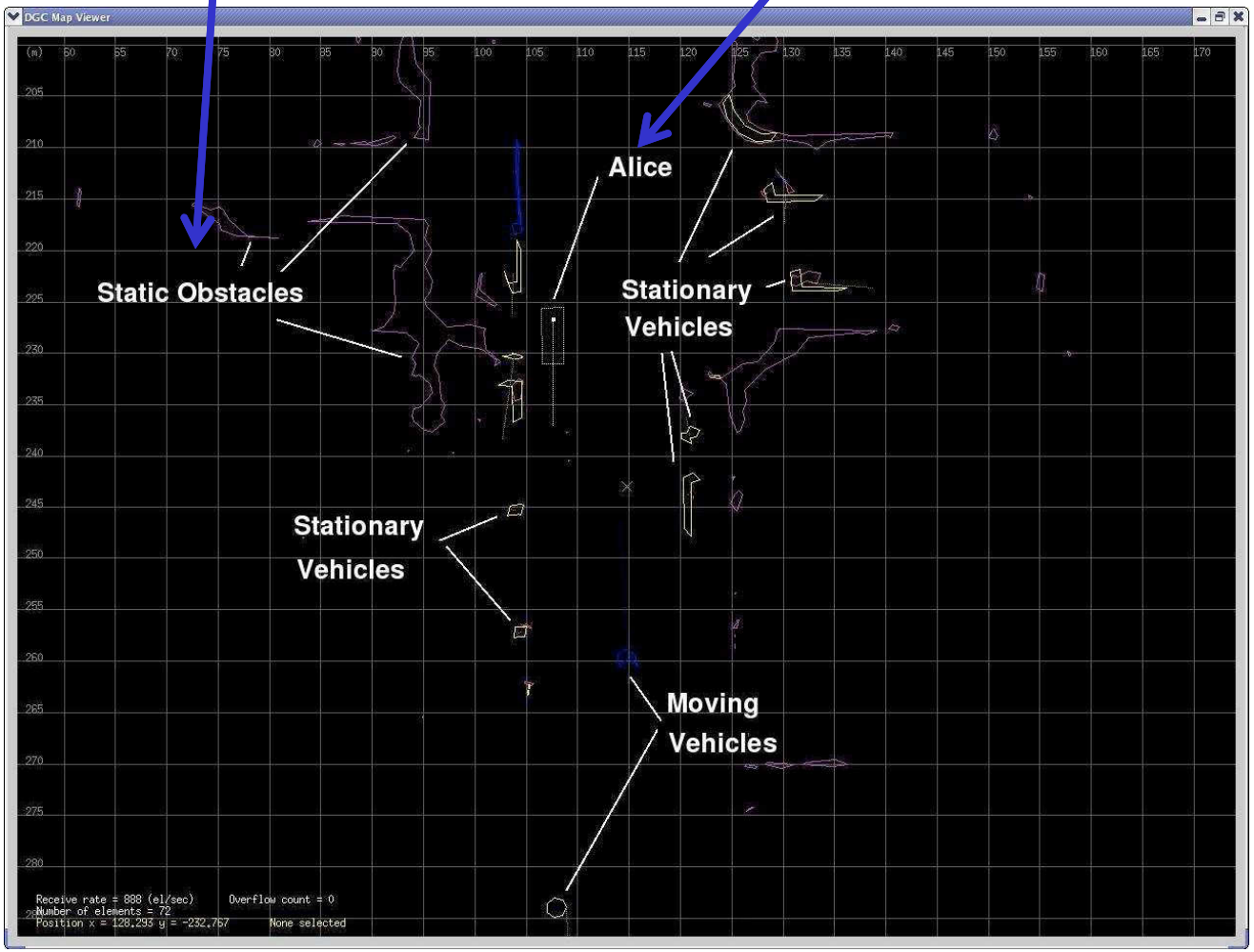
Caltech's DARPA Urban Challenge



Measurements:
GPS,
IMU,
Lidar,
Vision,
Radar
Odometry

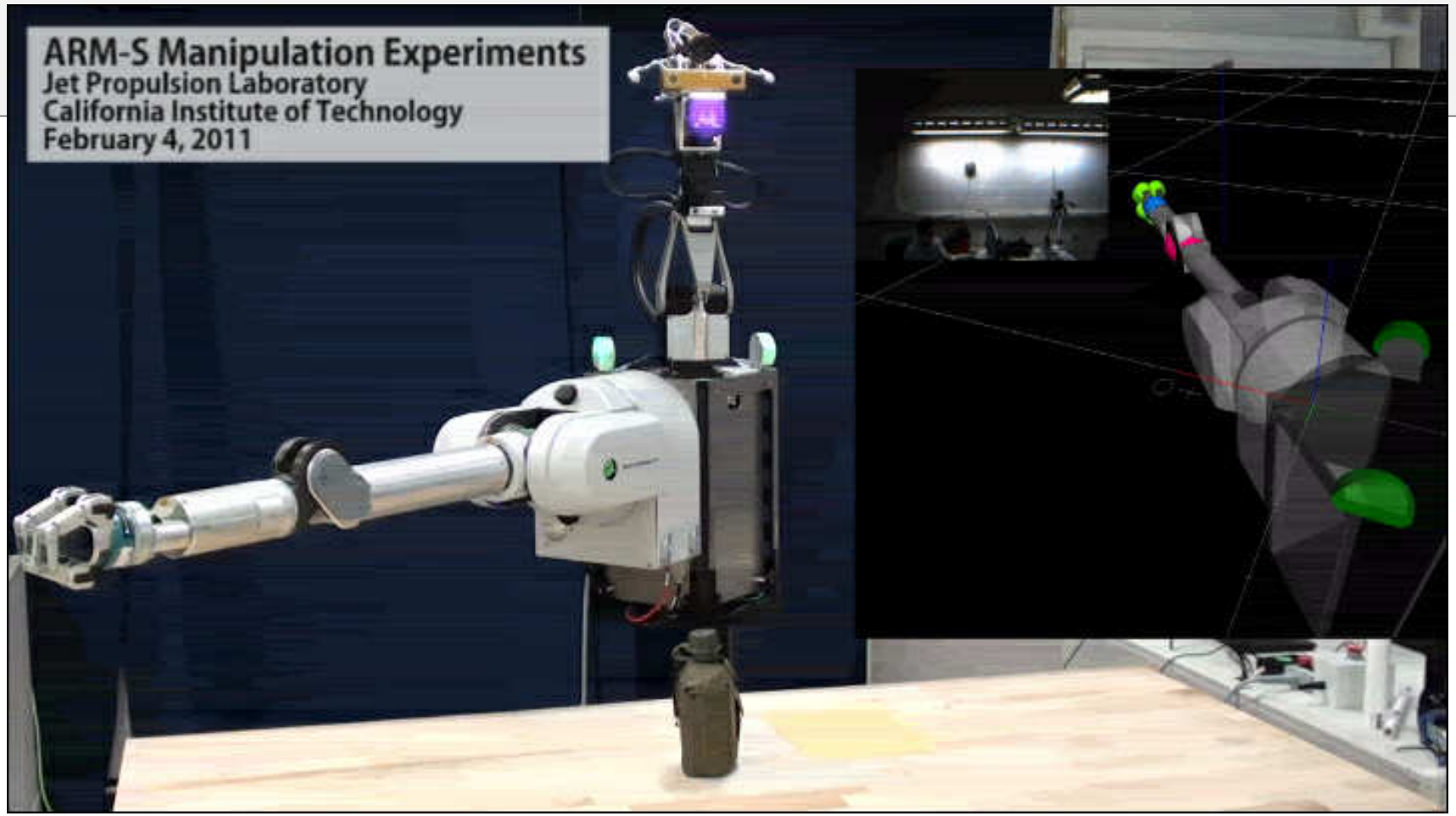
Map

Pose



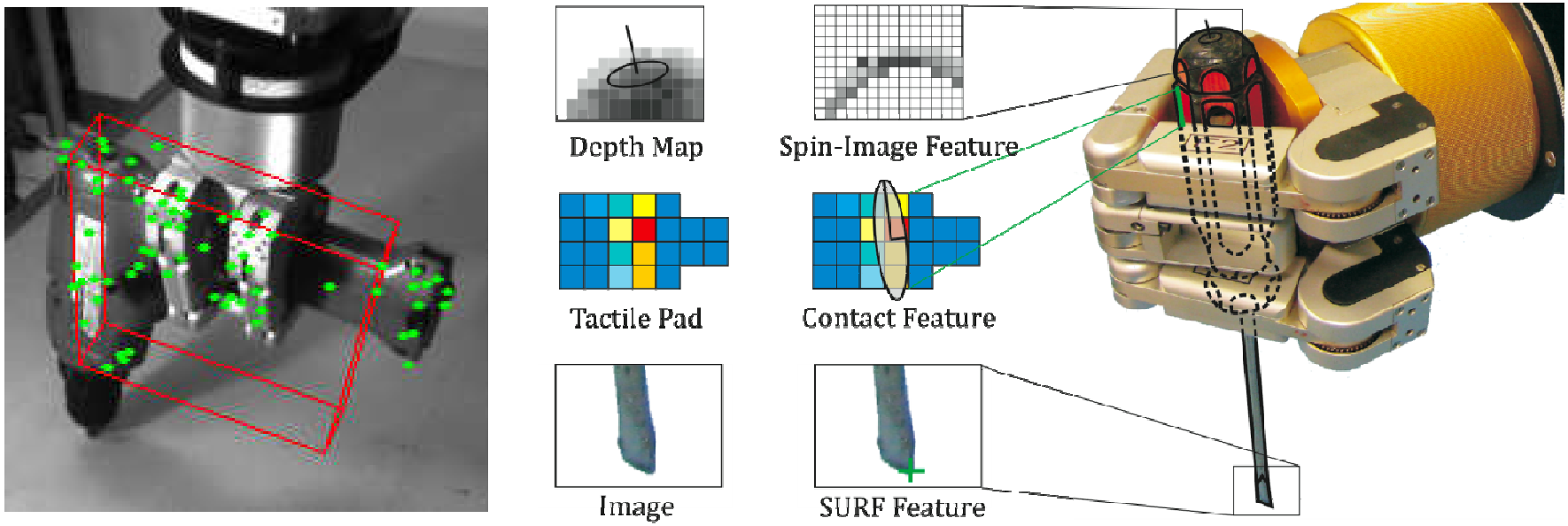
JPL's DARPA Autonomous Robotic Manipulation:

- 3D Object localization:
  - Estimator STATE = object pose



DARPA Autonomous Robotic Manipulation:

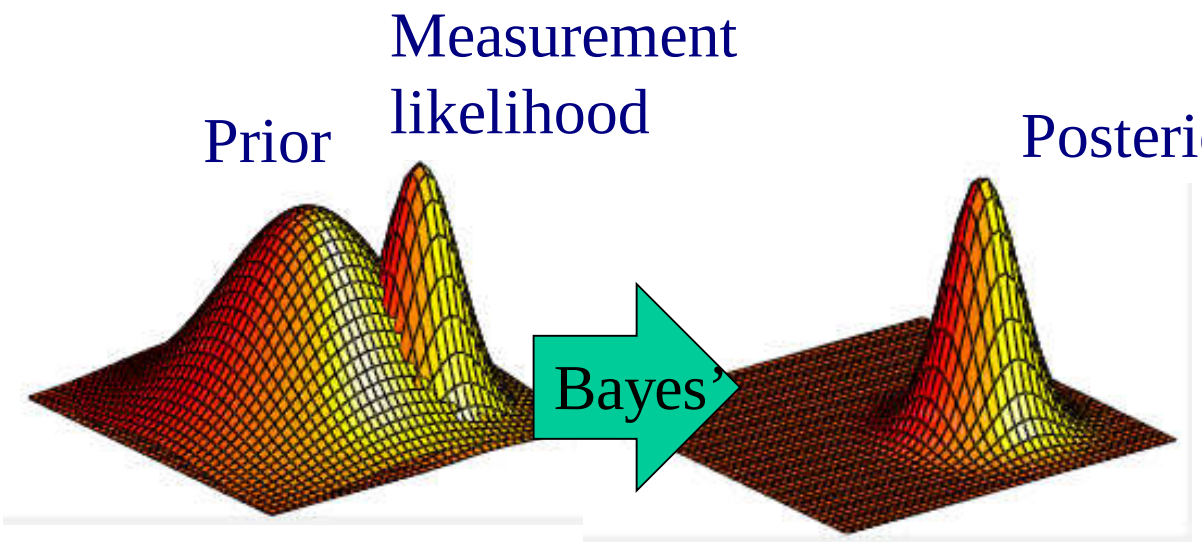
- What we're up to: SENSOR FUSION:
  - Stereo vision gives 3D feature locations of objects
  - Hand tactile pads give contact information
  - Force sensor can measure weight and center of mass



So what do all of these have in common?

- They all use probability theory and probability density functions as a foundation to infer the state of the system from measurements

e.g.



Review of Probability (Next)!

