Available online at www.sciencedirect.com

ScienceDirect

Robotics and Autonomous Systems 55 (2007) 3–20

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Using covariance intersection for SLAM

Simon J. Julier[a,*], Jeffrey K. Uhlmann[b]

[a] *Department of Computer Science, University College London, London, WC1E 6BT, UK*
[b] *327 Engineering Building West, University of Missouri — Columbia, Columbia, MO 65211-2060, USA*

Received 1 November 2005; received in revised form 1 April 2006; accepted 1 June 2006
Available online 18 October 2006

Abstract

One of the greatest obstacles to the use of Simultaneous Localization And Mapping (SLAM) in a real-world environment is the need to maintain the full correlation structure between the vehicle and all of the landmark estimates. This structure is computationally expensive to maintain and is not robust to linearization errors. In this tutorial we describe SLAM algorithms that attempt to circumvent these difficulties through the use of Covariance Intersection (CI). CI is the optimal algorithm for fusing estimates when the correlations among them are unknown. A feature of CI relative to techniques which exploit full correlation information is that it provides provable consistency with much less computational overhead. In practice, however, a tradeoff typically needs to be made between estimation accuracy and computational cost. We describe a number of techniques that span the range of tradeoffs from maximum computational efficiency with straight CI to maximum estimation efficiency with the maintenance of all correlation information. We present a set of examples illustrating benefits of CI-based SLAM.

Keywords: Mapping and localisation; Linearization errors; Unmodelled correlations; Covariance intersection; Scalability

1. Introduction

The seminal work on the rigorous application of the Kalman filter to the SLAM problem was carried out in the late 1980s by Smith, Self and Cheeseman [1] and was extended by Moutarlier and Chatila [2]. Prior to their work, a number of authors had attempted to address the problem of SLAM by assuming (often implicitly) that the landmark and vehicle estimates could be propagated independently of one another. However, all of these early efforts exhibited problems that manifested themselves in the form of slowly drifting the landmark estimates leading to inconsistent map and vehicle estimates [3,4]. What was shown in [1] was that a complete joint covariance matrix comprising the vehicle and all landmark estimates must be maintained in order to rigorously apply Kalman filtering. This Full Covariance SLAM (FC-SLAM) solution has many favourable theoretical properties. However, maintaining the joint covariance matrix has two significant difficulties associated with it. First, the computational and storage costs grow quadratically with number of landmarks. Second, the joint covariance structure is very sensitive to the effects of linearization errors, which are unavoidable in practical applications. These linearization errors accumulate through time and cause the filter to diverge.

To address these problems, many authors have developed alternative formulations and applied various types of non-Kalman filtering algorithms to the SLAM problem. Many of these approaches exploit the fact that SLAM has a highly specialized and extremely simple structure. One deficiency of most alternatives to the FC-SLAM approach is the lack of rigorous performance guarantees. In this tutorial we focus on one particular method, Covariance Intersection (CI) that does provide mathematically provable performance guarantees that are achievable in practice.

CI is a fusion algorithm for combining two or more estimates (e.g. running state estimates and sensor measurement) when the correlation among them is unknown. As such, CI does not succumb to the problems suffered by FC-SLAM resulting from an inability to maintain perfect knowledge of the correlations existing among all estimates. On the other hand, CI is not able

[*] Corresponding author.
E-mail addresses: S.Julier@cs.ucl.ac.uk (S.J. Julier), uhlmannj@missouri.edu (J.K. Uhlmann).
URLs: http://www.cs.ucl.ac.uk/people/S.Julier (S.J. Julier), http://www.cs.missouri.edu/facultypages/uhlmann.html (J.K. Uhlmann).

0921-8890/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.robot.2006.06.011

to exploit information that could be derived from correlation information if it were available. In this tutorial we show how CI can be applied so that a tradeoff can be made between estimation accuracy and computational efficiency. We argue that this approach can be adapted to provide the best achievable results for most practical systems.

We note that we do not consider the problems of data association and loop closing. Although these are extremely important topics for SLAM, we believe that they do not lie within the scope of this tutorial. Rather, we note that data association algorithms work best with accurate mean and covariance estimates.

The structure of this tutorial is as follows. Sections 2 and 3 introduce the full covariance formulation of the SLAM problem, analyse some of its properties, and identify its two main weaknesses: strong limits on scalability and lack of robustness to linearization errors. The Covariance Intersection (CI) algorithm is introduced in Section 4 and its properties are discussed. An extension to CI, known as Split Covariance Intersection, is introduced in Section 5. Together, these two algorithms form the building block for a hybrid SLAM approach described in Section 6. A summary is provided in Section 7.

2. The structure of full covariance SLAM

The Full Covariance (FC-SLAM) formulation of the SLAM problem was developed by the seminal papers from Smith, Self and Cheeseman [1]. The environment is assumed to be populated by a set of landmarks that can be detected by sensors fixed to a platform.

The state space for a map of N landmarks is:

$$\mathbf{x}(k) = \left[\mathbf{x}_v(k)\ \mathbf{p}_1^{\mathrm{T}}(k)\ \ldots\ \mathbf{p}_N^{\mathrm{T}}(k)\right]^{\mathrm{T}}$$

where $\mathbf{x}_v(k)$ is the vehicle state and $\mathbf{p}_i(k)$ is the location of the ith landmark. The mean and covariance of this estimate are:

$$\hat{\mathbf{x}}(k \mid k) = \left[\hat{\mathbf{x}}_v^{\mathrm{T}}(k \mid k) \ldots \hat{\mathbf{p}}_N^{\mathrm{T}}(k \mid k)\right]^{\mathrm{T}} \tag{1}$$

$$\mathbf{P}(k \mid k) = \begin{pmatrix} \mathbf{P}_{vv}(k \mid k) & \mathbf{P}_{v1}(k \mid k) & \ldots & \mathbf{P}_{vN}(k \mid k) \\ \mathbf{P}_{1v}(k \mid k) & \mathbf{P}_{11}(k \mid k) & \ldots & \mathbf{P}_{1N}(k \mid k) \\ \mathbf{P}_{2v}(k \mid k) & \mathbf{P}_{21}(k \mid k) & \ldots & \mathbf{P}_{2N}(k \mid k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{Nv}(k \mid k) & \mathbf{P}_{N1}(k \mid k) & \ldots & \mathbf{P}_{NN}(k \mid k) \end{pmatrix} \tag{2}$$

where $\mathbf{P}_{vv}(k \mid k)$ is the covariance of the vehicle's position estimate, $\mathbf{P}_{ii}(k \mid k)$ is the covariance of the position estimate of the ith landmark and $\mathbf{P}_{ij}(k \mid k)$ is the cross-correlation between the estimate of landmarks i and j.

By assumption, all of the landmarks are stationary, so there is no accumulation of process noise. Thus, the process model takes the form:

$$\mathbf{f}\left[\mathbf{x}_N(k-1), \mathbf{u}(k-1), \mathbf{v}(k-1)\right] = \begin{pmatrix} \mathbf{f}_v\left[\mathbf{x}_v(k-1), \mathbf{u}(k-1), \mathbf{v}(k-1)\right] \\ \mathbf{0} \end{pmatrix}$$

where $\mathbf{u}(k-1)$ is the control input and $\mathbf{v}(k-1)$ is the process noise. Similarly, the process noise covariance matrix is:

$$\mathbf{Q}(k) = \begin{bmatrix} \mathbf{Q}_v(k) & \ldots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix}.$$

The observation model for the vehicle's observation of the ith landmark is:

$$\mathbf{z}_i(k) = \mathbf{h}_i\left[\mathbf{x}_v(k), \mathbf{p}_i(k), \mathbf{w}(k)\right] \tag{3}$$

where $\mathbf{w}(k)$ is the observation noise with covariance $\mathbf{R}(k)$. Whenever the vehicle observes a landmark which is in the map, the joint system is updated using the Kalman filter update equations.

The Kalman filter update equations are:

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\boldsymbol{\nu}(k) \tag{4}$$

$$\mathbf{P}(k \mid k) = \mathbf{P}(k \mid k-1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}(k)^{\mathrm{T}} \tag{5}$$

where $\boldsymbol{\nu}(k)$ is the innovation vector, $\mathbf{S}(k)$ is its covariance, and $\mathbf{W}(k)$ is the Kalman gain. These values are computed by

$$\boldsymbol{\nu}(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k \mid k-1) \tag{6}$$

$$\hat{\mathbf{z}}(k \mid k-1) = \mathbf{h}_i\left[\hat{\mathbf{x}}_v(k \mid k-1), \hat{\mathbf{p}}_i(k \mid k-1), \mathbf{0}\right] \tag{7}$$

$$\mathbf{C}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^{\mathrm{T}}(k) \tag{8}$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{C}(k) + \mathbf{R}(k) \tag{9}$$

$$\mathbf{W}(k) = \mathbf{C}(k)\mathbf{S}(k)^{-1} \tag{10}$$

and $\mathbf{H}(k)$ is the Jacobian of $\mathbf{h}_i\left[\cdot, \cdot, \cdot\right]$ evaluated around $\hat{\mathbf{x}}(k \mid k-1)$ (hence $\hat{\mathbf{x}}_v(k \mid k-1)$ and $\hat{\mathbf{p}}_i(k \mid k-1)$).

When a vehicle observes a landmark that does not exist in the map, the landmark can be initialized from the inverse observation model:

$$\mathbf{p}_i(k) = \mathbf{g}_i\left[\mathbf{x}_v(k), \mathbf{z}_i(k), \mathbf{w}(k)\right]. \tag{11}$$

We require the condition that the estimate is: *consistent* [5]. In other words,

$$\mathbf{P}(i \mid j) - \mathrm{E}\left[\tilde{\mathbf{x}}(i \mid j)\tilde{\mathbf{x}}^{\mathrm{T}}(i \mid j)\right] \geq \mathbf{0} \tag{12}$$

where $\tilde{\mathbf{x}}(i \mid j) = \mathbf{x}(i) - \hat{\mathbf{x}}(i \mid j)$ is the error in the estimate and $\geq \mathbf{0}$ means that the result is positive semidefinite.[1]

3. Limitations with FC-SLAM

FC-SLAM has been used in several practical applications with apparent success. However, empirical results often do not show the effects of slow divergence or sensitivity to the particulars of a given scenario. For example, does the filter diverge if the scenario is extended in time or if the number of landmarks is increased? Increasing the vehicle covariance after

[1] Bar-Shalom also stipulates that the error is zero-mean [5]. However, because this requires the unrealistic assumptions that there are no errors in the process models, the observation models, or biases caused by linearization and other simplifications, we prefer our more general definition.

updates (sometimes regarded as 'tuning') can mask the visible effects of filter divergence for a while, but it cannot prevent it.

In practical applications of FC-SLAM there are two major limitations: scalability and robustness. We illustrate these in the following subsections.

3.1. Scalability

The first, and best known, difficulty associated with FC-SLAM is its scalability requirements: for a map of $O(N)$ landmarks, the storage is proportional to $O(N^2)$ (because it is dominated by the contribution of the beacon states in the covariance matrix), and the computational complexity of the updates are $O(N^2)$ because each observation of a single landmark causes the entire map to update. This means that the FC-SLAM algorithm can only be applied in real time for relatively small maps (with only a few hundred landmarks) and relatively low sensor update rates (a few tens of updates per second). Of the two, it is arguable that the high update cost is a more significant issue than the storage cost. This is because it is typically easier to increase the amount of storage available than it is to increase the processing speed.

To understand why the $O(N^2)$ update and storage complexities are necessary for FC-SLAM, consider the covariance structure depicted in (2), which has $O(N^2)$ cross covariance terms. In the Kalman filter equations, these cross terms define the correlations between the state variables. Terms of the form $\mathbf{P}_{vi}(k \mid k)$ represent the correlation between the vehicle estimate and the i landmark; correlations of the form $\mathbf{P}_{ij}(k \mid k)$ represent the correlations between pairs of landmarks. The cross covariances are typically nonzero, i.e. the estimates are all correlated, because all observations of landmarks include errors due to uncertainties in the vehicle's own position estimate. Because all the estimates are correlated, an update to one will provide information for all the others. More specifically, when one landmark is observed, *all* landmarks in the map are updated. Ignoring the cross covariances implies that all of the estimates are uncorrelated and that the joint covariance has the following structure:

$$\mathbf{P}(k \mid k) = \begin{pmatrix} \mathbf{P}_{vv}(k \mid k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{11}(k \mid k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}_{NN}(k \mid k) \end{pmatrix}.$$

In this case the storage complexity becomes $O(N)$ because only the diagonal blocks corresponding to the vehicle and landmark estimates need to be maintained. Furthermore, the Kalman filter equations in this case would require only $O(1)$ computations per update because the updated estimate is not correlated with any other estimates. Some authors have attempted the use of such a strategy in highly specialized contexts despite the fact that filter consistency cannot be assured. Welch, for example, assumed uncorrelated landmark estimates for a head tracking system [6].

