Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

Field Navigation of Mobile Robots

A. B. Lintott BE, PhD
Industrial Research Ltd
5 Sheffield Cres
Christchurch
New Zealand
A.Lintott@irl.cri.nz

Lan Le Ngoc BE, PhD
Industrial Research Ltd
5 Sheffield Cres
Christchurch
New Zealand
L.Lengoc@irl.cri.nz

Abstract

Robust navigation techniques for mobile robots are still the subject of much research and development. A number of approaches are being developed worldwide, each with their own merits. Research into the problem at Industrial Research Ltd has led to the development of a new technique based on probabilistic maps of the robot's local environment.

Notation

The following notational conventions are used in this report:

- The i'th element of vector $\boldsymbol{s}$ is denoted s_i. Similarly, A_{ij} is the element of matrix $\boldsymbol{A}$ at row i and column j.
- A vector quantity $\boldsymbol{s}$ may be expressed in any coordinate frame as a column matrix. A coordinate frame g is denoted by the symbol $\boldsymbol{\mathcal{F}}_g$. Where relevant, the coordinate frame that a quantity is expressed in is specified with a preceding superscript, e.g. ${}^g\boldsymbol{s}$ is vector $\boldsymbol{s}$ expressed in $\boldsymbol{\mathcal{F}}_g$ as a column matrix. Column matrix ${}^r\boldsymbol{s}$ represents the same physical vector quantity as ${}^g\boldsymbol{s}$, but the elements of each are different.
- If a vector $\boldsymbol{s}$ represents a stochastic quantity, then $\boldsymbol{\Sigma_s}$ is a positive semidefinite matrix representing the variance and covariance of $\boldsymbol{s}$, such that $\boldsymbol{\Sigma}_{ii}$ is the variance of s_i, and $\boldsymbol{\Sigma}_{ij}$ is the covariance of s_i and s_j. For brevity, the matrix $\boldsymbol{\Sigma_s}$ is referred to as the variance matrix, although it is more commonly known as the variance-covariance matrix.

1 Introduction

The objective of this work is to produce a robot navigation system suitable for use in forestry and horticultural applications. Investigation into probabilistic localisation techniques, such as the well-known Markov localisation technique, and the SLAM technique described below led to a new method which has been shown to be successful in computer simulations. This is described in §2. The method was extended to allow simple map building.

Mobile robot navigation may be broken down into 3 specific areas. Localisation, which is characterised by the question "Where am I?"; Purpose, or "Where am I going?"; and Path Planning, or "How do I get there?". This work focuses on the first of these areas. Given a robot with an array of range-finding sonar sensors, the aim was to construct a map of the robot's environment and determine the robot's position from that. The work here is based on environments containing point objects only.

1.1 Simultaneous Localisation and Map Building (SLAM)

Dissanayake *et al.* [2001] gave a method of performing SLAM which used a Kalman filter to moderate sensor noise and maintain a state vector which included map information as well as the robot's position. Several results were given which prove the numerical stability of the approach. The technique was implemented (ibid.) using a millimetre wave radar, which was capable of high range accuracy and high angular resolution, mounted on a motor vehicle. Elements of this technique were adopted in this work for sonar based navigation and the resulting system was shown to work well.

Using sonar to perform SLAM presents additional difficulties because of the poor angular resolution of the sonar sensors. The Pioneer robot has an array of 7 sonar sensors arranged about its console unit including 2 side facing sensors. The range readings are accurate to within approximately 20mm, but the poor angular resolution results in a large angular uncertainty, which is propagated to the Kalman filter. Despite this, the SLAM technique was found to perform very well for several trial tasks. The algorithm was able to maintain registration and the map would become more accurate as the trial progressed.

