TUTORIAL

Simultaneous Localization and Mapping (SLAM): Part II

BY TIM BAILEY AND HUGH DURRANT-WHYTE

Simultaneous localization and mapping (SLAM) is the process by which a mobile robot can build a map of the environment and, at the same time, use this map to compute its location. The past decade has seen rapid and exciting progress in solving the SLAM problem together with many compelling implementations of SLAM methods. The great majority of work has focused on improving computational efficiency while ensuring consistent and accurate estimates for the map and vehicle pose. However, there has also been much research on issues such as nonlinearity, data association, and landmark characterization, all of which are vital in achieving a practical and robust SLAM implementation.

This tutorial focuses on the recursive Bayesian formulation of the SLAM problem in which probability distributions or estimates of absolute or relative locations of landmarks and vehicle pose are obtained. Part I of this tutorial (*IEEE Robotics & Automation Magazine*, vol. 13, no. 2) surveyed the development of the essential SLAM algorithm in state-space and particle-filter form, described a number of key implementations, and cited locations of source code and real-world data for evaluation of SLAM algorithms. Part II of this tutorial (this article), surveys the current state of the art in SLAM research with a focus on three key areas: computational complexity, data association, and environment representation. Much of the mathematical notation and essential concepts used in this article are defined in Part I of this tutorial and, therefore, are not repeated here.

SLAM, in its naive form, scales quadratically with the number of landmarks in a map. For real-time implementation, this scaling is potentially a substantial limitation in the use of SLAM methods. The complexity section surveys the many approaches that have been developed to reduce this complexity. These include linear-time state augmentation, sparsification in information form, partitioned updates, and submapping methods. A second major hurdle to overcome in the implementation of SLAM methods is to correctly associate observations of landmarks with landmarks held in the map. Incorrect association can lead to catastrophic failure of the SLAM algorithm. Data association is particularly important when a vehicle returns to a previously mapped region after a long excursion, the so-called loop-closure problem. The data association section surveys current data association methods used in SLAM. These include batch-validation methods that exploit constraints inherent in the SLAM formulation, appearance-based methods, and multihypothesis techniques. The third development discussed in this tutorial is the trend towards richer appearance-based models of landmarks and maps. While initially motivated by problems in data association and loop closure, these methods have resulted in qualitatively different methods of describing the SLAM problem, focusing on trajectory estimation rather than landmark estimation. The environment representation section surveys current developments in this area along a number of lines, including delayed mapping, the use of nongeometric landmarks, and trajectory estimation methods.

SLAM methods have now reached a state of considerable maturity. Future challenges will center on methods enabling large-scale implementations in increasingly unstructured environments and especially in situations where GPS-like solutions are unavailable or unreliable: in urban canyons, under foliage, under water, or on remote planets.

Computational Complexity

The state-based formulation of the SLAM problem involves the estimation of a joint state composed of a robot pose and the locations of observed stationary landmarks. This problem formulation has a peculiar structure; the process model only affects vehicle pose states and the observation model only makes reference to a single vehicle-landmark pair. A wide range of techniques have been developed to exploit this special structure in limiting the computational complexity of the SLAM algorithm.

Techniques aimed at improving computational efficiency may be characterized as being optimal or conservative. Optimal algorithms aim to reduce required computation while still resulting in estimates and covariances that are equal to the full-form SLAM algorithm (as presented in Part I of this tutorial). Conservative algorithms result in estimates that have larger uncertainty or covariance than the optimal result. Usually, conservative algorithms, while less accurate, are computationally more efficient and, therefore, of value in real implementations. Algorithms with uncertainties or covariances less than those of the optimal solution are termed inconsistent and are considered invalid solutions to the SLAM (or any estimation) problem.

The direct approach to reducing computational complexity involves exploiting the structure of the SLAM problem in reformulating the essential time- and observation-update equations to limit required computation. The time-update computation can be limited using *state-augmentation* methods. The observation-update computation can be limited using a *partitioned form* of the update equations. Both these steps result in an optimal SLAM estimate with reduced computation. Reformulation of the standard space-space SLAM representation into information form allows *sparsification* of the resulting information matrix to be exploited in reducing computation. The resulting algorithms are usually conservative but still yield good estimates with much reduced computational effort. *Submapping* methods exploit the idea that a map can be broken up into regions with local coordinate systems and arranged in a hierarchical manner. Updates can occur in a local frame with periodic interframe updates. Submapping techniques generally provide a conservative estimate in the global frame.

State Augmentation

At a time k, the joint SLAM state vector $\mathbf{x}_k = [\mathbf{x}_{v_k}^T, \mathbf{m}^T]^T$ comprises two parts: the robot pose $\mathbf{x}_{v_k}$ and the set of map landmark locations $\mathbf{m}$. The vehicle model propagates only the pose states according to a set of control inputs $\mathbf{u}_k$ while leaving the map states unchanged

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)\\ &= \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k) \\ m \end{bmatrix},\end{aligned} \tag{1}$$