Although the cross covariances are the source of the large computational overhead associated with FC-SLAM, they serve a critical role in maximizing the amount of information that is exploited at each update. In particular, they define which errors are common among the vehicle and landmark estimates so that an observation that reduces the error in one estimate can reduce the same error in all other correlated estimates. Conversely, the cross covariance between the vehicle and each landmark also defines the amount of novel information each observation truly provides. For example, if the vehicle estimate and a landmark estimate are highly correlated, then an observation of that landmark by the vehicle will provide less update information than if the landmark estimate were uncorrelated with the vehicle estimate [7,8].

Vehicle
Beacon

Fig. 1. A simple one beacon example: the stationary vehicle attempts to build a map consisting of a single beacon.

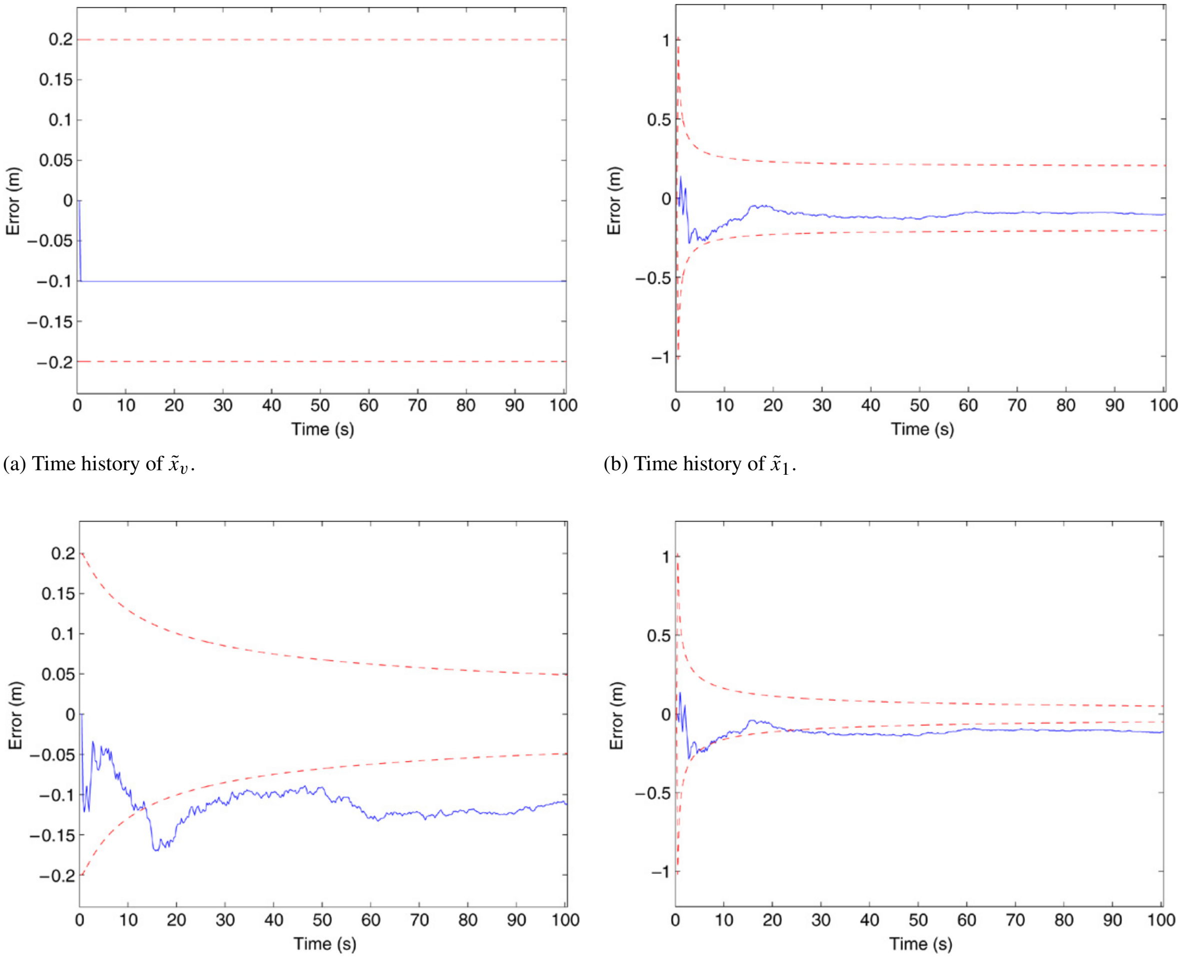
The effects of neglecting correlations can be illustrated by the simple example shown in Fig. 1: a stationary vehicle at (x_v, y_v, θ_v) observes a stationary landmark at (x_i, y_i) using a linear observation model. Full details of this example are given in Appendix A.

Fig. 2(a) and (b) show the behaviour of the vehicle position and landmark position estimates.[2] As can be seen, the vehicle estimate does not change but the landmark estimate improves. There are two reasons for this. First, because the vehicle can only measure the position of the landmark relative to itself, observing the same landmark in this static configuration means that no additional information about the vehicle's absolute position can be gained. Second, when the landmark is initialized, its estimate includes a component due to the observation error. The independence of the observation noise means that this contribution is filtered away over time.

Fig. 2(c) and (d) show the results with assumed independence. Because the filter treats the estimates as independent, it assumes that new information is continuously being provided. Consequently, the landmark estimates become spuriously small and 'converge' to incorrect positions. In Welch's tracking example, the filter appeared to work because the initial positions of the landmark estimates had errors with a standard deviation of 1 mm, and so the final position errors were of similarly small magnitude. In other words, the filter yielded inconsistent estimates, but the magnitude of the inconsistency was deemed tolerable for purposes of the particular application.[3] In such cases there is actually little reason to filter the landmark estimates at all.

[2] In this example the orientation is completely decoupled from position. Its value does not change and so we do not present its results here.

[3] Another, somewhat more refined version of this approach is the Sparse Extended Information Filter (SEIF) [9] The SEIF assumes that correlations below a threshold are 'small' and can be neglected. By eliminating 'small' correlations, the effects of the approximation are less severe than simply ignoring all the correlations irrespective of their magnitude. However, the resulting filter is not consistent [10].



(a) Time history of $\tilde{x}_v$.

(b) Time history of $\tilde{x}_1$.

(c) Time history of $\tilde{x}_v$.

(d) Time history of $\tilde{x}_1$.

Fig. 2. Time histories of the errors in the vehicle and platform x estimates. The figures in the first row show the results when the correlation structure is maintained, the second when the correlation information is ignored. The $\pm 1\sigma$ bounds are shown as the pair of dashed lines.

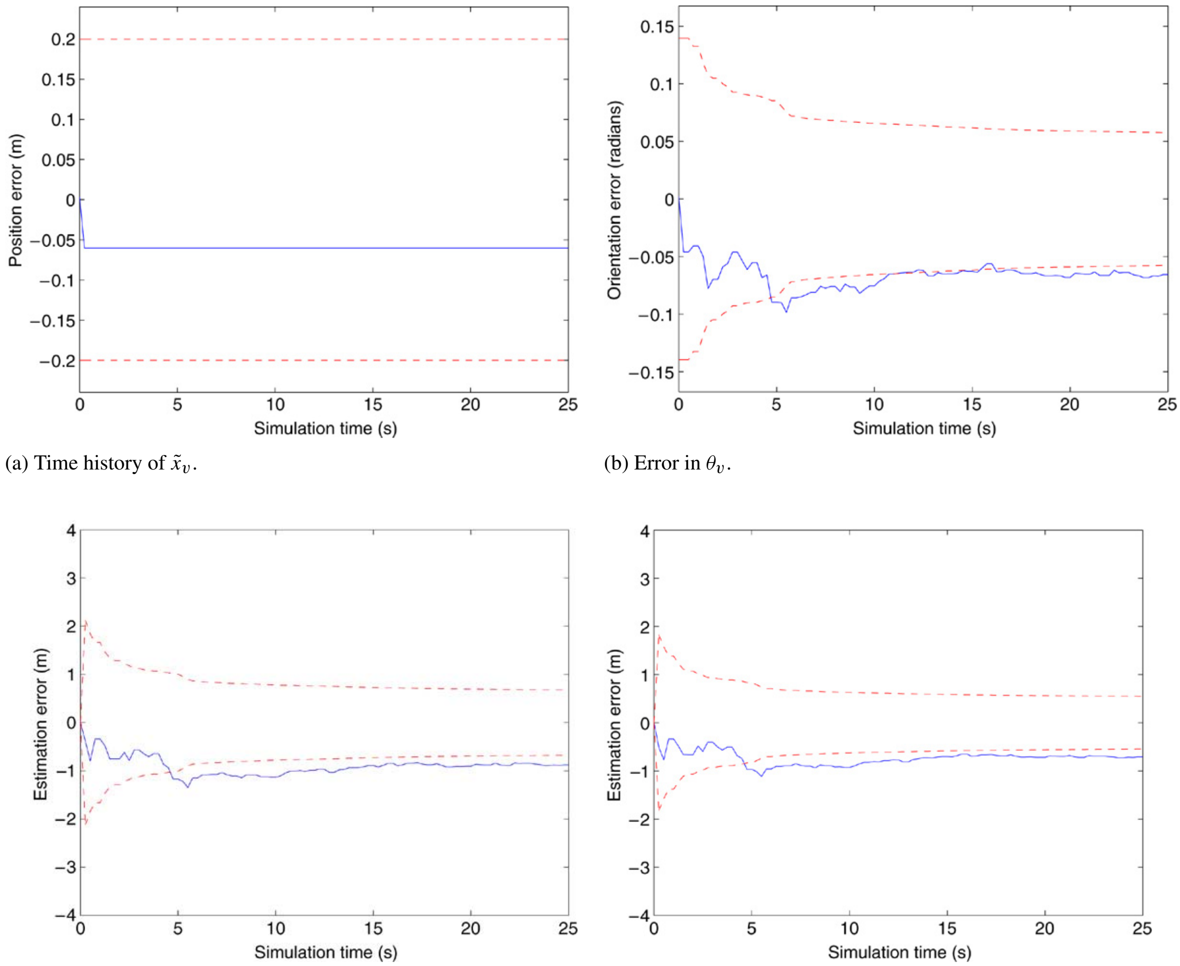
Some approaches for reducing computational costs while maintaining equivalence with FC-SLAM have been proposed. One method is postponement [11,12], where the full FC-SLAM is applied to a small subset of landmarks and the changes to all other landmarks are batched up and applied at a later time. A second method is to use map management schemes [13] in which most of the landmarks are discarded and only a minimal landmark "trail" is maintained. Postponement amortizes the computational costs but does not reduce the overall scaling and computational costs. Map management schemes can be used to reduce the total number of landmarks, but it cannot avoid the quadratic scaling. Thus, neither approach is generally practical.

3.2. *Consistency and linearization issues*

The second difficulty with FC-SLAM is that it requires precise cross covariance information. If a cross covariance is not *exactly* correct, updates will produce inconsistent results. For example, if the magnitude of the cross covariance between the vehicle and a landmark is too small, then an observation of the landmark will be treated as providing more novel information than what is actually available. Conversely, if the magnitude of the cross covariance is too large, then an update of the vehicle estimate will lead to a spuriously large update to the landmark because its errors are assumed to be more correlated with the vehicle estimate than what is actually the case.

The sensitivity of the FC-SLAM to imprecise cross covariances is a serious one because there is no general way to 'approximate' them in a way that avoids problems — no entry in the cross covariance matrix can be too large or too small. This means that the filter will inevitably succumb to the accumulation of errors from all sources ranging from numerical roundoff errors to unmodelled correlations in the sensor reports. One particularly significant contributor to incorrect cross covariances is linearization of nonlinear models.

Almost any practical SLAM application will involve nonlinear vehicle and/or observation models. These models can only be approximately applied within the Kalman filter equations using linearization of some kind. Unfortunately, the errors incurred by such approximations are introduced into both the vehicle and landmark estimates, but because the errors are unknown, the effect on the cross covariance cannot be determined. The effects can be demonstrated by returning to the stationary vehicle case described above [14]. Instead of equipping the vehicle with a sensor that directly measures



(a) Time history of $\tilde{x}_v$.

