SLAM

Borrowed Heavily from Tim Bailey

Why 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

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.

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 localization 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.

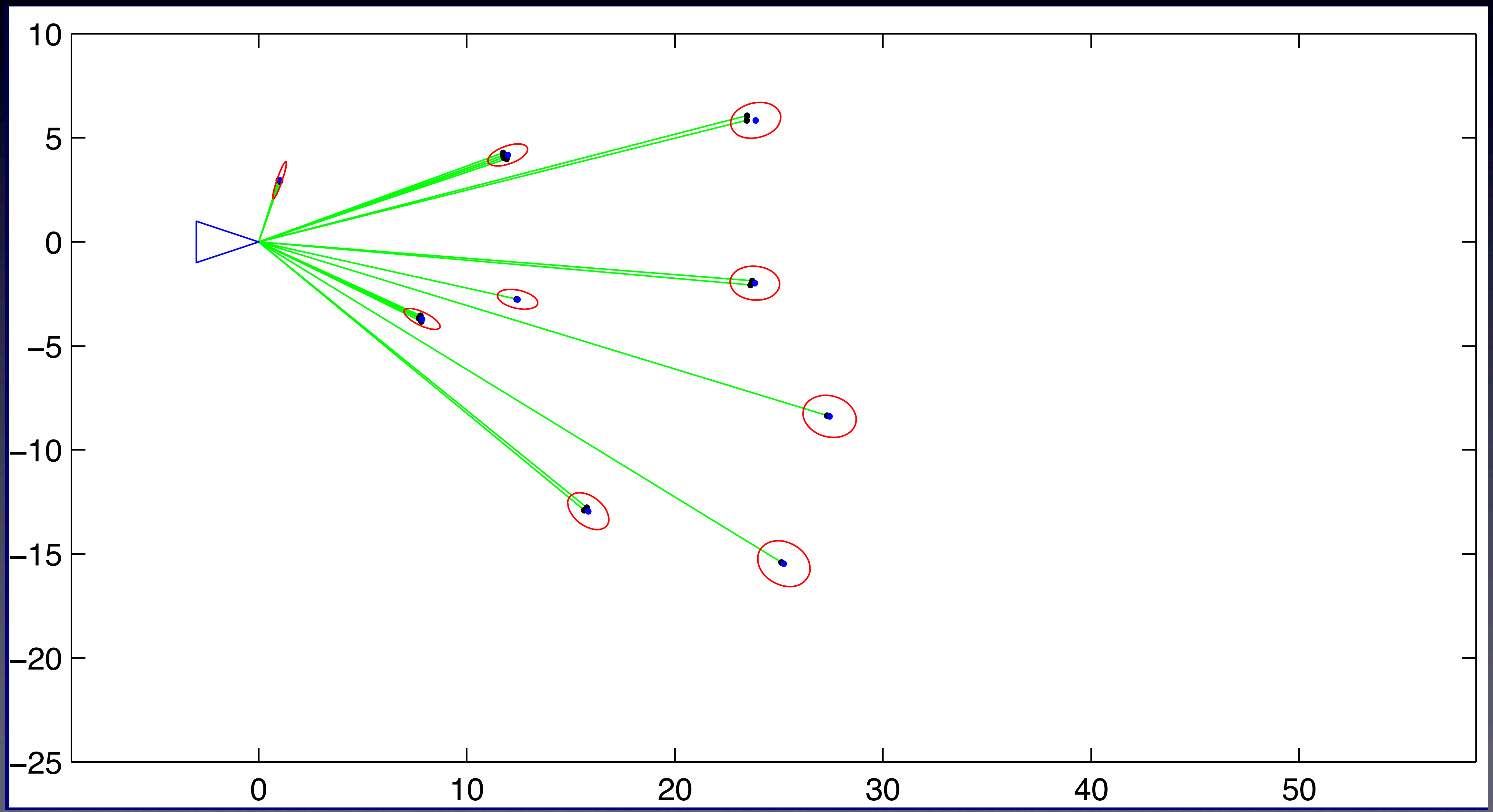
Implications Smith, Self, and Cheeseman

- A consistent full solution to the combined localization 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.

