Simultaneous Localization and Map Building Using Natural features in Outdoor Environments

Jose Guivant, Eduardo Nebot, Hugh Durrant Whyte
Australian Centre for Field Robotics
Department of Mechanical and Mechatronic Engineering
The University of Sydney, NSW 2006, Australia
{jguivant/nebot}@mech.eng.usyd.edu.au

Abstract. This work presents efficient algorithms for real time Simultaneous Localization and Map Building (SLAM). The accuracy of the algorithm is investigated with respect to standard localization algorithms using a 1cm kinematic GPS as ground truth. The issue of algorithm convergence when running in large areas of operation and for long period of time is investigated. Experimental results in unstructured environment are presented using more than 200 natural features. The complexity of feature extraction is also investigated with an approach to identify common types of outdoor natural features.

1 Introduction

Reliable localization is an essential component of any autonomous vehicle. The basic navigation loop is based on dead reckoning sensors that predict the vehicle high frequency manoeuvres and low frequency absolute sensors that bound the positioning errors [1]. During the past two years significant progress has been made towards the solution of the navigation and map building problem. One of the main problems of the SLAM algorithms has been the computational requirements, not so much in terms memory but in terms of CPU time requirements. It is well known that the complexity of the SLAM algorithms can be reduced to the order of N^2 [2], N being the number of landmarks in the map. For long duration missions the number of landmarks will increase and eventually the onboard computer resources will not be sufficient to update the map in real time. This problem arises because the full map is correlated. The correlation appears since the observation of a new landmark is being obtained with a sensor mounted on the mobile robot and obviously the error will be correlated with the uncertainty in the vehicle location and the other landmarks of the map. This correlation is of fundamental importance for the long-term convergence of the algorithm [3], and needs to be maintained for the full duration of the mission. Leonard et. al. [4], proposed to split the global map into a number of submaps, each with their own vehicle track. They present an approximation technique to address the update of the covariance in the transition between maps. Thrun [5] has also demonstrated navigation and map building in indoor environment using a probabilistic approach. Still the main barrier to overcome is the high computational cost required for real time implementation. This paper presents simplifications in the prediction and update stage of the SLAM algorithm. Sub-optimal algorithms based on maintaining conservative updates of the error covariances matrices are also presented. The convergence and accuracy of the algorithm are tested in large outdoor environments with more that 200 natural features

2 Navigation System

The navigation loop is based on encoders and range / bearing information provided by a laser sensor. A simple kinematic model is used for this experimentation [6]. The trajectory of the centre of the back axle is given by:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\phi}_c \end{bmatrix} = \begin{bmatrix} v_c \cdot \cos(\phi) \\ v_c \cdot \sin(\phi_c) \\ \frac{v_c}{L} \cdot \tan(\alpha) \end{bmatrix} \tag{1}$$

where υ_c is the velocity of the centre of the back axle and α is the orientation of the front wheels with respect to the forward direction. The equation that relates the observation with the states is

$$\begin{bmatrix} z_r^i \\ z_\beta^i \end{bmatrix} = h(X, x_i, y_i) = \begin{bmatrix} \sqrt{(x_L - x_i)^2 + (y_L - y_i)^2} \\ \text{atan}\left(\frac{(y_L - y_i)}{(x_L - x_i)}\right) - \phi + \pi/2 \end{bmatrix} \tag{2}$$

where z and $[x, y, \phi]$ are the observation and state values respectively, and (x_i, y_i) are the positions of the beacons or natural landmarks.

Simultaneous Localization and Map Building

The SLAM algorithms incorporate the location of the features or beacon as part of the state vector. The state vector is now given by:

$$X = \begin{bmatrix} x_v \\ x_L \end{bmatrix}, \; x_V = (x, y, \phi) \in R^3 \quad , \quad x_L = (x_1, y_1, .., x_n, y_n) \in R^N \tag{3}$$

Where x_v and x_L are the states of the vehicle and actual landmarks. The vehicle starts at an unknown position with a given uncertainty and obtains measurements of the environment relative to its position. This information is used to incrementally build and maintain a navigation map and localize with respect to this map. The landmarks can be natural features or special designed beacon located at unknown location. The dynamic model of the extended system that considers the new states can now be written:

$$\begin{aligned} x_V(k+1) &= f(x_V(k)) \\ x_L(k+1) &= x_L(k) \end{aligned} \tag{4}$$

