Localisation, Mapping and the Simultaneous Localisation and Mapping (SLAM) Problem

Hugh Durrant-Whyte
Australian Centre for Field Robotics
The University of Sydney
hugh@acfr.usyd.edu.au

Introduction

- SLAM asks the following question:

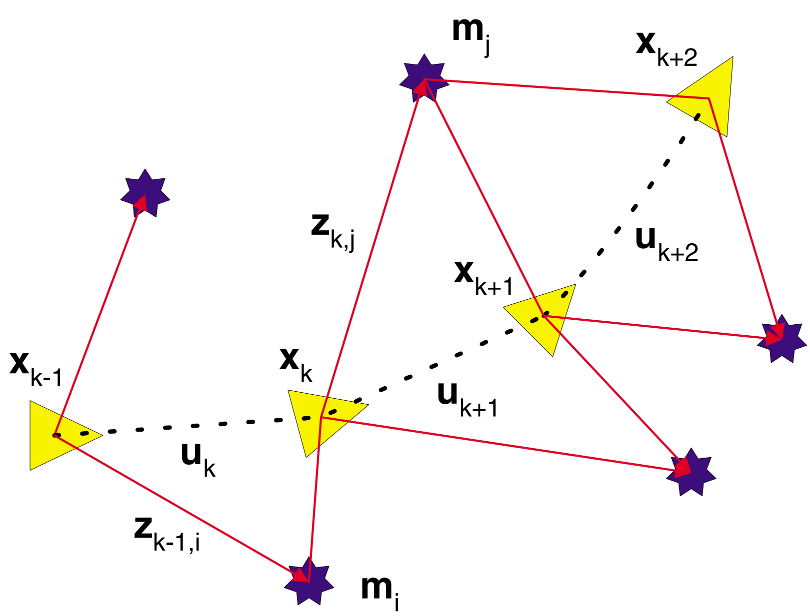
  "Is it possible for an autonomous vehicle to start in an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute vehicle location ?"

- 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.

Overview

1. Definition of the Localisation and Mapping Problem
2. Models of Sensors, Processes and Uncertainty
3. An EKF implementation of the localisation process
4. A short history of the SLAM problem
5. The essential EKF SLAM problem

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 $\mathbf{m}$.

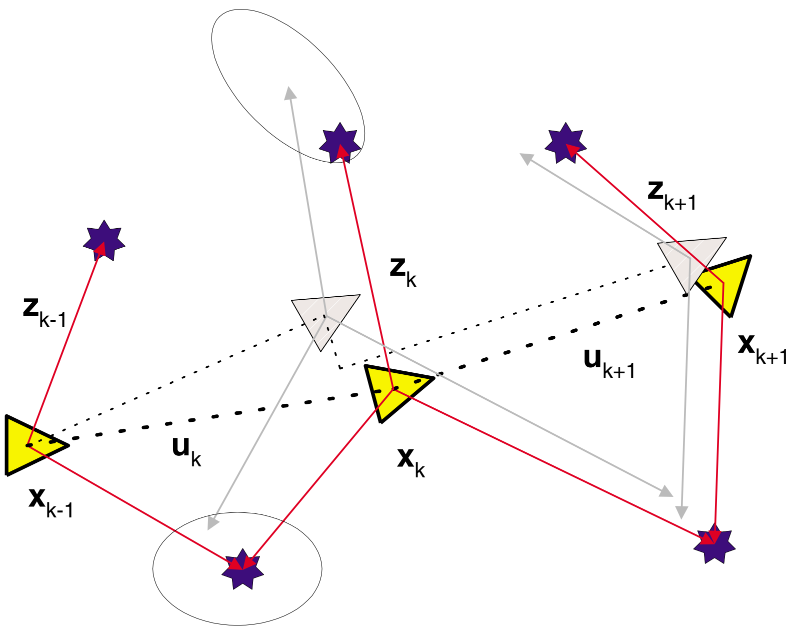
- Prior knowledge (a map) can be incorporated.
- Independent knowledge (inertial/GPS, for example) may also be used.

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 $\mathbf{X}^k$

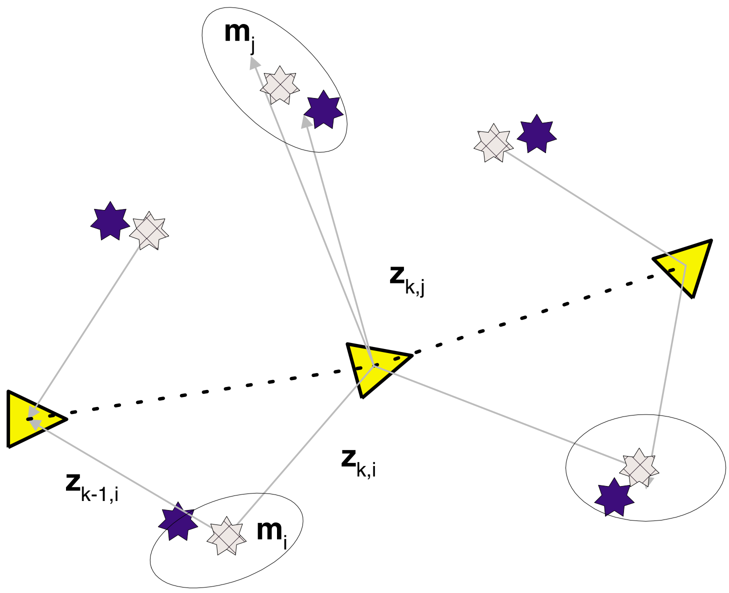
The Localisation Problem



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



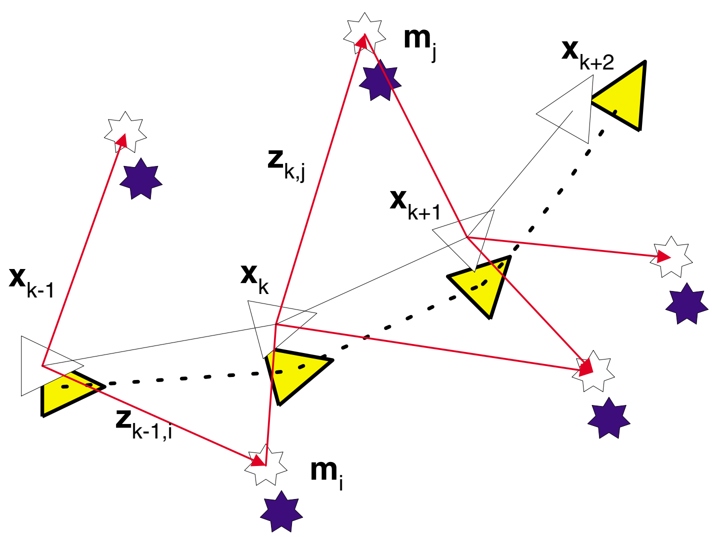
The Simultaneous Localisation and Mapping Problem

- 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

- Uncertainty lies at the heart of any inference and/or estimation problem
- Probabilistic models of sensing and motion are the most widely used method of quantifying 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

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

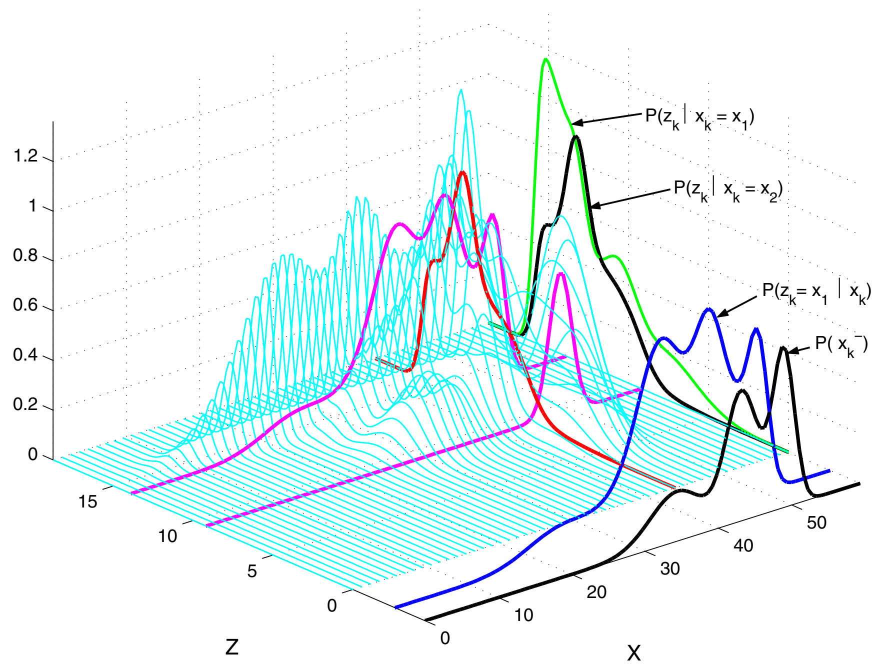
- and the observation

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

- Rearranging:

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

Observation Update Step



Time Update Step

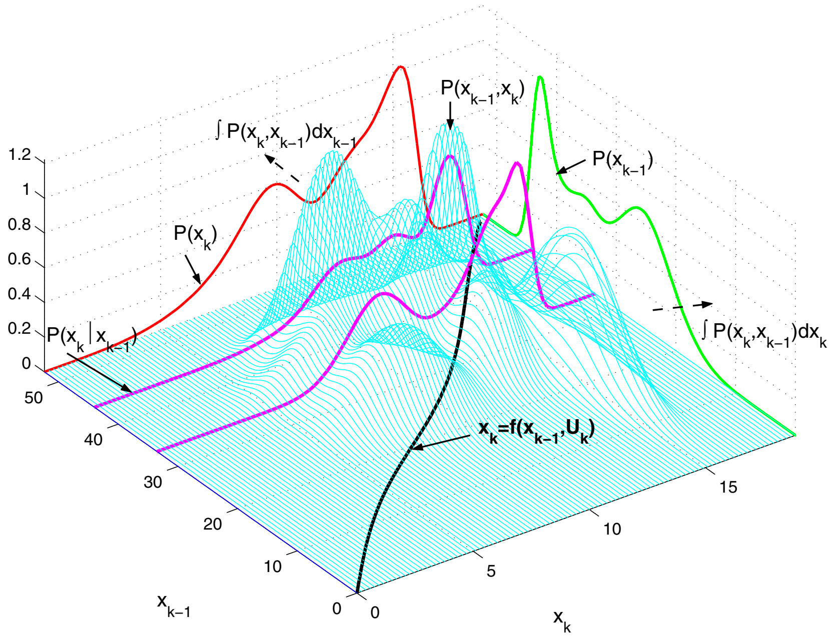
- Assume vehicle model is Markov:

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

- Then (Total Probability Theorem)

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

Time Update Step



Complete Recursive Calculation

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

The Essential Extended Kalman Filter (EKF)

- The Extended Kalman filter (EKF) is a linear recursive estimator for systems described by non-linear process models and/or observation models.
- The EKF is by far the most widely used algorithm for problems in localisation, mapping, and navigation (in an aerospace sense).
- The EKF is the basis for most current SLAM algorithms.

