Mtrx 4700: Experimental Robotics

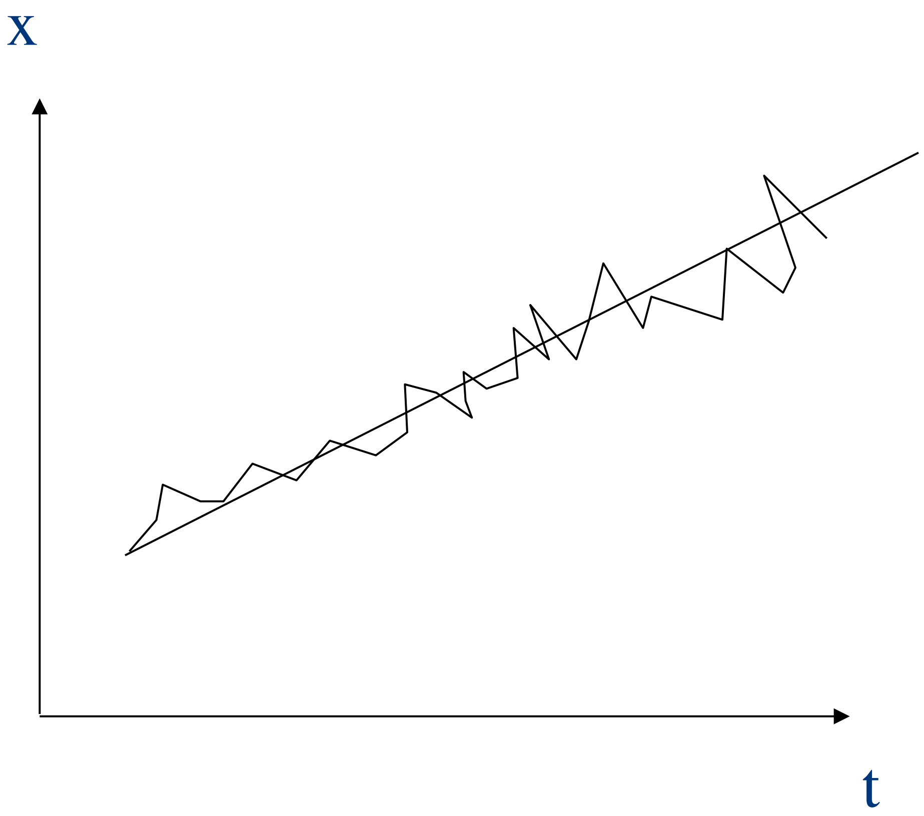
Data Fusion and Estimation

Dr. Stefan B. Williams

Course Outline

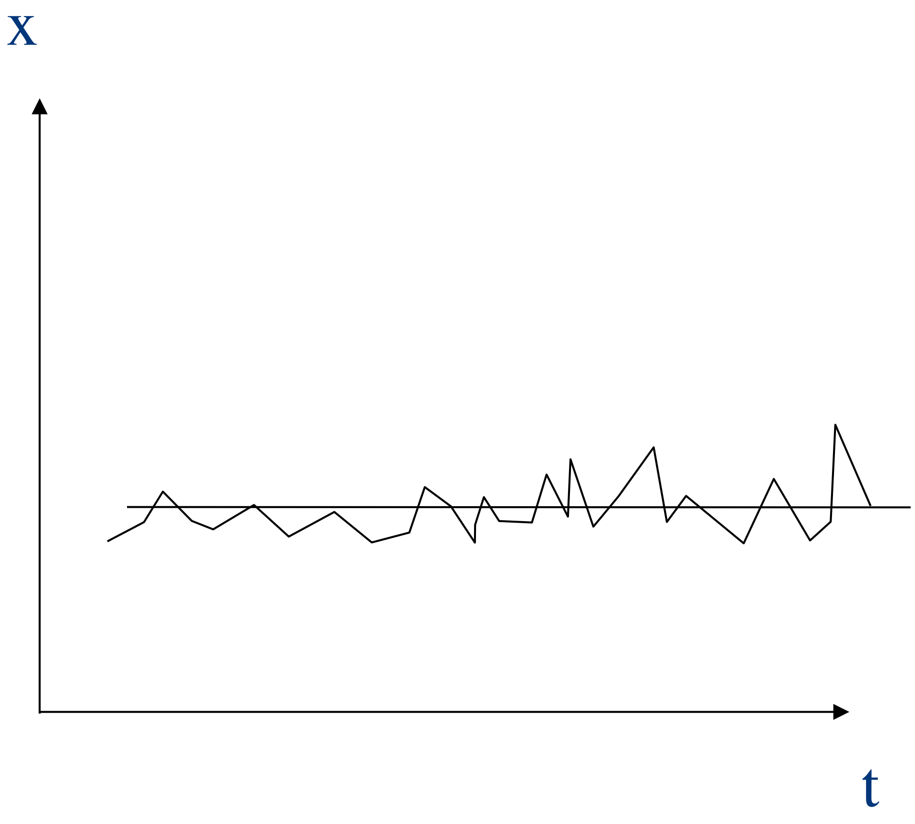
Week	Date	Content	Labs	Due Dates
1	5 Mar	Introduction, history & philosophy of robotics		
2	12 Mar	Robot kinematics & dynamics	Kinematics/Dynamics Lab	
3	19 Mar	Sensors, measurements and perception	"	
4	26 Mar	Robot vision and vision processing.	*No Tute (Good Friday)*	**Kinematics Lab**
	2 Apr	BREAK		
5	9 Apr	Localization and navigation	Sensing with lasers	
6	16 Apr	Estimation and Data Fusion	Sensing with vision	
7	23 Apr	Extra tutorial session (sensing)	Robot Navigation	**Sensing Lab**
8	30 Apr	Obstacle avoidance and path planning	Robot Navigation	
9	7 May	Extra tutorial session (nav demo)	Major project	**Navigation Lab**
10	14 May	Robotic architectures, multiple robot systems	"	
11	21 May	Robot learning	"	
12	28 May	Case Study	"	
13	4 June	Extra tutorial session (Major Project)	"	**Major Project**
14		Spare		

Random Signal



- There are many applications in which we might like to identify some signal of interest
- In many cases, the signal we are able to acquire may be corrupted by noise, bias or other effects which distort our understanding of the signal
- Data fusion and estimation techniques provide us with a mechanism for identifying the signal of interest

Estimation



- If we knew something about the underlying signal, we could select a method for identifying it
- For example, if we knew the value was some random constant, we could simply compute the *mean* of the noisy measurements

Estimating the Mean

- We could store each measurement and recompute the mean when each new measurement, z_i, arrives
  - $m_1 = z_1$
  - $m_2 = (z_1 + z_2)/2$
  - $m_3 = (z_1 + z_2 + z_3)/3$

$$m_n = \frac{\sum_{i=1}^{n} z_i}{n}$$

  - Over time the amount of memory needed and the complexity of recomputing the mean will increase as new samples are added

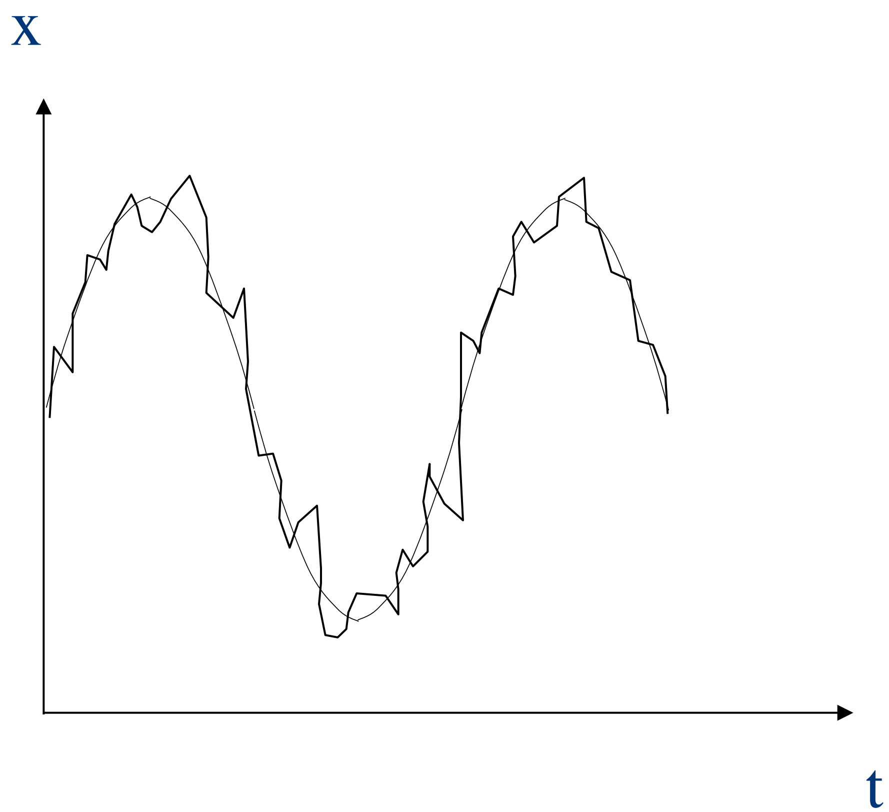
Recursively Estimating the Mean

- Alternatively, we could compute the mean recursively
  - $m_1 = z_1$
  - $m_2 = ½m_1 + ½\, z_1$
  - $m_3 = {}^2/_3 m_2 + {}^1/_3 z_3$