The trial tasks performed were:

1. Tracing a square path on carpet using 4 objects as landmarks. The robot was able to perform the task repeatedly without loss of registration. In some tests, sheets of paper were used to cause wheel slippage, while in others the robot was held back, rotated, and pushed sideways.
2. Following a path through a miniature orchard. The robot was able to complete multiple circuits of the orchard without losing registration. The same trial was carried out at high speed on flatter terrain with shorter grass and shown to work reliably as well. Both demonstrations are recorded on video [Lintott, 2002].
3. Interacting with the objects. The robot was programmed to complete a square circuit, then approach an object, shift it, complete another square circuit and then shift the object back. The SLAM localiser would ensure that the robot had a current estimate of the object positions for gripping based on 4 stationary objects that served as reference points. This demonstration was used repeatedly at a trade show. It was occasionally unsuccessful, mostly due to problems with the gripper, but also occasionally due to failure to localise the object sufficiently.

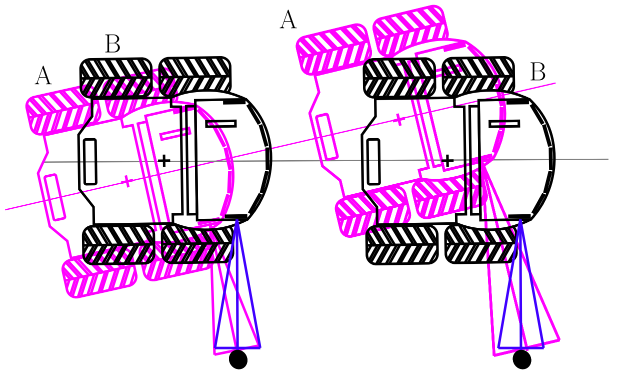
1.2 Rationale for probabilistic modelling

The results from the SLAM approach were very good, but the approach had certain deficiencies. Consider a blindfolded human navigating their way through an orchard of trees with a cane for guidance. Having touched one tree with a cane and then walked several paces before touching another, the human is able to glean a large amount of information. For instance, it should be possible to tell whether one is walking parallel to the row of trees or at some angle. It should also be possible to tell roughly how far the trees are apart, even though they weren't sensed at the same moment.

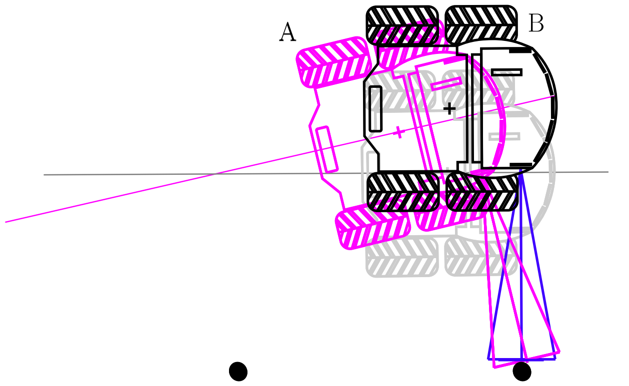
The SLAM approach is not capable of such rich inferences. Take for example the situation shown in Figure 1. The robot is misaligned with the row, but believes itself to be travelling parallel to it. It senses the first object, which agrees with its expectation of the object's position, but when it senses the second object it simply concludes that it has drifted sideways and doesn't correct its orientation. The only way that SLAM can correct its orientation is to detect both objects simultaneously, which is rarely possible for widely spaced objects. To put it another way, the robot needs to make a correction in $\Re^3$ from sensor information in $\Re^1$.

An improved approach would allow the effect of a previous observation to persist for a period so that it can be integrated into a localisation model. The observation should persist only as long as it is reliable. The accumulation of odometric error as the robot moves away from the object would make it less and less reliable the further the robot has moved. The importance or potency of an observation should be related to its certainty so that a very uncertain observation would not affect the localisation as much as a very recent and certain observation.

Such ideas are expressed in the probabilistic localisation model presented in the next section.



(a) The robot passes the first object and senses that it is on the right track. It continues to find it is further from the second object than anticipated (robot A). Robot B indicates the expected path.



(b) The robot corrects for its apparent positional error but since it has no record of having encountered the previous object it can only conclude that it has drifted sideways.

Figure 1: A problem with the SLAM approach to robot localisation.

