Introduction to Estimation and Data fusion

Part III: Simultaneous Localisation and Mapping (SLAM)

Hugh Durrant-Whyte

Australian Centre for Field Robotics

The University of Sydney

hugh@acfr.usyd.edu.au

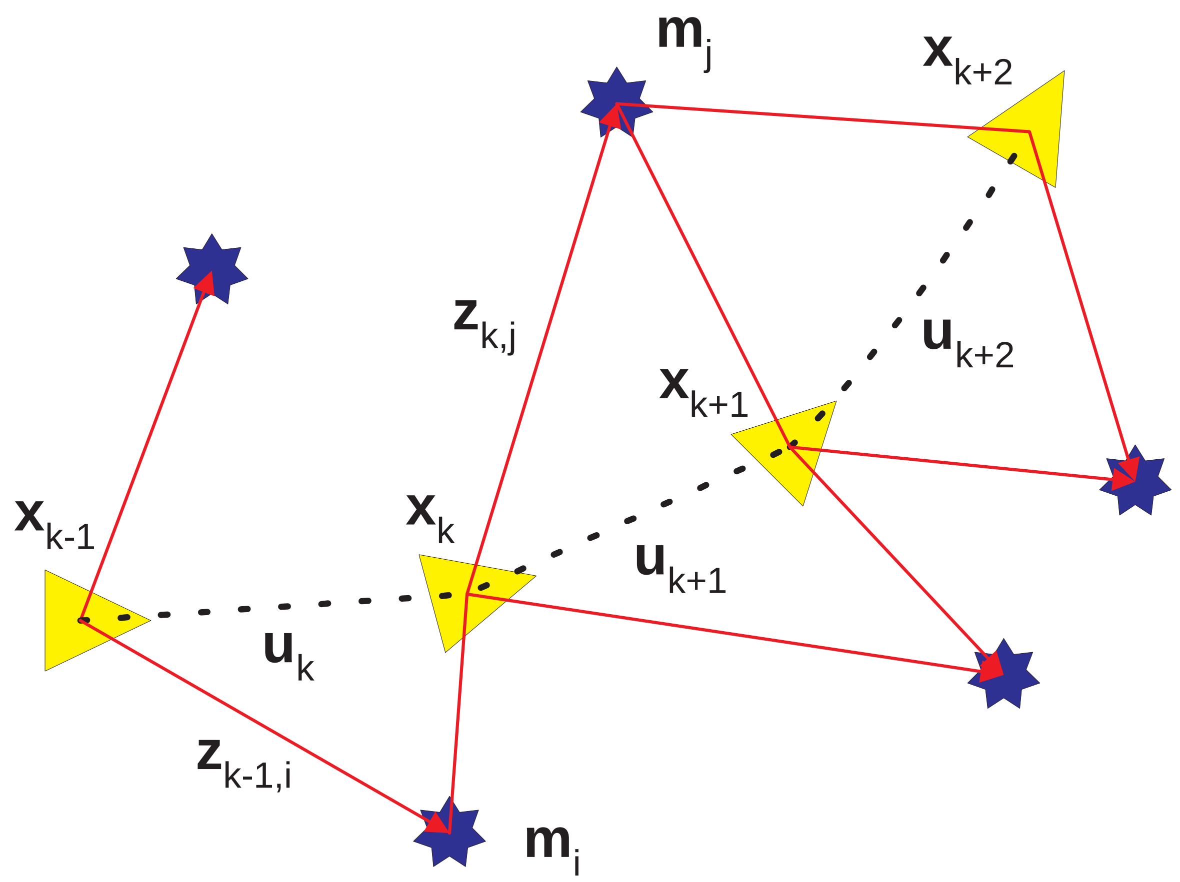
Introduction

- SLAM asks the following question:

  "Is it possible for an autonomous vehicle to start in 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 ?"

- A solution to the SLAM problem would allow robots to operate in an environment without *a priori* knowledge of a map and without access to independent position information.
- A solution to the SLAM problem would open up a vast range of potential applications for autonomous vehicles.
- A solution to the SLAM problem would make a robot truly autonomous
- Research over the last decade has shown that a solution to the SLAM problem is indeed possible.

Localisation and Mapping: Elements



Localisation and Mapping: General Definitions

- A discrete time index $k = 1, 2, \cdots$.
- $\mathbf{x}_k$: The true location of the vehicle at a discrete time k.
- $\mathbf{u}_k$: A control vector, assumed known, and applied at time $k-1$ to drive the vehicle from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ at time k.
- $\mathbf{m}_i$: The true location or parameterization of the ith landmark.
- $\mathbf{z}_{k,i}$: An observation (measurement) of the ith landmark taken from a location $\mathbf{x}_k$ at time k.
- $\mathbf{z}_k$: The (generic) observation (of one or more landmarks) taken at time k.

In addition, the following sets are also defined:

- The history of states: $\mathbf{X}^k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} = \{\mathbf{X}^{k-1}, \mathbf{x}_k\}$.
- The history of control inputs: $\mathbf{U}^k = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{U}^{k-1}, \mathbf{u}_k\}$.
- The set of all landmarks: $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_M\}$.
- The history of observations: $\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}^{k-1}, \mathbf{z}_k\}$.

The Localisation and Mapping Problem

- From knowledge of the observations $\mathbf{Z}^k$,
- Make inferences about the vehicle locations $\mathbf{X}^k$
- and/or inferences about the landmark locations $\mathbf{m}$.

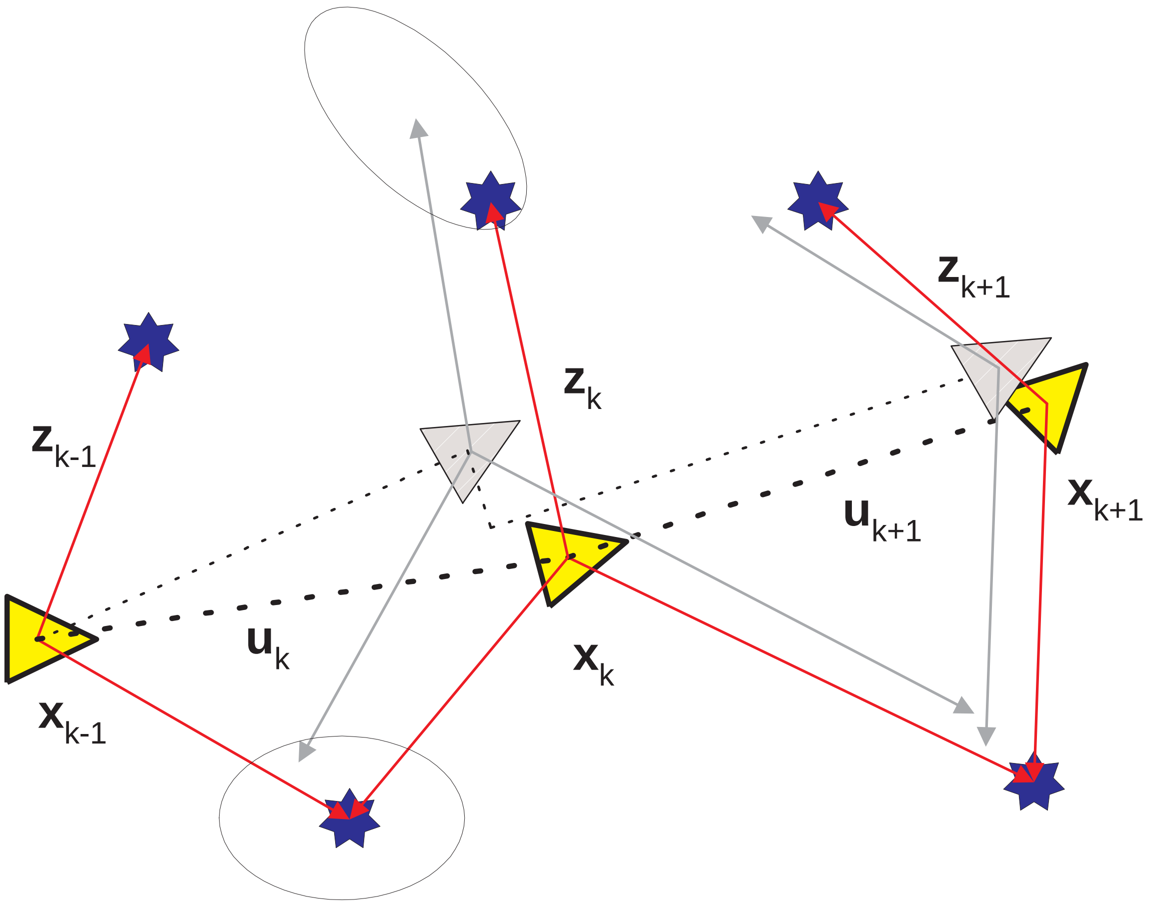
- Prior knowledge (a map) can be incorporated.
- Independent knowledge (inertial/GPS, for example) may also be used.

The Localisation Problem

- A map $\mathbf{m}$ is known *a priori*.
- The map may be a geometric map, a map of landmarks, a map of occupancy

- From a sequence of control actions $\mathbf{U}^k$
- Make inferences about the unknown vehicle locations $\mathbf{X}^k$

The Localisation Problem

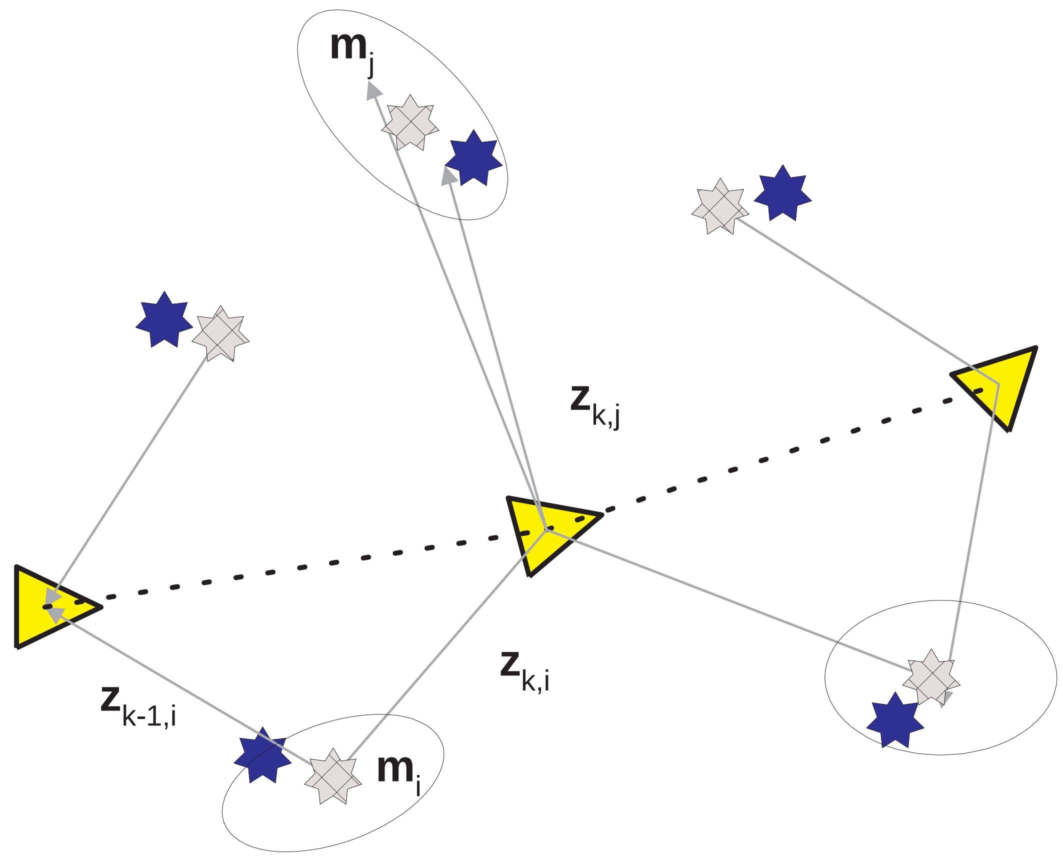


The Mapping Problem

- The vehicle locations $\mathbf{X}^k$ are provided (by some independent means).

- Make inferences about (build) the map $\mathbf{m}$
- The map may be a geometric map, a map of landmarks, a map of occupancy.

The Mapping Problem



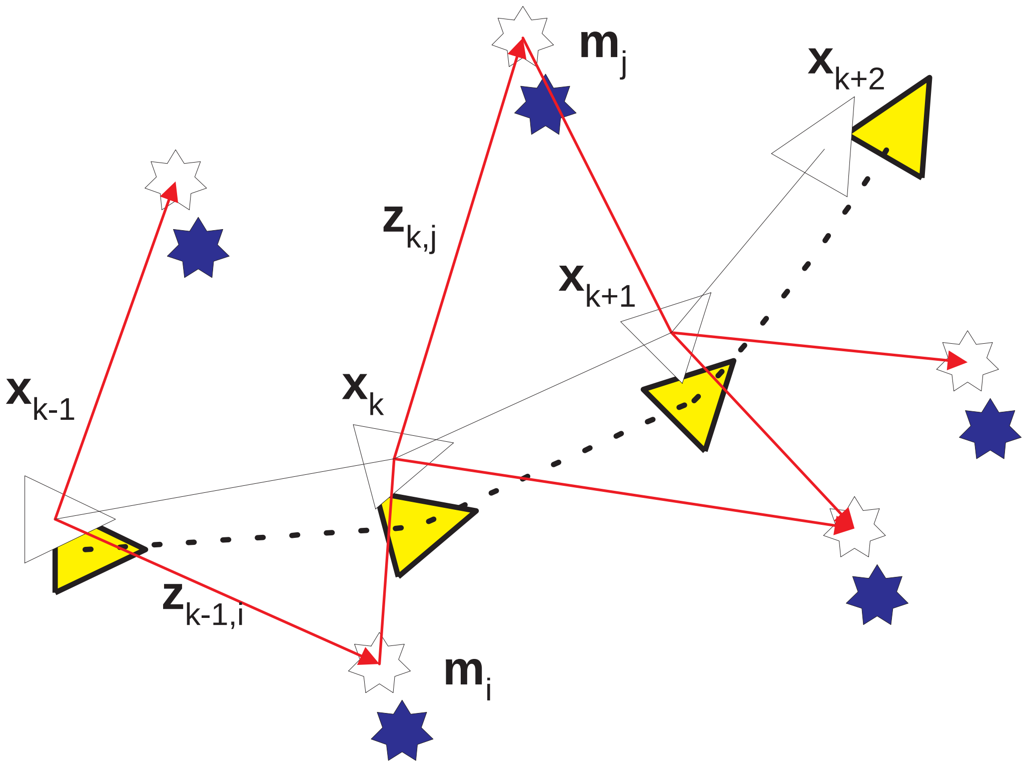
The Simultaneous Localisation and Mapping Problem

- No information about $\mathbf{m}$ is provided
- The initial location $\mathbf{x}_0$ is assumed known (the origin)
- The sequence of control actions $\mathbf{U}_k$ is given

- Build the Map $\mathbf{m}$
- At the same time inferences about the locations of the vehicle $\mathbf{X}^k$

- Recognise that the two inference problems are coupled.

The Simultaneous Localisation and Mapping Problem



The Simultaneous Localisation and Mapping Problem

- At the heart of the SLAM problem is the recognition that localisation and mapping are coupled problems.
- Fundamentally, this is because there is a single measurement from which two quantities are to be inferred.
- A solution can only be obtained if the mapping and localisation process are considered together.

Models of Sensors, Vehicles, Processes and Uncertainty

- Model sensors in the form of a likelihood $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$
- Model platform motion in terms of the conditional probability $P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k)$
- Recursively estimate the joint posterior $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0)$.

Sensor and Motion Models

- Observation model describes the probability of making an observation $\mathbf{z}_k$ when the true state of the world is $\{\mathbf{x}_k, \mathbf{m}\}$

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}).$$

- The observation model also has an interpretation as a likelihood function: the knowledge gained on $\{\mathbf{x}_k, \mathbf{m}\}$ *after* making the observation $\mathbf{z}_k$:

$$\Lambda(\mathbf{x}_k, \mathbf{m}) \stackrel{\triangle}{=} P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}).$$

- It is reasonable to assume conditional independence:

$$P(\mathbf{Z}^k \mid \mathbf{X}^k, \mathbf{m}) = \prod_{i=1}^{k} P(\mathbf{z}_i \mid \mathbf{X}^k, \mathbf{m}) = \prod_{i=1}^{k} P(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{m}).$$

Observation Update Step (Bayes Theorem)

- Expand joint distribution in terms of the state

$$\begin{aligned} P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) &= P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{z}_k, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\ &= P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{z}_k \mid \mathbf{Z}^{k-1} \mathbf{U}^k) \end{aligned}$$

- and the observation

$$\begin{aligned} P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) &= P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\ &= P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \end{aligned}$$

- Rearranging:

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) = \frac{P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0)}{P(\mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k)}.$$

Time Update Step

- Assume vehicle model is Markov:

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) = P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{X}^{k-2}, \mathbf{U}^{k-1}, \mathbf{m})$$

- Then (Total Probability Theorem)

$$\begin{aligned}
P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) &= \int P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\
&= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{m}, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\
&= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1}
\end{aligned}$$

Complete Recursive Calculation

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) = K.P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1}.$$

Kalman Filter Solutions to the SLAM Problem

Augmented State Model

- Vehicle model:

$$\mathbf{x}_v(k) = [x(k), y(k), \phi(k)]^T, \qquad \mathbf{u}(k) = [\omega(k), \gamma(k)]^T$$

- Landmark model

$$\mathbf{m}_i = [x_i, y_i]^T$$

- The augmented state model:

$$\mathbf{x}(k) \stackrel{\triangle}{=} = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_M \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\mathbf{x}_v(k-1), \mathbf{u}(k)\right) \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_M \end{bmatrix} + \begin{bmatrix} \mathbf{q}_v(k) \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

- Landmarks are assumed stationary

Estimation Process

- Observation model; relative observation of range and bearing

$$\mathbf{z}_i(k) = \begin{bmatrix} z_r^i(k) \\ z_\theta^i(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x(k))^2 + (y_i - y(k))^2} \\ \arctan\left(\frac{y_i - y(k)}{x_i - x(k)}\right) - \phi(k) \end{bmatrix} + \begin{bmatrix} r_r^i(k) \\ r_\theta^i(k) \end{bmatrix},$$

- In principle, estimation can now proceed in the same manner as a conventional EKF.
- Substantial computational advantage can be obtained by exploiting the structure of the process and observation models
- We now focus on the behaviour of the covariance matrix

Covariance Analysis

- The Covariance (in the EKF) tells us all we need to know about the errors involved in the SLAM process.
- Recall the recursion:

$$\begin{aligned} \mathbf{P}(k \mid k-1) &= \nabla \mathbf{f}_{\mathbf{x}}(k)\mathbf{P}(k-1 \mid k-1)\nabla^T \mathbf{f}_{\mathbf{x}}(k) + \mathbf{Q}(k) \\ \mathbf{P}(k \mid k) &= \mathbf{P}(k \mid k-1) + \mathbf{W}_i(k)\mathbf{S}_i(k)\mathbf{W}_i^T(k) \end{aligned}$$

- Where

$$\begin{aligned} \mathbf{S}_i(k) &= \nabla \mathbf{h}_{\mathbf{x},\mathbf{m_i}}(k)\mathbf{P}(k \mid k-1)\nabla^T \mathbf{h}_{\mathbf{x},\mathbf{m_i}}(k) + \mathbf{R}_i(k) \\ \mathbf{W}_i(k) &= \mathbf{P}(k \mid k-1)\nabla^T \mathbf{h}_{\mathbf{x},\mathbf{m_i}}(k)\mathbf{S}_i^{-1}(k) \end{aligned}$$

- and consider the form the matrix:

$$\mathbf{P}(i \mid j) = \begin{bmatrix} \mathbf{P}_{vv}(i \mid j) & \mathbf{P}_{vm}(i \mid j) \\ \mathbf{P}_{vm}^T(i \mid j) & \mathbf{P}_{mm}(i \mid j) \end{bmatrix}$$

Key Result I

The determinant of any sub-matrix of the map covariance matrix decreases monotonically as successive observations are made.

- with all square matrices *psd*,

$$\begin{aligned} \det \mathbf{P}(k \mid k) &= \det \left[\mathbf{P}(k \mid k-1) - \mathbf{W}_i(k)\mathbf{S}_i(k)\mathbf{W}_i^T(k)\right] \\ &\leq \det \mathbf{P}(k \mid k-1) \end{aligned}$$

- and noting

$$\mathbf{P}_{mm}(k \mid k-1) = \mathbf{P}_{mm}(k-1 \mid k-1)$$

- implies

$$\det \mathbf{P}_{mm}(k \mid k) \leq \det \mathbf{P}_{mm}(k-1 \mid k-1)$$

- and also for any sub-matrices of $\mathbf{P}_{mm}(k \mid k)$

Interpretation of Key Result I

- The determinant is a measure of volume,
- in this case measures the compactness of the Gaussian density function associated with the covariance matrix,
- is strictly proportional to the Shannon information associated with this density.

- As successive observations are made, map information increases monotonically.
- The correlations between landmark locations increase
- In effect, knowledge of the relative location of landmarks increases.

Key Result II

In the limit as successive observations are made, the errors in estimated landmark location become fully correlated.

- Lower limit of reduction in the determinant of the map covariance matrix:

$$\lim_{k\to\infty} [\det \mathbf{P}_{mm}(k \mid k)] = \mathbf{0}$$

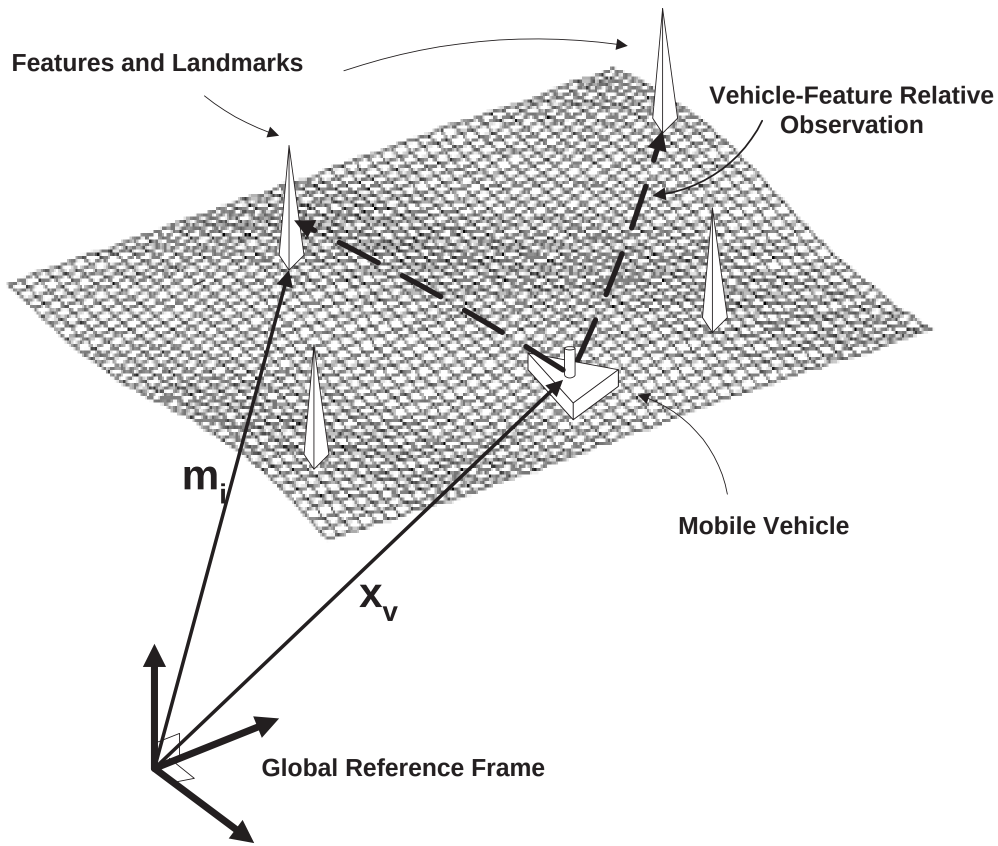
- True also for any sub-map

- The interpretation is that knowledge of the relative location of landmarks increases and, in the limit, becomes exact.