It can be seen that the dynamic of the states x_L is invariant since the landmarks are assumed to be static. Then the Jacobian matrix for the extended system becomes

$$\frac{\partial F}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_V} & \varnothing \\ \varnothing^T & I \end{bmatrix} = \begin{bmatrix} J_1 & \varnothing \\ \varnothing^T & I \end{bmatrix}, \quad J_1 \in R^{3x3}, \quad \varnothing \in R^{3xN}, \quad I \in R^{NxN} \tag{5}$$

The observations obtained with a range and bearing device are relative to the vehicle position. The observation equation is a function of the state of the vehicle and the states representing the position of the landmark:

$$\begin{aligned} r_i = h_r\left(X\right) = \left\|\left(x,y\right)-\left(x_i,y_i\right)\right\|_2 = \sqrt{\left(x-x_i\right)^2+\left(y-y_i\right)^2} \\ \alpha_i = h_\alpha\left(X\right) = \text{atan}\left(\frac{\left(y-y_i\right)}{\left(x-x_i\right)}\right)-\phi+\frac{\pi}{2} \end{aligned} \qquad (6)$$

Where *(x,y)* is the position of the vehicle, *(x$_i$,y$_i$)* the position of the landmark numbered *i* and *Φ* the orientation of the car. Then the Jacobian matrix of the vector *(r$_i$,α$_i$)* respect to the variables *(x,y, Φ,x$_j$,y$_j$)* can be evaluated using:

$$\frac{\partial h}{\partial X} = \begin{bmatrix} \dfrac{\partial h_r}{\partial X} \\ \dfrac{\partial h_\alpha}{\partial X} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial r_i}{\partial\left(x,y,\phi,\{x_j,y_j\}\right)} \\ \dfrac{\partial \alpha_i}{\partial\left(x,y,\phi,\{x_j,y_j\}\right)} \end{bmatrix} \qquad (7)$$

with

$$\begin{aligned} \frac{\partial h_r}{\partial X} &= \frac{1}{\Delta}\cdot\left[\Delta x,\Delta y,0,0,0,...,-\Delta x,-\Delta y,0,....,0,0\right] \\ \frac{\partial h_\alpha}{\partial X} &= \left[-\frac{\Delta y}{\Delta^2},\frac{\Delta x}{\Delta^2},-1,0,0,...,\frac{\Delta y}{\Delta^2},-\frac{\Delta x}{\Delta^2},0,....,0,0\right] \\ \Delta x = \left(x-x_j\right),& \quad \Delta y = \left(y-y_j\right), \quad \Delta = \sqrt{\left(\Delta x\right)^2+\left(\Delta y\right)^2} \end{aligned} \qquad (8)$$

These equations can be used to build and maintain a navigation map of the environment and to track the position of the vehicle.

Algorithm optimisation

In this section a number of simplifications for the FULL SLAM implementation are presented. The SLAM is implemented using and Extended kalman Filter (EKF). The EKF structure can be divided in 2 steps: The Prediction step:

$$\begin{aligned} \hat{X}\left(k+1\right) &= F\left(\hat{X}\left(k\right)\right) \\ P\left(k+1,k\right) &= J\cdot P\left(k,k\right)\cdot J^T + Q\left(k\right) \end{aligned} \qquad (9)$$

and the update stage

$$\begin{aligned} S\left(k+1\right) &= H\cdot P\left(k+1,k\right)\cdot H^T + R\left(k\right) \\ W\left(k+1\right) &= P\left(k+1,k\right)\cdot H^T\cdot S\left(k+1\right)^{-1} \\ Innov\left(k+1\right) &= z\left(k+1\right)-h\left(\hat{X}\left(k+1\right)\right) \\ \hat{X}\left(k+1,k+1\right) &= \hat{X}\left(k+1,k\right)+W\left(k+1\right)\cdot Innov\left(k+1\right) \\ P\left(k+1,k+1\right) &= P\left(k+1,k\right)-W\left(k+1\right)\cdot S\left(k+1\right)\cdot W\left(k+1\right)^T \end{aligned} \qquad (10)$$

