Simultaneous Localisation and Mapping: Probabilistic Formulation

Tim Bailey

Australian Centre for Field Robotics

Overview

- The SLAM problem
- History
- Models
- Bayesian estimation

- Last part has lots of maths [unfortunately]

What is SLAM?

- **SLAM asks the following question:**
  - Is it possible for an autonomous vehicle to start in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute vehicle location?
- **SLAM allows robots to operate in an environment without *a priori* knowledge of a map and without access to independent position information.**
- **SLAM is central to a range of indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles. For example,**
  - Explore and return to starting point, learn trained paths to different goal locations, traverse a region with complete coverage (eg, mine fields, lawns, reef monitoring), …

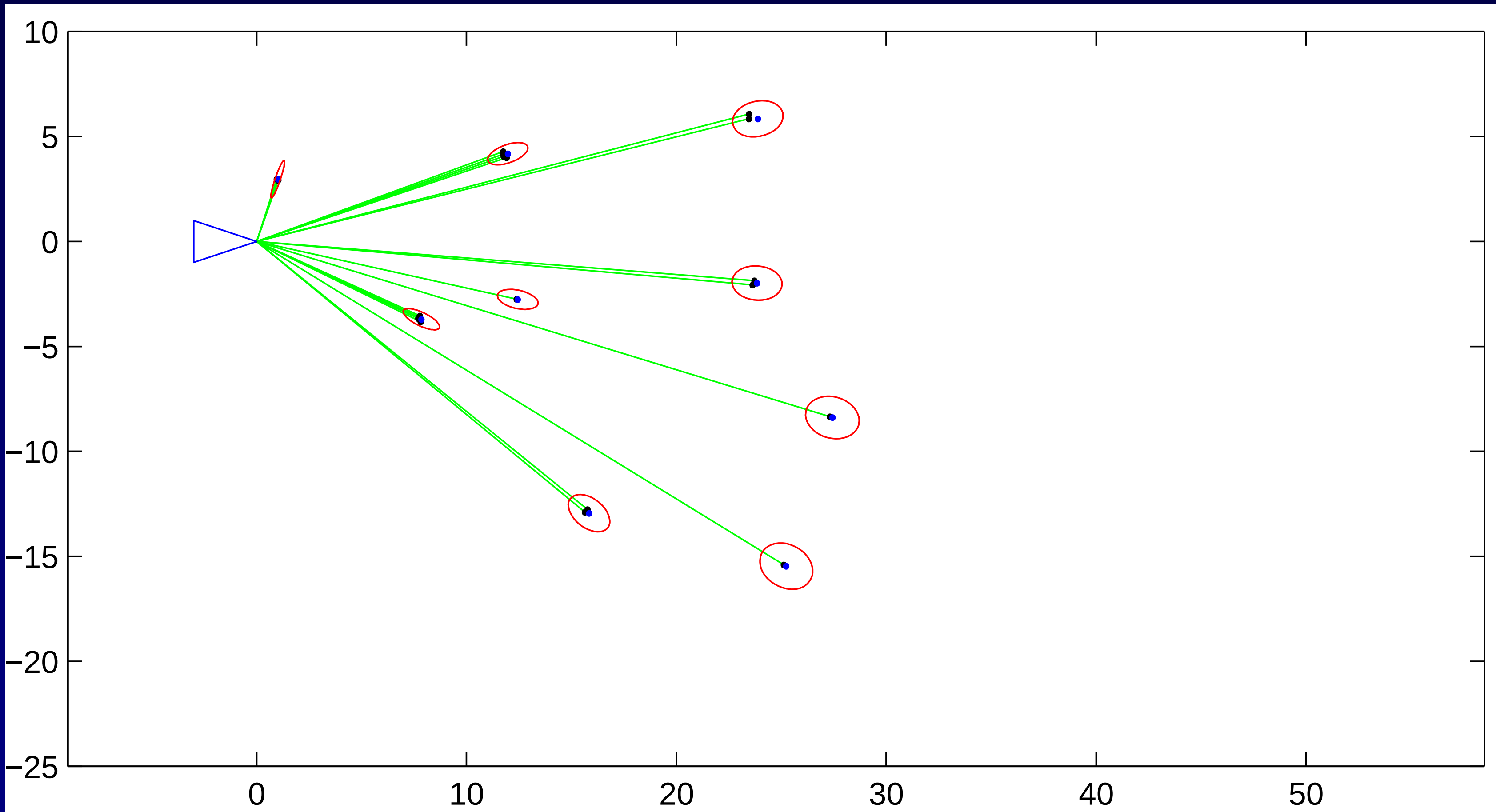
What is SLAM?

- **Localisation**
  – Determine pose given a priori map
- **Mapping**
  – Generate map when pose is accurately known from auxiliary source.
- **SLAM**
  – Define some arbitrary coordinate origin (usually the initial vehicle pose).
  – Generate a map from on-board sensors while, at the same time, computing pose from the map.
  – Errors in map and in pose estimate are dependent.