(b) Error in θ_v.

(c) Error in x_1.

(d) Error in y_1.

Fig. 3. Estimation errors and 2 standard deviation bounds. The standard deviation bounds are shown as dashed lines.

the distance between the platform and the beacon in (x, y) coordinates, the vehicle measures the range and bearing (r, ϕ).

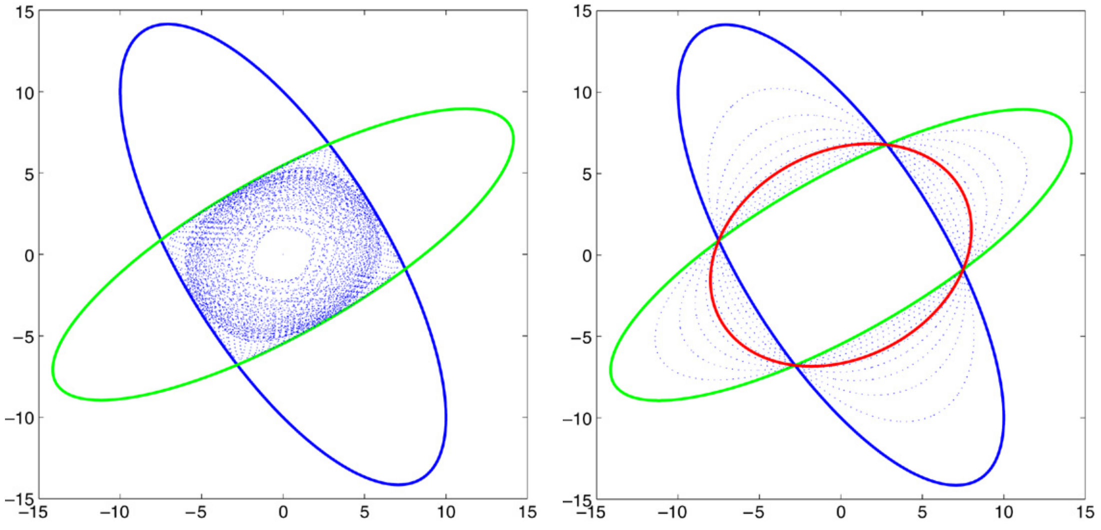
As before, we expect that the vehicle state should not change because the sensor only measures beacon positions relative to the vehicle. In other words, $\hat{\mathbf{x}}_v(k \mid k) = \hat{\mathbf{x}}_v(0 \mid 0)$ for all timesteps k. From (4), this occurs only if the Kalman gain can be written in the form:

$$\mathbf{W}(k) = \begin{bmatrix} \mathbf{0}_v \\ \mathbf{W}_p(k) \end{bmatrix}$$

where $\mathbf{0}_v$ is a zero vector with the same dimensions as $\mathbf{x}_v(k)$. In [14] it was proved that this only occurs if there is a particular mathematical relationship between the Jacobian of (11), evaluated at the time the landmark was intialized, and the Jacobian of (3), evaluated at the current observation time step. This condition always holds when the observation model is linear. However, it does not, in general, hold for nonlinear systems. One case where this fails for the range-bearing sensor is if the beacon is updated so that it no longer lies at its initialized position. The effect of observation noise is shown in Fig. 3: rather than remain constant, the vehicle covariance is reduced through time. Furthermore, because this condition is a *structural* relationship, normal tuning procedures (such as inflating the observation noise covariances) cannot circumvent the problem. If the vehicle estimate changes for one value of $\mathbf{R}(k)$, it will change for all (finite) values of $\mathbf{R}(k)$.

Castellanos has experimentally corroborated these observations in a scenario in which a vehicle moves around a closed circuit, and he argues that the effects become significant long before scalability becomes an issue [15]. Bailey has shown that the principle cause of this problem is the uncertainty in the rotation of the platform [16]. Furthermore, replacing the Kalman filter by more sophisticated methods (such as the Unscented Kalman Filter [17] or the Iterated Extended Kalman Filter [18]) does not significantly affect these results. The reason is that *any* errors will ultimately undermine the integrity of the filter.

It could be argued that some of these difficulties can be overcome by carefully structuring the environment. Probabilistic information is often used in data association problems, where a landmark is detected but its identity must be deduced probabilistically. It also arises in loop closing to identify when a vehicle revisits part of the map. Therefore, if all landmarks have a unique coding scheme (such as a bar code used in the VisTracker [19] or unique texture patterns such as the vision-tracking system developed by Davison [20]), association is not needed. However, Bailey



(a) Kalman results.

(b) CI results.

Fig. 4. The shape of the updated covariance ellipses for the Kalman and CI algorithms. The covariance ellipses for input estimates A and B are the thick outer ellipses. In the left figure, the dashed lines show the ellipses for various values of the KF update for different correlations between A and B [18]. The result with assumed independence is the thick solid line. The right figure shows the CI estimate C using different values of ω. The determinant minimizing solution ($\omega = 0.4615$) is shown with the thick solid line.

further notes that the inconsistency causes the vehicle estimate to move along a jagged path: the vehicle pose estimate shows dramatic jumps which tend to be disproportionately large when compared with the size of the measurement error. Such behaviour is problematic in safety-critical applications because it undermines the ability to confidently assess risk based on the available covariance information.

3.3. Summary

The analysis in the past two sections shows that FC-SLAM has two significant problems that preclude its use in all but the most controlled real-world applications. First, its computational and storage costs mean that it can only be used in maps that are composed of, at most, only a few hundred landmarks. Second, the accumulation of linearization errors mean that the filter will eventually become inconsistent.

Both of these problems stem from the need to propagate and maintain the cross correlations among the landmark estimates. One means of avoiding this restriction would be to develop an update rule that does not rely on knowledge of the cross correlations between the state estimates.

4. CI-SLAM

4.1. Covariance intersection

Covariance Intersection (CI) is a fusion rule for combining two estimates when the correlations between them are unknown. Suppose that two pieces of information, labelled A and B, are to be fused together to yield an output C. This is a general type of data fusion problem — A and B could be two different sensor measurements, or A could be a prediction from a system model and B could be sensor information. Consistent estimates for A and B are known and can be expressed as the mean and covariance pairs $\{\hat{\mathbf{a}}, \mathbf{A}\}$ and $\{\hat{\mathbf{b}}, \mathbf{B}\}$. However, the cross correlation between A and B are unknown.

The CI update yields the mean and covariance pair $\{\hat{\mathbf{c}}, \mathbf{C}\}$ from the equations [21,22]:

$$\mathbf{C}^{-1} = \omega\mathbf{A}^{-1} + (1-\omega)\mathbf{B}^{-1}$$
$$\hat{\mathbf{c}} = \mathbf{C}\left(\omega\mathbf{A}^{-1}\hat{\mathbf{a}} + (1-\omega)\mathbf{B}^{-1}\hat{\mathbf{b}}\right).$$

If $\omega \in [0, 1]$, the resulting estimate is guaranteed to be consistent. Moreover, it can be shown to be optimal for the case in which the cross covariance is unknown [21].

The intuition behind this form is illustrated in Fig. 4.[4] This figure shows the covariance ellipses (locus of points $\mathbf{x}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{x} = 1$) for A and B and various values of the update C. The left hand figure shows the update when the correlations between the estimates are known. As can be seen, the updated covariance ellipse lies within the intersection region of the covariance ellipses for A and B. Because the correlations are unknown, it is not possible to know which ellipse is the correct one. The CI algorithm is shown on the right. As can be seen, the covariance ellipses circumscribe the intersection region but do not lie within it. Furthermore, the ellipses intersect the corners of the intersection region, showing that no other family of robust update rules can yield a tighter estimate.

The figure also shows the role of ω: it chooses which, out of the set of ellipses, should be chosen as the update. ω must be chosen through an optimization process to minimize some criteria of uncertainty. In all of our experiments we use $\|\mathbf{C}\|$.

The CI algorithm can be written in a form very similar to that of the Kalman filter(4)–(10):

[4] It should be noted that CI does *not* assume that the means are coincident or the distributions are bounded. The analysis in [23] shows that $\{\hat{\mathbf{c}}, \mathbf{C}\}$ is consistent for *any* distribution.

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}(k)\boldsymbol{\nu}(k)$$

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

where

$$\mathbf{C}(k) = \frac{1}{\omega}\mathbf{P}(k \mid k-1)\mathbf{H}^{\mathrm{T}}(k)$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{C}(k) + \frac{1}{(1-\omega)}\mathbf{R}(k)$$

$$\mathbf{W}(k) = \mathbf{C}(k)\mathbf{S}(k)^{-1}.$$

4.2. Applying CI to SLAM

CI can be applied to the SLAM problem in the following way. The joint state space of the vehicle and beacons is decomposed into $N+1$ separate systems that are estimated independently. The first state space consists of the vehicle state $\{\hat{\mathbf{x}}_v(k \mid k), \mathbf{P}_{vv}(k \mid k)\}$. The remaining states correspond to each landmark $\{\hat{\mathbf{p}}_i(k \mid k), \mathbf{P}_{ii}(k \mid k)\}$. This decomposition of the joint filtering problem of $O(N^2)$ complexity into $O(N)$ independent filters dramatically simplifies the problem and increases the efficiency to $O(1)$ complexity per update. Unlike the simple decorrelation method described in Section 3.1, this method is guaranteed to be consistent.

The update is carried out in two independent steps: the vehicle is updated from the landmark, and the landmark is updated from the vehicle.

4.2.1. Updating the vehicle estimate

The observation to the landmark, given the vehicle estimate and the landmark estimate, is given by the observation model

$$\mathbf{z}_{v,i}(k) = \mathbf{h}_i\left[\mathbf{x}_v(k), \mathbf{p}_i(k), \mathbf{w}_v^*(k)\right]$$

where $\mathbf{w}_v^*(k)$ is the observation noise formed by concatenating together the sensor observation and the landmark observation to form a pseudo-vehicle observation noise,

$$\mathbf{w}_v^*(k) = \begin{bmatrix} \mathbf{w}(k) \\ \tilde{\mathbf{p}}_i(k \mid k) \end{bmatrix} \tag{13}$$

where $\tilde{\mathbf{p}}_i(k \mid k)$ is the error in the ith beacon estimate. The covariance of this pseudo-vehicle observation noise term is

$$\mathbf{R}_v^*(k) = \begin{pmatrix} \mathbf{R}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{ii}(k \mid k-1) \end{pmatrix}.$$

The Jacobians are calculated using the extended observation model and the update is performed using the CI algorithm.

4.2.2. Updating the landmark estimate

The landmark is updated in a similar way — the vehicle estimate is projected into the landmark's coordinate space using the initialization function

$$\mathbf{z}_{i,v}(k) = \mathbf{g}_i\left[\mathbf{x}_v(k), \mathbf{z}(k), \mathbf{w}_i^*(k)\right]$$

where $\mathbf{w}_i^*(k)$ is the observation noise formed by concatenating together the sensor observation and the landmark observation,

$$\mathbf{w}_i^*(k) = \begin{bmatrix} \mathbf{w}(k) \\ \tilde{\mathbf{x}}_v(k \mid k-1) \end{bmatrix} \tag{14}$$

with covariance

$$\mathbf{R}_i^*(k) = \begin{pmatrix} \mathbf{R}(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{vv}(k \mid k-1) \end{pmatrix}.$$

4.3. Example

To test the robustness to linearization errors, Fig. 5 shows the results applied to the stationary landmark case. As can be seen, the vehicle does not update. This is a consequence of the fact that, for all time, the landmark covariance exceeds that for the vehicle covariance. However, the landmark covariance does not reduce either. The reason is that there is no new additional information that CI can exploit. If the geometry changed because, for example, the vehicle moved to a new position, new information would become available and the landmark estimate would update.