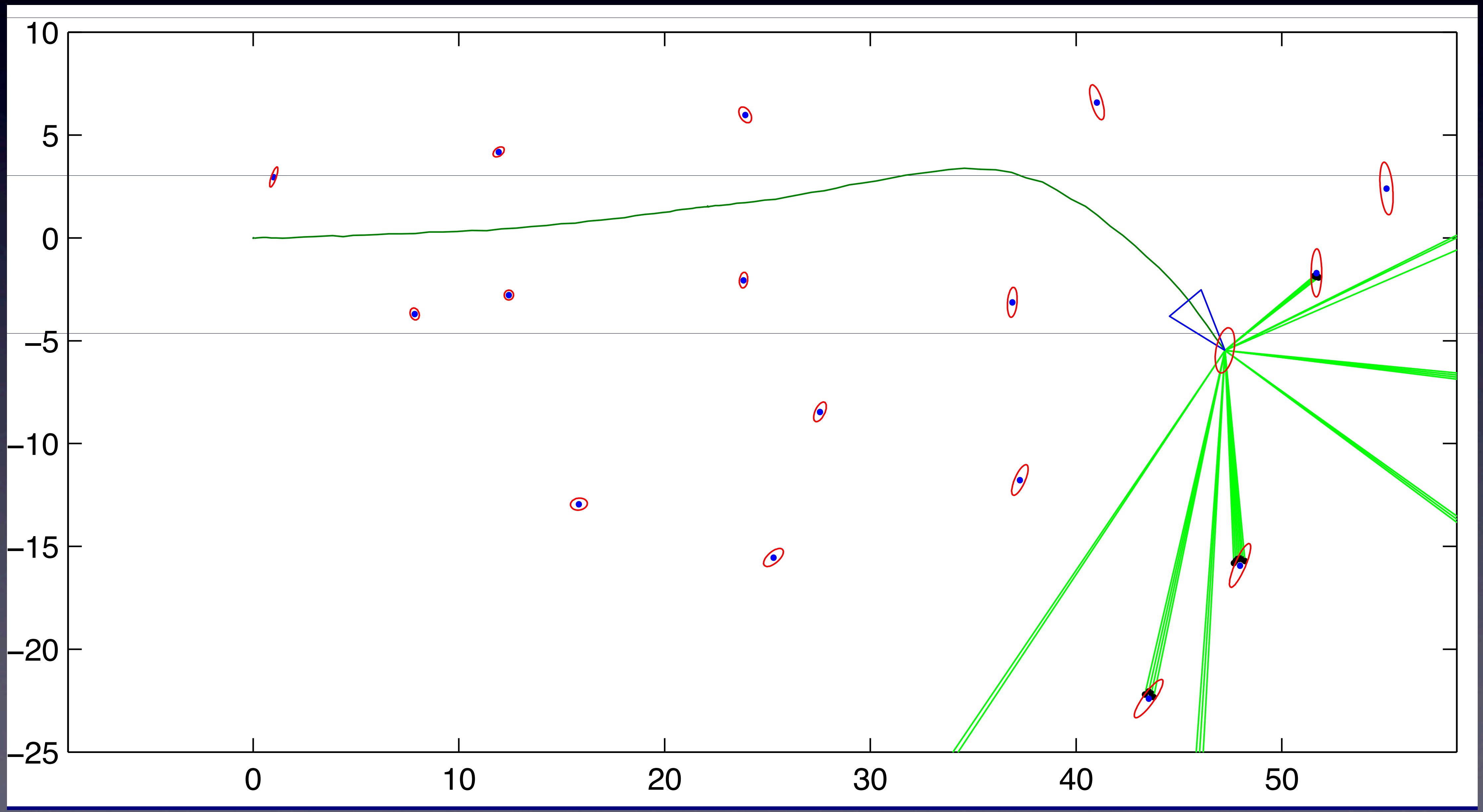
Map Errors

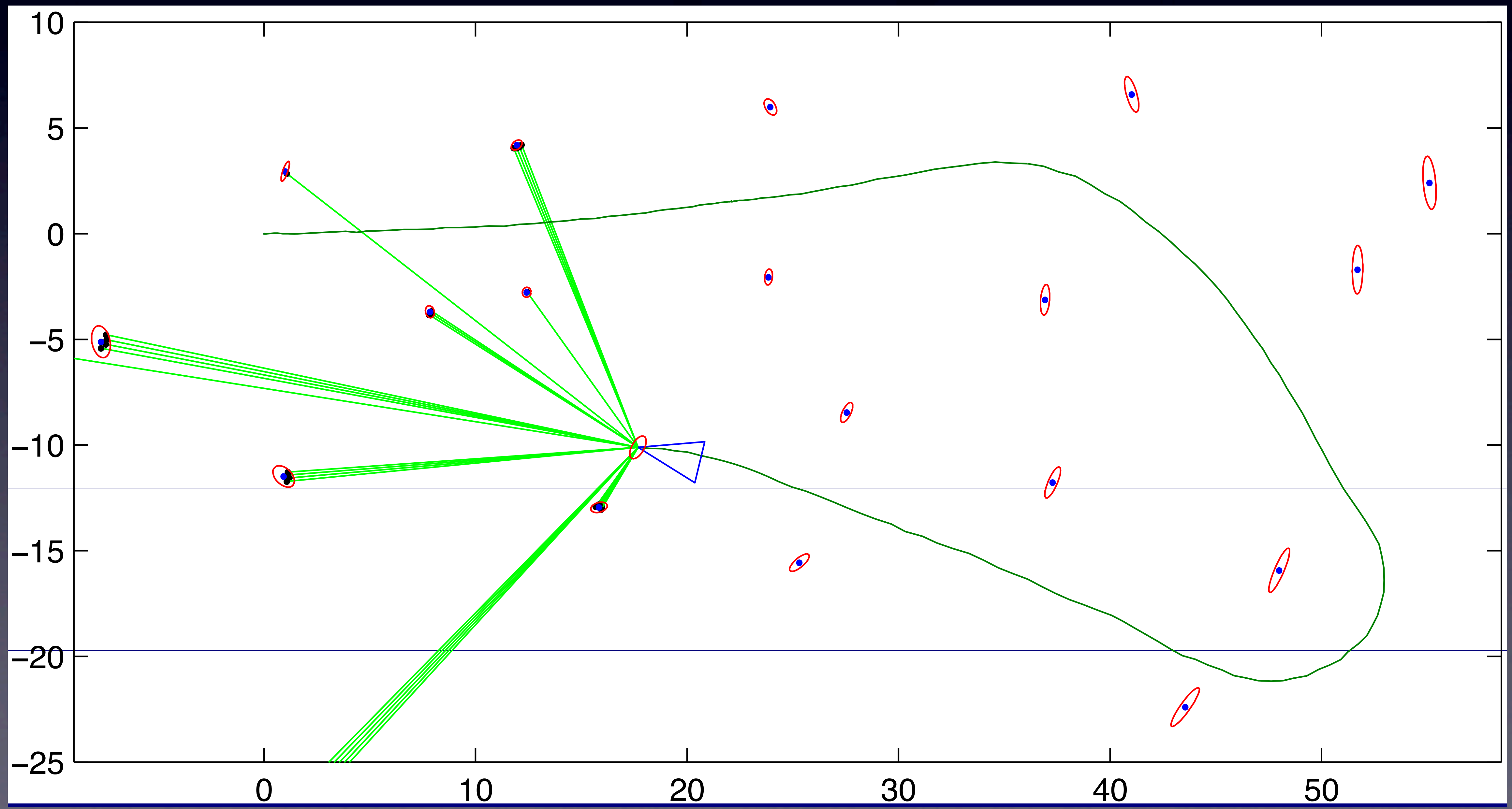
- It was assumed at the time that the estimated map errors would not converge and would instead exhibit a random walk behavior with unbounded error growth.
- Conceptual break-through: the combined mapping and localization problem, once formulated as a single estimation problem, is convergent (Csorba 96, Dissa 01).
    - Recognize 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.

Basic SLAM Components



10
5
0
−5
−10
−15
−20
−25
0
10
20
30
40
50


10
5
0
−5
−10
−15
−20
−25
0
10
20
30
40
50



10
5
0
−5