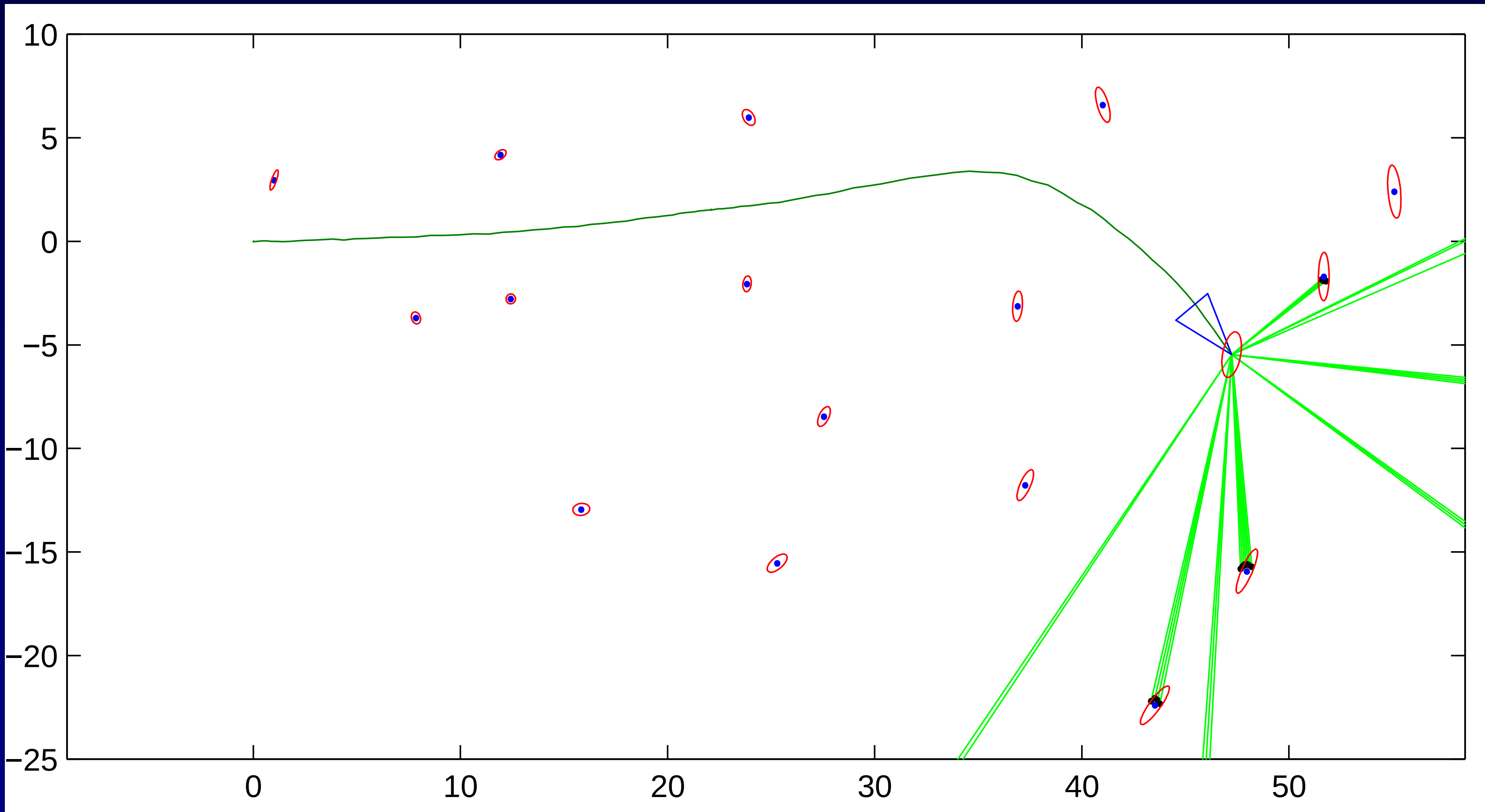
Basic SLAM Components

Example: SLAM in Victoria Park

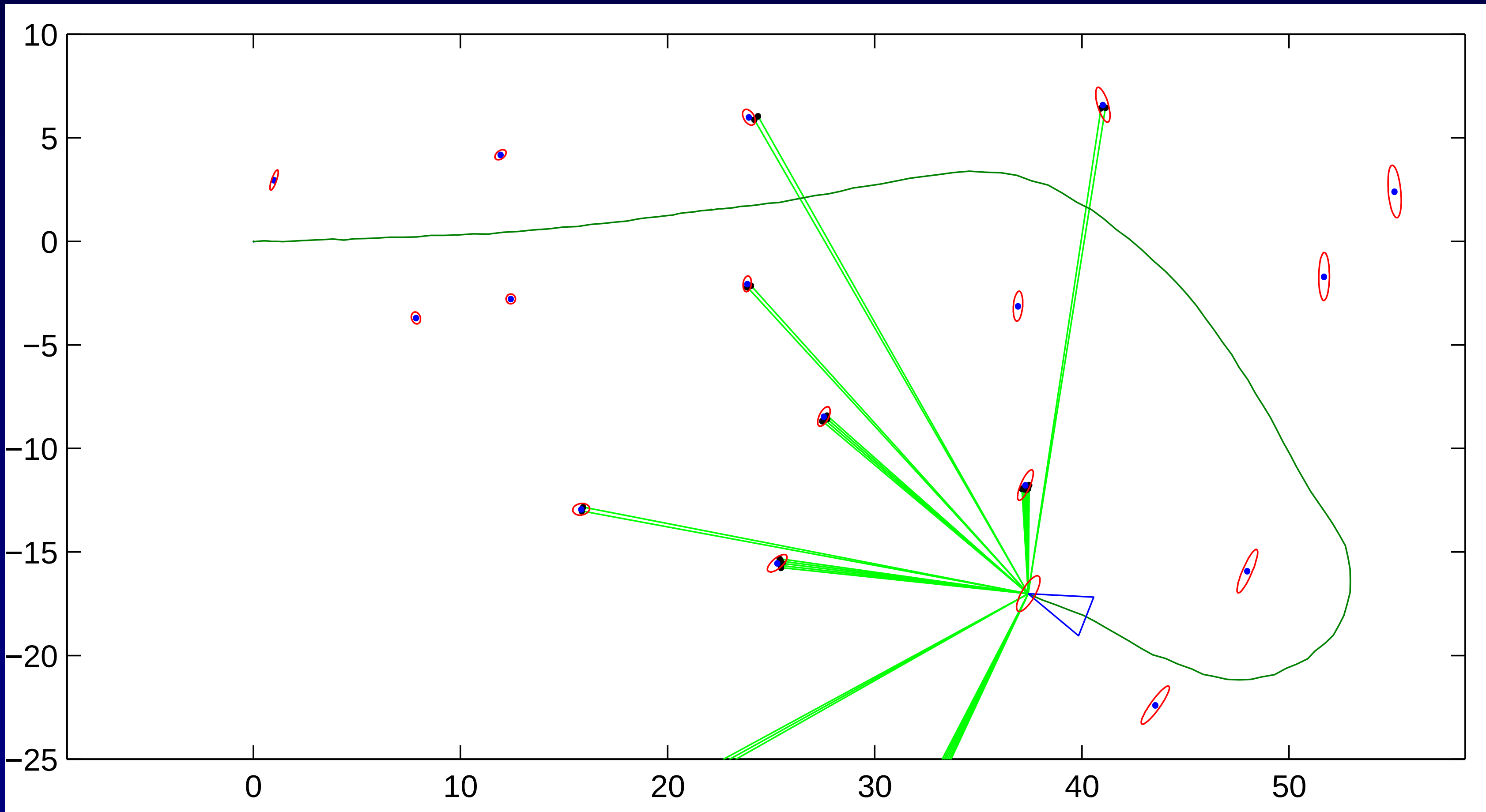
Basic SLAM Operation



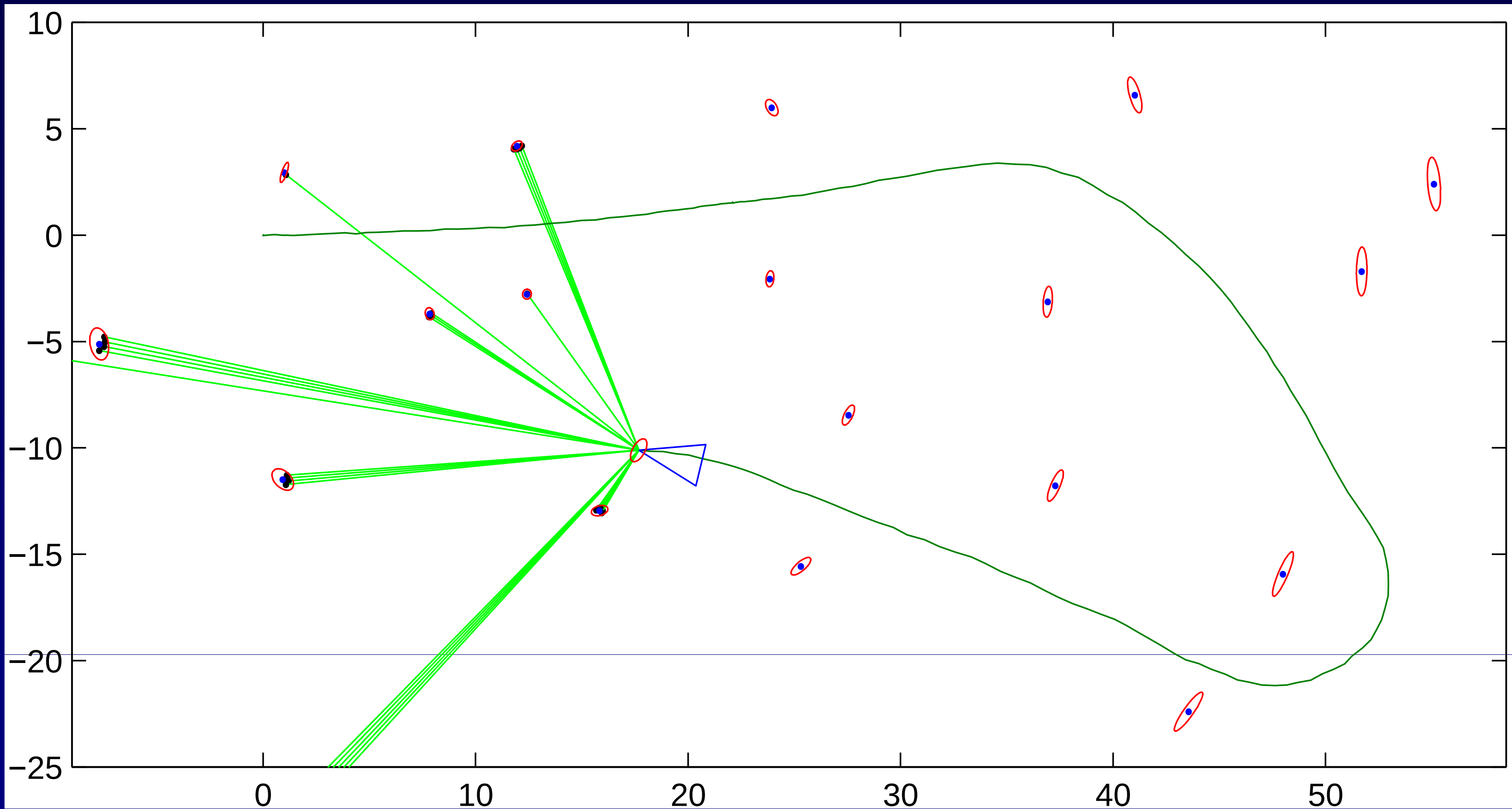
Basic SLAM Operation



Basic SLAM Operation



Basic SLAM Operation



A Little Bit of History

- **Addressing SLAM in a probabilistic setting began about 20 years ago at ICRA86 in San Francisco.**
  - Probabilistic methods were new to robotics and AI.
  - Peter Cheeseman, Jim Crowley, Raja Chatila, Olivier Faugeras and Hugh Durrant-Whyte were all looking at applying estimation-theoretic methods to mapping and localisation problems.
- **Landmark paper by Smith, Self and Cheeseman showed that as a mobile robot moves through an unknown environment taking relative observations of landmarks, the estimates of these landmarks are all necessarily correlated with each other because of the common error in estimated vehicle location.**
- **Important implication:**
  - A consistent full solution to the combined localisation and mapping problem would require a joint state composed of the vehicle pose and every landmark position, to be updated following each landmark observation.
  - An EKF estimator would need a huge state vector (of order the number of landmarks maintained in the map) with computation scaling as the square of the number of landmarks.
- **It was assumed at the time that the estimated map errors would not converge and would instead exhibit a random walk behaviour with unbounded error growth.**
- **Conceptual break-through: the combined mapping and localisation problem, once formulated as a single estimation problem, is convergent (Csorba 96, Dissa 01).**
  - Recognise that the correlations between landmarks, which previously people had tried to minimize, were actually the critical part of the problem and that, on the contrary, the more these correlations grew, the better the solution.