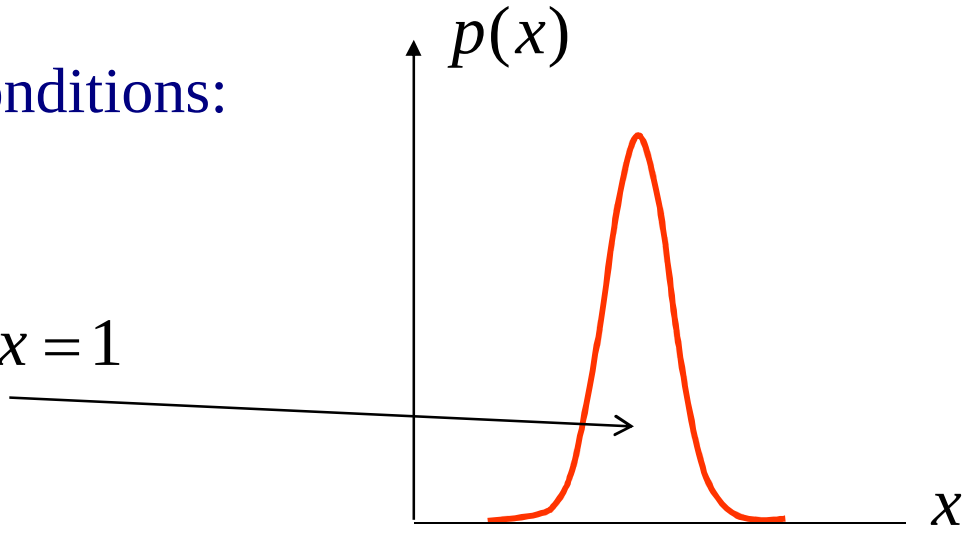
Brief Review of Probability

We are interested the probability density function (pdf) of continuous variables.

A pdf satisfies two conditions:

$$p(x) \geq 0$$

$$\int_{-\infty}^{\infty} p(x)dx = 1$$



The probability of the variable occurring in a set is the integral of the pdf over that set:

$$P(b > x > a) = \int_{a}^{b} p(x)dx$$

Terminology

Joint Probability

$p(x, y)$ *i.e.* $P(x, y \in D) = \int_D p(x, y)dxdy$

Conditional Probability

$p(x \mid z)$ *What is the probability of x given a value of z?*

Conditional Independence

$$p(x, y \mid z) = p(x \mid z)\, p(y \mid z)$$

Theorems

Marginalization [Sum Rule]

$$p(x) = \int p(x, y)dy$$

Product Rule

$$p(x, y) = p(x \mid y)\, p(y)$$

Theorem of Total Probability

$$p(x) = \int p(x \mid y)\, p(y)dy$$

Bayes Rule

$$p(y \mid x) = \frac{p(x \mid y)\, p(y)}{p(x)} = \frac{p(x \mid y)\, p(y)}{\int p(x \mid y)\, p(y)dy}$$

likelihood *prior*

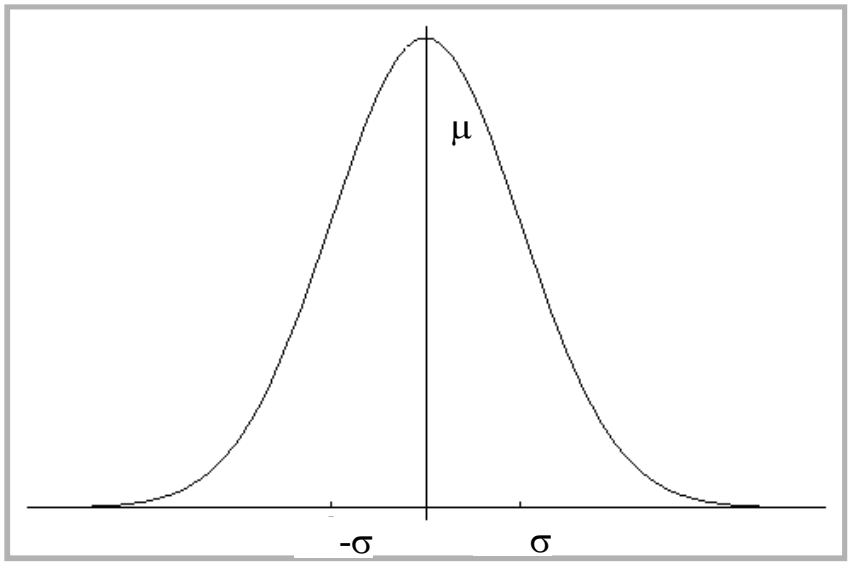
posterior

Probability Review: Gaussians

Univariate

$$p(x) \sim N(\mu, \sigma^2)$$

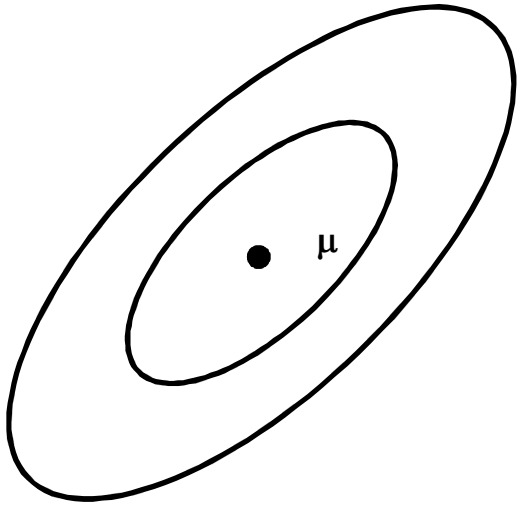
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$



Multivariate

$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad \text{where } \mathbf{x} \in \mathbf{R}^{\mathbf{d}}$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$



Expectation and Variance

- The average value of a function f(x) under a probability distribution p(x) is called the *expectation* of f(x):

$$E[f(x)] = \int f(x)\,p(x)\,dx$$

- Variation of a function

$$\begin{aligned}\mathrm{var}[f(x)] &= E\left[\left(f(x) - E\left[f(x)\right]\right)^2\right] \\ &= E\left[f(x)^2 - 2f(x)E\left[f(x)\right] + E\left[f(x)\right]^2\right] \\ &= E\left[f(x)^2\right] - E\left[f(x)\right]^2\end{aligned}$$

$$\int f(x)\left[\int f(x)\,p(x)\,dx\right]p(x)\,dx = \left[\int f(x)\,p(x)\,dx\right]^2$$

- Variation of a variable

$$\mathrm{var}[x] = E\left[x^2\right] - E\left[x\right]^2$$

Expectation and Covariance

Covariance (two variables)

$$\begin{aligned}\mathrm{cov}[x,y] &= E_{x,y}\left[\left(x-E[x]\right)\left(y-E[y]\right)\right] \\ &= E_{x,y}\left[xy - yE[x] - xE[y] + E[x]E[y]\right] \\ &= E_{x,y}\left[xy\right] - E\left[x\right]E[y]\end{aligned}$$

Expectation and Covariance

Covariance (two variables)

$$\begin{aligned}\mathrm{cov}[x,y] &= E_{x,y}\left[\left(x-E[x]\right)\left(y-E[y]\right)\right] \\ &= E_{x,y}\left[xy - yE[x] - xE[y] + E[x]E[y]\right] \\ &= E_{x,y}\left[xy\right] - E\left[x\right]E[y]\end{aligned}$$