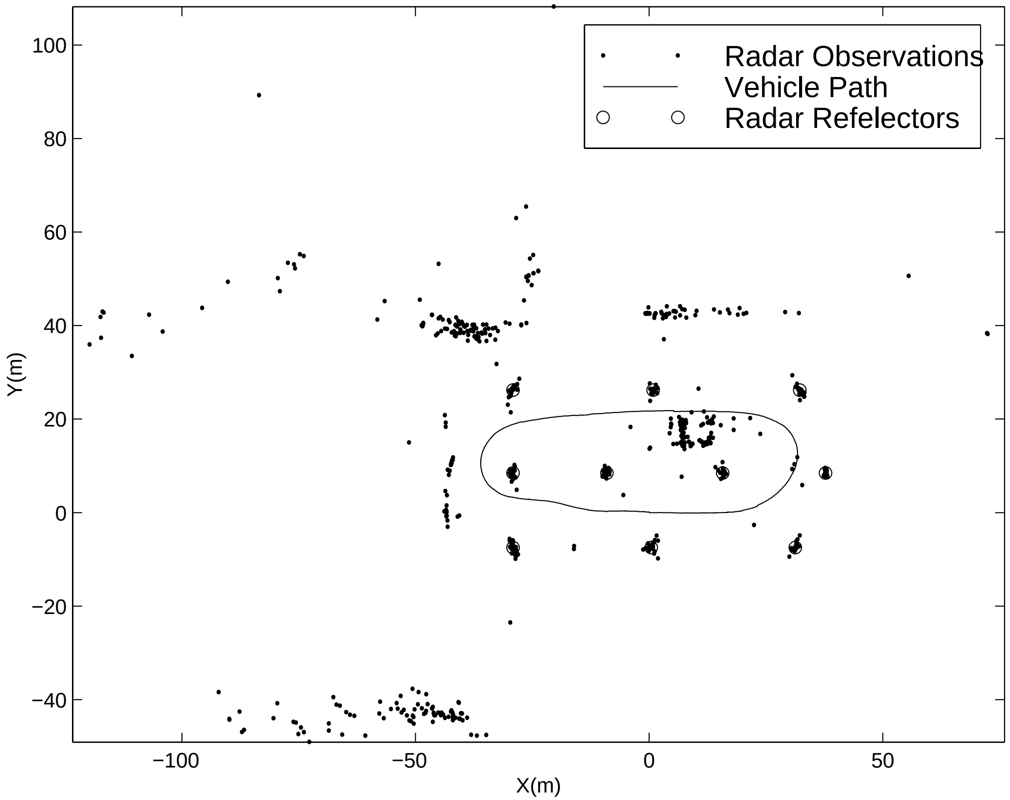
Result III

- In the limit, the absolute location of the landmark map is bounded only by the initial vehicle uncertainty $\mathbf{P}_{vv}(0 \mid 0)$.
- The Estimated location of the platform itself is therefore also bounded.

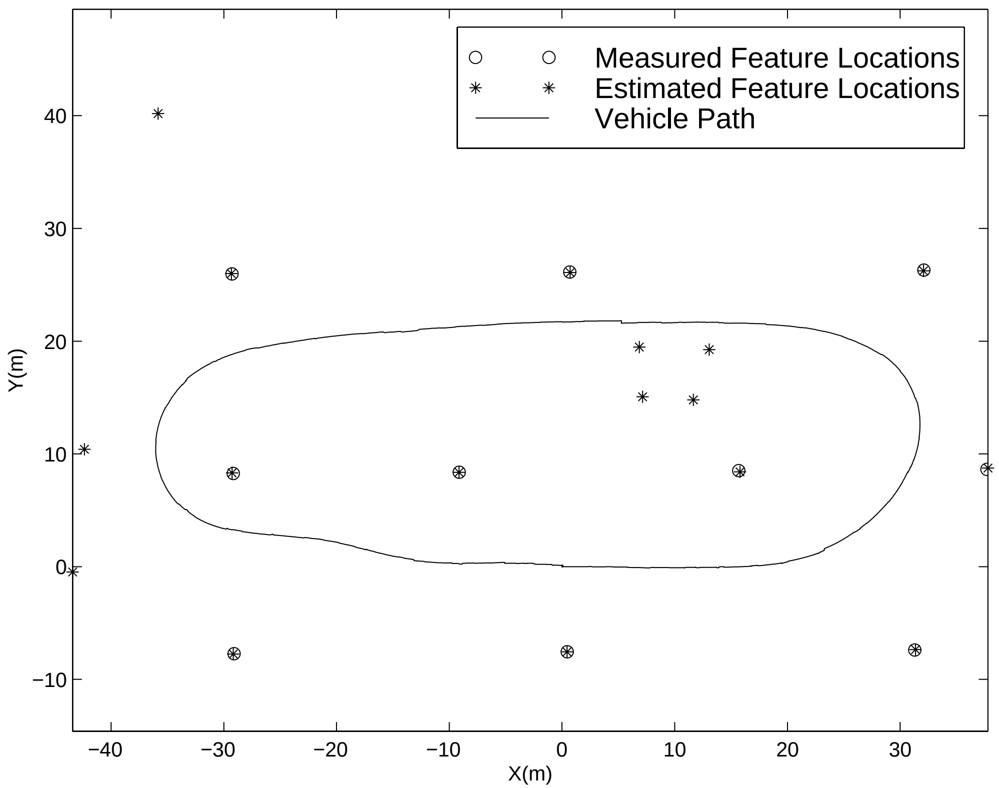
SLAM Structure



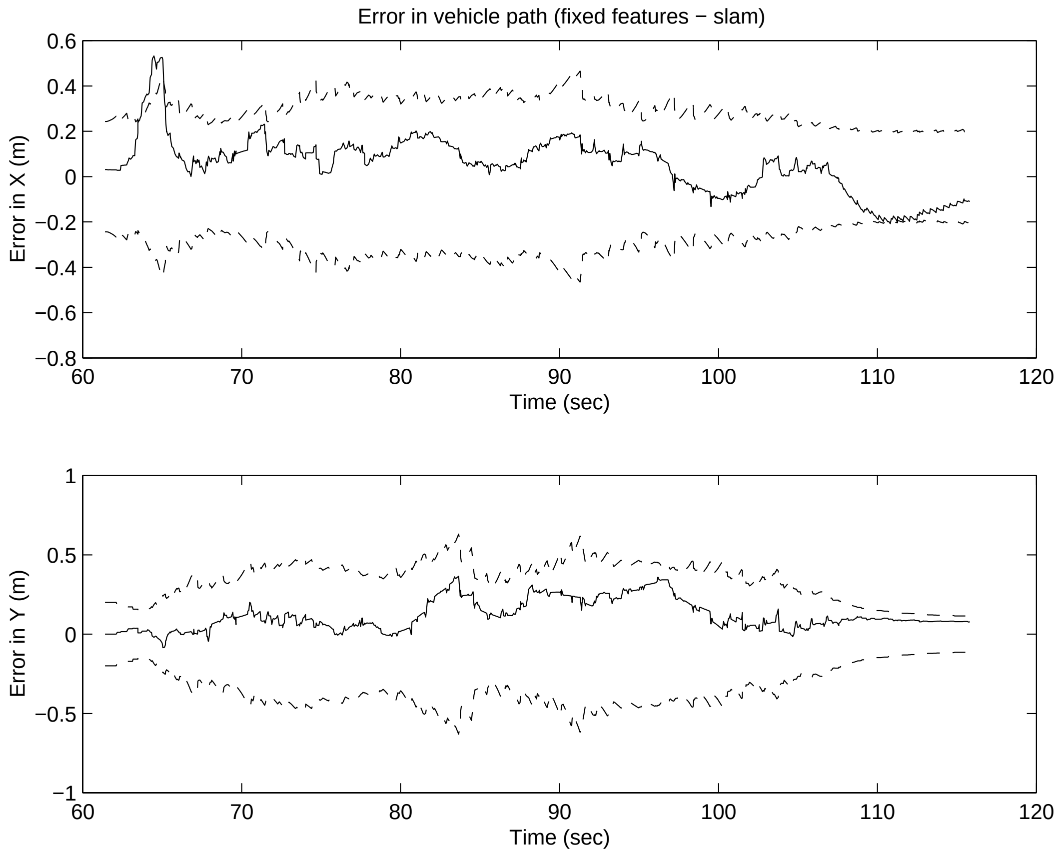
Characteristic Results: Raw Data



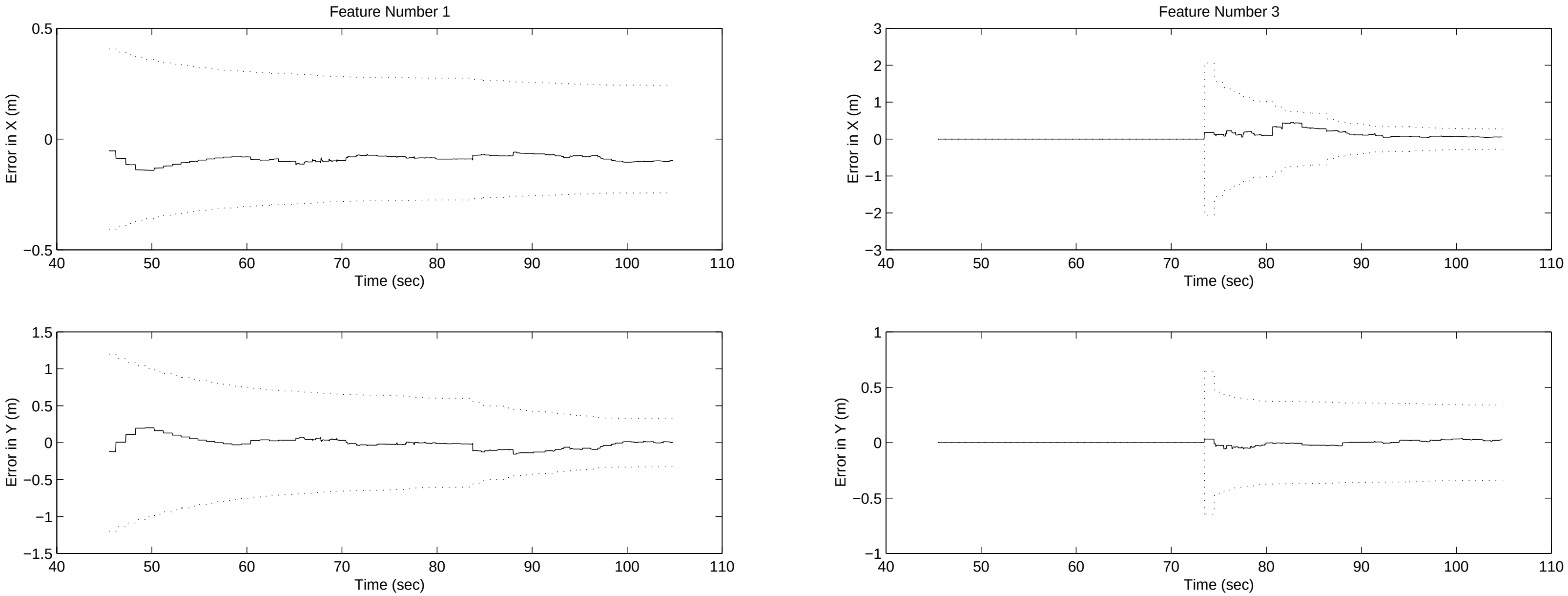
Characteristic Results: Vehicle Path and Landmark Locations



Characteristic Results: Vehicle Position Errors

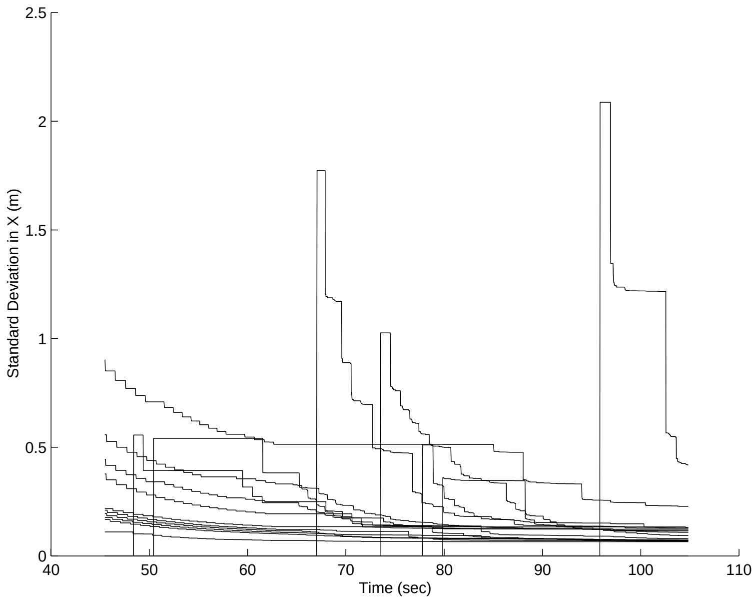


Characteristic Results: Example Land Mark Errors

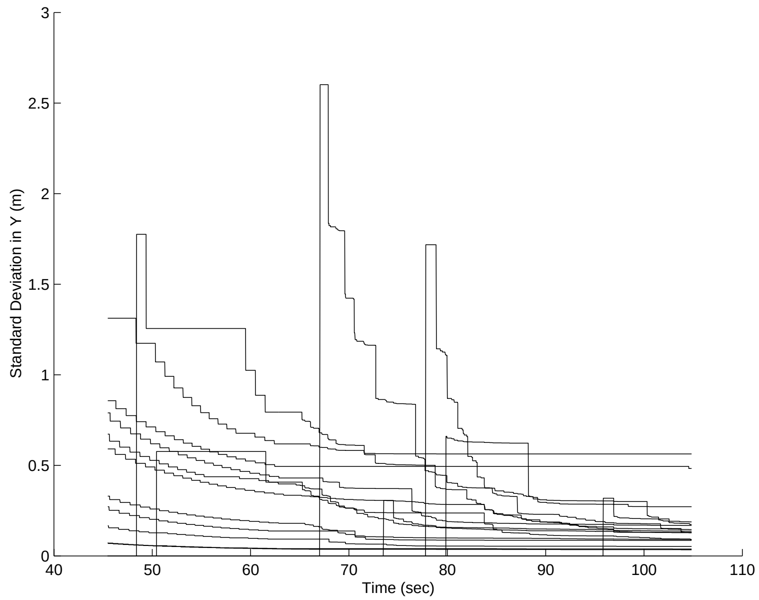


(a) (b)

Characteristic Results: All Land Mark Errors



(a)



(b)

Closed Form Results I

- Possible to get a closed-form solution to the basic 1d linear problem which provides some insight into the nature of errors in the map and the rates of convergence.
- Simple process model:

$$\begin{aligned}\dot{x}(t) &= x(t) + w \\ \dot{m}_i &= 0, \quad i = 1, \cdots, m_M \\ \mathbf{x}(t) &= [x(t), m_1, m_2, \cdots, m_M]^T\end{aligned}$$

- and observation model

$$z_i(t) = m_i - x(t) + v$$

- with $q = \mathrm{E}\{w^2\}$ and $r = \mathrm{E}\{v^2\}$
- In Riccati Equation of the form:

$$\dot{\mathbf{P}}(t) = 2\mathbf{P}(t) + \mathbf{G}q\mathbf{G}^T - \mathbf{P}^T(t)\mathbf{H}^T\mathbf{H}\mathbf{P}(t)/r$$

- Gives:

Closed Form Results II

$$\mathbf{P}(t) = \frac{1}{(\alpha+1)+(\alpha-1)e^{-2\alpha t}} \begin{bmatrix} q(1-e^{-2\alpha t}) + \frac{2q}{\alpha}(1-e^{-\alpha t})^2 & \cdots & \frac{q}{\alpha}(1-e^{-\alpha t})^2 & \cdots & \frac{q}{\alpha}(1-e^{-\alpha t})^2 & \cdots \\ \vdots & \ddots & \vdots & & \vdots & \\ \frac{q}{\alpha}(1-e^{-\alpha t})^2 & \cdots & \frac{r_i(I_T - r_i^{-1})}{(t+1)I_T} + \frac{q}{\alpha}(1+e^{-2\alpha t}) & \cdots & -\frac{1}{(t+1)I_T} + \frac{q}{\alpha}(1+e^{-2\alpha t}) & \cdots \\ \vdots & & \vdots & \ddots & \vdots & \\ \frac{q}{\alpha}(1-e^{-\alpha t})^2 & \cdots & -\frac{1}{(t+1)I_T} + \frac{q}{\alpha}(1+e^{-2\alpha t}) & \cdots & \frac{r_j(I_T - r_j^{-1})}{(t+1)I_T} + \frac{q}{\alpha}(1+e^{-2\alpha t}) & \cdots \\ \vdots & & \vdots & & \vdots & \end{bmatrix} \quad (1)$$

- where the characteristic equation of the system is

$$D(t) = (\alpha+1) + (\alpha-1)e^{-2\alpha t}$$

- and the total Fisher information available to the filter

$$\mathbf{I}_T = \sum_{i=1}^{n} r_i^{-1}$$

- the dominant time constant for the system.

$$\alpha = \sqrt{q\mathbf{I}_T}$$

A Brief History of the SLAM Problem I

- Initial work by Smith et al. and Durrant-Whyte established a statistical basis for describing geometric uncertainty and relationships between features or landmarks (1985-1986).
- At the same time Ayache and Faugeras, and Chatila and Laumond were undertaking early work in visual navigation of mobile robots using Kalman filter-type algorithms.
- Discussions on how to do the SLAM problem at ICRA'86 (Cheesman, Chatila, Crowley, DW) resulting soon after in the key paper by Smith, Self and Cheeseman.
- This paper 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.

A Brief History of the SLAM Problem II

- Work then focused on Kalman-filter based approaches to indoor vehicle navigation Especially:
  - Leonard/Durrant-Whyte, Sonar and data association.
  - Chatila et.al; visual navigation and mapping
  - Faugeras et. al. visual navigation/motion
- Most approaches to the problem involved decoupling localisation and mapping; especially Leonard, Rencken, Stevens, (1990-1994)
- In 1991/92 "Chicken and Egg" paper identified some of the key issues in solving the SLAM problem.
- A realisation that the two problems must be solved together (around 1991, then 1993-94).

A Brief History of the SLAM Problem III

- For *me* the big break-through was understanding and then demonstrating that the SLAM problem would converge if considered as a whole (Csorba 1995).
- The SLAM acronym coined in 1995 (ISRR).
- Generating proofs of convergence and some of the first demonstrations of the SLAM algorithm, Especially:
  - Dissanayake's work with indoor vehicles and lasers (1996-1997)
  - Leonard/Feder work with sonar modeling, data association and CML (1996-1999)
  - Dissanayake, Newman et.al. outdoor radar and sub-sea SLAM and final convergence proofs (1997)
  - Independently Thrun's indoor vehicle localisation and mapping work (1997-1999).

Some History of the SLAM Problem IV

- ISRR 1999 session on navigation/SLAM was a key event (Leonard, Thrun, DW).
- ICRA 2000 SLAM workshop also got many other researchers interested in the problem.
- Key problems identified and then subsequent work on:
  - Computationally efficient implementations (Leonard, Nebot, Newman, Tardos)
  - Large-scale implementations (Nebot, Dissa)
  - Data Association (Castellanos, Tardos, Leonard)
  - Understanding the applicability of probabilistic methods (Thrun et.al, DW et.al)
  - Multiple Vehicle SLAM (Nettleton, Thrun, Williams)
  - Implementations indoor, on land, air and sub-sea.
- By ICRA 2002, many new methods and ideas with groups working at ANU, CMU, EPTL, KTH, MIT, Oxford, Sydney, Zaragoza
- Most of which you will now hear about ...

Implementation of the Simultaneous Localization and Mapping (SLAM) Algorithm

Eduardo Nebot

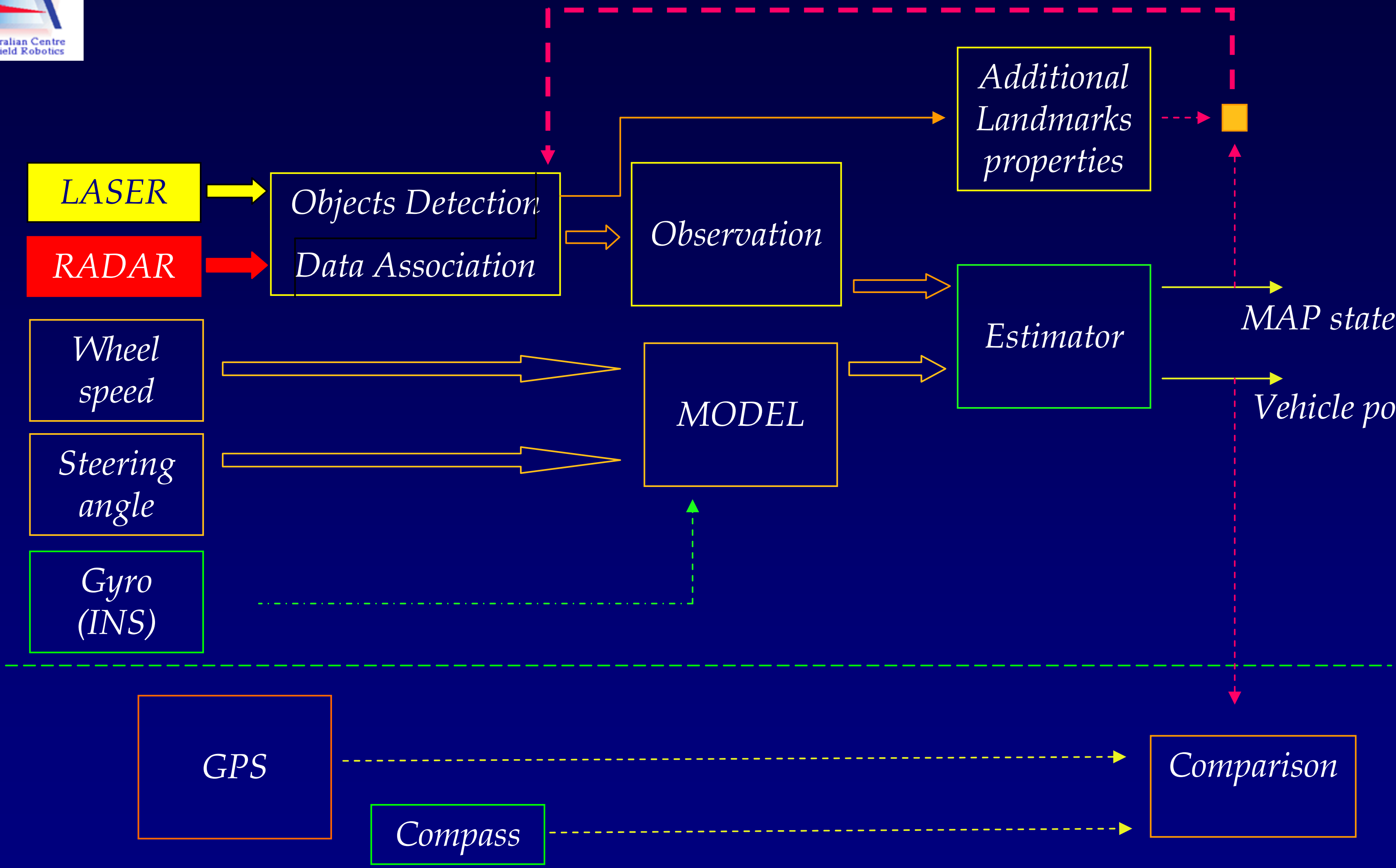
Australian Centre for Field Robotics
nebot@acfr.usyd.edu.au
http://acfr.usyd.edu.au/homepages/academic/enebot

Overview

- The SLAM problem
- Models
- Standard EKF Implementation
- Simplifications in the Prediction and Update Stages
- Optimal Implementation of EKF SLAM
- Relative Map Representation
- Sub-optimal Filters
- Exploration
- Labs

Basic Principle of SLAM

System Description



Kalman Implementation

- **Prediction Stage**

$$x(k) = f(x(k-1), u(k-1), k) + v(k-1)$$

$$P(k/k-1) = \nabla f_x(k) P(k-1/k-1) \nabla f_x^T(k) + Q$$

Can also model noise in the inputs

- **This step is performed each time a set of inputs is available (High Frequency information).**
- **The uncertainty in the pose of the vehicle will grow according to the uncertainty of the model**

Kalman Implementation

- **Update Stage**
  - The update stage is performed once a feature is associated to a known feature.