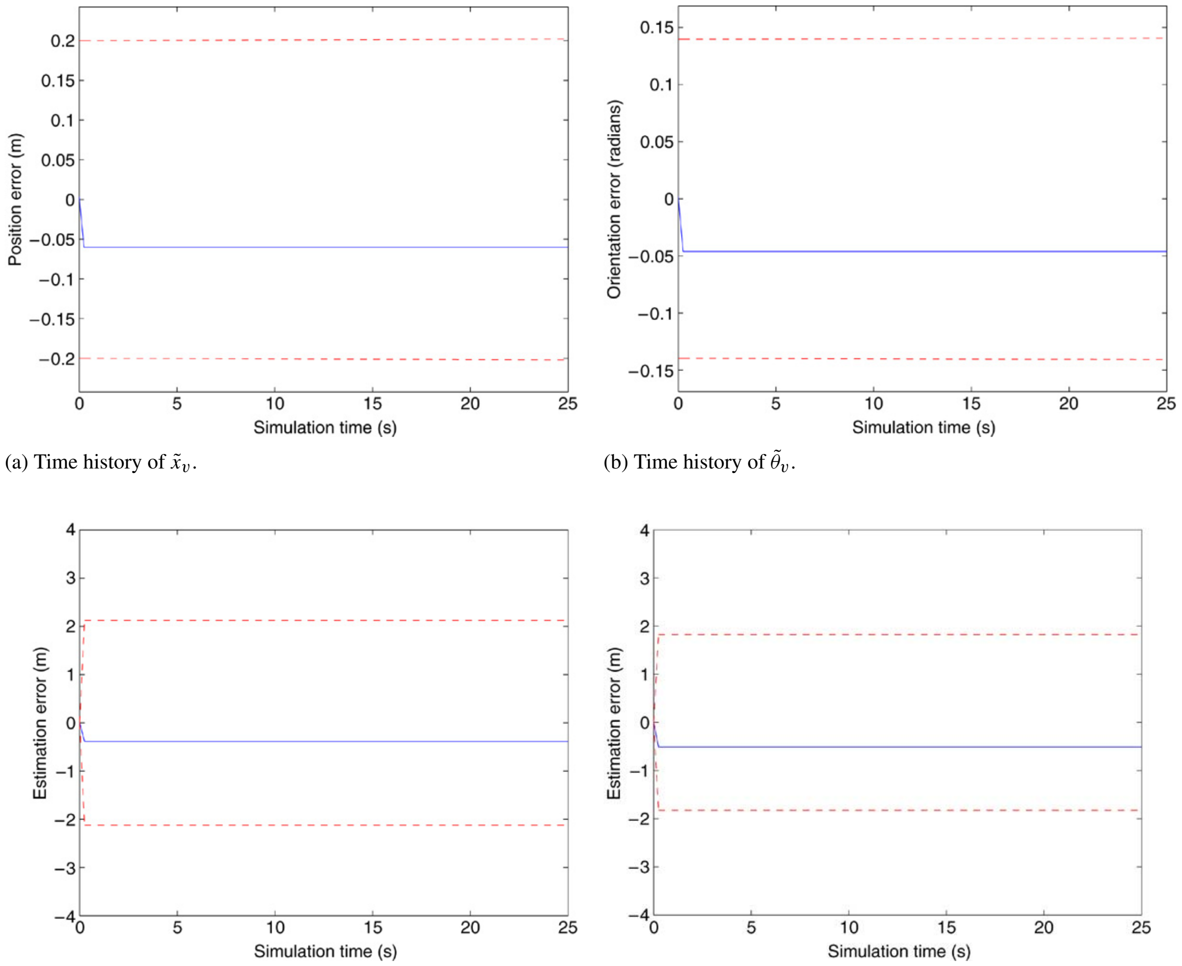
These effects are quantified in the second scenario shown in Fig. 6. The vehicle is placed in the middle of a 400 m by 400 m field populated with 250 randomly placed landmarks. The vehicle initially spirals out from the middle of the map until it nears the periphery, and then begins to spiral back in again. This is a challenging case for SLAM: the vehicle initially and continuously moves away from regions where it has surveyed landmarks. Because of the linearization issues pointed out earlier, the orientation error must be bounded. Therefore, the vehicle uses an ideal compass to measure the orientation of the vehicle. The system equations can be found in Appendix A.

Fig. 7 shows the resulting estimates in x_v and θ_v for the FC-SLAM and CI-SLAM algorithms after 17,883 time steps. At this point, 184 landmarks have been initialized and an average of 3.02 landmarks were observed at each time step. The full SLAM results, in Fig. 7(a) and (b) show that the position variance initially increases and decreases again. The CI algorithm performs extremely well. Despite the fact that, in this example, it utilizes less than 1% of the storage and computational costs of the full SLAM algorithm, the 2 standard deviation bounds are less than a factor of 3 times larger than those of FC-SLAM. This result is very significant for practical applications because it reveals how little performance gain may be provided by the cross covariance information for the amount of computational resources that must be applied.

Schlegel [24] has shown how CI can be applied to a graph-theoretic form for loop closing and illustrated the results by constructing a map using approximately 400 laser scans. Austin [25] has applied CI to the problem of data association in SLAM.

4.4. Summary

In this section we have described how CI can be applied to the SLAM problem to develop an algorithm that addresses the computational and storage requirements constraint of FC-SLAM. Furthermore, CI does not exhibit the same fragility with respect to linearization errors. However, it does so at the cost of ignoring all potential correlation information. In this particular example, the result is a covariance estimate that is 3-times larger than the optimal value. Whether this factor of 3



(a) Time history of $\tilde{x}_v$.

(b) Time history of $\tilde{\theta}_v$.

(c) Time history of $\tilde{x}_1$.

(d) Time history of $\tilde{y}_1$.

Fig. 5. Estimation errors and 2 standard deviation bounds. The standard deviation bounds are shown as dashed lines.

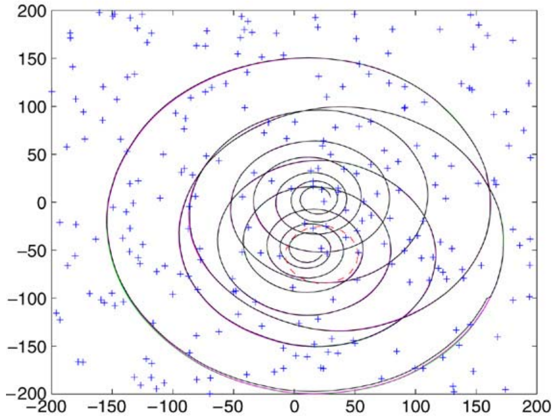


Fig. 6. The test scenario. The vehicle starts from the middle and spirals outwards. Landmark positions are '+'. The dashed circle indicates the range over which the vehicle can detect landmarks.

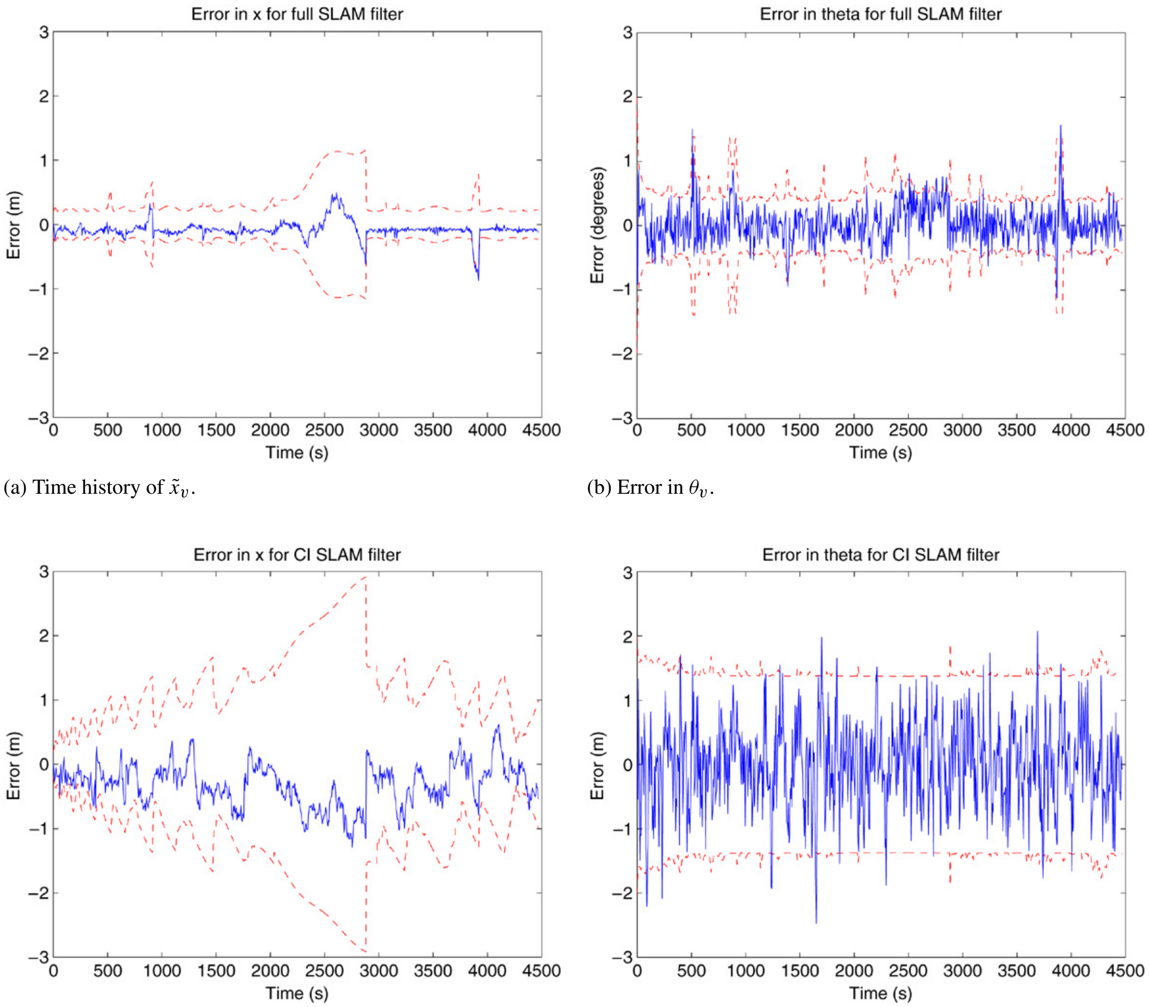
represents an absolute error that is significant depends on the application. For example, the difference between a 1 mm and a 3 mm error for a typical indoor vehicle may be negligible, but the difference between a 1 m and a 3 m error for an outdoor vehicle that must traverse tightly-spaced obstacles may be significant.

A critical question is whether the structure of the SLAM problem can exploit additional information within the CI framework to achieve even better filter accuracy. One means of obtaining a more accurate estimate would be to exploit the known independent information which is available in the system. For example, the observation noise $\mathbf{w}(k)$ appears in both (13) and (14). Because this observation has not been used in the landmark or vehicle estimates before, it can be updated using the conventional Kalman filter. To exploit this additional information, CI must be extended to maintain estimates whose correlation structure is partially known. This is discussed in the next section.

5. SCI-SLAM

5.1. Split covariance intersection (SCI)

The Split Covariance Intersection (SCI) algorithm [22] extends CI by allowing some information about the cross



(a) Time history of $\tilde{x}_v$.

(b) Error in θ_v.

(c) Time history of $\tilde{x}_v$.

(d) Error in θ_v.

Fig. 7. The estimated error and 2σ bounds for FC-SLAM and CI-SLAM. In each figure, the estimated error is the solid line and the $\pm 2\sigma$ bounds are the dashed lines.

correlations to be exploited. Specifically, we assume that the error in the input estimate A can be decomposed into two terms[5]

$$\tilde{\mathbf{a}} = \tilde{\mathbf{a}}_I + \tilde{\mathbf{a}}_C.$$

The first term is the fraction of the error that is known to be independent of all other sources. The second term is the fraction of error that may be correlated with errors in other estimates. The mean of each term cannot be known individually. However, the covariance of each term is $\mathbf{A}_I$ and $\mathbf{A}_C$ and the overall covariance $\mathbf{A}$ is:

$$\mathbf{A} = \mathbf{A}_I + \mathbf{A}_C.$$

With this more expressive representation of state, a more general definition of consistency is required. We adopt a consistency measure that requires two tests. First, the overall covariance in A must exceed the mean squared error in the estimate:

$$\mathbf{A} - \mathrm{E}\left[\tilde{\mathbf{a}}\tilde{\mathbf{a}}^{\mathrm{T}}\right] \geq \mathbf{0}. \tag{15}$$

Second, the amount of known independent information must not be overestimated:

$$\mathbf{A}_I - \mathrm{E}\left[\tilde{\mathbf{a}}_I\tilde{\mathbf{a}}_I^{\mathrm{T}}\right] \leq \mathbf{0}. \tag{16}$$