In a naive implementation of the extended Kalman filter (EKF) for SLAM, the covariance prediction is computed from

$$\mathbf{P}_{k|k-1} = \nabla\mathbf{f}_\mathbf{x}\mathbf{P}_{k-1|k-1}\nabla\mathbf{f}_\mathbf{x}^T + \nabla\mathbf{f}_\mathbf{u}\mathbf{U}k\nabla\mathbf{f}_\mathbf{u}^T, \tag{2}$$

where $\nabla\mathbf{f}_\mathbf{x} = \frac{\partial\mathbf{f}}{\partial\mathbf{x}_{k-1}}$, $\nabla\mathbf{f}_\mathbf{u} = \frac{\partial\mathbf{f}}{\partial\mathbf{u}_k}$ and $\mathbf{U}_k$ is a covariance characterising uncertainty on the control vector. This operation has cubic complexity in the number of landmarks due to matrix multiplication of the Jacobian $\nabla\mathbf{f}_\mathbf{x}$ and the covariance matrix $\mathbf{P}_{k-1|k-1}$. However, as only the pose states are affected by the vehicle model, the covariance prediction can be rewritten in a form which has linear complexity in the number of landmarks [53, Sec. 2.4.1]

$$\mathbf{P}_{k|k-1} = \begin{bmatrix} \nabla\mathbf{f}_{i_\mathbf{x}}\mathbf{P}_{vv}\nabla\mathbf{f}_{i_\mathbf{x}}^T + \nabla\mathbf{f}_{i_\mathbf{u}}\mathbf{U}_k\nabla\mathbf{f}_{i_\mathbf{u}}^T & \nabla\mathbf{f}_{i_\mathbf{x}}\mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T\nabla\mathbf{f}_{i_\mathbf{x}}^T & \mathbf{P}_{mm} \end{bmatrix}, \tag{3}$$

where $\nabla\mathbf{f}_{i_\mathbf{x}} = \frac{\partial\mathbf{f}_i}{\partial\mathbf{x}_{v_{k-1}}}$, $\nabla\mathbf{f}_{i_\mathbf{u}} = \frac{\partial\mathbf{f}_i}{\partial\mathbf{u}_k}$ and where

$$\mathbf{P}_{k-1|k-1} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}.$$

The process of adding a new landmark to the state vector has a similar form. A new map landmark is initializd as a function of the robot pose and an observation $\mathbf{z}_k$

$$\mathbf{m}_{new} = \mathbf{g}(\mathbf{x}_{vk}, \mathbf{z}_k). \tag{4}$$

The augmented states are then a function of only a small number of existing states

$$\mathbf{x}_k^+ = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{m} \\ \mathbf{g}(\mathbf{x}_{v_k}, \mathbf{z}_k) \end{bmatrix}. \tag{5}$$

The general idea of state augmentation can be applied whenever new states are a function of a subset of existing states

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{f}(\mathbf{x}_2, \mathbf{q}) \end{bmatrix}, \tag{6}$$

$$\begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{12}\nabla\mathbf{f}_{\mathbf{x}_2}^T \\ \mathbf{P}_{12}^T & \mathbf{P}_{22} & \mathbf{P}_{22}\nabla\mathbf{f}_{\mathbf{x}_2}^T \\ \nabla\mathbf{f}_{\mathbf{x}_2}\mathbf{P}_{12}^T & \nabla\mathbf{f}_{\mathbf{x}_2}\mathbf{P}_{22} & \nabla\mathbf{f}_{\mathbf{x}_2}\mathbf{P}_{22}\nabla\mathbf{f}_{\mathbf{x}_2}^T + \nabla\mathbf{f}_\mathbf{q}\mathbf{Q}\nabla\mathbf{f}_\mathbf{q}^T \end{bmatrix}. \tag{7}$$

A comparison of (1) and (3) with (6) and (7) shows that the SLAM prediction step is a special case of state augmentation in which the state is augmented by the new pose $\mathbf{x}_{vk}$ and where the previous pose $\mathbf{x}_{vk-1}$ is removed by marginalization. In this form, both the EKF prediction step and the process of adding new landmarks can be reduced to calculations that are linear in the number of landmarks. The predictions made are clearly optimal.

Partitioned Updates

A naive implementation of the SLAM observation-update step updates all vehicle and map states every time a new measurement is made. For an EKF update, the computational effort scales quadratically with the number of landmarks held in the map. A number of partitioned update methods have been devised to reduce this computational effort. These confine sensor-rate updates to a small local region and update the global map only at a much lower frequency. These partition methods all produce optimal estimates.

There are two basic types of partitioned update. The first operates in a local region of the global map and maintains globally referenced coordinates. This approach is taken by the compressed EKF (CEKF) [21] and the postponement algorithm [28]. The second generates a short-term submap with its own local coordinate frame. This is the approach of the constrained local submap filter (CLSF) [53] and the local map sequencing algorithm [45]. We focus on this latter approach as it is simpler and, by performing high-frequency operations in

IEEE Proof

a local coordinate frame, it avoids very large global covariances and, therefore, is more numerically stable and less affected by linearization errors.

The local submap algorithm maintains at all times two independent SLAM estimates

$$\mathbf{x}_G = \begin{bmatrix} \mathbf{x}_F^G \\ \mathbf{m}_G \end{bmatrix}, \quad \mathbf{x}_R = \begin{bmatrix} \mathbf{x}_v^R \\ \mathbf{m}_R \end{bmatrix}, \tag{8}$$

where $\mathbf{x}_G$ is a map composed of a set of globally referenced landmarks $\mathbf{m}_G$, together with the global reference pose of a submap coordinate frame $\mathbf{x}_F^G$, and where $\mathbf{x}_R$ is the local submap with a locally referenced vehicle pose $\mathbf{x}_v^R$ and locally referenced landmarks $\mathbf{m}_R$ as shown in Figure 1(a) and 1(b), respectively.

As observations are made, conventional SLAM updates are performed entirely within the local submap and with only those landmarks held in the local submap. It is possible to obtain a global vehicle pose estimate at any time by simple vector summation of the locally referenced pose and the global estimate of the submap coordinate frame. An optimal global estimate is obtained periodically by registering the submap with the global map, see Figure 1(c), and applying constraint updates upon any features common to both maps. At this point a new submap is created and the process continues.

The submap method has a number of advantages. First, the number of landmarks that must be updated at any one time is limited to only those that are described in the local submap coordinate frame. Thus, the observation-rate update is independent of the total map size. The full update, and the propagation of local estimates, can be carried out as a background task at a much lower update rate while still permitting observation-rate global localzation. A second advantage is that there is lower uncertainty in a locally referenced frame, so approximations due to linearization are reduced. Finally, submap registration can use batch-validation gating, thereby improving association robustness.