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

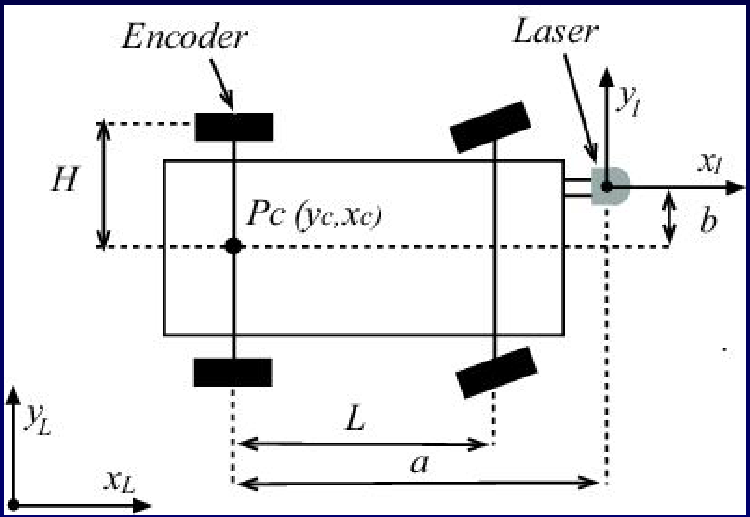
$$P(k/k) = P(k/k-1) - W(k)S(k)W^T(k)$$

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

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

$$\tilde{x}(k/k) = \hat{x}(k/k-1) + W(k)\mu(k)$$

Vehicle Model



$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_c \cos(\phi) \\ v_c \sin(\phi) \\ v_c \tan(\alpha) \end{bmatrix}$$

We translate the centre to the GPS antenna position (top of the Laser)

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \begin{bmatrix} x_c + a\cos\phi - b\sin\phi \\ y_c + a\sin\phi + b\cos\phi \end{bmatrix}$$

The velocity is measured with and encoder located in the back left wheel. The velocity Vc is then:

$$v_c = \frac{v_e}{\left(1 - \tan(\alpha)\frac{H}{L}\right)}$$

Vehicle Model

Continuous Model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_c \cdot \cos(\phi) - \frac{v_c}{L} \cdot \left(a \cdot \sin(\phi) + b \cdot \cos(\phi)\right) \cdot \tan(\alpha) \\ v_c \cdot \sin(\phi) + \frac{v_c}{L} \cdot \left(a \cdot \cos(\phi) - b \cdot \sin(\phi)\right) \cdot \tan(\alpha) \\ \frac{v_c}{L} \cdot \tan(\alpha) \end{bmatrix}$$

Discrete Model

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = f(x,u) = \begin{bmatrix} x(k) + \Delta T(v_c \cos(\phi) - \frac{v_c}{L} \tan(\phi)(a \sin(\phi) + b \cos(\phi))) \\ y(k) + \Delta T(v_c \sin(\phi) + \frac{v_c}{L} \tan(\phi)(a \cos(\phi) + -b \sin(\phi))) \\ \phi(k) + \Delta T \frac{v_c}{L} \tan(\alpha) \end{bmatrix}$$

Vehicle Model

Discrete Model

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = f(x,u) = \begin{bmatrix} x(k) + \Delta T(v_c\cos(\phi) - \frac{v_c}{L}\tan(\phi)(a\sin(\phi) + b\cos(\phi))) \\ y(k) + \Delta T(v_c\sin(\phi) + \frac{v_c}{L}\tan(\phi)(a\cos(\phi) + -b\sin(\phi))) \\ \phi(k) + \Delta T\frac{v_c}{L}\tan(\alpha) \end{bmatrix}$$

Jacobian

$$\frac{\partial f}{\partial X} = \begin{bmatrix} 1 & 0 & -\Delta T(v_c\sin(\phi) + \frac{v_c}{L}\tan\alpha(a\cos(\phi) - b\sin(\phi))) \\ 0 & 1 & \Delta T(v_c\cos(\phi) - \frac{v_c}{L}\tan\alpha(a\sin(\phi) + b\cos(\phi))) \\ 0 & 0 & 1 \end{bmatrix}$$

Observation Model

- **Jacobians**

$$\begin{bmatrix} z_r \\ z_\beta \end{bmatrix} = h(x) = \begin{bmatrix} \sqrt{(x_L - x_v)^2 + (y_L - y_v)^2} \\ \operatorname{atan}\left(\dfrac{(y_L - y_v)}{(x_L - x_v)} \right) - \phi + \dfrac{\pi}{2} \end{bmatrix}$$

$$\frac{\partial h}{\partial X} = \begin{bmatrix} \dfrac{\partial h_r}{\partial x} \\ \dfrac{\partial h_\beta}{\partial x} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial z_r}{\partial (x_v, y_v, \phi_v, \{x_L, y_L\})} \\ \dfrac{\partial z_\beta}{\partial (x_v, y_v, \phi_v, \{x_L, y_L\})} \end{bmatrix}$$

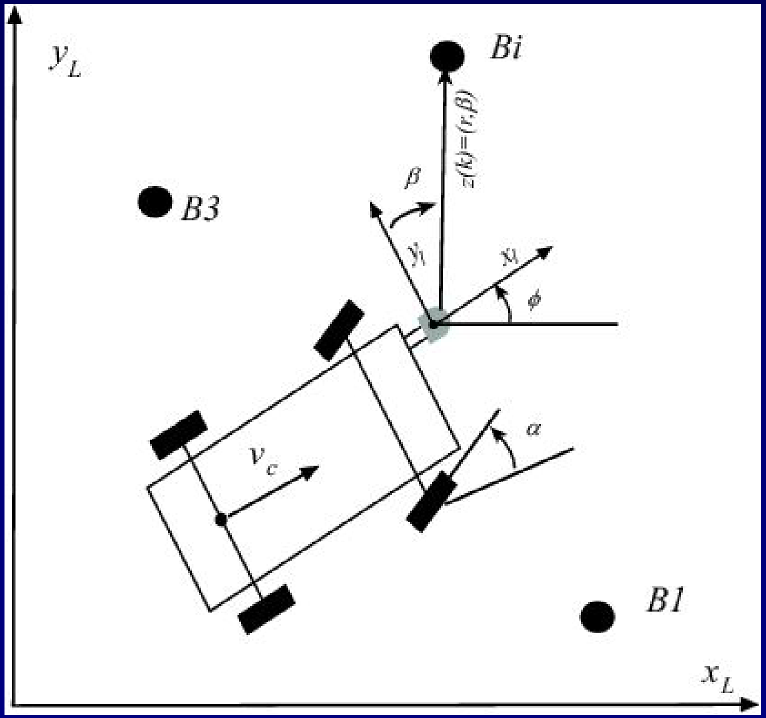
with

$$\frac{\partial h_r}{\partial X} = \frac{1}{\Delta}\left[-\Delta x, -\Delta y, 0, 0, 0, \ldots. \Delta x, \Delta y, 0, 0 \ldots. 0\right]$$

$$\frac{\partial h_\beta}{\partial X} = \left[\frac{\Delta y}{\Delta^2}, -\frac{\Delta x}{\Delta^2}, -1, 0, 0, \ldots., -\frac{\Delta y}{\Delta^2}, \frac{\Delta x}{\Delta^2}, 0, 0 \ldots 0\right]$$

$$\Delta x = (x_L - x_v) \qquad \Delta y = (y_L - y_v)$$

$$\Delta = \sqrt{\Delta x^2 + \Delta y^2}$$



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
  - Not all the Feature need to be checked since the vehicle pose is known with a given uncertainty (Selective Search)

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

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

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

New Features

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

- **We do not want to incorporate an spurious observation as a new feature**
  - It will not be observed again and will consume computational time and memory
  - The features are assumed to be static. We don 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
    - Absolutely necessary for Low frequency external sensors (ratio between vehicle velocity and feature rate information)
  - **Disadvantages:**
    - The implementation is more complicated

Incorporation of New Features

- We have the vehicle sates and previous map

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

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

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

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

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

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

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

Analytical Approach

- **Assume the following System**

$$\begin{bmatrix} X_v \\ X_m \\ X_n \end{bmatrix} = \begin{bmatrix} X_v \\ X_m \\ h(X_v, Z) \end{bmatrix}$$

$$h(X,Z) = \begin{bmatrix} x_v + \cos(\varphi_v + \alpha - \pi / 2 \\ y_v + \sin(\varphi_v + \alpha - \pi / 2 \end{bmatrix}$$

We need P1

$$P_1 = J\, P_0\, J^T + J_z\, R\, J_z^T$$

$$J = (\partial F / \partial X_v, X_m)_{X_v(k,k), X_m(k,k), Z(k)}$$

$$J_z = (\partial F / \partial Z)_{X_v(k,k), X_m(k,k), Z(k)}$$

$$J = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \xi & 0 & 0 \end{bmatrix} \quad , \quad J_z = \begin{bmatrix} 0 \\ 0 \\ \eta \end{bmatrix}$$

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

$$J_z\ P_0\ J_z^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & B \end{bmatrix}$$

$$B = \eta R \eta$$

$$\eta = (\partial h(X_v, Z) / \partial z)_{x_v(k,k), Z(k)}$$

Analytical Approach

- **The analytical form of the covariance matrix for the new feature:**

$$P_1 = \begin{bmatrix} P^0_{v,v} & P^0_{v,m} & P^0_{v,v}\,\xi^T \\ P^0_{m,v} & P^0_{m,m} & P^0_{m,v}\,\xi^T \\ \xi\, P^0_{v,v} & \xi\, P^0_{v,m} & \xi\, P^0_{v,v}\,\xi^T + B \end{bmatrix}$$

$$\xi = \partial h / \partial (x_v, y_v, \varphi_v) = \begin{bmatrix} 1 & 0 & \cos(\varphi_v + \alpha - \pi/2) \\ 0 & 1 & \sin(\varphi_v + \alpha - \pi/2) \end{bmatrix}_{(\varphi_v, r, \alpha) = (\varphi_v(k/k), r(k), \alpha(k))}$$

$$B = \eta R \eta$$

$$\eta = (\partial h(X_v, Z) / \partial z)_{x_v(k,k), Z(k)}$$

$$\eta = (\partial h(X_v, Z)/\partial z) = \partial h / \partial (r, \alpha) = \begin{bmatrix} \cos(\varphi_v + \alpha - \pi/2) & -r\sin(\varphi_v + \alpha - \pi/2) \\ \sin(\varphi_v + \alpha - \pi/2) & r\cos(\varphi_v + \alpha - \pi/2) \end{bmatrix}$$

Now We know

- Structure of the Problem
- Models
- Kalman Filter Equations
- Data Association
- Feature Incorporation

Problem Solved ?

Computational Requirements

$$X = \begin{bmatrix} X_v \\ X_L \end{bmatrix}$$

$$X_v = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}, \qquad X_L = \begin{bmatrix} x_1 \\ y_1 \\ \dots \\ x_N \\ y_N \end{bmatrix}$$

$$J_1 \in R^{3x3}, \quad \varnothing \in R^{3xN}, \quad I \in R^{NxN}$$

$$\frac{\partial F}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_V} & \varnothing \\ \varnothing^T & I \end{bmatrix} = \begin{bmatrix} J_1 & \varnothing \\ \varnothing^T & I \end{bmatrix}$$

The computational requirements for each update will be proportional to $\rightarrow$ N^3

Simplifications (Prediction Stage)

- In most navigation system the dead reckoning information is available at high frequency.
- The full error covariance matrix is only needed when a new set of observation is available

$$\hat{X}(k+1) = F\left(\hat{X}(k)\right)$$

$$P(k+1,k) = J \cdot P(k,k) \cdot J^T + Q(k)$$

Simplifications (Prediction Stage)

Considering the zeros in the Jacobian matrices.

$$J \cdot P \cdot J^T + Q = \begin{bmatrix} J_1 & \varnothing \\ \varnothing^T & I \end{bmatrix} \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} J_1^T & \varnothing^T \\ \varnothing & I^T \end{bmatrix} + \begin{bmatrix} Q_V & \varnothing \\ \varnothing & \varnothing_2 \end{bmatrix}$$

$$J_1 \in R^{3x3}, \quad \varnothing \in R^{3x2N}, \quad I \in R^{2Nx2N}$$

$$P_{11} \in R^{3x3}, \quad P_{12} \in R^{3x2N}, \quad P_{21} = P_{12}^T, \quad P_{22} \in R^{2Nx2N}$$

$$J \cdot P = \begin{bmatrix} J_1 & \varnothing \\ \varnothing^T & I \end{bmatrix} \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ I \cdot P_{21} & I \cdot P_{22} \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

$$J \cdot P \cdot J^T = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} J_1^T & \varnothing^T \\ \varnothing & I \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} \cdot J_1^T & J_1 \cdot P_{12} \cdot I \\ P_{21} \cdot J_1^T & P_{22} \cdot I \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} \cdot J_1^T & J_1 \cdot P_{12} \\ \left(J_1 \cdot P_{12}\right)^T & P_{22} \end{bmatrix}$$

Simplifications (Prediction Stage)

$$J \cdot P = \begin{bmatrix} J_1 & \varnothing \\ \varnothing^T & I \end{bmatrix} \cdot \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ I \cdot P_{21} & I \cdot P_{22} \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

$$J \cdot P \cdot J^T = \begin{bmatrix} J_1 \cdot P_{11} & J_1 \cdot P_{12} \\ P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} J_1^T & \varnothing^T \\ \varnothing & I \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} \cdot J_1^T & J_1 \cdot P_{12} \cdot I \\ P_{21} \cdot J_1^T & P_{22} \cdot I \end{bmatrix} = \begin{bmatrix} J_1 \cdot P_{11} \cdot J_1^T & J_1 \cdot P_{12} \\ \left(J_1 \cdot P_{12}\right)^T & P_{22} \end{bmatrix}$$

$$P(k+2/k) = \begin{bmatrix} P_{11}\left(k+2,k\right) & G_1 \cdot P_{12}\left(k,k\right) \\ \left(G_1 \cdot P_{12}\left(k,k\right)\right)^T & P_{22}\left(k,k\right) \end{bmatrix}$$

with

$$G_1 = J_1\left(k+1\right) \cdot J_1\left(k\right)$$

$$P_{11}\left(k+2,k\right) = J_1\left(k+1\right) \cdot \left(J_1\left(k\right) \cdot P_{11}\left(k,k\right) \cdot J_1^T\left(k\right) + Q_{11}\left(k\right)\right) \cdot J_1^T (k -$$

Simplifications (Prediction Stage)

For n predictions:

$$P(k+n,k) = \begin{bmatrix} P_{11}(k+n,k+n) & G_1 \cdot P_{12}(k,k) \\ \left(G_1 \cdot P_{12}(k,k)\right)^T & P_{22}(k,k) \end{bmatrix}$$

with

$$G_1 = G_1(k,n) = \prod_{i=0}^{n-1} J_1(k+i) = J_1(k+n-1)\cdot \ldots \cdot J_1(k)$$

$$J_1 \in R^{3x3}, \quad \emptyset \in R^{3x2N}, \quad I \in R^{2Nx2N}$$
$$P_{11} \in R^{3x3}, \quad P_{12} \in R^{3x2N}, \quad P_{21} = P_{12}^T, \quad P_{22} \in R^{2Nx2N}$$

P_{11} is required at each step, P_{22} remains constant and P_{21} P_{12} are required only at the update stage

Simplification Update stage

The evaluation of the gain W requires P H'

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

Simplification Update stage

The Jacobian of the Observation matrix H will normally have a large number of zeros since very few landmarks will be observed at a given time:

$$H = H(k) = \left.\frac{\partial h}{\partial X}\right|_{X=X(k)} = \left[H_1, \varnothing_1, H_2, \varnothing_2\right] \in R^{2xM}, \quad M = (N+3)$$

$$H_1 = \left.\frac{\partial h}{\partial X_V}\right|_{X=X(k)} = \left.\frac{\partial h}{\partial(x, y, \phi)}\right|_{X=X(k)} \in R^{2x3}$$

$$H_2 = \left.\frac{\partial h}{\partial X_i}\right|_{X=X(k)} = \left.\frac{\partial h}{\partial(x_i, y_i)}\right|_{X=X(k)} \in R^{2x2}$$

$$\varnothing_1, \varnothing_2 = \text{null matrixs} \left(\frac{\partial h}{\partial X_j} = \varnothing \quad \forall j \neq i\right).$$

Simplification Update stage

The operations can be simplified with

$$P \cdot H^T = P_1 \cdot H_1^T + P_2 \cdot H_2^T$$

$$P_1 \in R^{Mx3} \quad , \quad P_2 \in R^{Mx2}$$

$$S = H \cdot P \cdot H^T + R \ \in R^{2*2}$$

$$W = P \cdot H^T \cdot S^{-1} \in R^{Mx2}$$

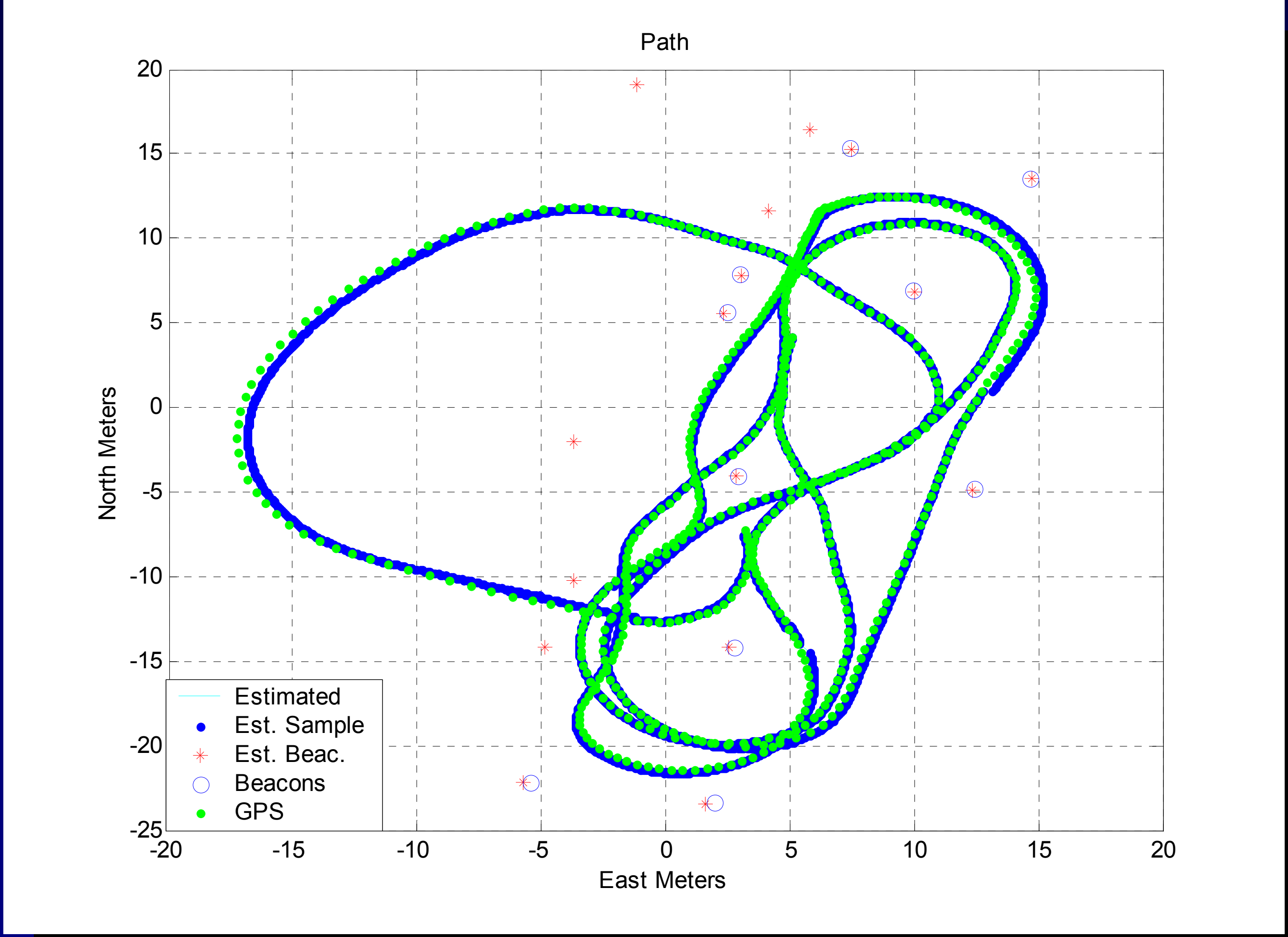
Still the main computational requirement is in the update of P (order 4 M*M)