Assuming that B can be decomposed in a similar manner with error terms $\tilde{\mathbf{b}}_I$ and $\tilde{\mathbf{b}}_C$ with covariances $\mathbf{B}_I$ and $\mathbf{B}_C$, and obey consistency conditions (15) and (16), the SCI algorithm is:

$$\mathbf{C}^{-1} = \mathbf{A}_*^{-1} + \mathbf{B}_*^{-1}$$

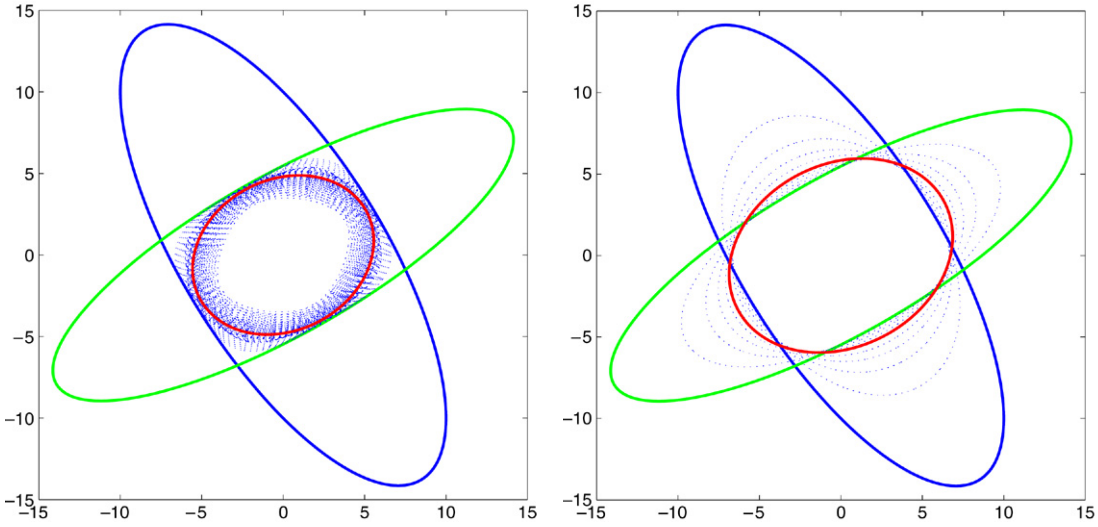
$$\mathbf{c} = \mathbf{C}\left(\mathbf{A}_*^{-1}\mathbf{a} + \mathbf{B}_*^{-1}\mathbf{b}\right)$$

where:

$$\mathbf{A}_* = \mathbf{A}_I + \frac{1}{\omega}\mathbf{A}_C$$

$$\mathbf{B}_* = \mathbf{B}_I + \frac{1}{1-\omega}\mathbf{B}_C,$$

[5] It should be noted that other methods of capturing partial correlation information can be devised. Reece, for example, has developed an algorithm known as *Bounded Covariance Inflation* that quantifies known information in terms of a bound on the cross correlation matrix [26]. They have demonstrated the use of this algorithm for multiple vehicle SLAM. It is possible that this algorithm can be used in single vehicle SLAM, but this remains an open research question.



(a) Kalman results.

(b) CI results.

Fig. 8. The shape of the updated covariance ellipses for the Kalman and SCI algorithms. In the left figure, the dashed lines show the ellipses for various values of the KF update for different correlations between A_C and B_C only. The result with assumed independence is the thick solid line. The right figure shows the SCI estimate C using different values of ω. The determinant minimizing solution ($\omega = 0.4780$) is the inner, rounder covariance ellipse.

and $\omega \in [0, 1]$. By collecting the independent error terms and taking outer products, $\mathbf{C}$ can be decomposed into known correlation and unknown correlation terms:

$$\begin{aligned}\mathbf{C}_I &= \mathbf{C}\mathbf{A}_*^{-1}\mathbf{A}_I\mathbf{A}_*^{-1}\mathbf{C} + \mathbf{C}\mathbf{B}_*^{-1}\mathbf{B}_I\mathbf{B}_*^{-1}\mathbf{C},\\ \mathbf{C}_c &= \mathbf{C}\mathbf{A}_*^{-1}\mathbf{A}_C\mathbf{A}_*^{-1}\mathbf{C} + \mathbf{C}\mathbf{B}_*^{-1}\mathbf{B}_C\mathbf{B}_*^{-1}\mathbf{C}.\end{aligned} \tag{17}$$

It can be shown [22] that the updated estimate is guaranteed to be consistent with respect to (15) and (16).

The split estimates are beneficial because the ω scaling penalty is only applied to the unknown correlated components. It can be shown that SCI is optimal given the available information and is consistent for all choices of $0 \leq \omega \leq 1$. Furthermore, as the correlated terms go to 0, SCI converges to the conventional Kalman filter. As the independent terms go to 0, SCI converges to the CI algorithm.

The results of this algorithm are shown in Fig. 8. This uses the same estimates as those used to construct Fig. 4, but $\mathbf{A}_I = 0.5\mathbf{A}$, $\mathbf{B}_I = 0.5\mathbf{B}$. The effect of the known independence is shown on the left by the fact that the estimate is more tightly bound within the intersection region. The result of using this information is illustrated on the right hand side: the ellipses calculated by the SCI algorithm now lie partially within the intersection region.

As with CI, SCI can be expressed in terms of the Kalman filter equations. Defining:

$$\mathbf{P}_*(k \mid k-1) = \frac{1}{\omega}\mathbf{P}_C(k \mid k-1) + \mathbf{P}_I(k \mid k-1)$$

$$\mathbf{R}_*(k) = \frac{1}{1-\omega}\mathbf{R}_C(k) + \mathbf{R}_I(k),$$

the split CI algorithm is[6]

$$\begin{aligned}\mathbf{C}_*(k) &= \mathbf{P}_*(k \mid k-1)\mathbf{H}^{\mathrm{T}}(k)\\ \mathbf{S}_*(k) &= \mathbf{H}(k)\mathbf{C}_*(k) + \mathbf{R}_*(k)\\ \mathbf{W}_*(k) &= \mathbf{C}_*(k)\mathbf{S}_*(k)^{-1}\\ \hat{\mathbf{x}}(k \mid k) &= \hat{\mathbf{x}}(k \mid k-1) + \mathbf{W}_*(k)\boldsymbol{\nu}(k)\\ \mathbf{P}(k \mid k) &= \mathbf{P}_*(k \mid k-1) - \mathbf{W}_*(k)\mathbf{S}_*(k)\mathbf{W}_*(k)^{\mathrm{T}}.\end{aligned} \tag{18}$$

5.2. Using split CI with SLAM

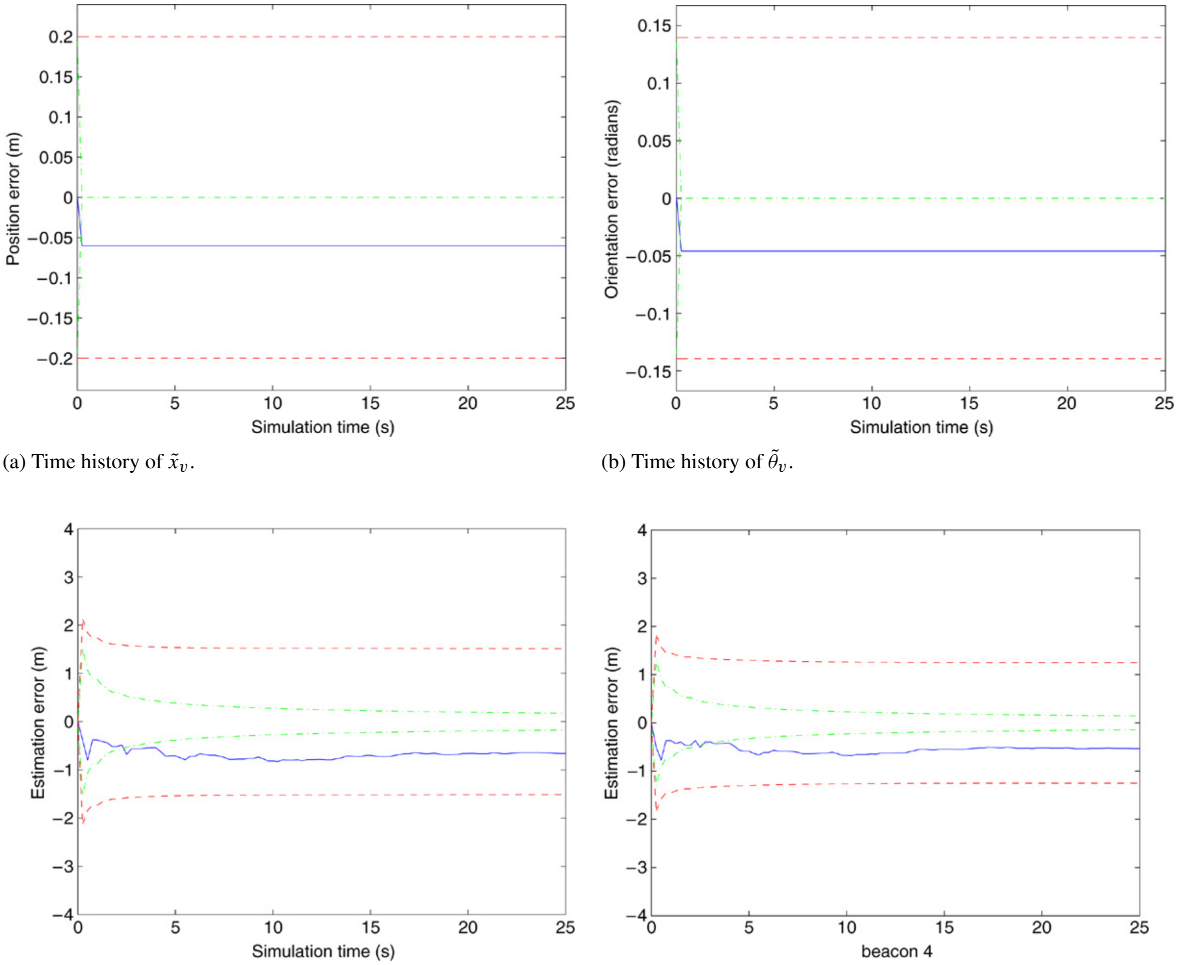
The SLAM problem naturally decomposes into independent and correlated terms. The independent terms arise from the process noise which accumulates between times when the vehicle observes landmarks, and the independent observation noises when the vehicle observes landmarks. The correlated terms arise from the vehicle-landmark cross-correlation terms which are not maintained.

The basic update can be performed using the two pass approach applied with CI. However, it is possible to exploit additional information from the availability of some cross covariance information. Suppose, for example, that in the first stage the vehicle updates from a landmark estimate. As a result the observation noise is now incorporated into the vehicle estimate. However, if the landmark is not updated, the noise from the observation is not incorporated into the landmark estimate. As a result, no additional unmodelled error terms arise. Appendix B enumerates four rules for modifying the independent and correlated terms in the landmark and vehicle estimates. These rules simply look at the ω values post the update, and thus operate in constant time.

5.3. Example

The results for the stationary scenario are shown in Fig. 9. As with the CI case, the vehicle does not update. However, the landmark estimates continue to update. The inner line corresponds to the independent observation information and, as

[6] This result was later independently derived by Xu in the context of SLAM using images to construct a mosaic-based map [27]. However, Xu did not employ the independent update formulation described in (17).



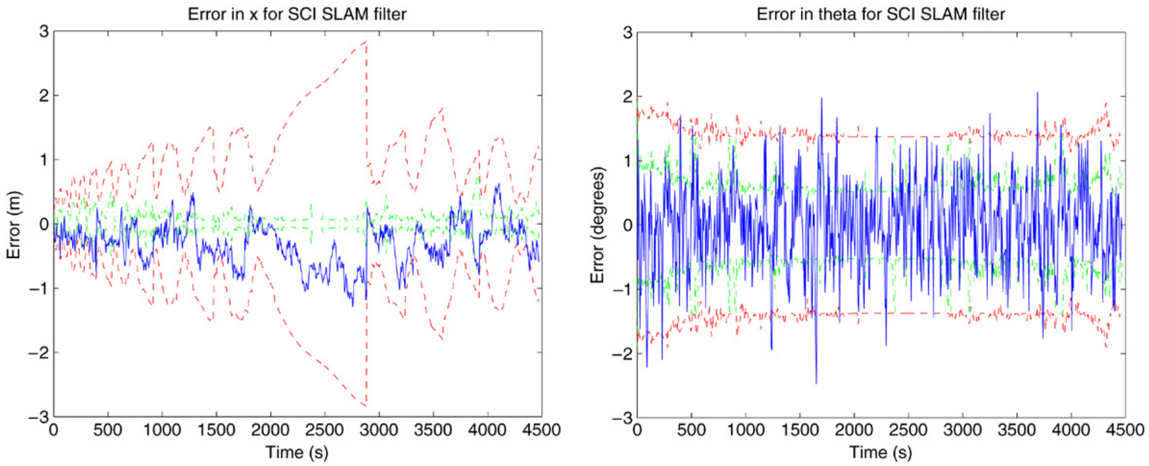
(a) Time history of $\tilde{x}_v$.

