Probabilistic Robotics: Localization

METR 4202: Advanced Control & **Robotics**

Dr Surya Singh -- Lecture # 8

September 16, 2015

metr4202@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~metr4202/

Schedule

Week	Date	Lecture (W: 12:05-1:50, 50-N201)
1	29-Jul	Introduction
2	5-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	12-Aug	Robot Kinematics Review (& *Ekka Day*)
4	19-Aug	Robot Dynamics
5	26-Aug	Robot Sensing: Perception
6	2-Sep	Robot Sensing: Multiple View Geometry
7	9-Sep	Robot Sensing: Feature Detection (as Linear Observers)
8	**16-Sep**	**Probabilistic Robotics: Localization**
9	23-Sep	**Quiz** & Guest Lecture (SLAM?)
	30-Sep	*Study break*
10	7-Oct	Motion Planning
11	14-Oct	State-Space Modelling
12	21-Oct	Shaping the Dynamic Response
13	28-Oct	LQR + Course Review

Quiz Next Week: Two Example Problems

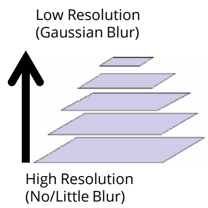
1. Is the following 3x3 rotation matrix (within practical numerical limits) a valid rotation matrix?

$$R_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0.8660 & 0.5000 & 0 \\ 0.5000 & 0.8660 & 0 \end{bmatrix}$$

A: No. The determinant is 0.5.
Note that (3,1) should be -0.5.

2. Where does the Scale-Invariant in SIFT Comes From?

A: Scale-Space/
The Scale Pyramid



Localization: SFM → SLAM

- ACFR View:
  - Treat as joint estimation problem
- [New] Oxford View:
  - Treat as (feature) placement optimization problem
  - Bundle Adjustment (borrow from computer vision)

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

How to quantify uncertainty ? Probability to the rescue…

Bertsekas & Tsitsiklis, Introduction to Probability.

- FATHER(F): Nurse, what is the probability that the drug will work?
- NURSE (N): I hope it works, we'll know tomorrow.
- F: Yes, but what is the probability that it will?
- N: Each case is different, we have to wait.
- F: But let's see, out of a hundred patients that are treated under similar conditions, how many times would you expect it to work?
- N (somewhat annoyed): I told you, every person is different, for some it works, for some it doesn't.
- F (insisting): Then tell me, if you had to bet whether it will work or not, which side of the bet would you take?
- N (cheering up for a moment): I'd bet it will work.
- F (somewhat relieved): OK, now, would you be willing to lose two dollars if it doesn't work, and gain one dollar if it does?
- N (exasperated): What a sick thought! You are wasting my time!

Probability review 1/4: Probabilistic Modeling

- View:
  - Experiments with random outcome.
  - Quantifiable properties of the outcome.
- Three components:
  - Sample space: Set of all possible outcomes.
  - Events: Subsets of sample space.
  - Probability: Quantify how likely an event occurs.

Probability review 2/4: Probability

- Probability: A function that maps events to real numbers satisfying these axioms:
  1. Non-negativity: $P(E) \geq 0$, where E is an event.
  2. Normalization: $P(S) = 1$, where S is the sample space.
  3. Additivity of finite / countably infinite events.

$$P\left(\bigcup_{i=1}^{\infty/n} E_i\right) = \sum_{i=1}^{\infty/n} P(E_i),$$

where E_i are disjoint / mutually exclusive, i : natural number.

Probability review 3/4: Random Variables

- Interest is on numerical values associated w. samples, e.g.:
  - Sample 50 students enrolled in METR4202, what's the major of most of the students.
  - Roll a fair dice, get \$5 if the outcome is even, & loose \$5 if the outcome is odd.
- Random variable X is a function $X : S \rightarrow Num.$
  - Num: countable set (e.g., integer) → discrete random variable.
  - Num: uncountable set (e.g., real) → continuous random variable.