$$P = P - W \cdot S \cdot W^T$$

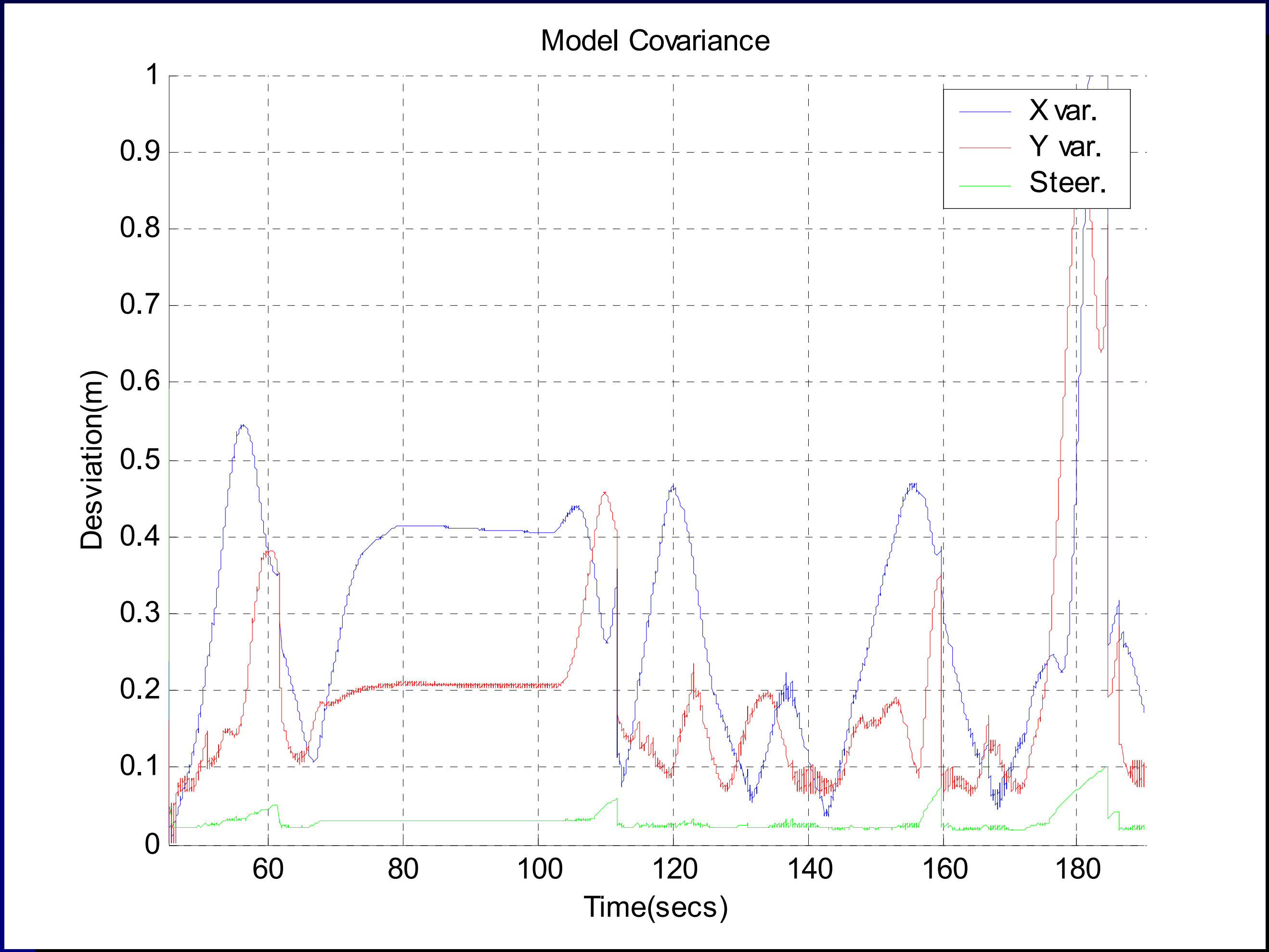
Experimental Results

Experimental Results

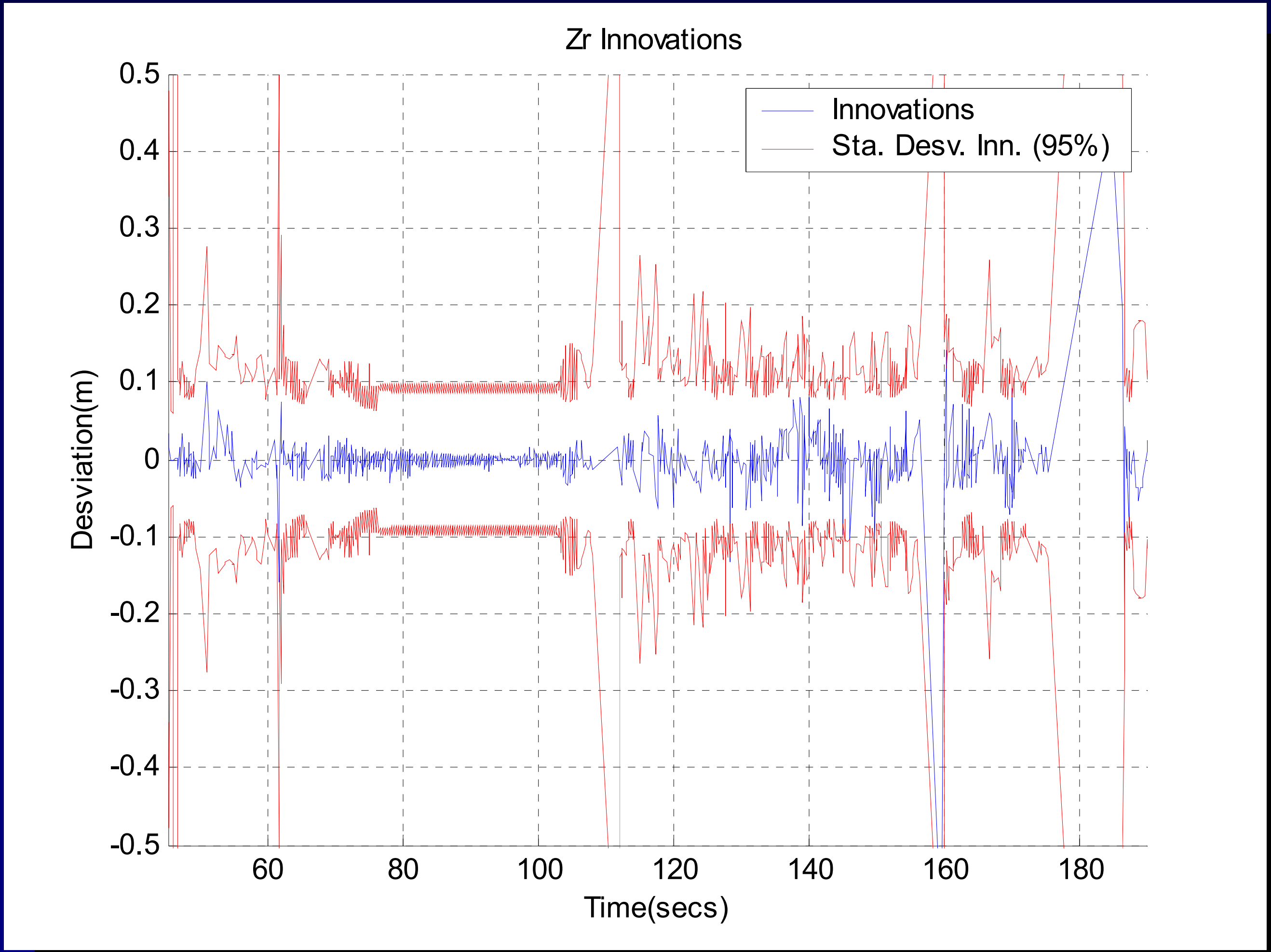
Estimated Trajectory using beacons



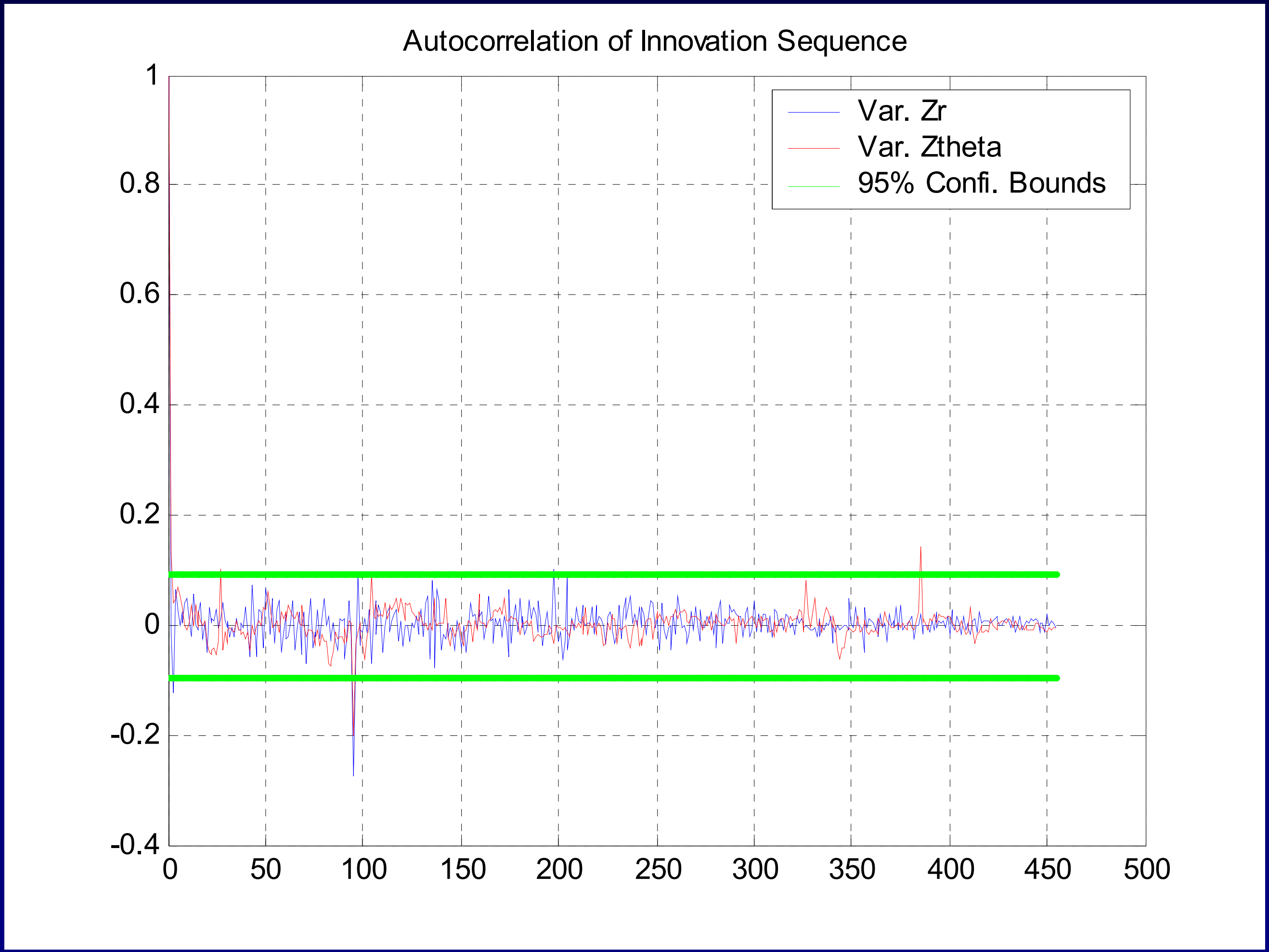
Estimated error covariance



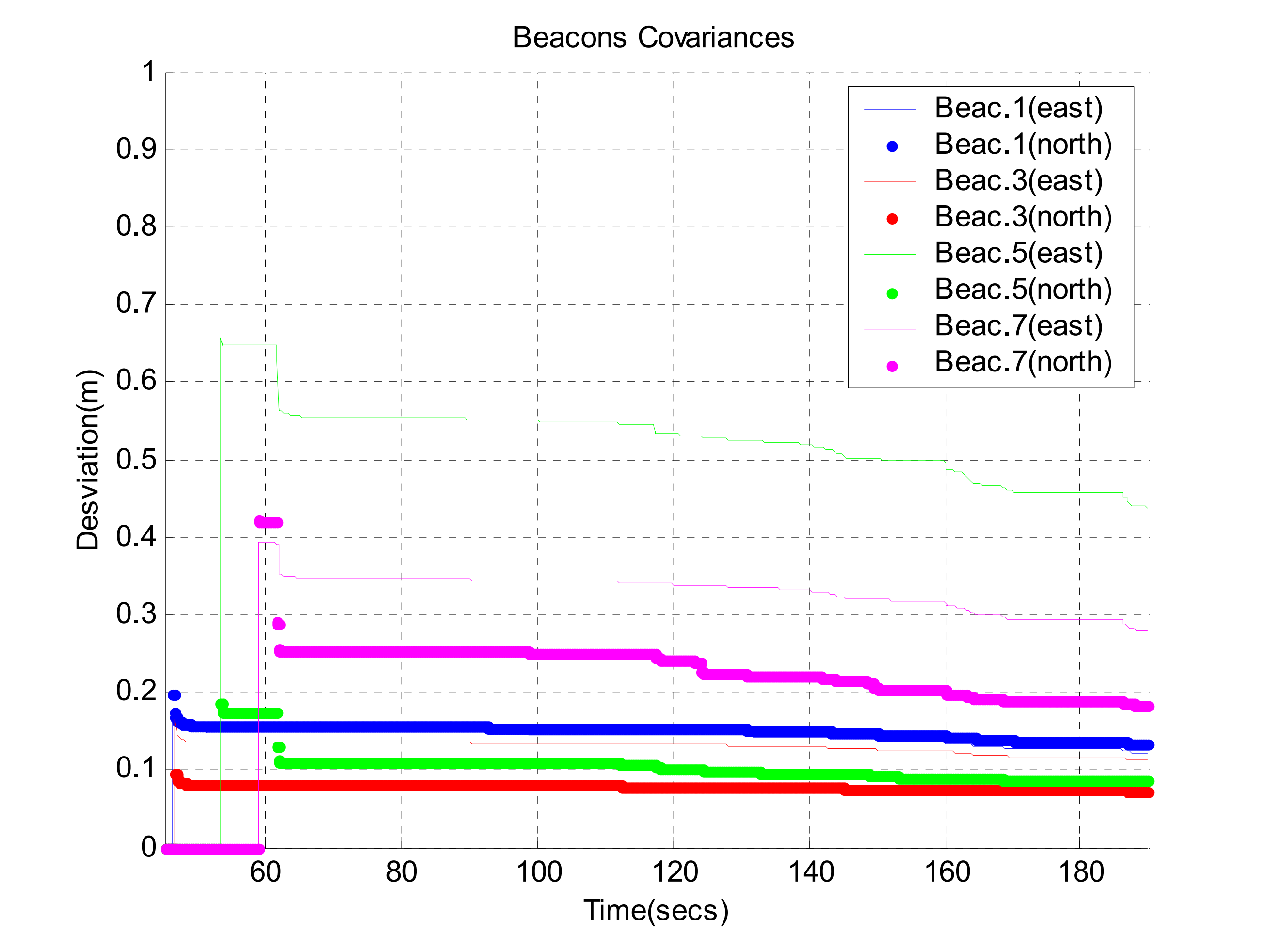
Innovation Sequence



Autocorrelation of Innovation Sequence

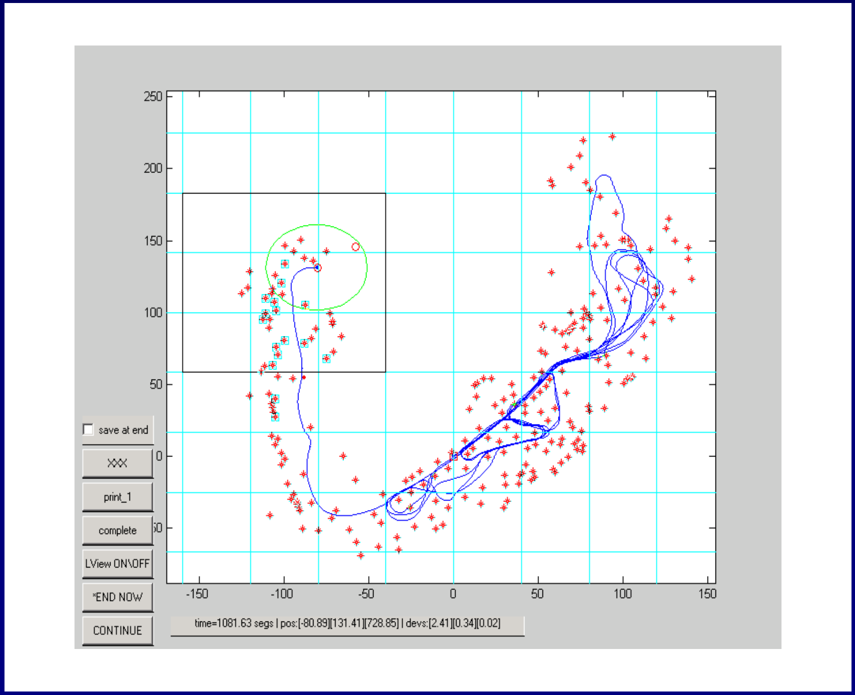


Beacons Standard Deviations



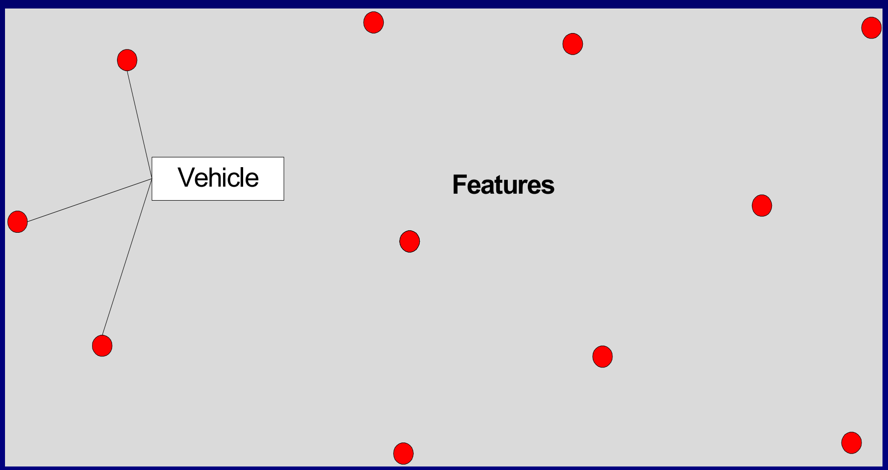
Efficient EKF implementation of SLAM

- For environment with large number of landmarks the standard Kalman Filter implementation is still very expensive **N * N**



Compressed Filter (CEKF)

- **Key Concept:**
  - When the vehicle navigates in a local area observing a group of features the information gained is a function of only the observed features.
  - This information can be saved and then transferred in one iteration to the rest of the map



Compressed EKF (CEKF)

Assume a system with dynamic and observation models:

$$X(k+1) = F\left(X(k), u(k), k\right), \quad X \in R^n$$
$$Y(k) = G\left(X(k), k\right), \quad Y \in R^m$$

With:

$$X = \begin{bmatrix} X_{A_i} \\ X_{B_i} \end{bmatrix}, \quad X_{A_i} \in R^{N_{A_i}}, \quad X_{B_i} \in R^{N_{B_i}}, \quad X \in R^N, \quad N_{A_i} < N_{B_i} < N$$

$$\Omega = \bigcup_{i=1}^{I} \Omega_i, \quad \Omega_i = [k_{i-1}, k_i]$$

$$\begin{bmatrix} X_{A_i}(k+1) \\ X_{B_i}(k+1) \end{bmatrix} = F(X, u, k) = \begin{bmatrix} f_i\left(X_{A_i}, u, k\right) \\ X_{B_i}(k) \end{bmatrix}$$

$$y = G(X, k) = h\left(X_{A_i}, k\right)$$

$$\forall (X, u, k) \quad / \quad k_{i,1} \le k \le k_{i,2}$$

Compressed EKF (CEKF)

When k=k1 $\qquad \phi_{k_1} = I, \quad \psi_{k_1} = \overline{\overline{0}}, \quad \theta_{k_1} = \overline{0}$

uxiliary ops. order Na

k1 < k < k2

Each Prediction and update

KF order Na

prediction step
$$\begin{cases} \text{standard EKF predicting for } P_{aa}, X_a \\ \phi_k = J_{aa} \cdot \phi_{k-1}, \quad \psi_k = \psi_{k-1}, \quad \theta_k = \theta_{k-1} \end{cases}$$

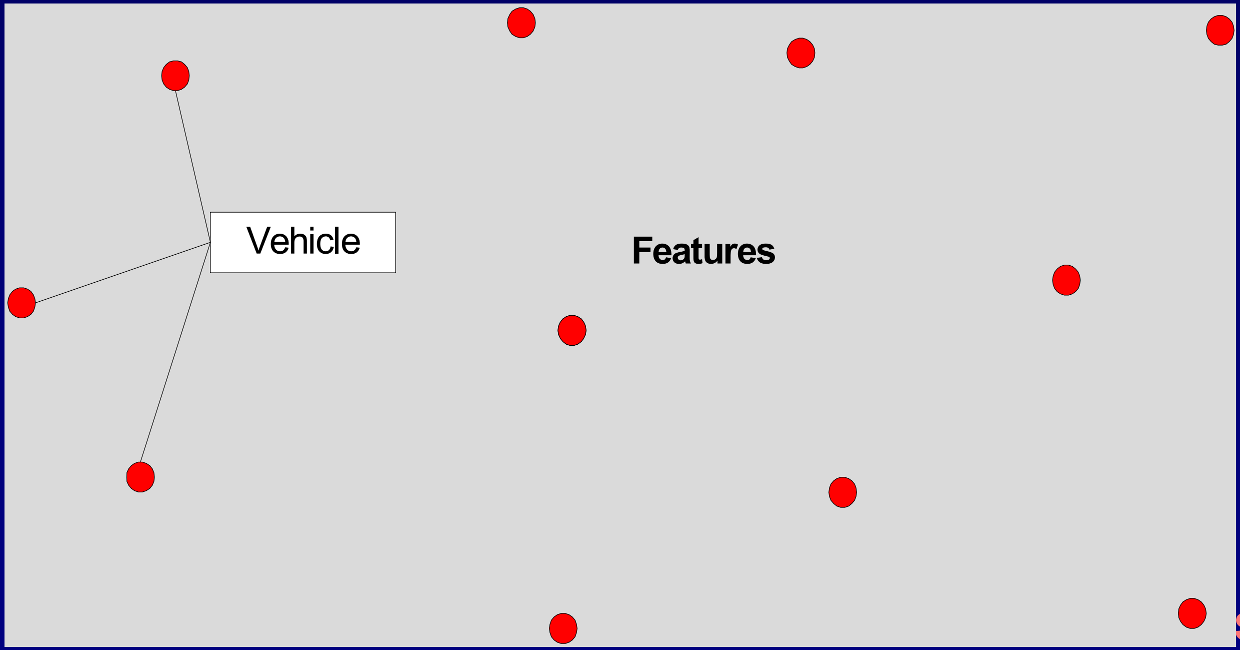
update step
$$\begin{cases} \text{standard EKF updating for } P_{aa}, X_a \\ \phi_k = (I - \mu_k) \cdot \phi_{k-1} \qquad \begin{cases} \beta_k = H_{a,k}^T \cdot S_k^{-1} \cdot H_{a} \\ \mu_k = P_{aa,k} \cdot \beta_k \end{cases} \\ \psi_k = \psi_{k-1} + \phi_{k-1}^T \cdot \beta_k \cdot \phi_{k-1} \\ \theta_k = \theta_{k-1} + \phi_{k-2}^T \cdot H_{a,k-1}^T \cdot S_{k-1}^{-1} \cdot z_{k-1} \end{cases}$$

k=k2 **Global update**

$$P_{ab,(k_2)} = \phi_{(k_2-1)} \cdot P_{ab,(k_1)}$$
$$P_{bb,(k_2)} = P_{bb,(k_1)} - P_{ba,(k_1)} \cdot \psi_{(k_2-1)} \cdot P_{ab,(k_1)}$$
$$X_{b,(k_2)} = X_{b,(k_1)} - P_{ba,(k_1)} \cdot \theta_{k_2}$$

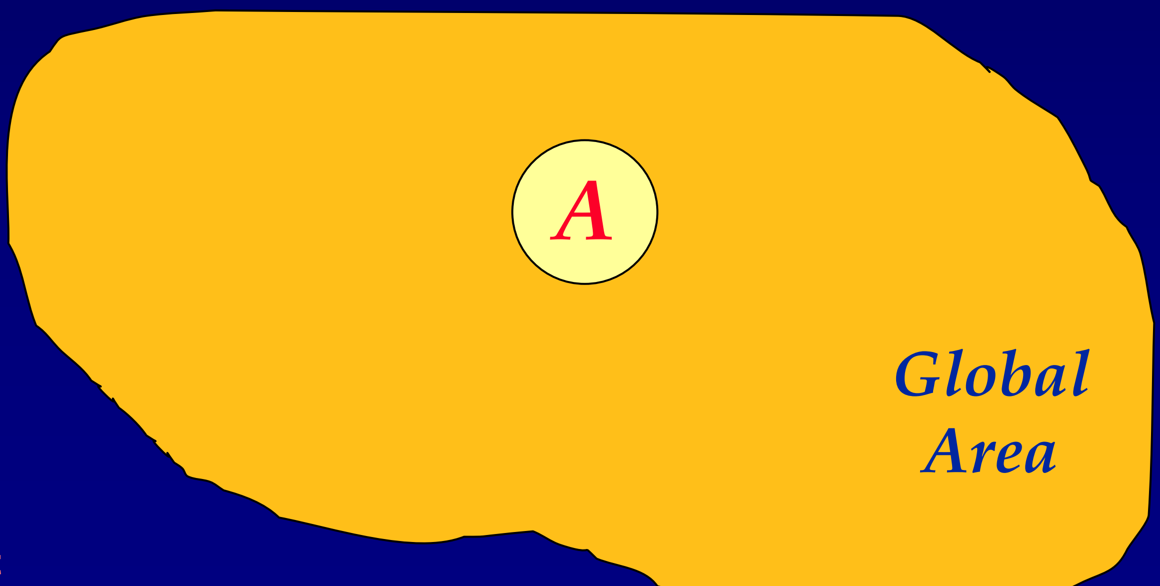
Advantages of the CEKF Algorithm

- **Constant Computational Requirements**
  - Independent of the total number of features in the global map
- **Full use of High Frequency sensors**



Compressed Filter

- **The computational cost of the SLAM proportional to 2Na * 2Na**
  - (Na landmarks in the local area)
- **Full update is only required when the vehicle leaves the local Area A**
  - Only the features in the new area need to be updated