- The EKF employs analytic models of vehicle motion and observation.
- The EKF assumes a motion error distribution which is unimodal and has zero mean
- The EKF assumes an observation error distribution which is also unimodal and has zero mean
- Various other assumptions are added which make people nervous.
- However, the EKF works well and has been proved successful in a wide range of applications.

- Goal: Develop and Introduce the essential EKF localisation problem.

System and Observation Models

- Non-linear discrete-time state transition equation

$$\mathbf{x}(k) = \mathbf{f}\left(\mathbf{x}(k-1), \mathbf{u}(k), k\right) + \mathbf{v}(k),$$

- Non-linear observation equation in the form

$$\mathbf{z}(k) = \mathbf{h}\left(\mathbf{x}(k)\right) + \mathbf{w}(k)$$

- Errors are assumed zero mean

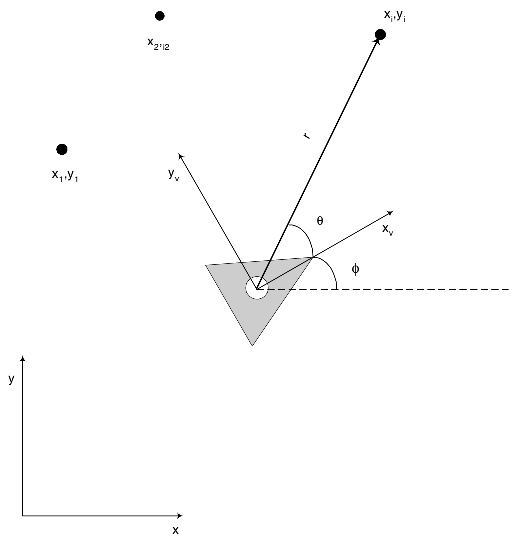
$$\mathrm{E}[\mathbf{v}(k)] = \mathrm{E}[\mathbf{w}(k)] = \mathbf{0}, \qquad \forall k,$$

- and temporally uncorrelated

$$\mathrm{E}[\mathbf{v}(i)\mathbf{v}^T(j)] = \delta_{ij}\mathbf{Q}(i), \quad \mathrm{E}[\mathbf{w}(i)\mathbf{w}(j)] = \delta_{ij}\mathbf{R}(i).$$

$$\mathrm{E}[\mathbf{v}(i)\mathbf{w}^T(j)] = \mathbf{0}, \qquad \forall i, j.$$

Example System and Observation Models



- Vehicle state: $\mathbf{x}(k) = [x(k), y(k), \phi(k)]^T$.
- Vehicle control $\mathbf{u}(k) = [V(k), \psi(k)]^T$.

- Vehicle motion:

$$\begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + TV(k)\cos(\phi(k-1) + \psi(k)) \\ y(k-1) + TV(k)\sin(\phi(k-1) + \psi(k)) \\ \phi(k-1) + T\frac{V(k)}{B}\sin(\psi(k)) \end{bmatrix} + \begin{bmatrix} q_x(k) \\ q_y(k) \\ q_\phi(k) \end{bmatrix}$$

- Sensor: range and bearing to landmarks $\mathbf{B}_i = [X_i, Y_i]^T$, $i = 1, \cdots, N$
- Measurement model:

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

State Prediction

- Assume an estimate at time $k-1$ which is approximately equal to the conditional mean,

$$\hat{\mathbf{x}}(k-1 \mid k-1) \approx \mathrm{E}[\mathbf{x}(k-1) \mid Z^{k-1}]$$

- Find a prediction $\hat{\mathbf{x}}(k \mid k-1)$ based on this.
- Expand state model as a Taylor series about the prediction $\hat{\mathbf{x}}(k-1 \mid k-1)$:

$$\begin{aligned}\mathbf{x}(k) &= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) + \nabla\mathbf{f}_{\mathbf{x}}(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] \\ &\quad + O\left(\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right]^2\right) + \mathbf{v}(k)\end{aligned}$$

- $\nabla\mathbf{f}_{\mathbf{x}}(k)$ is the Jacobian of $\mathbf{f}$ evaluated at $\mathbf{x}(k-1) = \hat{\mathbf{x}}(k-1 \mid k-1)$.
- Truncating expansion at first order, and taking expectations conditioned gives

$$\begin{aligned}\hat{\mathbf{x}}(k \mid k-1) &= \mathrm{E}[\mathbf{x}(k) \mid \mathbf{Z}^{k-1}] \\ &\approx \mathrm{E}\left[\mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right)\right. \\ &\quad \left. + \nabla\mathbf{f}_{\mathbf{x}}(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] + \mathbf{v}(k) \mid \mathbf{Z}^{k-1}\right] \\ &= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right)\end{aligned}$$

Covariance Prediction

- State estimate error $\tilde{\mathbf{x}}(i \mid j) \triangleq \mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j)$
- State covariance $\mathbf{P}(i \mid j) \triangleq \mathrm{E}\{\tilde{\mathbf{x}}(i \mid j)\tilde{\mathbf{x}}^T(i \mid j) \mid \mathbf{Z}^j\}$
- State prediction error:

$$\begin{aligned}\tilde{\mathbf{x}}(k \mid k-1) &= \mathbf{x}(k) - \hat{\mathbf{x}}(k \mid k-1) \\ &= \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) + \nabla\mathbf{f}_{\mathbf{x}}(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] \\ &\quad + O\left(\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right]^2\right) + \mathbf{v}(k) - \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k), k\right) \\ &\approx \nabla\mathbf{f}_{\mathbf{x}}(k)\left[\mathbf{x}(k-1) - \hat{\mathbf{x}}(k-1 \mid k-1)\right] + \mathbf{v}(k) \\ &= \nabla\mathbf{f}_{\mathbf{x}}(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k)\end{aligned}$$

- State covariance:

$$\begin{aligned}\mathbf{P}(k \mid k-1) &\triangleq \mathrm{E}\{\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^{k-1}\} \\ &\approx \mathrm{E}\{\left(\nabla\mathbf{f}_{\mathbf{x}}(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k)\right)\left(\nabla\mathbf{f}_{\mathbf{x}}(k)\tilde{\mathbf{x}}(k-1 \mid k-1) + \mathbf{v}(k)\right)^T \mid \mathbf{Z}^{k-1}\} \\ &= \nabla\mathbf{f}_{\mathbf{x}}(k)\mathrm{E}\{\tilde{\mathbf{x}}(k-1 \mid k-1)\tilde{\mathbf{x}}^T(k-1 \mid k-1) \mid \mathbf{Z}^{k-1}\}\ \nabla^T\mathbf{f}_{\mathbf{x}}(k) + \mathrm{E}\{\mathbf{v}(k)\mathbf{v}^T(k)\} \\ &= \nabla\mathbf{f}_{\mathbf{x}}(k)\mathbf{P}(k-1 \mid k-1)\nabla^T\mathbf{f}_{\mathbf{x}}(k) + \mathbf{Q}(k)\end{aligned}$$

Prediction Example

- Mobile vehicle example:
- Previous estimate: $\hat{\mathbf{x}}(k-1 \mid k-1) = [\hat{x}(k-1 \mid k-1), \hat{y}(k-1 \mid k-1), \hat{\phi}(k-1 \mid k-1)]^T$
- Applied control $\mathbf{u}(k) = [V(k), \psi(k)]^T$
- Predicted state:

$$\begin{bmatrix} \hat{x}(k \mid k-1) \\ \hat{y}(k \mid k-1) \\ \hat{\phi}(k \mid k-1) \end{bmatrix} = \begin{bmatrix} \hat{x}(k-1 \mid k-1) + TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ \hat{y}(k-1 \mid k-1) + TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ \hat{\phi}(k-1 \mid k-1) + T\frac{V(k)}{B}\sin(\psi(k)) \end{bmatrix}.$$

- Jacobian evaluated at $\mathbf{x}(k) = \hat{\mathbf{x}}(k-1 \mid k-1)$

$$\nabla \mathbf{f}_{\mathbf{x}}(k) = \begin{bmatrix} 1 & 0 & -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ 0 & 1 & +TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k)) \\ 0 & 0 & 1 \end{bmatrix}$$

- Assume (for simplicity) $\mathbf{P}(k-1 \mid k-1)$ is diagonal, with $\mathbf{P}(k-1 \mid k-1) = diag\{\sigma_x^2, \sigma_y^2, \sigma_\phi^2, \}$,
- and that the process noise covariance is also diagonal $\mathbf{Q}(k) = diag\{q_x^2, q_y^2, q_\phi^2, \}$.

Prediction Example

- Prediction covariance:

$$\mathbf{P}(k \mid k-1) = \begin{bmatrix} \sigma_x^2 + T^2V^2(k)\sin^2(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 + q_x^2 & -T^2V^2(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 & -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ -T^2V^2(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 & \sigma_y^2 + T^2V^2(k)\cos^2(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 + q_y^2 & TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 \\ -TV(k)\sin(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 & TV(k)\cos(\hat{\phi}(k-1 \mid k-1) + \psi(k))\sigma_\phi^2 & \sigma_\phi^2 + q_\phi^2 \end{bmatrix}$$

- Along-path prediction error dependent only on position uncertainty
- Cross-path prediction error dependent on orientation uncertainty, and increases in proportion to distance traveled.

