Stefan B. Williams
Hugh Durrant-Whyte

Australian Centre for Field Robotics, J04
The University of Sydney, 2006, Australia
stefanw@acfr.usyd.edu.au
hugh@acfr.usyd.edu.au

Gamini Dissanayake

Department of Mechanical and Manufacturing Engineering
University of Technology Sydney, 2006, Australia
gdissa@eng.uts.edu.au

Constrained Initialization of the Simultaneous Localization and Mapping Algorithm

Abstract

In this paper we present a novel feature initialization technique for the Simultaneous Localization and Mapping (SLAM) algorithm. The initialization scheme extends previous approaches for identifying new confirmed features and is shown to improve the steady-state performance of the filter by incorporating tentative features into the filter as soon as they are observed. Constraints are then applied between multiple feature estimates when a feature is confirmed. Observations that are subsequently deemed as spurious are removed from the state vector after an appropriate timeout. It is shown that information that would otherwise be lost can therefore be used consistently in the filter. Results of this algorithm applied to data collected using a submersible vehicle are also shown.

KEY WORDS—simultaneous localization and mapping, feature initialization, constraint, submersible

1. Introduction

Simultaneous Localization and Mapping (SLAM) is the process of concurrently building a feature-based map of the environment and using this map to obtain estimates of the location of the vehicle. In essence, the vehicle relies on its ability to extract useful navigation information from the data returned by its sensors. The vehicle typically starts at an unknown location with no a priori knowledge of landmark locations. From relative observations of landmarks, it simultaneously computes an estimate of vehicle location and an estimate of landmark locations. While continuing in motion, the vehicle builds a complete map of landmarks and uses these to provide continuous estimates of the vehicle location. By tracking the relative position between the vehicle and identifiable features in the environment, both the position of the vehicle and the position of the features can be estimated simultaneously. The potential for this type of navigation system for autonomous robotic systems operating in unknown environments is enormous.

In this paper we present a novel feature initialization technique for the SLAM algorithm. In Section 2 we begin by introducing the formulation of the SLAM algorithm used for generating vehicle and landmark feature position estimates based on observations taken relative to the position of the vehicle. The development of appropriate vehicle and landmark models is shown together with a discussion of map management issues associated with the filter. In Section 3 we introduce a novel initialization scheme that extends the approach presented in Dissanayake et al. (2000) for identifying new confirmed features. The novel feature initialization scheme can be used to improve the steady-state performance of the filter by incorporating tentative features into the filter as soon as they are observed. Constraints are then applied between multiple feature estimates when a feature is confirmed. Observations that are subsequently deemed as spurious are removed from the state vector after an appropriate timeout. It will be shown that information that would otherwise be lost can therefore be used consistently in the filter. In Section 4 we show the results of this algorithm applied to data collected using a submersible vehicle. Finally, in Section 5 we summarize the contributions of the paper and provide concluding remarks.

2. Simultaneous Localization and Mapping

The SLAM algorithm has recently seen a considerable amount of interest from the mobile robotics community as a tool to enable fully autonomous navigation (Castellanos et al. 2000; Dissanayake et al. 2000; Feder, Leonard, and Smith 1999;

The International Journal of Robotics Research
Vol. 22, No. 7–8, July–August 2003, pp. 541-564,

Leonard and Durrant-Whyte 1991; Newman 1999; Rencken 1993; Thrun, Fox, and Burgard 1998). This algorithm originally appeared in seminal work by Smith, Self and Cheeseman (1987, 1990) and was first demonstrated in work by Ayache and Faugeras (1989). The prospect of deploying a robotic vehicle that can build a map of its environment while simultaneously using that map to localize itself promises to allow these vehicles to operate autonomously for long periods of time in unknown environments. Much of this work has focused on the use of stochastic estimation techniques to build and maintain estimates of vehicle and map feature locations. In particular, the extended Kalman filter (EKF) has been proposed as a mechanism by which the information gathered by the vehicle can be consistently fused to yield bounded estimates of vehicle and landmark locations in a recursive fashion (Dissanayake et al. 2000; Leonard and Durrant-Whyte 1992). Much of the recent work reported in the literature addresses the computational complexity associated with the algorithm (Guivant, Nebot, and Durrant-Whyte 2000; Leonard and Feder 1999; Williams, Dissanayake, and Durrant-Whyte 2002).

While the Kalman filter approach to the SLAM problem has received considerable interest, alternative philosophies also appear in the literature. Some recent work has examined a particle-based approach for tracking the vehicle state estimate to reduce the complexity of the algorithm (Montemerlo et al. 2002). A number of research teams have tackled the problem of map building and localization using batch estimation techniques (Gutmann and Konolige 2000; Lu and Milios 1997; Thrun, Fox, and Burgard 1998). By storing the data collected by the vehicle during its run, they are able to process sensor returns in a batch manner to build maps of the environment in which the vehicle operated. Still other approaches to the problem of map building and localization have done away with the rigorous mathematical models of the vehicle and sensing properties and have relied instead on more qualitative knowledge of the nature of the environment (Brooks 1986; Kuipers and Byun 1991; Levitt and Lawton 1990). These methods have a certain appeal in that they eliminate the need for accurate models of the vehicle motion and sensing processes, limit the computational requirements of map building and have a certain anthropomorphic appeal. While all of these alternative approaches to the problem have their own particular strengths, in this paper we are concerned primarily with a recursive, online approach to the problem and will rely on the EKF as the primary means of simultaneously building a map while localizing the vehicle.

2.1. System States

The SLAM algorithm represents the state of the environment and the state of the vehicle within it, as shown in Figure 1. The vehicle travels through the environment using its sensors to observe features around it. The state of the system at time k can therefore be represented by the augmented state vector, $\mathbf{x}(k)$, consisting of the n_v states representing the vehicle, $\mathbf{x}_v(k)$, and the n_f states describing the observed features, $\mathbf{x}_i(k)$, $i = 1, \ldots, n_f$

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{x}_1(k) \\ \vdots \\ \mathbf{x}_{n_f}(k) \end{bmatrix}. \tag{1}$$

The system state vector can be written more concisely by lumping the map features into the common term $\mathbf{x}_m(k)$

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ \mathbf{x}_m(k) \end{bmatrix}. \tag{2}$$

2.2. The Vehicle and Landmark Models

The process model for a system describes how the system states change as a function of time and is usually written as a first-order non-linear vector differential equation or state model of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{v}(t) \tag{3}$$

where $\mathbf{x}(t) \in R^n$ is a vector of the states of interest at time t, $\mathbf{u}(t) \in R^r$ is a known control input, $\mathbf{f}(\cdot, \cdot, \cdot)$ is a model of the rate of change of system state as a function of time, and $\mathbf{v}(t)$ is a random vector describing both dynamic driving noise and uncertainties in the state model itself.