General Compressed Filters

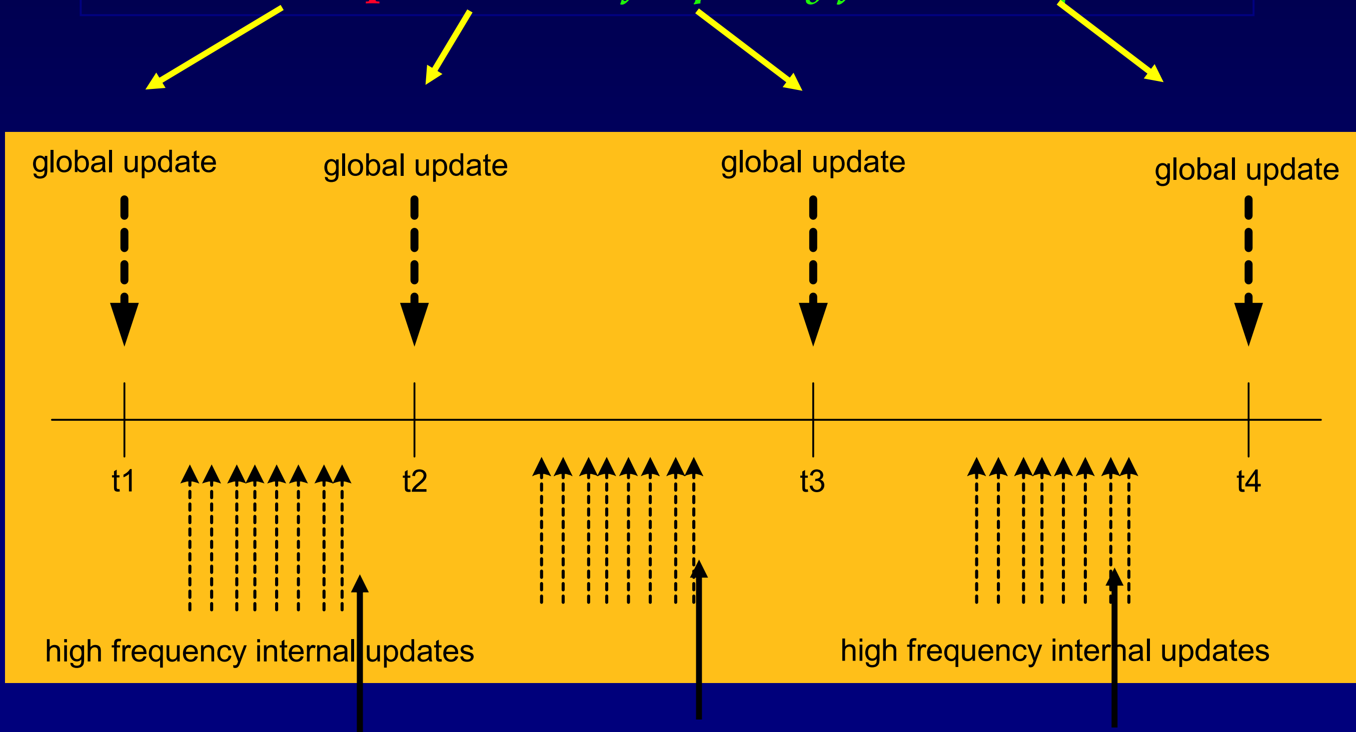
$$\cdots \left\{ X = \begin{bmatrix} X_{A_1} \\ X_{B_1} \end{bmatrix},\quad \begin{bmatrix} X_{A_1}(k+1) \\ X_{B_1}(k+1) \end{bmatrix} = \begin{bmatrix} f_1\left(X_{A_1},u,k\right) \\ X_{B_1}(k) \end{bmatrix},\quad y(k) = h_1\left(X_{A_1},k\right) \right\}$$

$$\left\{ X = \begin{bmatrix} X_{A_2} \\ X_{B_2} \end{bmatrix},\quad \begin{bmatrix} X_{A_2}(k+1) \\ X_{B_2}(k+1) \end{bmatrix} = \begin{bmatrix} f_2\left(X_{A_2},u,k\right) \\ X_{B_2}(k) \end{bmatrix},\quad y(k) = h_2\left(X_{A_2},k\right) \right\}$$

$$\left\{ X = \begin{bmatrix} X_{A_3} \\ X_{B_3} \end{bmatrix},\quad \begin{bmatrix} X_{A_3}(k+1) \\ X_{B_3}(k+1) \end{bmatrix} = \begin{bmatrix} f_3\left(X_{A_3},u,k\right) \\ X_{B_3}(k) \end{bmatrix},\quad y(k) = h_3\left(X_{A_3},k\right) \right\} \cdots$$

Compressed filter operation

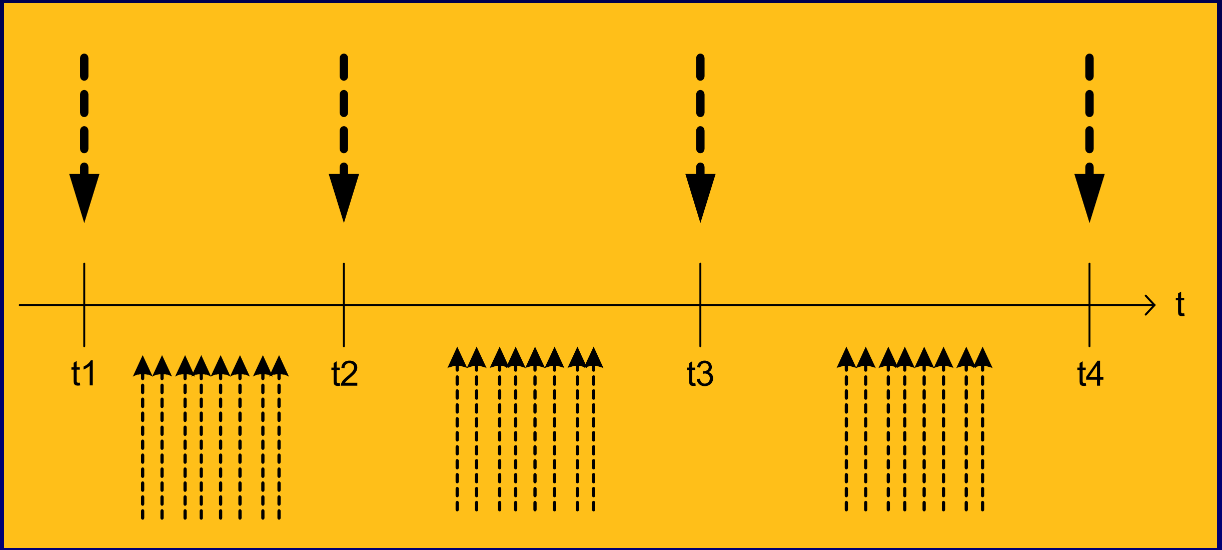
- **External Estimator**
- **Global updates**: *low frequency full EKF update*.



- **Internal estimator** *(predictions and observations) at **high frequency**. Estimator running on a **reduced system*** X_A

Compressed filter operation

Global Update (Large system estimation): EKF



* *high frequency*
* *Small system*

*EKF

*"Unscented Filter"

*S.O.G. ….

Global update

*It is a **low frequency event**.*

The reduced system estimator transfers the collected information to the full system doing an full EKF update.

If necessary, It performs a Gaussian Approximation of $p(Xa(k)\,|\,Z(k))$

Full EKF update Computational Cost

$$\begin{cases} \text{general case } O\left(N_A^2 \cdot N_B^2\right) \\ \text{SLAM case } O(L \cdot N_B^2) \le O(N_A \cdot N_B^2) \end{cases}$$

*An **EKF / EKF** combination (**CEKF**) gives identical result to the full EKF.*

Compressed filters Properties:

Internal estimator (small system)** *can be more* ***sophisticated.

** Sub optimal simplifications run at the global update (low frequency)*

- *High frequency.*
- *Good numerical stability.*
- *Adequate for non linear problems.*

- *It can be less conservative…if Sub Optimal simplifications are used.*

Australian Centre for Field Robotics

EXAMPLE

When k=k1

$$\phi_{k_1} = I, \quad \psi_{k_1} = \bar{\bar{0}}, \quad \theta_{k_1} = \bar{0}$$

uxiliary ops. order Na

k1 < k < k2

Each Prediction and update

EKF order Na

prediction step
$$\begin{cases} \text{standard EKF predicting for } P_{aa}, X_a \\ \phi_k = J_{aa} \cdot \phi_{k-1}, \quad \psi_k = \psi_{k-1}, \quad \theta_k = \theta_{k-1} \end{cases}$$

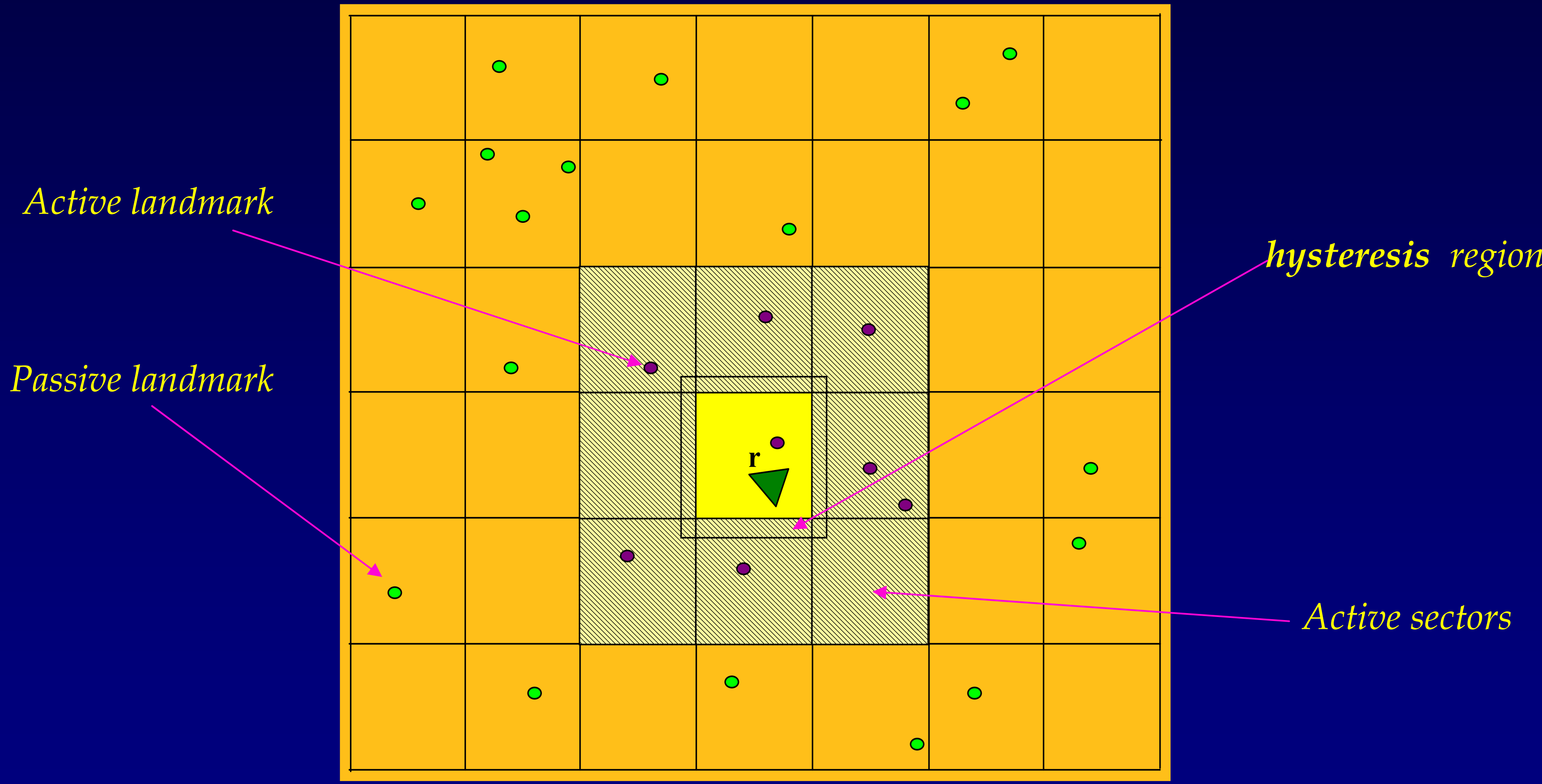
update step
$$\begin{cases} \text{standard EKF updating for } P_{aa}, X_a \\ \phi_k = (I - \mu_k) \cdot \phi_{k-1} \\ \psi_k = \psi_{k-1} + \phi_{k-1}^T \cdot \beta_k \cdot \phi_{k-1} \\ \theta_k = \theta_{k-1} + \phi_{k-2}^T \cdot H_{a,k-1}^T \cdot S_{k-1}^{-1} \cdot z_{k-1} \end{cases} \quad \begin{cases} \beta_k = H_{a,k}^T \cdot S_k^{-1} \cdot H_{a,k} \\ \mu_k = P_{aa,k} \cdot \beta_k \end{cases}$$

k=k2

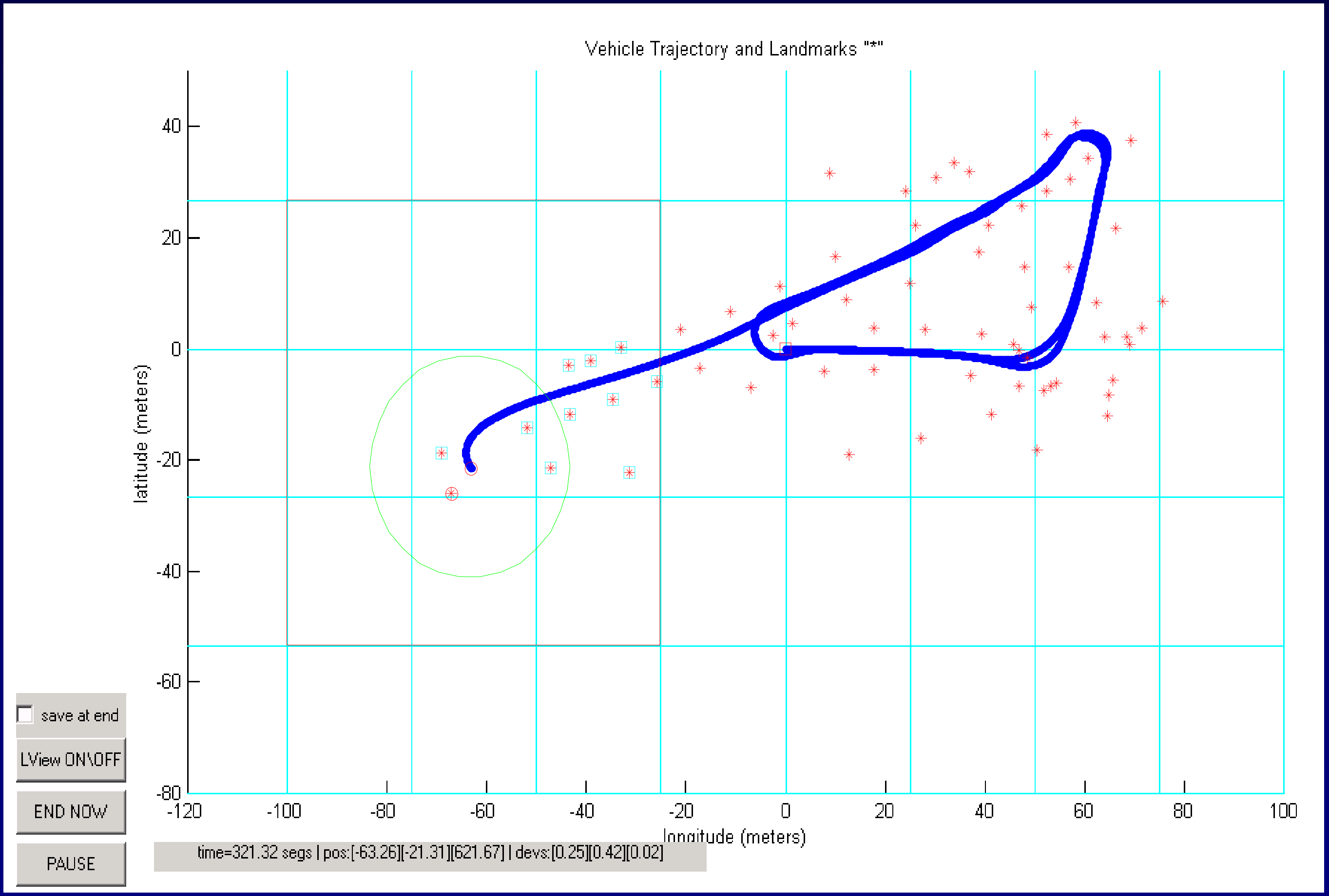
Global update

$$P_{ab,(k_2)} = \phi_{(k_2-1)} \cdot P_{ab,(k_1)}$$
$$P_{bb,(k_2)} = P_{bb,(k_1)} - P_{ba,(k_1)} \cdot \psi_{(k_2-1)} \cdot P_{ab,(k_1)}$$
$$X_{b,(k_2)} = X_{b,(k_1)} - P_{ba,(k_1)} \cdot \theta_{k_2}$$