−10
−15
−20
−25
0
10
20
30
40
50

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

Alternatives

- 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

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ m_1 \\ \dots \\ m_n \end{bmatrix}$$

Joint State with Pose History

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{x}_{v_{k-1}} \\ \dots \\ \mathbf{x}_{v_0} \\ m_1 \\ \dots \\ m_n \end{bmatrix}$$

Control Inputs/Observations

Control Inputs

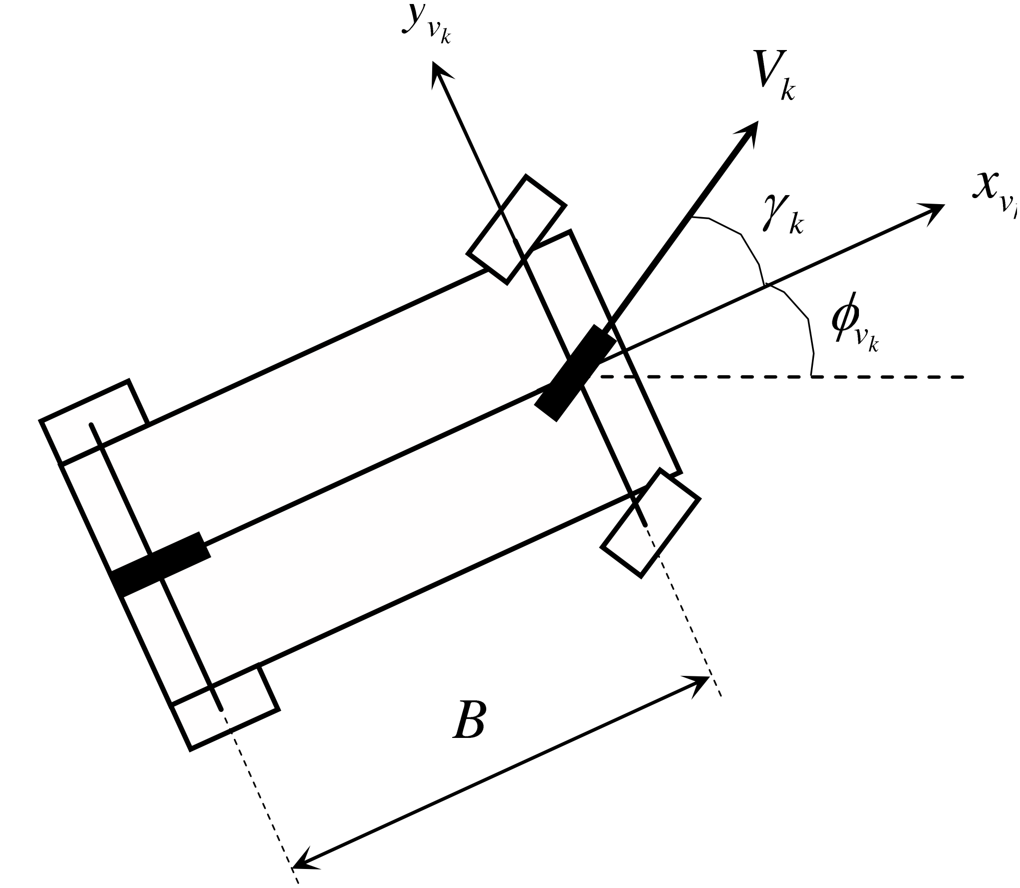
$$\mathbf{U}_{0:k} = \{\mathbf{u}_0, \ldots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u_k}\}$$