In general, process models are continuous and must be discretized for implementation on a digital computer. In this paper we consider discrete process and observation models directly but it should be remembered that these models are derived from their continuous time counterparts. The discrete equivalent of eq. (3) can be written as

$$\mathbf{x}(t_k) = \mathbf{f}(\mathbf{x}(t_{k-1}), \mathbf{u}(t_k), t_k) + \mathbf{v}(t_k) \tag{4}$$

where the function $\mathbf{f}(\cdot, \cdot, \cdot)$ now maps the state $\mathbf{x}(t_{k-1})$ at time t_{k-1} and the control input $\mathbf{u}(t_k)$ at time t_k to the state $\mathbf{x}(t_k)$ at time t_k. In almost all cases considered here, the time interval $\Delta t(k) \triangleq t_k - t_{k-1}$ between successive samples of the state remains constant. In this case, it is common practice to drop the time argument from eq. (4) and simply index the variables by the sample number

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k)) + \mathbf{v}(k). \tag{5}$$

2.2.1. Vehicle Model

A vehicle model attempts to capture the fundamental relationship between the vehicle's past state, $\mathbf{x}_v(k-1)$, and its current state, $\mathbf{x}_v(k)$, given a control input, $\mathbf{u}(k)$

$$\mathbf{x}_v(k) = \mathbf{f}_v(\mathbf{x}_v(k-1), \mathbf{u}(k)) + \mathbf{v}_v(k) \tag{6}$$

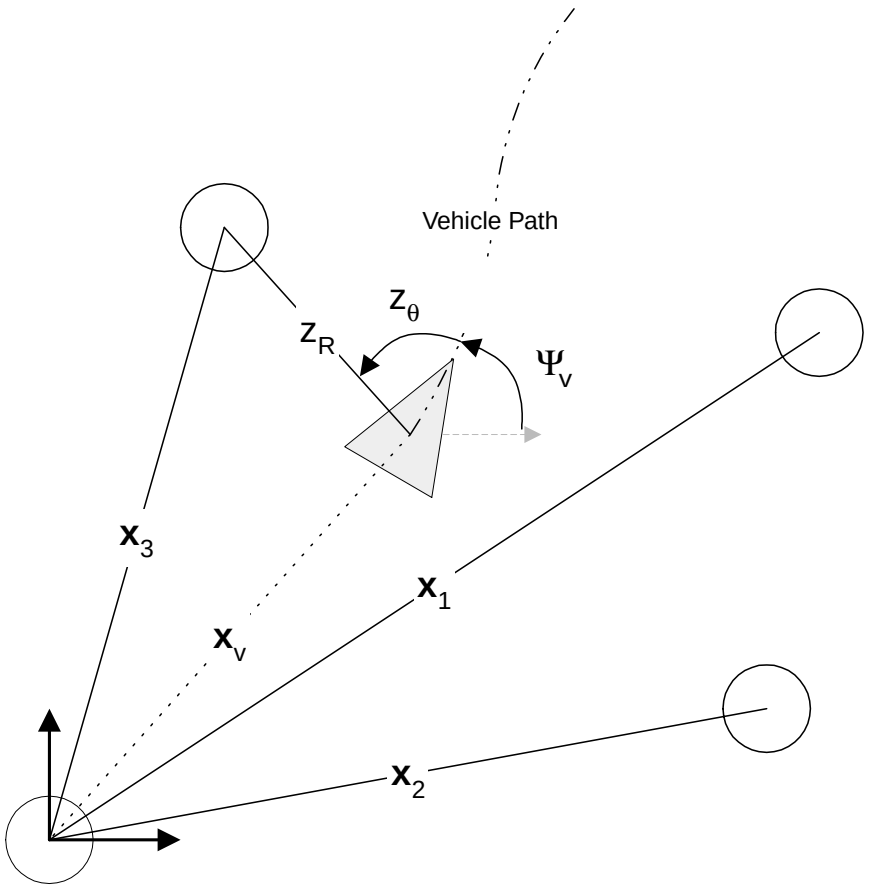


Fig. 1. The system states involved in the SLAM algorithm. The vehicle travels through the environment taking observations to features using its on-board sensors.

where $\mathbf{x}_v(k) \in R^{n_v}$ are the vehicle states at time step k, and $\mathbf{v}_v(k)$ is a random vector describing both dynamic driving noise and uncertainties in the vehicle state model itself.

An accurate vehicle model is an essential component of most navigation schemes. Vehicle models can be of varying degrees of complexity. Depending on the reliability with which the motion of the vehicle can be modeled and the sensors available for predicting the change of state of the vehicle, highly accurate estimates of vehicle motion can be generated. Ideally, of course, a vehicle model would capture the motion of the vehicle without any uncertainty and, for each state transition, the vehicle state would be precisely known. This is a practical impossibility, however, as the models used do not, in general, perfectly capture the vehicle motion and sensors and actuators are subject to noise that will slowly corrupt the accuracy of the state estimate.

2.2.2. Landmark Model

In the context of SLAM, a landmark is a feature of the environment that can be consistently and reliably observed using the vehicle's sensors. Landmarks must be described in parametric form to allow them to be incorporated into a state model. Point landmarks, corners, line and polyline feature models have all been reported in the literature (Castellanos et al. 1999; Leonard and Durrant-Whyte 1991). For the SLAM algorithm, the feature states are usually assumed to be stationary. While not an essential condition of the SLAM algorithm, tracking moving map features in the environment is not considered of much value for the purposes of navigation. Moving features can, however, be accommodated given the formulation of the SLAM algorithm using an EKF and the novel initialization routine proposed in this paper can serve to defer the identification of stationary landmarks until a number of observations have been taken. This is important if the vehicle is operating in a dynamic environment.

For the work considered here, in which only static targets are included in the map, the dynamic portion of the process model consists only of a vehicle model. This leads to the simple landmark model

$$\mathbf{x}_m(k) = \mathbf{x}_m(k-1) \tag{7}$$

where $\mathbf{x}_m(k) \in R^{n_f}$ are the landmark states at time step k.

2.2.3. Sensor Models

The state observation process can also be modeled in state-space notation by a non-linear vector function in the form

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \tag{8}$$

where $\mathbf{z}(t) \in R^m$ is the observation made at time t, $\mathbf{h}(\cdot, \cdot, \cdot)$ is a model of the observation of system states as a function of time, and $\mathbf{w}(t)$ is a random vector describing both measurement corruption noise and uncertainties in the measurement model itself. These models describe the true observation at time t given the true state of the system $\mathbf{x}(t)$.

The observation process lends itself well to the discrete observation model

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k) \tag{9}$$

as the state observations are often captured using a digital computer system.

2.3. The Estimation Process

Given the process and observation models described in the previous sections, the localization and map building process consists of generating the best estimate for the system states given the information available to the system. This can be accomplished using a recursive, three-stage procedure comprising prediction, observation and update steps known as the EKF (Dissanayake et al. 2000). The Kalman filter is a recursive, least-squares estimator and produces at time i a minimum mean-squared error estimate $\hat{\mathbf{x}}(i|j)$ of the state $\mathbf{x}(i)$ given a sequence of observations up to time j, $\mathbf{Z}^j = \{\mathbf{z}(1)...\mathbf{z}(j)\}$

$$\hat{\mathbf{x}}(i|j) = E[\mathbf{x}(i)|\mathbf{Z}^j]. \tag{10}$$

The development of the EKF equations detailed here can be found in numerous texts on the subject (Durrant-Whyte 2001; Gelb 1996; Maybeck 1982; Miller and Leskiw 1987).

Adopting the notation of Gelb (1996), the posterior estimate of the state at time k conditioned on the information up to time k will be written as $\hat{\mathbf{x}}^+(k)$

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}(k|k). \tag{11}$$

The prior estimate of the state at time k given the information up to time $k-1$, also referred to as the one-step-ahead prediction, will be written $\hat{\mathbf{x}}^-(k)$

$$\hat{\mathbf{x}}^-(k) = \hat{\mathbf{x}}(k|k-1). \tag{12}$$

The filter fuses a prior state estimate $\hat{\mathbf{x}}^-(k)$ with an observation $\mathbf{z}(k)$ of the state $\mathbf{x}(k)$ at time k to produce the updated estimate $\hat{\mathbf{x}}^+(k)$.