Observation Prediction and Innovation

- Predicted observation
- Expand observation equation as a Taylor series about the state prediction $\hat{\mathbf{x}}(k \mid k-1)$

$$\begin{aligned}\mathbf{z}(k) &= \mathbf{h}\left(\mathbf{x}(k)\right) + \mathbf{w}(k) \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] \\ &\quad + O\left(\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right]^2\right) + \mathbf{w}(k)\end{aligned}$$

- $\nabla\mathbf{h}_{\mathbf{x}}(k)$ is the Jacobian of $\mathbf{h}$ evaluated at $\mathbf{x}(k) = \hat{\mathbf{x}}(k \mid k-1)$
- Truncating at first order, and taking expectations:

$$\begin{aligned}\hat{\mathbf{z}}(k \mid k-1) &\triangleq \mathrm{E}\{\mathbf{z}(k) \mid \mathbf{Z}^{k-1}\} \\ &\approx \mathrm{E}\{\mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k) \mid \mathbf{Z}^{k-1}\} \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right)\end{aligned}$$

Observation Prediction and Innovation

- Innovation

$$\nu(k) = \mathbf{z}(k) - \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right)$$

- Predicted observation error

$$\begin{aligned}\tilde{\mathbf{z}}(k \mid k-1) &\triangleq \mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1) \\ &= \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) + \nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] \\ &\quad + O\left(\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right]^2\right) + \mathbf{w}(k) \\ &\quad - \mathbf{h}\left(\hat{\mathbf{x}}(k \mid k-1)\right) \\ &\approx \nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k)\end{aligned}$$

- Note distinction between the 'estimated' observation error $\tilde{\mathbf{z}}(k \mid k-1)$ and the actual or measured observation error, the innovation, $\nu(k)$.
- Squaring observation error and taking expectation conditions

$$\begin{aligned}\mathbf{S}(k) &= \mathrm{E}\{\tilde{\mathbf{z}}(k \mid k-1)\tilde{\mathbf{z}}^T(k \mid k-1)\} \\ &= \mathrm{E}\{\left(\nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k)\right)\left(\nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{w}(k)\right)^T\} \\ &= \nabla\mathbf{h}_{\mathbf{x}}(k)\mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k) + \mathbf{R}(k)\end{aligned}$$

Example Observation Prediction and Innovation

- Assume a predicted vehicle location, $\hat{\mathbf{x}}(k \mid k-1) = [\hat{x}(k \mid k-1), \hat{y}(k \mid k-1), \hat{\phi}(k \mid k-1)]^T$.
- Predict observations

$$\begin{bmatrix} \hat{z}_r^i(k \mid k-1) \\ \hat{z}_\theta^i(k \mid k-1) \end{bmatrix} = \begin{bmatrix} \sqrt{(X_i - \hat{x}(k \mid k-1))^2 + (Y_i - \hat{y}(k \mid k-1))^2} \\ \arctan\left(\frac{Y_i - \hat{y}(k|k-1)}{X_i - \hat{x}(k|k-1)}\right) - \hat{\phi}(k \mid k-1) \end{bmatrix}$$

- Jacobian evaluated at $\hat{\mathbf{x}}(k \mid k-1)$

$$\nabla \mathbf{h}_{\mathbf{x}}(k) = \begin{bmatrix} \frac{\hat{x}(k|k-1) - X_i}{d} & \frac{\hat{y}(k|k-1) - Y_i}{d} & 0 \\ -\frac{\hat{y}(k|k-1) - Y_i}{d^2} & \frac{\hat{x}(k|k-1) - X_i}{d^2} & -1 \end{bmatrix},$$

- where $d = \sqrt{(X_i - \hat{x}(k \mid k-1))^2 + (Y_i - \hat{y}(k \mid k-1))^2}$ is the predicted distance between vehicle and beacon.

Example Observation Prediction and Innovation

- Assume prediction covariance matrix $\mathbf{P}(k \mid k-1)$ is diagonal (not true in practice)
- with $\mathbf{P}(k \mid k-1) = diag\{\sigma_x^2, \sigma_y^2, \sigma_\phi^2, \}$,
- And observation noise covariance also diagonal $\mathbf{R}(k) = diag\{r_r^2, r_\theta^2\}$.
- Innovation covariance:

$$\mathbf{S}(k) = \frac{1}{d^2} \begin{bmatrix} (X_i - \hat{x}(k \mid k-1))^2 \sigma_x^2 + (Y_i - \hat{y}(k \mid k-1))^2 \sigma_y^2 + r_r^2 & (X_i - \hat{x}(k \mid k-1))(Y_i - \hat{y}(k \mid k-1)) \left(\sigma_y^2 - \sigma_x^2\right)/d \\ (X_i - \hat{x}(k \mid k-1))(Y_i - \hat{y}(k \mid k-1)) \left(\sigma_y^2 - \sigma_x^2\right)/d & (Y_i - \hat{y}(k \mid k-1))^2 \sigma_x^2 / d^2 + (X_i - \hat{x}(k \mid k-1))^2 \sigma_y^2 / d^2 + d^2(\sigma_\phi^2 + r_\theta^2) \end{bmatrix}$$

- Orientation error does not affect range innovation, only bearing innovation (only when the location to be estimated and the sensor location coincide).

Update Equations

- Find a recursive linear estimator for $\mathbf{x}(k)$
- Assume a prediction $\hat{\mathbf{x}}(k \mid k-1)$ and an observation $\mathbf{z}(k)$
- Assume estimator in the form of an unbiased average of the prediction and innovation

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right].$$

- As before, find an appropriate gain matrix $\mathbf{W}(k)$ which minimizes conditional mean-squared estimation error.
- Estimation error

$$\begin{aligned}
\tilde{\mathbf{x}}(k \mid k) &= \hat{\mathbf{x}}(k \mid k) - \mathbf{x}(k) \\
&= [\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)] + \mathbf{W}(k)\left[\mathbf{h}(\mathbf{x}(k)) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right] + \mathbf{W}(k)\mathbf{w}(k) \\
&\approx [\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)] - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\left[\hat{\mathbf{x}}(k \mid k-1) - \mathbf{x}(k)\right] + \mathbf{W}(k)\mathbf{w}(k) \\
&= \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\right]\tilde{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\mathbf{w}(k)
\end{aligned}$$

Update Equations

- Covariance

$$\begin{aligned}
\mathbf{P}(k \mid k) &\triangleq \mathrm{E}\{\tilde{\mathbf{x}}(k \mid k)\tilde{\mathbf{x}}^T(k \mid k) \mid \mathbf{Z}^k\} \\
&\approx \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\right]\mathrm{E}\{\tilde{\mathbf{x}}(k \mid k-1)\tilde{\mathbf{x}}^T(k \mid k-1) \mid \mathbf{Z}^{k-1}\}\left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\right]^T \\
&\quad + \mathbf{W}(k)\mathrm{E}\{\mathbf{w}(k)\mathbf{w}^T(k)\}\,\mathbf{W}^T(k) \\
&\approx \left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\right]\mathbf{P}(k \mid k-1)\left[\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k)\right]^T + \mathbf{W}(k)\mathbf{R}(k)\mathbf{W}^T(k)
\end{aligned}$$

- Gain matrix $\mathbf{W}(k)$ is chosen to minimize mean squared estimation error

$$L(k) = \mathrm{E}[\tilde{\mathbf{x}}^T(k \mid k)\tilde{\mathbf{x}}(k \mid k)] = \mathrm{trace}[\mathbf{P}(k \mid k)].$$

- Differentiate and set equal to zero

$$\frac{\partial L}{\partial \mathbf{W}(k)} = -2(\mathbf{I} - \mathbf{W}(k)\nabla\mathbf{h}_{\mathbf{x}}(k))\mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k) + 2\mathbf{W}(k)\mathbf{R}(k) = \mathbf{0}.$$

- Rearranging gives

$$\begin{aligned}
\mathbf{W}(k) &= \mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k)\left[\nabla\mathbf{h}_{\mathbf{x}}(k)\mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k) + \mathbf{R}(k)\right]^{-1} \\
&= \mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k)\mathbf{S}^{-1}(k)
\end{aligned}$$

Summary

- Prediction:

$$\hat{\mathbf{x}}(k \mid k-1) = \mathbf{f}\left(\hat{\mathbf{x}}(k-1 \mid k-1), \mathbf{u}(k)\right)$$

$$\mathbf{P}(k \mid k-1) = \nabla\mathbf{f}_{\mathbf{x}}(k)\mathbf{P}(k-1 \mid k-1)\nabla^T\mathbf{f}_{\mathbf{x}}(k) + \mathbf{Q}(k)$$

- Update:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}(k \mid k-1))\right]$$

$$\mathbf{P}(k \mid k) = \mathbf{P}(k \mid k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k)$$

- where

$$\mathbf{W}(k) = \mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k)\mathbf{S}^{-1}(k)$$

$$\mathbf{S}(k) = \nabla\mathbf{h}_{\mathbf{x}}(k)\mathbf{P}(k \mid k-1)\nabla^T\mathbf{h}_{\mathbf{x}}(k) + \mathbf{R}(k).$$

Understanding the Extended Kalman Filter

- The extended Kalman filter algorithm is very similar to the linear Kalman filter algorithm,
- with the substitutions $\mathbf{F}(k) \rightarrow \nabla\mathbf{f}_{\mathbf{x}}(k)$ and $\mathbf{H}(k) \rightarrow \nabla\mathbf{h}_{\mathbf{x}}(k)$
- Similarity as filtering on state errors
- EKF works like the KF except
  - The Jacobians $\nabla\mathbf{f}_{\mathbf{x}}(k)$ and $\nabla\mathbf{h}_{\mathbf{x}}(k)$ are not constant, being functions of both state and timestep.
  - Linearised model derived by perturbing the true state around a predicted or nominal trajectory, great care must be taken to ensure that perturbations are sufficiently small
  - The EKF employs a separating model which must be computed from an approximate knowledge of the state. Means that filter must be accurately initialized.