$$m_n = \left(\frac{n-1}{n}\right) m_{n-1} + \left(\frac{1}{n}\right) z_n$$

- Now our mean depends only on the last value plus the current observation
- This procedure will work fine as long as the variable we are estimating is, in fact, a constant

Recursive Filtering



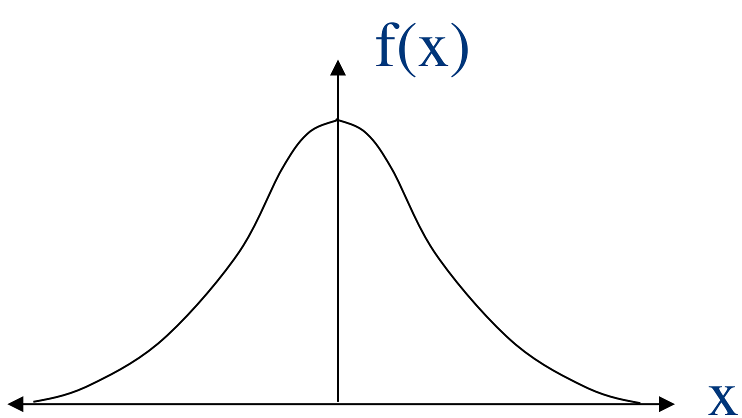
- What if we didn't know much about the underlying signal?
- Here we can see that the signal of interest appears to have significantly lower frequency than the noise
- A low pass filter will reduce the noise

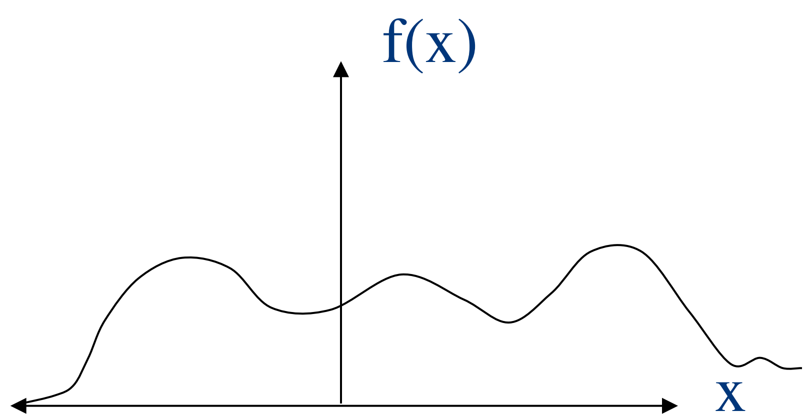
$$\begin{aligned}\hat{x}_k &= \hat{x}_{k-1} + \alpha\left(z_k - \hat{x}_{k-1}\right) \\ &= (1-\alpha)\hat{x}_{k-1} + \alpha z_k\end{aligned}$$

- Selecting the value of α will depend on the relative bandwidth of the signals

Probability Distributions

- A random variable is a variable which can take on some value
- We can describe a random variable using a *probability density function* (pdf) f(x)
- This describes the probability that the random variable will take on a particular value

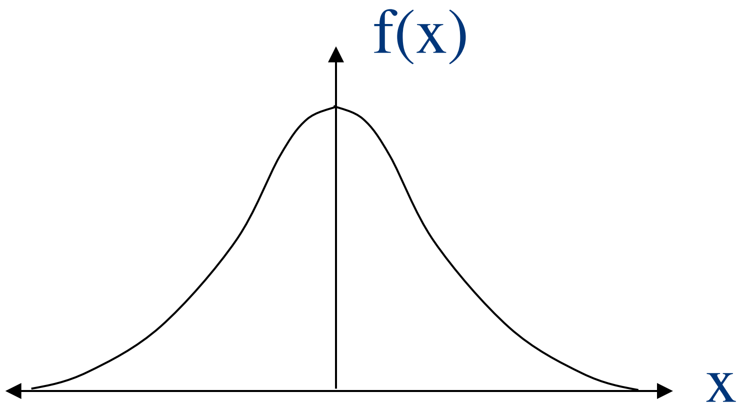




Gaussian Distribution

- The Gaussian distribution occurs commonly in many applications

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



- By the Central Limit Theorem, the sum of a number of independent variables has a Gaussian distribution regardless of their original distribution
- Furthermore, a Gaussian can be characterized by its first and second moments (mean and variance)

Probability Distributions

- PDFs are characterized by the following properties
  - f(x) is positive for all values of x

    $$f(x) \geq 0 \quad \forall x \in \Re$$

  - The area under the curve describes the probability that x falls within a particular range

    $$p(a \leq x \leq b) = \int_a^b f(x)dx$$

  - The total probability mass assigned to the set of X is 1

    $$\int_{-\infty}^{\infty} f(x)dx = 1 \qquad \text{or} \qquad \sum_{x \in \Re} f(x) = 1$$

Expected Value

- The expected value is the weighted average where the pdf provides the weighting function

$$\bar{x} = E[x] \triangleq \int_{-\infty}^{\infty} x f(x)\, dx$$

- In general this is *not* the same as the most likely value
- We may also be interested in the variance of the variable

$$Var(x) \triangleq E[(x-\bar{x})(x-\bar{x})^T]$$
$$= \int_{-\infty}^{\infty} (x-\bar{x})(x-\bar{x})^T f(x)\, dx$$

Joint Probability Distributions

- The joint probability describes the likelihood of a pair of variables taking on particular values
  - f(x, y) is positive for all values of x

$$f(x,y) \geq 0 \quad \forall x \in \Re,\ y \in \Re$$

  - The total probability mass assigned to the set of X and Y is 1

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,dx\,dy = 1$$

  - For a discrete pdf we find

$$\sum_{y\in\Re}\sum_{x\in\Re} f(x,y) = 1$$

Covariance

- The covariance describes the interdependence between two variables

$$
\begin{aligned}
Covar(x, y) &\triangleq E[(x-\bar{x})(y-\bar{y})^T] \\
&= \int_{-\infty}^{\infty} (x-\bar{x})(y-\bar{y})^T f(x, y)\, dx
\end{aligned}
$$

Marginal Probability

- The marginal probability describes the likelihood of one variable taking on a particular value over all values of the second

$$f(x) = \int_{-\infty}^{\infty} f(x, y)\,dy$$

Conditional Probability

- The conditional probability describes the likelihood of one variable taking on a particular value if the other is fixed

$$f(x \mid y) \triangleq \frac{f(x, y)}{f(y)}$$

- Two values are said to be conditionally independent if the value of one is independent of the other

$$f(x \mid y) = f(x)$$

- In this case, the joint probability is simply the product of the two independent probabilities