For the SLAM algorithm, the EKF is used to estimate the pose of the vehicle $\hat{\mathbf{x}}_v^+(k)$ along with the positions of the n_f observed features $\hat{\mathbf{x}}_i^+(k)$, $i = 1...n_f$. The augmented state vector, eq. (1), gives rise to the augmented state estimate consisting of the current vehicle state estimates as well as those associated with the observed features

$$\hat{\mathbf{x}}^+(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^+(k) \\ \hat{\mathbf{x}}_1^+(k) \\ \vdots \\ \hat{\mathbf{x}}_{n_f}^+(k) \end{bmatrix}. \tag{13}$$

The covariance matrix for this state estimate is defined through

$$\mathbf{P}^+(k) = E[(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))(\mathbf{x}(k) - \hat{\mathbf{x}}^+(k))^{\mathrm{T}}|\mathbf{Z}^k]. \tag{14}$$

This defines the mean-squared error and error correlations in each of the state estimates. The covariance matrix takes on the following form

$$\mathbf{P}^+(k) = \begin{bmatrix} \mathbf{P}_{vv}^+(k) & \mathbf{P}_{v1}^+(k) & \dots & \mathbf{P}_{vn}^+(k) \\ \mathbf{P}_{v1}^{+\mathrm{T}}(k) & \mathbf{P}_{11}^+(k) & \dots & \mathbf{P}_{1n}^+(k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{vn}^{+\mathrm{T}}(k) & \mathbf{P}_{1n}^{+\mathrm{T}}(k) & \dots & \mathbf{P}_{nn}^+(k) \end{bmatrix}. \tag{15}$$

The covariance matrix can be written more concisely using $\mathbf{P}_{mm}^+(k)$ to represent the map covariance and $\mathbf{P}_{vm}^+(k)$ to represent the cross-covariance between the vehicle and the map:

$$\mathbf{P}^+(k) = \begin{bmatrix} \mathbf{P}_{vv}^+(k) & \mathbf{P}_{vm}^+(k) \\ \mathbf{P}_{vm}^{+\mathrm{T}}(k) & \mathbf{P}_{mm}^+(k) \end{bmatrix}. \tag{16}$$

It is important to note that it is the correlation terms between the vehicle and map feature estimates, resulting from the fact that the vehicle is observing features using its on-board sensors, that allows the SLAM filter to converge (Dissanayake et al. 2001). These correlations must be maintained in order to preserve the consistency of the filter.

2.3.1. Prediction

The prediction stage of the filter uses the model of the motion of the vehicle defined in eq. (6) to generate an estimate of the vehicle position, $\hat{\mathbf{x}}_v^-(k)$, at instant k given the information available to instant $k-1$ as

$$\begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_m^-(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}_v^+(k-1), \mathbf{u}(k)) \\ \hat{\mathbf{x}}_m^+(k-1) \end{bmatrix}. \tag{17}$$

The covariance matrix must also be propagated through the vehicle model as part of the prediction. The EKF linearizes the propagation of uncertainty about the current state estimate $\hat{\mathbf{x}}^+(k-1)$ using the Jacobian $\nabla_{\mathbf{x}}\mathbf{f}(k)$ of $\mathbf{f}$ evaluated at $\hat{\mathbf{x}}^+(k-1)$ as

$$\mathbf{P}^-(k) = \nabla_{\mathbf{x}}\mathbf{f}(k)\mathbf{P}^+(k-1)\nabla_{\mathbf{x}}\mathbf{f}^{\mathrm{T}}(k) + \mathbf{Q}(k). \tag{18}$$

For the SLAM algorithm, this step in the filter can be simplified because of the assumption that the feature states are

stationary. This allows the complexity of computing the predicted covariance to be reduced by requiring that only the variances associated with the vehicle and the cross-covariance terms between the vehicle and the map are updated during the prediction step

$$\begin{bmatrix} \mathbf{P}^-_{vv}(k) & \mathbf{P}^-_{vm}(k) \\ \mathbf{P}^{-\mathrm{T}}_{vm}(k) & \mathbf{P}^-_{mm}(k) \end{bmatrix} = \tag{19}$$

$$\begin{bmatrix} \nabla_{\mathbf{v}}\mathbf{f}(k)\mathbf{P}^+_{vv}(k-1)\nabla_{\mathbf{v}}\mathbf{f}^{\mathrm{T}}(k) + \mathbf{Q}_{vv}(k) & \nabla_{\mathbf{v}}\mathbf{f}(k)\mathbf{P}^+_{vm}(k-1) \\ (\nabla_{\mathbf{v}}\mathbf{f}^{\mathrm{T}}(k)\mathbf{P}^+_{vm}(k-1) & \mathbf{P}^+_{mm}(k-1) \end{bmatrix}$$

where the noise term, $\mathbf{Q}_{vv}(k)$, here represents the lumped process and control noise terms for conciseness.

2.3.2. Observation

The fusion of the observation into the state estimate is accomplished by first calculating a predicted observation, $\hat{\mathbf{z}}^-(k)$, using the observation model, $\mathbf{h}$ as

$$\hat{\mathbf{z}}^-(k) = \mathbf{h}(\hat{\mathbf{x}}^-(k)). \tag{20}$$

When observations are received from the vehicle's on-board sensors they must be associated with particular features in the environment. The difference between the actual observation, $\mathbf{z}(k)$, received from the system's sensors and the predicted observation, $\hat{\mathbf{z}}^-(k)$, is termed the innovation $\nu(k)$

$$\nu(k) = \mathbf{z}(k) - \hat{\mathbf{z}}^-(k). \tag{21}$$

The innovation covariance, $\mathbf{S}(k)$, is computed from the current state covariance estimate, $\mathbf{P}^-(k)$, the Jacobian of the observation model, $\nabla_{\mathbf{x}}\mathbf{h}(k)$, and the covariance of the observation model $\mathbf{R}(k)$:

$$\mathbf{S}(k) = \nabla_{\mathbf{x}}\mathbf{h}(k)\mathbf{P}^-(k)\nabla_{\mathbf{x}}\mathbf{h}^{\mathrm{T}}(k) + \mathbf{R}(k). \tag{22}$$

As will be shown, the innovations and their associated covariances can be used to validate measurements before they are incorporated into the filtered estimates. The calculation of the innovation covariance can be simplified by noting that each observation is only a function of the feature being observed. The Jacobian of the observation function, $\nabla_{\mathbf{x}}\mathbf{h}(k)$, is therefore a sparse matrix of the form

$$\nabla_{\mathbf{x}}\mathbf{h}(k) = \begin{bmatrix} \nabla_{\mathbf{v}}\mathbf{h}(k) & 0 & \dots & 0 & \nabla_{\mathbf{i}}\mathbf{h}(k) & 0 & \dots \end{bmatrix}. \tag{23}$$

Evaluating the product in eq. (22) using the sparse Jacobian results in

$$\begin{aligned} \mathbf{S}(k) = {} & \nabla_{\mathbf{v}}\mathbf{h}(k)\mathbf{P}^-_{vv}(k)\nabla_{\mathbf{v}}\mathbf{h}^{\mathrm{T}}(k) + \nabla_{\mathbf{i}}\mathbf{h}(k)\mathbf{P}^-_{vi}(k)\nabla_{\mathbf{v}}\mathbf{h}^{\mathrm{T}}(k) + \\ & \nabla_{\mathbf{v}}\mathbf{h}(k)\mathbf{P}^{-\mathrm{T}}_{vi}(k)\nabla_{\mathbf{i}}\mathbf{h}^{\mathrm{T}}(k) + \nabla_{\mathbf{i}}\mathbf{h}(k)\mathbf{P}^-_{ii}(k)\nabla_{\mathbf{i}}\mathbf{h}^{\mathrm{T}}(k) + \mathbf{R}(k). \end{aligned} \tag{24}$$