Observations

$$\mathbf{Z}_{0:k} = \{\mathbf{z}_0, \ldots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z_k}\}$$

Motion Model

Ackerman steered vehicles:
Bicycle model



Discrete time model:

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

SLAM Motion Model

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

- Joint State Landmarks are Stationary

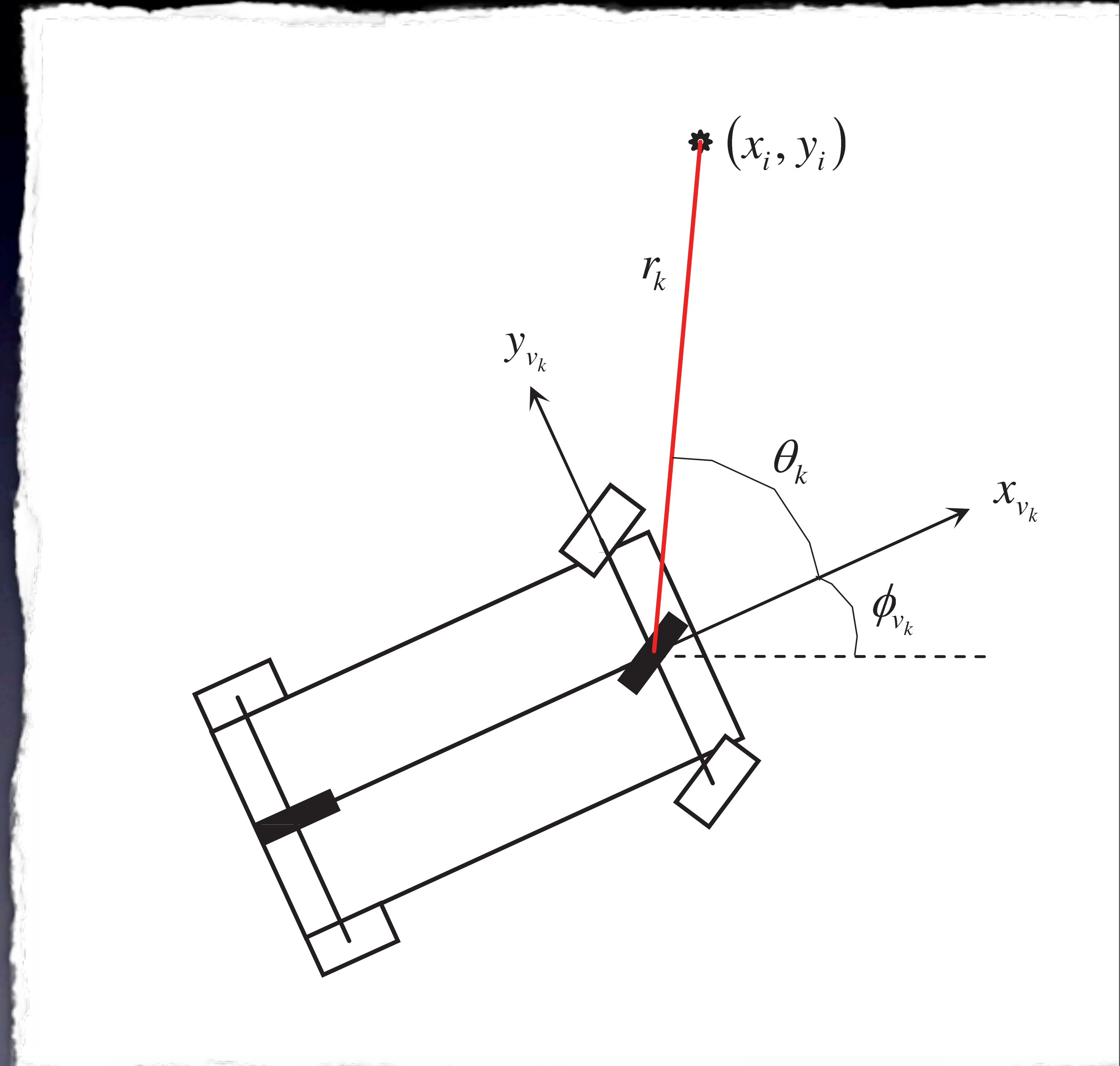
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix}$$