Theorem of Total Probability

- From the definition of conditional probability and the axioms of probability measures we can derive the total probability as

$$f(x) = \int f(x \mid y) f(y) dy$$

Bayes Rule

- By combining the conditional probability rules, Bayes rule provides us with a method for estimating a quantity of interest

$$f(x \mid y) \triangleq \frac{f(x,y)}{f(y)}$$

$$f(y \mid x) \triangleq \frac{f(x,y)}{f(x)}$$

$$\therefore f(y \mid x) = \frac{f(x \mid y)\, f(y)}{f(x)}$$

Bayes Estimation

- What if we know something about the statistics of the process we are modelling?
- We can employ Bayes rule to help us in the estimation process

$$f(x|Z^n) = \frac{f(Z^n \mid x) f(x)}{f(Z^n)}$$

$$= \frac{f(z_1, z_2 \cdots z_n \mid x) f(x)}{f(z_1, z_2 \cdots z_n)}$$

- If we assume that the observations are independent given the true state of the system we find

$$f(x|Z^n) = \frac{f(x)}{f(Z^n)} \prod_{i=1}^{n} f(z_i \mid x)$$

Recursive Bayes Estimation

- We can rearrange the previous equation to put it into a recursive form

$$f(x|Z^k) = \frac{f(z_k | x) f(x|Z^{k-1})}{f(z_k | Z^k)}$$

Estimation

- In essence, estimation methods are designed to provide us with the best estimate of the our states of interest x_k given the information available to us

$$P(x_k \mid Z^k, U^{k-1}, x_0)$$

where

  - x_k is the state at time k
  - Z^k is a sequence of observations up to time k
  - U^{k-1} is a sequence of actions up to time k-1
  - x_0 is the initial state
- How *best* is defined depends on the situation. We also need to make decisions about how to model any potential errors in the sensors

Bayesian Estimation

z = observation
u = action
x = state

$$\boxed{Bel(x_t)} = P(x_t \mid U^{t-1}, Z^t)$$

Bayes
$$= \eta \; P(z_t \mid x_t, U^{t-1}, Z^{t-1}) \; P(x_t \mid U^{t-1}, Z^{t-1})$$

Markov
$$= \eta \; P(z_t \mid x_t) \; P(x_t \mid U^{t-1}, Z^{t-1})$$

Total prob.
$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid U^{t-1}, Z^{t-1}, x_{t-1}) \; P(x_{t-1} \mid U^{t-1}, Z^{t-1}) \; dx_{t-1}$$

Markov
$$= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; P(x_{t-1} \mid U^{t-1}, Z^{t-1}) \; dx_{t-1}$$

$$= \boxed{\eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \, Bel(x_{t-1}) \, dx_{t-1}}$$

Bayesian Estimation

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \; \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

- We require the following:
  - State Representation x_t
  - A belief Model $Bel(x_t)$
  - Process Model $P(x_t \mid u_{t-1}, x_{t-1})$
  - Observation Model $P(z_t \mid x_t)$

Bayesian Estimation

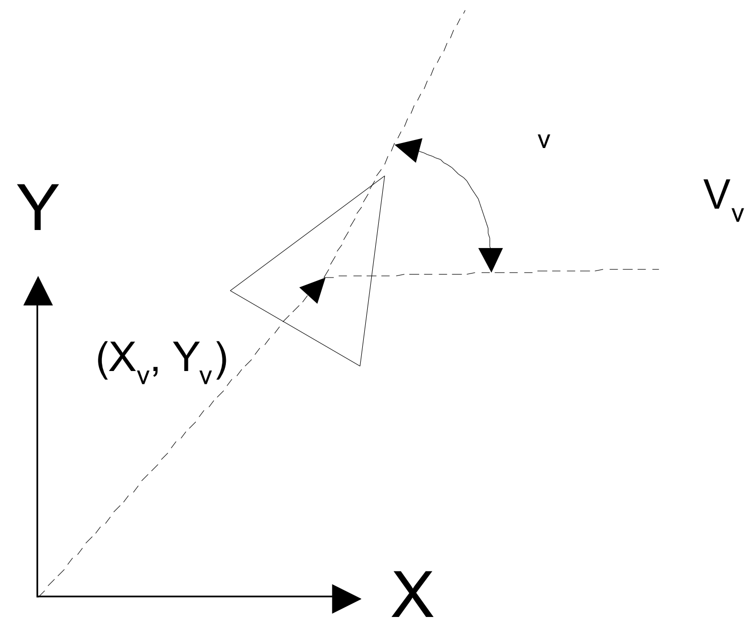
$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_{t-1}, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

- Kalman filters (linear, extended, unscented)
- Particle filters (Rao-Blackwellized)
- Hidden Markov models
- Dynamic Bayesian networks
- Estimator in Partially Observable Markov Decision Processes (POMDPs)

State Space Representations

- For a mobile vehicle, we are usually interested in describing the state of the vehicle by its pose

$$\mathbf{x} = \begin{bmatrix} x_v \\ y_v \\ \psi_v \end{bmatrix}$$



Process Models

- Process models describe the evolution of the state by a first order non-linear differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{v}(t)$$

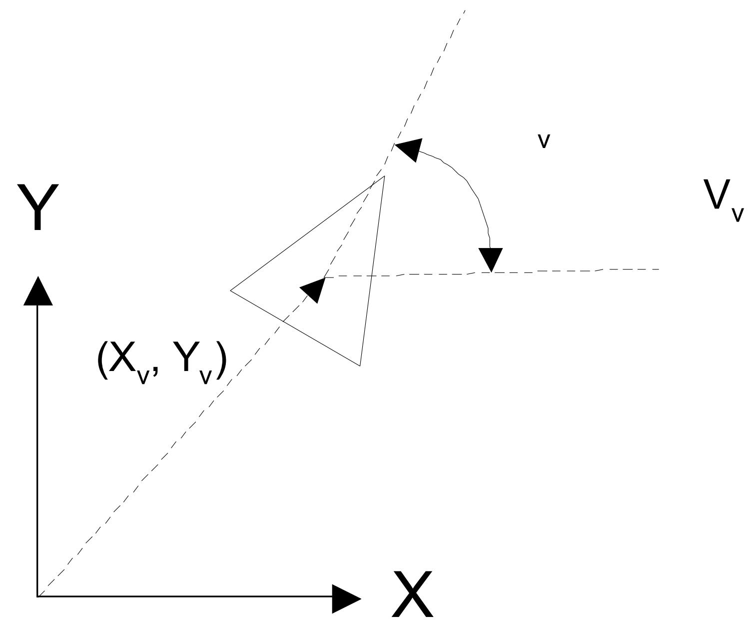
Vehicle Model

- If we can measure the vehicle velocity and sense heading changes we can write a differential equation describing the evolution of the vehicle pose