Sparsification

Conventional EKF-SLAM produces a state estimate $\hat{\mathbf{x}}_k$ and covariance matrix $\mathbf{P}_k$, which implicitly describe the first two central moments of a Gaussian probability density on the true state $\mathbf{x}_k$. An alternative representation for this same Gaussian is in canonical or information form using the information vector $\hat{\mathbf{y}}_k$ and information matrix $\mathbf{Y}_k$. These are related to the moment form parameters as

$$\mathbf{Y}_k = \mathbf{P}_k^{-1}, \tag{9}$$

$$\hat{\mathbf{y}}_k = \mathbf{Y}_k \hat{\mathbf{x}}_k. \tag{10}$$

The advantage of the information form for SLAM is that, for large-scale maps, many of the off-diagonal components of the *normalized* information matrix are very close to zero. Thrun et al. [47], [48] have exploited this observation to propose a sparsification procedure that allows near-zero elements of the normalized information matrix to be set to zero. With the information matrix now sparse, very efficient update procedures for information estimates can be obtained with relatively little loss in optimality of the maps produced. Although this initial solution was subsequently shown not to be consistent by Eustice et al. [17], the idea of sparsification has sparked considerable interest in the information-form SLAM problem and several consistent sparse solutions [12], [18], [19], [42], [50], [52] have been presented. Of particular note are those solutions that are optimal and exactly sparse [12], [18], [19].

The key to exact sparsification of the information form of the SLAM problem is to notice that state augmentation is a sparse operation. Consider the moment-form augmentation identity in (6) and (7). These have an equivalent information-form identity

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 - \nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{Q}^{-1} [\mathbf{f}(\mathbf{x}_2) - \nabla \mathbf{f}_{\mathbf{x}_2} \mathbf{x}_2] \\ \mathbf{Q}^{-1} [\mathbf{f}(\mathbf{x}_2) - \nabla \mathbf{f}_{\mathbf{x}_2} \mathbf{x}_2] \end{bmatrix}, \tag{11}$$

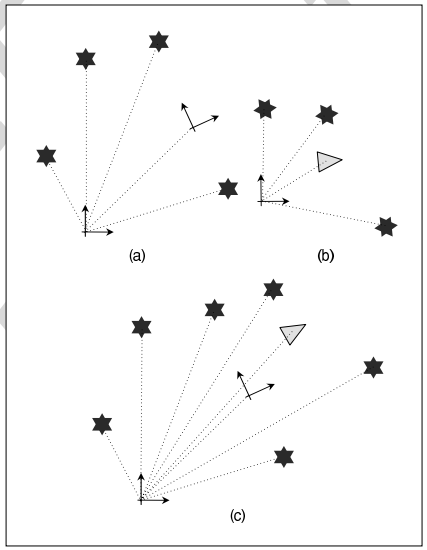

Figure 1. *The constrained local submap filter. The SLAM frontier is constructed in (b) a local map, which periodically registers with (a) a global map to produce (c) an optimal global estimate.*

$$\begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & 0 \\ \mathbf{Y}_{12}^T & \mathbf{Y}_{22} + \nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{Q}^{-1} \nabla \mathbf{f}_{\mathbf{x}_2} & -\nabla \mathbf{f}_{\mathbf{x}_2}^T \mathbf{Q}^{-1} \\ 0 & -\mathbf{Q}^{-1} \nabla \mathbf{f}_{\mathbf{x}_2} & \mathbf{Q}^{-1}, \end{bmatrix}. \tag{12}$$

where, for simplicity, it is assumed that noise is zero-mean additive $\mathbf{f}(\mathbf{x}_2, \mathbf{q}) = \mathbf{f}(\mathbf{x}_2) + \mathbf{q}$. Assuming the subset of states $\mathbf{x}_1$ comprises the bulk of the map states, then (12) is sparse and has constant-time complexity compared to (7), which has linear complexity in the dimension of $\mathbf{x}_1$.

Therefore, in the information-form SLAM problem, an exactly sparse solution can be obtained by augmenting the state with the new vehicle pose estimate at each time step and retaining all past robot poses,

$$\mathbf{x}_k = \left[\mathbf{x}_{v_k}^T, \mathbf{x}_{v_{k-1}}^T, \ldots, \mathbf{x}_{v_1}^T, \mathbf{m}^T\right]^T. \tag{13}$$

In this way, the off-diagonal terms of the information matrix are nonzero only for poses and landmarks that are directly related by measurement data [see Figure 2(a)]. Observation updates are also a sparse operation, producing links only between measured states.

However, marginalization, which is necessary to remove past pose states, introduces links between all state elements connected to the removed states. Marginalizing all past states produces a dense information matrix as shown in Figure 2(c). Nevertheless, it is possible to retain a reasonably sparse estimate without having to keep an entire pose history [19]. By judicious selection of anchoring poses to decouple different regions of the map, a great proportion of poses can be marginalized away without inducing excessive density as shown in Figure 2(b).

Despite the attraction of its sparse representation, there remain serious caveats with regard to practical implementation of information-form SLAM. For realistic use, it is necessary to recover the mean and covariance of the state at every time step. This is potentially very expensive. The mean estimate is required to perform linearization of the process and observation models. It can be recovered fairly efficiently using the conjugate gradients method [16]. The mean and covariance are both required to compute validation gates for data association. While efficient solutions have been devised for simple gating [16], [48], the robust batch gating methods, described in the following data association section, potentially involve recovery of the full covariance matrix, which has a cubic complexity in the number of landmarks.