Alternative SLAM Solutions

- **In this talk, we focus on a particular SLAM solution**
  - Building a map of discrete landmarks
- **Landmark based SLAM is not the only solution, but**
  - We are convinced that *all* solutions should have a probabilistic basis to deal with uncertainty
- **Some alternatives:**
  - Trajectory-based (or view-based) SLAM
    - Probability over vehicle trajectory, so as to align all views
    - Implementations typically neglect "map" correlations – implicit in reusing view information
  - Topological SLAM
    - Accuracy requirements of metric map is relaxed
    - Emphasis is instead on reliable recognition of places
    - Primarily a data association problem

Models

- **Models are central to creating a representation of the world.**
- **Must have a mapping between sensed data (eg, laser, cameras, odometry) and the states of interest (eg, vehicle pose, stationary landmarks)**
- **Two essential model types:**
  - Vehicle motion
  - Sensing of external objects

States, Controls, Observations

Joint state with momentary pose

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Joint state with pose history

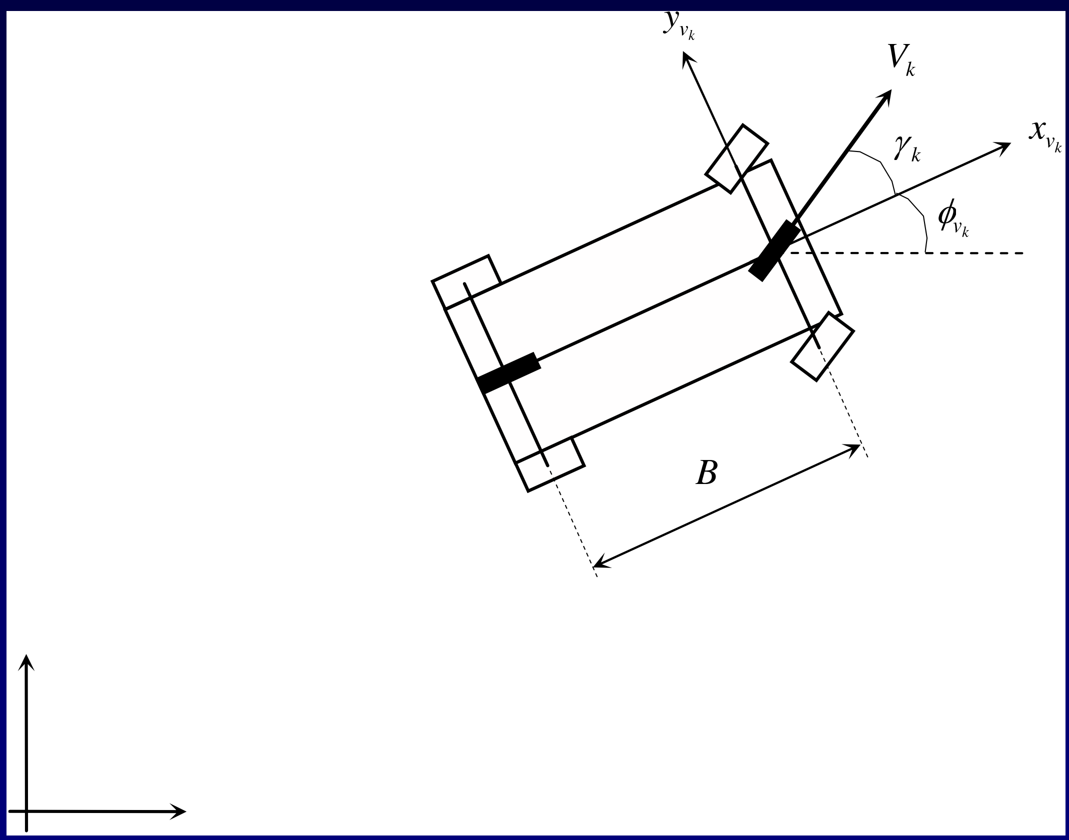
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Control inputs: $\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$

Observations: $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$

Vehicle Motion Model

- **Ackerman steered vehicles: Bicycle model**



- **Discrete time model:**

$$\mathbf{x}_{v_k} = \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}$$

SLAM Motion Model

$$\mathbf{x}_{v_k} = \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix}$$

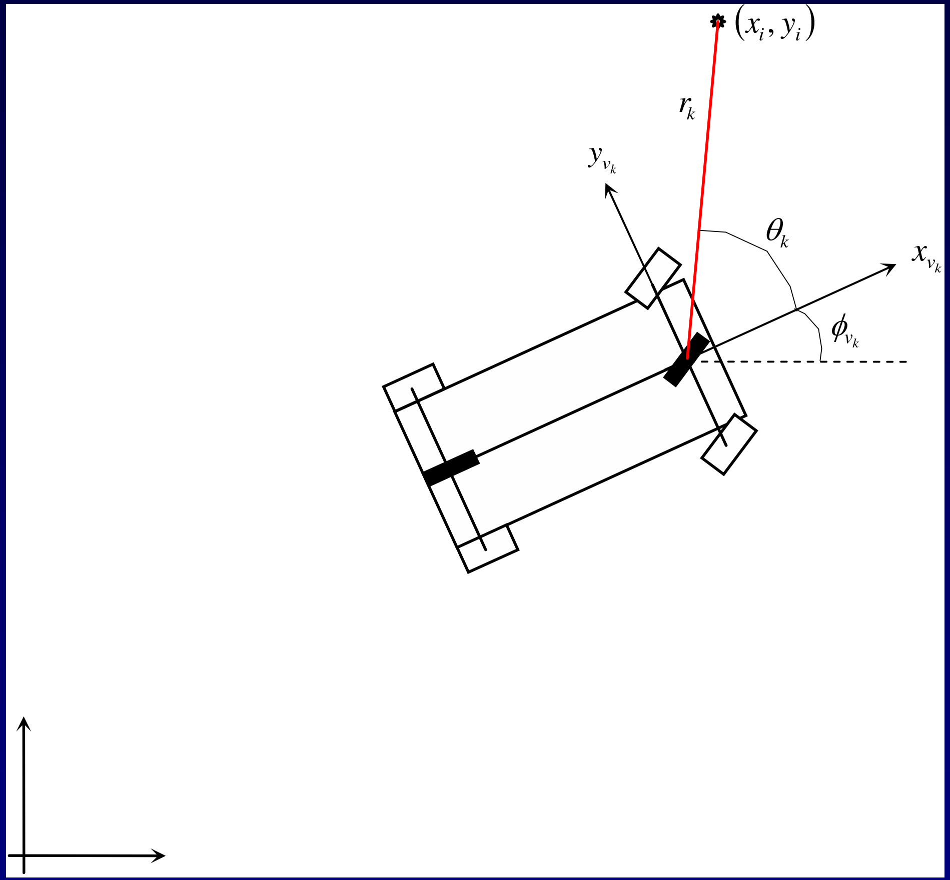
- Joint state: Landmarks are assumed stationary