2 Localisation by the maximum likelihood criterion

The following formulation is based on the assumptions listed below:

1. The position of each artifact $\boldsymbol{a}_i \in \Re^2$ in the map is known with an associated variance $\boldsymbol{\Sigma}_{\boldsymbol{a}_i}$.
2. The position of each observation $\boldsymbol{o}_i \in \Re^2$ is measured with an associated variance $\boldsymbol{\Sigma}_{\boldsymbol{o}_i}$. Calculation of $\boldsymbol{\Sigma}_{\boldsymbol{o}_i}$ is covered in detail in §2.1.

3. All observations correspond to artifacts (this assumption will be relaxed later, cf. §2.3).
4. Observations are statistically independent.
5. The robot's position is described by 2 cartesian parameters $\boldsymbol{x}_c = \begin{bmatrix} {}^g x_r & {}^g y_r \end{bmatrix}^T$ and an angular parameter θ_r. Collectively the position parameters are $\boldsymbol{x} = \begin{bmatrix} {}^g x_r & {}^g y_r & \theta_r \end{bmatrix}^T$.

We wish to find $\boldsymbol{x} \in \Re^3$ that minimizes

$$L_{\boldsymbol{x}} = -\log\big(\text{prob}\,(\boldsymbol{x}|\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n, \boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m)\big) \tag{1}$$

The difference between an observation and its associated artifact is $(\boldsymbol{o}_i - \boldsymbol{a}_j)$, and it is χ^2 distributed.

$$\chi^2_{ij} = (\boldsymbol{o}_i - \boldsymbol{a}_j)^T \left(\boldsymbol{\Sigma}_{\boldsymbol{o}_i} + \boldsymbol{\Sigma}_{\boldsymbol{a}_j}\right)^{-1} (\boldsymbol{o}_i - \boldsymbol{a}_j) \tag{2}$$

The probability distribution function of χ^2 is

$$P\big(\chi^2_{ij}\big) = \frac{1}{\Gamma\left(\frac{\nu}{2}\right)} \int_{\chi^2_{ij}/2}^{\infty} e^{-t} t^{\nu/2-1} dt \tag{3}$$

where ν is the degree of freedom of the parameters (2 in this situation). Assuming observations are statistically independent,

$$L_{\boldsymbol{x}} = -\sum_{i=1}^{n} P\big(\chi^2_{ij}\big) + C \tag{4}$$

where j corresponds to the single artifact $\boldsymbol{a}_j$ that matches observation $\boldsymbol{o}_i$, and C is some unknown constant.

A non-linear minimisation algorithm was used to solve equation 4 for an $\boldsymbol{x}$ that minimizes $L_{\boldsymbol{x}}$, or at least finds a sub-optimal approximation to the minimal solution. This is discussed in section §2.2.

2.1 Geometric model

The geometric model describes the way that raw sensor information is transformed into observation data, and how observations are updated with respect to new odometry data. The physical robot is represented in Figure 2. The sonar sensors return a range value r_s for each sonar s which indicates the shortest distance to any object within the sonar beam. The geometry and sensitivity of the sonar beam depends on a large number of environmental and atmospheric factors, but is modelled here as a simple cone with apex at the centre of the sonar sensor, and an apical angle of 20°. Let

$$\boldsymbol{T}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5}$$

The position of the object that caused the sonar to return range r_s is estimated by

$$\boldsymbol{o}_i = \boldsymbol{v}_s + \hat{\boldsymbol{v}}_b r_s + \boldsymbol{x}_c \tag{6}$$

where all vectors are expressed in $\boldsymbol{\mathcal{F}}_g$. In $\boldsymbol{\mathcal{F}}_r$