Expectation is a linear operator:

$$\begin{aligned}E_{xy}[x+y] &= \int\int (x+y)\; p(x)p(y)dxdy \\ &= \int\int x\; p(x)p(y)dxdy + \int\int y\; p(x)p(y)dxdy \\ &= \int\left[\int x\; p(x)dx\right]p(y)dy + \ldots \\ &= E[x]\int p(y)dy + \ldots \\ &= E[x] + E[y]\end{aligned}$$

$$\begin{aligned}E_x\left[xE[y]\right] &= \int xp(x)E[y]dx \\ &= \int xp(x)dx\,E[y] \\ &= E[x]E[y]\end{aligned}$$

Expectation and Covariance

Covariance (two variables)

$$\begin{aligned}\text{cov}[x,y] &= E_{x,y}\left[\left(x-E[x]\right)\left(y-E[y]\right)\right] \\ &= E_{x,y}\left[xy - yE[x] - xE[y] + E[x]E[y]\right] \\ &= E_{x,y}\left[xy\right] - E\left[x\right]E[y]\end{aligned}$$

Covariance (two vectors)

$$\begin{aligned}\text{cov}[\mathbf{x},\mathbf{y}] &= E_{x,y}\left[\left(\mathbf{x}-E[\mathbf{x}]\right)\left(\mathbf{y}^{\mathbf{T}}-E[\mathbf{y}^{\mathbf{T}}]\right)\right] \\ &= E_{x,y}\left[\mathbf{x}\mathbf{y}^{\mathbf{T}}\right] - E\left[\mathbf{x}\right]E[\mathbf{y}^{\mathbf{T}}]\end{aligned}$$

Covariance (of a single vector)

$$\text{cov}[\mathbf{x}] = \text{cov}[\mathbf{x},\mathbf{x}]$$

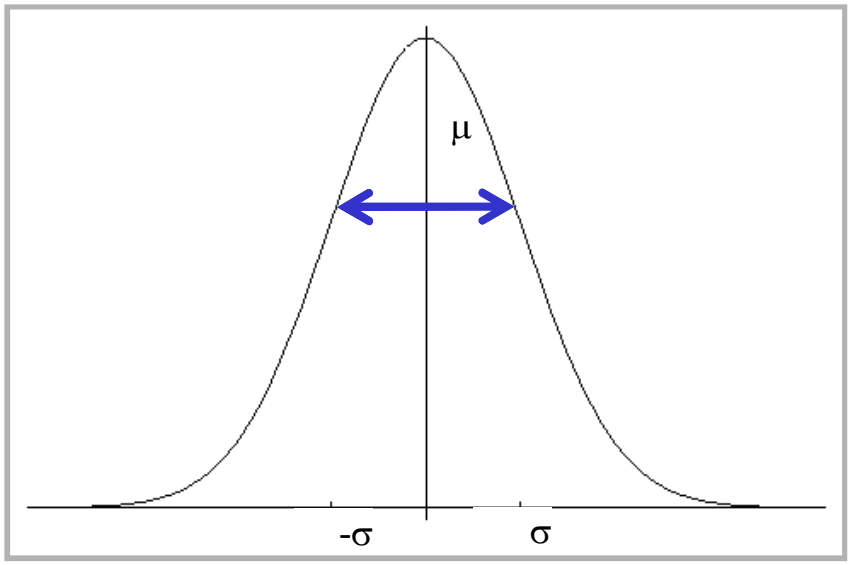
Probability Review: Gaussians

Univariate

$$p(x) \sim N(\mu, \sigma^2)$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

$$\sigma^2 = \text{var}[x] = \text{cov}[x, x] = E\left[\left(x - E[x]\right)^2\right]$$

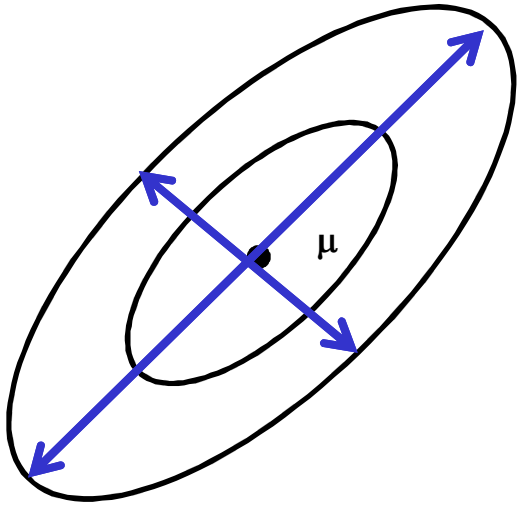


Multivariate

$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad \text{where } \mathbf{x} \in \mathbf{R}^{\mathbf{d}}$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

with $E[\mathbf{x}] = \boldsymbol{\mu}$ $\text{cov}[\mathbf{x}] = \boldsymbol{\Sigma}$



ML Estimators

- Maximum Likelihood Estimation: suppose a measurement *z* is related to the system state *x.* Measurements are corrupted by noise and can be specified in a Likelihood function:

$$L \equiv p(z \mid x)$$

- This is a *conditional* probability- where the distribution of z depends on x.

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

ML Estimators

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 $\mathcal{L} \triangleq p(\mathbf{z}|\mathbf{x})$.

$$\hat{\mathbf{x}}_{m.l} = \arg\max_{\mathbf{x}} p(\mathbf{z}|\mathbf{x}) \qquad (3.3)$$

In this example: $$p(\mathbf{z}|\mathbf{x}) = \frac{1}{C} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{x})^T \mathbf{P}^{-1}(\mathbf{z}-\mathbf{x})}$$