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

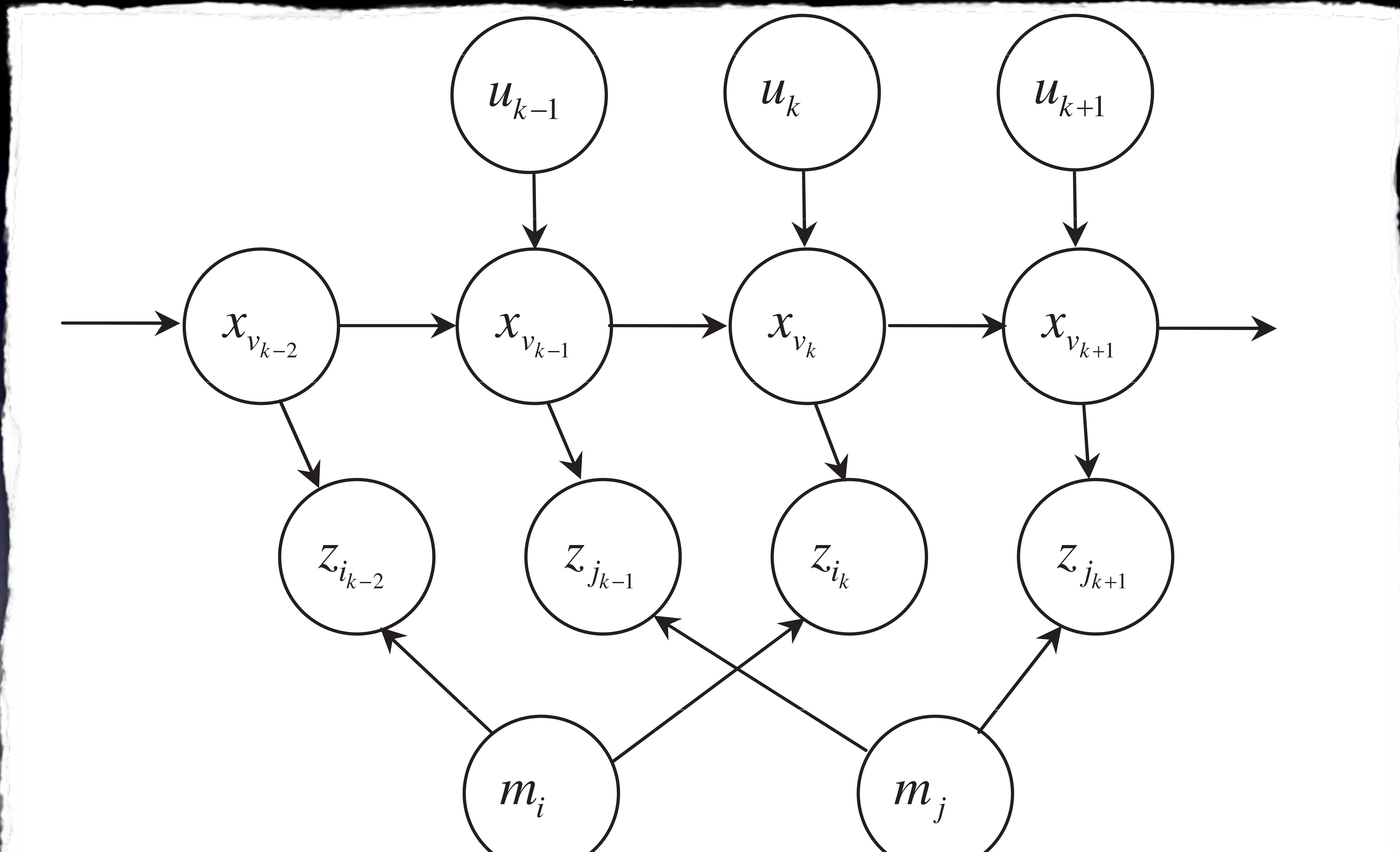
Observation Model

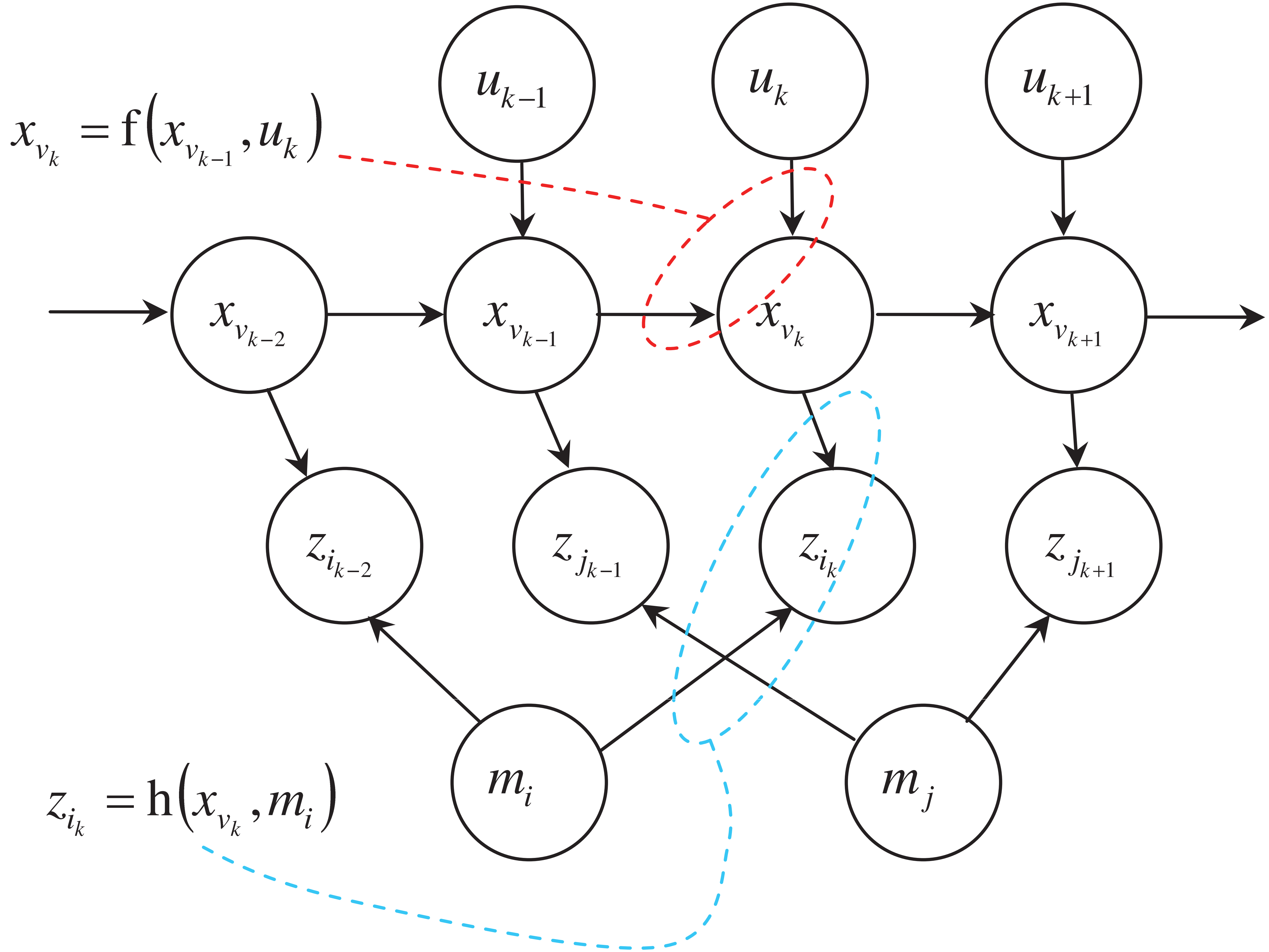
- Range Bearing Measurement

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



SLAM Graphical Model




$x_{v_k} = \mathrm{f}(x_{v_{k-1}}, u_k)$
u_{k-1}
u_k
u_{k+1}
$x_{v_{k-2}}$
$x_{v_{k-1}}$
x_{v_k}
$x_{v_{k+1}}$
$z_{i_{k-2}}$
$z_{j_{k-1}}$
z_{i_k}
$z_{j_{k+1}}$
m_i
m_j
$z_{i_k} = \mathrm{h}(x_{v_k}, m_i)$