Implementation of Extended Kalman Filter

- Linear approximations for non-linear functions should be treated with care.
- However, the EKF has seen a huge variety of successful applications ranging from missile guidance to process plant control – so it can be made to work.
- Guidelines for the linear Kalman filter apply, with twice the importance, to the extended Kalman Filter:
- Understand your sensor and understand your process;
- Innovation measures provide the main means of analysing filter performance; but these are complicated by changes in covariance, observation and state prediction caused by the dependence of the state model on the states themselves.
- Testing of an EKF requires the consideration of rather more cases than is required in the linear filter.
- Detecting modeling errors is much more difficult for an EKF. Use of a "truth model" is a good method (see Maybeck Chapter 6 for an excellent introduction to the problem of error budgeting)

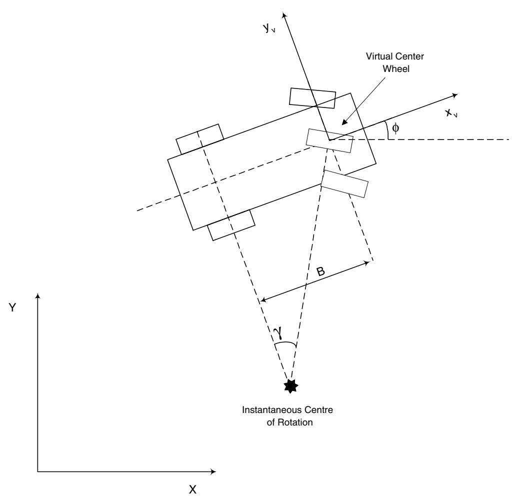
Example Implementation of the Extended Kalman Filter

- A land-mark based navigation system developed for an Autonomous Guided Vehicle (AGV).
- High speed, variable terrain.
- Detailed modeling of process, observation and error models

The Process Model

- Description of the nominal motion of the vehicle
- Description of uncertainties arise in prediction:
  - Errors in drive velocity (slipping),
  - Errors in steer traction (skidding),
  - Changes in effective wheel radius,
  - Errors vary with vehicle state.
- Procedure:
  - Develop a "nominal" process model
  - Model of how errors in velocity, steer, wheel radius and previous state values are propagated through time.
  - A linearised model of the error propagation equations to provide equations for state estimate covariance propagation.

Nominal Process Model



Nominal Process Model

- Bicycle model:

$$
\begin{aligned}
\dot{x}(t) &= R(t)\omega(t)\cos(\phi(t)+\gamma(t)) \\
\dot{y}(t) &= R(t)\omega(t)\sin(\phi(t)+\gamma(t)) \\
\dot{\phi}(t) &= \frac{R(t)\omega(t)}{B}(\sin\gamma(t)) \\
\dot{R}(t) &= 0,
\end{aligned}
$$

- Vehicle location referenced to the centre of the front axle.
- Control inputs are steer angle γ, and ground speed $V(t) = R(t)\omega(t)$ of the front wheel.
- Ground speed set equal to rotational wheel speed $\omega(t)$ (a measured quantity) multiplied by the wheel radius $R(t)$.
- Makes explicit wheel radius variations.

Nominal Process Model

- Convert to discrete-time state transition equation.
- Assume a synchronous sampling interval ΔT for both drive and steer encoders
- Approximate all derivatives by first-order forward differences,
- All control signals assumed approximately constant over the sample period,
- All continuous times replaced with discrete time index $t = k\Delta T \stackrel{\triangle}{=} k$.
- Then

$$
\begin{aligned}
x(k+1) &= x(k) + \Delta T R(k)\omega(k)\cos\left[\phi(k)+\gamma(k)\right] \\
y(k+1) &= y(k) + \Delta T R(k)\omega(k)\sin\left[\phi(k)+\gamma(k)\right] \\
\phi(k+1) &= \phi(k) + \Delta T\frac{R(k)\omega(k)}{B}\sin\gamma(k) \\
R(k+1) &= R(k).
\end{aligned}
$$

Nominal Process Model

- State vector at a time k:

$$\mathbf{x}(k) = [x(k), y(k), \phi(k), R(k)]^T,$$

- Control vector:

$$\mathbf{u}(k) = [\omega(k), \gamma(k)]^T$$

- Nominal (error-free) state transition:

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right)$$

Prediction

- During operation, the true vehicle state $\mathbf{x}(k)$ will never be known.
- Instead, an estimate of the state is computed.
- Estimate of state $\mathbf{x}(k)$:

$$\hat{\mathbf{x}}^+(k) = [\hat{x}^+(k), \hat{y}^+(k), \hat{\phi}^+(k), \hat{R}^+(k)]^T$$

- Mean measured (from encoders) value of the true control vector $\mathbf{u}(k)$:

$$\overline{\mathbf{u}}(k) = [\overline{\omega}(k), \overline{\gamma}(k),]^T$$

- generate a prediction $\hat{\mathbf{x}}^-(k+1)$,

$$\hat{\mathbf{x}}^-(k+1) = [\hat{x}^-(k+1), \hat{y}^-(k+1), \hat{\phi}^-(k+1), \hat{R}^-(k+1)]^T,$$

- of the true state $\mathbf{x}(k+1)$ at time $k+1$ as

$$\hat{\mathbf{x}}^-(k+1) = \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right)$$

Error Prediction Model

- Drive error is modeled as a combination of additive disturbance and multiplicative (slip) error:

$$\omega(k) = \overline{\omega}(k)\,[1 + \delta q(k)] + \delta\omega(k),$$

- $\overline{\omega}(k)$is mean measured wheel rotation and $\omega(k)$ is defined as true mean wheel rotation rate.
- Steer error is modeled as a combination of additive disturbance and multiplicative (skid) error:

$$\gamma(k) = \overline{\gamma}(k)\,[1 + \delta s(k)] + \delta\gamma(k),$$

- $\overline{\gamma}(k)$is mean measured (encoder) steer angle
- Wheel radius error is additive disturbance rate (a random walk):

$$R(k) = \hat{R}^{+}(k) + \Delta T \delta R(k).$$

- Source errors $\delta q(k)$, $\delta\omega(k)$, $\delta s(k)$, $\delta\gamma(k)$, and $\delta R(k)$ are modeled as constant, zero mean, uncorrelated white sequences, with variances σ_q^2, σ_ω^2, σ_s^2, σ_γ^2 and σ_R^2 respectively.

Error Prediction Model

- Error models are designed to capture
  - **Multiplicative**: Increased uncertainty in vehicle motion as speed and steer angles increase (slipping and skidding) actually due to linear and rotational inertial forces acting at the interface between tyre and road.
  - **Additive**: Stationary uncertainty and motion model errors such as axle offsets. Also important to stabilize the estimator algorithm.
  - Random walk model for wheel radius is intended to allow adaptation of the estimator to wheel radius changes caused by uneven terrain and by changes in vehicle load.

Error Propagation Equations

- Error between the true state and estimated state, and between the true state and the prediction are given by

$$\delta\mathbf{x}^+(k) = \mathbf{x}(k) - \hat{\mathbf{x}}^+(k), \text{ and } \qquad \delta\mathbf{x}^-(k+1) = \mathbf{x}(k+1) - \hat{\mathbf{x}}^-(k+1),$$

- Difference between true and measured control input:

$$\delta\mathbf{u}(k) = \mathbf{u}(k) - \overline{\mathbf{u}}(k)$$

- Then

$$\begin{aligned} \delta\hat{\mathbf{x}}^-(k+1) &= \mathbf{x}(k+1) - \hat{\mathbf{x}}^-(k+1) \\ &= \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}(k)\right) - \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right) \\ &= \mathbf{f}\left(\hat{\mathbf{x}}^+(k) + \delta\mathbf{x}^+(k), \overline{\mathbf{u}}(k) + \delta\mathbf{u}(k)\right) - \mathbf{f}\left(\hat{\mathbf{x}}^+(k), \overline{\mathbf{u}}(k)\right) \end{aligned}$$

Error Propagation Equations

- Evaluating this and neglecting all second-order error products

$$\begin{aligned} \delta\hat{x}^-(k+1) = \delta\hat{x}^+(k) &+ \Delta T \cos\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Omega(k) + \varpi(k)\delta R(k)\right] \\ &- \Delta T \sin\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Gamma(k) + \hat{R}^+(k)\varpi(k)\delta\phi(k)\right] \end{aligned}$$

$$\begin{aligned} \delta\hat{y}^-(k+1) = \delta\hat{y}^+(k) &+ \Delta T \sin\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Omega(k) + \varpi(k)\delta R(k)\right] \\ &+ \Delta T \cos\left(\hat{\phi}^+(k) + \overline{\gamma}(k)\right)\left[\delta\Gamma(k) + \hat{R}^+(k)\varpi(k)\delta\phi(k)\right] \end{aligned}$$

$$\begin{aligned} \delta\hat{\phi}^-(k+1) = \delta\hat{\phi}^+(k) &+ \Delta T \tfrac{\sin\overline{\gamma}(k)}{B}\left[\delta\Omega(k) + \varpi(k)\delta R(k)\right] \\ &+ \Delta T \tfrac{\cos\overline{\gamma}(k)}{B}\delta\Gamma(k) \end{aligned}$$

$$\delta\hat{R}^-(k+1) = \delta\hat{R}^+(k) + \Delta T \delta R(k)$$

Error Propagation Equations

- where

$$\delta\Omega(k) = \hat{R}^+(k)\overline{\omega}(k)\delta q(k) + \hat{R}^+(k)\delta\omega(k)$$

is the composite along-track rate error describing control induced error propagation along the direction of travel,

- and

$$\delta\Gamma(k) = \hat{R}^+(k)\overline{\omega}(k)\overline{\gamma}(k)\delta s(k) + \hat{R}^+(k)\overline{\omega}(k)\delta\gamma(k)$$

is the composite cross-track rate error describing control induced error propagation perpendicular to the direction of travel.