2.3.3. Update

Once the observation has been associated with a particular feature in the map, the state estimate can be updated using the optimal gain matrix $\mathbf{W}(k)$. This gain matrix provides a weighted sum of the prediction and observation and is computed using the innovation covariance, $\mathbf{S}(k)$, and the predicted state covariance, $\mathbf{P}^-(k)$. The weighting factor is proportional to $\mathbf{P}^-(k)$ and inversely proportional to the innovation covariance (Smith, Self, and Cheesman 1990). This is used to compute the state update $\hat{\mathbf{x}}^+(k)$ as well as the updated state covariance $\mathbf{P}^+(k)$

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\nu(k) \tag{25}$$

$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^{\mathrm{T}}(k) \tag{26}$$

where

$$\mathbf{W}(k) = \mathbf{P}^-(k)\nabla_{\mathbf{x}}\mathbf{h}^{\mathrm{T}}(k)\mathbf{S}^{-1}(k). \tag{27}$$

2.3.4. Feature Initialization

When a new feature is observed, its estimate must be properly initialized and added to the state vector. Given a current state estimate, $\hat{\mathbf{x}}^-(k)$, comprised of the vehicle state, $\hat{\mathbf{x}}^-_v(k)$, and the map states, $\hat{\mathbf{x}}^-_m(k)$, a relative observation between the vehicle and the new feature, $\mathbf{z}(k)$, and a feature initialization model, $\mathbf{g}_i(\cdot,\cdot)$, that maps the current vehicle state estimate and observation to a new feature estimate, the initial estimate of the feature state is

$$\hat{\mathbf{x}}^+_i(k) = \mathbf{g}_i(\hat{\mathbf{x}}^-_v(k), \mathbf{z}(k)). \tag{28}$$

These new state estimates are then appended to the state vector as new map feature elements.

The covariances of the new feature estimates must also be properly initialized since the initial estimate depends on the current vehicle estimate and is therefore correlated with the rest of the vehicle and other map state estimates. Ignoring the correlation between the new state estimates and the remainder of the map can lead to inconsistency in the filtering process (Csorba 1997). The covariance matrix is first augmented with the observation covariance and the cross-covariance terms between the existing state elements and the new state estimates are computed. Assume the initial covariance matrix, $\mathbf{P}^-(k)$, is

$$\mathbf{P}^-(k) = \begin{bmatrix} \mathbf{P}^-_{vv}(k) & \mathbf{P}^-_{vm}(k) \\ \mathbf{P}^{-\mathrm{T}}_{vm}(k) & \mathbf{P}^-_{mm}(k) \end{bmatrix} \tag{29}$$

The covariance matrix is then augmented with the observation covariance, $\mathbf{R}(k)$

$$\mathbf{P}^{*-}(k) = \begin{bmatrix} \mathbf{P}^-_{vv}(k) & \mathbf{P}^-_{vm}(k) & 0 \\ \mathbf{P}^{-\mathrm{T}}_{vm}(k) & \mathbf{P}^-_{mm}(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix}. \tag{30}$$

The final covariance is computed by projecting the augmented covariance matrix through the Jacobian $\nabla_{\mathbf{x}}\mathbf{g}(k)$ of the initialization function, $\mathbf{g}_i$, with respect to the augmented states

$$\mathbf{P}^{+}(k) = \nabla_{\mathbf{x}}\mathbf{g}(k)\mathbf{P}^{*-}(k)\nabla_{\mathbf{x}}\mathbf{g}^{\mathrm{T}}(k) \tag{31}$$

with

$$\nabla_{\mathbf{x}}\mathbf{g}(k) = \begin{bmatrix} \mathbf{I}_v & 0 & 0 \\ 0 & \mathbf{I}_m & 0 \\ \nabla_{\mathbf{v}}\mathbf{g}(k) & 0 & \nabla_{\mathbf{z}}\mathbf{g}(k) \end{bmatrix}. \tag{32}$$

This $O(n^3)$ operation can be simplified by exploiting the sparse nature of the Jacobian, $\nabla_{\mathbf{x}}\mathbf{g}(k)$

$$\mathbf{P}^{+}(k) = \tag{33}$$
$$\begin{bmatrix} \mathbf{P}^{-}_{vv}(k) & \mathbf{P}^{-}_{vm}(k) & (\mathbf{P}^{-\mathrm{T}}_{vv}(k)\nabla_{\mathbf{v}}\mathbf{g}^{\mathrm{T}}(k)) \\ \mathbf{P}^{-\mathrm{T}}_{vm}(k) & \mathbf{P}^{-}_{mm}(k) & (\mathbf{P}^{-\mathrm{T}}_{vm}(k)\nabla_{\mathbf{v}}\mathbf{g}^{\mathrm{T}}(k)) \\ \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}^{-}_{vv}(k) & \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}^{-}_{vm}(k) & \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}^{-}_{vv}(k)\nabla_{\mathbf{v}}\mathbf{g}^{\mathrm{T}}(k) \\ & & +\nabla_{\mathbf{z}}\mathbf{g}(k)\mathbf{R}(k)\nabla_{\mathbf{z}}\mathbf{g}^{\mathrm{T}}(k) \end{bmatrix}.$$

Proper initialization of the feature estimates is necessary to maintain their consistency and to generate the correct cross-covariances between the feature and vehicle estimates.

2.4. Filter Management

To successfully manage the estimation process, and to supply it with reliable and robust feature observations, a number of additional considerations must be taken into account. These include the methods used for extracting and identifying features and for data association when new observations are received. In this section we examine some of these issues in more detail.

2.4.1. Feature Extraction

The development of autonomous feature-based navigation relies on the ability of a sensor system to extract appropriate and reliable features with which to build maps. The feature extraction process is highly application-dependent and depends on the anticipated environment in which the vehicle will operate, and the sensors used to observe this environment. Some environments, such as those found in typical offices, lend themselves to the extraction of corner and line features using such sensors as lasers and vision systems. In unstructured environments, simple corner or line features are not commonly observed. Additionally, outdoor sensors very often cannot provide the same accuracy or detail as can be obtained from indoor navigation sensors. Should the vehicle be required to operate in more unstructured environments, these features may prove to be insufficient for modeling the vehicle's surroundings. Alternative methods for terrain modeling have recently appeared in the literature (Majumder 2001) and it will be interesting to see how some of this work develops over the coming years.

In this paper we are concerned primarily with SLAM using point features. Point features are simply defined as a point in the environment that yields consistent, reliable and viewpoint invariant sensor returns. What constitutes a good point feature will once again be application-dependent, and in particular will be affected by the quality of available sensors. However, the techniques presented are not limited to the estimation of point features. They can readily be extended to include the estimation of any alternative feature types that can be parametrized for use in the Kalman filter framework.

2.4.2. Data Association

Data association is the process matching observations that are received by the filter to the features to which they correspond. Given that the state estimation process relies on generating statistical estimates of the locations of features in the environment, statistical methods are used for establishing these associations. The most common method for associating observations to features in the map relies on nearest-neighbor techniques (Dissanayake et al. 2000; Feder 1999; Leonard and Feder 1999; Newman 1999). A nearest-neighbor association is taken in this case to be the closest association in a statistical sense. A common statistical discriminator is based on the normalized innovation squared between two estimates. Given an observation $\mathbf{z}(k)$ comprising a range and bearing to the observed landmark, the innovation, $\nu(k)$, and innovation covariance, $\mathbf{S}(k)$, can be calculated as shown in eqs. (21) and (22), respectively. The normalized innovation squared between the observation and the estimated feature location is then compared against a validation gate, d_{min}, for the association being considered.

$$d_{fi} = \nu^{\mathrm{T}}(k)\mathbf{S}^{-1}(k)\nu(k) < d_{min}. \tag{34}$$

The normalized innovation squared forms a χ^2 distribution that can be used to accept or reject a particular association with a given confidence level by the appropriate selection of d_{min}.