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

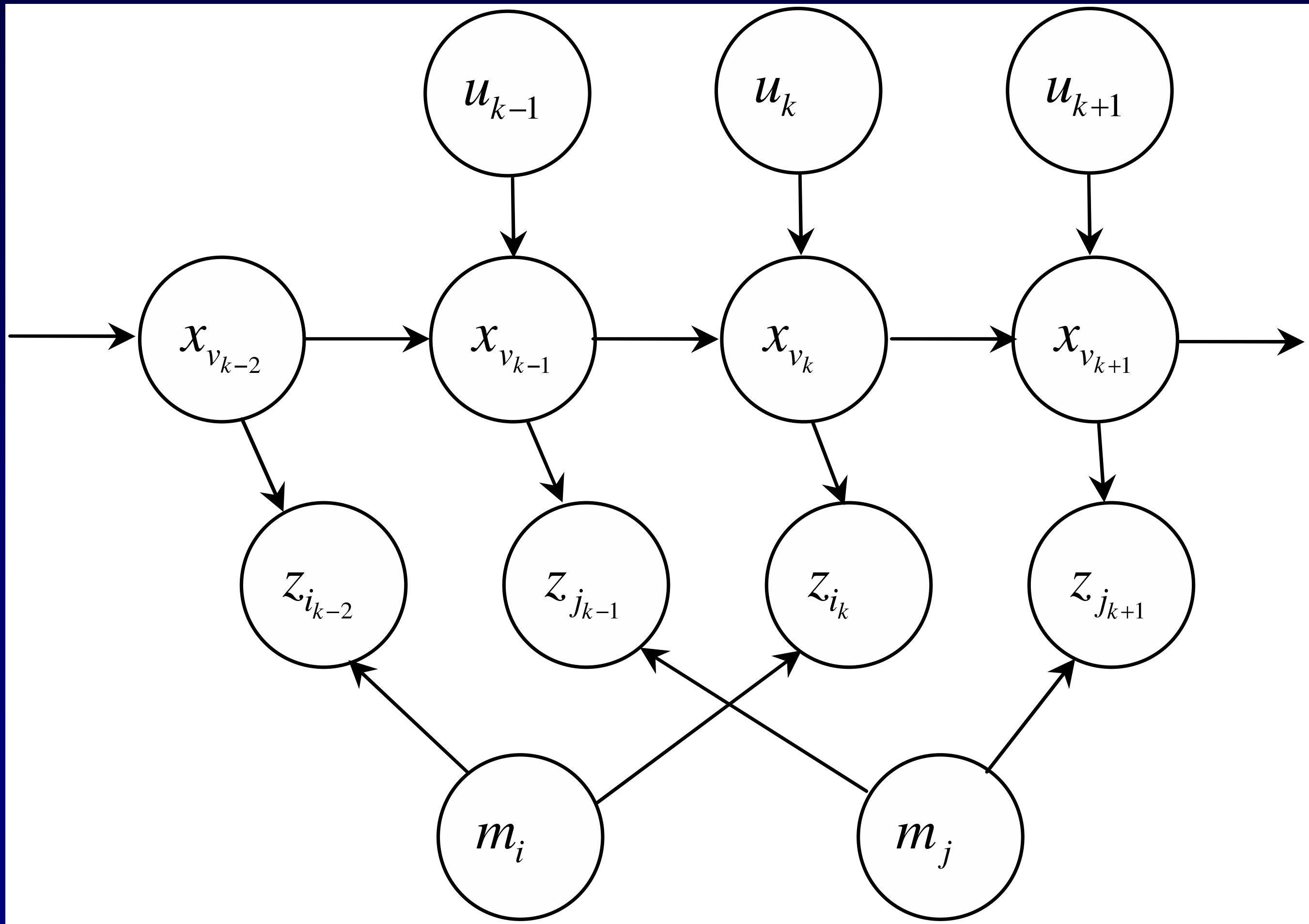
Observation Model

- **Range-bearing measurement**

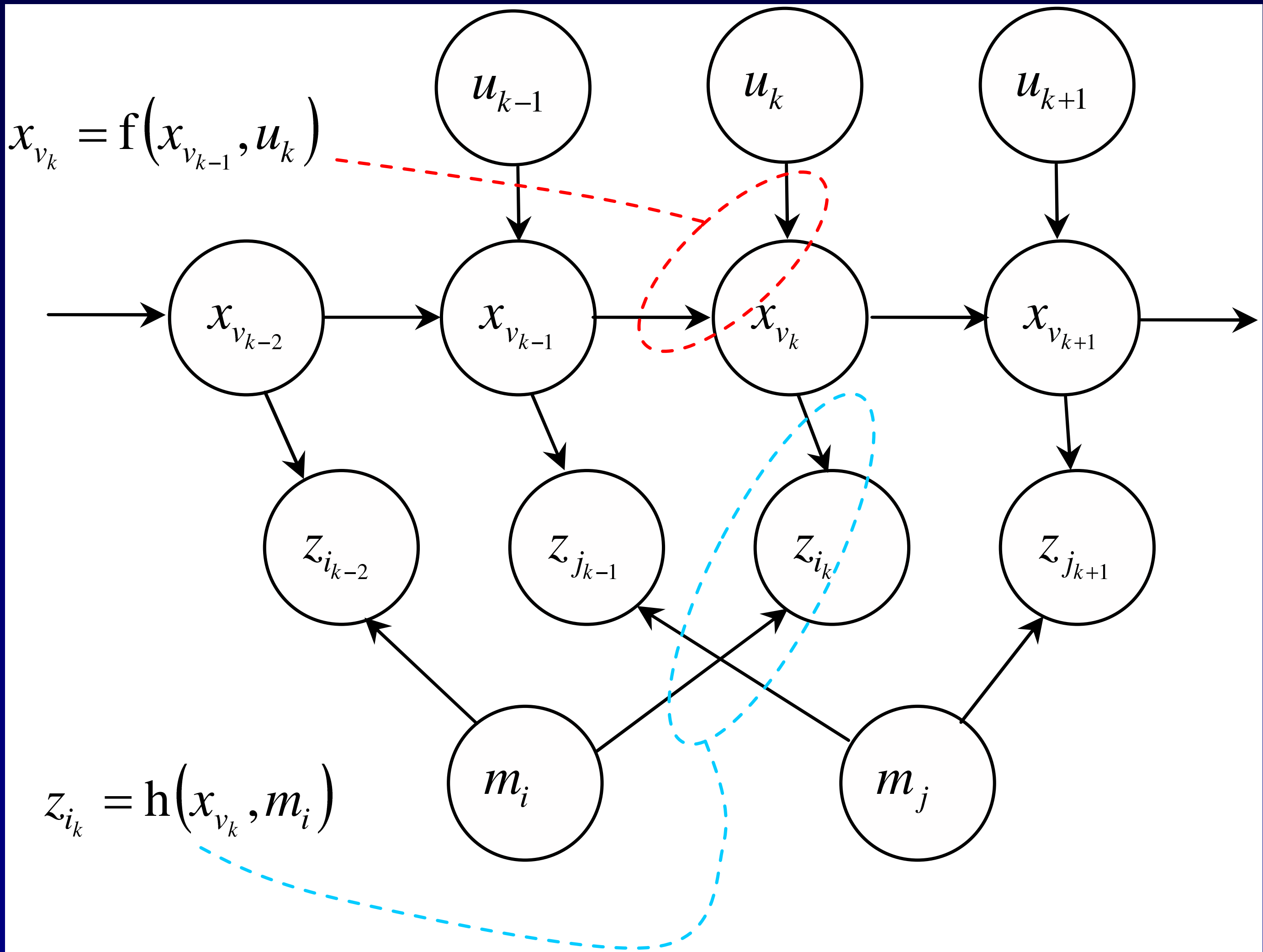


$$\mathbf{z}_{i_k} = \mathbf{h}_i(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(x_i - x_{v_k})^2 + (y_i - y_{v_k})^2} \\ \arctan \frac{y_i - y_{v_k}}{x_i - x_{v_k}} - \phi_{v_k} \end{bmatrix}$$

SLAM Graphical Model



Models in Graphical Model



Perfect World: Deterministic

- **Exact pose from motion model**
- **Global localisation by triangulation**
  - Even if range-only or bearing-only sensors, can localise given enough measurements
  - Solve simultaneous equations: N equations for N unknowns

Real World: Uncertain

- **All measurements have errors**
- **In SLAM, measurement errors induce dependencies in the landmark and vehicle pose estimates**
  - Everything is correlated