Error Transfer Equations

- State error transfer matrix

$$\mathbf{F}(k) = \begin{bmatrix} 1 & 0 & -\Delta T\hat{R}^+(k)\overline{\omega}(k)\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \Delta T\overline{\omega}(k)\cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) \\ 0 & 1 & \Delta T\hat{R}^+(k)\overline{\omega}(k)\cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \Delta T\overline{\omega}(k)\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) \\ 0 & 0 & 1 & \Delta T\overline{\omega}(k)\frac{\sin\overline{\gamma}(k)}{B} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Source error transfer matrix

$$\mathbf{G}(k) = \begin{bmatrix} \cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & -\sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & 0 \\ \sin(\hat{\phi}^+(k)+\overline{\gamma}(k)) & \cos(\hat{\phi}^+(k)+\overline{\gamma}(k)) & 0 \\ \frac{\sin\overline{\gamma}(k)}{B} & \frac{\cos\overline{\gamma}(k)}{B} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- With $\delta\mathbf{w}(k) = [\delta\Omega(k), \delta\Gamma(k), \delta R(k)]^T$
- Then

$$\delta\mathbf{x}^-(k+1) = \mathbf{F}(k)\delta\mathbf{x}^+(k) + \Delta T\mathbf{G}(k)\delta\mathbf{w}(k)$$

Error Transfer Equations

- Define

$$\begin{aligned}\mathbf{P}^-(k+1) &= \mathrm{E}\left[\delta\mathbf{x}^-(k+1)\delta\mathbf{x}^-(k+1)^T\right],\\ \mathbf{P}^+(k) &= \mathrm{E}\left[\delta\mathbf{x}^+(k)\delta\mathbf{x}^+(k)^T\right],\\ \mathbf{\Sigma}(k) &= \mathrm{E}\left[\delta\mathbf{w}(k)\delta\mathbf{w}(k)^T\right],\end{aligned}$$

- Assume $\mathrm{E}[\delta\mathbf{x}^+(k)\delta\mathbf{w}(k)^T] = \mathbf{0}$.
- Square, take expectations: propagation of covariance

$$\mathbf{P}^-(k+1) = \mathbf{F}(k)\mathbf{P}^+(k)\mathbf{F}^T(k) + \Delta T^2\mathbf{G}(k)\mathbf{\Sigma}(k)\mathbf{G}^T(k)$$

- where,

$$\mathbf{\Sigma}(k) = \begin{bmatrix} \left[\hat{R}^+(k)\right]^2\left([\varpi(k)]^2\sigma_q^2 + \sigma_\omega^2\right) & 0 & 0 \\ 0 & \left[\hat{R}^+(k)\varpi(k)\right]^2\left(\overline{\gamma}(k)^2\sigma_s^2 + \sigma_\gamma^2\right) & 0 \\ 0 & 0 & \sigma_R^2 \end{bmatrix}.$$

Observation Model

- Observation of range and bearing made by radar to a number of beacons placed at fixed and known locations in the environment.
- Processing: .
  1. The measurement is converted into a Cartesian observation referenced to the vehicle coordinate system.
  2. The vehicle-centered observation is transformed into base-coordinates using knowledge of the predicted vehicle location at the time the observation was obtained.
  3. The observation is then matched to a map of beacons maintained by the AGV in base-coordinates.
  4. The matched beacon is transformed back into a vehicle centered coordinate system where it is used to update vehicle location according to the standard extended Kalman filter equations.
- Measurements are taken at a discrete time instant k when a prediction of vehicle location is already available.

Observation Model

- Also recall: If two random variables $\mathbf{a}$ and $\mathbf{b}$ are related by the non-linear equation $\mathbf{a} = \mathbf{g}(\mathbf{b})$, then the mean $\overline{\mathbf{a}}$ of $\mathbf{a}$ may be approximated in terms of the mean $\overline{\mathbf{b}}$ of $\mathbf{b}$ by

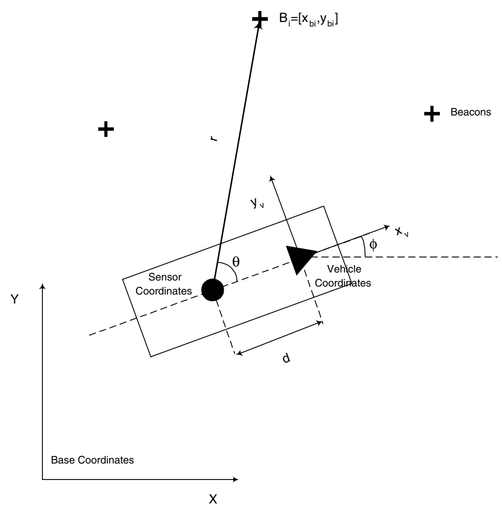
$$\overline{\mathbf{a}} = \mathbf{g}\left(\overline{\mathbf{b}}\right)$$

and that the variance $\mathbf{\Sigma}_a$ of $\mathbf{a}$ may be approximated in terms of the variance $\mathbf{\Sigma}_b$ of $\mathbf{b}$ by

$$\mathbf{\Sigma}_a = \nabla \mathbf{g}_b \mathbf{\Sigma}_b \nabla \mathbf{g}_b^T,$$

where $\nabla \mathbf{g}_b$ is the Jacobian of $\mathbf{g}(\cdot)$ taken with respect to $\mathbf{b}$, evaluated at the mean $\overline{\mathbf{b}}$.

Observation Model



Observation Processing I

- The radar provides observations of range $r(k)$ and bearing $\theta(k)$ to a fixed target in the environment.
- The radar itself is located on the centerline of the vehicle with a longitudinal offset d from the vehicle centered coordinate system.
- The observations $\mathbf{z}_v(k)$, in Cartesian coordinates, referred to the vehicle frame are given by

$$\mathbf{z}_v(k) = \begin{bmatrix} z_{xv}(k) \\ z_{yv}(k) \end{bmatrix} = \begin{bmatrix} d + r(k)\cos\theta(k) \\ r(k)\sin\theta(k) \end{bmatrix}$$

- Assume that errors in range and bearing are Gaussian and uncorrelated with constant variances σ_r^2 and σ_θ^2 respectively.
- Observation variance $\mathbf{\Sigma}_z(k)$ in vehicle coordinates:

$$\mathbf{\Sigma}_z(k) = \begin{bmatrix} \cos\theta(k) & -\sin\theta(k) \\ \sin\theta(k) & \cos\theta(k) \end{bmatrix} \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & r^2\sigma_\theta^2 \end{bmatrix} \begin{bmatrix} \cos\theta(k) & \sin\theta(k) \\ -\sin\theta(k) & \cos\theta(k) \end{bmatrix}$$

Observation Processing II

- Transform vehicle-centered observation into absolute world coordinates so that it can be matched to the map of beacon locations.
- Predicted vehicle location in base coordinates is $[\hat{x}^-(k), \hat{y}^-(k), \hat{\phi}^-(k)]^T$,
- Then observation $\mathbf{z}_b(k)$ in cartesian base coordinates:

$$\mathbf{z}_b(k) = \begin{bmatrix} z_{xb}(k) \\ z_{yb}(k) \end{bmatrix} = \begin{bmatrix} \hat{x}^-(k) + z_{xv}(k)\cos\hat{\phi}^-(k) - z_{yv}(k)\sin\hat{\phi}^-(k) \\ \hat{y}^-(k) + z_{xv}(k)\sin\hat{\phi}^-(k) + z_{yv}(k)\cos\hat{\phi}^-(k) \end{bmatrix}$$

- Observation variance transformed into base coordinates:

$$\mathbf{\Sigma}_b(k) = \mathbf{T}_x(k)\mathbf{P}^-(k)\mathbf{T}_x^T(k) + \mathbf{T}_z(k)\mathbf{\Sigma}_z(k)\mathbf{T}_z^T(k)$$

- where

$$\mathbf{T}_x(k) = \begin{bmatrix} 1 & 0 & -z_{xv}(k)\sin\hat{\phi}^-(k) - z_{yv}(k)\cos\hat{\phi}^-(k) & 0 \\ 0 & 1 & -z_{xv}(k)\cos\hat{\phi}^-(k) - z_{yv}(k)\sin\hat{\phi}^-(k) & 0 \end{bmatrix},$$

- and

$$\mathbf{T}_z(k) = \begin{bmatrix} \cos\hat{\phi}^-(k) & -\sin\hat{\phi}^-(k) \\ \sin\hat{\phi}^-(k) & \cos\hat{\phi}^-(k) \end{bmatrix},$$

and where $\mathbf{P}^-(k)$ is the predicted vehicle state covariance.

Observation Processing III

- Matching (data association) of observations to beacons
- Beacon locations $\mathbf{b}_i = [x_{bi}, y_{bi}], i = 1, \cdots, N$
- Matching gate:

$$(\mathbf{b}_i - \mathbf{z}_b(k))^T \, \mathbf{\Sigma}_b^{-1}(k) \, (\mathbf{b}_i - \mathbf{z}_b(k)) < \alpha.$$

- The gate size α is normally taken to be quite small (0.5) to ensure low false alarm rates.

Observation Processing IV

- Update stage: done in vehicle centered coordinates because of rotation sensitivity.
- Single beacon match $\mathbf{b} = [x_b, y_b]^T$
- Transformed to vehicle coordinates

$$\hat{\mathbf{z}}_v = \begin{bmatrix} \hat{z}_{vx} \\ \hat{z}_{vy} \end{bmatrix} = \begin{bmatrix} \cos\hat{\phi}^-(k) & \sin\hat{\phi}^-(k) \\ -\sin\hat{\phi}^-(k) & \cos\hat{\phi}^-(k) \end{bmatrix} \begin{bmatrix} x_b - \hat{x}^-(k) \\ y_b - \hat{y}^-(k) \end{bmatrix}$$

- Update with the usual equations:

$$\hat{x}^+(k) = \hat{x}^-(k) + \mathbf{W}(k) \left[\mathbf{z}_v(k) - \hat{\mathbf{z}}_v \right],$$

$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k),$$

- where

$$\mathbf{W}(k) = \mathbf{P}^-(k)\mathbf{H}^T(k)\mathbf{S}^{-1}(k),$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}^-(k)\mathbf{H}^T(k) + \mathbf{\Sigma}_z(k),$$

- and

$$\mathbf{H}(k) = \begin{bmatrix} -\cos\hat{\phi}^-(k) & -\sin\hat{\phi}^-(k) & -(x_b - \hat{x}^-(k))\sin\hat{\phi}^-(k) + (y_b - \hat{y}^-(k))\cos\hat{\phi}^-(k) & 0 \\ \sin\hat{\phi}^-(k) & -\cos\hat{\phi}^-(k) & -(x_b - \hat{x}^-(k))\cos\hat{\phi}^-(k) - (y_b - \hat{y}^-(k))\sin\hat{\phi}^-(k) & 0 \end{bmatrix}$$

System Analysis

- Implementation of this navigation system as a detailed example
- Code is implemented in Matlab in two main parts;

1. The specification of a vehicle path, followed by the generation of true vehicle trajectory, true vehicle control inputs and simulated observations of a number of beacons.
2. The filtering of observations with control inputs in an extended Kalman filter to provide estimates of vehicle position, heading and mean wheel radius, together with associated estimation errors.