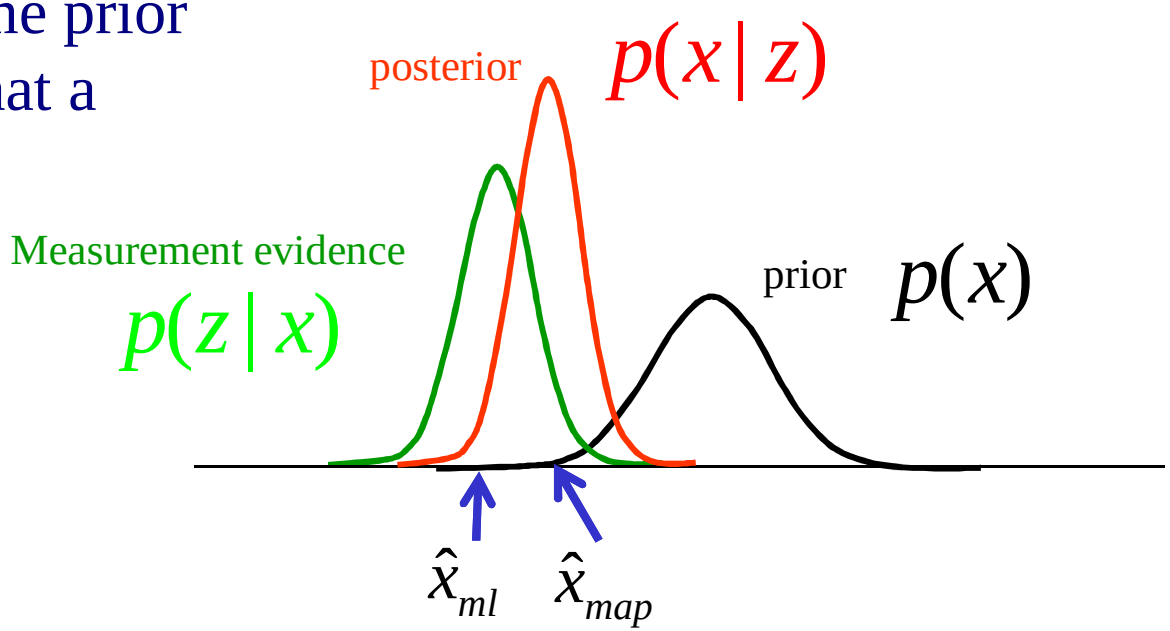
the ML estimate is $\hat{x} = z$

(here we really view the likelihood as a function of z, not x)

Maximum A-Posteriori

Suppose we have some prior information about what a variable should be?

$$p(x \mid z) = \frac{p(z \mid x)\, p(x)}{p(z)}$$
$$= C \times p(z \mid x)\, p(x)$$



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

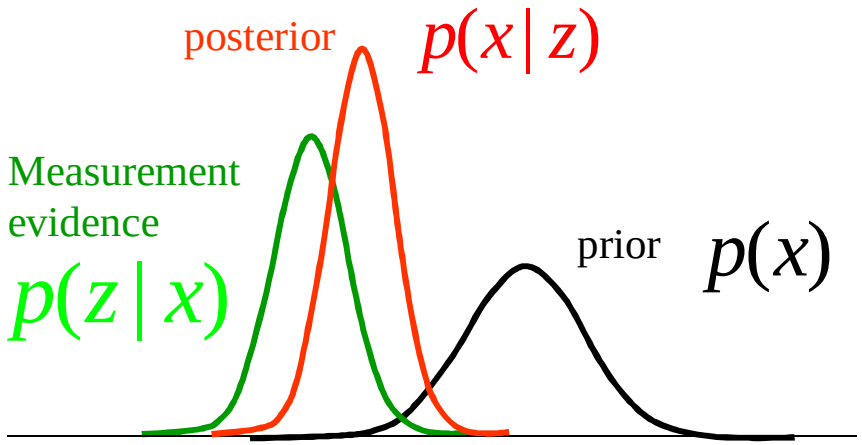
$$\hat{\mathrm{x}}_{map} = \arg\max_{\mathrm{x}} p(\mathrm{z}|\mathrm{x})p(\mathrm{x}) \qquad (3.4)$$

MAP Example

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

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

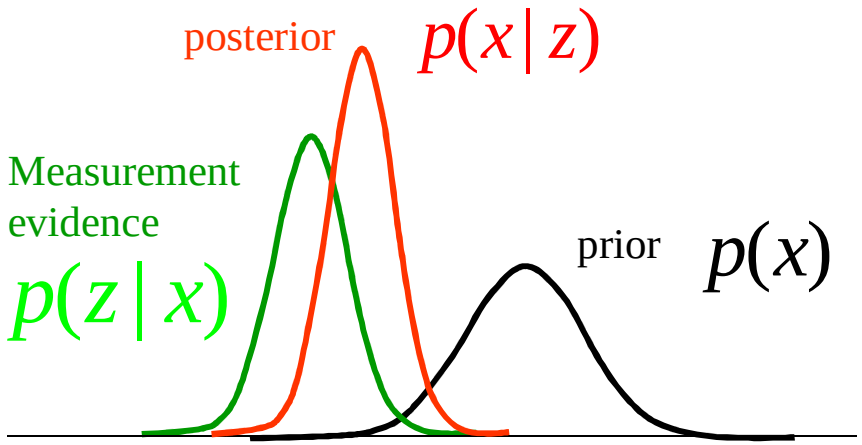
$$\begin{aligned} p(\mathbf{x}|\mathbf{z}) &= \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \\ &= C(\mathbf{z}) \times p(\mathbf{z}|\mathbf{x}) \times p(\mathbf{x}) \\ &= C(\mathbf{z}) \exp\{-\frac{(\mathbf{x}-\mu_p)^2}{2\sigma_p^2} - \frac{(\mathbf{z}-\mathbf{x})^2}{2\sigma_z^2}\} \end{aligned}$$



MAP Example

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

$$= C(\mathbf{z}) \exp\{-\frac{(x-\alpha)^2}{\beta^2}\}$$



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

Note: $\lim_{\sigma_p \to \infty} \alpha = z$

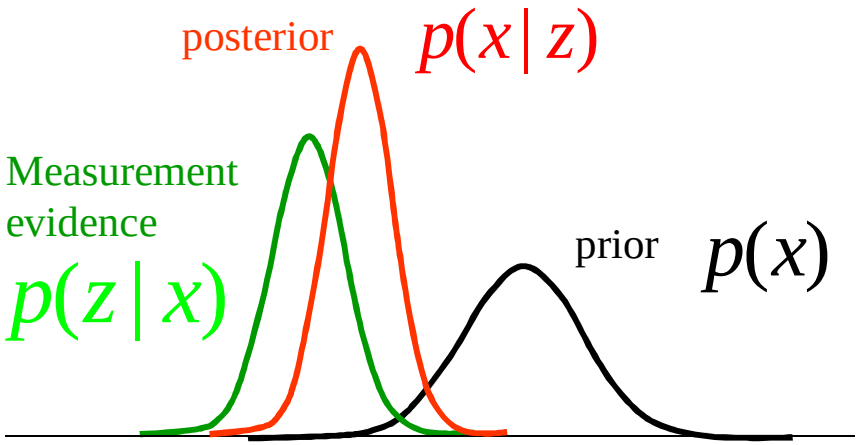
$$\beta^2 = \frac{\sigma_z^2 \sigma_p^2}{\sigma_z^2 + \sigma_p^2}$$ variance

Note: $\beta^2 < \sigma_z^2$

as: $$\frac{1}{\beta^2} = \frac{\sigma_z^2 + \sigma_p^2}{\sigma_z^2 \sigma_p^2} = \frac{1}{\sigma_p^2} + \frac{1}{\sigma_z^2}$$

MAP Example

If the prior has a large variance, then the MAP and ML estimates converge



We have gained information about the state! The posterior is more informative than the prior or likelihood.