$${}^r\boldsymbol{o}_i = {}^r\boldsymbol{v}_s + {}^r\boldsymbol{v}_b \tag{7}$$

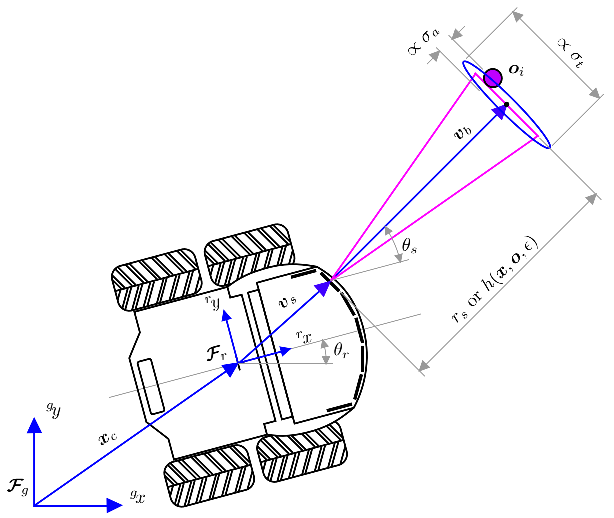


Figure 2: The Pioneer AT mobile robot detects an object $\boldsymbol{o}_i$ at range r_s

and

$$\boldsymbol{o}_i = \boldsymbol{T}(\theta_r)\,{}^r\boldsymbol{o}_i + \boldsymbol{x}_c \tag{8}$$

where ${}^r\boldsymbol{o}_i$ is the position of observation i expressed in frame $\boldsymbol{\mathcal{F}}_r$, or "relative to the robot". In reality we have no reason to believe that the object is located at the centre of the sonar beam. If the variance of the sonar reading is σ_a^2 along the beam and σ_t^2 across the beam, then

$$\boldsymbol{\Sigma}_s = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_t^2 \end{bmatrix} \tag{9}$$

and

$${}^r\boldsymbol{\Sigma}_{\boldsymbol{o}_i} = r_s \boldsymbol{T}(\theta_s)\,\boldsymbol{\Sigma}_s \boldsymbol{T}(\theta_s)^T \tag{10}$$

where θ_s is the orientation of the sonar sensor relative to frame $\boldsymbol{\mathcal{F}}_r$. The variance matrix defined by equation 10 is the variance of observation $\boldsymbol{o}_i$ expressed in $\boldsymbol{\mathcal{F}}_r$. In $\boldsymbol{\mathcal{F}}_g$ the variance has an additional component derived from the variance of $\boldsymbol{x}$. From equation 8,

$$\frac{\partial \boldsymbol{o}_i}{\partial \boldsymbol{x}} = \begin{bmatrix} \boldsymbol{I}_2 & \frac{\partial}{\partial\theta}\boldsymbol{T}(\theta)\,{}^r\boldsymbol{o}_i \end{bmatrix} \tag{11}$$

and to first order

$$\boldsymbol{\Sigma}_{\boldsymbol{o}_i} = \frac{\partial\,{}^r\boldsymbol{o}_i}{\partial \boldsymbol{x}}\,{}^r\boldsymbol{\Sigma}_{\boldsymbol{o}_i} \frac{\partial\,{}^r\boldsymbol{o}_i}{\partial \boldsymbol{x}}^T + \boldsymbol{\Sigma}_{\boldsymbol{x}_c} \tag{12}$$

Let $\boldsymbol{u} = \begin{bmatrix} u & \phi \end{bmatrix}^T$. If the robot rotates a small amount ϕ and then translates a small distance u

$$\boldsymbol{x}' = \boldsymbol{x} + \begin{bmatrix} u\cos(\theta_r + \phi) \\ u\sin(\theta_r + \phi) \\ \phi \end{bmatrix} \tag{13}$$

$$\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}} = \begin{bmatrix} \cos(\theta_r + \phi) & -u\sin(\theta_r + \phi) \\ \sin(\theta_r + \phi) & u\cos(\theta_r + \phi) \\ 0 & 1 \end{bmatrix} \tag{14}$$

As the robot accumulates odometry error by moving, the variance of the observations relative to the robot will increase. Let the positions of an observation expressed in $\mathcal{F}_r$ before and after a small movement described by $\boldsymbol{u}$ be ${}^r\boldsymbol{o}_i$ and ${}^r\boldsymbol{o}'_i$ respectively.