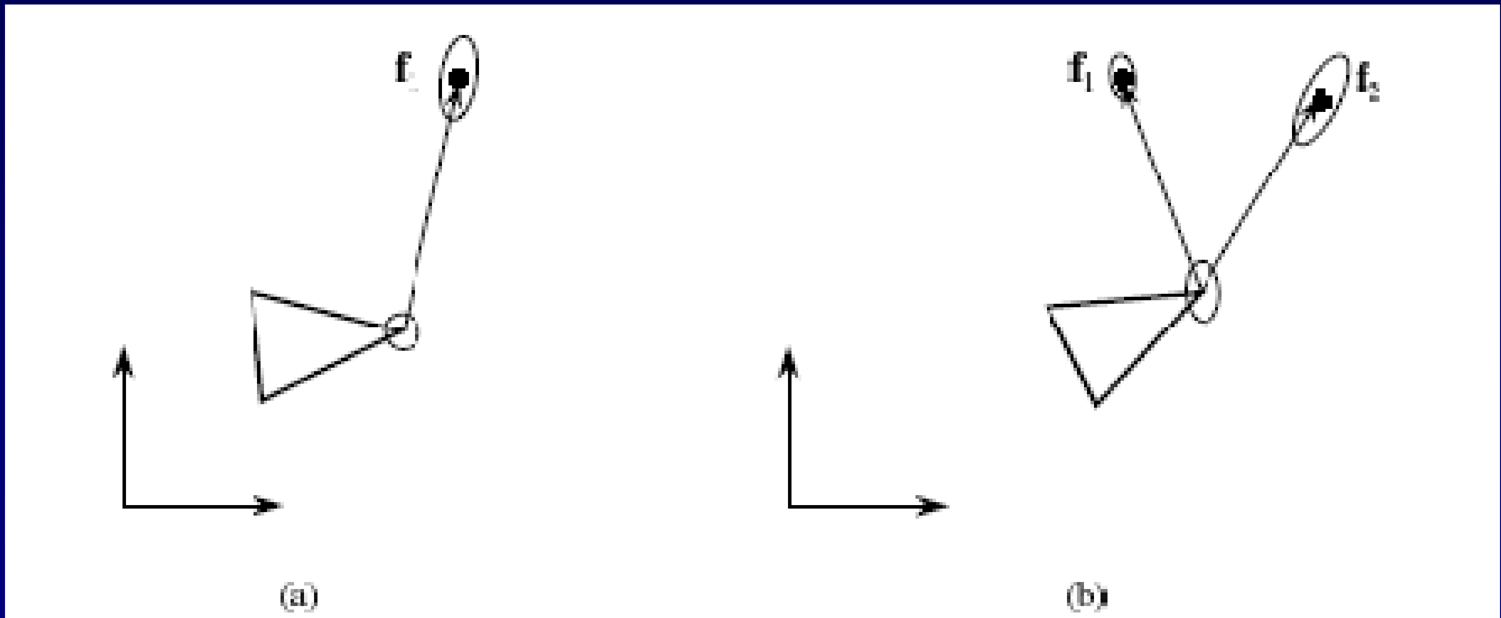
Before All the Hard Maths

- **Key property of stochastic SLAM**
  – Largely a *parameter* estimation problem

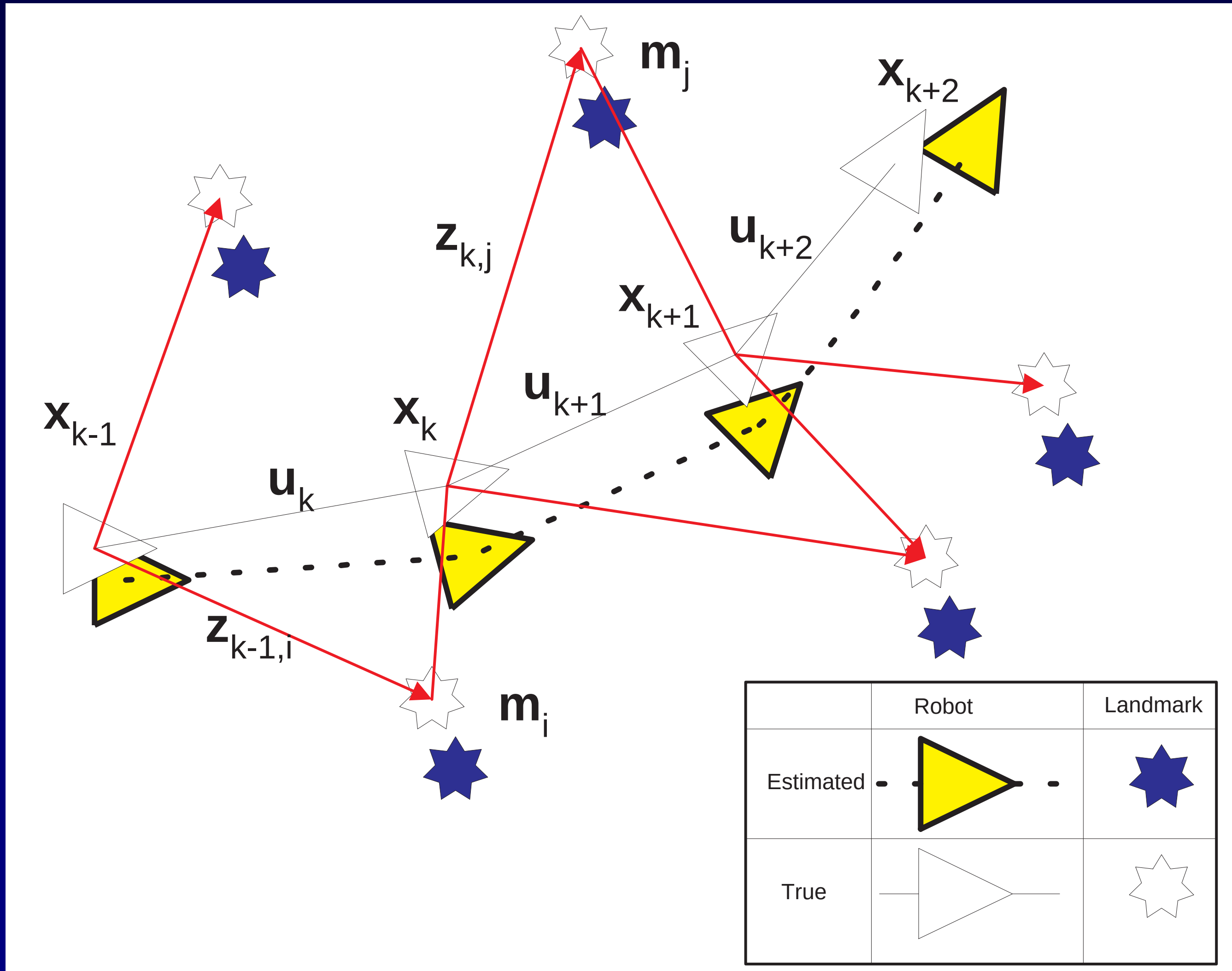
- **Since the map is stationary**
  – No process model, no process noise

- **For Gaussian SLAM**
  – Uncertainty in each landmark reduces monotonically after landmark initialisation
  – Map converges