Perfect World: Deterministic

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

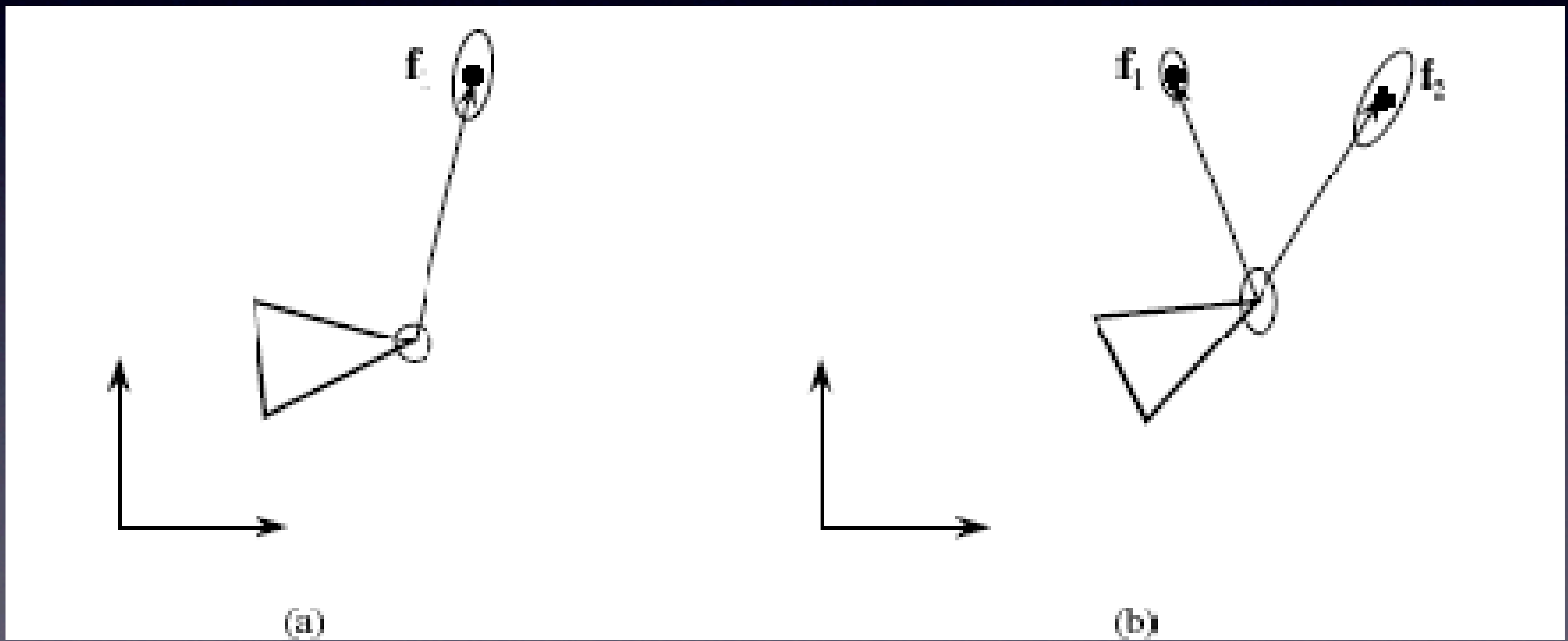
Real World: Uncertain

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

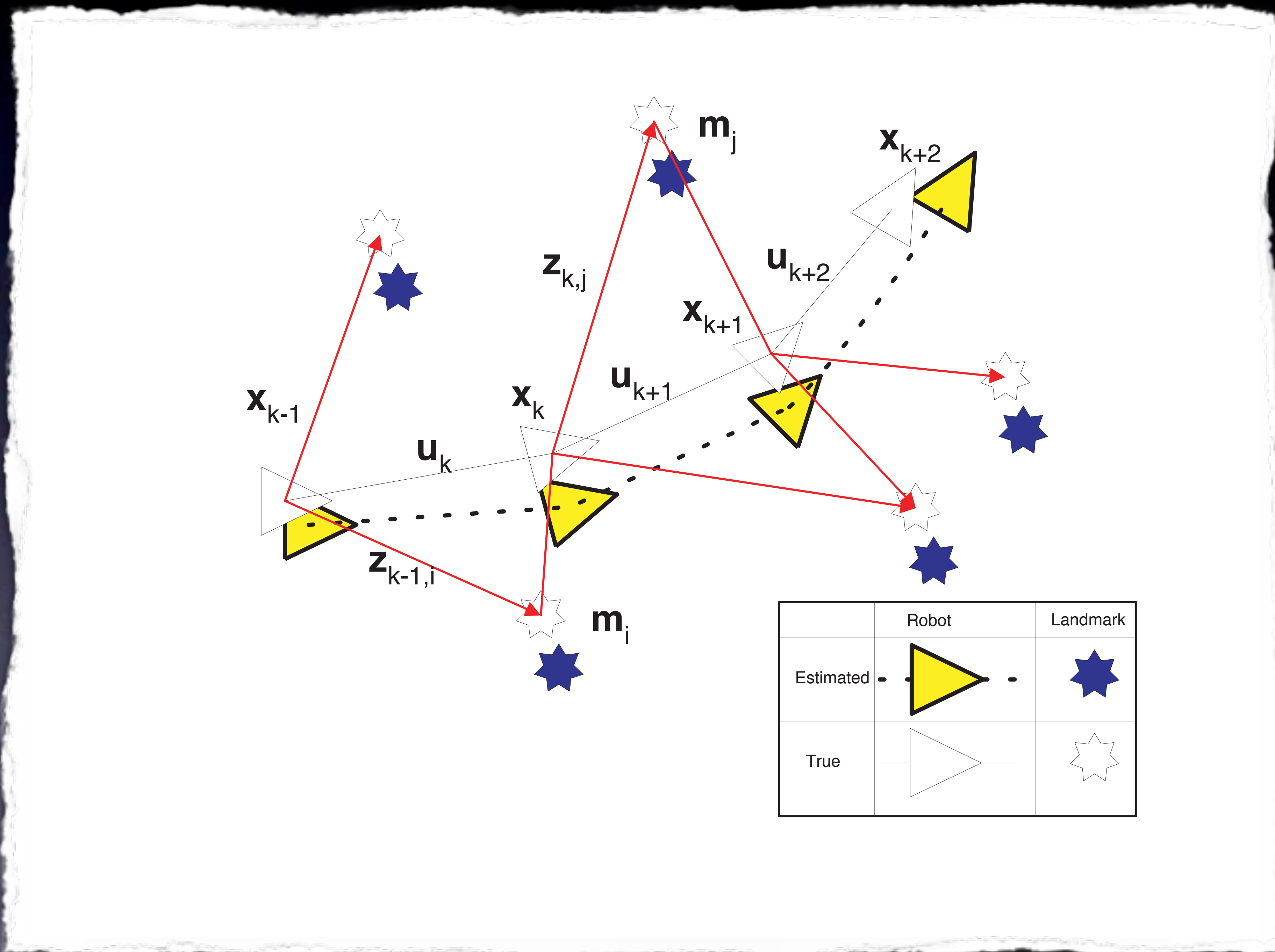
Key property of stochastic SLAM

- 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 initialization
    - Map converges