$$ {}^r\boldsymbol{o}'_i = \boldsymbol{T}(\phi)^T\ {}^r\boldsymbol{o}_i - \begin{bmatrix} u\cos(\theta_r+\phi) \\ u\sin(\theta_r+\phi) \end{bmatrix} \tag{15} $$

$$ \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial u} & \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \phi} \end{bmatrix} \tag{16} $$

$$ \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial u} = \begin{bmatrix} -\cos(\theta_r+\phi) \\ -\sin(\theta_r+\phi) \end{bmatrix} \tag{17} $$

$$ \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \phi} = \begin{bmatrix} -\sin\phi & \cos\phi \\ -\cos\phi & -\sin\phi \end{bmatrix} {}^r\boldsymbol{o}_i + \begin{bmatrix} u\sin(\theta_r+\phi) \\ -u\cos(\theta_r+\phi) \end{bmatrix} \tag{18} $$

To a first order approximation

$$ \boldsymbol{\Sigma}_{\boldsymbol{x}'} = \boldsymbol{\Sigma}_{\boldsymbol{x}} + \frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}} \boldsymbol{\Sigma}_{u,\phi} \frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}^T \tag{19} $$

$$ {}^r\boldsymbol{\Sigma}_{\boldsymbol{o}'_i} = {}^r\boldsymbol{\Sigma}_{\boldsymbol{o}_i} + \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \boldsymbol{u}} \boldsymbol{\Sigma}_{u,\phi} \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \boldsymbol{u}}^T \tag{20} $$

After each small movement,

$$ \boldsymbol{\Sigma}_{\boldsymbol{o}_i} = \boldsymbol{\Sigma}_{\boldsymbol{o}_i} + \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \boldsymbol{u}} \boldsymbol{\Sigma}_{u,\phi} \frac{\partial\, {}^r\boldsymbol{o}'_i}{\partial \boldsymbol{u}}^T \tag{21} $$

The sonar takes a reading every 100 ms. If the same artifact is observed more than once by the sonar then it makes sense to combine observations in some fashion to decrease the number of observations that need to be stored. This is achieved by means of a Kalman filtered estimate. The measurement model is

$$ h(\boldsymbol{x}, \boldsymbol{o}, \epsilon) = (\boldsymbol{o} - \boldsymbol{x} - \boldsymbol{v}_s) \cdot \boldsymbol{T}(\theta_r)\ {}^r\hat{\boldsymbol{v}}_b + \epsilon \tag{22} $$

$$ \boldsymbol{H} = \frac{\partial h}{\partial \boldsymbol{x}} \tag{23} $$

$$ = -\boldsymbol{T}(\theta_r)\ {}^r\hat{\boldsymbol{v}}_b \tag{24} $$

The covariance of the new observation $\boldsymbol{o}_z$ is given by equation 12 but the measurement model is only sensitive to errors in the direction along the sonar beam therefore

$$ \hat{\boldsymbol{v}}_b = \boldsymbol{T}(\theta_r)\ {}^r\hat{\boldsymbol{v}}_b \tag{25} $$

$$ \boldsymbol{\Sigma}_\epsilon = \hat{\boldsymbol{v}}_b \boldsymbol{\Sigma}_{\boldsymbol{o}_z} \hat{\boldsymbol{v}}_b \tag{26} $$

The gain matrix is defined as

$$ \boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{o}_k} \boldsymbol{H}^T \left(\boldsymbol{H} \boldsymbol{\Sigma}_{\boldsymbol{o}_k} \boldsymbol{H}^T + \boldsymbol{\Sigma}_\epsilon \right)^{-1} \tag{27} $$

The position and covariance of the updated observation $\boldsymbol{o}_{k+1}$ is

$$ \boldsymbol{o}_{k+1} = \boldsymbol{o}_k + \boldsymbol{K}\left(r_s - h(\boldsymbol{x}, \boldsymbol{o}_k, 0)\right) \tag{28} $$

$$ \boldsymbol{\Sigma}_{\boldsymbol{o}_{k+1}} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\, \boldsymbol{\Sigma}_{\boldsymbol{o}_k} \tag{29} $$

2.2 Minimization algorithm

The function $L_{\boldsymbol{x}}$ defined in equation 4 is generally continuous around its minimum, although it may posses discontinuities and other local minima (if the modification described in §2.3 is applied). The function is unlikely to have value 0 at its minimum, and computing the function value can be relatively demanding if there are large numbers of observations and artifacts.