Dependent Errors



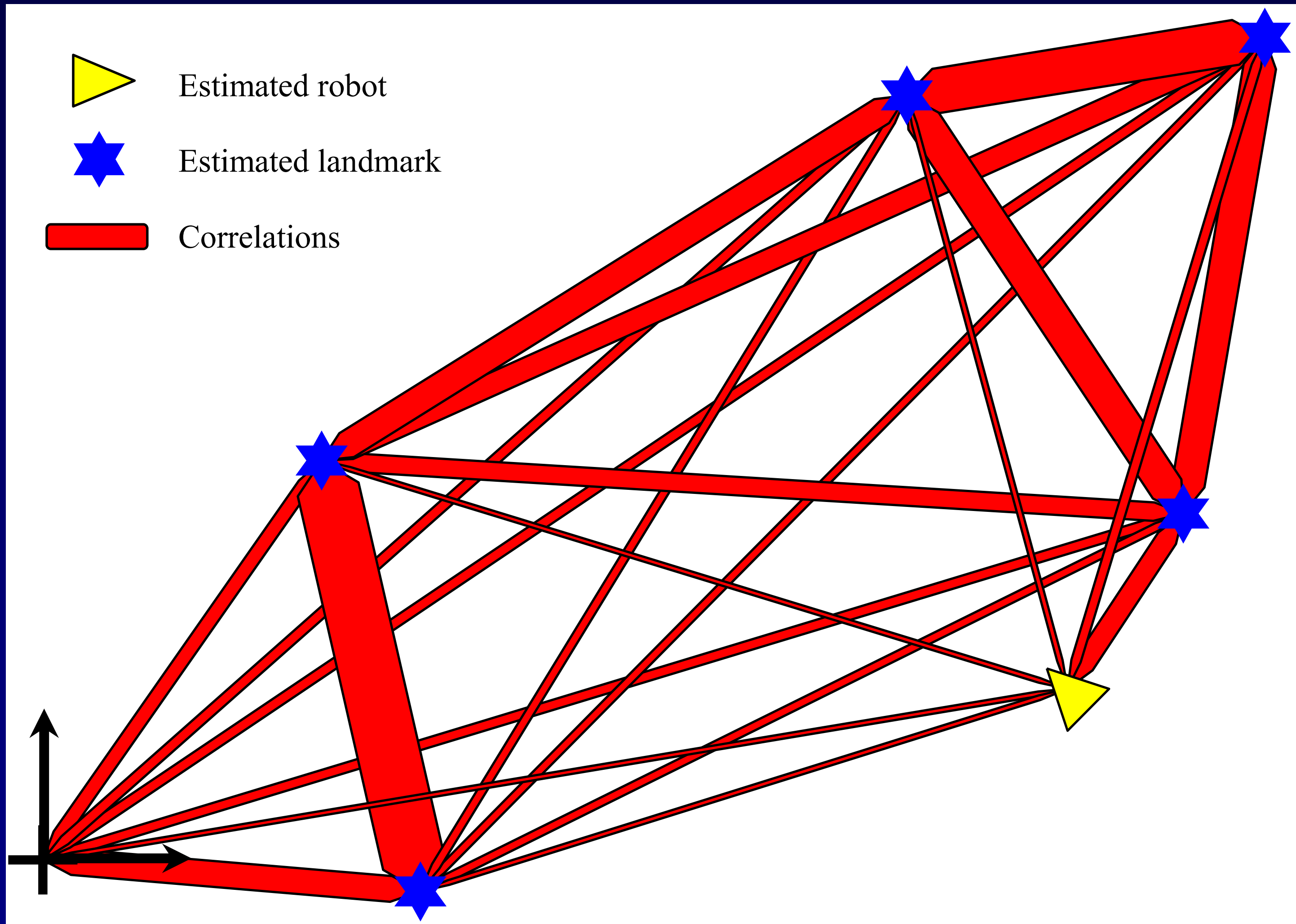
Correlated Estimates



SLAM Convergence

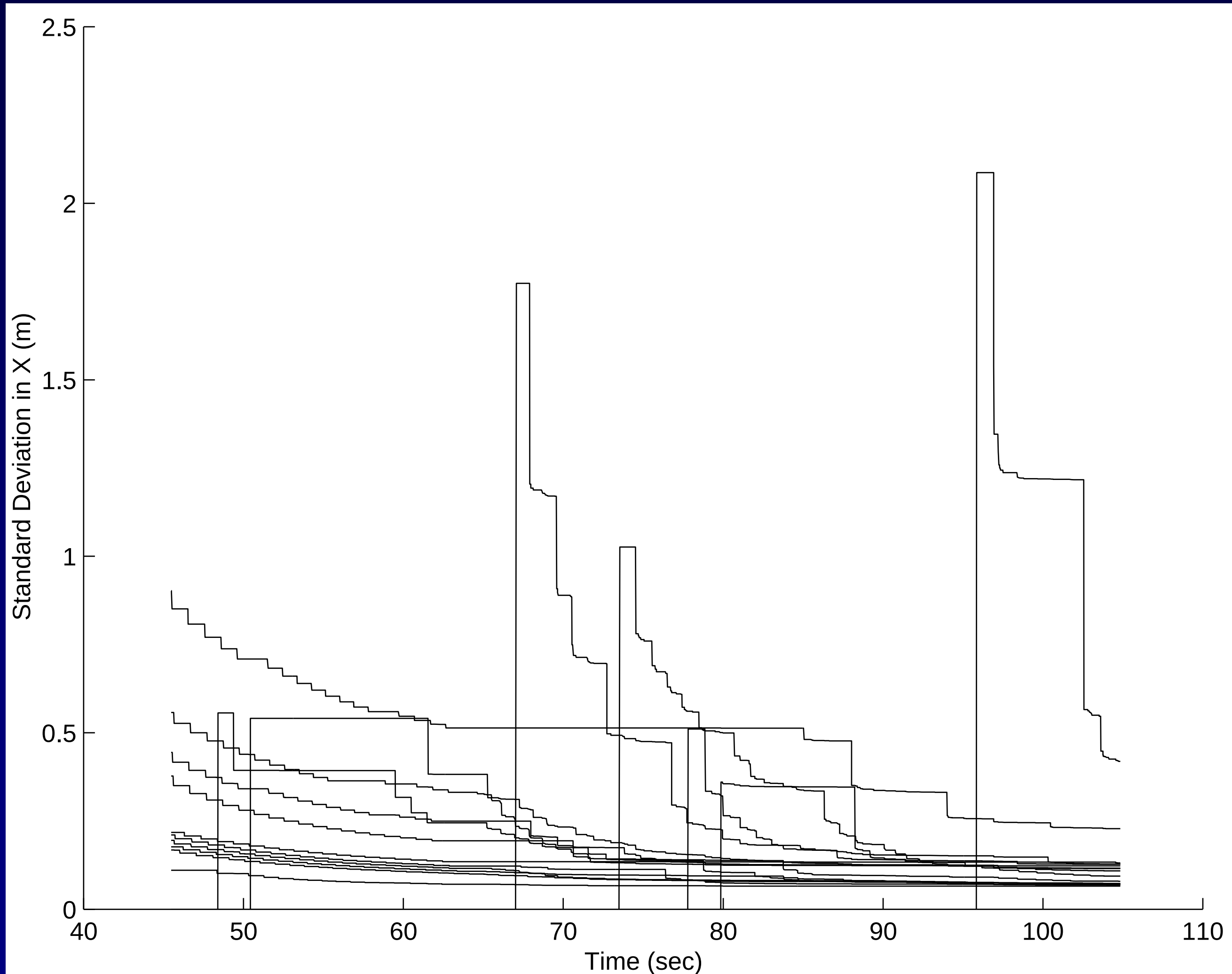
- An observation in a neighbourhood acts like a displacement to a spring system such that it's effect is great in the immediate neighbourhood and, dependent on local stiffness (correlation) properties, diminishes with distance to other landmarks.
- As the robot moves through this environment and takes observations of the landmarks, the springs become increasingly (and monotonically) stiffer.
- In the limit, a rigid map of landmarks or an accurate *relative* map of the environment is obtained.
- As the map is built, the location accuracy of the robot measured relative to the map is bounded only by the quality of the map and relative measurement sensor.
- In the theoretical limit, robot relative location accuracy becomes equal to the localisation accuracy achievable with an *a priori* map.

Spring Analogy (show movie)



Monotonic Convergence

- When a new landmark is initialised, its uncertainty is maximum
- Landmark uncertainty decreases monotonically with each new observation



Non-Gaussian SLAM

- **Convergence results proved for linear Gaussian case**
- **Results do not hold in general for non-Gaussian SLAM even with ideal Bayesian filter**
  - Can contrive (conflicting) likelihood functions that actually *increase* uncertainty when fused
- **However, for all real world scenarios, the convergence results should always hold**
  - Parameter estimation (ie, no process noise) typically gives rise to shrinking uncertainty
- **Note, with approximate estimation all bets are off**

Bayesian Estimation

- Standard theory for dealing with uncertain information in a consistent manner

Brief Overview of Probability Theory

- **Probability density function (PDF) over N-D state space $\mathbf{x} \in \mathcal{X}$ is denoted $p(\mathbf{x})$**

- **Properties of a PDF**

$$\mathbb{R}^N \mapsto \mathbb{R}$$

$$p(\mathbf{x}) \geq 0, \quad \forall\ \mathbf{x} \in \mathcal{X}$$

$$\int_{\mathcal{X}} p(\mathbf{x})\, d\mathbf{x} = 1$$

Brief Overview of Probability Theory

- **State vector**
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}$$
- **Joint PDF is** $p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$
- **Conditional PDF of x_1 *given* x_2 and x_3**
$$p(\mathbf{x}_1 | \mathbf{x}_2, \mathbf{x}_3)$$
- ***Conditional independence*: if x_1 is independent of x_2 given x_3 then**
$$p(\mathbf{x}_1 | \mathbf{x}_2, \mathbf{x}_3) \stackrel{\text{indep}}{=} p(\mathbf{x}_1 | \mathbf{x}_3)$$

Two Essential Rules for Manipulating Probabilities

- **Sum rule**

$$p\left(\mathbf{x}_1|\mathcal{H}\right) \triangleq \int p\left(\mathbf{x}_1, \mathbf{x}_2|\mathcal{H}\right) d\mathbf{x}_2$$

- **Product rule**

$$\begin{aligned} p\left(\mathbf{x}_1, \mathbf{x}_2|\mathcal{H}\right) &\triangleq p\left(\mathbf{x}_1|\mathbf{x}_2, \mathcal{H}\right) p\left(\mathbf{x}_2|\mathcal{H}\right) \\ &\triangleq p\left(\mathbf{x}_2|\mathbf{x}_1, \mathcal{H}\right) p\left(\mathbf{x}_1|\mathcal{H}\right) \end{aligned}$$

Implications of the Product Rule

- **Conditionals**

$$p\left(\mathbf{x}_1|\mathbf{x}_2,\mathcal{H}\right)=\frac{p\left(\mathbf{x}_1,\mathbf{x}_2|\mathcal{H}\right)}{p\left(\mathbf{x}_2|\mathcal{H}\right)}$$

- **Independence**

$$p\left(\mathbf{x}_1,\mathbf{x}_2|\mathcal{H}\right)\overset{\text{indep}}{=}p\left(\mathbf{x}_1|\mathcal{H}\right)p\left(\mathbf{x}_2|\mathcal{H}\right)$$

- **Markov Models**

$$p\left(\mathbf{x}_1|\mathcal{H}\right)=\int p\left(\mathbf{x}_1|\mathbf{x}_2,\mathcal{H}\right)p\left(\mathbf{x}_2|\mathcal{H}\right)d\mathbf{x}_2$$