Note: $\lim_{\sigma_p \to \infty} \alpha = z$

Note: $\beta^2 < \sigma_z^2$

as: $$\frac{1}{\beta^2} = \frac{\sigma_z^2 + \sigma_p^2}{\sigma_z^2 \sigma_p^2} = \frac{1}{\sigma_p^2} + \frac{1}{\sigma_z^2}$$

Recursive Estimation

We can take this idea of a prior further:

Say we have multiple measurements

$$p(x \mid z_{1:k}) = \frac{p(z_{1:k} \mid x)\, p(x)}{p(z_{1:k})} \qquad z_{1:k} = [z_1, z_2, ..., z_k]$$

We typically assume the measurements are conditionally independent

$$p(z_{1:k} \mid x) = p(z_1 \mid x)\, p(z_2 \mid x) ... p(z_3 \mid x)$$

Then we can recursively update with each new measurement

$$p(x \mid z_{1:k}) = \frac{p(z_k \mid x, z_{1:k-1})\, p(x \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}$$

Recursive Estimation

With just one measurement this is the same as the last example:

$$p(x \mid z_1) = \frac{p(z_1 \mid x)\, p(x)}{p(z_1)}$$

With another measurement we can further refine or estimate:

$$p(x \mid z_1, z_2) = \frac{p(z_2 \mid x)\, p(x \mid z_1)}{p(z_2 \mid z_1)}$$

$$\vdots$$

$$p(x \mid z_{1:k}) = \frac{p(z_k \mid x, z_{1:k-1})\, p(x \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}$$

The rest of todays Lecture:

- Focus on how to estimate a fixed state:
- In this case the state does not “move”
- We will process the state in a “batch” process, where have multiple measurements at once.
  - Least Squares
  - Non-linear least squares

- Next lecture we will talk about how the state x will evolve in time!

Least Squares

Consider a vector of measurements ($\mathbf{z}$) related to the state ($\mathbf{x}$) by the linear equation:

$$\mathbf{z} = H\mathbf{x}$$

You may already know that the inverse won't exist unless H is square:

$$\mathbf{x} = H^{-1}\mathbf{z}$$

Do you recall pseudo inverses? :

$$\mathbf{x} = (H^T H)^{-1} H^T \mathbf{z}$$

We will show this in our estimation framework.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| H\mathbf{x} - \mathbf{z} \right\|$$

$$= \arg\min_{\mathbf{x}} \left(\left(H\mathbf{x} - \mathbf{z} \right)^T \left(H\mathbf{x} - \mathbf{z} \right) \right)$$

Least Squares

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\left(H\mathbf{x} - \mathbf{z} \right)^T \left(H\mathbf{x} - \mathbf{z} \right) \right)$$

We can solve the above convex optimization problem using minimization:

$$0 = \frac{\partial}{\partial \mathbf{x}} \left(\left(H\mathbf{x} - \mathbf{z} \right)^T \left(H\mathbf{x} - \mathbf{z} \right) \right)$$

$$0 = \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{x}^{\mathbf{T}} H^{\mathbf{T}} H\mathbf{x} - \mathbf{x}^{\mathbf{T}} H^{\mathbf{T}} \mathbf{z} - \mathbf{z} H\mathbf{x} \right)$$

$$0 = 2H^T H\mathbf{x} - 2H^{\mathbf{T}} \mathbf{z}$$

$$\Rightarrow \mathbf{x} = \left(H^T H \right)^{-1} H^T \mathbf{z}$$

Least Squares – Polynomial Example

Find the parameters of the quadratic equation ('line fitting') using a set of inputs *u* and measurements z

$$z = a_0 + a_1 u + a_2 u^2 + noise$$

$$\mathbf{z} = \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(n) \end{bmatrix} \qquad H = \begin{bmatrix} 1 & u(1) & u(1)^2 \\ 1 & u(2) & u(2)^2 \\ & \vdots & \\ 1 & u(n) & u(n)^2 \end{bmatrix} \qquad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

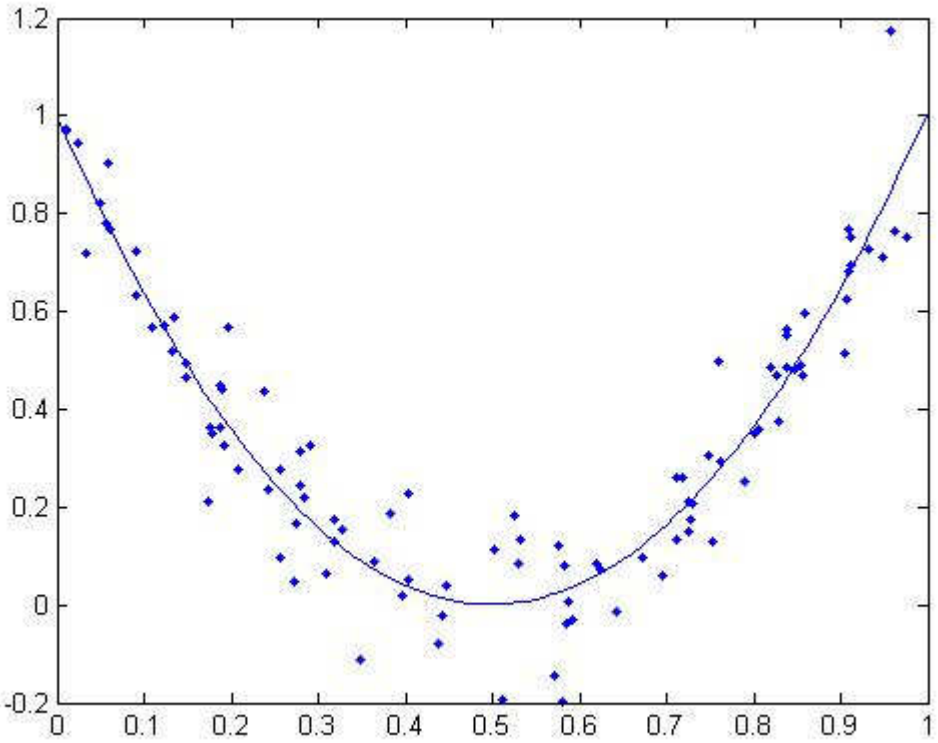
Note that "a" is the state – not "x"

$$\hat{\mathbf{a}} = \left(H^T H\right)^{-1} H\mathbf{z}$$

Polynomial Example

```
a=[1 -4 4];

%generate data:
for i=1:100
    x(i)= rand;
    z(i)= a(1)+a(2)*x(i)+a(3)*x(i)^2 + 0.1*randn;
    H(i,:)=[1 x(i) x(i)^2];
end
 %plot the data z
plot(x,z,'.')

%LEAST SQUARES:
a_est=inv(H'*H)*H'*z';

%plot the least square curve
x=0:0.01:1;
y=a_est(1)+a_est(2).*x+a_est(3).*x.^2;

hold on;
plot(x,y)
hold off;
```

```
a_est =

    0.9719
   -3.9566
    4.0057
```



Bayesian method for least squares

We can also do this in the Bayesian recursive framework
(but it is a little more complicated!)