(b) Time history of $\tilde{\theta}_v$.

(c) Time history of $\tilde{x}_1$.

(d) Time history of $\tilde{y}_1$.

Fig. 9. Estimation errors and 2 standard deviation bounds. The standard deviation bounds for the combined component are the dashed lines, for the independent component the dash-dotted lines.



(a) Error in x.

(b) Error in θ_v.

Fig. 10. The estimated error and 2σ bounds for the SCI-SLAM algorithm. The 2σ bounds for the independent components are shown as the inner pair of dashed lines.

can be seen, is filtered away through time. The overall landmark variance reduces only a small amount, however, because it is dominated by the initial uncertainty in the vehicle position.

The results for the moving vehicle scenario are shown in Fig. 10. At first sight, this figure appears very similar to the results from the CI filter shown in Fig. 7. However, the

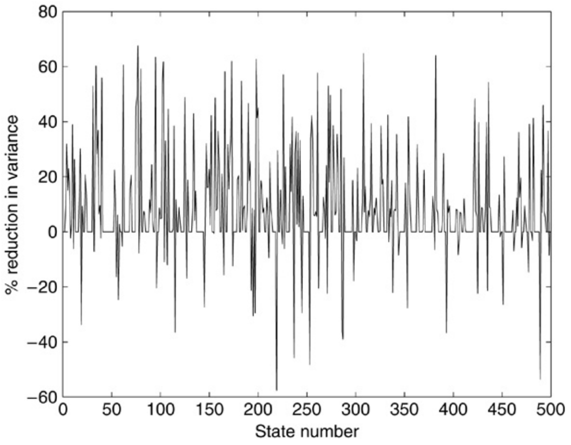


Fig. 11. The percentage reduction in the variance in each landmark state from CI to SCI.

determinant of $\mathbf{P}_{vv}(k \mid k)$ is less than half of that for the vehicle using the CI update. Furthermore, there have been significant increases in the accuracy of the estimated landmark locations. Fig. 11 plots the change in covariance of the landmark estimates. The average reduction in landmark variance is approximately 32% which corresponds to a 17% reduction in the standard deviation in the landmark estimates. Again, it should be noted that the computational and storage costs of the SCI-SLAM algorithm are much smaller (approximately 2%) than those required for FC-SLAM.

The plots also show that independent information is consistently maintained within much of the operation of the filter and that the level remains approximately constant. For the position estimate, the contribution is relatively small compared to the overall position estimate and is probably due to the injection of process noise in the prediction step. It is a much greater proportion of the orientation error and is probably due to the observation noise from the compass.

Xu [27] independently investigated this approach (calling it Extended Covariance Intersection) and demonstrated its use in an image mosaicking application. An underwater autonomous vehicle collected a series of images and that had to be mosaicked together to build a map of the environment. Because of correlations in the mosaicking process, the Kalman filter could not be effectively applied.

5.4. Summary

This section has described Split Covariance Intersection, an extension to CI that can be used to maintain some fraction of an estimate with a known correlation structure. A constant time algorithm has been described, and it has been shown that it can improve the performance of the filter. However, these results seem disappointing: despite the great increase in complexity, the variance of the SCI-SLAM algorithm is still approximately three times that of FC-SLAM. These results are probably due to the highly conservative (but constant time) algorithm used and other non constant time (but bounded complexity) algorithms could be used. Furthermore, CI solutions have been developed which exploit information about bounds on the cross correlation terms [28,26] and it is possible that these could be applied to SLAM as well.

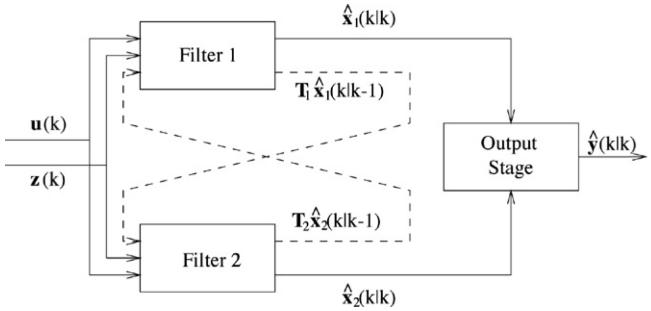

$\hat{\mathbf{x}}_1$(k|k)
Filter 1
$\mathbf{T}_1\hat{\mathbf{x}}_1$(k|k-1)
u(k)
z(k)
Output Stage
$\hat{\mathbf{y}}$(k|k)
$\mathbf{T}_2\hat{\mathbf{x}}_2$(k|k-1)
Filter 2
$\hat{\mathbf{x}}_2$(k|k)

Fig. 12. The architecture of a hybrid SLAM system. Each filter implements its own SLAM algorithm and the filters can exchange information with one another.

The significance of SCI is that it provides the mechanism to mix information with known and unknown correlation structures together. This gives the freedom to develop hybrid approaches to SLAM that are provably robust and, at the same time, can exploit multiple SLAM algorithms to encode independent information even more effectively.

6. Hybrid approaches

6.1. Basic requirements

One means for combining the particular performance advantages of different SLAM algorithms in a single system is illustrated in Fig. 12. Suppose F different filters have been implemented. Each filter uses its own representations of the vehicle and the environment as well as its own fusion algorithm, but every filter receives the same observation and control input information. Therefore, each filter maintains its own estimate of the state of the environment. This is related to the concept of *model fusion* [29,30]. The original derivation of model fusion was applied to the problem of modelling a system whose process and observation models are imperfectly known. The solution involved a set of models defined under different assumptions (i.e. models) of the system dynamics and behaviours. Assuming that each model can be tuned so that it yields consistent estimates, the different models can in principle provide differing amounts of information about different states and can thus collectively provide more information than is available from any of the models individually. The challenge is to be able to combine the estimates from different models, which are generally correlated in practice to an unknown extent, to obtain a consistent fused estimate.

There are two ways to support interactions among the algorithms — *direct* and *indirect*.

6.1.1. Direct hybrid architectures

Let the common space for a pair of filters i and j be $\mathbf{S}_{ij}$. Let $\mathbf{t}_{i\to ij}\left[\mathbf{x}_i(k)\right]$ and $\mathbf{t}_{j\to ij}\left[\mathbf{x}_j(k)\right]$ be the functions which map $\mathbf{x}_i(k-1)$ and $\mathbf{x}_j(k-1)$ to $\mathbf{S}_{ij}$. These parameters obey the condition that

$$\mathbf{t}_{i\to ij}\left[\mathbf{x}_i(k)\right] = \mathbf{t}_{j\to ij}\left[\mathbf{x}_j(k)\right]. \qquad (19)$$

In other words, the true value of the state in both filters should project to the same value in the common space. One example is that filter i could store all landmarks in a single global coordinate frame, whereas filter j could describe a submap whose transformation to world coordinates is known. In this case, $\mathbf{S}_{ij}$ is the space of beacons in world coordinates. Conversely, i could be a map of beacons in a global frame and j could use relative maps. In this case $\mathbf{S}_{ij}$ is the space of relative beacon estimates.

Given this relationship, the estimate from one filter can be used as an observation on a second filter by constructing an observation model of the form [29]

$$\mathbf{z}_i(k) = \left(\mathbf{t}_{j\rightarrow ij}\left[\mathbf{x}_j(k)\right]\ \right). \tag{20}$$

In general the estimates in each filter will not be independent of one another. To enforce practical computational and storage constraints, the maintenance of complete cross covariance information must be avoided through the use of CI or SCI.

6.1.2. Indirect hybrid architectures

Indirect hybrid architectures arise when there is no direct transfer of information between the different map representations. Rather, the results from one map can be used to guide update in another map. One example could be that a local map could be constructed to aid in associating sensor reports with landmarks. The association information would be used by other maps to update their estimates. An example of this is the relocation step which occurs in Atlas [37].

6.2. Example

In this section we describe an implementation that scales to more than one million landmarks in real time. To our knowledge, this is the largest map that has been built to date in real time, both in terms of the number of landmarks within the map and the number of timesteps used to construct it.[7] Such numbers may seem well beyond what might be needed in practice, but systems that are envisaged to track all discernible features in an environment can quickly reach this number of landmarks. For example, an autonomous robot travelling in an urban environment can potentially map hundreds of features from a single position [31]. This number of landmarks is necessary to provide sufficient position estimation redundancy to ensure that the vehicle can travel safely at all times.

Consider the covariance structure of the SLAM system as given by (1). Qualitatively, this covariance matrix contains three types of terms: the covariance of the state estimates, the cross correlations between the vehicle and the landmarks, and the cross correlations among the landmarks. This suggests the following simple strategy: If it is possible to develop 3 SLAM algorithms, each of which focuses on a particular correlation structure, then the combined filter will give better results than each filter acting on its own. We choose a SLAM algorithm that has the three components shown in Fig. 13: a local map for initialization, a relative map to maintain and update the correlations between the landmarks, and a Split CI map to maintain the vehicle and landmark states in world coordinates.

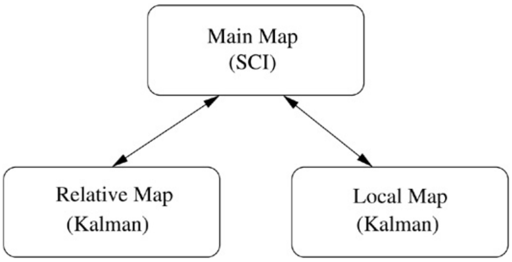


Fig. 13. The structure of the hybrid algorithm. The central representation is the split CI map. Data from the relative and global maps is fused into the central map.

6.2.1. Maintaining correlations between the beacons and the vehicle through a local map

The local map is an FC-SLAM map that contains a small number of landmarks $N_{\mathrm{lm}} \ll N$. Its purpose is to maintain the correlations between the vehicle and the beacon estimates. As discussed in Section 5.3, these correlations are particularly important when a beacon is first initialised because they are used to filter out the observation noise in the initialized beacon position.

Suppose that, at time k, the vehicle observes a new landmark j. If the local map is not active, a local map is created using the current vehicle state estimate (from the SCI map) and the initialized position of landmark j. If the local map is active and the number of landmarks is less than N_{lm}, the new landmark is inserted into the map. If the map has N_{lm} landmarks already, one landmark is deleted from the local map (but not from the SCI or relative maps discussed below)[8] before the new landmark estimate is created. In all cases, a new entry for the landmark is created and inserted in the SCI map and, if necessary, the relative map as well.

As the vehicle progresses through the environment, the vehicle estimate is predicted and updated using the normal FC-SLAM equations. Because the local map uses a subset of the available landmarks, the variance on the vehicle will, in general, end up being larger than that of the vehicle in the SCI map. Therefore, it is necessary to periodically reinitialize the local map. Since the purpose of the local map is to aid in beacon initialization, we use the simple heuristic that if a new beacon has not been initialized for a set period of time, the local map is deactivated. At that point, CI is used to fuse the estimates of the beacon and vehicle positions into the SCI map.

[7] Recently Thrun has published the GraphSLAM algorithm which uses graph methods to solve implicitly the SLAM problem [31]. The algorithm has been reported to solve systems with up to 10^7 variables in 30 s. However, the algorithm is an offline batch solution.

[8] In our implementation, we eliminated the landmark which had been updated the least recently. However, other more refined strategies such as Dissanayke's map management algorithm [32] could be used instead.

6.2.2. Maintaining the correlations between the beacons using a relative map

The *relative map* approach to SLAM, was pioneered by Csorba [33] and further developed by Newman [34]. It implicitly maintains the cross covariance information between the estimates of pairs of beacons and makes it possible for known information about pairs of beacons to propagate slowly throughout the whole map.

Rather than maintain a global estimate of all landmarks in the same coordinate space, the relative map estimates the relative configuration between a group of landmarks. A relative landmark estimate is defined to be

$$\mathbf{p}_{ij}(k) = \mathbf{p}_j(k) \ominus \mathbf{p}_i(k)$$

where the $\ominus$ defines an appropriate "difference" operator. Csorba [33] used the distance between pairs of landmarks and the angle subtended by three landmarks, whereas Newman [34] used Cartesian difference.

A relative map can be maintained if it is possible to construct a direct observation of $\mathbf{p}_{ij}(k)$ without the need to use the vehicle state or the observations of any other landmarks.