Probability review 4/4: Characterizing Random Variables

- Cumulative distribution function (cdf)

$$F_X(x) = P(X \leq x) = P\left(\left\{s \middle| X(s) \leq x, s \in S\right\}\right)$$

- Discrete: Probability mass function (pmf)

$$f_X[x] = P(X = x)$$

- Continuous: Probability density function/probability distribution function (pdf)

$$f_X(x) = \frac{dF_X(x)}{dx} \quad ; \quad P(a \leq X \leq b) = \int_a^b f_X(x)dx$$

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

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

Marginalisation: Remove old states

- As per the sum rule

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

- 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 $\quad p(\mathbf{x}|\mathbf{z}) = \dfrac{p(\mathbf{z}|\mathbf{x})\, p(\mathbf{x})}{p(\mathbf{z})}$
- Observation model $\quad \mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{r})$
- 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 $\quad \Lambda(\mathbf{x}) = p(\mathbf{z} = \mathbf{z}_0 | \mathbf{x})$

Bayes Update

- Update $$p(\mathbf{x}|\mathbf{z}=\mathbf{z}_0) = \frac{\Lambda(\mathbf{x})p(\mathbf{x})}{\int \Lambda(\mathbf{x})p(\mathbf{x})\,d\mathbf{x}}$$

- 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

Bayesian Estimation

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

SLAM!
(Better than SMAL!)

What is SLAM?

- SLAM asks the following question:

  Is it possible for an autonomous vehicle to start at an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute vehicle location?

- SLAM has many indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

- Examples
  - Explore and return to starting point (Newman)
  - Learn trained paths to different goal locations
  - Traverse a region with complete coverage (eg, mine fields, lawns, reef monitoring)
  - …

Components of 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
  - Generate a map from on-board sensors
  - Compute pose from this map
  - Errors in map and in pose estimate are dependent.

History of SLAM

- It all started about 20 years ago at ICRA86 in San Francisco.
  - Probabilistic methods were new to robotics and AI
  - Several researchers were looking at applying estimation-theoretic methods to mapping and localisation problems
- They saw that:
  - Consistent probabilistic mapping was a fundamental problem
  - Major conceptual and computational issues needed to be addressed
- Key papers were written on geometric uncertainty (Smith and Cheeseman, HDW).
  - They showed that estimates exhibit a high degree of correlation between geometric features (ie, landmark locations in a map).

History of SLAM

- Landmark paper by Smith, Self and Cheeseman
  - Landmark estimates correlated via vehicle pose estimate
- Important implication
  - A consistent full solution requires a joint state composed of the vehicle pose and every landmark position
  - Many landmarks means huge state vector
  - Estimate to be updated following each landmark observation
- At the time, estimation meant Kalman filters
  - Computation and storage costs scale quadratically with the number of landmarks

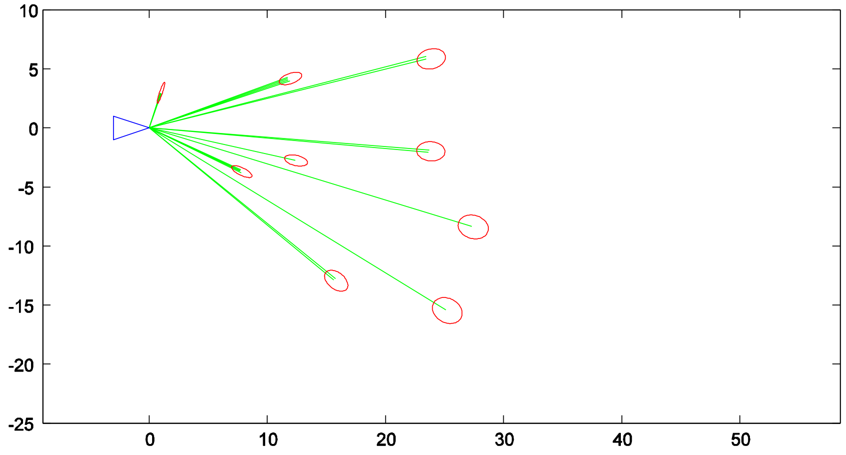
History of SLAM

- Researchers initially thought SLAM would not converge
  - Assumed estimated map errors would exhibit a random walk behaviour
  - Unbounded map uncertainty
  - Leonard (1991): Simultaneous mapping and localisation, which came first, the chicken or the egg ?
- Researchers tried to *minimise* correlations between landmarks
  - Applied approximations to minimise or eliminate correlations, or simply assumed they were zero
  - Reduced the full filter to a series of decoupled landmark to vehicle filters.
- Theoretical work on SLAM came to a temporary halt
  - Work focused on either mapping or localisation as separate problems.