$$z = Fx + N(0, \sigma^2 I) \qquad \begin{array}{l} z \in \mathbb{R}^m \\ x \in \mathbb{R}^n \\ F \in \mathbb{R}^{m \times n} \end{array}$$

Given the prior $p(x) = N(\mu_0, \Sigma_0)$ and the data z, we can find:

$$p(x \mid z, \sigma^2) = N(\mu, \Sigma)$$

$$\mu = \mu_0 + \Sigma_0 F^T (\sigma^2 I + F \Sigma_0 F^T)^{-1} (z - F\mu_0)$$

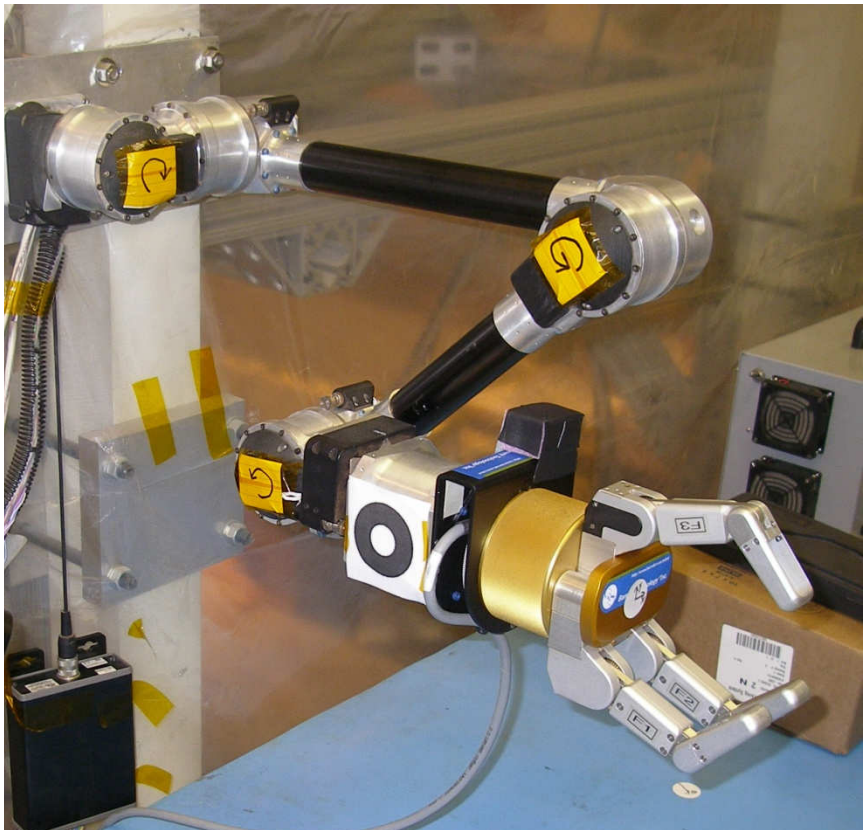
$$\Sigma = \left(\Sigma_0 + \frac{1}{\sigma^2} F^T F \right)^{-1}$$

What's cool about this is it's always invertible because of the prior

Least Squares Example: Learning Stiffness

- Estimate the stiffness K of the environment

  Force = K * displacement



Least Squares Example: Learning Stiffness

- Estimate the stiffness K of the environment

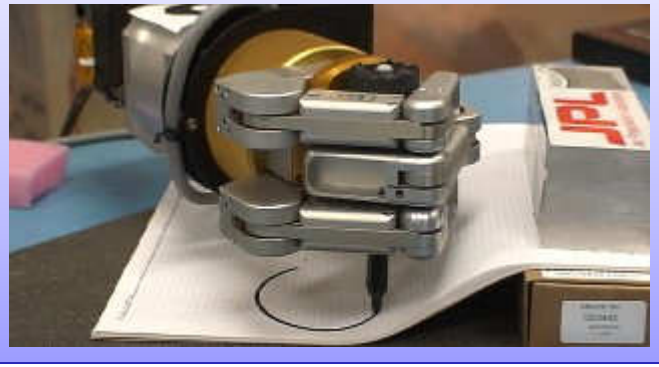
  Force = K * displacement

Contact with surface: ***Learn stiffness*** → Force / Position Control: ***Adapted gains*** → Finished:

Hard Surface



Soft Surface


Least Squares Example: Learning Stiffness

Work with Ruslan Kurdyumov (Caltech SURF- Now a ME at Stanford)

Non-linear Least Squares

Say we now have a non-linear measurement model:

$$\mathbf{z} = h(\mathbf{x})$$

We can still solve the minimization problem :

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| h(\mathbf{x}) - \mathbf{z} \right\|^2$$

We do this in an iterative way :

Starting with an initial guess: x_0

We now minimize: $\delta = \arg\min_{\delta} \left\| h(\mathbf{x}_1) - \mathbf{z} \right\|^2$ where $x_1 = x_0 + \delta$

Non-linear Least Squares

To minimize: $\delta = \arg\min_{\delta} \left\| h(\mathbf{x}_1) - \mathbf{z} \right\|^2$ where $x_1 = x_0 + \delta$

We do a Taylor Series Expansion:

$$h(x_0 + \delta) = h(x_0) + \nabla H_{x_0} \delta$$

Now
$$\left\| h(x_1) - z \right\|^2 = \left\| h(x_0) + \nabla H_{x_0} \delta - z \right\|^2$$
$$= \left\| \nabla H_{x_0} \delta - (z - h(x_0)) \right\|^2$$

where
$$\nabla H_{x_0} = \frac{\delta h}{\delta x} = \begin{bmatrix} \frac{\delta h_1}{\delta x_1} & \cdots & \frac{\delta h_1}{\delta x_m} \\ \vdots & & \vdots \\ \frac{\delta h_n}{\delta x_1} & \cdots & \frac{\delta h_n}{\delta x_m} \end{bmatrix}$$

Non-linear Least Squares

Note that calculating δ is a linear least squares problem:

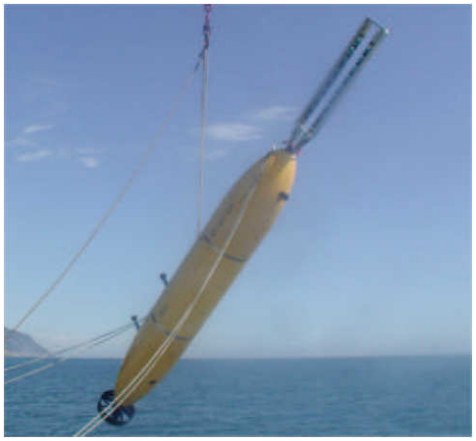
$$\left\|h(x_1)-z\right\|^2=\left\|\underbrace{\nabla H_{x_0}}_{A}\delta-\underbrace{(z-h(x_0))}_{b}\right\|^2$$

$$\Rightarrow\quad \delta=\left(\nabla H_{x_0}{}^T\nabla H_{x_0}\right)^{-1}\nabla H_{x_0}{}^T\left[z-h(x_0)\right]$$