6.2.3. SCI map

This map maintains the estimates of the vehicle and all landmarks within a single global coordinate space. It uses the implementation described in Section 5.2 and thus the storage is linear in the number of landmarks and the computational costs are constant time. However, information from the local and relative maps can be fused into this map to ensure that the estimate is continuously improved.

6.2.4. Fusing the filters together

Because each SLAM algorithm shares observation and process noise terms, the correlations between the different estimates are unknown and CI must be used to fuse the estimates together.

Suppose the relative estimate $\hat{\mathbf{p}}_{ij}(k \mid k)$ is to be used to update the estimate of landmarks j. This is achieved by first constructing, from landmark i and the relative estimate, an estimate of the position of j:

$$\tilde{\mathbf{z}}_j(k) = \hat{\mathbf{p}}_i(k \mid k) \oplus \hat{\mathbf{p}}_{ij}(k \mid k).$$

However, because the same observations can be used in both the SCI and the relative maps, and because the same relative map estimate can update the same landmark multiple times, the correlations between $\hat{\mathbf{p}}_i(k \mid k)$ and $\hat{\mathbf{p}}_{ij}(k \mid k)$ are unknown. Therefore, the covariance of this estimate has to be calculated using *Covariance Addition* (CA) [22]. CA, which is closely related to CI, finds the smallest covariance that is guaranteed to be consistent when two variables with unknown correlations are added together. Its value is given by:

$$\tilde{\mathbf{R}}_j(k) = \frac{1}{\omega}\mathbf{P}_i(k \mid k) + \frac{1}{1-\omega}\mathbf{P}_{ij}(k \mid k)$$

where $0 \le \omega \le 1$ is chosen to minimize $\|\tilde{\mathbf{R}}_j(k)\|$. However, because $\tilde{\mathbf{z}}_j(k)$ is correlated with $\hat{\mathbf{p}}_j(k \mid k)$, we must use CI to fuse $\tilde{\mathbf{z}}_j(k)$ with $\hat{\mathbf{p}}_j(k \mid k)$. Combining the CA and CI steps together, the update for landmark j is

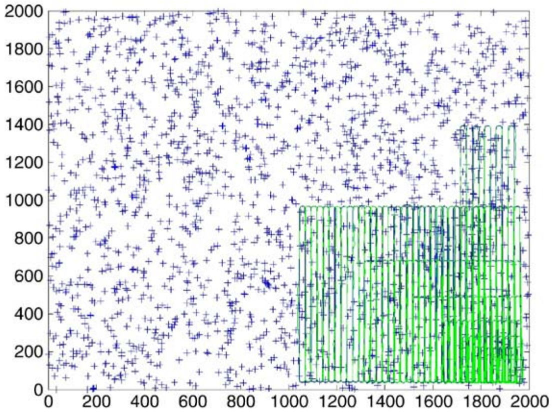


Fig. 14. A subset of the full scenario. This portion consists of approximately 2000 landmarks. The path of the vehicle is shown as the closely spaced lines and curves.

$$\mathbf{P}_j^{-1}(k \mid k) = \omega_1\mathbf{P}_j^{-1}(k \mid k-1) + (1-\omega_1)\tilde{\mathbf{R}}_j^{-1}(k)$$

$$\hat{\mathbf{p}}_j(k \mid k) \; = \; \mathbf{P}_j(k \mid k)\Big(\omega_1\mathbf{P}^{-1}(k \mid k-1)\hat{\mathbf{p}}_j(k \mid k-1) + \; (1-\omega_1)\tilde{\mathbf{R}}_j^{-1}(k)\tilde{\mathbf{z}}_j(k)\Big)$$

$$\tilde{\mathbf{R}}_j(k) = \frac{1}{\omega_2}\mathbf{P}_i(k \mid k) + \frac{1}{1-\omega_2}\mathbf{P}_{ij}(k \mid k)$$

where this equation must be optimized with respect to both ω_1 and ω_2. Since this is a convex optimisation problem, approaches such as Powell's Method [35] can be used.

The filtering of estimates in the relative map ensures that all landmark estimates converges to the same covariance in the limit as $k \to \infty$. The reason is that as $k \to \infty$, $\mathbf{P}_{ij}(k \mid k) \to \mathbf{0}$. Therefore, in the limit the mean and covariance of the projected estimate is

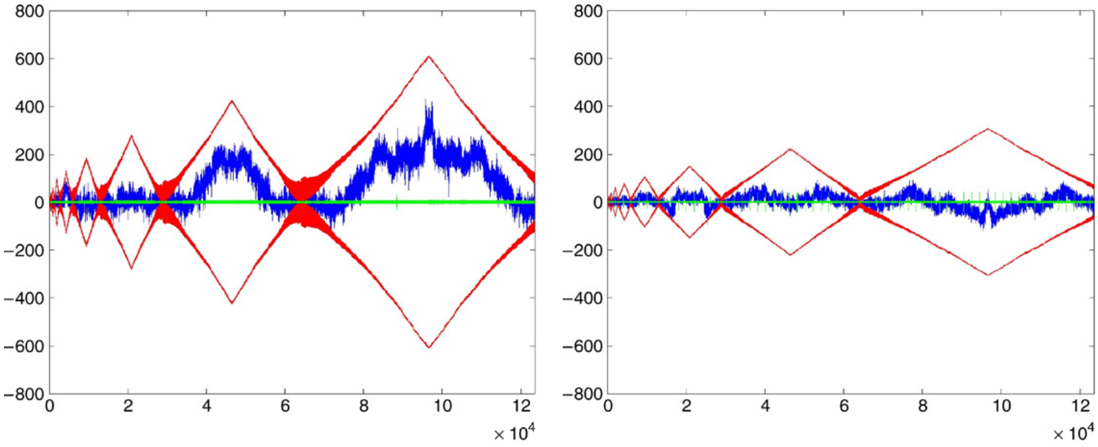
$$\mathbf{z}_j(k) = \hat{\mathbf{p}}_i(k \mid k) \oplus \hat{\mathbf{p}}_{ij}(k \mid k)$$

$$\tilde{\mathbf{R}}_j(k) = \mathbf{P}_i(k \mid k).$$

In other words, this is equivalent to the application of CI to fuse the estimate of landmark i directly with landmark j. This will occur for each landmark, connected via a relative estimate, throughout the map.

6.3. Results

The hybrid algorithm from the previous section was tested on a scenario that consists of approximately 1.1 million landmarks lying in a square of dimensions 40,000 m × 40,000 m. A subset of the overall scenario together with part of the path taken by the platform is shown in Fig. 14. The vehicle explores the environment in a set of squares of increasing size. To traverse the first square, the platform heads north (up) until it has moved about 200 m from the origin. It then turns to the left through 180° and heads down to the bottom of the region. At the bottom it turns through 180° to the left again, and continues to move out until it is 200 m to the left of the start position. At



(a) Variance in vehicle x.

(b) Variance in vehicle y.

Fig. 15. The standard deviation and state errors for the vehicle in the large scenario.

that point, the platform executes turns to the right until it has returned to the origin. The pattern is then repeated with area of the square doubling each time. The maximum sensor detection range is 50 m and an average of 4 landmarks are visible.

The standard deviation in the error in the vehicle is shown in Fig. 15. The characteristic wave pattern reflects the motion of the vehicle: The vehicle starts at the defined origin with its minimum possible covariance, traverses radially away from the origin with an accumulation of process noise, and progressively re-acquires position information as it reobserves previously mapped landmarks during its return to the origin. The variance peaks when the vehicle is furthest from the origin (the top left hand corner of Fig. 14) and is minimum when it is in proximity to the origin. There are several important points to note with this example. The fact that the estimate remains consistent shows that this system does not suffer from subtle correlation errors, which can require long periods of time to accumulate and ultimately undermine the integrity of a filter. Second, even though the variance of the vehicle in absolute terms at the far corner of the map is fairly large ($\sigma = 100$ m), it is extremely small compared to the distance travelled. Specifically, the rate of error growth is significantly less than 1% of the distance travelled, and the average error accumulated from each landmark is approximately 10^{-4} m. Third, although the SCI map alone may not be sufficiently accurate to perform operations such as gating, such operations can be carried out within the relative or local maps. This illustrates the advantage of the hybrid approach — information in global coordinates can be continuously obtained while other maps can be used, where necessary, to perform other operations.

6.4. Summary

This section has described how a CI and SCI can be used as the building blocks to develop a hybrid, large-scale SLAM solution. We have shown how these methods can be used to develop solutions to large scale mapping problems.

7. Summary

In this tutorial, we have considered a number of methods for applying the CI algorithm to SLAM. As we explained in the introduction, this algorithm was motivated by the fundamental limitations in FC-SLAM: both the computational costs and linearization errors lead to inconsistent estimates because of the need to precisely maintain the cross correlations between the estimates. CI, however, does not rely on these assumptions. As a result, it can be used to derive constant-time linear storage algorithms that are robust to linearization errors. We have shown how the CI algorithm can be combined with other SLAM algorithms to develop hybrid solutions, and we have shown how these methods can scale to extremely large maps of a practically significant size.

The most important conclusion to be drawn from this tutorial is that the full covariance Kalman filter solution to the SLAM problem cannot be applied rigorously in practice. Unlike the more common tracking and control applications of the Kalman filter, for which it is possible to compensate for many types of errors to maintain filter consistency, the structure of the SLAM problem effectively guarantees eventual filter inconsistency. Therefore, the full covariance Kalman solution should be regarded as being optimal in theory only. We argue in this tutorial that the Split Covariance Intersection framework provides the best performance possible when filter consistency must be provably guaranteed.

Appendix A. System model for the example

In this appendix we describe the system models for the extended example used throughout the paper.

A.1. Process model

The vehicle state consists of the position and orientation of the vehicle,

$$\mathbf{x}_v(k) = \begin{bmatrix} x_v(k) \\ y_v(k) \\ \theta_v(k) \end{bmatrix}.$$

The estimate of the ith landmark is its position in 2D,

$$\mathbf{p}_i(k) = \begin{bmatrix} x_i(k) \\ y_i(k) \end{bmatrix}.$$

The vehicle motion is modeled as the movement of a standard bicycle model [36]:

$$\mathbf{x}_v(k+1) = \begin{bmatrix} x_v(k) + V(k)\Delta t \cos(\delta(k) + \theta_v(k)) \\ y_v(k) + V(k)\Delta t \sin(\delta(k) + \theta_v(k)) \\ \theta_v(k) + \dfrac{V(k)\Delta t \sin(\delta(k))}{B} \end{bmatrix},$$

where the timestep is Δt, the control inputs are the wheel speed $V(k)$ and steer angle $\delta(k)$ and the vehicle wheel base is B. The process noises are additive disturbances which act on $V(k)$ and $\delta(k)$. The standard deviations on these terms are 0.1 ms^{-1} and 0.5°.

The initial covariance is diagonal with $\sigma_x = 1$, $\sigma_y = 1$, $\sigma_\theta = 4°$.

A.2. Linear observation model

In Section 2 the vehicle uses a linear $(\Delta x, \Delta y)$ position sensor. Therefore,

$$\mathbf{h}_i\left[\mathbf{x}_v(k), \mathbf{p}_i(k), \mathbf{w}(k)\right] = \begin{bmatrix} \Delta x_i(k) \\ \Delta y_i(k) \end{bmatrix} = \begin{bmatrix} x_i(k) - x_v(k) + w_x(k) \\ y_i(k) - y_v(k) + w_y(k) \end{bmatrix}.$$

A new landmark is initialized from

$$\mathbf{g}_i\left[\mathbf{x}_v(k), \mathbf{z}_i(k), \mathbf{w}(k)\right] = \begin{bmatrix} \Delta x_i(k) + x_v(k) \\ \Delta y_i(k) + y_v(k) \end{bmatrix}.$$

The observation covariance is

$$\mathbf{R}(k) = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}.$$

A.3. Range-bearing observation model

In the remaining sections, a nonlinear range-bearing sensor is used. The observation model is:

$$\mathbf{h}_i\left[\mathbf{x}_v(k), \mathbf{p}_i(k), \mathbf{w}(k)\right] = \begin{bmatrix} r(k) \\ \phi(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_v)^2 + (y_i - y_v)^2} + w_r(k) \\ \tan^{-1}\left(\dfrac{y_i - y_v}{x_i - x_v}\right) - \theta_v + w_\phi(k) \end{bmatrix}.$$

A new landmark is initialized from

$$\mathbf{g}_i\left[\mathbf{x}_v(k), \mathbf{z}_i(k), \mathbf{w}(k)\right] = \begin{bmatrix} x_v(k) + r(k)\cos(\theta_v(k) + \phi(k)) \\ y_v(k) + r(k)\sin(\theta_v(k) + \phi(k)) \end{bmatrix}.$$

For the stationary example, $\sigma_r = 0.16$ m and $\sigma_\theta = 4°$ (to exaggerate the effects of linearization errors). For the moving example, $\sigma_r = 0.04$ m and $\sigma_\theta = 0.5°$. The maximum detection range is 10 m — i.e., if $\sqrt{(x_i - x_v)^2 + (y_i - y_v)^2} \leq 10$, the sensor returns an observation.