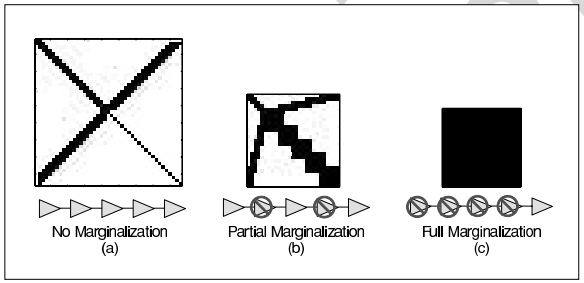

Figure 2. *Exact information matrix SLAM: These information matrices all represent optimal map estimates but show the tradeoff between the number of retained pose states and matrix sparsity; (b) keeps just four out of 20 of past poses (i.e., 20%) while remaining quite sparse.*

Global Submaps

Submap methods are another means of addressing the issue of computation scaling quadratically with the number of landmarks during measurement updates. Submap methods come in two fundamental varieties: globally referenced and locally referenced, as shown in Figure 3. The common thread to both types is that the submap defines a local coordinate frame and nearby landmarks are estimated with respect to the local frame. The local submap estimates are obtained using the standard, optimal SLAM algorithm using only the locally referenced landmarks. The resulting submap structures are then arranged in a hierarchy leading to computational efficiency but also lack of optimality.

Global submap methods estimate the global locations of submap coordinate frames relative to a common base frame. This is the approach adopted in the relative landmark representation (RLR) [22], hierarchical SLAM [15], and constant time SLAM (CTS) [30] methods. These approaches reduce computation from a quadratic dependence on the number of landmarks to a linear or constant time dependence by maintaining a conservative estimate of the global map. However, as submap frames are located relative to a common base coordinate frame, global submaps do not alleviate linearization issues arising from large pose uncertainties.

Relative Submaps

Relative submap methods differ from global submaps in that there is no common coordinate frame. The location of any given submap is recorded only by its neighboring submaps, and these are connected in a graphical network. Global estimates can be obtained by vector summation along a path in the network. By eschewing any form of global-level data fusion, relative submaps address both computation and nonlinearity issues.

The original notion of relative submaps was introduced by Chong and Kleeman [8]. This was further developed by Williams [53] in the form of the constrained relative submap filter (CRSF). However,

IEEE Proof

Future challenges will center on methods enabling large-scale implementations in increasingly unstructured environments and especially in situations where GPS-like solutions are unavailable or unreliable

CRSF does not exhibit global-level convergence without forfeiting the decoupled submap structure. The Atlas framework [6], [7] and network coupled feature maps (NCFM) [2] rectified this problem by realizing that conservative global convergence could be achieved using the covariance intersect algorithm [26] for estimating connections. These algorithms result in a network of optimal SLAM submaps connected by conservative links.

The relative submap framework has a number of advantages. In particular, it produces locally optimal maps with computational complexity independent of the size of the compete map. Further, by treating updates locally, it is numerically very stable, allows batch association between frames, and minimizes problems arising from linearization in a global frame.

Data Association

Data association has always been a critical issue for practical SLAM implementations. Before fusing data into the map, new measurements are associated with existing map landmarks, and, after fusion, these associations cannot be revised. The problem is that a single incorrect data association can induce divergence into the map estimate, often causing catastrophic failure of the localization algorithm. SLAM algorithms will be fragile when 100% correct associations are mandated for correct operation.

Batch Validation

Almost all SLAM implementations perform data association using only statistical validation gating, a method inherited from the target-tracking literature for culling unlikely associations [4]. Early SLAM implementations considered each measurement-to-landmark association individually by testing whether an observed landmark is close to a predicted location. Individual gating is extremely unreliable if the vehicle pose is very uncertain and fails in all but the most sparsely populated and structured environments.

An important advance was the concept of batch gating, where multiple associations are considered simultaneously. Mutual association compatibility exploits the geometric relationship between landmarks. The two existing forms of batch gating are the joint compatibility branch and bound (JCBB) [37] method, which is a tree-search, and combined constraint data association (CCDA) [2], which is a graph search (see Figure 4). The latter (and also a randomized variant of JCBB [38]) is able to perform reliable association with no knowledge of vehicle pose whatsoever.

Batch gating alone is often sufficient to achieve reliable data association: If the gate is sufficiently constrained, association errors have an insignificant effect [5], and if a false association is made with an incorrect landmark that is physically close to the right one, then the inconsistency is minor. This may not always be valid and, especially in large complex environments, more comprehensive data association mechanisms (such as multihypothesis tracking [4]) may be necessary.

Appearance Signatures

Gating on geometric patterns alone is not the only avenue for reliable data association. Many sensing modalities, such as vision, provide rich information about shape, color, and texture, all of which may be used to find a correspondence between two data sets. For SLAM, appearance signatures are useful to predict a possible association, such as closing a loop, or for assisting conventional gating by providing additional discrimination information.