Data association is essential to the operation of the SLAM algorithm. The estimated location of landmark positions relies on the accuracy of the vehicle location estimate. An incorrect association of an observation to the map can cause the filter to diverge from a consistent estimate, effectively rendering all future predicted observations incorrect.

Unfortunately, it is quite difficult to detect and recover from an incorrect association by relying exclusively on nearest-neighbor techniques, as associations depend on calculating a predicted observation based on the current estimate of vehicle location. If the vehicle location estimate is in error, then an observation to a known landmark will be estimated to have occurred from a different map position. In this case, there is a risk that the filter may incorrectly associate the observation with another landmark in the map or initialize a new feature at this position.

Recent work in this area has examined the possibility of using multiple hypothesis approaches (Smith, Feder, and Leonard 1998) or joint compatibility criteria (Neira and Tardos 2001) to improve the performance of data association when compared with simple nearest-neighbor approaches.

2.4.3. Map Management

Once features have been identified, they must be matched against known landmarks in the environment. The first step is to perform data association between the observed feature and the features currently in the map. This step is one of the most crucial in the mapping process. Erroneous data association can destroy the integrity of the map. When data are received from a sensor, there is a possibility that the data may in fact be spurious measurements. Some spurious measurements can be eliminated by the development of appropriate feature extraction routines; by studying and modeling the physical phenomena that are being measured by the sensor, it is possible to reduce the number of spurious measurements picked up by a feature extractor. Regardless of the care that is taken in designing the feature extractor, some spurious measurements may still be passed to the localization and mapping algorithm and it is important to have a mechanism for rejecting these. A two-step matching algorithm is used in order to reduce the number of landmarks that are added to the map (see Figure 2).

When a new range and bearing observation is received from the feature extraction process, the estimated position of the feature is computed using the current estimate of vehicle position. This position is then compared with the estimated positions of the features in the map using the data association strategies given in eq. (34) (Dissanayake et al. 2000). If the observation can be associated with a single feature, the EKF is used to generate a new state estimate. An observation that can be associated with multiple landmarks is rejected, since false associations can destroy the integrity of the estimation process.

If the observation does not match any landmarks in the current map, it is compared against a list of tentative landmarks. Each tentative landmark maintains a counter indicating the number of associations that have been made with the feature as well as the last observed position of the feature. If a match is made, the counter is incremented and the observed position is updated. When the counter passes a threshold value, the feature is considered to be sufficiently stable and is added to the map. If the potential feature cannot be associated with any of the tentative features, a new tentative feature is added to the list. Tentative features that are not reobserved are removed from the list after a fixed time interval has elapsed.

3. Constrained Initialization

In this section we introduce a novel feature initialization scheme that can be used to improve the steady-state performance of the filter. By incorporating tentative feature observations into the filter and then applying constraints when the feature is confirmed, information that would otherwise be lost can be used in the filter. This is a valuable tool in instances where observation rates are low compared with the speed of the vehicle or in areas of high signal clutter, where data association is difficult. Both cases can result in relatively few confirmed observations of each feature. The approach presented here considers the case when the observation consists of a range and bearing to a target implying that a feature estimate is fully observable with a single observation. For analogous methods for the case in which this is not the case, refer to the work of Rikoski, Leonard, and Newman (2002) and Deans (2002) in which more temporal information is stored in the filter.

3.1. Associating Observations

When observations are received by the SLAM process, the first step is to perform data association between the observed feature and the features currently in the map. This step is one of the most crucial in the mapping process. Erroneous data association can destroy the integrity of the map. In many instances, however, ambiguities can arise due to the uncertainty in the vehicle position. This uncertainty translates through the large vehicle covariance into a relatively large area of plausible vehicle locations. This in turn may lead to multiple features yielding potentially consistent associations.

As discussed in Section 2.4.2, the normalized innovation squared distance is often used as a statistical gate with which to validate the association between an observation and an estimated feature state. There are instances, however, when this validation procedure might result in ambiguous data associations. Should multiple features match to within the confidence bounds set for the algorithm, there is a risk that the wrong feature will be matched to the observation. Fusing this erroneous observation into the state estimate can cause the filter to collapse and the estimate to diverge from the true state. This situation can be difficult to detect in implementations of the SLAM algorithm given that all future associations will be dependent on the estimated state of the vehicle. It is therefore important to ensure that the risks of false associations are minimized. Methods for deferring data association to allow a more informed choice of association possibilities will be examined in some detail in the remainder of this section.

3.2. Initializing the Mapping Process

When an observation is received by the filter, the decision must be made as to whether it comes from a known feature in the environment, from a new feature in the environment or is simply a spurious sensor reading. In this section we develop a constrained estimator that allows the initialization phase of the algorithm to be deferred until a potential feature is confirmed by additional observations.

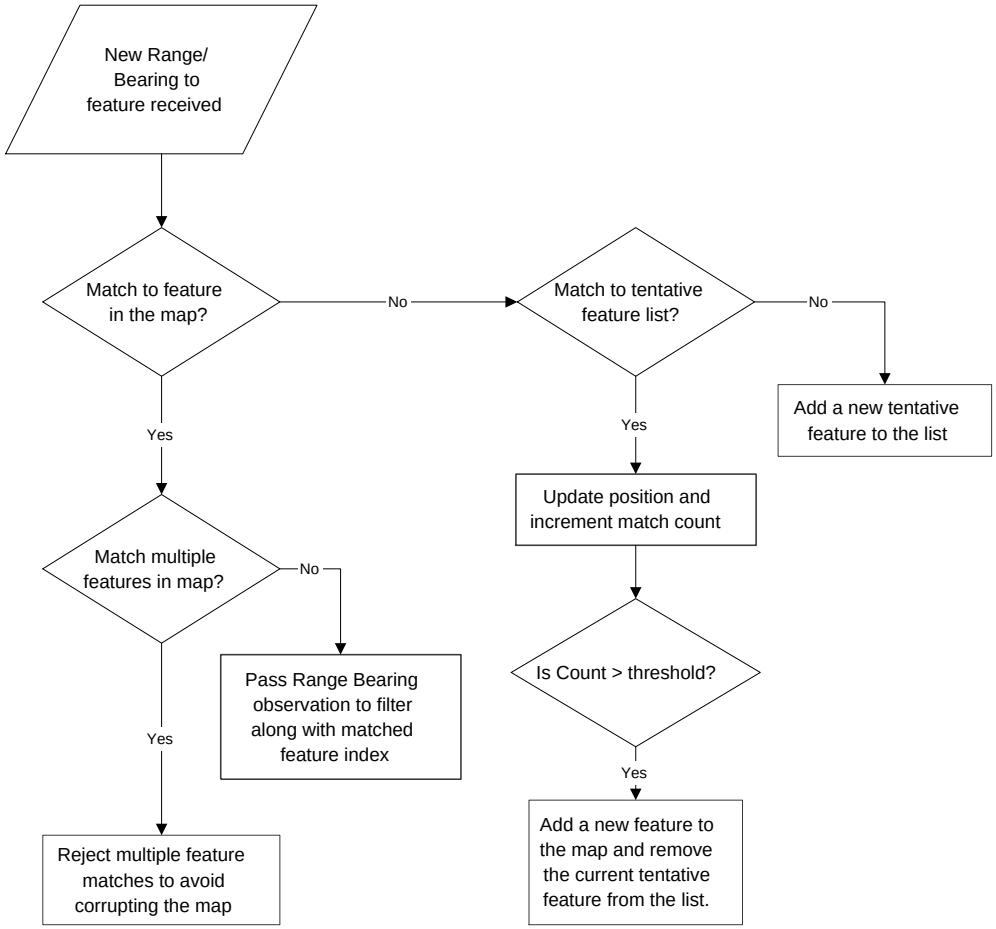


Fig. 2. The feature matching algorithm.