It can be proved that considering the zeros in the matrices given in equations (5) and (8) the computational requirements becomes of the order of N^2 instead of N^3 for the case where observations are taken sequentially with the predictions. Normally more than one prediction step is executed between 2 update steps. This is due to the fact that the prediction stage is driven by the high frequency sensors information that acts as inputs to the dynamic model of the car. The low frequency sensors report the observation used in the estimation stage of the EKF and is processed at much lower frequency. For example, the steering angle and wheel speed are sampled every 20 milliseconds but the laser frames are obtained with a sample time of 200 milliseconds. In this case a relation of approximately 10 prediction steps for one update step is present. The general form of the error matrix that is propagated forward in time is given equation (9). In SLAM applications the Jacobian *J* will have the form given in equation (5). Then the first term in the propagation equation can be simplified:

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

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

These results can be generalized for n prediction steps:

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

where

$$G_1 = G_1\left(k,n\right) = \prod_{i=1}^{n} J_1\left(k+i\right) \qquad (13)$$

In Equation (12) P_{11} needs to be evaluated for every prediction step since the quality of the estimated position is required all the time. P_{22} remains constant between updates. Finally the cross-covariances, P_{12} (and P_{21}), have to be evaluated only before the estimation procedure. The evaluation of P_{12} (or P_{21}) becomes the major computational requirement in the computation of P in the prediction step. From the previous development, for the case of n prediction steps without an update, the computational requirements becomes approximately equal to $n*3*9 + 9*M=27n+9M$. This is significant smaller than the standard form, which is greater than $n*M*9$. In this case M is the number of landmarks plus the number of vehicle states.

Suboptimal Solution.

Most of the computational requirements of the EKF is needed during the update process of the error covariance matrix. This section presents a sub-optimal approach to reduce the number of calculations considering a subset of the navigation landmarks. It is also be proved that this approach does not generate overconfident results. Once an observation is being validated and associated to a given landmark, the covariance error matrix of the states is updated;

$$\begin{gathered} P = P - \Delta P \\ \Delta P = W \cdot S \cdot W^T = \rho \cdot \rho^T \end{gathered} \qquad (14)$$

In SLAM applications the state vector is formed with the states of the vehicle and the states representing the position of each landmark in the actual map. This state vector can be divided in 2 groups

$$X = \begin{bmatrix} X_A \\ X_B \end{bmatrix}, \quad X_A \in R^{N_A}, \quad X_B \in R^{N_B}, \quad X \in R^N, \quad N = N_A + N_B \qquad (15)$$

With this partition it will be possible to generate conservative estimates not updating the covariance and cross-covariance corresponding to the subvector B. The covariance matrix can then be written in the following form:

$$P = \begin{bmatrix} P_{AA} & P_{AB} \\ P_{AB}^T & P_{BB} \end{bmatrix}, \quad \Delta P = \begin{bmatrix} \Delta P_{AA} & \Delta P_{AB} \\ \Delta P_{AB}^T & \Delta P_{BB} \end{bmatrix} = W \cdot S \cdot W^T = \rho \cdot \rho^T \qquad (16)$$

As mention before a more conservative update can be obtained if the nominal update matrix ΔP is replaced by the sub-optimal $\Delta P*$

$$\Delta P^* = \begin{bmatrix} \Delta P_{AA} & \Delta P_{AB} \\ \Delta P_{BA} & \varnothing \end{bmatrix} = \Delta P - \begin{bmatrix} \varnothing & \varnothing \\ \varnothing & \Delta P_{BB} \end{bmatrix}, \quad P^* = P - \Delta P^* = P - \Delta P + \begin{bmatrix} \varnothing & \varnothing \\ \varnothing & \Delta P_{BB} \end{bmatrix} \tag{17}$$

In the appendix it is demonstrated that the simplification proposed does not generate overconfident covariance estimates. In this case the submatrices evaluated are P_{AA} , P_{AB} and P_{BA}. P_{BB} is not evaluated. The fundamental problem is the selection to the partition A and B of the state vector.