$$\dot{x}_v = V_v \cos(\psi_v) + v_x$$

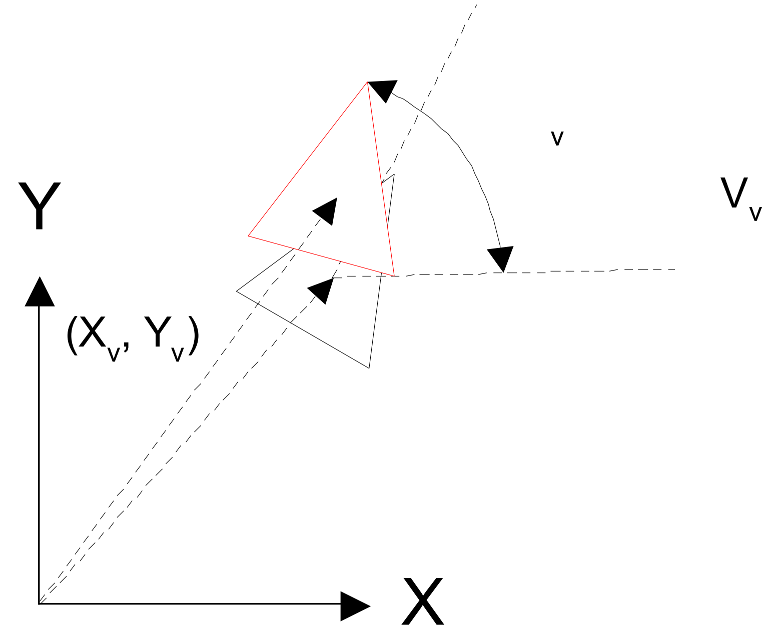
$$\dot{y}_v = V_v \sin(\psi_v) + v_y$$

$$\dot{\psi}_v = (\dot{\psi}_{turnrate}) + v_\psi$$



Vehicle Model

- To implement this on a digital controller, we discretize the process equations

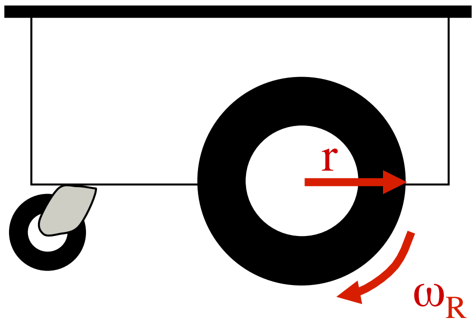


$$x_v(k) = x_v(k-1) + \Delta t \cdot V_v(k-1)\cos(\psi_v(k-1)) + v_x$$

$$y_v(k) = y_v(k-1) + \Delta t \cdot V_v(k-1)\sin(\psi_v(k-1)) + v_y$$

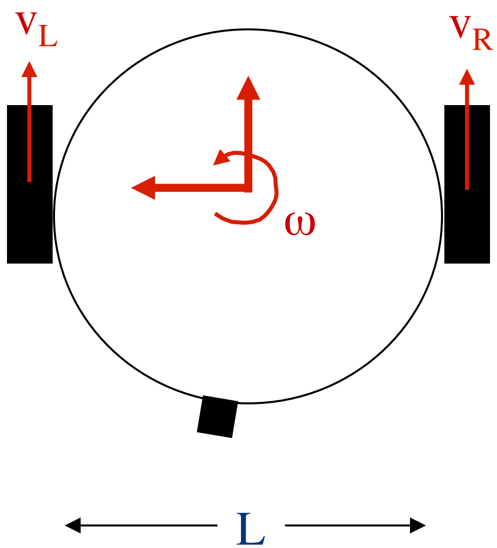
$$\psi_v(k) = \psi_v(k-1) + \Delta t \cdot \dot{\psi}_{turnrate}(k-1) + v_\psi$$

Vehicle Model – Differential Drive



$$v_L = r_L \times \omega_L$$

$$v_R = r_R \times \omega_R$$

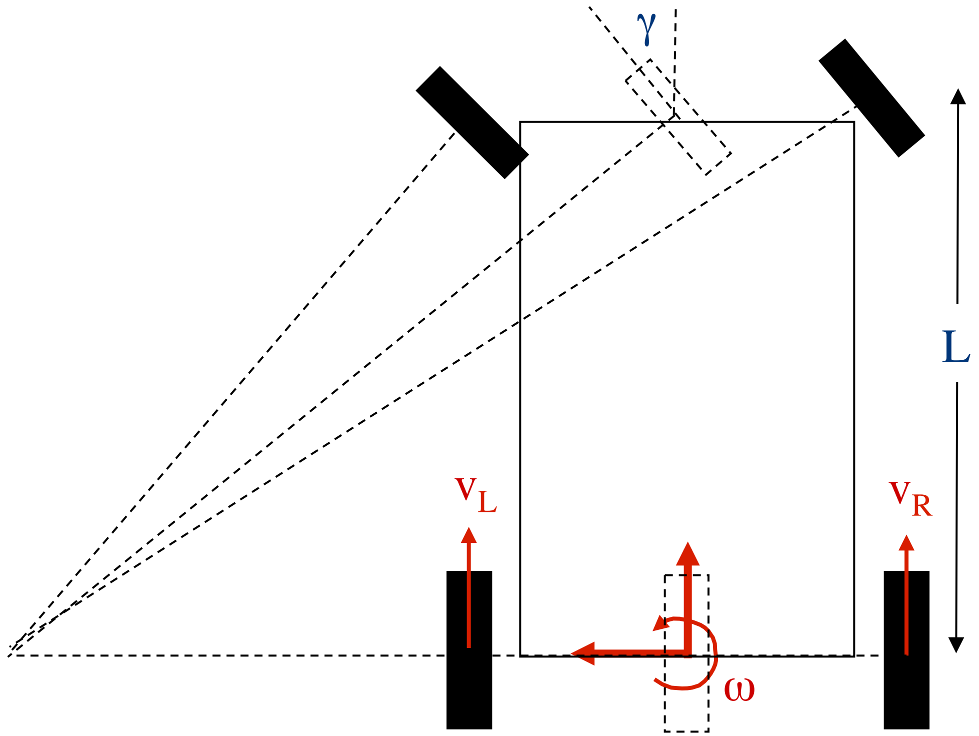


$$v = \frac{v_L + v_R}{2}$$

$$\omega = \frac{v_L - v_R}{L}$$

- A vehicle like our pioneers relies on differential drive (i.e. two powered wheels) velocity and turn rate is achieved by turning the two wheels
- If one wheel turns, the body centre will move at half the instantaneous velocity. The body will rotate about the stationary wheel

Vehicle Model – Differential Drive



$$v_L = r_L \times \omega_L$$

$$v_R = r_R \times \omega_R$$

$$v = \frac{v_L + v_R}{2}$$

$$\omega = \frac{v \tan(\gamma)}{L}$$

- More complex vehicles, such as a car, are often modelled using the tricycle model
- Velocity and turn rate is measured about the centre of the rear axis
- The angle, γ, of the front steering wheel, determines the turn rate of the vehicle

Observation Models

- An observation of the state is usually described by an observation model

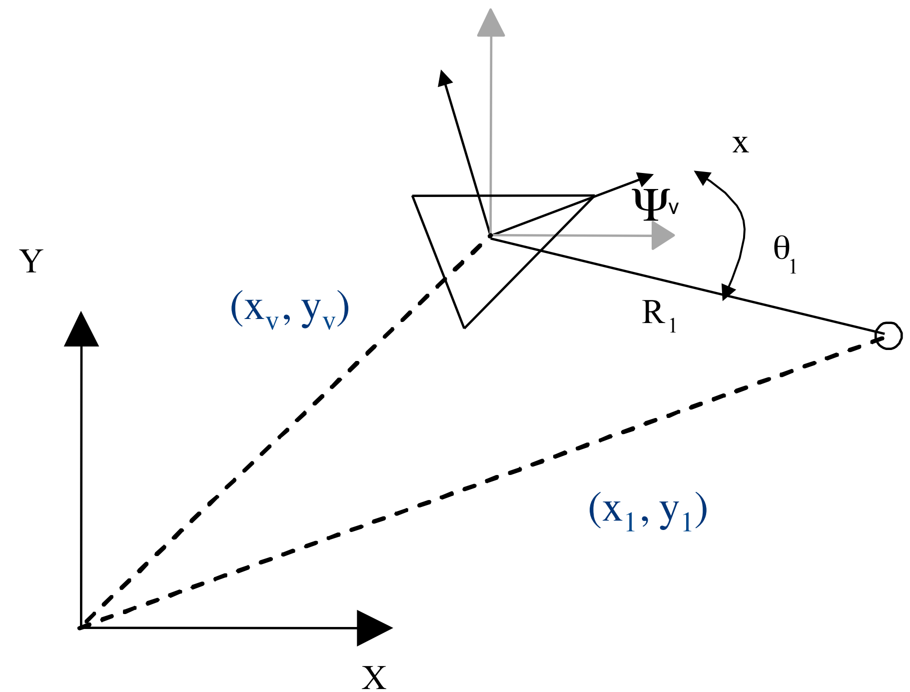
$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t), t) + \mathbf{w}(t)$$

- Observations may not observe all the states but we may be able to infer something about these states

Observation Models

- What if I take an observation to a feature in the world?
- Clearly, observations of the relative position between myself and this beacons would tell me something about my own state

$$\mathbf{z} = \begin{bmatrix} R_1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \sqrt{(x_v - x_1)^2 + (y_v - y_1)^2} \\ \tan^{-1} \dfrac{y_v - y_1}{x_v - x_1} - \psi_v \end{bmatrix} + \begin{bmatrix} w_R \\ w_\theta \end{bmatrix}$$



Estimation

- At this point we have a model of how we believe the state will evolve over time
- We also have a model of the observations we are likely to take of the environment
- How do we combine this information to yield the best estimate of the state of the vehicle in the world?
- Estimation is the process of generating an estimate of the state of our system

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_v \\ \hat{\psi}_v \end{bmatrix}$$

Estimation

- Least squares estimation produces an estimate that minimizes the sum of the squared error between the observation and the model

$$\hat{\mathbf{x}}^{LS}(k) = \arg\min_{x \in X} \sum_{j=1}^{k} [z(j) - h(j,x)]^T [z(j) - h(j,x)]$$

- Minimum Mean Squared Error estimation produces an estimate that minimizes the mean squared error

$$\hat{\mathbf{x}}^{MMSE}(k) = \arg\min_{x \in X} E[(\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x}) \mid Z^k]$$