Map Management



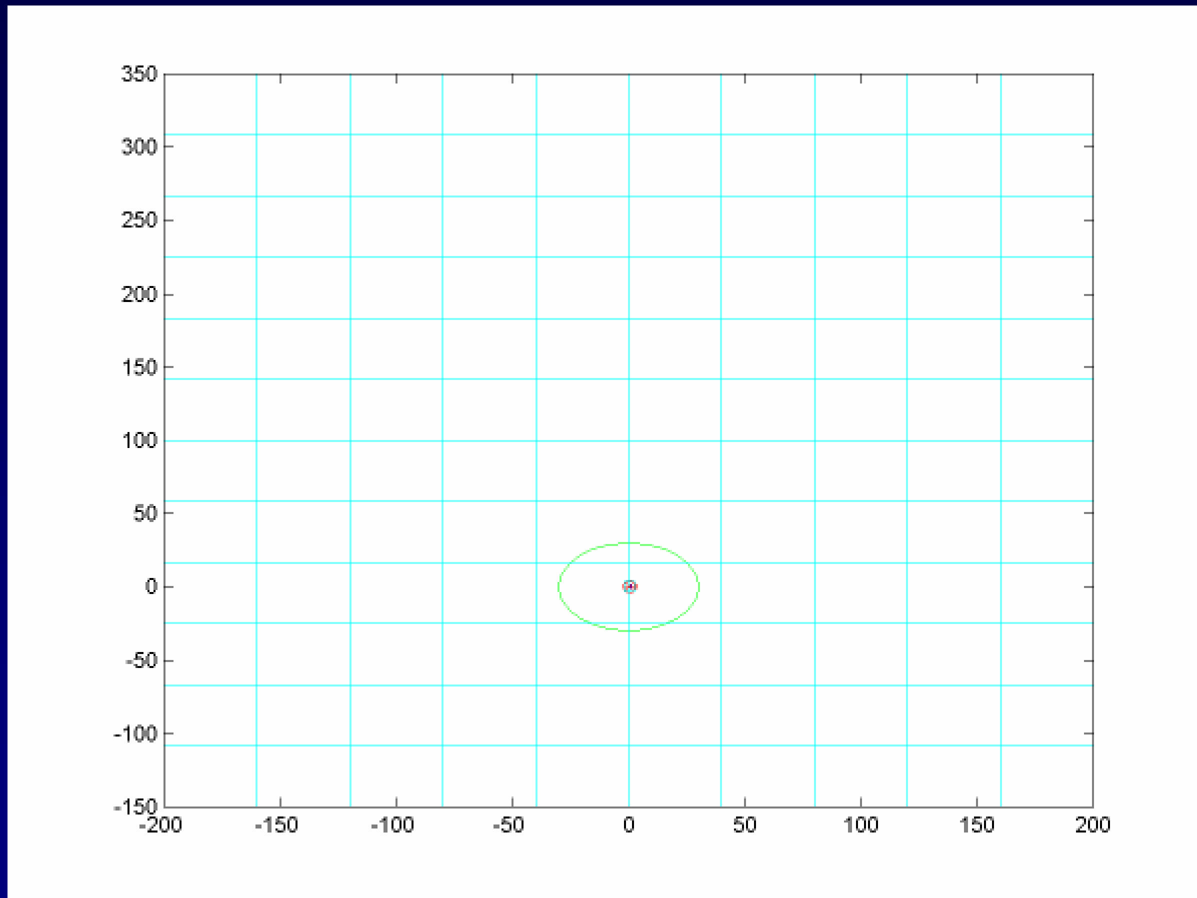
Map Management



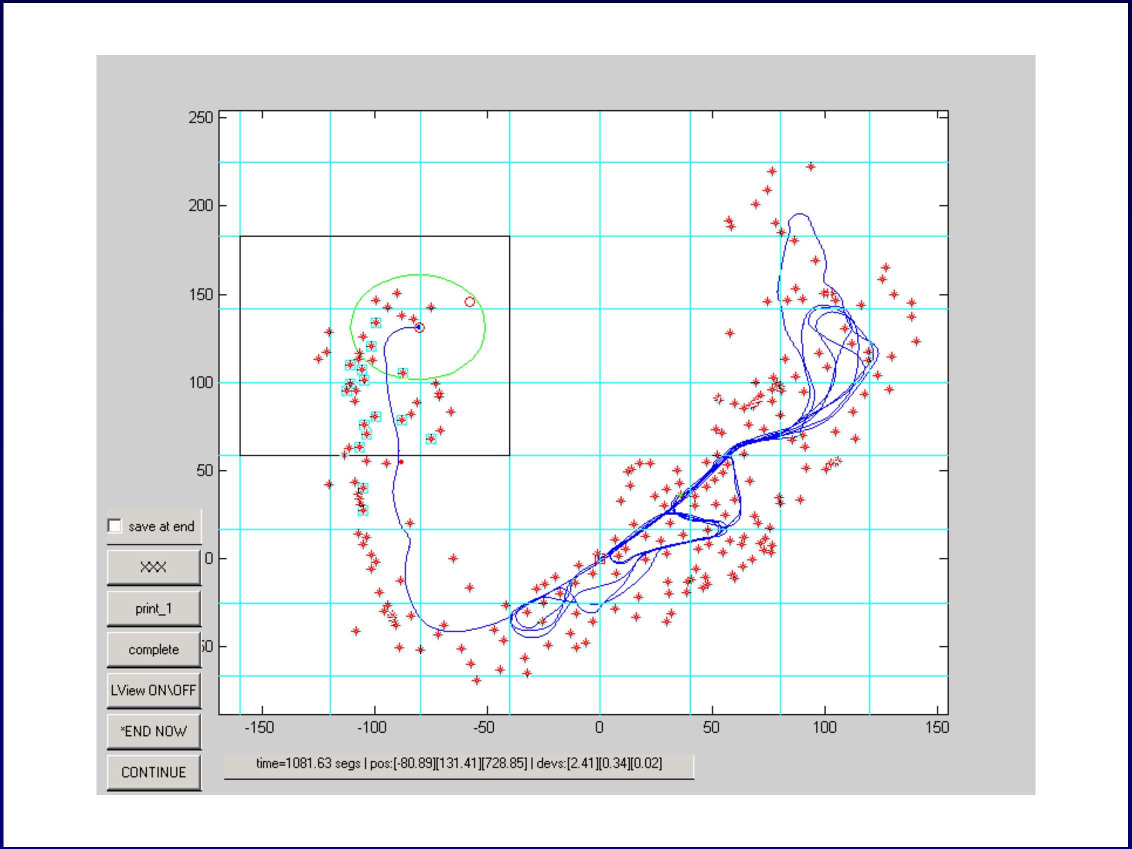
Experimental Run

Outdoor Environment

SLAM


350
300
250
200
150
100
50
0
-50
-100
-150
-200
-150
-100
-50
0
50
100
150
200

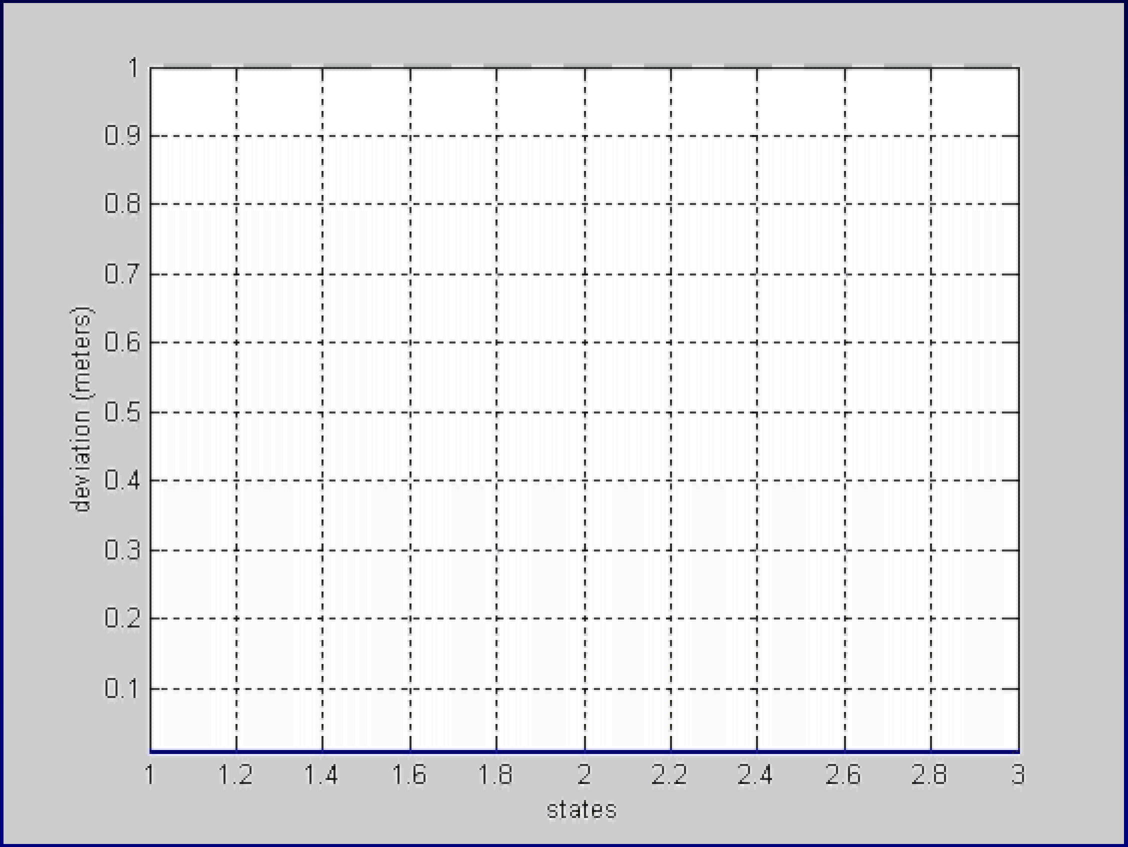
Map and Trajectory



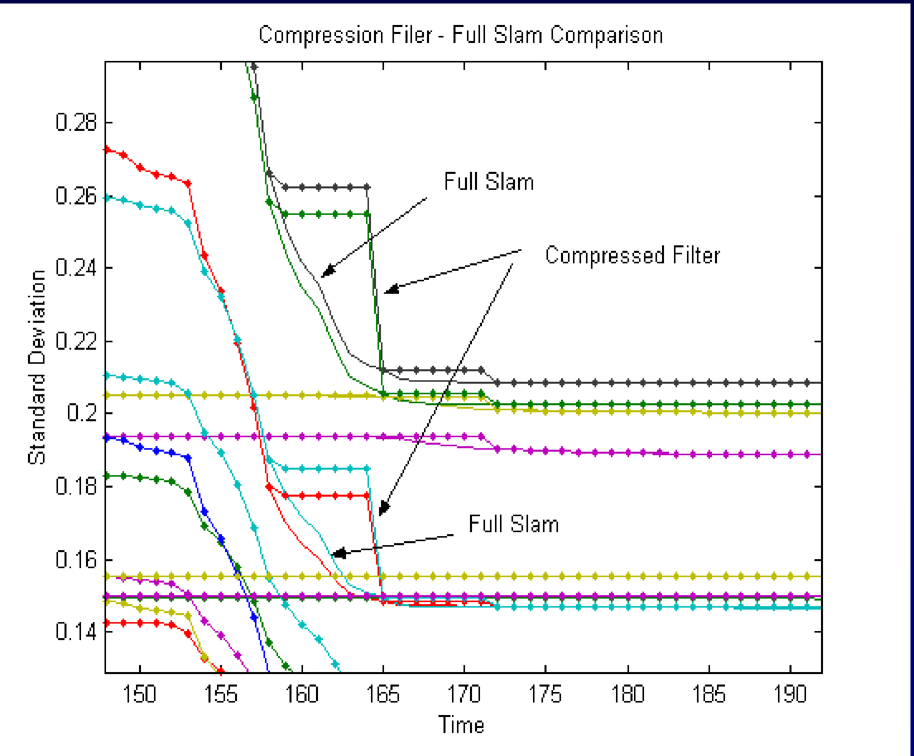
Landmarks

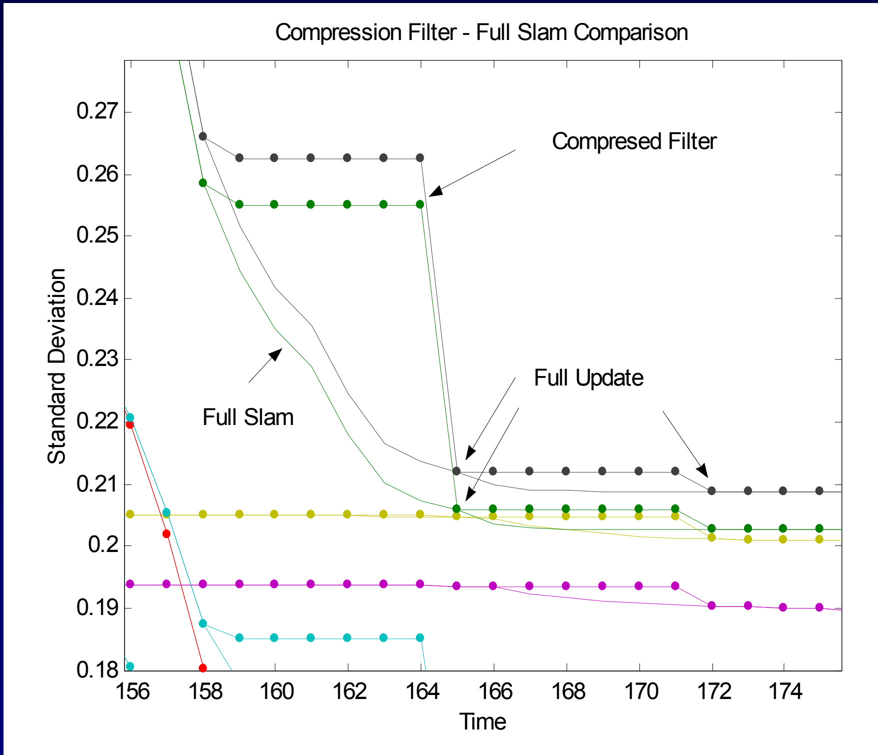
Covariance

Landmark Covariance

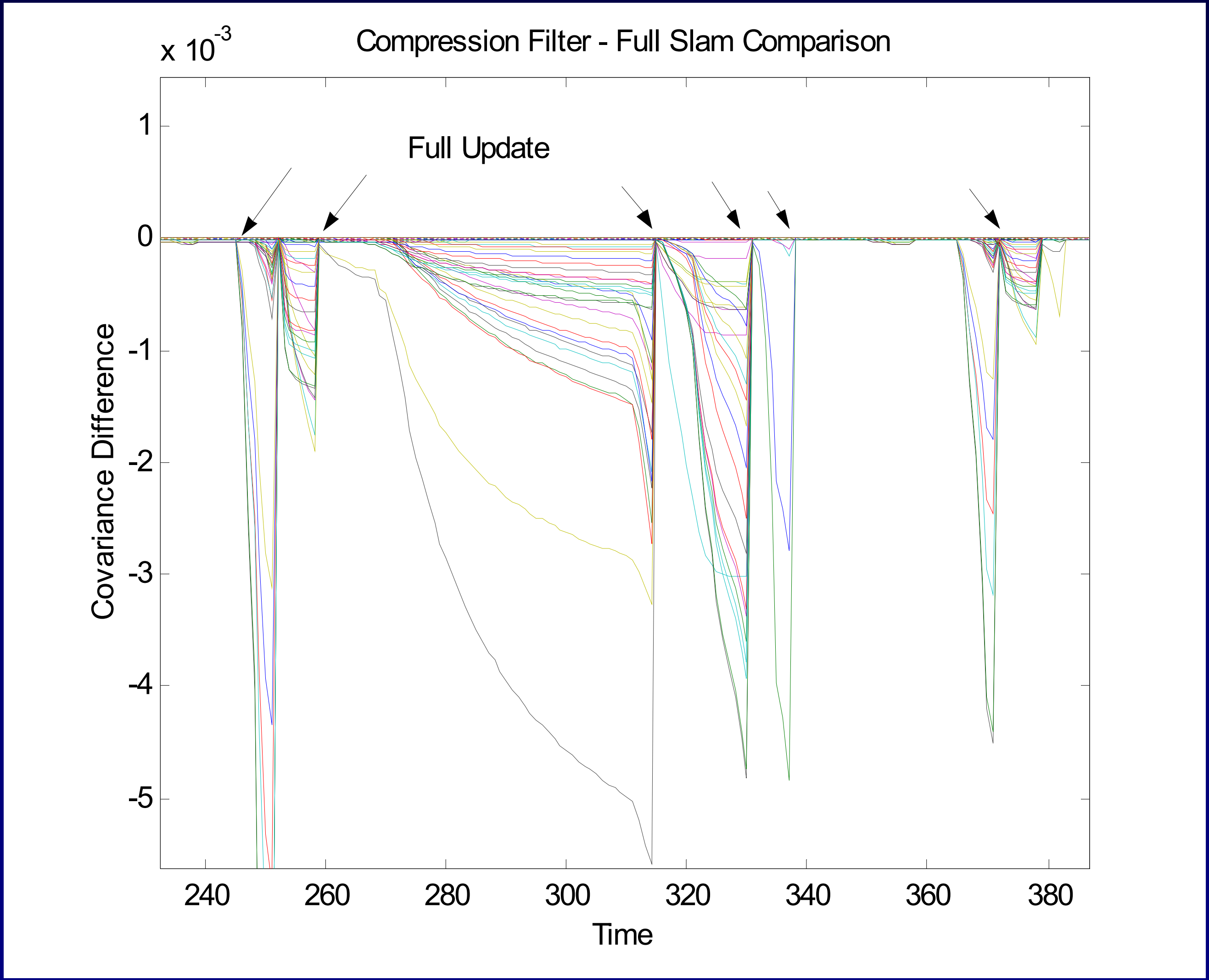


CEKF–SLAM vs Full EKF SLAM



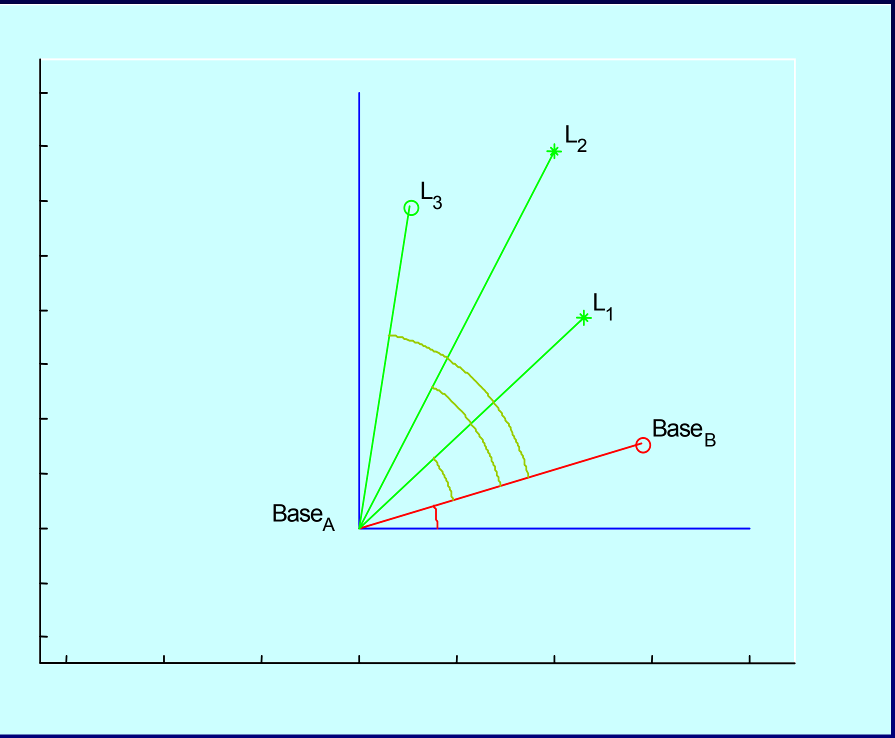


Compressed Algorithm



Relative Representation

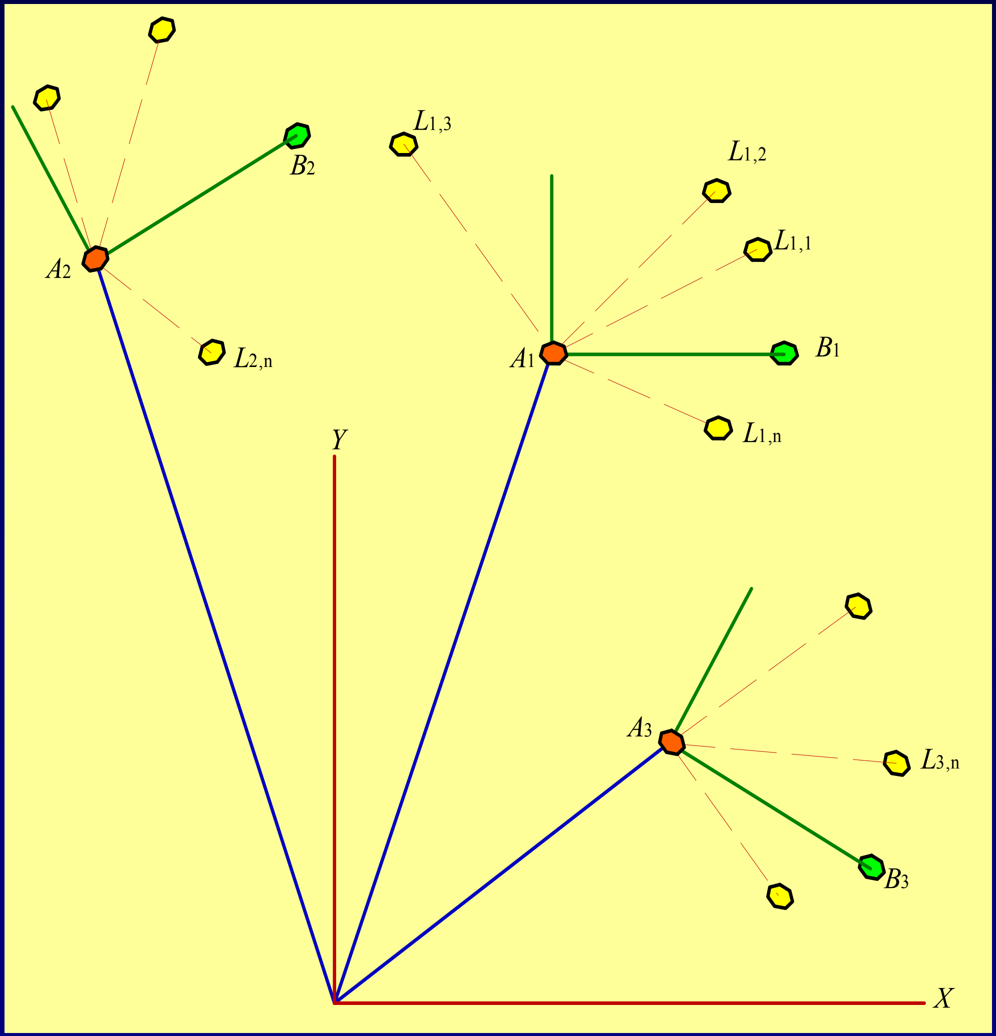
Landmark classification



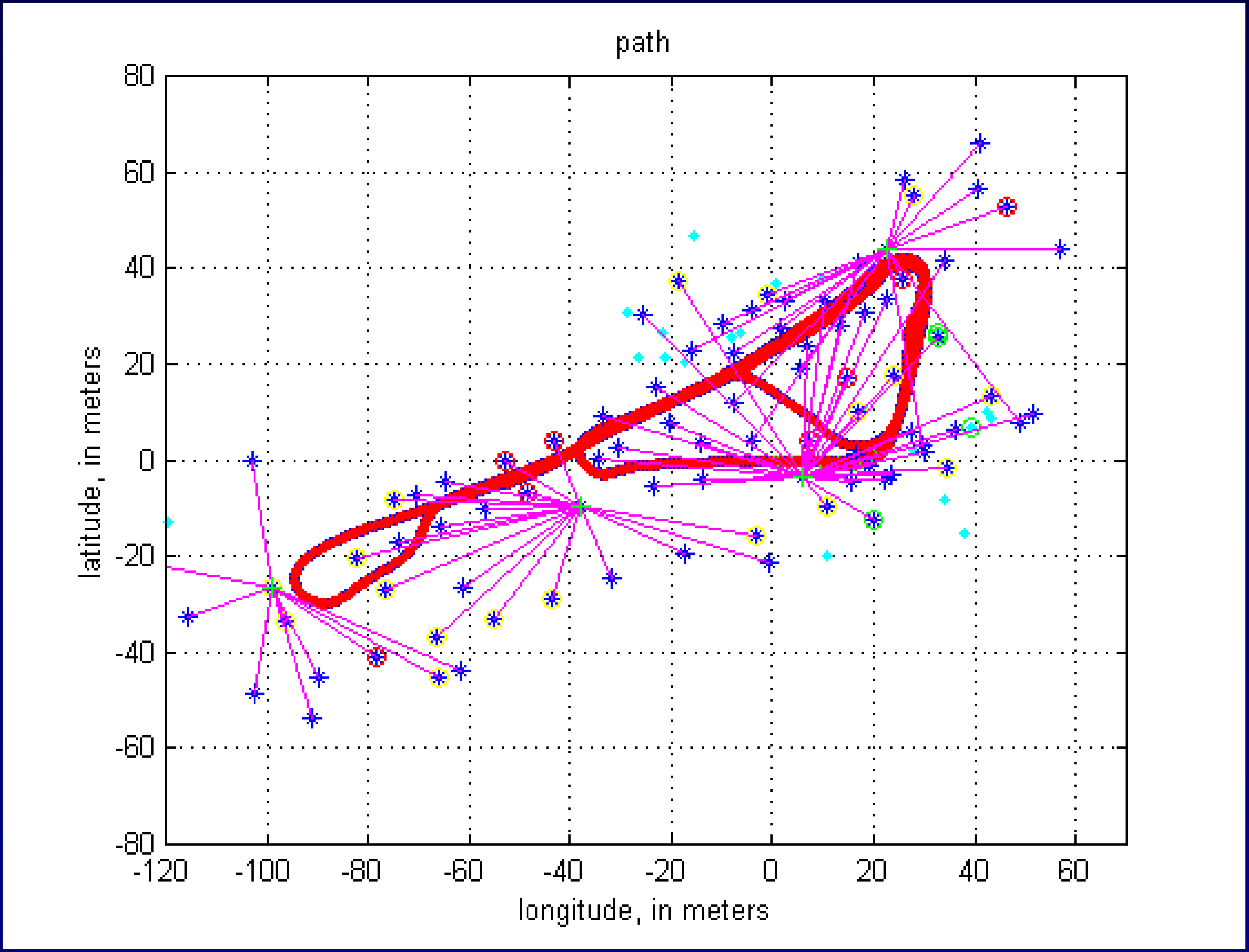
Landmark types

- *"A" Bases : Absolute*
- *"B" Bases : Semi-absolute*
- *Li normal landmarks: Relative*

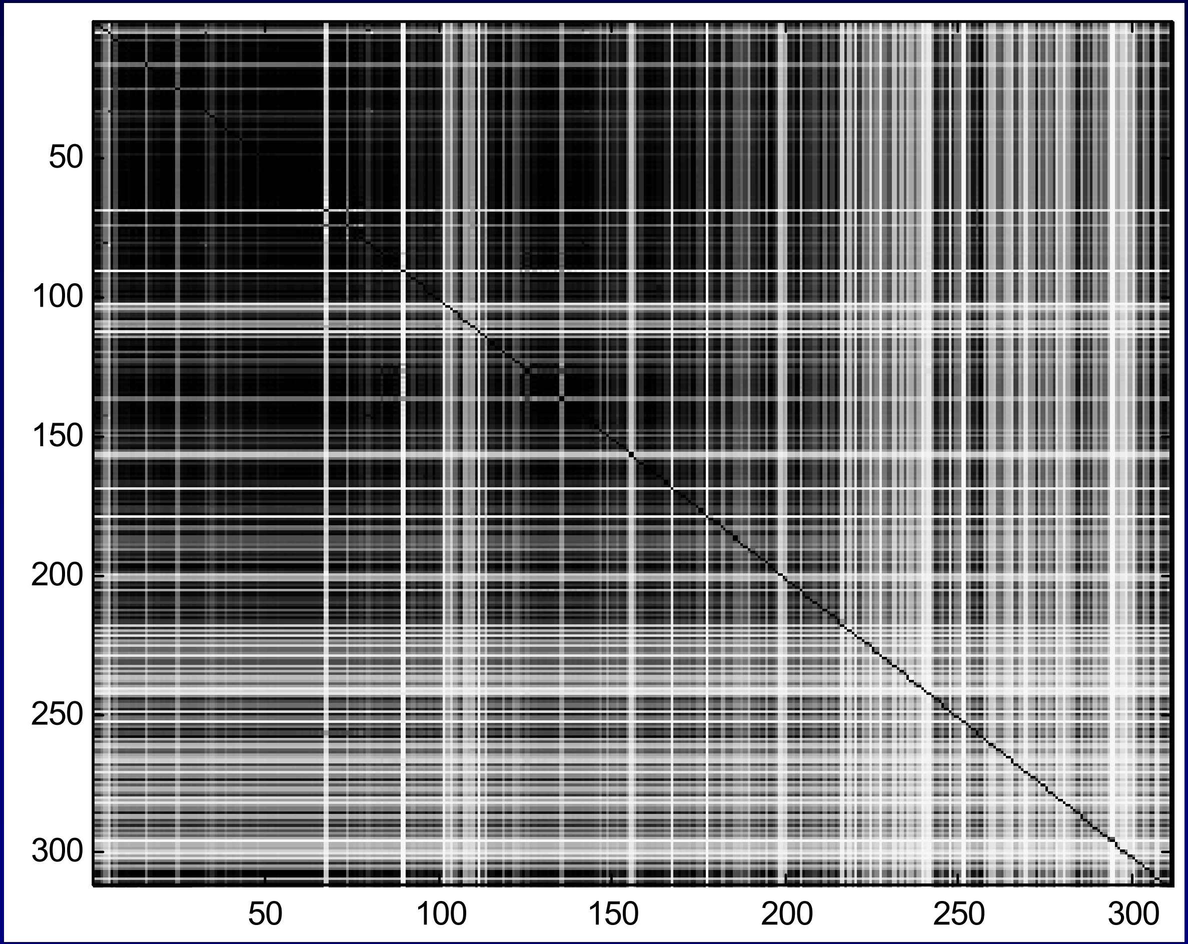
Constellation Map



SLAM Map and trajectory (Victoria Park)

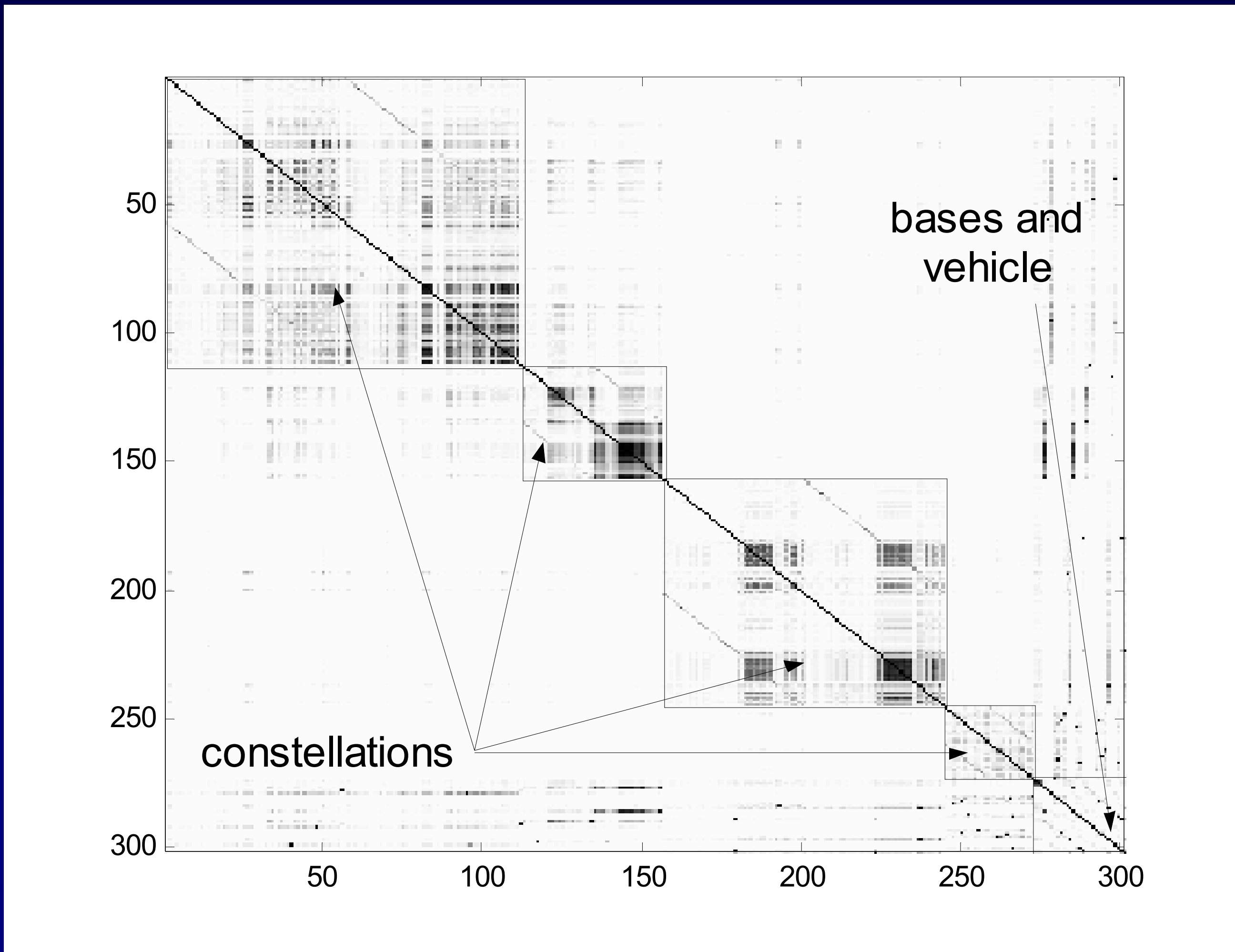


Correlation Coefficients for the absolute representation



$$c(i,j) = \frac{\left|P(i,j)\right|}{\sqrt{P(i,i)\cdot P(j,j)}} \in [\ 0\ ,\ 1\ ]$$