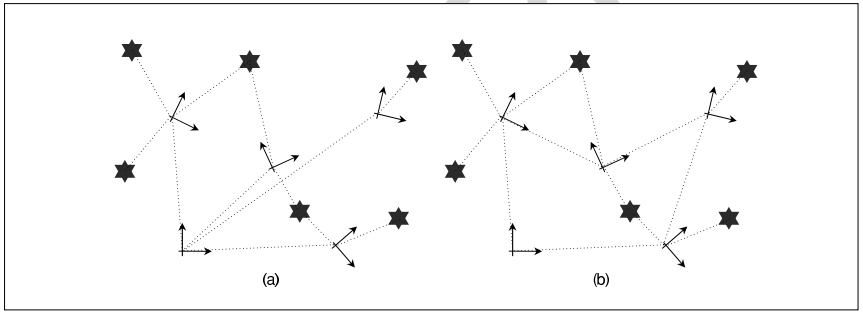

Figure 3. *Global and relative submaps.*

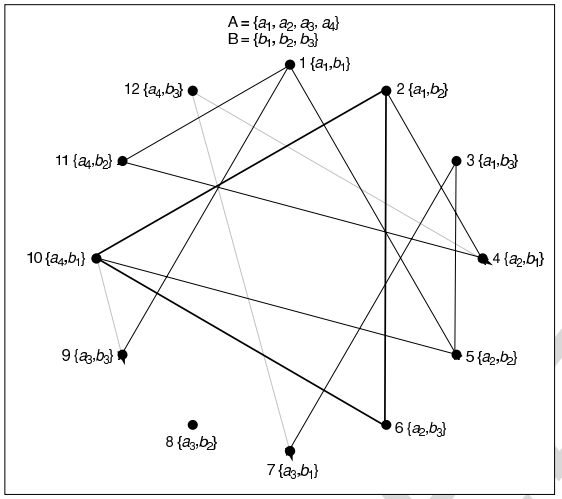

Figure 4. *Combined constraint data association (CCDA) performs batch-validation gating by constructing and searching a correspondence graph. The graph nodes represent associations that are possible when considered individually. The edges indicate compatible associations, and a clique is a set of mutually compatible associations (e.g., the clique 2, 6, 10 implies that associations* $a_1 \rightarrow b_2, a_2 \rightarrow b_3, a_4 \rightarrow b_1$ *may coexist).*

Historically, appearance signatures and image similarity metrics have been developed for indexing image databases [43] and for recognizing places in topological mapping [1], [49]. More recently, appearance measures have been applied to detecting loops in SLAM [23], [39]. The work on visual appearance signatures for loop detection by Newman et al. [39] introduces two significant innovations. A similarity metric over a sequence of images, rather than a single image, is computed, and an eigenvalue technique is employed to remove common-mode similarity. This approach considerably reduces the occurrence of false positives by considering only matches that are interesting or uncommon.

Multihypothesis Data Association

Multihypothesis data association is essential for robust target tracking in cluttered environments [4]. It resolves association ambiguities by generating a separate track estimate for each association hypothesis, creating over time an ever-branching tree of tracks. The number of tracks is typically limited by the available computational resources, and low-likelihood tracks are pruned from the hypothesis tree.

Multihypothesis tracking (MHT) is also important for robust SLAM implementation, particularly in large complex environments. For example, in loop closure, a robot should ideally maintain separate hypotheses for suspected loops and also a "no-loop" hypothesis for cases where the perceived environment is structurally similar. While MHT has been applied to mapping problems [9], this has yet to be applied in the SLAM context. A major hurdle is the computational overhead of maintaining separate map estimates for each hypothesis. Tractable solutions may be possible using sparsification or submap methods. The FastSLAM algorithm is inherently a multihypothesis solution, with each particle having its own map estimate. A significant attribute of the FastSLAM algorithm is its ability to perform per-particle data association [36].

Environment Representation

Early work in SLAM assumed that the world could reasonably be modeled as a set of simple discrete landmarks described by geometric primitives such as points, lines, or circles. In more complex and unstructured environments—outdoor, underground, subsea—this assumption often does not hold.

Partial Observability and Delayed Mapping

Environment modeling depends both on the complexity of the environment and on the limitations of the sensing modality. Two common examples are sonar and vision. Sonar sensors typically produce accurate range measurements but often have large beam width and side lobes, making the bearing estimate unusable [31]. Measurements from a single camera, on the other hand, provide bearing information without an accurate indication of range.

SLAM with range-only sensors [32], [33] and bearing-only sensors [3], [11] shows that a single measurement is insufficient to constrain a landmark location. Rather, it must be observed from multiple vantage points as shown in Figure 5. More precisely, a single measurement generates a non-Gaussian distribution over landmark location, and multiple measurements are needed to obtain an estimate. Generalized distributions, such as mixture models, permit immediate, nondelayed landmark tracking [44]. One way to

IEEE PROOF

obtain a Gaussian landmark estimate is to delay initialization and, instead, accumulate raw measurement data. To permit consistent delayed fusion, it is necessary to record the vehicle pose for each deferred measurement. Thus, the SLAM state is augmented with recorded pose estimates

$$\mathbf{x}_k = \left[\mathbf{x}_{v_k}^T, \mathbf{x}_{v_{k-1}}^T, \ldots, \mathbf{x}_{v_{k-n}}^T, \mathbf{m}^T\right]^T, \tag{14}$$

and the corresponding measurements are stored in an auxiliary list $\{\mathbf{z}_k, \ldots, \mathbf{z}_{k-n}\}$. Once sufficient information over a period n has been collected, a landmark is initialized by a batch update. Recorded poses that do not have any other associated measurements are then simply removed from the state.

Delayed fusion addresses far more than just partial observability. It is a general concept for increasing robustness by accumulating information and permitting delayed decision making. Given an accumulated data set, an improved estimate can be obtained by performing a batch update, such as bundle adjustment [11] or iterated smoothing, which dramatically reduces linearization errors. Deferred data also facilitate batch validation gating and, therefore, aid reliable data association.

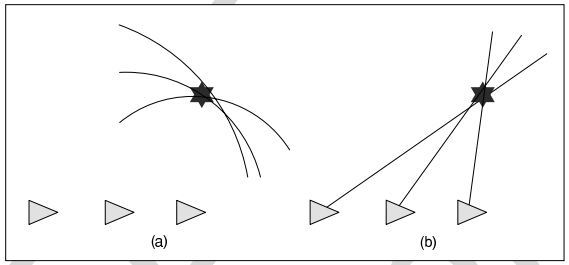

Figure 5. *Partial observation. Some sensing modalities cannot directly observe a landmark location and require observations from multiple vantage points.*

Nongeometric Landmarks

While EKF-SLAM is usually applied to geometric landmarks (often misnamed *point landmarks*), the simple expedient of attaching a coordinate frame to an arbitrary object allows the same methods to be applied to much more general landmark descriptions. A recent contribution by Nieto et al. [40] shows that landmarks of arbitrary shape may be dealt with by using EKF-SLAM to reconcile landmark locations separately from the estimation of shape parameters.