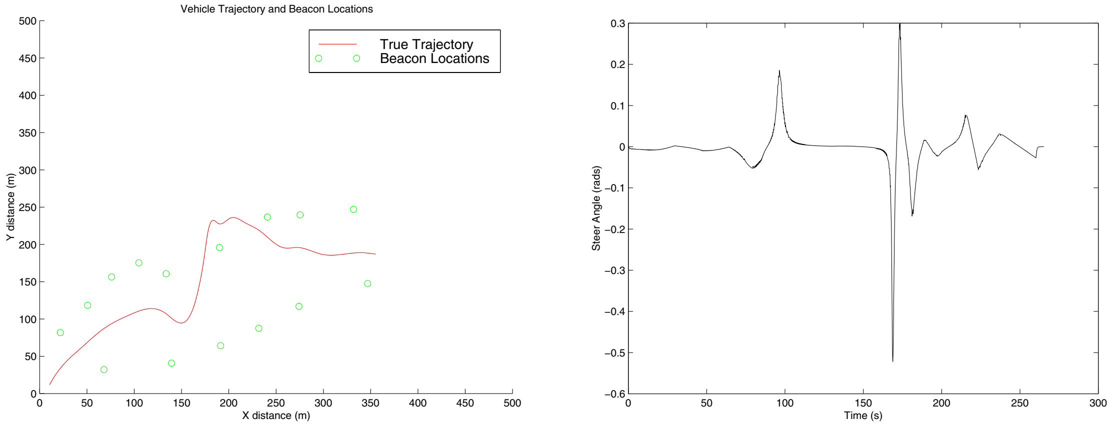
- This structure allows a single vehicle run to be generated and subsequently to evaluate the effect of different filter parameters, initial conditions and injection errors.

Trajectory Generation

- The vehicle trajectory is generated in the following stages:

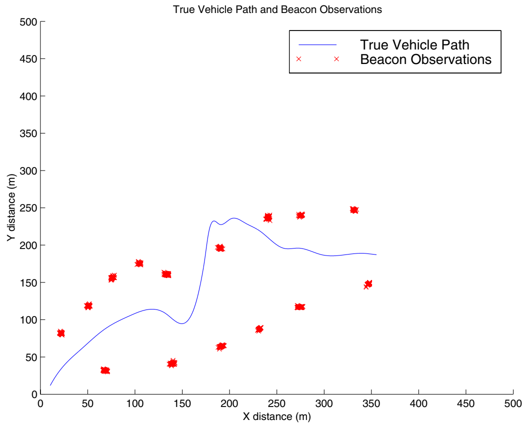
1. A number of spline points are defined. At the same time the beacon locations are defined.
2. A smooth spline curve is fitted to these points.
3. A velocity profile for the vehicle is defined.
4. A proportional control algorithm is used to control the steering angle to maintain the vehicle on-path.
5. The resulting true vehicle location is recorded as it follows the path, together with the steer and speed control inputs.

Trajectory Generation



(a) (b)

Observation Generation



- Sensor scans 360^o at a fixed scan rate of 2Hz.
- Intersection algorithm plus additive noise.

Filter Structure and Initialisation

- Prediction, Observation Matching, and Update.
- Predictions are made synchronously at 10Hz on the basis of true (measured) steer and drive control inputs.
- If there is no beacon observation made in the sample time, then the prediction becomes the estimate at that time.
- If an observation is made, it is matched to one of the beacons in the initial map and a filter update is performed at that time.

- The nominal estimated values for the error source terms were defined as follows:

Multiplicative Slip Error σ_q (%/100):	0.02
Additive Slip Error σ_ω (rads/s):	0.1
Multiplicative Skid Error σ_s (%/100):	0.01
Additive Skid Error σ_γ (rads):	0.035
Wheel Radius Error Rate σ_R (m/s):	0.001
Radar Range Error σ_r (m):	0.3
Radar Bearing Error σ_θ (rads):	0.035

- Performance of the vehicle navigation system is relatively insensitive to specific values of process model errors up to a factor of 2-4, while being quite sensitive to estimated observation errors.

- The navigation is initialised with the initial true vehicle location. The initial position errors are taken as

$$\sqrt{\mathbf{P}(0 \mid 0)} = \begin{bmatrix} 0.3 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.05 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.01 \end{bmatrix}$$

Nominal Filter Performance

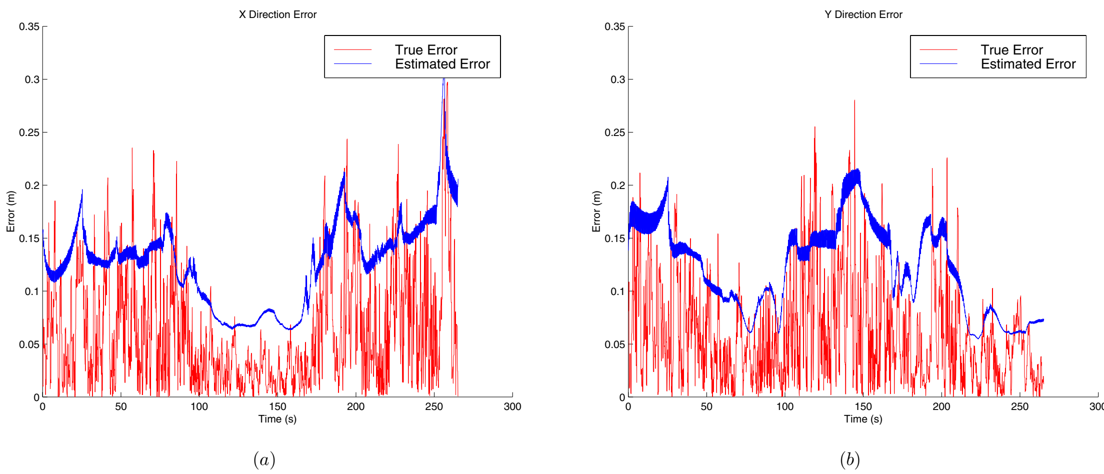


Figure 1: Actual and estimated vehicle position errors in (a) x-direction and (b) y-direction. Actual error (the error between true vehicle path and estimated vehicle path) is not normally available in a real system. The estimated error (standard deviation or square-root of covariance) in vehicle path is generated by the filter.

Nominal Filter Performance

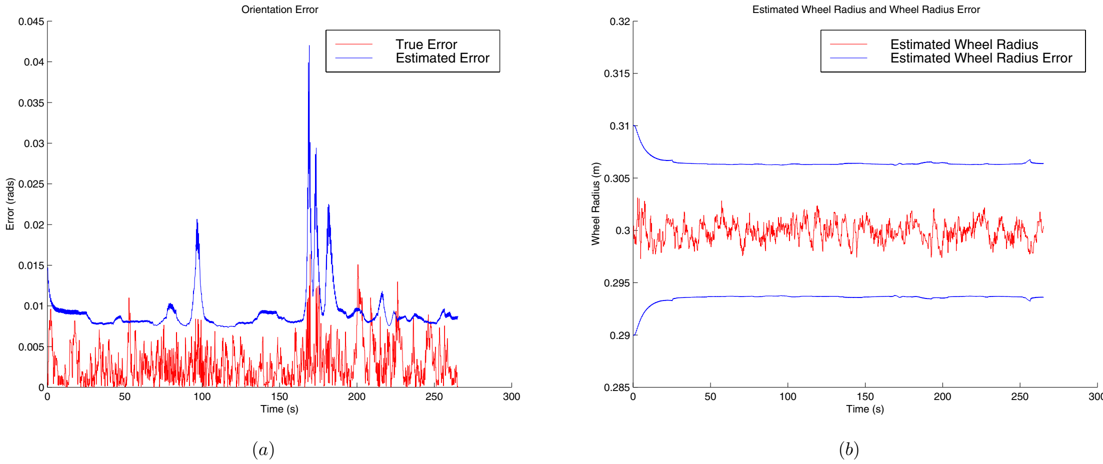


Figure 2: (a) Actual and estimated vehicle orientation error. (b) Estimated vehicle mean wheel radius with 1-σ estimated error bounds.

Nominal Filter Performance

- Difference between these plots and those of a linear system is that the estimated errors are not constant and do not reach a steady-state value.
- Position error plots show that the filter is well matched with at least 60% of actual position errors falling within the estimated (1-σ) error bound.
- The position errors vary substantially over the run. Caused by the vehicle changing orientation: along-track and cross-track errors, locally observable geometry of the beacons.
- Orientation error also shows a non-constant behaviour. The obvious feature in these plots are the sudden spikes in estimated orientation error correspond to the times at which the vehicle is steering sharply.
- Wheel radius error shows a much more constant behaviour. As errors in wheel radius feed through into vehicle position error, so observation of position while the vehicle is in motion allows estimation (observability) of the wheel radius.