By inspecting the diagonal elements of ΔP it can be seen that many of its terms are very small compared to the corresponding previous covariance value in the matrix P. This indicates that the new observation does not have a significant information contribution to this particular state. This is an indication to select a particular state as belonging to the subset B. The diagonal matrix of ΔP can be evaluated on-line with low computational cost. The other criterion used is based on the value of the actual covariance of the state. If it is below certain threshold it can be a candidate for the subvector B. In many practical situations a large number of landmarks can usually be associated to the subvector B. This will introduce significant computational savings since P_{BB} can potentially become larger than P_{AA}. The cross-correlation P_{AB} and P_{BA} are still maintained but are of much lower order. Finally the selection criteria to obtain the partition of the state vector can be given with the union of the following I_i sets:

$$I_1 = \left\{ i \quad \backslash \quad \Delta P(i,i) < c_1 \cdot P(i,i) \right\}, \quad I_2 = \left\{ i \quad \backslash \quad P(i,i) < c_2 \right\}, \quad I = I_1 \cup I_2 \tag{18}$$

Then ΔP^* is evaluated as follows:

$$\begin{aligned} \Delta P^*(i,j) &= 0 & \forall i,j \backslash i \in I \quad and \quad j \in I \\ \Delta P^*(i,j) &= \Delta P(i,j) & \forall i,j \backslash i \notin I \quad or \quad j \notin I \end{aligned} \tag{19}$$

Finally the error covariance matrix is updated with the simplified ΔP

$$P^*(k+1,k+1) = P(k+1,k) - \Delta P^* \tag{20}$$

The meaning of the set I_1, is that with the appropriate selection of c_1 we can reject negligible update of covariances. As mentioned before the selection of I_1 requires the evaluation of the diagonal elements of the matrix ΔP . The evaluation of the $\Delta P(i,i)$ elements requires a number of operations proportional to the number of states instead of the quadratic relation required for the evaluation of the complete ΔP matrix.

The second subset defined by I_2 is related to the states whose covariances are small enough to be considered practically zero. In the case of natural landmarks they become almost equivalent to beacons at known positions. The number of elements in the set I_2 will increase with time and can eventually make the SLAM algorithms of computational requirements comparable to the standard beacon localisation algorithms .

Finally, the order of the saving in the calculations depends of the size of the set I . With appropriate exploration polices (real time mission planning) this set can be maintained within the bounds of the computational capabilities of the on-board resources.

3 Experimental Results

The navigation system was tested in two different environments with a utility vehicle retrofitted with the sensors described. The first testing site corresponds to the top level of the car park building of the university campus. This testing site was chosen to compare the SLAM results with the ground truth provided by a Kinematic Glonass/GPS system of 1 cm accuracy. Three different implementations are presented. The first set of results corresponds to the localization algorithms using reflective beacons at known locations. Figure 1 presents the 95 % confidence bounds of the estimated position of the vehicle, continuous line, with the true error, dotted line. It can be seen that most of the errors are bounded by the confidence bounds estimated by the filter. It is also important to note that the localizer is able to estimate the position of the vehicle with and error of approximate 6 centimetres.

The second experimental results correspond to SLAM using only beacons. The system built a map of the environment and localized itself. The accuracy of this map is determined by the initial vehicle position uncertainty and the quality of the combination of dead reckoning and external sensors. In this experimental run an initial uncertain of 10 cm in coordinates x and y was assumed. Figure 2 presents the absolute error and the predicted standard deviation (2 σ bounds, 95 % confidence bounds). These plots show that the bounds are consistent with the actual error. It is also important to remark that the uncertainty in position does not become smaller than the initial vehicle uncertainty. This is expected since the laser information is obtained relative to the vehicle position. The only way the uncertainty can be reduced is by incorporating additional information that is not correlated to the vehicle position, such as GPS position information or recognizing a beacon with known position.

The final experimental results correspond to SLAM using all the features available in the environment. Figure 3 shows the initial part of the experimental run while the system is still incorporating new landmarks. Figure 4 presents the absolute error with the estimated standard deviation. These tests proved that under ideal conditions the SLAM algorithms can provide similar accuracy when compared to the standard localization algorithms. The second set of experimental runs is performed in a totally unstructured environment. Figure 5 shows the experimental car in the testing environment. In this case the car is running on different type of surfaces. Modelling errors are expected due to slip and the 3-D nature of the environment. Another objective of the test was to determine the convergence characteristics of the algorithm when running in large areas for long period of time. In this case the vehicle was running for more than 20 minutes in the area shown in figure 6 . This figure presents part of the estimated trajectory and the landmark used up to this point.