A landmark is described by a shape model which has an embedded coordinate frame defining the landmark origin as shown in Figure 6(a). This model is auxiliary to the SLAM process and may have any representation that permits data alignment (e.g., a grid). When the robot observes the landmark, the shape model is aligned with the measurement data as shown in Figure 6(b). Assuming this alignment is approximately Gaussian, the vehicle-centric estimate of the model coordinate frame is an observation suitable for an EKF-SLAM update, where the map is composed of landmark frame locations as in Figure 6(c).

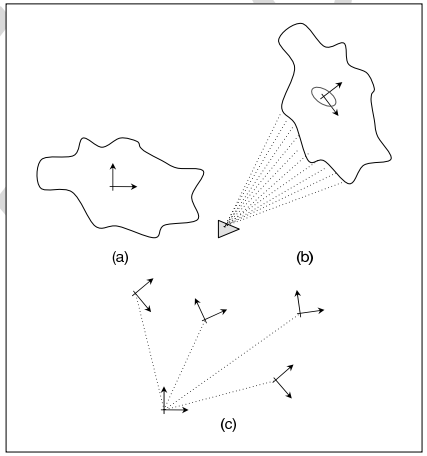

Figure 6. *SLAM with arbitrary shaped landmarks. Aligning a shape model with sensed data produces a suitable observation model for SLAM.*

3-D SLAM

Implementing SLAM in three dimensions is, in principle, a straightforward extension of the two-dimensional (2-D) case. However, it involves significant added complexity due to the more general vehicle motion model and, most importantly, greatly increased sensing and feature modeling complexity.

Three essential forms of three-dimensional (3-D) SLAM exist. The first is simply 2-D SLAM with additional map building capabilities in the third dimension, for example, horizontal laser-based SLAM with a second orthogonal laser mapping vertical slices [35], [46]. This approach is appropriate when the vehicle motion is confined to a plane. The second form is a direct extension of 2-D SLAM to three dimensions, with the extraction of discrete landmarks and joint estimation of the map and vehicle pose. This has been implemented with monocular vision sensing by Davison et al. [10] and permits full six degree-of-freedom motion (see also [27] for an airborne application). The third form involves an entirely different SLAM formulation, where the joint state is composed of a history of past vehicle poses [16], [39]. At each pose, the vehicle obtains a 3-D scan of the environment, and the pose estimates are aligned by correlating the scans.

Trajectory-Oriented SLAM

The standard SLAM formulation, as described in Part I of this tutorial, defines the estimated state as the vehicle pose and a list of observed landmarks

$$\mathbf{x}_k = \left[\mathbf{x}_{v_k}^T, \mathbf{m}^T\right]^T. \tag{15}$$

An alternative formulation of the SLAM problem that has gained recent popularity is to estimate the vehicle trajectory instead

$$\mathbf{x}_k = \left[\mathbf{x}_{v_k}^T, \mathbf{x}_{v_{k-1}}^T, \ldots, \mathbf{x}_{v_1}^T\right]^T. \tag{16}$$

This formulation is particularly suited to environments where discrete identifiable landmarks are not easily discerned and direct alignment of sensed data is simpler or more reliable. Notice that the map is no longer part of the state to be estimated but instead forms an auxiliary data set. Indeed, this formulation of the SLAM problem has no explicit map; rather, each pose estimate has an associated scan of sensed data, and these are aligned to form a global map. Figure 7 shows an example of this approach from [39].

The FastSLAM algorithm may also be considered an example of trajectory estimation, with each particle defining a particular trajectory hypothesis. Several recent FastSLAM hybrids use pose-aligned scans or grids in place of a landmark map [14], [20], [24]. Another variation of trajectory-based SLAM has developed from topological mapping [34], where poses are connected in a graphical network rather than a joint state vector. This framework, known as *consistent pose estimation* (CPE) [23], [29], is a promising alternative to state-space SLAM and is capable of producing large-scale maps. The advent of sparse-information-form SLAM has led to a third type of trajectory-based SLAM [12], [18], [39], with sparse estimation of (16).

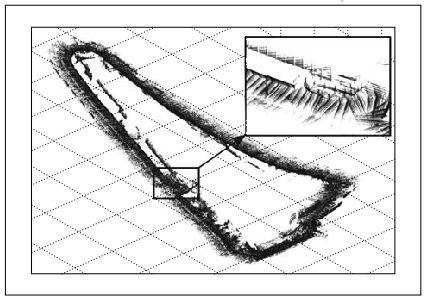

Figure 7. *Trajectory-based SLAM. Scans taken at each pose are aligned according to their pose estimate to form a global map. (Picture courtesy of [39].)*

While trajectory SLAM has many positive characteristics, these come with caveats. Most importantly, its state-space grows unbounded with time, as does the quantity of stored measurement data. For very long-term SLAM, it will eventually become necessary to coalesce data into a format similar to the traditional SLAM map to bound storage costs.

Embedded Auxiliary Information

Trajectory-based SLAM lends itself to representing spatially located information. Besides scan data for mapping, it is possible to associate auxiliary information with each pose, soil salinity, humidity, temperature, or terrain characteristics, for example. The associated information may be used to assist mapping, to aid data association, or for purposes unrelated to the mapping task, such as path planning or data gathering.