These factors lead to the choice of a quasi-Newton minimization method as described by Fletcher [1987] and commonly known as the BFGS method. This method maintains an approximation to the inverse Hessian of the cost function, which is updated after every step. A line search is performed every step, but it is terminated whenever a simple set of conditions known as the "Wolfe conditions" is satisfied. The Wolfe conditions do not require the minimum along the search direction to be found, but merely require that a suitable descent has occurred. Convergence is generally superlinear.

In practise the algorithm has been found to converge well when the number of steps taken is limited to 1. Thus a single line search is performed every time the sensor data is updated. The inverse Hessian matrix provides a good estimate of the covariance matrix of the result.

2.3 Updating the map

In a real environment, it is a very common occurrence for the robot to encounter an object that is not in its world map. If the assumption that all observations correspond to artifacts is relaxed, then the method of section 2 is inadequate.

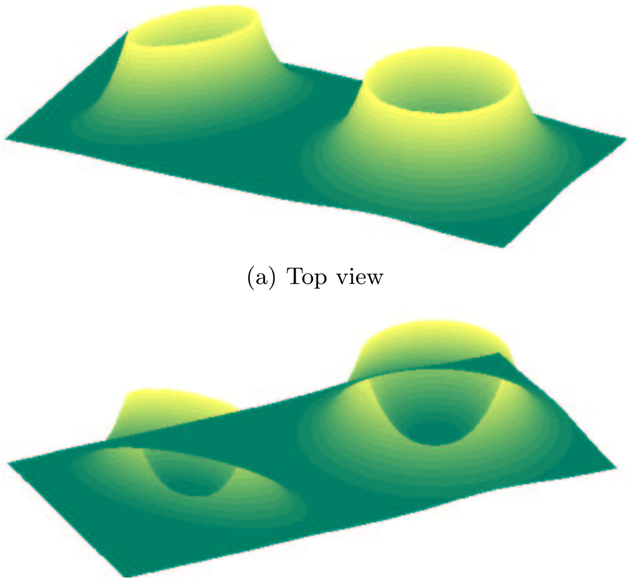
An improved formulation evaluates the likelihood that an observation corresponds to an artifact and assigns it to the most likely artifact. If it is more likely that the observation does not correspond to an artifact then it is designated a rogue observation and the probability that it does not correspond to an observation is added to the likelihood function. For a given observation $\boldsymbol{o}_i$

$$ Q_i = \max\Big(P(\chi^2_{i1}), P(\chi^2_2), \ldots, P(\chi^2_m), \prod_{j=1}^{m} \Big(1 - P(\chi^2_{ij})\Big) \Big) \tag{30} $$

The last term in equation 30 represents the probability that $\boldsymbol{o}_i$ is a rogue observation. The log likelihood function then becomes

$$ L_{\boldsymbol{x}} = -\sum_{i=1}^{n} Q_i + C \tag{31} $$

The shape of the log-likelihood function $L_{\boldsymbol{x}}$ is shown in Figure 3. The map has a single observation and 2 artifacts. The robot's position $\boldsymbol{x}$ is varied over a 2-dimensional region. The function is bowl shaped with a



(a) Top view

(b) Bottom view

Figure 3: Typical log-likelihood function $L_{\boldsymbol{x}}$ evaluated over a region containing 2 artifacts.

clearly defined minimum in the vicinity of the artifacts. Away from the artifacts, the function has a low value, reflecting the probability that the observation does not belong to either artifact.

If an object exists but is not included in the world map, then the robot will accumulate observations of it but not attribute them to a known artifact. Using the technique of §2.1 in which observations are combined with a Kalman filter, it is possible to determine when an object has been observed with enough certainty to merit adding it to the world map. In trials, it was found that the least singular value (LSV) of the observation's covariance matrix was suitable for this. When the LSV of $\boldsymbol{\Sigma}_{\boldsymbol{o}_i}$ is less than a certain threshold, a map artifact is created with the same position and covariance matrix as the observation.