Correlation Coefficients for the absolute representation



$$c(i,j) = \frac{|P(i,j)|}{\sqrt{P(i,i) \cdot P(j,j)}}$$

$$\in [0, 1]$$

Sub-Optimal SLAM

Sub-optimal Solutions: De-correlation Algorithms

- **In the general case it is possible to de-correlate the covariance submatrices corresponding to two groups of states, Xa and Xb.**

$$P_1 = \begin{bmatrix} A & D & E \\ D^T & B & F \\ E^T & F^T & C \end{bmatrix} \qquad \begin{array}{c} A \in R^{n\,x\,n}, B \in R^{m\,xm} \\ C \in R^{l\,xl} \\ D, E, F, \ldots \end{array}$$

$$D = \begin{bmatrix} d_{11} & .. & .. & d_{1m} \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ d_{n1} & .. & .. & d_{nm} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} A+\alpha & \bar{\bar{0}} & E \\ \bar{\bar{0}} & B+\beta & F \\ E^T & F^T & C \end{bmatrix}$$

$$P_1 \le P_2$$

De-correlation Procedure

$$P=\begin{bmatrix}\alpha & C\\ C^T & \beta\end{bmatrix}=\begin{bmatrix}\alpha+\tilde{\alpha} & 0\\ 0 & \beta+\tilde{\beta}\end{bmatrix}-\begin{bmatrix}\tilde{\alpha} & -C\\ -C^T & \tilde{\beta}\end{bmatrix}\le\begin{bmatrix}\alpha+\tilde{\alpha} & 0\\ 0 & \beta+\tilde{\beta}\end{bmatrix}$$

$$\tilde{\alpha},\tilde{\beta}\quad/\quad\tau=\begin{bmatrix}\tilde{\alpha} & -C\\ -C^T & \tilde{\beta}\end{bmatrix}\ge 0$$

$$\tilde{\alpha}\quad/\quad\tilde{\alpha}_{i,j}=\begin{cases}\sum_{k=1}^{m}\kappa_{i,k}\cdot\left|c_{i,k}\right| & ,\quad i=j\\ 0 & ,\quad i\ne j\end{cases}$$

$$\tilde{\beta}/\tilde{\beta}_{i,j}=\begin{cases}\sum_{k=1}^{n}\tilde{\kappa}_{i,k}\cdot\left|\tilde{c}_{i,k}\right|=\sum_{k=1}^{n}\frac{1}{\kappa_{k,i}}\cdot\left|c_{k,i}\right| & ,\quad i=j\\ 0 & ,\quad i\ne j\end{cases}$$

$$\kappa_{i,k}>0\quad\forall\quad i,k$$

$$\kappa_{i,k}=\tilde{\kappa}_{k,i}=1$$

$$\tilde{\alpha}_{i,i}=\sum_{k=1}^{m}\left|c_{i,k}\right|,\quad\tilde{\beta}_{j,j}=\sum_{k=1}^{n}\left|c_{k,j}\right|$$

Selection of Passive and Active States

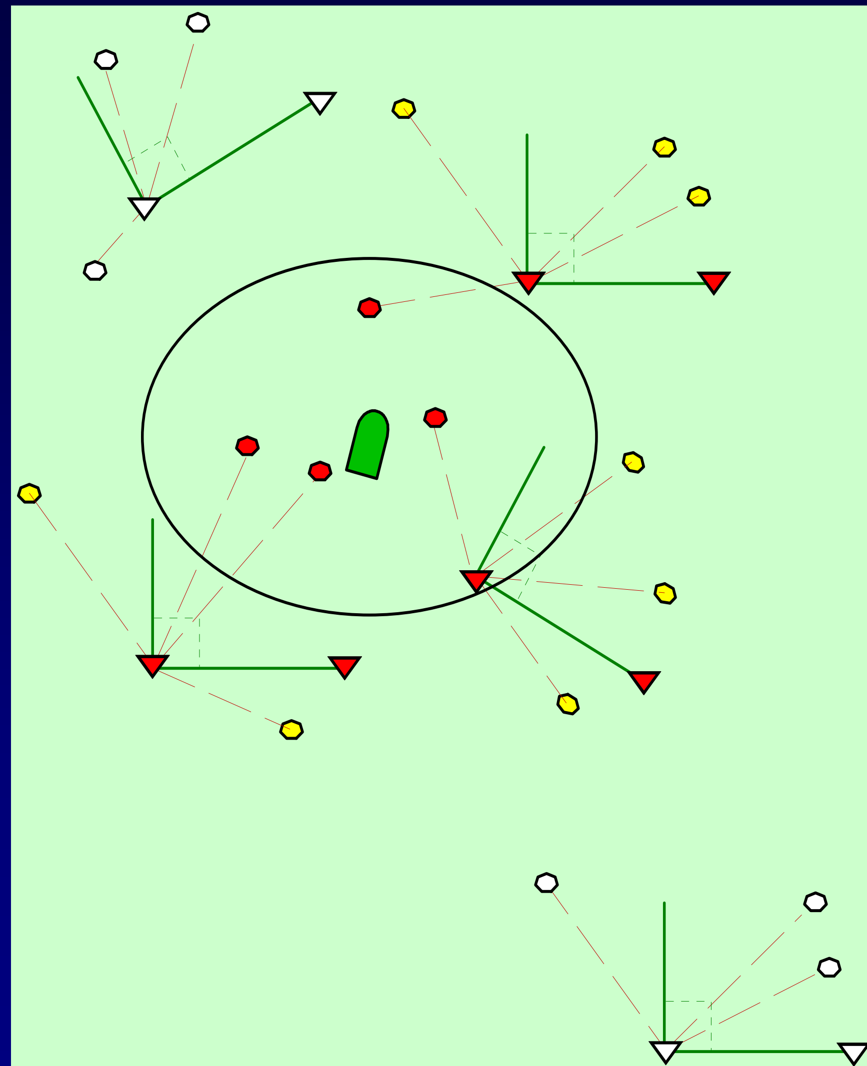
Active base landmarks

Active relative landmarks.

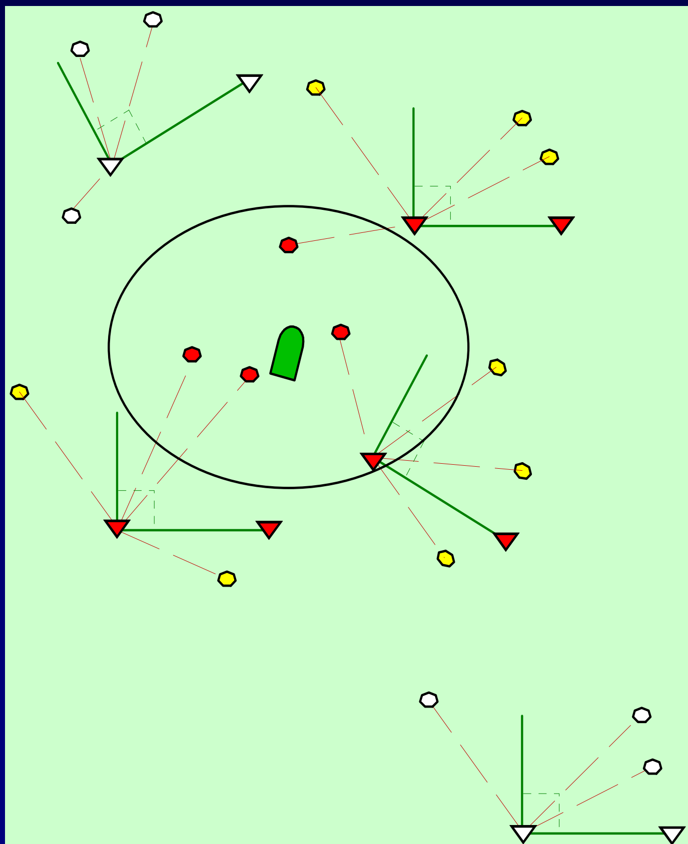
Passive close relative landmarks.

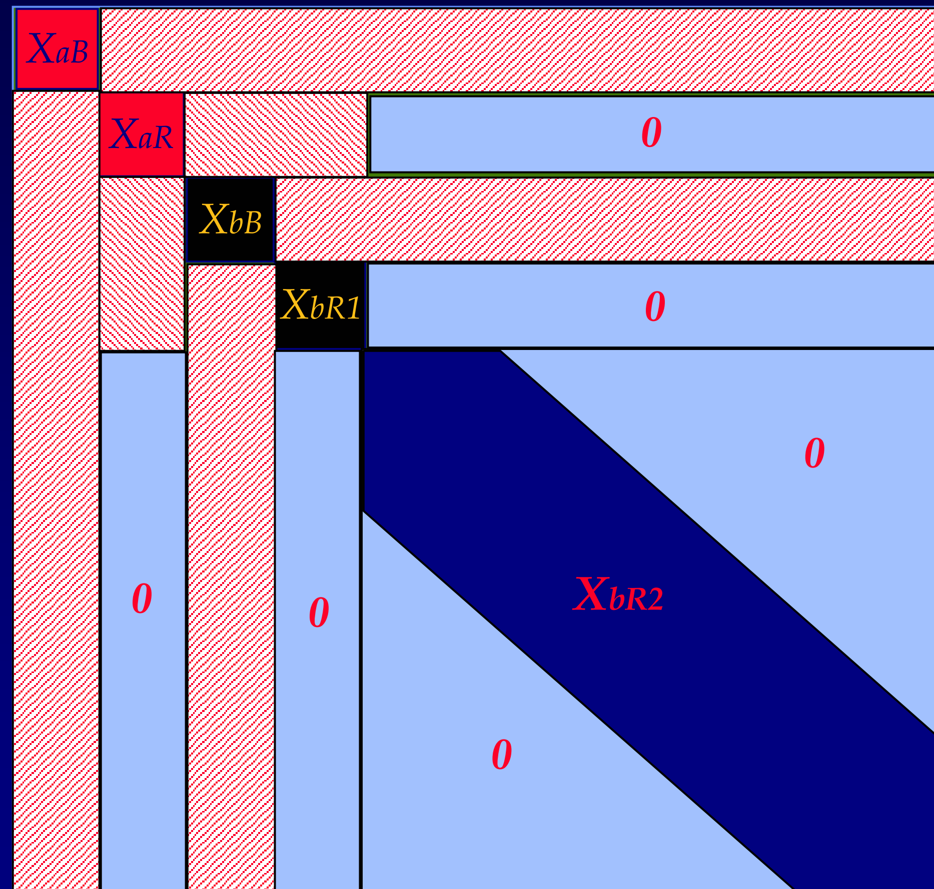
Passive far relative landmarks.

.