- **Bayes theorem**

$$p\left(\mathbf{x}_1|\mathbf{x}_2,\mathcal{H}\right)=\frac{p\left(\mathbf{x}_2|\mathbf{x}_1,\mathcal{H}\right)p\left(\mathbf{x}_1|\mathcal{H}\right)}{p\left(\mathbf{x}_2|\mathcal{H}\right)}$$

Bayesian Estimation over a Dynamic State Space

- SLAM has a *dynamic* state space

- Grow state-space: via augmentation
- Reduce state-space: via marginalisation
- Fuse new information: via Bayes theorem

Augmentation: Introduce new states into joint vector

- **Given a prior PDF $p(\mathbf{x}_1)$ and a functional relationship $\mathbf{x}_2 = \mathbf{f}(\mathbf{x}_1, \mathbf{q})$ where q is an independent vector with PDF $p(\mathbf{q})$**
- **We have a conditional PDF**

$$p(\mathbf{x}_2|\mathbf{x}_1) = \int p(\mathbf{x}_2|\mathbf{x}_1, \mathbf{q})\, p(\mathbf{q})\, d\mathbf{q}$$
$$= \int \delta(\mathbf{x}_2 - \mathbf{f}(\mathbf{x}_1, \mathbf{q}))\, p(\mathbf{q})\, d\mathbf{q}$$

- **And form the joint PDF as**

$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_2|\mathbf{x}_1)\, p(\mathbf{x}_1)$$

Marginalisation: Remove old states

- As per the sum rule

$$p(\mathbf{x}_1) = \int p(\mathbf{x}_1, \mathbf{x}_2)\, d\mathbf{x}_2$$
$$= \int p(\mathbf{x}_1 | \mathbf{x}_2)\, p(\mathbf{x}_2)\, d\mathbf{x}_2$$

- **Marginal says: what is PDF of x_1 when we don't care what value x_2 takes; ie, $p(x_1)$ regardless of x_2**
- **Important distinction: x_1 is still dependent on x_2, but $p(x_1)$ is not a *function* of x_2**

Bayesian Update: Inverse probability

- **Bayes theorem** $$p(\mathbf{x}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{x})\, p(\mathbf{x})}{p(\mathbf{z})}$$
- **Observation model** $\mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{r})$
- **Conditional probability**

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}) &= \int p(\mathbf{z}|\mathbf{x}, \mathbf{r})\, p(\mathbf{r})\, d\mathbf{r} \\ &= \int \delta(\mathbf{z} - \mathbf{h}(\mathbf{x}, \mathbf{r}))\, p(\mathbf{r})\, d\mathbf{r} \end{aligned}$$

- **Likelihood function** $\Lambda(\mathbf{x}) = p(\mathbf{z} = \mathbf{z}_0|\mathbf{x})$

Bayes Update

- **Update**

$$p\left(\mathbf{x}|\mathbf{z}=\mathbf{z}_0\right)=\frac{\Lambda(\mathbf{x})p\left(\mathbf{x}\right)}{\int\Lambda(\mathbf{x})p\left(\mathbf{x}\right)d\mathbf{x}}$$

- **Denominator term often seen as just a normalising constant, but is important for saying how likely a model or hypothesis is**
  - Used in FastSLAM for determining particle weights
  - Used in multi-hypothesis data association

Applying Bayes to SLAM: Available Information

- **States** $\mathbf{x}_k$ **(Hidden or inferred values)**
  - Vehicle poses
  - Map; typically composed of discrete parts called landmarks or features
- **Controls** $\mathbf{U}_{0:k}$
  - Velocity
  - Steering angle
- **Observations** $\mathbf{Z}_{0:k}$
  - Range-bearing measurements

Augmentation: Adding new poses and landmarks

- **Add new pose**

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

- **Conditional probability is a Markov Model**

$$\begin{aligned} p\left(\mathbf{x}_{v_k} | \mathbf{x}_{k-1}\right) &= \int p\left(\mathbf{x}_{v_k} | \mathbf{x}_{k-1}, \mathbf{u}_k\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= \int \delta\left(\mathbf{x}_{v_k} - \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right)\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\ &= p\left(\mathbf{x}_{v_k} | \mathbf{x}_{v_{k-1}}\right) \end{aligned}$$

Augmentation

$$
\begin{aligned}
p\left(\mathbf{x}_{v_k} | \mathbf{x}_{k-1}\right) &= \int p\left(\mathbf{x}_{v_k} | \mathbf{x}_{k-1}, \mathbf{u}_k\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\
&= \int \delta\left(\mathbf{x}_{v_k} - \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right)\right) p\left(\mathbf{u}_k\right) d\mathbf{u}_k \\
&= p\left(\mathbf{x}_{v_k} | \mathbf{x}_{v_{k-1}}\right)
\end{aligned}
$$

- Product rule to create joint PDF $p(x_k)$

$$
p\left(\mathbf{x}_{v_k}, \mathbf{x}_{k-1}\right) = p\left(\mathbf{x}_{v_k} | \mathbf{x}_{v_{k-1}}\right) p\left(\mathbf{x}_{v_{k-1}}, \ldots, \mathbf{x}_{v_0}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right)
$$

- Same method applies to adding new landmark states

Marginalisation: Removing past poses and obsolete landmarks

- Augmenting with new pose and marginalising the old pose gives the classical SLAM prediction step

$$p\left(\mathbf{x}_{v_k}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right) = \int p\left(\mathbf{x}_{v_k}, \mathbf{x}_{v_{k-1}}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right) d\mathbf{x}_{v_{k-1}}$$

Fusion: Incorporating observation information

- **Conditional PDF according to observation model**

$$p\left(\mathbf{z}_{i_k}|\mathbf{x}_k\right) = \int p\left(\mathbf{z}_{i_k}|\mathbf{x}_{v_k}, \mathbf{m}_i, \mathbf{r}_k\right) p\left(\mathbf{r}_k\right) d\mathbf{r}_k$$
$$= \int \delta\left(\mathbf{z}_{i_k} - \mathbf{h}\left(\mathbf{x}_{v_k}, \mathbf{m}_i, \mathbf{r}_k\right)\right) p\left(\mathbf{r}_k\right) d\mathbf{r}_k$$

- **Bayes update: proportional to product of likelihood and prior**

$$p\left(\mathbf{x}_k|\mathbf{Z}_{0:k}\right) = \frac{p\left(\mathbf{z}_{i_k} = \mathbf{z}_0|\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{Z}_{0:k-1}\right)}{p\left(\mathbf{z}_{i_k} = \mathbf{z}_0\right)}$$

Implementing Probabilistic SLAM

- **The problem is that Bayesian operations are intractable in general.**
  - General equations are good for analytical derivations, not good for implementation
- **We need approximations**
  - Linearised Gaussian systems (EKF, UKF, EIF, SAM)
  - Monte Carlo sampling methods (Rao-Blackwellised particle filters)

EKF SLAM

- **The complicated Bayesian equations for augmentation, marginalisation, and fusion have simple and efficient closed form solutions for linear Gaussian systems**
- **For non-linear systems, just linearise**
  - EKF, EIF: Jacobians
  - UKF: use deterministic samples

EKF Augmentation

- **Add new pose (adding new landmarks is the same)**
  - Compute mean vector directly from non-linear model

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \mathbf{f}_v\left(\hat{\mathbf{x}}_{v_{k-1}}, \mathbf{u}_k\right) \\ \hat{\mathbf{x}}_{v_{k-1}} \\ \vdots \\ \hat{\mathbf{x}}_{v_0} \\ \hat{\mathbf{m}}_1 \\ \vdots \\ \hat{\mathbf{m}}_N \end{bmatrix}$$

  - Compute covariance by linearisation

Covariance Augmentation

- **Need Jacobians of vehicle motion model with respect to all uncertain variables**
  - Presume, without loss of generality, that all motion uncertainty is contained in control variables u_k and has covariance U_k

$$\nabla \mathbf{f}_{\mathbf{x}} = \left. \frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_{v_{k-1}}} \right|_{(\hat{\mathbf{x}}_{v_{k-1}}, \mathbf{u}_k)}$$

$$\nabla \mathbf{f}_{\mathbf{u}} = \left. \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}_k} \right|_{(\hat{\mathbf{x}}_{v_{k-1}}, \mathbf{u}_k)}$$

- **To simplify notation on next slide, let**

$$\mathbf{x}_\alpha \triangleq \begin{bmatrix} \mathbf{x}_{v_{k-2}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

Covariance Augmentation

- **Covariance before augmentation**

$$\mathbf{P}_{k-1} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_{v_{k-1}}} & \mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} & \mathbf{P}_{\mathbf{x}_\alpha} \end{bmatrix}$$