Nominal Filter Performance:Innovations

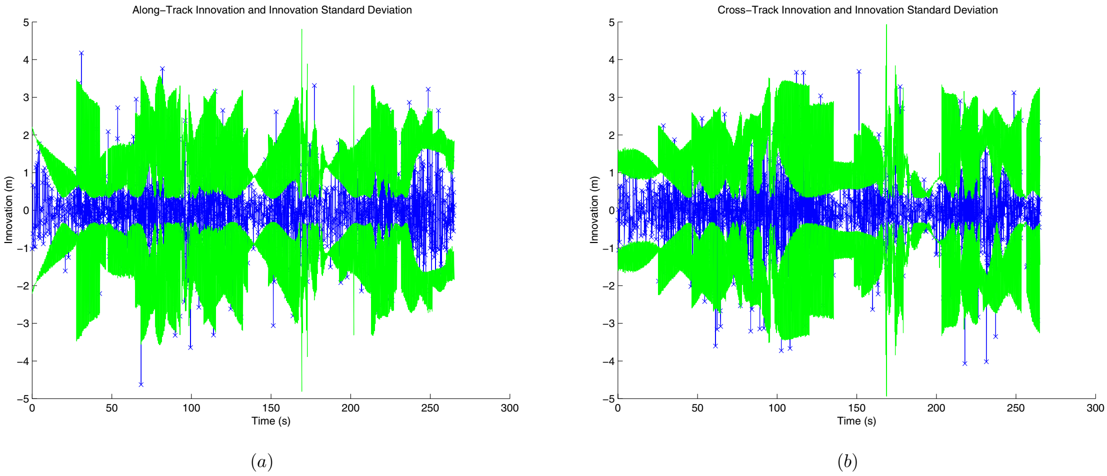


Figure 3: Filter innovations and innovation standard deviations in (a) Along-track direction and (b) Cross-Track direction.

Nominal Filter Performance:Innovations

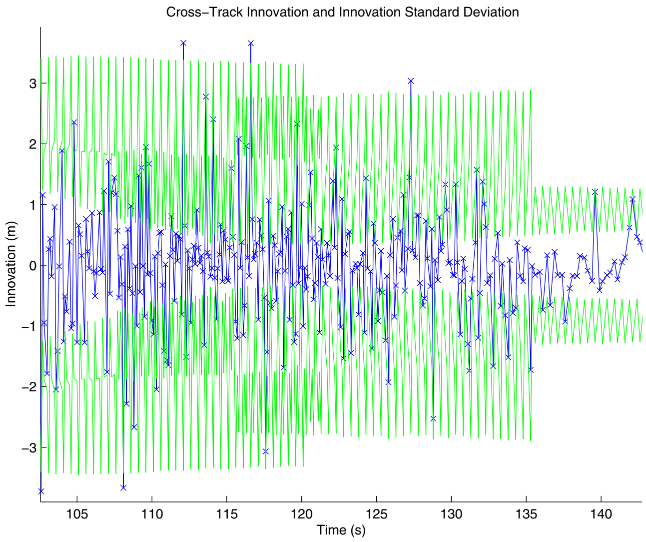
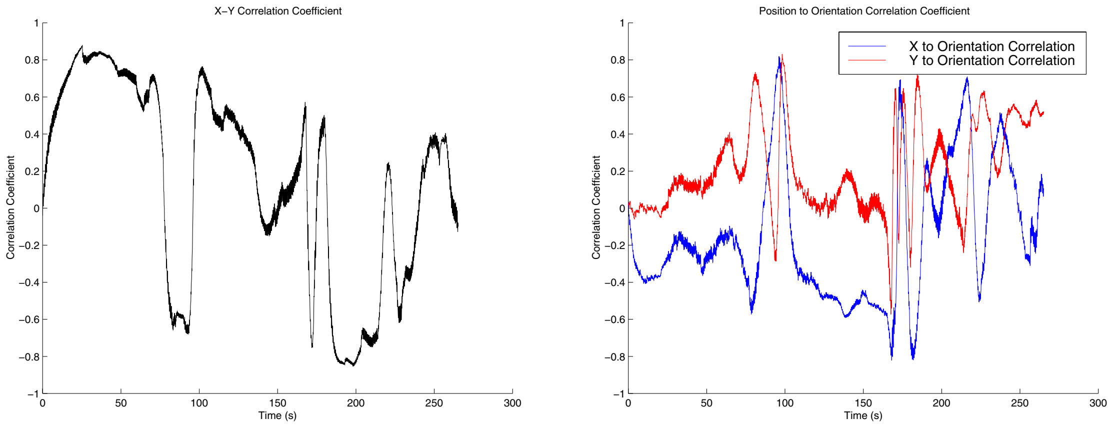


Figure 4: Detail of Cross-track innovations.

Nominal Filter Performance:Innovations

- To tune estimate vehicle errors, the along-track and cross-track innovations are the most helpful.
- Innovations show the filter is well matched with most innovations falling within the 1-*sigma* standard deviation.
- Innovations show substantial variation over time. The periodic (triangular) rise and fall of the innovation standard deviations is due to specific beacons coming into and then leaving view.
- Detail shows several different beacons being observed; a measure of the relative information contribution of each beacon.

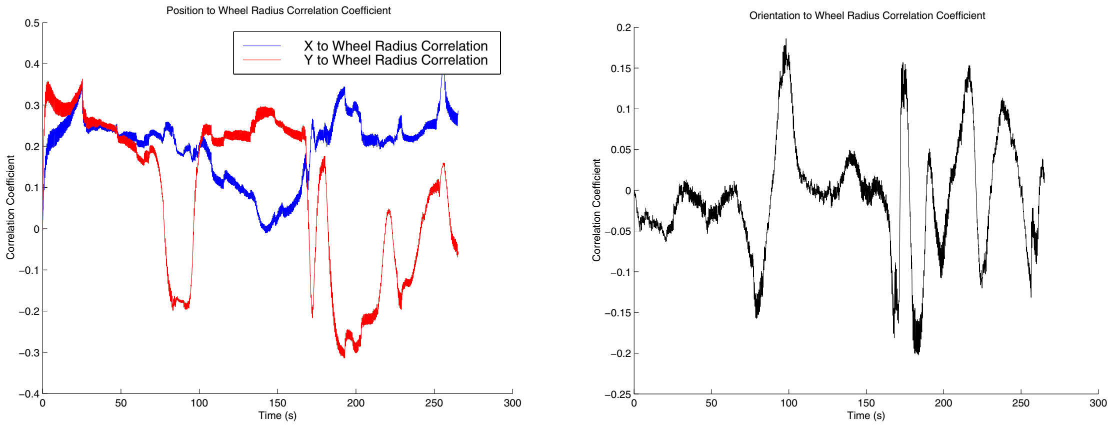
Nominal Filter Performance:Correlations



(*a*) (*b*)

Figure 5: Filter correlation coefficients (a) x to y, (b) x and y to orientation.

Nominal Filter Performance:Correlations



(c) (d)

Figure 6: Filter correlation coefficients (c) x and y to wheel radius, (d) orientation to wheel radius.

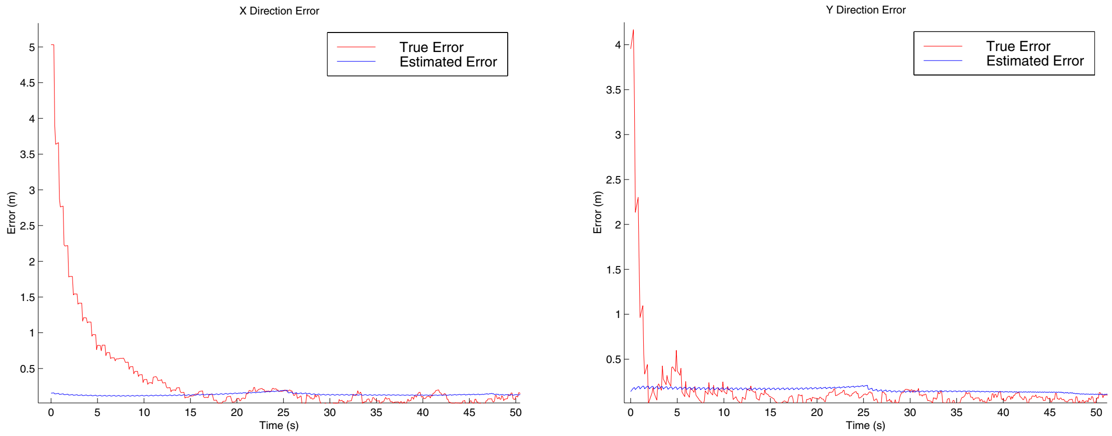
Nominal Filter Performance:Correlations

- Correlation coefficients

$$\rho_{ij} = \frac{P^+_{ij}(k)}{\sqrt{P^+_{ii}(k)P^+_{jj}(k)}}.$$

- When $\rho_{ij} = 0$, states are uncorrelated,
- when $|\rho_{ij}| \to 1$, the states are highly correlated.
- Correlation between states is desirable to aid estimation performances, but can cause quite complex and counter-intuitive behaviour in the filter.
- Position estimates correlated because primary errors are injected cross and along track.
- Vehicle location estimates are moderately and consistently correlated with vehicle orientation estimates; due to steering rates.
- Wheel radius estimate is only weakly correlated with position and orientation; essential in allowing the wheel radius to be observed and estimated.

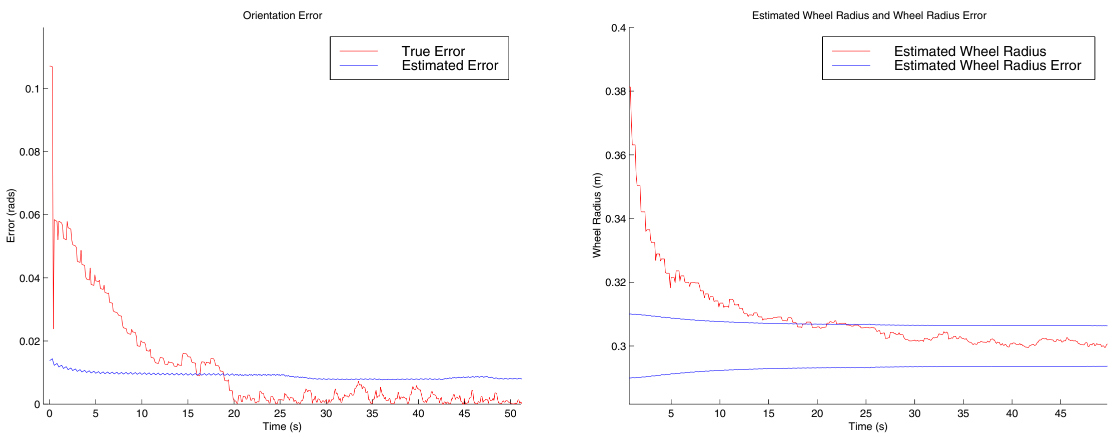
Errors and Faults



(a) (b)

Figure 7: Initial transients in filter state estimates following an initialisation error $[\delta x, \delta y, \delta\phi, \delta R] = [5.0, 5.0, 0.2, 0.1]$: estimates in (a) x, (b) y.