The standard deviation error of the vehicle and a few selected landmarks positions during the total duration of the run are shown in Figure 7 . It can be seen that the uncertainty of the landmarks decreased constantly with time but not below the initial vehicle uncertainty. Figure 8 presents the history of the covariance evolution of all the landmarks and their final value respectively. The x axis in the last figure indicates the landmark number. It can be seen that at the end of the run the uncertainty of most of the landmarks is below 20 cm, value that is reasonable considering the uncertainty in the determination of the position of the landmarks in this particular application. Finally the simplifications were implemented not updating the states with errors smaller than 8cm. Figure 9 presents the total number of landmarks and the ones that are not updated. It shows that the reduction in the number of states is approximately 20%.

The landmarks detection process is essential for SLAM. In this case the laser obtained profiles of trees, as shown in Figure 10 , and determined the most likely centre of the trunk. A Kalman filter was also implemented to reduce the errors due to the different profiles obtained when observing the trunk of the trees from different locations.

4 Conclusion

This work presented efficient algorithms for SLAM. The accuracy of the algorithm was investigated in relation to standard localization algorithms using 1cm kinematic GPS as ground truth. The convergence of the algorithm with the proposed simplification is presented with experimental runs in an unstructured environment running for long periods of time. Experimental results are also presented that demonstrated the computational savings obtained without noticeable degradation of the algorithm. The complexity of feature extraction was also investigated. Future work will address the integrity issues of the algorithm when working in larger areas of operations.

Appendix

This appendix demonstrates that the simplification proposed generate consistent error covariance estimates. The covariance error matrix $P^{*}(k+1)$ can be rewritten as follows

$$P^{*}(k+1) = P(k) - \Delta P^{*} = P(k) - \Delta P + \mu \qquad (21)$$

where

$$\Delta P^{*} = \begin{bmatrix} \Delta P_{AA} & \Delta P_{AB} \\ \Delta P_{BA} & \varnothing \end{bmatrix} = \Delta P - \mu, \quad with \quad \Delta P = \begin{bmatrix} \Delta P_{AA} & \Delta P_{AB} \\ \Delta P_{BA} & \Delta P_{BB} \end{bmatrix} \geq 0 \qquad \mu = \begin{bmatrix} \varnothing & \varnothing \\ \varnothing & \Delta P_{BB} \end{bmatrix} \geq 0 \qquad (22)$$

The matrices ΔP and μ are positive semi-definite since:

$$\Delta P = \begin{bmatrix} \Delta P_{AA} & \Delta P_{AB} \\ \Delta P_{AB}^T & \Delta P_{BB} \end{bmatrix} = W \cdot S \cdot W^T \geq \quad 0 \qquad (23)$$

$$and \qquad \Delta P_{BB} = W_B \cdot S_B \cdot W_B{}^T \geq 0$$

As expressed in equation (21), the total update is formed by the optimal update plus an additional positive semi-definite noise matrix $\boldsymbol{\mu}$. The matrix $\boldsymbol{\mu}$ will increase the covariance uncertainty:

$$P^*(k+1) = P(k+1) + \mu \qquad (24)$$

then the sub optimal update of P^* becomes more conservative than the full update.

References

[1] Nebot E., Durrant-Whyte H., “High Integrity Navigation Architecture for Outdoor Autonomous Vehicles”, Journal of Robotics and Autonomous Systems, Vol. 26, February 1999, p 81-97.

[2] Motalier P., Chatila R., “Stochastic multisensory data fusion for mobile robot location and environmental modelling”, In Fifth Symposium on Robotics Research, Tokyo, 1989.

[3] Castellanos J., Tardos J, Schmidt G., “Building a global map of the environment of a robot: The importance of correlations”, IEEE conference on Robotics and Automation, 1997, pp. 1053-1059.