Algorithm:

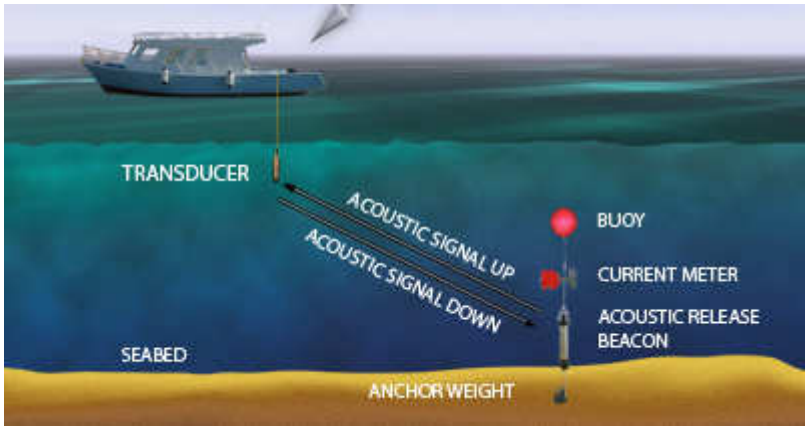
1. Begin with an initial guess $\hat{\mathbf{x}}$
2. Evaluate
$$\delta=\left(\nabla\mathbf{H}_{\hat{\mathbf{x}}}{}^T\mathbf{R}^{-1}\nabla\mathbf{H}_{\hat{\mathbf{x}}}\right)^{-1}\nabla\mathbf{H}_{\hat{\mathbf{x}}}{}^T\mathbf{R}^{-1}[\mathbf{z}-\mathbf{h}(\hat{\mathbf{x}})]$$
3. Set $\hat{\mathbf{x}}=\hat{\mathbf{x}}+\delta$
4. If $||\ \mathbf{h}(\hat{\mathbf{x}})-\mathbf{z}\ ||^2>\epsilon$ goto 1 else stop.

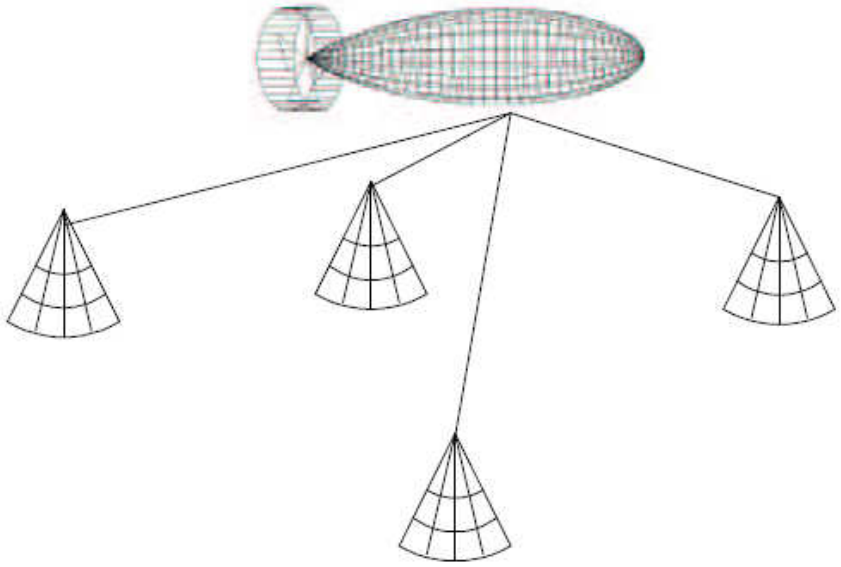
Example - Localization



Consider an UUV localizing using multiple acoustic beacon transponders.

- The UUV sends out a “call” signal
- Each beacon that gets the signal sends out an “acknowledge” signal.





Example - Localization

The difference between the "call" and "acknowledge" signals gives the distance to each beacon. (as the velocity *c* of sound is a constant).

We want to find the pose of the robot $\mathbf{x}_v = [x, y, z]^T$

Given the positions of each beacon $\mathbf{x}_{bi} = [x_{bi}, y_{bi}, z_{bi}]^T$

The observation model for all measurements is:

$$\mathbf{z} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \end{bmatrix}^T = h(\mathbf{x}_v)$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \frac{2}{c} \begin{bmatrix} || \mathbf{x}_{b1} - \mathbf{x}_v || \\ || \mathbf{x}_{b2} - \mathbf{x}_v || \\ || \mathbf{x}_{b3} - \mathbf{x}_v || \\ || \mathbf{x}_{b4} - \mathbf{x}_v || \end{bmatrix}$$

Example – Long Baseline Localization

The difference between the "call" and "acknowledge" signals gives the distance to each beacon. (as the velocity *c* of sound is a constant).

We want to find the pose of the robot $\mathbf{x}_v = [x, y, z]^T$

Given the positions of each beacon $\mathbf{x}_{bi} = [x_{bi}, y_{bi}, z_{bi}]^T$

The observation model for all measurements is:

$$\mathbf{z} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \end{bmatrix}^T = h(\mathbf{x}_v)$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \frac{2}{c} \begin{bmatrix} || \mathbf{x}_{b1} - \mathbf{x}_v || \\ || \mathbf{x}_{b2} - \mathbf{x}_v || \\ || \mathbf{x}_{b3} - \mathbf{x}_v || \\ || \mathbf{x}_{b4} - \mathbf{x}_v || \end{bmatrix}$$

$$\mathbf{z} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \end{bmatrix}^T = h(\mathbf{x}_v)$$

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \frac{2}{c} \begin{bmatrix} || \mathbf{x}_{b1} - \mathbf{x}_v || \\ || \mathbf{x}_{b2} - \mathbf{x}_v || \\ || \mathbf{x}_{b3} - \mathbf{x}_v || \\ || \mathbf{x}_{b4} - \mathbf{x}_v || \end{bmatrix}$$

$$\nabla \mathbf{H}_{\mathbf{x_v}} = -\frac{2}{rc} \begin{bmatrix} \Delta_{x1} & \Delta_{y1} & \Delta_{y1} \\ \Delta_{x2} & \Delta_{y2} & \Delta_{y2} \\ \Delta_{x3} & \Delta_{y3} & \Delta_{y3} \\ \Delta_{x4} & \Delta_{y4} & \Delta_{y4} \end{bmatrix}$$

$$\Delta_{xi} = x_{bi} - x$$
$$\Delta_{yi} = y_{bi} - y$$
$$\Delta_{zi} = z_{bi} - z$$
$$r = \sqrt{(x_{bi} - x)^2) + (y_{bi} - y)^2) + (z_{bi} - z)^2)}$$

With Real Data: (From MIT)