Particle Filter

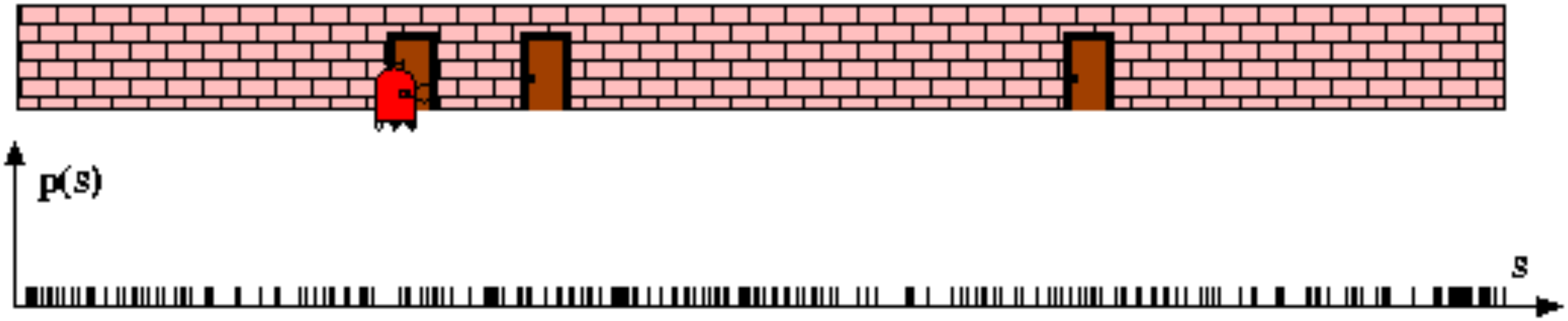
- The Particle filter represents the uncertainty using a discrete set of samples
- Represent belief by random samples
- Estimation of non-Gaussian, nonlinear processes

- Monte Carlo filter, Survival of the fittest, Condensation, Bootstrap filter, Particle filter

- Filtering: [Rubin, 88], [Gordon et al., 93], [Kitagawa 96]
- Computer vision: [Isard and Blake 96, 98]
- Dynamic Bayesian Networks: [Kanazawa et al., 95]d

Particle Filters

$$Bel(x_0)$$

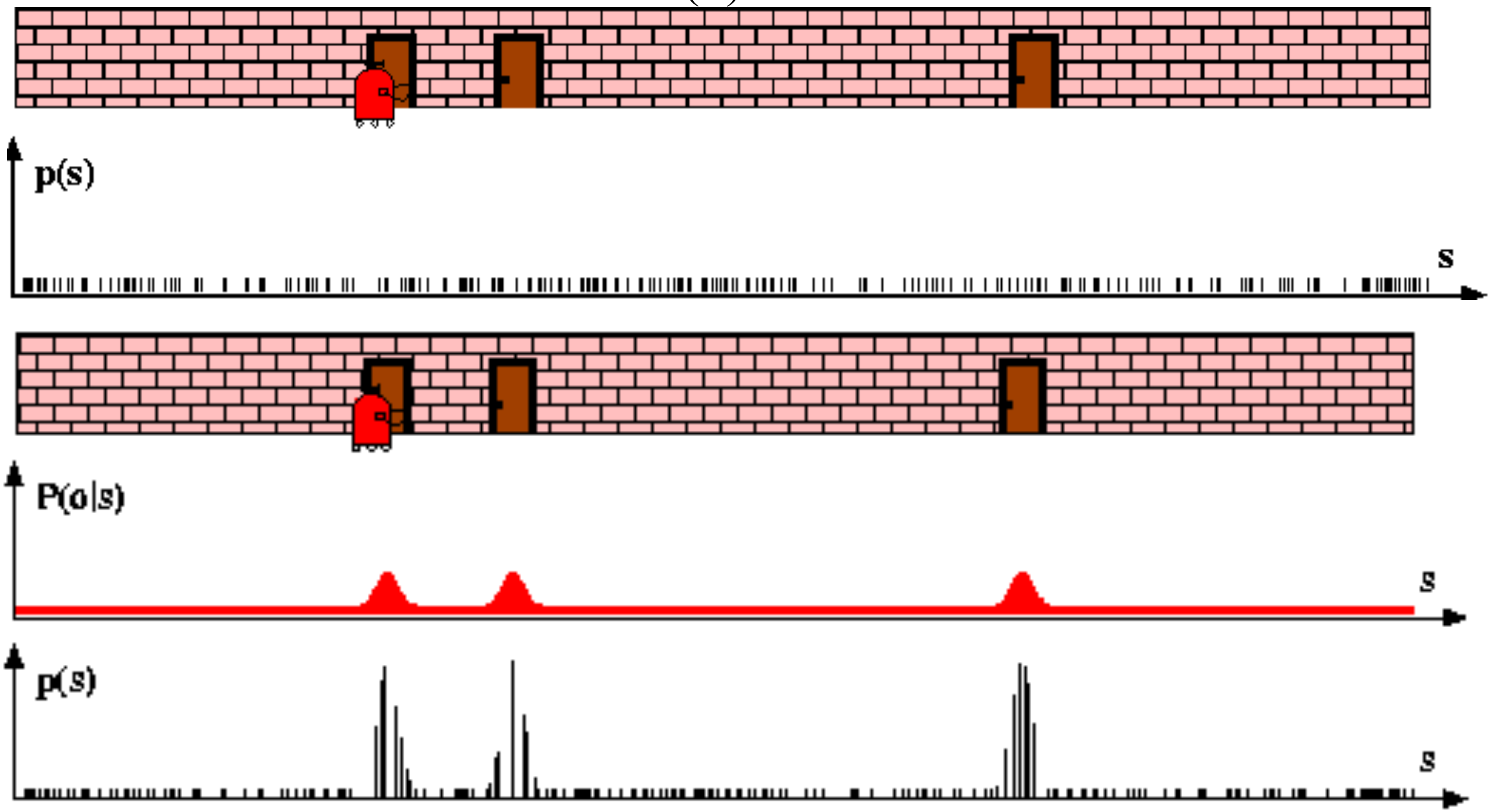


The following slides are courtesy of Dieter Fox

Sensor Information: Importance Sampling

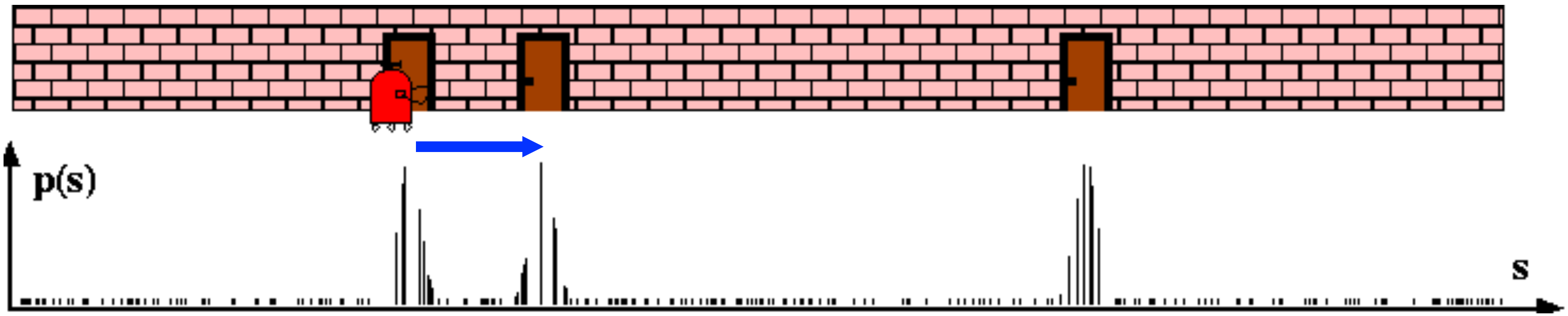
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