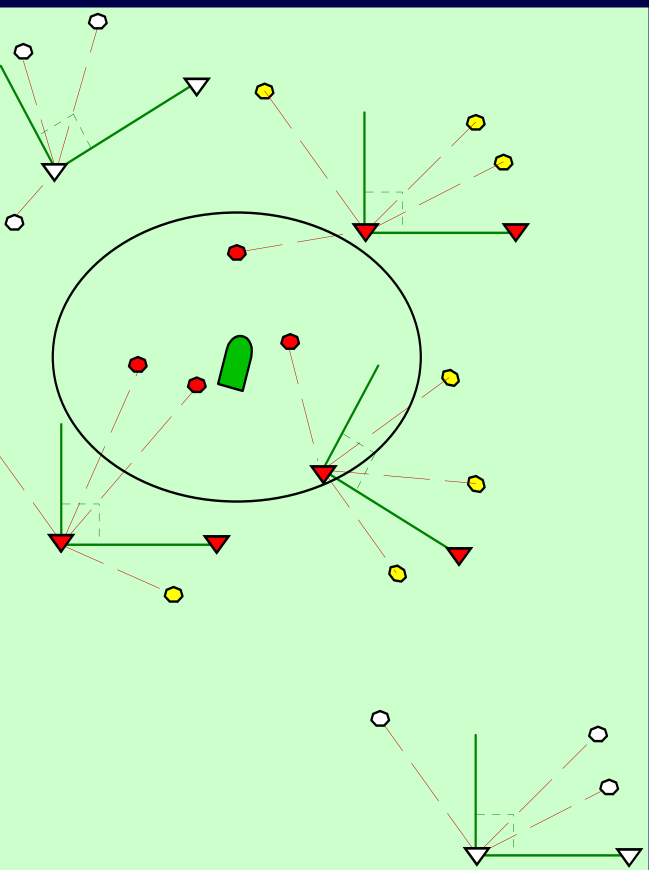


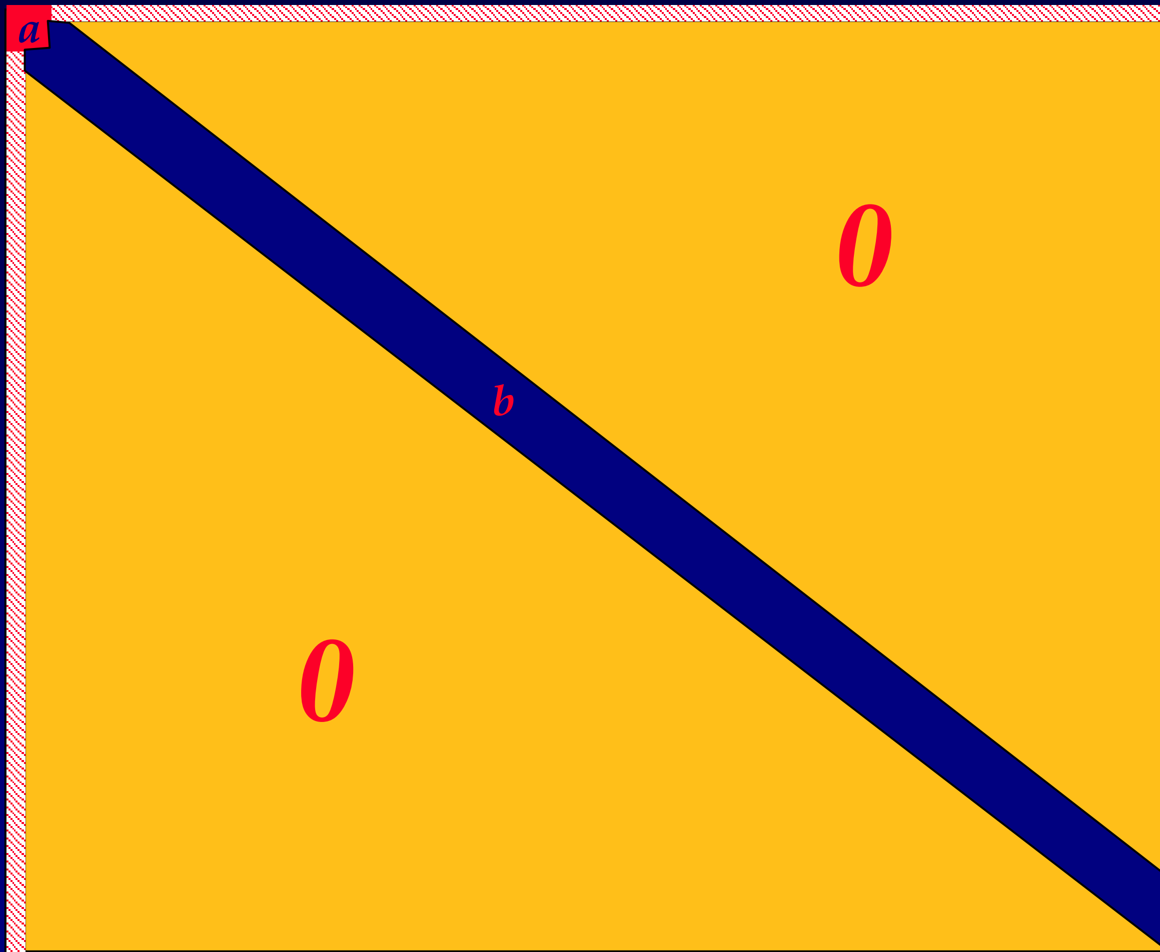
Decorrelation Matrices



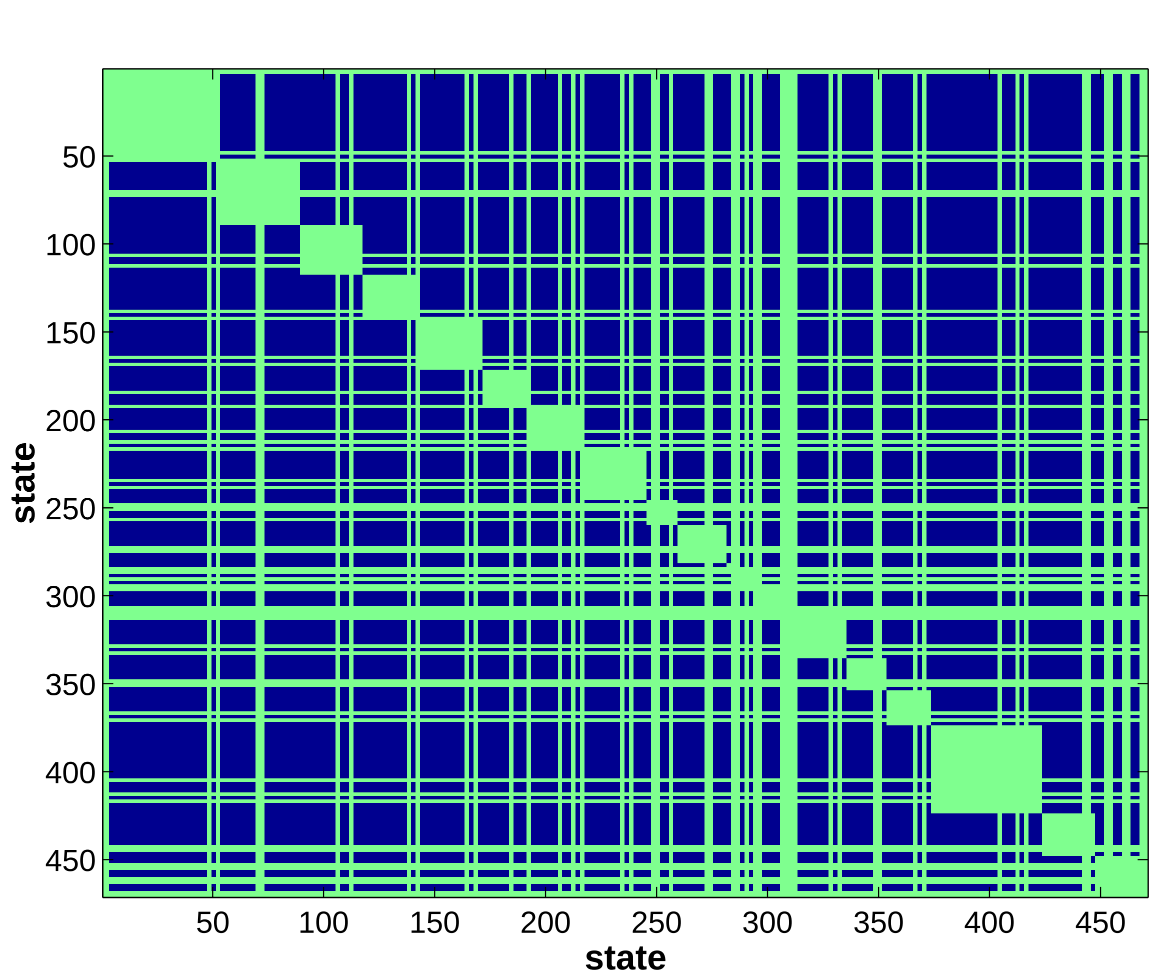


Reduced Covariance Matrix

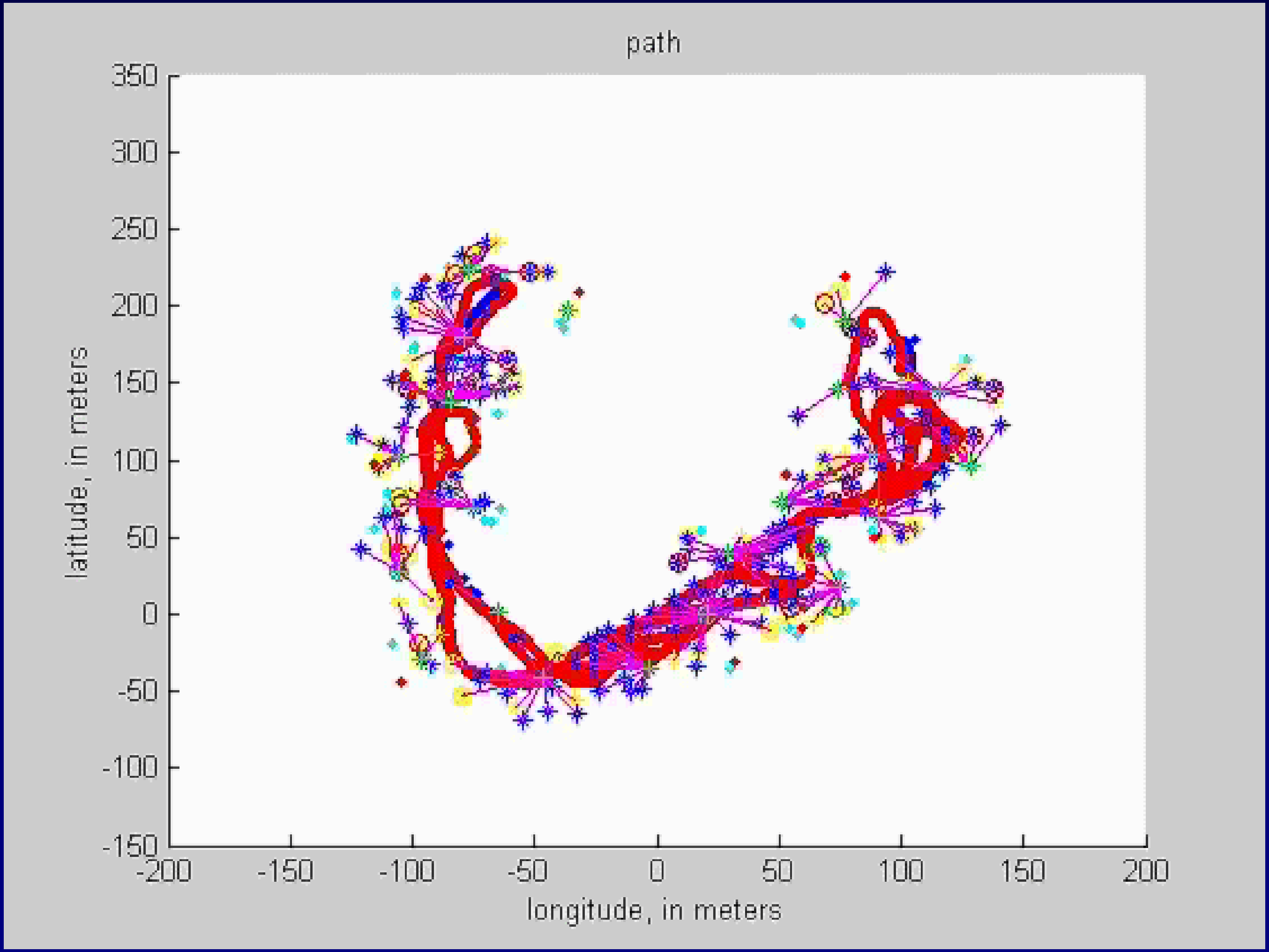




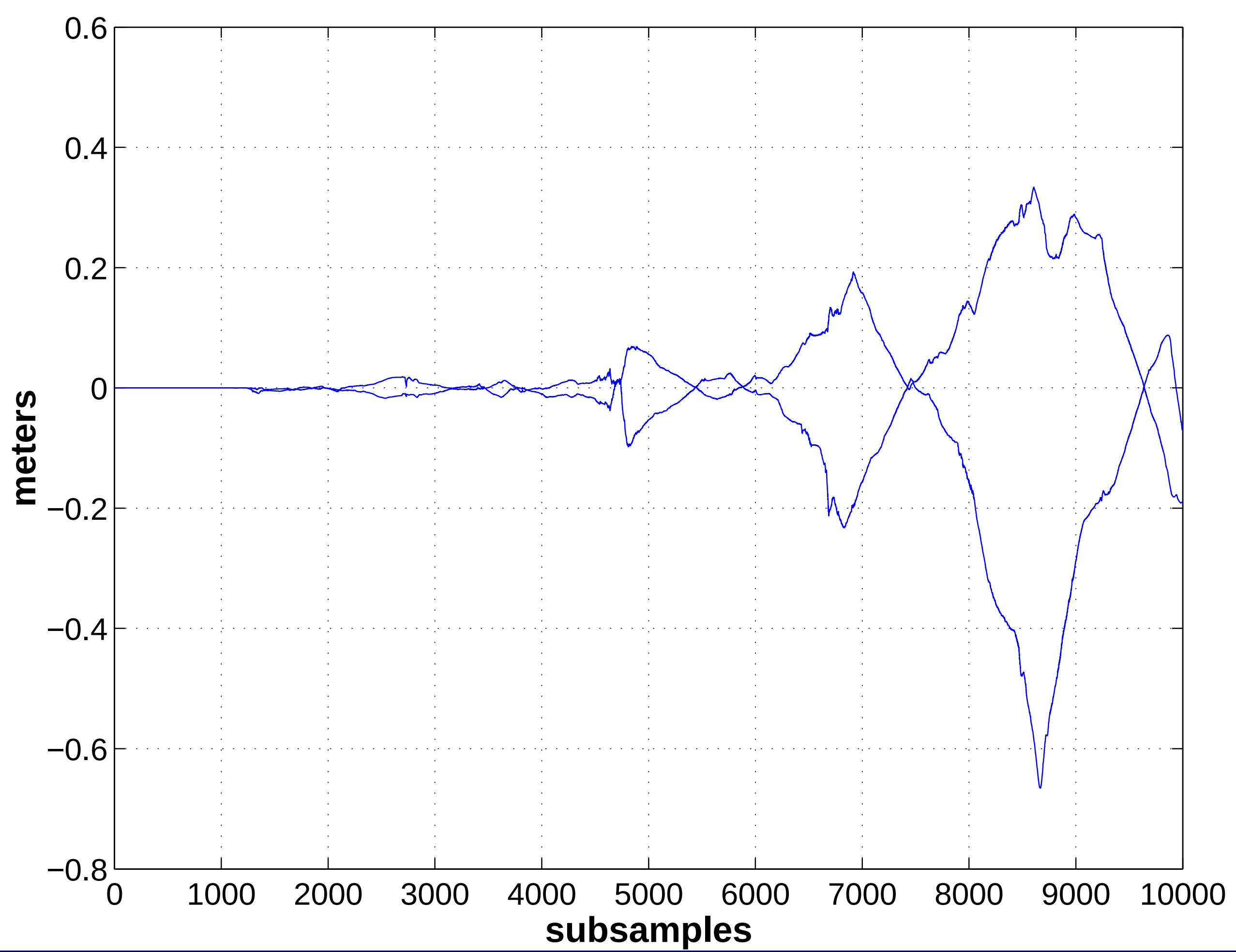
Experimental results



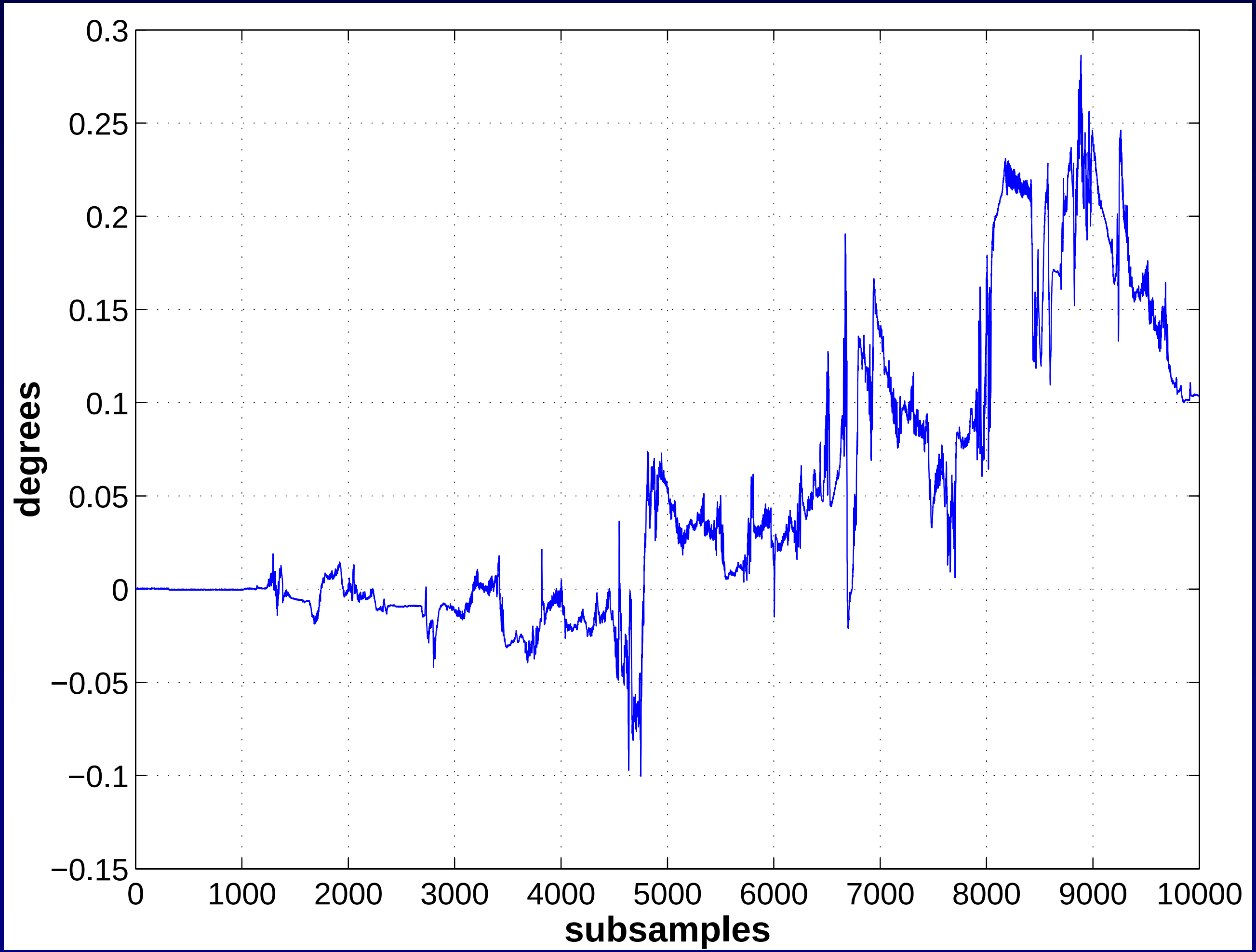
Compressed / Simplifications



Difference in position estimation

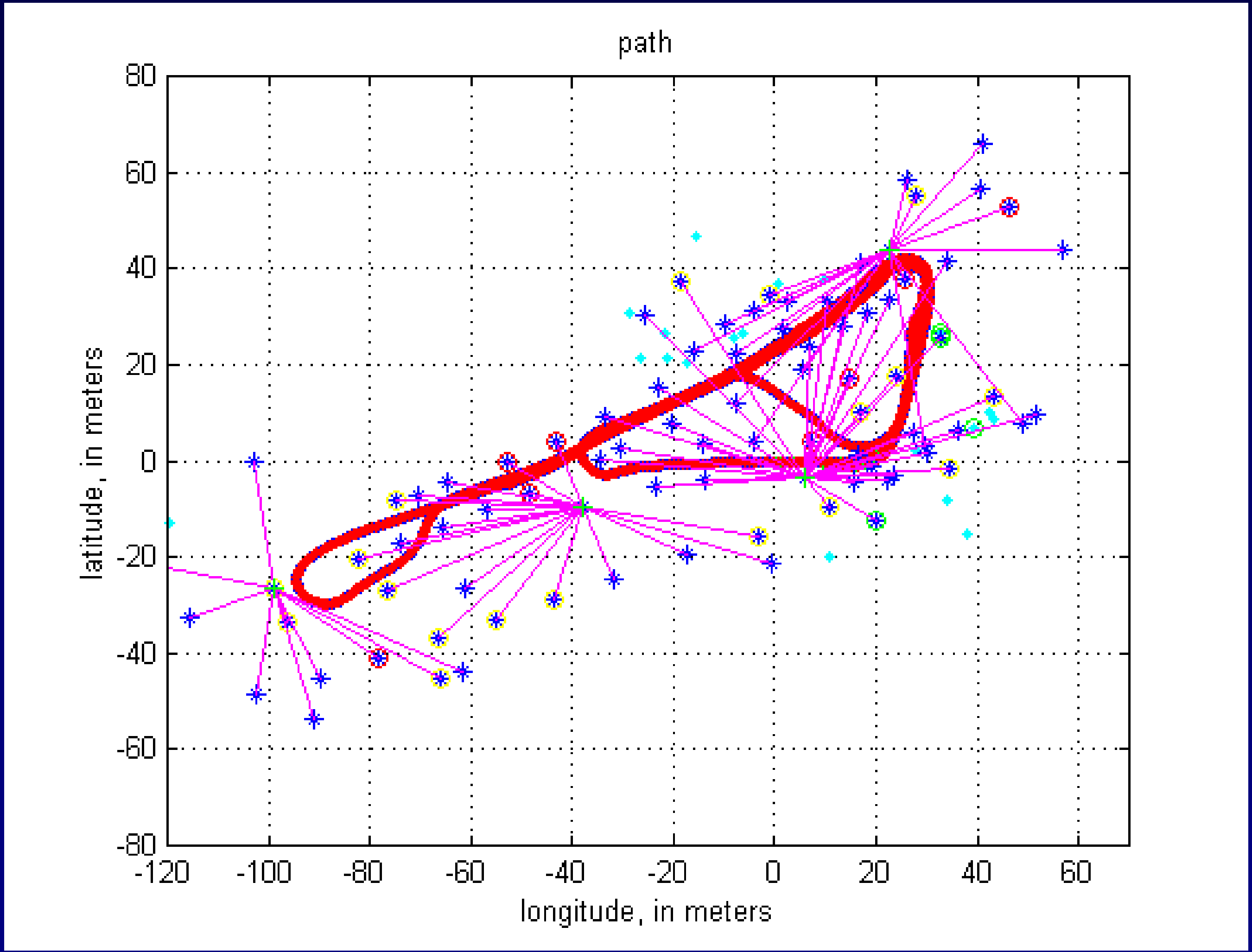


Difference in orientation estimation

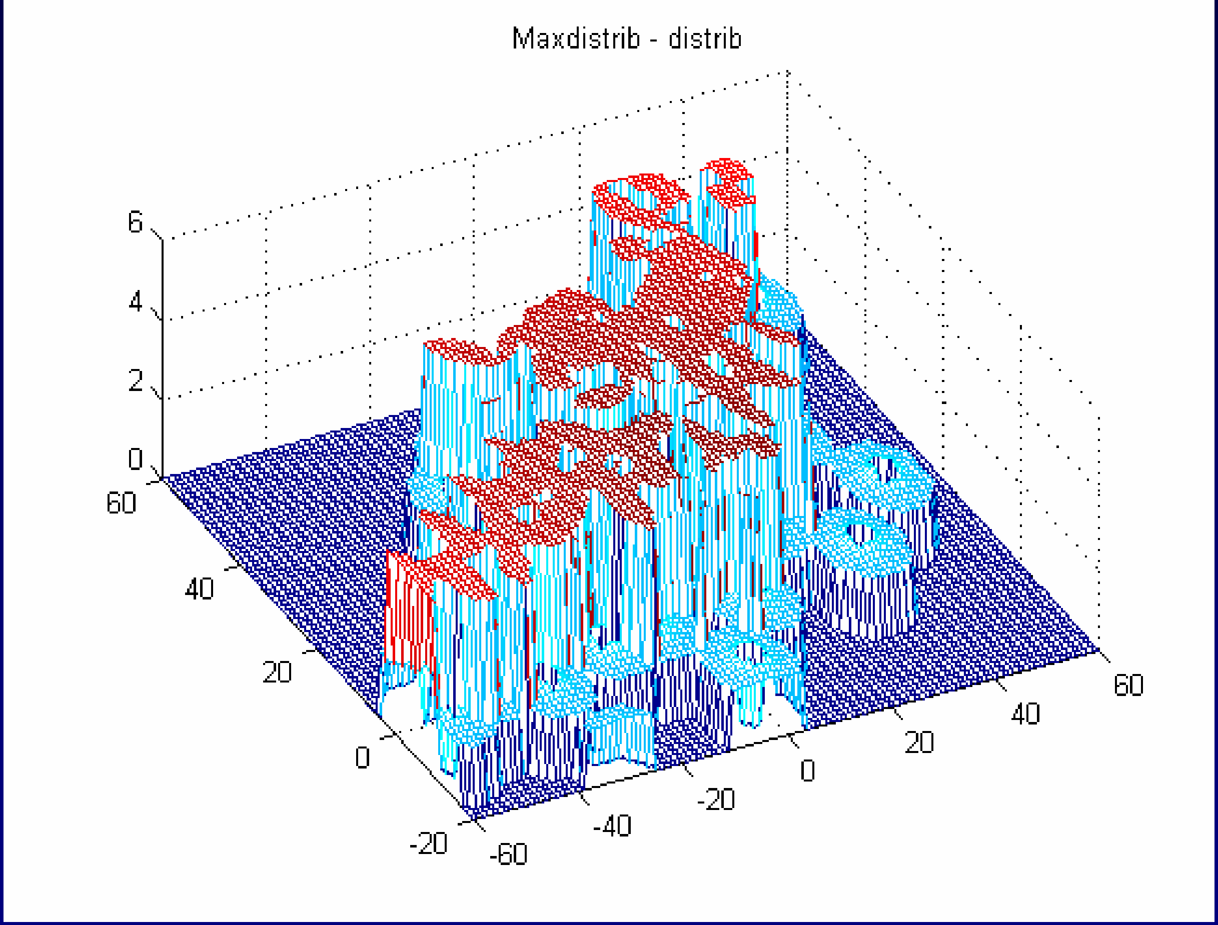


Exploration

Map (section of Victoria Park)



Information maps



Example, using laser observations

$$P_0 = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix}, \quad M = 1000$$

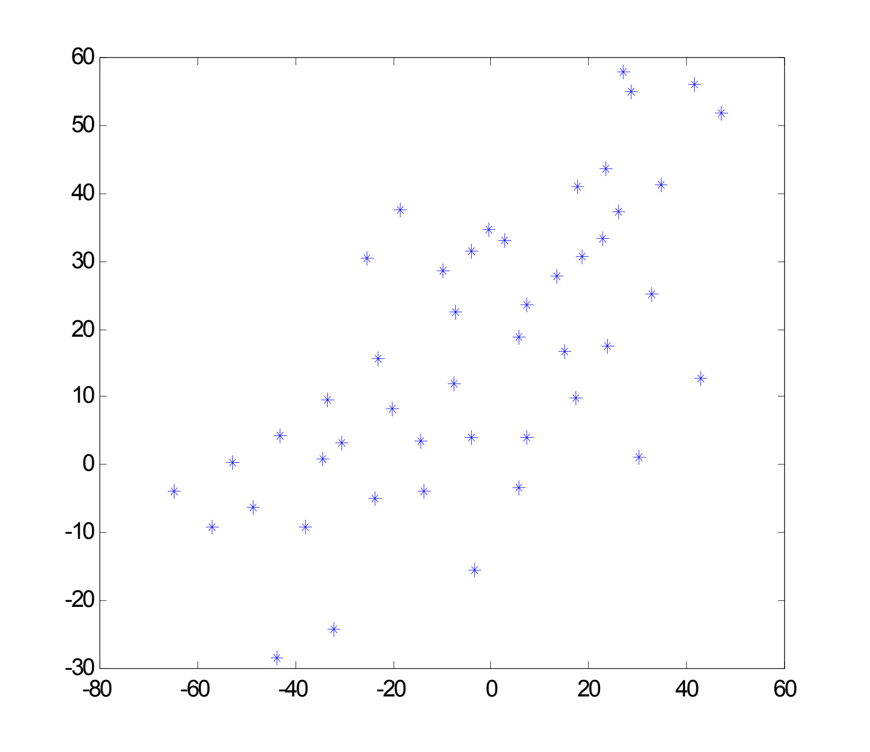
Using omni-directional sensors

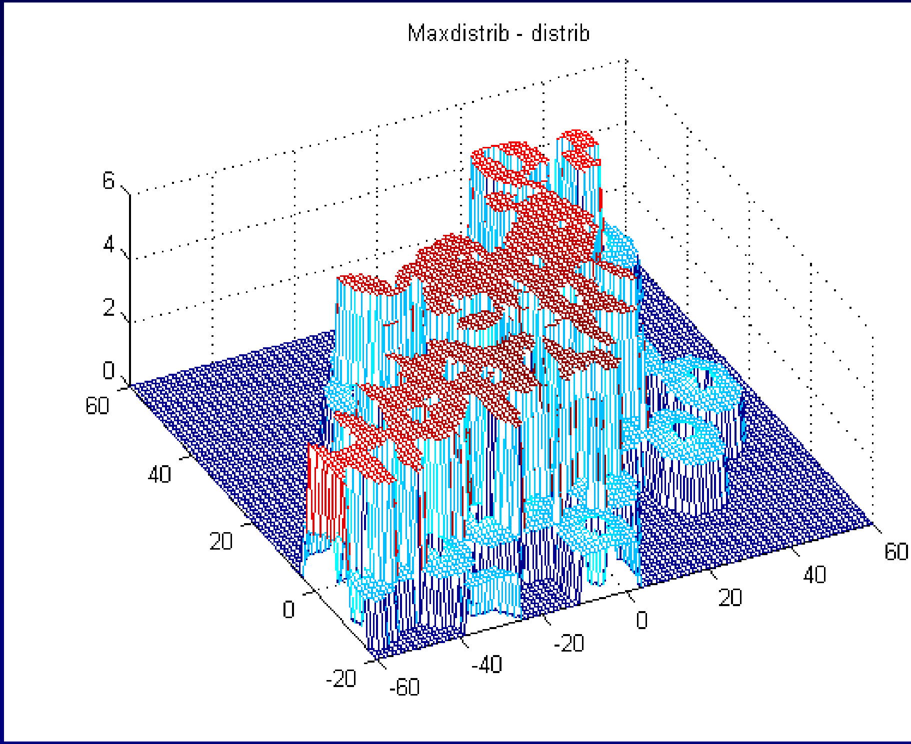
$$f(x, y, \varphi) = F(x, y)$$

$$f(x, y, \varphi) = \det(P_0 - \Delta P)$$

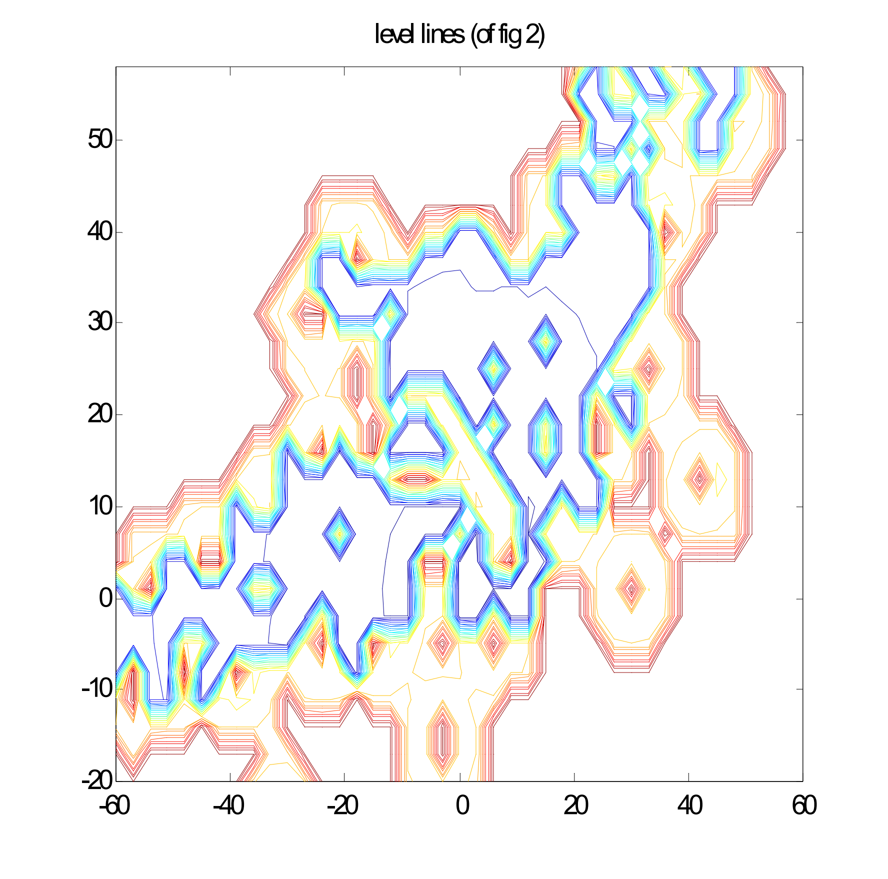
(360º laser or two 180º lasers)

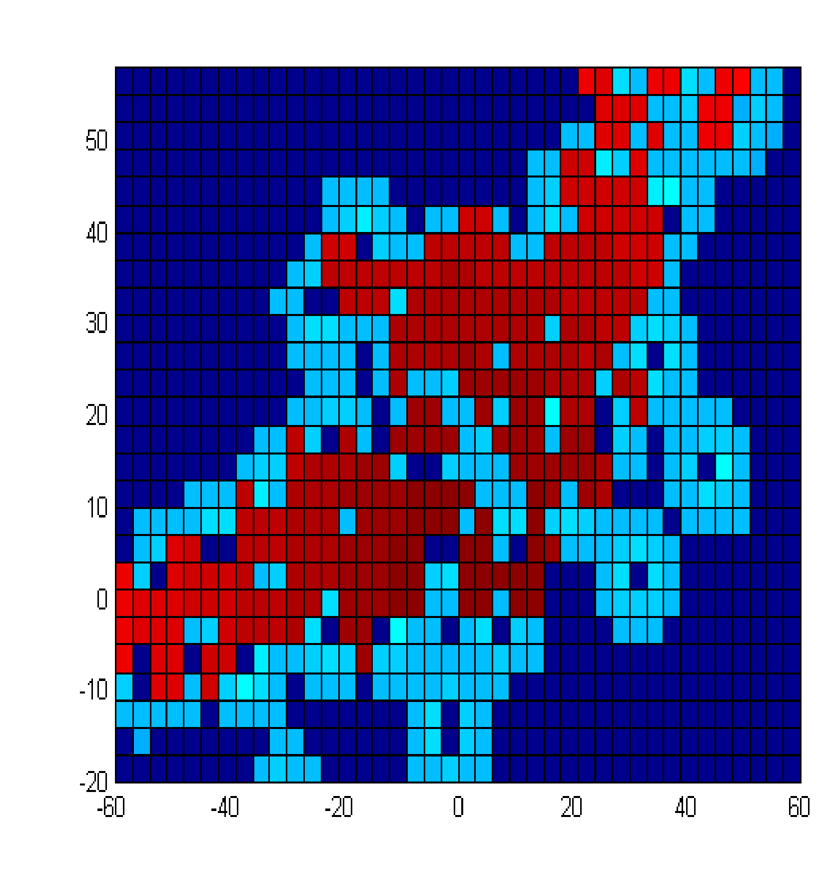
Original map and Information Map





Level sets


level lines (of fig 2)
50
40
30
20
10
0
-10
-20
-60
-40
-20
0
20
40
60


50
40
30
20
10
0
-10
-20
-60
-40
-20
0
20
40
60

SLAM Research

- **Robust SLAM Implementation**
  - Incorporation Absolute / Relative Information
  - Closing Large Loops
  - Link control with navigation
  - Extension to 3-D (Inertial / 3D sensors)
- **Multiple Vehicle problem**
- **Environment Representation**
  - Non-feature based representation
  - Terrain Description
  - Generic definition of Terrain Features
- **All terrain Navigation**
  - Dead Reckoning from external information
  - Incorporation of 3D sensory information
  - Inertial aided with relative information

Labs

- **SLAM code**
  - GPS (running for the first few seconds to obtain heading)
  - Feature Incorporation (validation)
- **Victoria Data**
  - ViewLsr (Return range and bearing to trees)
  - Use same vehicle models

END