When data are received from a sensor, there is a possibility that the data may in fact be spurious measurements. Some spurious measurements can be eliminated by the development of appropriate feature extraction routines; by studying and modeling the physical phenomena that are being measured by the sensor it is possible to reduce the number of spurious measurements picked up by a feature extractor. Regardless of the care that is taken in designing the feature extractor, some spurious measurements may still be passed to the localization and mapping algorithm and it is important to have a mechanism for rejecting these.

As described in Section 2.4.3, this implementation of the SLAM algorithm relies on maintaining a list of tentative features in the environment. Observations are matched against these tentative features and a new feature is initialized by the mapping algorithm only after the tentative feature is confirmed through multiple sightings. This approach helps to reduce the clutter that would result from adding every observation to the feature map as a confirmed feature. One key problem with this approach is that important observation information about the features is discarded since they are not incorporated into the map. In the next section we develop a constrained estimator that can be used to recover the initial features observations and allows them to be fused consistently into the map.

3.3. Constrained Feature Initialization

One outstanding issue with the method for identifying new map features described in Section 2.4.3 arises from the fact that observations of the tentative features are not incorporated into the estimation process. Since the covariance between the observed feature and the vehicle and map estimates are not maintained, this information cannot be used without the estimate being inconsistent. This results in a loss of information about the vehicle and map feature estimates. This is particularly problematic when the algorithm is first being initialized.

Given that there are no features in the map to begin with, the vehicle must rely exclusively on dead reckoning until the first map feature(s) are initialized. With a relatively poor

dead-reckoning model, the growth of uncertainty in the vehicle position can be quite rapid and the vehicle position uncertainty becomes large. This large uncertainty is reflected in the feature position estimates which also become quite uncertain. The steady-state covariance of the feature estimates is a function of the vehicle covariance when the first feature is initialized (Csorba 1997; Dissanayake et al. 2000) suggesting that it is important to use the earliest vehicle state estimate possible when initializing new landmarks. A novel initialization routine is proposed here that allows the tentative observations to be used consistently in the estimation process. As will be shown, this method can also be used to overcome ambiguous data association since the data association decisions can be deferred until more information becomes available.

In the case where a feature estimate, $\hat{\mathbf{x}}_i^-(k)$, exists along with the estimates of the vehicle, $\hat{\mathbf{x}}_v^-(k)$, and the rest of the map, $\hat{\mathbf{x}}_m^-(k)$, the state vector can be written as follows:

$$\hat{\mathbf{x}}^-(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_m^-(k) \\ \hat{\mathbf{x}}_i^-(k) \end{bmatrix} \tag{35}$$

with covariance

$$\mathbf{P}^-(k) = \begin{bmatrix} \mathbf{P}_{vv}^-(k) & \mathbf{P}_{vm}^-(k) & \mathbf{P}_{vi}^-(k) \\ \mathbf{P}_{vm}^{-\mathrm{T}}(k) & \mathbf{P}_{mm}^-(k) & \mathbf{P}_{mi}^-(k) \\ \mathbf{P}_{vi}^{-\mathrm{T}}(k) & \mathbf{P}_{mi}^{-\mathrm{T}}(k) & \mathbf{P}_{ii}^-(k) \end{bmatrix}. \tag{36}$$

If an observation, $\mathbf{z}_i(k)$, of feature i is made at time k this observation would normally be used to update the full map estimate. Consider for a moment the linear case in which the observation prediction equation can be written as

$$\hat{\mathbf{z}}_i^-(k) = \mathbf{H}\hat{\mathbf{x}}^-(k) \tag{37}$$

$$= -\mathbf{H}_v\hat{\mathbf{x}}_v^-(k) + \mathbf{H}_i\hat{\mathbf{x}}_i^-(k) \tag{38}$$

with covariance $\mathbf{H}\mathbf{P}^-(k)\mathbf{H}^{\mathrm{T}}$. Using the standard Kalman filter update equations, the state estimate can be updated as

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\nu(k) \tag{39}$$

with

$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^{\mathrm{T}}(k) \tag{40}$$

where $\mathbf{W}(k)$ is the Kalman gain, $\nu(k)$ represents the innovation and $\mathbf{S}(k)$ represents the innovation covariance (Maybeck 1982).

Alternatively, the observation of the feature, $\mathbf{x}_i$, can be used to initialize a new estimate of the state. The state vector is first augmented with the observation and the linear observation initialization model is applied to yield the new estimate

$$\hat{\mathbf{x}}^{*+}(k) = \mathbf{G}\hat{\mathbf{x}}^{*-}(k) \tag{41}$$

with

$$\hat{\mathbf{x}}^{*-}(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_m^-(k) \\ \hat{\mathbf{x}}_i^-(k) \\ \mathbf{z}(k) \end{bmatrix} \tag{42}$$

and

$$\mathbf{G} = \begin{bmatrix} I_v & 0 & 0 & 0 \\ 0 & I_m & 0 & 0 \\ 0 & 0 & I_i & 0 \\ \mathbf{H}_i^{\dagger}\mathbf{H}_v & 0 & 0 & \mathbf{H}_i^{\dagger} \end{bmatrix} \tag{43}$$

where $\mathbf{H}_i^{\dagger}$ represents the generalized inverse of the landmark observation model, and I_v, I_m and I_i represent the appropriately dimensioned identity matrices.

The updated state covariance estimate is

$$\mathbf{P}^{*+}(k) = \mathbf{G}(k)\mathbf{P}^{*-}(k)\mathbf{G}^{\mathrm{T}}(k) \tag{44}$$

where

$$\mathbf{P}^{*-}(k) = \begin{bmatrix} \mathbf{P}_{vv}^-(k) & \mathbf{P}_{vm}^-(k) & \mathbf{P}_{vi}^-(k) & 0 \\ \mathbf{P}_{vm}^{-\mathrm{T}}(k) & \mathbf{P}_{mm}^-(k) & \mathbf{P}_{mi}^-(k) & 0 \\ \mathbf{P}_{vi}^{-\mathrm{T}}(k) & \mathbf{P}_{mi}^{-\mathrm{T}}(k) & \mathbf{P}_{ii}^-(k) & 0 \\ 0 & 0 & 0 & \mathbf{R} \end{bmatrix}. \tag{45}$$

While it might seem that information is being lost at this point, due to the initialization of the new feature, it is possible to use a virtual observation in the form of a constraint to recover the true feature estimate. In the case considered above, the following constraint between the states must hold

$$\hat{\mathbf{x}}_i^+(k) - \hat{\mathbf{x}}_{i*}^+(k) = 0. \tag{46}$$

In Appendix A we present the development of a constrained estimator based on the Kalman filter as proposed in Newman (1999). The constraint in eq. (46) can be satisfied using the constraint equation with the following values:

$$\begin{aligned} \mathbf{C}\hat{\mathbf{x}}^+(k) &= \mathbf{b} \\ \mathbf{C} &= \begin{bmatrix} 0 & 0 & 1 & -1 \end{bmatrix} \\ \mathbf{b} &= 0. \end{aligned} \tag{47}$$