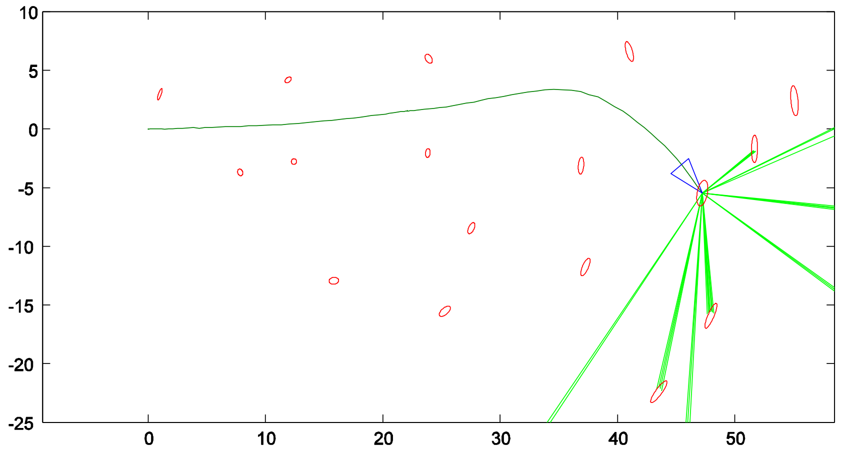
Basic SLAM Operation

Example: SLAM in Victoria Park

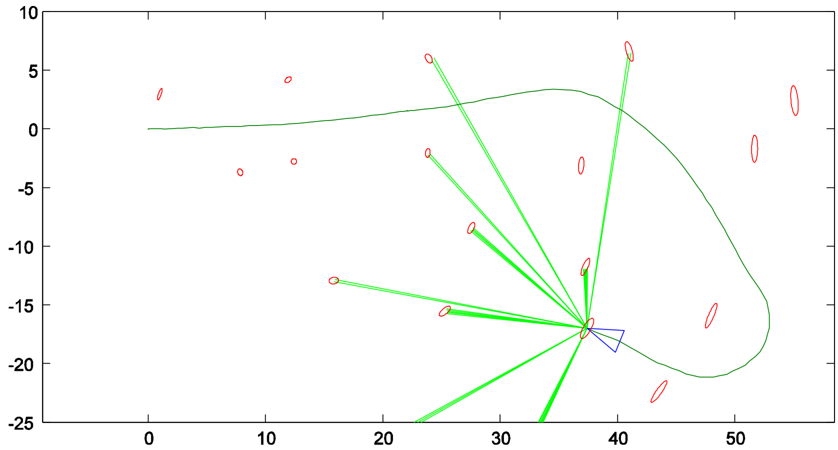
Basic SLAM Operation



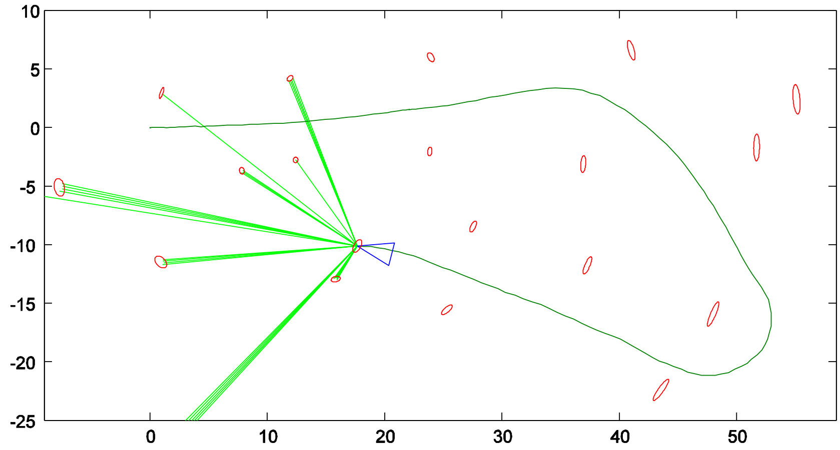
Basic SLAM Operation



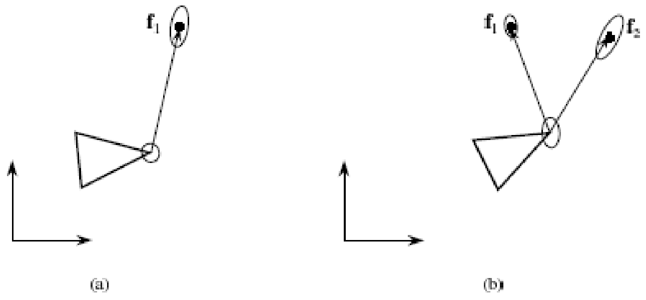
Basic SLAM Operation



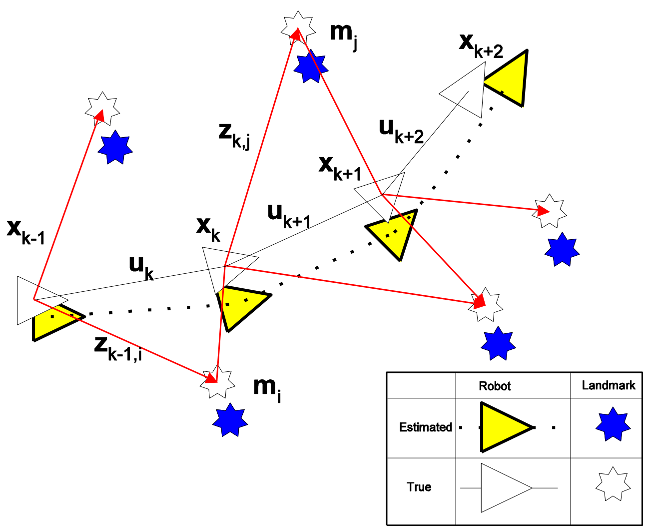
Basic SLAM Operation



Dependent Errors



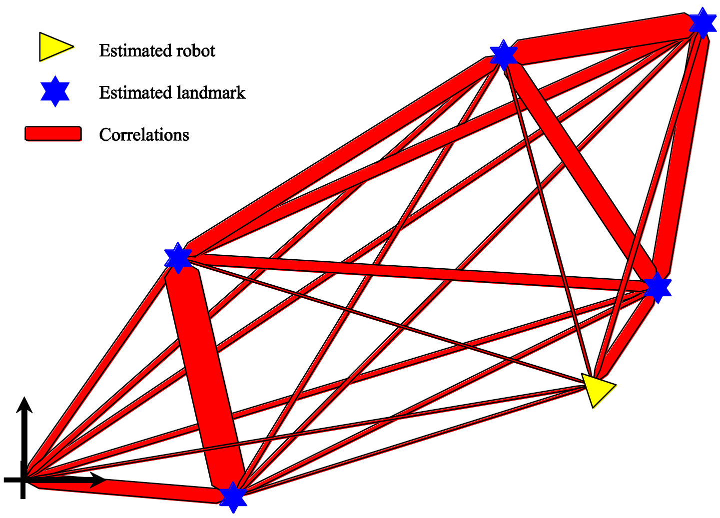
Correlated Estimates



SLAM Convergence

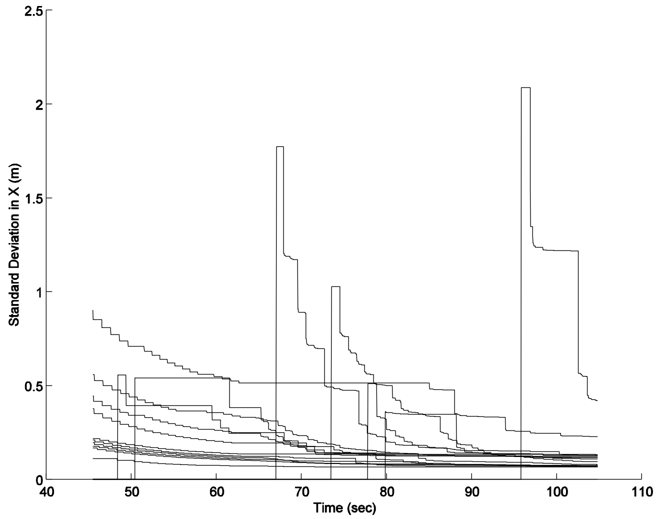
- An observation acts like a displacement to a spring system
  - Effect is greatest in a close neighbourhood
  - Effect on other landmarks diminishes with distance
  - Propagation depends on local stiffness (correlation) properties
- With each new observation the springs become increasingly (and monotonically) stiffer.
- In the limit, a rigid map of landmarks is obtained.
  - A perfect *relative* map of the environment
- The location accuracy of the robot is bounded by
  - The current quality of the map
  - The relative sensor measurement