Just as a particularly well defined observation can lead to the addition of a map artifact, other observations can improve the accuracy of existing map artifacts. Each time a new observation is recorded and associated with a known artifact, the estimate of the artifact's position can be improved with the use of a Kalman filter.

The measurement model is similar to equation 22, except that the map artifact is substituted for the observation.

$$h(\boldsymbol{x}, \boldsymbol{a}, \epsilon) = (\boldsymbol{a} - \boldsymbol{x} - \boldsymbol{v}_s) \cdot \boldsymbol{T}(\theta_r)\, \hat{\boldsymbol{v}}_b + \epsilon \tag{32}$$

$$\boldsymbol{H} = \frac{\partial h}{\partial \boldsymbol{x}} \tag{33}$$

$$= -\boldsymbol{T}(\theta_r)\, \hat{\boldsymbol{v}}_b \tag{34}$$

and

$$\hat{\boldsymbol{v}}_b = \boldsymbol{T}(\theta_r)\, {}^r\hat{\boldsymbol{v}}_b \tag{35}$$

$$\boldsymbol{\Sigma}_\epsilon = \hat{\boldsymbol{v}}_b \boldsymbol{\Sigma}_{\boldsymbol{a}_z} \hat{\boldsymbol{v}}_b \tag{36}$$

The gain matrix is defined as

$$\boldsymbol{K} = \boldsymbol{\Sigma}_{\boldsymbol{a}_k} \boldsymbol{H}^T \left(\boldsymbol{H} \boldsymbol{\Sigma}_{\boldsymbol{a}_k} \boldsymbol{H}^T + \boldsymbol{\Sigma}_\epsilon \right)^{-1} \tag{37}$$

The position and covariance of the updated map artifact $\boldsymbol{a}_{k+1}$ is

$$\boldsymbol{a}_{k+1} = \boldsymbol{a}_k + \boldsymbol{K} \left(r_s - h(\boldsymbol{x}, \boldsymbol{a}_k, 0) \right) \tag{38}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{a}_{k+1}} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\, \boldsymbol{\Sigma}_{\boldsymbol{a}_k} \tag{39}$$

2.4 Discussion

The probabilistic localisation procedure was tested by postprocessing sonar data from field trials because the computational demands of solving the minimization problem in the Matlab environment repeatedly in real time were too high. Overall the approach worked effectively and could be made to work in real time if re-coded in a compiled language. The technique, if developed more rigorously may provide an alternative to the commonly used Markov localisation technique [Thrun *et al.*, 1998, 2000]. The chief difference between the techniques is that the technique presented here limits the amount of information that needs to be stored about the global map, reducing the computational demands. Old observations have less influence as the robot progresses and eventually they are discarded. This is useful in dynamic environments where objects may move.

A shortcoming of the method was that rogue observations exerted a small repulsive effect on the robot's position, which is not realistic. If there were not well identified artifacts in the model, the repulsive effects would dominate and the robot would drift away from the rogue observation. This could be fixed by refining the cost function so that $Q_i = 0$ if the last term in equation 30 is the largest. The minimization algorithm would need to be adjusted to be more robust in situations where all observations are rogue observations if this change were made.

Another shortcoming was that there is currently no mechanism for removing a map artifact if repeated observations indicate that it does not exist. This involves a predictive model, which is capable of increasing the variance of a map artifact if it is not sensed. Objects with large variance can be removed from the map.

Occasionally, if the map had regularly repeating features, the BFGS solver would find itself at a bifurcation in the cost function, which resulted in the robot jumping to the wrong location. This is helped by starting with a realistic estimate of the inverse Hessian, so that the line search is well directed and properly scaled.