[4] Leonard J. J. and Feder H. J. S., “A computationally efficient method for large-scale concurrent mapping and localization”, Ninth International Symposium on Robotics Research, pp 316-321, Utah, USA, October 1999

[5] Thrun S., Fox D., Bugard W., “Probabilistic Mapping of and Environment by a Mobile Robot”, Proc. Of 1998 IEEE, Belgium , pp 1546-1551

[6] Guivant J., Nebot E., Baiker J, “High Accuracy Navigation Using Laser Range Sensors in outdoor Applications”, IEEE International Conference on Robotics and Automation, Jan 2000

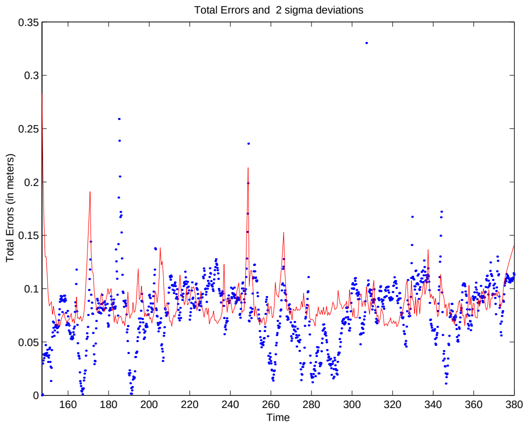

Figure 1 Standard deviation with beacons

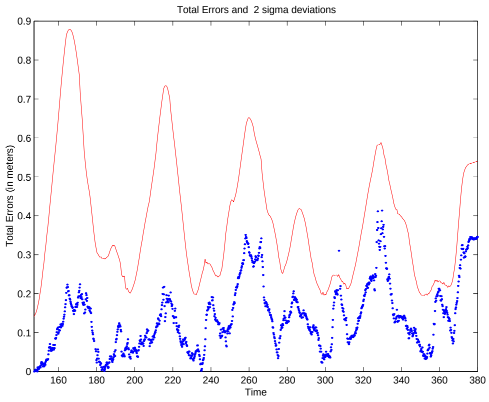

Figure 2 Standard deviation with SLAM.

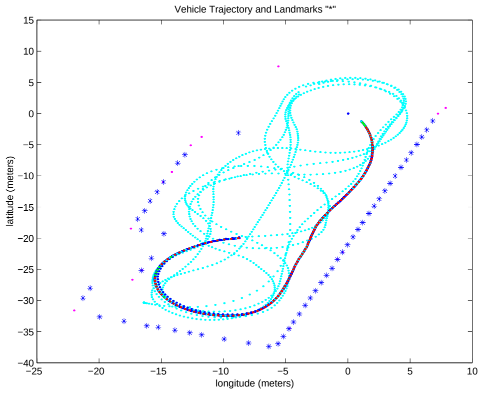

Figure 3. Initial part of the trajectory using SLAM

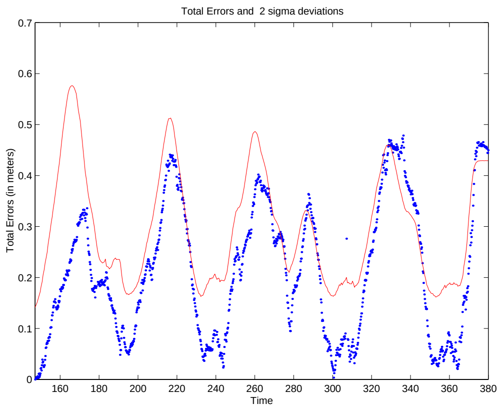

Figure 4 Absolute error and standard deviation

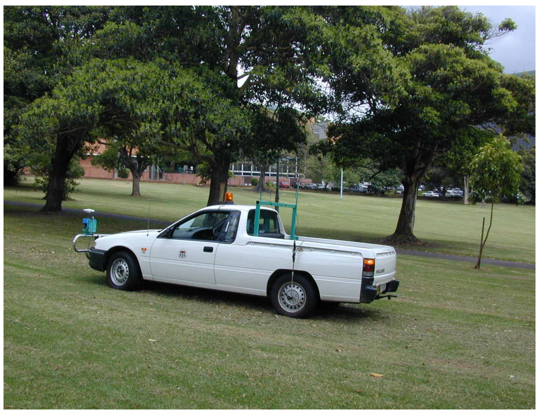

Figure 5. Utility car used for the experiments.