This concept of embedding auxiliary data is more difficult to incorporate within the traditional SLAM framework. The SLAM state is composed of discrete landmark locations and is ill suited to the task of representing dense spatial information. Nieto et al. [41] have devised a method called DenseSLAM to permit such an embedding. As the robot moves through the environment, auxiliary data is stored in a suitable data structure, such as an occupancy grid, and the region represented by each grid cell is determined by a set of local landmarks in the SLAM map. As the map evolves, and the landmarks move, the locality of the grid region is shifted and warped accordingly. The result is an ability to consistently maintain spatial locality of dense auxiliary information using the SLAM landmark estimates.

Dynamic Environments

Real-world environments are not static. They contain moving objects, such as people, and temporary structures that appear static for a while but are later moved, such as chairs and parked cars. In dynamic environments, a SLAM algorithm must somehow manage moving objects. It can detect and ignore them; it can track them as moving landmarks, but it must not add a moving object to the map and assume it is stationary.

The conventional SLAM solution is highly redundant. Landmarks can be removed from the map without loss of consistency, and it is often possible to remove large numbers of landmarks with little change in convergence rate [13]. This property has been exploited to maintain a contemporaneous map by removing landmarks that have become obsolete due to changes in the environment [2, Sec. 5.1]. To explicitly manage moving objects, Hähnel et al. [25] implement an auxiliary identification routine and then remove the dynamic information from a data scan before sending it to their SLAM algorithm. Conversely, Wang et al. [51] add moving objects to their estimated state and provide models for tracking both stationary and dynamic targets. Simultaneous

IEEE PROOF

estimation of moving and stationary landmarks is very costly due to the added predictive model. For this reason, the implemented solution first involves a stationary SLAM update followed by separate tracking of moving targets.

The past decade, in particular, has seen substantial progress in our understanding of the SLAM problem and in the development of efficient, consistent, and robust SLAM algorithms.

SLAM: Where to Next?

The SLAM method provides a solution to the key competency of mapping and localization for any autonomous robot. The past decade, in particular, has seen substantial progress in our understanding of the SLAM problem and in the development of efficient, consistent, and robust SLAM algorithms. The standard state-space approach to SLAM is now well understood, and the main issues in representation, computation, and association appear to be resolved. The information form of the SLAM problem has significant unexplored potential in large-scale mapping, problems involving many vehicles and potentially in mixed environments with sensor networks and dynamic landmarks. The delayed data-fusion concept complements batch association and iterative smoothing to improve estimation quality and robustness. Appearance- and pose-based SLAM methods offer a radically new paradigm for mapping and location estimation without the need for strong geometric landmark descriptions. These methods are opening up new directions and making links back to fundamental principles in robot perception.

The key challenges for SLAM are in larger and more persuasive implementations and demonstrations. While progress has been substantial, the scale and structure of many environments are limited. The challenge now is to demonstrate SLAM solutions to large problems where robotics can truly contribute: driving hundreds of kilometers under a forest canopy or mapping a whole city without recourse to global positioning system (GPS) and to demonstrate true autonomous localization and mapping of structures such as the Barrier Reef or the surface of Mars. SLAM has brought these possibilities closer.

References

[1] S. Argamon-Engelson, "Using image signatures for place recognition," *Pattern Recognit. Lett.*, vol. 19, no. 4, pp. 941–951, 1998.

[2] T. Bailey, "Mobile robot localisation and mapping in extensive outdoor environments," Ph.D. dissertation, Univ. Sydney, Australian Ctr. Field Robotics, 2002.

[3] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003, pp. 1966–1971.

[4] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*. New York: Academic Press, 1988.

[5] S.S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[6] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003, pp. 1899–1906.

[7] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework," *Int. J. Robot. Res.*, vol. 23, no. 12, pp. 1113–1140, 2004.

[8] K.S. Chong and L. Kleeman, "Feature-based mapping in real, large scale environments using an ultrasonic array," *Int. J. Robot. Res.*, vol. 18, no. 1, pp. 3–19, 1999.

[9] I.J. Cox and J.J. Leonard, "Modeling a dynamic environment using a bayesian multiple hypothesis approach," *Artif. Intell.*, vol. 66, no. 2, pp. 311–344, 1994.

[10] A.J. Davison, Y.G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. IFAC/EURON Symp. Intelligent Autonomous Vehicles*, 2004.

[11] M. Deans and M. Hebert, "Experimental comparison of techniques for localization and mapping using a bearing-only sensor," in *Experimental Robotics VII*. D. Rus and S. Singh, Eds. New York: Springler Verlag, 2001.

[12] F. Dellaert, "Square root SAM: Simultaneous location and mapping via square root information smoothing," in *Proc. Robotics: Science and Systems*, 2005.

[13] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Proc. IEEE Int. Conf. Robotics Automation*, 2000, vol. 2, pp. 1009–1014.

[14] A.I. Eliazar and R. Parr, "DP-SLAM 2.0," in *Proc. IEEE Int. Conf. Robotics Automation*, 2004, pp. 1314–1320.

[15] C. Estrada, J. Neira, and J.D. Tardós, "Hierarchical SLAM: Realtime accurate mapping of large environments," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 588–596, 2005.

[16] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the RMS Titanic with SLAM information filters," in *Proc. Robotics: Science and Systems*, 2005.

[17] R. Eustice, M. Walter, and J. Leonard, "Sparse extended information filters: Insights into sparsification," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2005, pp. 3281–3288.

[18] R.M. Eustice, H. Singh, and J.J. Leonard, "Exactly sparse delayed-state filters," in *Proc. IEEE Int. Conf. Robotics Automation*, 2005, pp. 2417–2424.

[19] J. Folkesson and H.I. Christensen, "Graphical SLAM—a selfcorrecting map," in *Proc. IEEE Int. Conf. Robotics Automation*, 2004, pp. 791–798.

[20] G. Grisetti, C. Stachniss, and W. Burgard, "Improving gridbased SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robotics Automation*, 2005, pp. 667–672.