Spring Analogy



Monotonic Convergence

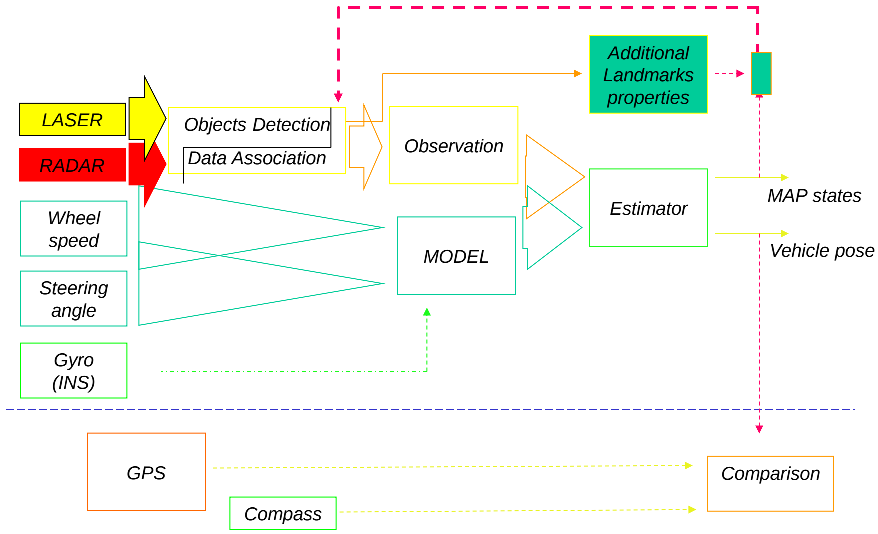
- With each new observation, the determinant decreases over the map and for any submatrix in the map.



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

An Example System



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

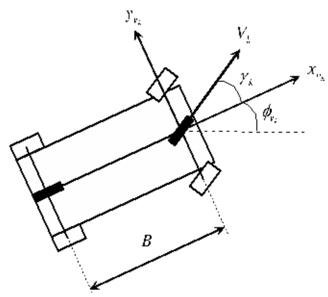
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

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

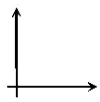
Observations: $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$

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

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

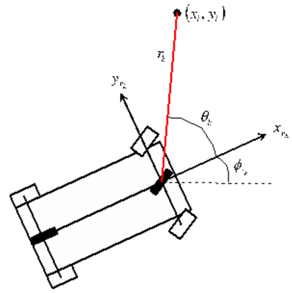
- Joint state: Landmarks are assumed stationary

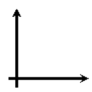
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix} \qquad \mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{x}_{v_{k-1}} \\ \vdots \\ \mathbf{x}_{v_0} \\ \mathbf{m}_1 \\ \vdots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

Observation Model

- Range-bearing measurement

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





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 $p(x_k)$

$$p\left(\mathbf{x}_{v_k}, \mathbf{x}_{k-1}\right) = p\left(\mathbf{x}_{v_k} | \mathbf{x}_{v_{k-1}}\right) p\left(\mathbf{x}_{v_{k-1}}, \ldots, \mathbf{x}_{v0}, \mathbf{m}_1, \ldots, \mathbf{m}_N\right)$$

- 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

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

- 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

Kalman Implementation

- So can we just plug the process and observation models into the standard EKF equations and turn the crank?