- **After augmentation**

$$\mathbf{P}_k = \begin{bmatrix} \nabla\mathbf{f}_\mathbf{x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}}\nabla\mathbf{f}_\mathbf{x}^T + \nabla\mathbf{f}_\mathbf{u}\mathbf{U}_k\nabla\mathbf{f}_\mathbf{u}^T & \nabla\mathbf{f}_\mathbf{x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}} & \nabla\mathbf{f}_\mathbf{x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \mathbf{P}_{\mathbf{x}_{v_{k-1}}}\nabla\mathbf{f}_\mathbf{x}^T & \mathbf{P}_{\mathbf{x}_{v_{k-1}}} & \mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha}\nabla\mathbf{f}_\mathbf{x}^T & \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} & \mathbf{P}_{\mathbf{x}_\alpha} \end{bmatrix}$$

EKF Marginalisation

- **Marginalisation of covariance-form Gaussian is entirely trivial: just extract relevant parts**

$$\mathbf{P}_k = \left[\begin{array}{c|c|c} \nabla\mathbf{f_x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}}\nabla\mathbf{f}_\mathbf{x}^T + \nabla\mathbf{f_u}\mathbf{U}_k\nabla\mathbf{f}_\mathbf{u}^T & \nabla\mathbf{f_x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}} & \nabla\mathbf{f_x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \hline \mathbf{P}_{\mathbf{x}_{v_{k-1}}}\nabla\mathbf{f}_\mathbf{x}^T & \mathbf{P}_{\mathbf{x}_{v_{k-1}}} & \mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \hline \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha}\nabla\mathbf{f}_\mathbf{x}^T & \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} & \mathbf{P}_{\mathbf{x}_\alpha} \end{array}\right]$$

$$\mathbf{P}_k = \begin{bmatrix} \nabla\mathbf{f_x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}}\nabla\mathbf{f}_\mathbf{x}^T + \nabla\mathbf{f_u}\mathbf{U}_k\nabla\mathbf{f}_\mathbf{u}^T & \nabla\mathbf{f_x}\mathbf{P}_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha} \\ \mathbf{P}^T_{\mathbf{x}_{v_{k-1}}\mathbf{x}_\alpha}\nabla\mathbf{f}_\mathbf{x}^T & \mathbf{P}_{\mathbf{x}_\alpha} \end{bmatrix}$$

EKF Fusion

- Presume we have a non-linear model with additive noise r that has covariance R

$$\mathbf{z}_{i_k} = \mathbf{h}\left(\mathbf{x}_{v_k}, \mathbf{m}_i\right) + \mathbf{r}$$

- Standard EKF update step: an N^2 operation

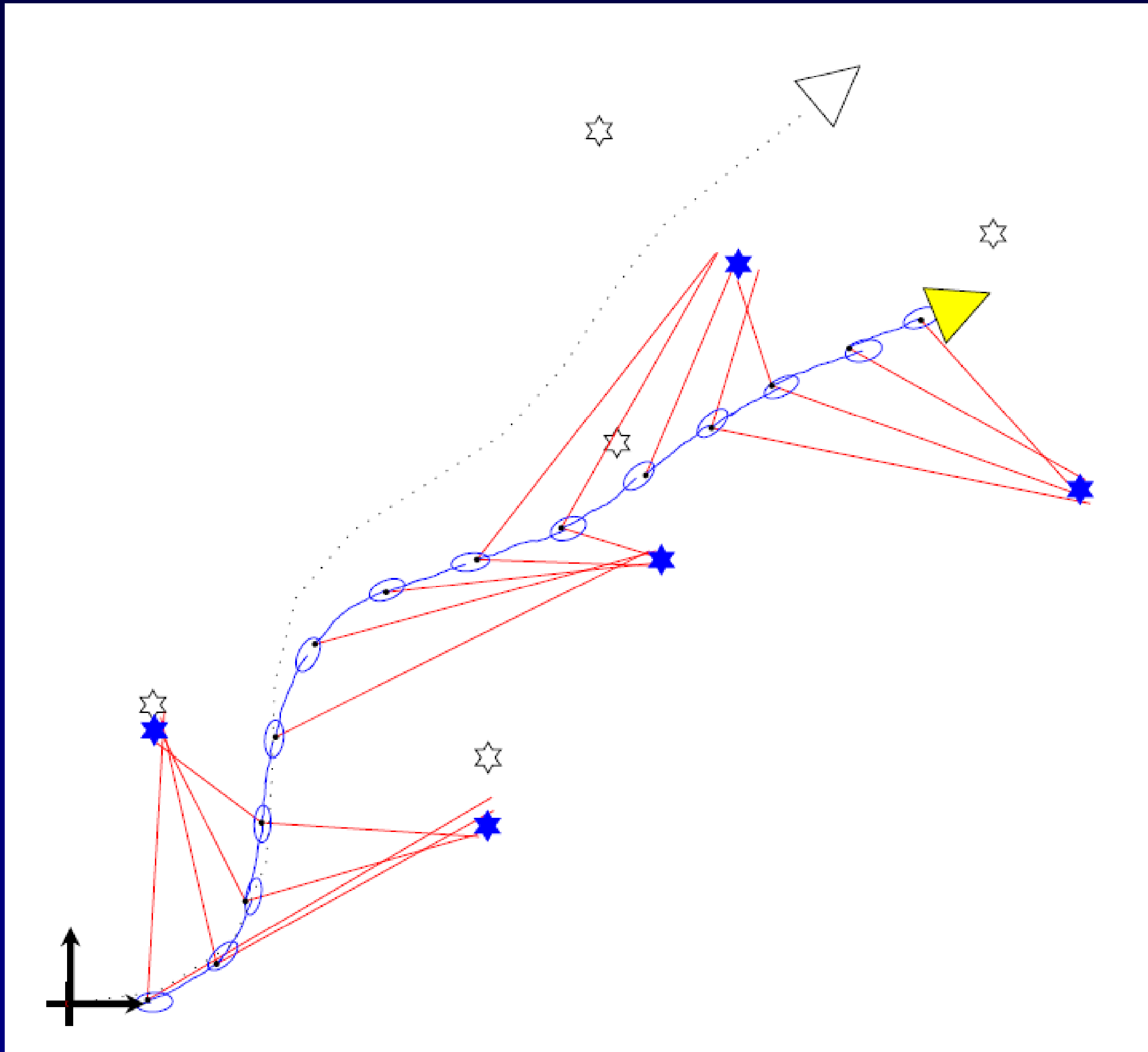
Problem with EKF SLAM

- **Difficult to manage data association ambiguity efficiently**
  - Especially difficult if environment is cluttered, dynamic, or has structural similarities
- **Linearisation of models can badly corrupt statistics**
  - Biggest issues seems to be variation in linearisation point

Particle Filter SLAM

- **The FastSLAM algorithm introduced by Montemerlo and Thrun**
- **Rao-Blackwellised particle filter**
  - Particles for vehicle pose states
  - Each particle represents an entire *pose history* or trajectory
  - Each particle has bank of independent EKFs for landmark states
- **Deals well with non-linear vehicle motion model and ambiguous data association**

FastSLAM: Propagation of a single particle



Problem with FastSLAM

- **Suffers from a problem common to all particle filter estimators with stationary parameters.**
- **Particle weights diverge over successive observations (weight degeneracy)**
  - Left with a single particle of significant weight
- **Weight degeneracy mitigated by resampling**
  - Causes loss of historical diversity
- **Poor statistical quality over time**

Hybrid SLAM

- **Combine FastSLAM and EKF-SLAM using submapping methods (Brooks 08)**
- **FastSLAM frontend**
  - Deals with nonlinearities and data association
- **EKF backend**
  - Retains long term correlation information

What about Data Association?

- **Data association is the problem of assigning a given measurement to a particular portion of the state vector (eg, which landmark was observed?)**
- **Is in fact a model selection problem**
- **Fits properly into the Bayesian framework as discrete probabilities.**
  - We have a finite set of observation models assigning a measurement to each landmark, including a previously unseen landmark.
- **Full evaluation hard. Usually approximate:**
  - Validation gating (eg, JCBB, CCDA)
    - Note: one should always reject ambiguous measurements, never do nearest neighbour association
  - Multi-hypothesis (eg, FastSLAM)

In Conclusion

- **This talk gave a whirlwind overview of the probabilistic foundations of SLAM**
  – Sorry about all those complicated equations
- **Solution is composed of**
  – Motion and observation models; notably stationary landmarks
  – Basic mechanics of Bayesian estimation
- **Estimation involves**
  – Rules of augmentation, marginalisation, fusion (which are built on two fundamentals: sum and product rules)
  – Also, we have data association, which is probabilities over discrete states
- **Real systems are approximations to the ideal non-linear Bayesian estimator**
  – Linear Gaussian
  – Monte Carlo sampling
- **Many practical details for robust and efficient implementation, which will be presented in subsequent talks**