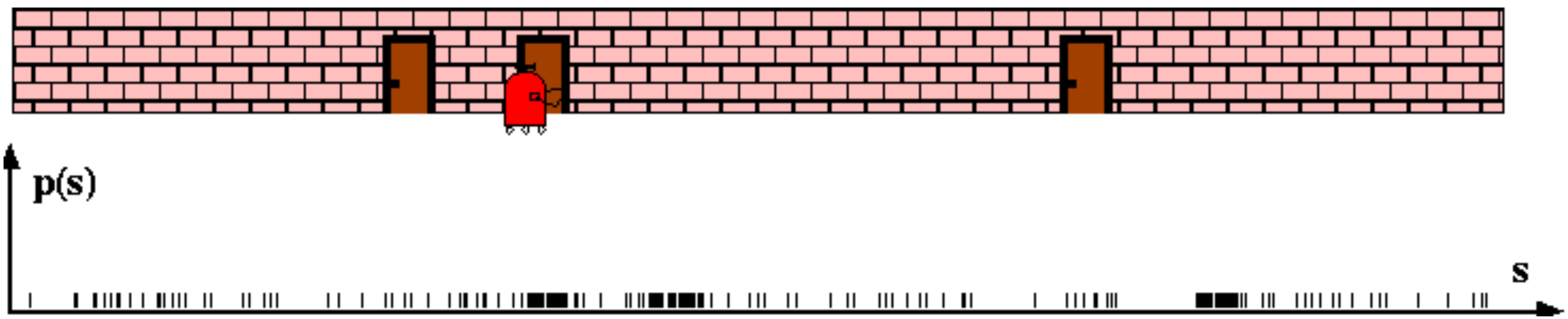
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x')\, Bel(x') \;\, \mathrm{d}x'$$

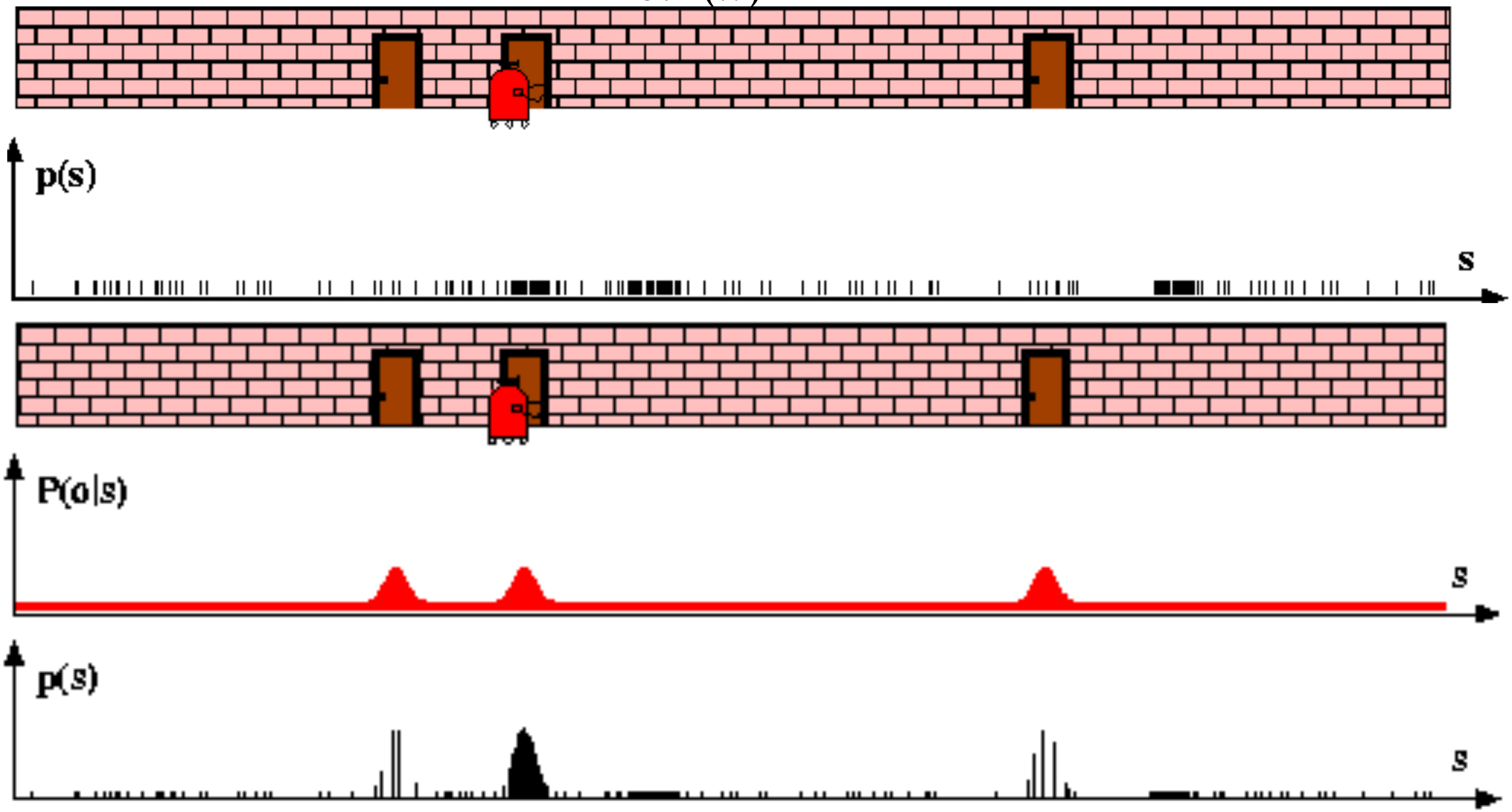




Sensor Information: Importance Sampling

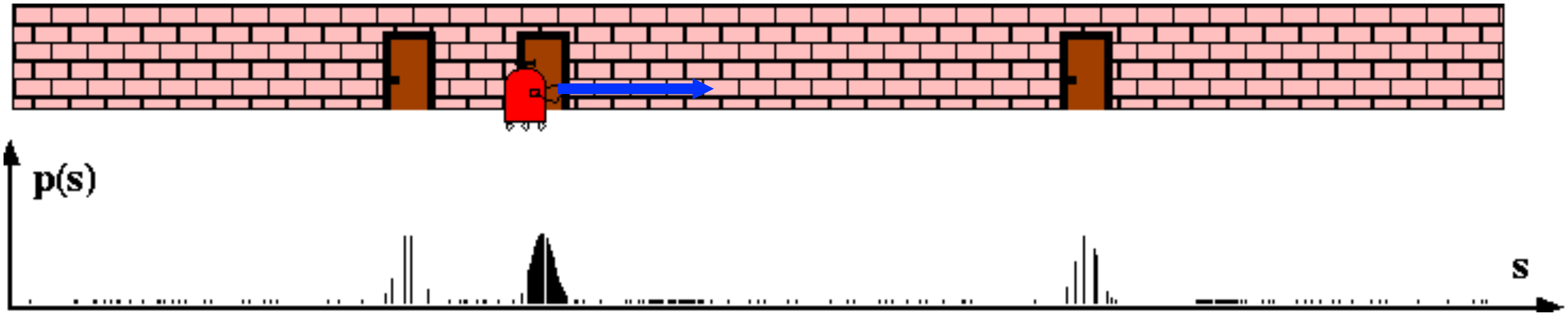
$$Bel(x) \leftarrow \alpha\, p(z \mid x)\, Bel^{-}(x)$$

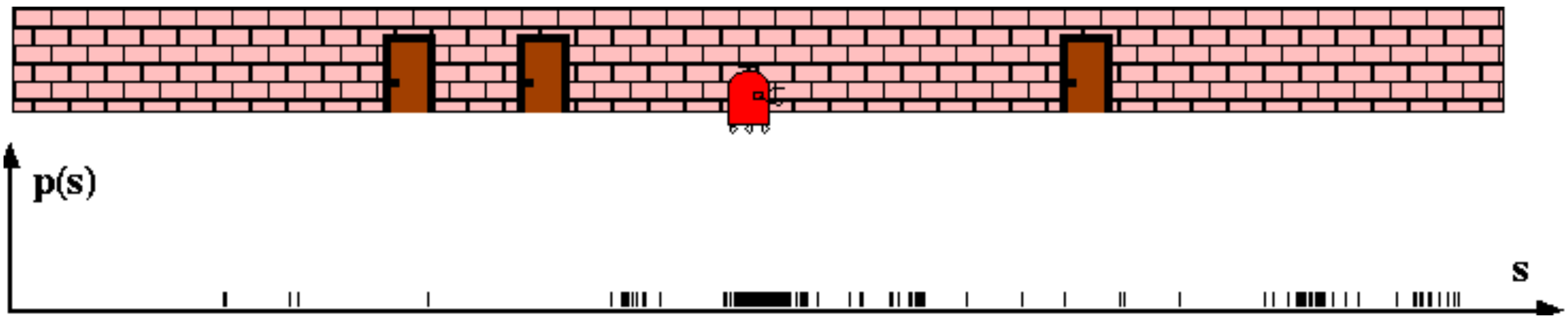
$$w \leftarrow \frac{\alpha\, p(z \mid x)\, Bel^{-}(x)}{Bel^{-}(x)} = \alpha\, p(z \mid x)$$