Applying the constraints to the posterior estimates $\hat{\mathbf{x}}^{*+}(k)$and $\mathbf{P}^{*+}(k)$ considered above and rearranging terms yields the following constrained estimate:

$$\hat{\mathbf{x}}_c^+(k) = \hat{\mathbf{x}}^{*+}(k) + \mathbf{W}_c(k)(-\mathbf{C}\hat{\mathbf{x}}^{*+}(k)) \tag{48}$$

$$\mathbf{P}_c^+(k) = \mathbf{P}^{*+}(k) - \mathbf{W}_c(k)\mathbf{S}_c(k)\mathbf{W}_c^{\mathrm{T}}(k) \tag{49}$$

with

$$\mathbf{W}_c(k) = \mathbf{P}^{*+}(k)\mathbf{C}^{\mathrm{T}}\mathbf{S}_c^{-1}(k) \tag{50}$$

and

$$\mathbf{S}_c(k) = \mathbf{C}\mathbf{P}^{*+}(k)\mathbf{C}^{\mathrm{T}}. \tag{51}$$

The resulting constrained estimates are equivalent to those of eqs. (39) and (40) if the duplicate state estimates are removed from the state vector. See Appendix B for a demonstration of this fact. Ambiguity in the data association during

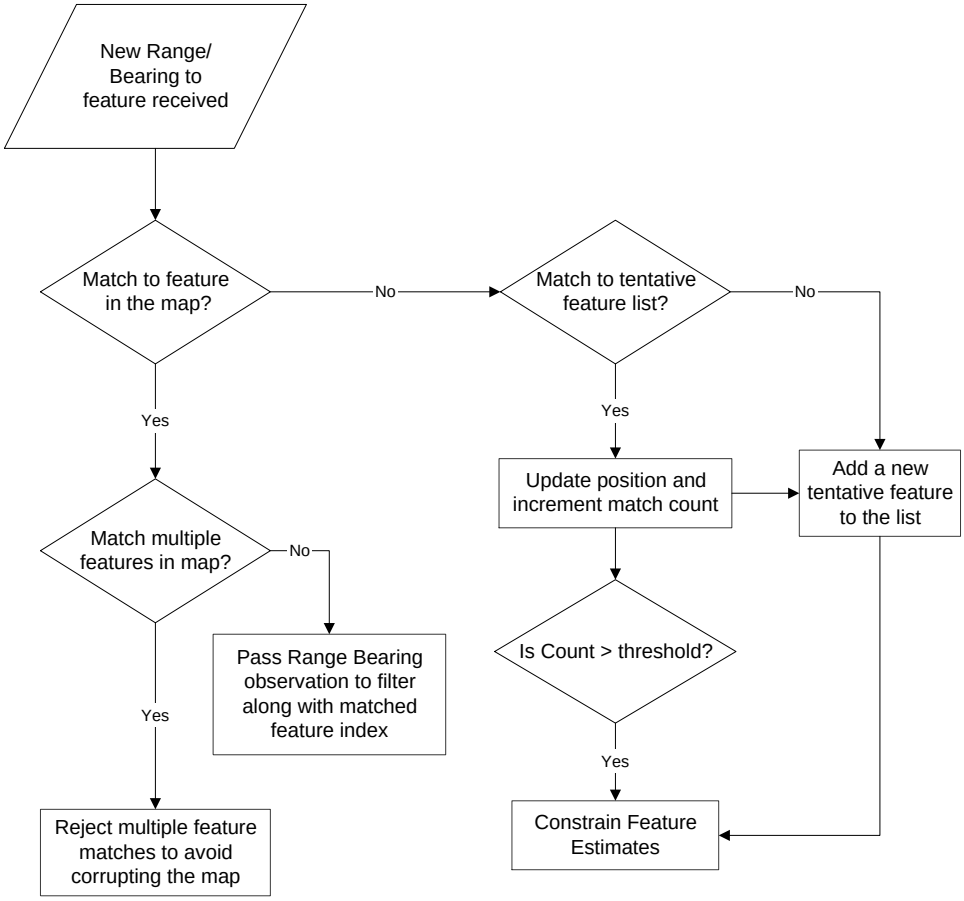


Fig. 3. The constrained feature matching algorithm. As tentative features are identified, they are added to the state vector. When features are confirmed through multiple sightings, these independent estimates are consolidated through the use of constraints.

the initialization phase of the algorithm can therefore be resolved by initializing each observation as a new feature. When a number of observations are found to correspond to a single feature, a constraint can be applied to consolidate all of the observation information in a single estimate of the feature (see Figure 3). The state vector can then be collapsed by removing the redundant feature estimates. Any observations that are not reconfirmed with additional observations can simply be removed from the state vector after some appropriate time period has elapsed.

Ambiguous data associations that may occur at any point in the course of the algorithm can also be resolved using this approach. The results shown here suggest that this ambiguity can be resolved by the introduction of new feature estimates and the application of appropriate constraints.

4. Experimental Results

The goal of any field robotics project should ultimately be the deployment of a vehicle into a natural terrain environment. In this paper we have presented the development of novel techniques designed to improve the performance of the SLAM algorithm. It remains to be shown that these techniques can be adopted in a real world setting.

Current work on undersea vehicles at the Australian Centre for Field Robotics concentrates on the development of terrain-aided navigation techniques. Key elements of this work include sensor fusion, vehicle control architectures for real-time platform control, the development of sonar feature models, the tracking and use of these models in mapping and position estimation, and the development of low-speed platform models for vehicle control. In this section we present the results of the application of the SLAM techniques described in this paper to the deployment of an autonomous underwater vehicle (AUV).

4.1. Oberon: An Underwater Research Platform

The experimental platform used for the work reported in this paper is a mid-size submersible robotic vehicle called Oberon,

Fig. 4. Oberon at sea.

designed and built at the Australian Centre for Field Robotics (see Figure 4). This vehicle is used to demonstrate the methods and algorithms proposed in this paper. The vehicle is equipped with two scanning low-frequency terrain-aiding sonars and a color CCD camera, together with bathyometric depth sensors, a fiber optic gyroscope and a magneto-inductive compass with integrated two-axis tilt sensor (Williams et al. 1999). This vehicle is intended primarily as a research platform on which to test novel sensing strategies and control methods. Autonomous navigation using the information provided by the vehicle's on-board sensors represents one of the ultimate goals of the project (Newman and Durrant-Whyte 1997; Williams 2001).

4.2. System States

In the current implementation, the vehicle pose is made up of the two-dimensional position (x_v, y_v) and orientation ψ_v of the vehicle. A schematic diagram of the vehicle model is shown in Figure 5. Estimates of the vehicle ground speed, V_v, slip angle, γ_v, and the gyro rate bias, $\dot{\psi}_{bias}$, are also generated by the algorithm. The "slip angle", γ_v, is the angle between the vehicle axis and the direction of the velocity vector. Although the thrusters that drive the vehicle are oriented in the direction of the vehicle axis, the slip angle is often non-zero due to disturbances caused by ocean currents, wave effects and the deployed tether.

The landmarks tracked in this implementation of the SLAM algorithm are assumed to be point features. During the trials described in this section, sonar reflectors are deployed in the area in which the vehicle operates. As will be shown, these reflectors present the vehicle with easily identifiable return signatures that can be characterized as point features for mapping purposes.