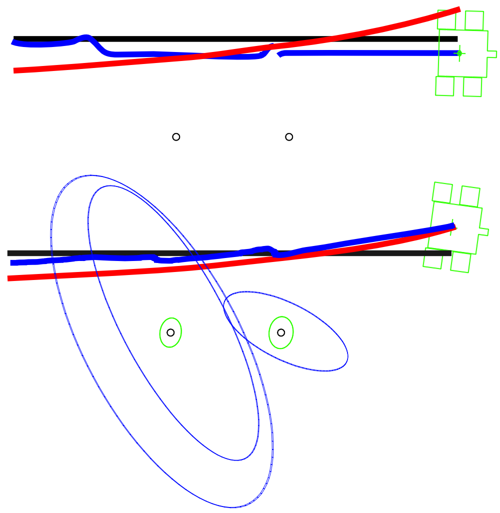
The following tests were performed:

1. Localisation relative to 2 objects, correcting for position and orientation despite sensing the objects at different times.
2. Localisation in a known map consisting of 6 objects arranged in 2 rows. The robot maintained localisation and was able to add artifacts to the map representing a number of surrounding objects.
3. Map building in an environment containing 5 objects. The robot started with an empty map and was moved about the environment. Eventually all 5 objects were identified along with part of a wall situated at the edge of the test area. The robot was able to use the newly identified artifacts for localisation during the map building process.

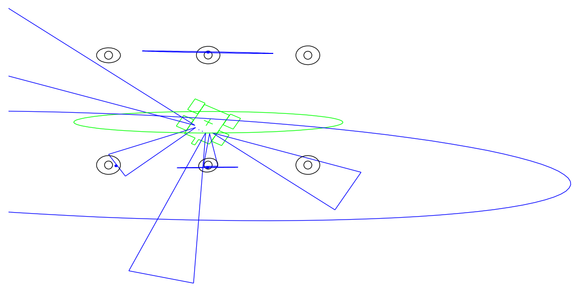
3 Conclusion

This work has demonstrated that the SLAM technique put forward by Dissanayake *et al.* [2001] can be successfully adapted to work with sonar sensors. This has been shown to work in outdoor environments containing point objects.

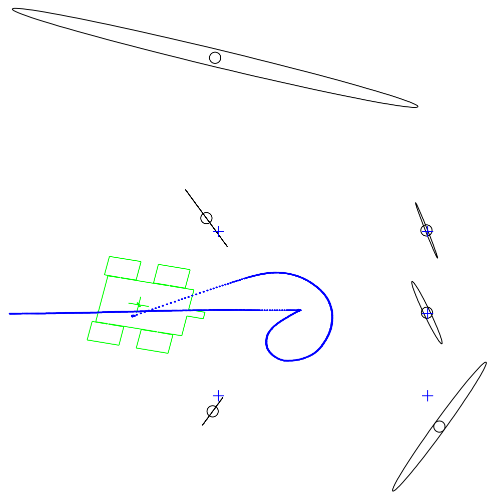
A new approach to localisation and map building has been proposed and simulated in simple environments. The method has been shown to be capable of rich inferences from sonar data that is widely spaced in space and time. The method requires further development in order to become robust enough for rough terrain and cluttered environments.



(a) Localisation relative to 2 objects. Top: SLAM localisation. Bottom: Probabilistic localisation. The red line is the world line of the robot, the black line is the ideal position, and the blue line is the localisation algorithm's position estimate.



(b) Localisation using probabilistic modelling. The green ellipse is a 95% confidence region on the robot's position, the blue ellipses (some very narrow) are confidence regions about observations.



(c) Map building from an initially empty map. Ellipses are 95% confidence regions about the expected object location. Blue crosses indicate the true positions of the objects. The top ellipse is a nearby wall that was inadvertently sensed.

Figure 4: Trials of the probabilistic method.

References

[Dissanayake, 2001] M. W. M. Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions of Robotics and Automation*, 17(3):229–241, jun 2001.

[Fletcher, 1987] R. Fletcher. *Practical Methods of Optimisation*. John Wiley and Sons, second edition, 1987.

[Lintott, 2002] Andrew Lintott. Mobile robot navigation field trial 18-1-2002. Short Video, January 2002. Available from Industrial Research Limited, PO Box 20-028, Christchurch.

[Thrun, 1998] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probablistic approach to concurrent mapping and localization for mobile robots. In *Machine Learning*, volume 31, pages 29–53. 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands., 1998.

[Thrun, 2000] S. Thrun, M. Beetz, M. Bennewitz, et al. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research*, 19(11):972–999, nov 2000.