[21] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map building algorithm for real time implementation," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 242–257, 2001.

[22] J. Guivant and E. Nebot, "Improving computational and memory requirements of simultaneous localization and map building algorithms," in *Proc. IEEE Int. Conf. Robotics Automation*, 2002, pp. 2731–2736.

[23] J.S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Computational Intelligence Robotics Automation*, 1999, pp. 318–325.

[24] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fast-SLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2003, pp. 206–211.

[25] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003, pp. 1557–1563.

[26] S.J. Julier and J.K. Uhlmann, "A nondivergent estimation algorithm in the presence of unknown correlations," in *Proc. American Control Conf.*, 1997, vol. 4, pp. 2369–2373.

[27] J. Kim and S. Sukkarieh, "Autonomous airborne navigation in unknown terrain environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 1031–1045, 2004.

[28] J. Knight, A. Davison, and I. Reid, "Towards constant time SLAM using postponement," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 2001, pp. 405–413.

[29] K. Konolige, "Large-scale map-making," in *Proc. Nat. Conf. AI (AAAI)*, 2004, pp. 457–463.

[30] J. Leonard and P. Newman, "Consistent, convergent, and constant-time SLAM," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2003.

[31] J.J. Leonard and H.F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Norwell, MA: Kluwer, 1992.

[32] J.J. Leonard and R.J. Rikoski, "Incorporation of delayed decision making into stochastic mapping," in *Experimental Robotics VII*, D. Rus and S. Singh, Eds. New York: Springler Verlag, 2001

[33] J.J. Leonard, R.J. Rikoski, P.M. Newman, and M.C. Bosse, "Mapping partially observable features from multiple uncertain vantage points," *Int. J. Robot. Res.*, vol. 21, no. 10–11, pp. 943–975, 2002.

[34] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, no. 6, pp. 333–349, 1997.

[35] I. Mahon and S. Williams, "Three-dimensional robotic mapping," in *Proc. Australasian Conf. Robotics Automation*, 2003.

[36] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003, pp. 1985–1991.

[37] J. Neira and J.D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 890–897, 2001.

[38] J. Neira, J.D. Tardós, and J.A. Castellanos, "Linear time vehicle relocation in SLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003.

[39] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," in *Proc. IEEE Int. Conf. Robotics Automation*, 2006.

[40] J. Nieto, T. Bailey, and E. Nebot, "Scan-SLAM: Combining EKFSLAM and scan correlation," in *Proc. Int. Conf. Field Service Robotics*, 2005.

[41] J. Nieto, J. Guivant, and E. Nebot, "The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, 2004, pp. 391–396.

[42] M.A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," Comput. Sci. Div., Univ. California, Tech. Rep., 2002.

[43] Y. Rubner, C. Tomasi, and L.J. Guibas, "A metric for distributions with applications to image databases," in *Proc. IEEE Int. Conf. Computer Vision*, 1998.

[44] J. Solà, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2005, pp. 2751–2756.

[45] J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 311–330, 2002.

[46] S. Thrun, W. Bugard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. Int. Conf. Robotics Automation*, 2000, pp. 321–328.

[47] S. Thrun, D. Koller, Z. Ghahmarani, H. Durrant-Whyte, and A. Ng, "Simultaneous localization and mapping with sparse extended information filters: Theory and initial results," Carnegie Mellon Univ., Tech. Rep., 2002.

[48] S. Thrun, Y. Liu, D. Koller, A. Ng, and H. Durrant-Whyte, "Simultaneous localisation and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7–8, pp. 693–716, 2004.

[49] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. IEEE Int. Conf. Robotics Automation*, 2000, pp. 1023–1029.

[50] M. Walter, R. Eustice, and J. Leonard, "A provably consistent method for imposing sparsity in feature-based SLAM information filters," in *Proc. Int. Symp. Robotics Research*, 2005.

[51] C.C. Wang, C. Thorpe, and S. Thrun, "On-line simultaneous localisation and mapping with detection and tracking of moving objects," in *Proc. IEEE Int. Conf. Robotics Automation*, 2003, pp. 2918–2924.

[52] Z. Wang, S. Huang, and G. Dissanayake, "Implementation issues and experimental evaluation of D-SLAM," in *Proc. Int. Conf. Field Service Robotics*, 2005.

[53] S.B. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, Univ. Sydney, Australian Ctr. Field Robotics, 2001.

Tim Bailey received his B.E. in mechanical and mechatronic engineering at the University of Sydney in 1997 and his Ph.D. at the Australian Centre for Field Robots, University of Sydney, in 2003. He is currently a research fellow at the ACFR. His research interests include mobile robot localization and mapping, Bayesian estimation techniques, and probabilistic sensor modeling.

Hugh Durrant-Whyte received the B.Sc. in nuclear engineering from the University of London, United Kingdom, in 1983, and the M.S.E. and Ph.D. degrees, both in systems engineering, from the University of Pennsylvania, in 1985 and 1986, respectively. From 1987–1995, he was a senior lecturer in engineering science, the University of Oxford, United Kingdom, and a fellow of Oriel College Oxford. From 1995–2002 he was a professor of mechatronic engineering at the University of Sydney. In 2002, he was awarded an Australian Research Council (ARC) Federation Fellowship. He also now leads the ARC Centre of Excellence in Autonomous Systems. His research work focuses on autonomous vehicle navigation and decentralized data fusion methods. His work in applications includes automation in cargo handling, mining, defense, and marine systems. He has published over 300 technical papers and has won numerous awards and prizes for his work.

Address for Correspondence: Tim Bailey, Australian Centre for Field Robotics (ACFR) J04, The University of Sydney, Sydney NSW 2006, Australia. E-mail: tbailey@acfr.usyd.

IEEE Proof