- Several additional issues:
  - Structure of the SLAM problem permits more efficient implementation than naïve EKF.
  - Data association.
  - Feature initialisation.

Structure of SLAM

- 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

- Examine computational consequences of this structure in next session.

Data Association

- Before the Update Stage we need to determine if the feature we are observing is:
  - An old feature
  - A new feature

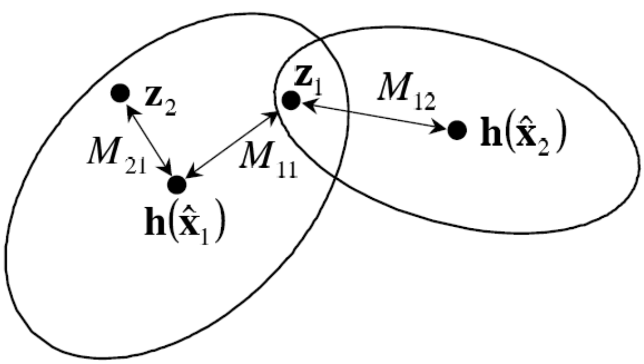
- If there is a match with only one known feature, the Update stage is run with this feature information.

$$\mu(k) = z(k) - h(\hat{x}(k/k-1))$$

$$S(k) = \nabla h_x(k) P(k/k-1) \nabla h_x^T(k) + R$$

$$\alpha = \mu^T(k) S^{-1}(k)\ \mu(k) < \chi^2_{0.95}$$

Validation Gating



New Features

- If there is no match then a potential new feature has been detected

- We do not want to incorporate a spurious observation as a new feature
  - It will not be observed again and will consume computational time and memory
  - It will add clutter, increasing risk of future mis-associations
  - The features are assumed to be static. We don't not want to accept dynamic objects as features: cars, people etc.

Acceptance of New Features

- **APPROACH 1**

  - Get the feature in a list of potential features
  - Incorporate the feature once it has been observed for a number of times

  - Advantages:
    - Simple to implement
    - Appropriate for High Frequency external sensor

  - Disadvantages:
    - Loss of information
    - Potentially a problem with sensor with small field of view: a feature may only be seen very few times

Acceptance of New Features

- **APPROACH 2**
  - The state vector is extended with past vehicle positions and the estimation of the cross-correlation between current and previous vehicle states is maintained. With this approach improved data association is possible by combining data form various points
    - J. J. Leonard and R. J. Rikoski. Incorporation of delayed decision making into stochastic mapping
    - Stephan Williams, PhD Thesis, 2001, University of Sydney
  - Advantages:
    - No Loss of Information
    - Well suited to low frequency external sensors (ratio between vehicle velocity and feature rate information)
    - Absolutely necessary for some sensor modalities (eg, range-only, bearing-only)
  - Disadvantages:
    - Cost of augmenting state with past poses
    - The implementation is more complicated

Incorporation of New Features

- **We have the vehicle states and previous map**

$$P_0 = \begin{bmatrix} P^0_{v,v} & P^0_{v,m} \\ P^0_{m,v} & P^0_{m,m} \end{bmatrix}$$

We observed a new feature and the covariance and cross-covariance terms need to be evaluated

$$P_1 = \begin{bmatrix} P^0_{v,v} & P^0_{v,m} & ? \\ P^0_{m,v} & P^0_{m,m} & ? \\ ? & ? & ? \end{bmatrix}$$

Incorporation of New Features

- Approach 1

$$P_0 = \begin{bmatrix} P_{vv}^0 & P_{vm}^0 & 0 \\ P_{mv}^0 & P_{mm}^0 & 0 \\ 0 & 0 & A \end{bmatrix}$$

With A very large

$$W(k) = P(k/k-1)\nabla h_x^T(k) S^{-1}(k)$$
$$S(k) = \nabla h_x(k) P(k/k-1) \nabla h_x^T(k) + R$$
$$P(k/k) = P(k/k-1) - W(k)S(k)W^T(k)$$

$$P_1 = \begin{bmatrix} P_{vv}^1 & P_{vm}^1 & P_{vn}^1 \\ P_{mv}^1 & P_{mm}^1 & P_{mn}^1 \\ P_{nv}^1 & P_{nm}^1 & P_{nn}^1 \end{bmatrix}$$