A.4. Compass observation model

The compass measures the orientation of the vehicle directly. Therefore,

$$\mathbf{h}_\theta\left[\mathbf{x}_v(k), \mathbf{w}(k)\right] = \left[\theta_v(k) + w_\theta(k)\right]$$

where $w_\theta(k)$ is assumed to be additive with standard deviation $\sigma_{\theta\theta} = 2°$.

Appendix B. Covariance adjustment rules for split covariance intersection

This appendix describes a set of rules that can be used with SCI to maintain the known independent terms in the vehicle and landmark estimates. These rules were designed with two objectives in mind. First, to ensure that the rules can execute in constant time, they only look at the current state of the vehicle and landmark estimate immediately after the update.[9] Second, to ensure that (16) is obeyed, the rules should calculate a *lower bound* on the independent terms.

Assume that, at time step k, both the vehicle and landmark have been updated using the SCI algorithm. The value of ω for the vehicle update is ω_v and for the landmark it is ω_p. Four cases can be identified:

- $\omega_v = 1$ *and* $\omega_p = 1$. Neither the vehicle nor the landmark have been updated with the observation. This could arise if, for example, the observation were extremely noisy. As a result, the correlated and independent terms remain the same and

$$\mathbf{P}^I_{vv}(k \mid k) = \bar{\mathbf{P}}^I_{vv}(k \mid k),$$
$$\mathbf{P}^C_{vv}(k \mid k) = \bar{\mathbf{P}}^C_{vv}(k \mid k),$$
$$\mathbf{P}^I_{ii}(k \mid k) = \bar{\mathbf{P}}^I_{ii}(k \mid k),$$
$$\mathbf{P}^C_{ii}(k \mid k) = \bar{\mathbf{P}}^C_{ii}(k \mid k).$$

- $\omega_v = 1$ *and* $\omega_p < 1$. The landmark estimate is updated but the vehicle estimate is not. Because the vehicle can update with the same landmark again at any time in the future, the independent terms in the landmark are reassigned to its correlated terms. Furthermore, the independent contribution from the landmark to the vehicle estimate must be reassigned as a correlated term in the landmark estimate:

$$\mathbf{P}^I_{vv}(k \mid k) = \bar{\mathbf{P}}^I_{vv}(k \mid k) - \mathbf{W}^*_v(k)\mathbf{R}^I_v(k)\mathbf{W}^{*\mathrm{T}}_v(k)$$
$$\mathbf{P}^C_{vv}(k \mid k) = \bar{\mathbf{P}}^C_{vv}(k \mid k) + \mathbf{W}^*_v(k)\mathbf{R}^I_v(k)\mathbf{W}^{*\mathrm{T}}_v(k)$$
$$\mathbf{P}^I_{ii}(k \mid k) = \mathbf{0},$$
$$\mathbf{P}^C_{ii}(k \mid k) = \bar{\mathbf{P}}^C_{ii}(k \mid k) + \bar{\mathbf{P}}^I_{ii}(k \mid k).$$

- $\omega_p = 1$ *and* $\omega_v < 1$. the vehicle updates but the landmark does not. In this case, all of the vehicle independent terms become correlated:

[9] A fixed length window could also be used as well but we did not examine this case for simplicity.

$$\mathbf{P}^{I}_{vv}(k \mid k) = \mathbf{0}$$
$$\mathbf{P}^{I}_{vv}(k \mid k) = \bar{\mathbf{P}}^{I}_{vv}(k \mid k) + \bar{\mathbf{P}}^{C}_{vv}(k \mid k)$$
$$\mathbf{P}^{C}_{ii}(k \mid k) = \bar{\mathbf{P}}^{C}_{ii}(k \mid k) + \mathbf{W}_{*}(k)\mathbf{R}^{I}_{p}(k)\mathbf{W}^{*\mathrm{T}}_{p}(k)$$
$$\mathbf{P}^{I}_{ii}(k \mid k) = \bar{\mathbf{P}}^{I}_{ii}(k \mid k) - \mathbf{W}^{*}_{p}(k)\mathbf{R}^{I}_{i}(k)\mathbf{W}^{*\mathrm{T}}_{p}(k).$$

- $\omega_p < 1$ *and* $\omega_v < 1$. Both the vehicle and the landmark are updated. In this situation, all independent information is lost and so:

$$\mathbf{P}^{I}_{vv}(k \mid k) = \mathbf{0},$$
$$\mathbf{P}^{I}_{vv}(k \mid k) = \bar{\mathbf{P}}^{I}_{vv}(k \mid k) + \bar{\mathbf{P}}^{C}_{vv}(k \mid k),$$
$$\mathbf{P}^{I}_{ii}(k \mid k) = \mathbf{0},$$
$$\mathbf{P}^{C}_{ii}(k \mid k) = \bar{\mathbf{P}}^{C}_{ii}(k \mid k) + \bar{\mathbf{P}}^{I}_{ii}(k \mid k).$$

References

[1] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, in: Autonomous Robot Vehicles, Springer-Verlag, 1989.

[2] P. Moutarlier, R. Chatila, Stochastic multisensory data fusion for mobile robot location and environment modelling, LAAS Technical Note.

[3] C.G. Harris, Tracking With Rigid Models, in: A. Blake, A. Yuille (Eds.), Active Vision, MIT Press, Cambridge, MA, USA, 1993, pp. 59–73.

[4] Z. Zhang, O. Faugeras, 3D Dynamic Scene Analysis, Springer-Verlag, 1992.

[5] Y. Bar-Shalom, T.E. Fortmann, Tracking and Data Association, Academic Press, New York, NY, USA, 1988.

[6] G. Welch, G. Bishop, SCAAT: Incremental tracking with incomplete information, in: Proceedings of SIGGRAPH ’97, 1997, pp. 333–344.

[7] J.A. Castellanos, J.D. Tardós, J. Neira, Building a global map of the environment of a mobile robot: The importance of correlations, in: Proceedings of the 1997 Conference on Robotics and Automation, Albuquerque, USA, 1997, pp. 1053–1059.

[8] J. Castellanos, J. Montiel, J. Neira, J. Tardós, The SPmap: A probabilistic framework for simultaneous localization and map building, IEEE Transactions on Robotics and Automation 15 (5) (1996) 948–953.

[9] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, H.F. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters, The International Journal of Robotics Research 23 (7–8) (2004) 693–716.

[10] R. Eustice, M. Walter, J. Leonard, Sparse extended information filters: Insights into sparsification, in: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Edmonton, Alberta, Canada, 2005, pp. 641–648.

[11] A. Davison, Mobile robot navigation using active vision, Ph. D. Thesis, University of Oxford, 1999.

[12] J. Knight, A. Davison, I. Reid, Towards constant time SLAM using postponement, in: Proc. IEEE International Conference on Intelligent Robots and Systems, Maui, 2001, pp. 406–412.

[13] G. Dissanayake, H.F. Durrant-Whyte, T. Bailey, A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem, in: Proc IEEE International Conference on Robotics and Automation, Albuquerque, USA, 2000, pp. 1009–1014.

[14] S.J. Julier, J.K. Uhlmann, A counter example to the theory of simultaneous localization and map building, in: Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 4238–4243.

[15] J.A. Castellanos, J. Neira, J.D. Tardòs, Limits to the consistency of EKF-based SLAM, in: Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles, IAV’04, Lisbon, Portugal, 2004, pp. 1244–1249.

[16] T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot, Consistency of the EKF-SLAM algorithm, in: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2006.

[17] S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation, IEEE Review 92 (3).

[18] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, San Diego, CA, 1970.

[19] E. Foxlin, L. Naimark, VIS-Tracker: A wearable vision-inertial self-tracker, in: Proceedings of the 2003 IEEE VR Conference, Los Angeles, CA, USA, 2003, pp. 199–206.

[20] A.J. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Proc. International Conference on Computer Vision, Nice, France, 2003, pp. 1403–1410.

[21] J.K. Uhlmann, Dynamic map building and localization for autonomous vehicles, Ph. D. Thesis, University of Oxford, 1995.

[22] S.J. Julier, J.K. Uhlmann, General decentralized data fusion with covariance intersection (CI), in: D. Hall, J. Llinas (Eds.), Handbook of Data Fusion, CRC Press, Boca Raton FL, USA, 2001, pp. 12–1–12–25 (Chapter 12).

[23] S.J. Julier, J.K. Uhlmann, A non-divergent estimation algorithm in the presence of unknown correlations, in: Proceedings of the IEEE American Control Conference, vol. 4, Albuquerque, NM, USA, 1997, pp. 2369–2373.

[24] C. Schlegel, T. Kämpke, Filter design for simultaneous localization and map building (SLAM), in: Proc IEEE International Conference on Robotics and Automation, Washington, DC, 2002, pp. 2737–2742.

[25] D. Austin, P. Jensfelt, Using multiple Gaussian hypotheses to represent probability distributions for mobile robot localization, in: Proceedings of the 2000 IEEE Conference on Robotics and Automation, San Francisco, CA, USA, 2000, pp. 1036–1041.

[26] S. Reece, S. Roberts, Robust, low-bandwidth, multi-vehicle mapping, in: Proceedings of the 2005 FUSION Conference, Philadelphia, PA, USA, 2005, pp. 1319–1326.

[27] X. Xu, Vision-based ROV system, Ph. D. Thesis, University of Miami, May 2000.

[28] U.D. Hanebeck, K. Briechle, J. Horn, A tight bound for the joint covariance of two random vectors with unknown but constrainted cross-correlation, in: Proceedings of the 2005 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI’2001, Baden, Germany, 2001, pp. 85–90.

[29] S.J. Julier, H.F. Durrant-Whyte, A horizontal model fusion paradigm, in: The Proceedings of the SPIE AeroSense Conference: Navigation and Control Technologies for Unmannged Systems, vol. 2738, Orlando, FL, USA, 1996, pp. 37–48.

[30] S.J. Julier, Comprehensive process models for high-speed navigation, Ph. D. Thesis, University of Oxford, October 1997.

[31] S. Thrun, M. Montemerlo, The graph slam algorithm with applications to large-scale mapping of urban structures, The International Journal of Robotics Research 25 (5–6) (2006) 403–429.

[32] G. Dissayanake, S.B. Williams, H.F. Durrant-Whyte, Map management for efficient simultaneous and mapping (SLAM), Autonomous Robots 12 (3) (2002) 267–286.

[33] M. Csorba, Simultaneous localisation and map building, Ph.D. Thesis, University of Oxford, 1997.

[34] P. Newman, On the structure and solution of the simultaneous localisation and map building problem, Ph. D. Thesis, University of Sydney, 1999.

[35] W.H. Press, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in C, 2nd edition, Cambridge University Press, 1992, pp. 412–420 (Chapter 10).

[36] H.F. Durrant-Whyte, An autonomous guided vehicle for cargo handling applications, International Journal of Robotics Research 15 (5) (1996) 407–440.

[37] M. Bosse, P. Newman, J. Leonard, S. Teller, Simultaneous localization and map building in large cyclic environments using the Atlas framework, The International Journal of Robotics Research 3 (12) (2004) 1113–1139.

Dr. Simon Julier is a Senior Lecturer at the Vision, Imaging and Virtual Environments Group, in the Computer Science Department at UCL. Before joining UCL, Dr. Julier worked for nine years at the 3D Mixed and Virtual Environments Laboratory at the Naval Research Laboratory in Washington DC. There he was PI of the Battlefield Augmented Reality System (BARS), a research effort to develop man-wearable systems for providing situation awareness information. He served as the Associate Director of the 3DMVEL from 2005–2006. His research interests include user interfaces, distributed data fusion, nonlinear estimation, and simultaneous localisation and mapping.

Prof. Jeffrey Uhlmann has a doctorate in robotics from the University of Oxford, an MSc in Computer Science, and a BA in Philosophy from the University of Missouri-Columbia. He has received numerous publication, performance, and invention awards for his work in the areas of target tracking, multi-dimensional search, and quantum computing. His frequent collaborators include Simon Julier, Marco Lanzagorta, and gentleman-scholar Mil Mascaras.