Robot Motion

$$Bel^{-}(x) \quad \leftarrow \quad \int p(x \mid u, x')\, Bel(x') \;\, \mathrm{d}\, x'$$





Particle Filter Algorithm

1. Algorithm **particle_filter**(S_{t-1}, u_{t-1} z_t):
2. $S_t = \emptyset, \quad \eta = 0$
3. **For** $i = 1 \ldots n$ ***Generate new samples***
4. Sample index *j(i)* from the discrete distribution given by w_{t-1}
5. Sample x_t^i from $p(x_t \mid x_{t-1}, u_{t-1})$ using $x_{t-1}^{j(i)}$ and u_{t-1}
6. $w_t^i = p(z_t \mid x_t^i)$ ***Compute importance weight***
7. $\eta = \eta + w_t^i$ ***Update normalization factor***
8. $S_t = S_t \cup \{< x_t^i, w_t^i >\}$ ***Insert***
9. **For** $i = 1 \ldots n$
10. $w_t^i = w_t^i / \eta$ ***Normalize weights***

Particle Filter Algorithm

$$Bel(x_t) = \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})\, dx_{t-1}$$

draw x^i_{t-1} from $Bel(x_{t-1})$

draw x^i_t from $p(x_t \mid x^i_{t-1}, u_{t-1})$

Importance factor for x^i_t:

$$w^i_t = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})}{p(x_t \mid x_{t-1}, u_{t-1})\, Bel(x_{t-1})} \propto p(z_t \mid x_t)$$

The Kalman Filter

- The Kalman Filter is a minimum mean-squared error estimator
- It makes the assumption that noise in the process and observation models are white, zero-mean Gaussian
- With this assumption, we can recursively estimate the mean and covariance of a Gaussian describing the most likely state of the system

Prediction

- During the prediction step, we take the current estimate of the state of the system and apply the process model

$$\hat{\mathbf{x}}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1}$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^T + \mathbf{Q}$$

  - where F is the model
    - u is the control input
    - Q is the model of the uncertainty in the process
- This results in an update to the mean and variance of our estimate of the state of the world

Observation

- At some time an observation is received and we will compare the actual observation against what we would *expect* to observe if our current estimate were correct

$$\hat{\mathbf{z}}_k = \mathbf{H}\hat{\mathbf{x}}_k^-$$

$$\nu = \mathbf{z}_k - \hat{\mathbf{z}}_k$$

$$\mathbf{S} = \mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}$$

where h is the observation model

ν is the innovation

S is the innovation Covariance

R is the observation Covariance

Update

- We then compute an optimal weighting factor W that is used to update our estimate of the mean and covariance based on the information at hand

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

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{W}\nu$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{W}\mathbf{S}\mathbf{W}^T$$

where W is the Kalman Gain

Deriving the Posterior Estimate Covariance

- By definition

$$\mathbf{P}_k^+ = \mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)$$

- Substituting values

$$\mathbf{P}_k^+ = \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_k^- + \mathbf{W}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-))$$

$$= \mathrm{cov}(\mathbf{x}_k - (\hat{\mathbf{x}}_k^- + \mathbf{W}_k(\mathbf{H}\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-))$$

- Collecting terms

$$\mathbf{P}_k^+ = \mathrm{cov}((I - \mathbf{W}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)) + \mathrm{cov}(\mathbf{W}_k\mathbf{v}_k)$$

$$= (I - \mathbf{W}_k\mathbf{H})\,\mathrm{cov}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(I - \mathbf{W}_k\mathbf{H})^T + \mathbf{W}_k\,\mathrm{cov}(\mathbf{v}_k)\mathbf{W}_k^T$$

$$= (I - \mathbf{W}_k\mathbf{H})\mathbf{P}_k^-(I - \mathbf{W}_k\mathbf{H})^T + \mathbf{W}_k\mathbf{R}_k\mathbf{W}_k^T$$

$$= \mathbf{P}_k^- - \mathbf{W}_k(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}_k)\mathbf{W}_k^T$$

Example

- Imagine that we wish to estimate the one dimensional position and speed of a particle

$$\mathbf{x}_k = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

- Assuming the particle moves under constant acceleration, the process model can be written as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k$$

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \; and \; \mathbf{G} = \begin{bmatrix} \dfrac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}$$

Example

- If we receive some noisy observation of the position of the particle, we have

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_{k-1} + \mathbf{v}_{\mathbf{k}}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Estimation for Navigation

- Recursive three stage update procedure

① Prediction

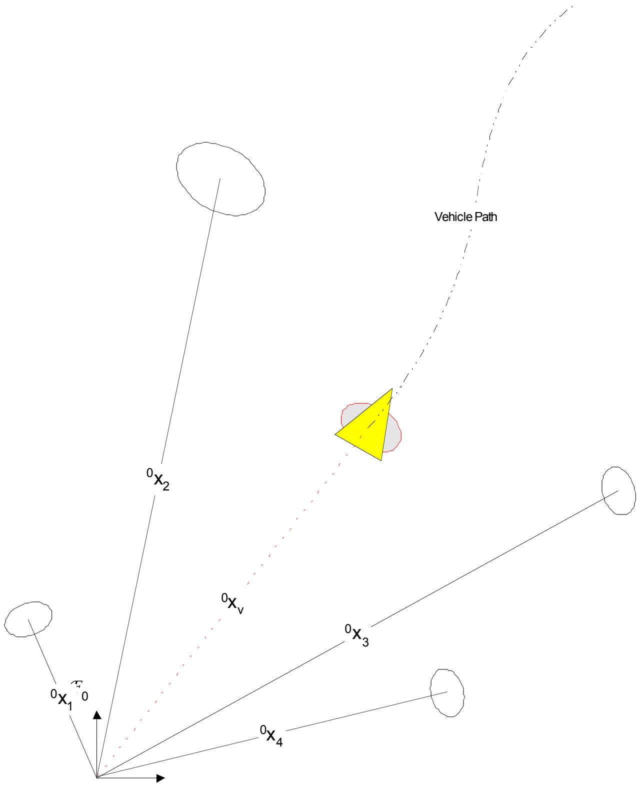
Use vehicle model to predict vehicle position

② Observation

Take feature observation(s)

③ Update

Validated observations used to generate optimal estimate



$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_v \\ \hat{y}_v \\ \hat{\psi}_v \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} P_{xx} & P_{xy} & P_{x\psi} \\ P_{xy}^T & P_{yy} & P_{y\psi} \\ P_{x\psi}^T & P_{y\psi}^T & P_{\psi\psi} \end{bmatrix}$$

EKF Prediction

- During the prediction step, we take the current estimate of the state of the system and apply the process model

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1})$$

$$\mathbf{P}_k^- = \nabla\mathbf{f}\mathbf{P}_{k-1}^+\nabla\mathbf{f}^T + \mathbf{Q}$$

  - where f is the model
    - u is the control input
    - Q is the model of the uncertainty in the process
- This results in an update to the mean and variance of our estimate of the state of the world

EKF Observation

- At some time an observation is received and we will compare the actual observation against what we would *expect* to observe if our current estimate were correct

$$\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}_k^-)$$

$$\nu = \mathbf{z}_k - \hat{\mathbf{z}}_k$$

$$\mathbf{S} = \nabla\mathbf{h}_k \mathbf{P}_k^- \nabla\mathbf{h}^T{}_k + \mathbf{R}$$

where h is the observation model

ν is the innovation

S is the innovation Covariance

R is the observation Covariance

EKF Update

- We then compute an optimal weighting factor W that is used to update our estimate of the mean and covariance based on the information at hand