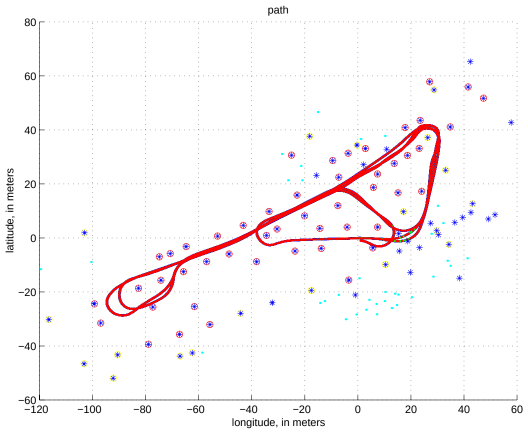

Figure 6 Vehicle trajectory and landmarks

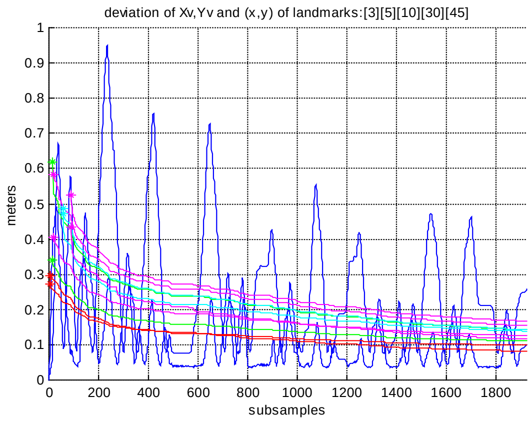

Figure 7 Landmark's Standard Deviation

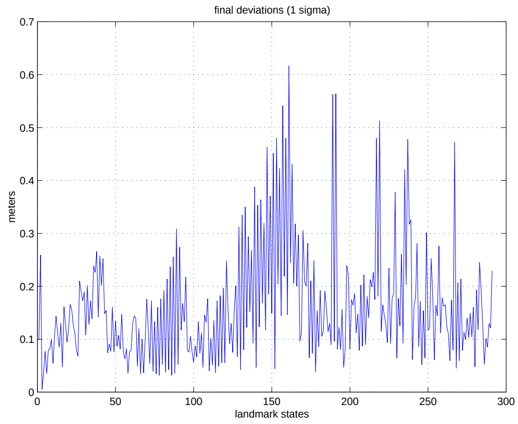

Figure 8 Final value of landmark's covariance

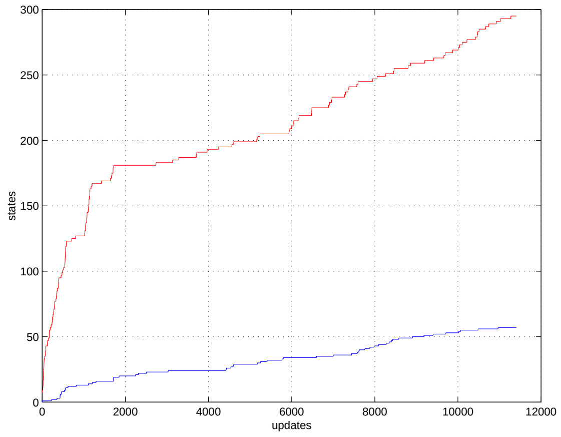

Figure 9 States not updated

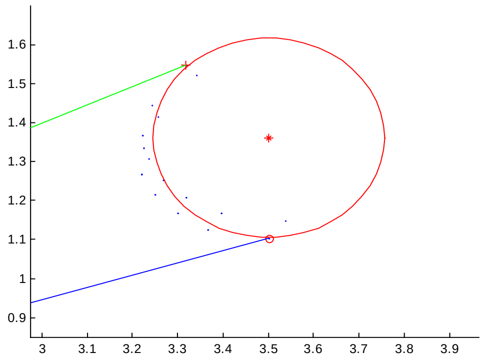

Figure 10 Tree profile and approximation