- **Easy to understand and implement**
- **Very large values of A may introduce numerical problems**

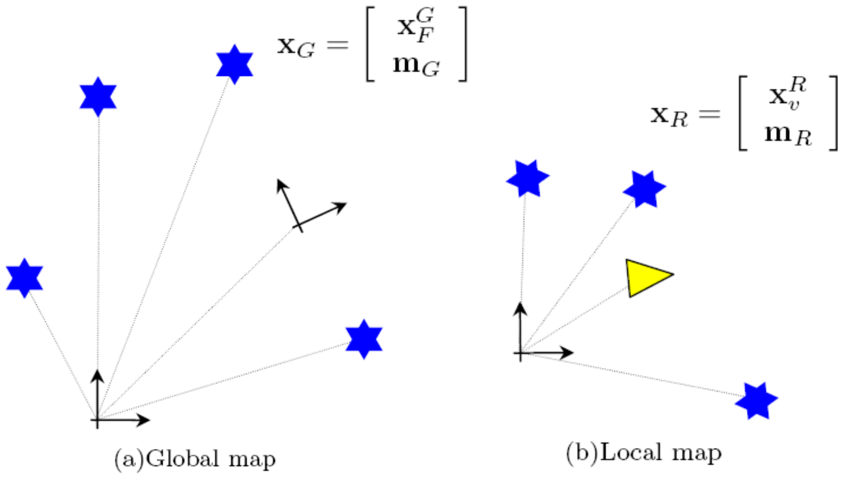
Analytical Approach

$$P_0 = \begin{bmatrix} P_{v,v}^0 & P_{v,m}^0 \\ P_{m,v}^0 & P_{m,m}^0 \end{bmatrix}$$

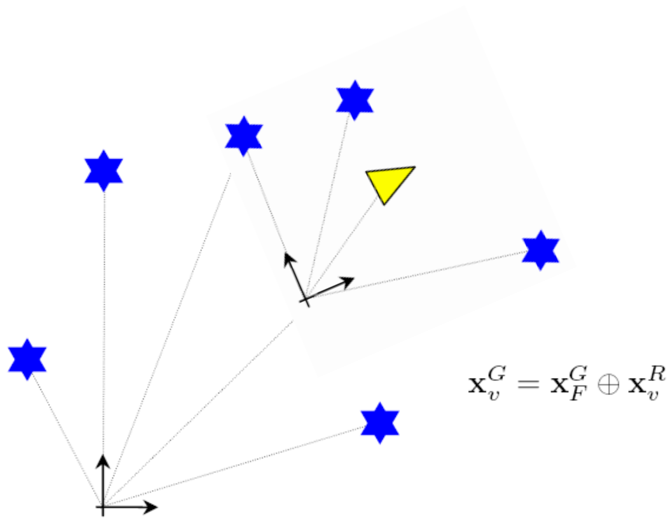
- **We can also evaluate the analytical expressions of the new terms**

$$P_1 = \begin{bmatrix} P_{v,v}^0 & P_{v,m}^0 & ? \\ P_{m,v}^0 & P_{m,m}^0 & ? \\ ? & ? & ? \end{bmatrix}$$

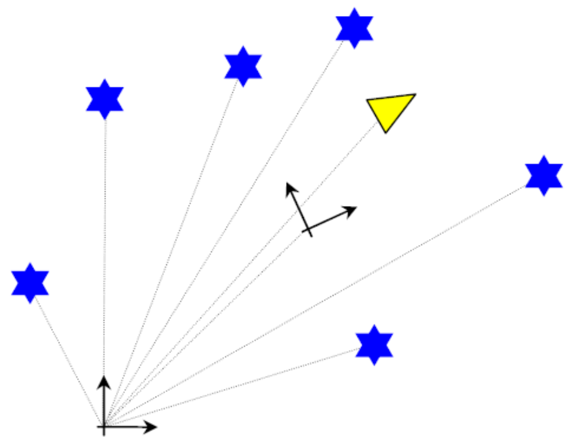
Constrained Local Submap Filter



CLSF Registration



CLSF Global Estimate



More Cool Robotics Share!