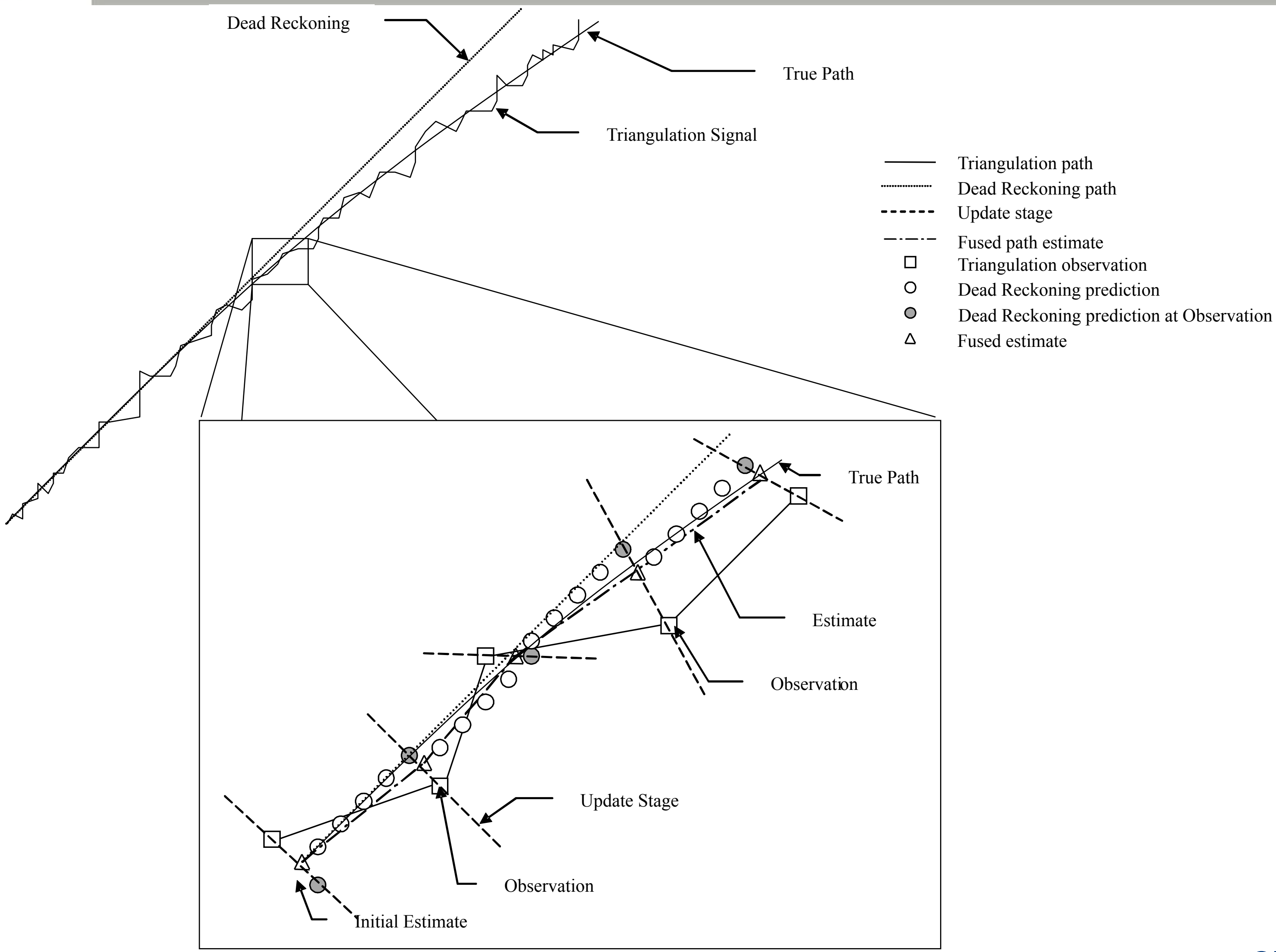
$$\mathbf{W} = \mathbf{P}_k^- \nabla \mathbf{h}_k^T \mathbf{S}^{-1}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{W}\nu$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{WSW}^T$$

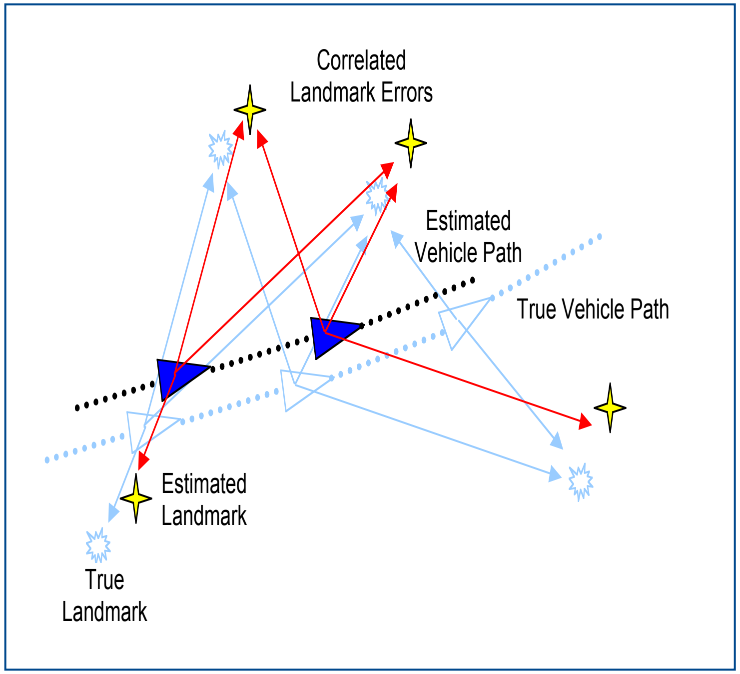
where W is the Kalman Gain

Data Fusion



Slide 53

The SLAM Problem



- Simultaneous Localisation and Map Building (SLAM)
- Start at an unknown location with no a priori knowledge of landmark locations
- From relative observations of landmarks, compute estimate of vehicle location and estimate of landmark locations
- While continuing in motion, build complete map of landmarks and use these to provide continuous estimates of vehicle location

The Estimation Process

- Recursive three stage update procedure

① Prediction

Use vehicle model to predict vehicle position

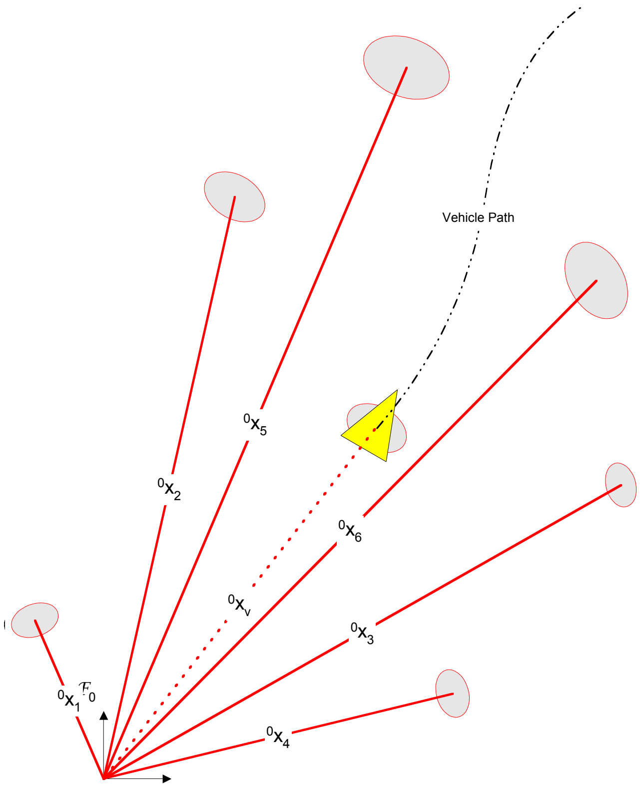
② Observation

Take feature observation(s)

③ Update

Validated observations used to generate optimal estimate

Initialise new target



Conclusions

- Data fusion is an important mechanism for combining noisy or uncertain data
- Many methods for data fusion rely on Bayesian techniques for consistently fusing data
- Have a look at

http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf

which contains a good description of the Kalman Filter and its applications

References

- Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- R.G. Brown and P.Y.C. Hwang. *Introduction to Applied Kalman Filtering*. John Wiley,
- 1992.
- D.E. Catlin. *Estimation, Control and the Discrete Kalman Filter*. Springer Verlag, 1984.
- H.F. Durrant-Whyte. An autonomous guided vehicle for cargo handling applications. *Int. J. Robotics Research*, 15(5):407–440, 1996.
- M. Gelb. *Applied Optimal Estimation*. MIT press, 1974.
- M.S. Grewal and A.P. Andrews. *Kalman Filtering Theory and Practise*. Prentice Hall, 1993.
- P.S. Maybeck. *Stochastic Models, Estimaton and Control, Vol. I*. Academic Press, 1979.
- Sebastian Thrun, Dieter Fox and Wolfram Burgard, *Probabilistic Robotics*, The MIT Press, 2005