Correlated Error



Correlated Estimates

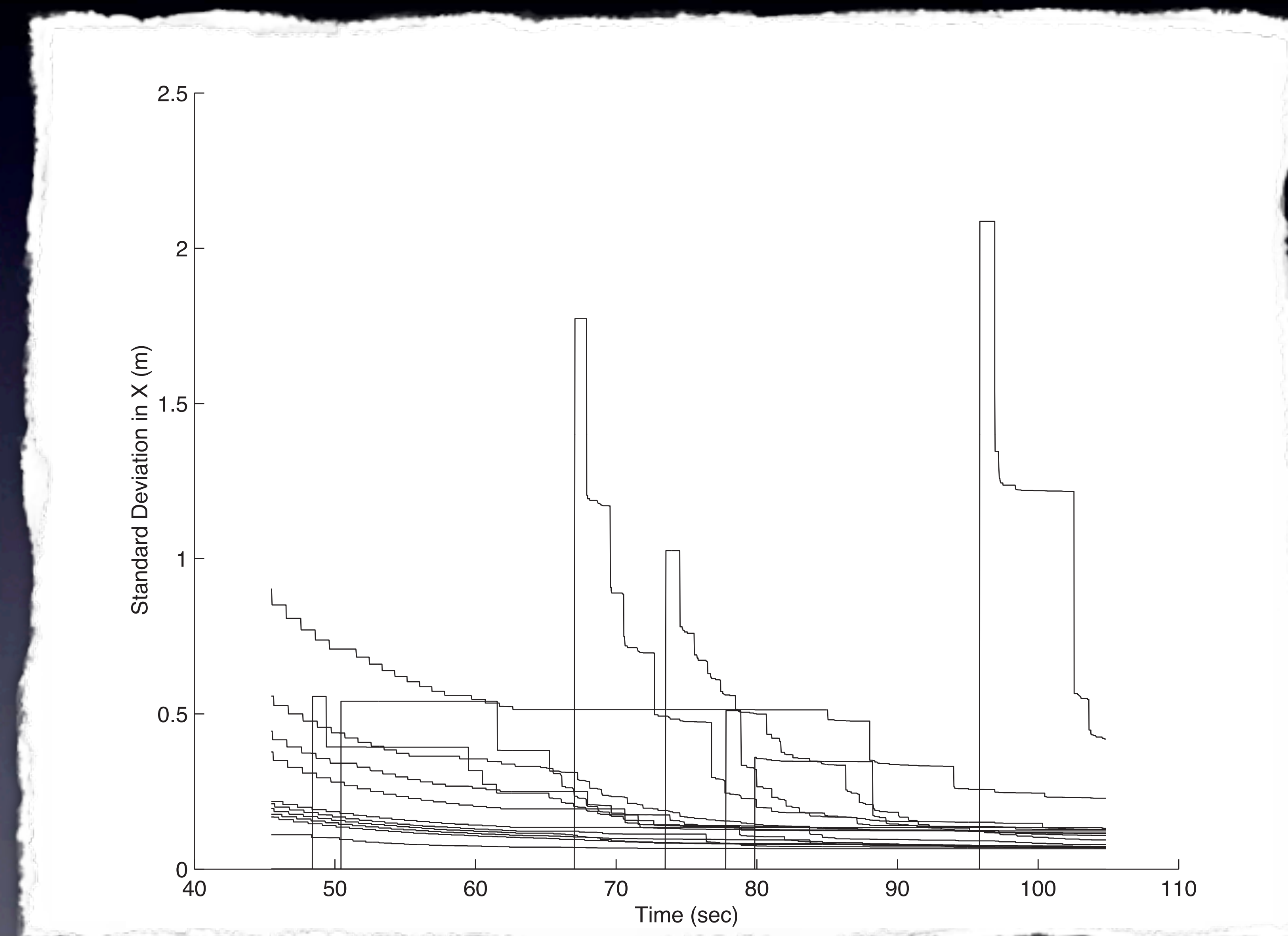


SLAM Convergence

- An observation in a neighborhood acts like a displacement to a spring system such that it's effect is great in the immediate neighborhood 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 localization accuracy achievable with an a priori map.

Monotonic Convergence

- When a new landmark is initialized, 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 (Linearization, Monte Carlo)

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
  - Compute covariance by linearisation

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

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 uk and has covariance Uk

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

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

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

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

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