Errors and Faults



(c) (d)

Figure 8: Initial transients in filter state estimates following an initialisation error $[\delta x, \delta y, \delta\phi, \delta R] = [5.0, 5.0, 0.2, 0.1]$: estimates in (c) ϕ and (d) R.

Errors and Faults

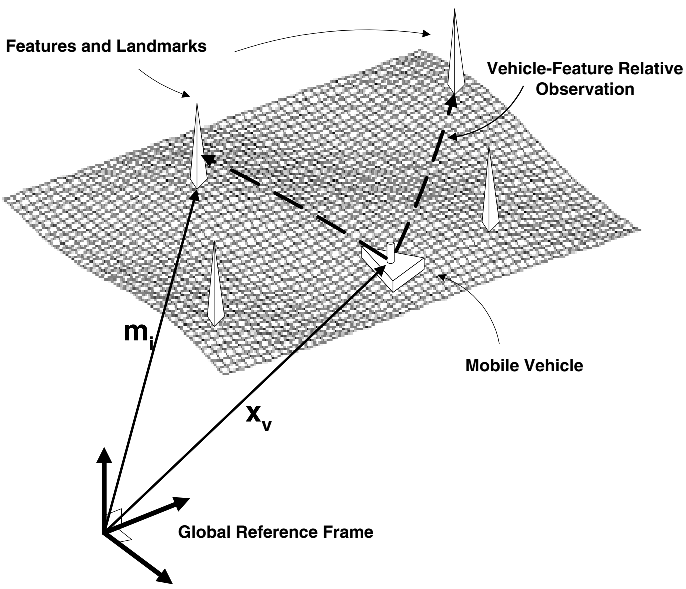
- The linear Kalman filter is stable for any initial conditions and for any perturbation.
- No such guarantees are possible for the extended Kalman filter.
- Initialisation error is often one of the most difficult practical problems to address in any real-world application of the EKF.
- There are no general solutions to this problem either.
- A practical solution is to use a batch process estimate to provide initial values for state variables.
- What initial error can be tolerated: depends on stability of process model.
- Generally good for navigation or kinematic models.

Tuning Filter Parameters

- Unlike the case for linear filters, there is no general methodology for estimating filter noise parameters for extended Kalman filters.
- Start with reasonable ranges of values for the different filter parameters.
- Deduced from observation of the system, simple experiments or simulation.
- In complex problems, it is good practice to establish an "error budget" and to systematically analyse contributions to overall estimation performance.
- Good practice (and generally produces best results) if the *true source* of error is identified and analysed.

Introduction to the SLAM Problem

SLAM Structure



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:

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

- 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:

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

- Where

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

- 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:

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

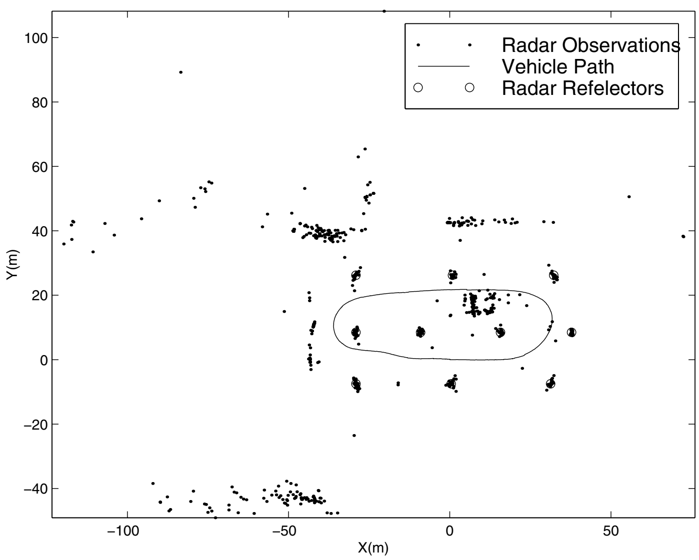
- True also for any sub-map

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

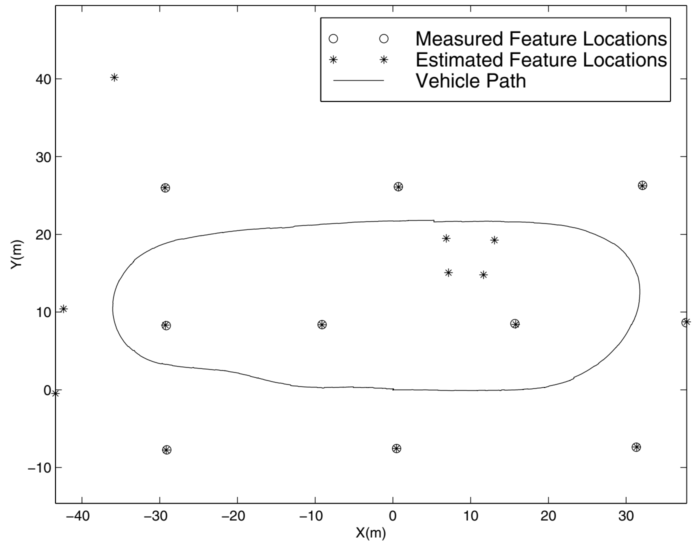
Result III

- 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.

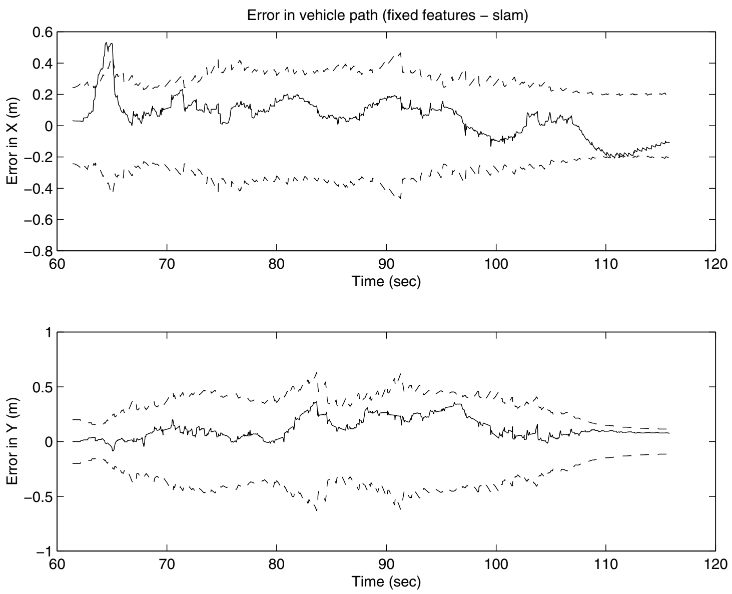
Characteristic Results: Raw Data



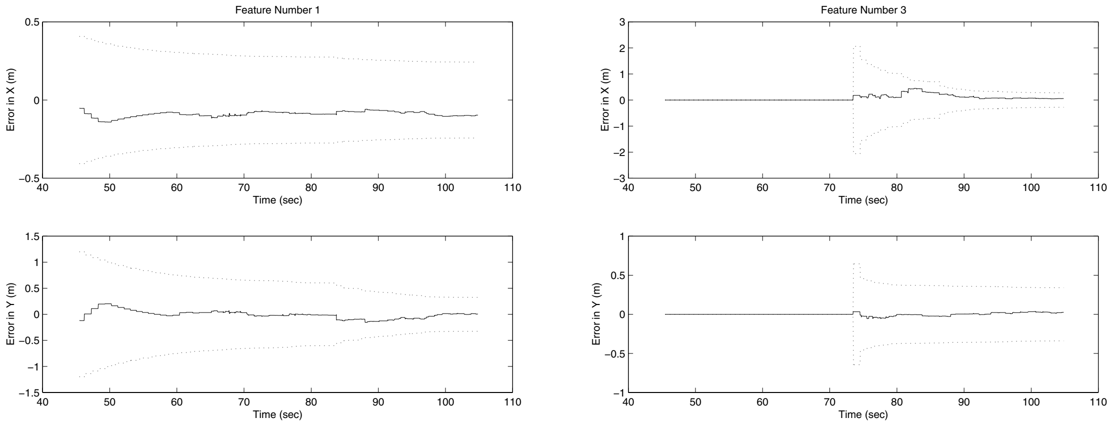
Characteristic Results: Vehicle Path and Landmark Locations



Characteristic Results: Vehicle Position Errors

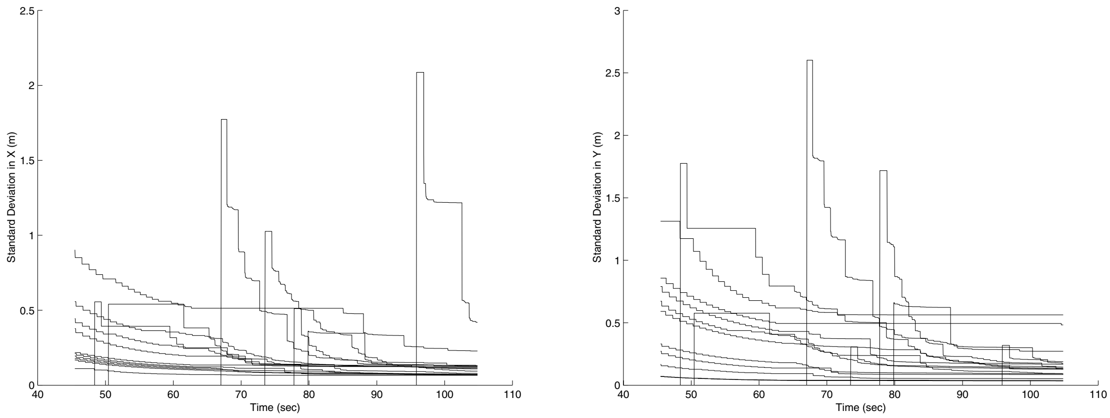


Characteristic Results: Example Land Mark Errors



(a) (b)

Characteristic Results: All Land Mark Errors



(a) (b)

One Last Interesting Result

- 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:

One Last Interesting Result

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

- 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 ...