The augmented state matrix therefore consists of the six states that describe the vehicle and the n_f states that describe the position of the observed landmarks:

$$\begin{aligned}\mathbf{x}(k) &= \begin{bmatrix}\mathbf{x}_v(k)\\ \mathbf{x}_m(k)\end{bmatrix}\\ &= \begin{bmatrix}x_v(k) & y_v(k) & \psi_v(k) & V_v(k) & \gamma_v(k)\\ & \dot{\psi}_{bias}(k) & x_1(k) & y_1(k) & \ldots\end{bmatrix}^{\mathrm{T}}. \end{aligned} \quad (52)$$

4.3. The Vehicle and Landmark Models

In the current implementation of the filter, the vehicle is modeled as a rigid body operating in a 2.5-dimensional world. Depth information is kept separate from the position and orientation of the vehicle. This is not an entirely accurate reflection of the motion of the vehicle in the underwater domain but serves to allow the algorithms to be developed and tested. For many of the missions envisaged for this vehicle, the two-dimensional model is adequate given that the vehicle will generally be operating in close proximity to the sea floor.

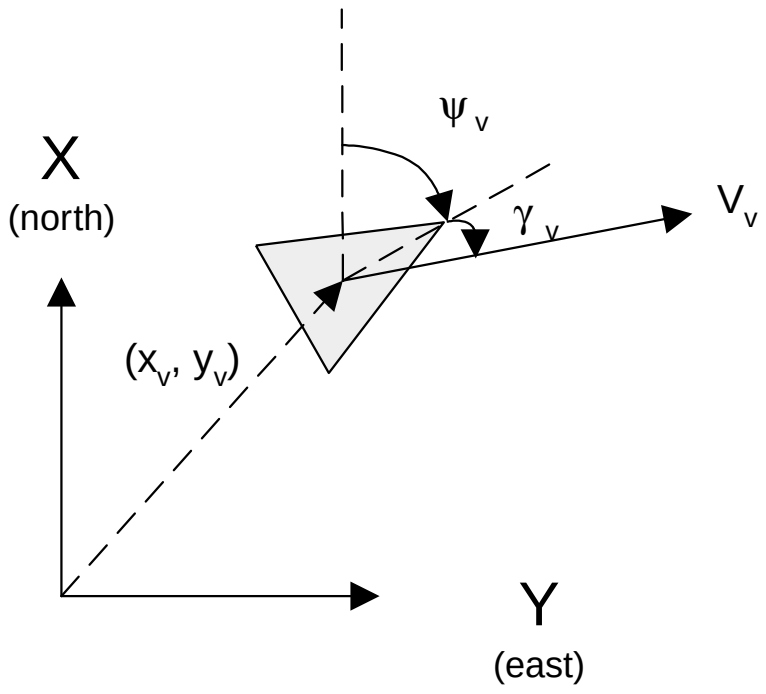


Fig. 5. The vehicle model currently employed with the submersible vehicle. The positioning filter estimates the vehicle position (x_v, x_v), orientation ψ_v, velocity V_v, and slip angle γ_v. The frame of reference used is based on the North-East-Down alignment commonly used in aeronautical engineering applications. The x-axis is aligned with the compass generated North reading.

4.3.1. Vehicle Model

A constant acceleration model, shown in eq. (53), is used for the purpose of estimating the vehicle state transitions:

$$\begin{aligned} \dot{x}_v(t) &= V_v(t)\cos(\psi_v(t)+\gamma_v(t)) + v_x(t) \\ \dot{y}_v(t) &= V_v(t)\sin(\psi_v(t)+\gamma_v(t)) + v_y(t) \\ \dot{\psi}_v(t) &= \dot{\psi}_{gyro}(t) - \dot{\psi}_{bias}(t) + v_\psi(t) \\ \dot{V}_v(t) &= v_V(t) \\ \dot{\gamma}_v(t) &= v_\gamma(t) \\ \dot{\psi}_{bias}(t) &= v_{bias}(t). \end{aligned} \tag{53}$$

Here, v_x, v_y, v_ψ, v_V, v_γ and v_{bias} are assumed to be zero-mean, temporally uncorrelated Gaussian process noise errors with variance σ_x^2, σ_y^2, σ_ψ^2, σ_V^2, σ_γ^2 and σ_{bias}^2, respectively. The standard deviations for these noise parameters are shown in Table 1.

The rate of change of vehicle ground speed, $\dot{V}_v(t)$, and slip angle, $\dot{\gamma}_v(t)$, are assumed to be driven by white noise. The fiber-optic gyroscope measures the vehicle yaw rate and is used as a control input to drive the orientation estimate. Given the small submerged inertia, relatively slow motion and large drag-coefficients induced by the open frame structure of the vehicle and the deployed tether, the model described by eq. (53) is able to capture the motion of the vehicle.

In order to implement the filter, the discrete form of the vehicle model is used to predict the vehicle state $\mathbf{x}_v(k)$ given the previous state $\mathbf{x}_v(k-1)$:

$$\begin{aligned} x_v(k) &= x_v(k-1) + \Delta t(k) V_v(k-1)\cos(\psi_v(k-1) \\ &\quad +\gamma_v(k-1)) \\ y_v(k) &= y_v(k-1) + \Delta t(k) V_v(k-1)\sin(\psi_v(k-1) \\ &\quad +\gamma_v(k-1)) \\ \psi_v(k) &= \psi_v(k-1) + \Delta t(k)(\dot{\psi}_{gyro}(k-1) \\ &\quad -\dot{\psi}_{bias}(k-1)) \\ V_v(k) &= V_v(k-1) \\ \gamma_v(k) &= \gamma_v(k-1) \\ \dot{\psi}_{bias}(k) &= \dot{\psi}_{bias}(k-1). \end{aligned} \tag{54}$$

This defines the discrete, non-linear vehicle prediction equation

$$\mathbf{x}_v(k) = \mathbf{f}_v(\mathbf{x}_v(k-1), \mathbf{u}(k)). \tag{55}$$

The filter parameters used in this application are shown in Table 1.

Table 1. SLAM Filter Parameters

Sampling period	$\Delta t(k)$	0.1 s
Vehicle X process noise std dev	σ_x	0.025 m
Vehicle Y process noise std dev	σ_y	0.025 m
Vehicle heading process noise std dev	σ_ψ	0.6°
Vehicle velocity std dev	σ_v	0.01 m s^{-1}
Vehicle slip angle std dev	σ_γ	1.4°
Gyro bias std dev	σ_{bias}	0.3° s^{-1}
Gyro measurement std dev	σ_{gyro}	0.6° s^{-1}
Compass std dev	$\sigma_{compass}$	2.9°
Range measurement std dev	σ_R	0.1 m
Bearing measurement std dev	σ_B	1.4°
Sonar range		20 m
Sonar resolution		0.1 m

4.3.2. Vehicle Observation Model

There are two types of observations involved in the map building process as implemented on the vehicle. The first is the observation of the orientation from the output of the magneto-inductive compass. The compass observations are assumed to be corrupted by zero-mean, temporally uncorrelated white noise with variance $\sigma_{compass}$:

$$\mathbf{z}_{compass}(k) = \psi(k) + w_{compass}. \tag{56}$$

There is always danger that a compass will be affected by ferrous objects in the environment and transient magnetic fields induced by large electric currents, such as those generated by the vehicle's thrusters. In practice, the compass does not seem to be affected to a significant degree by the vehicle's thrusters. In addition, the unit is equipped with a magnetic field strength alarm. When the strength of the magnetic field increases, the alarm is signaled indicating that the current observation may be in doubt.

Terrain feature observations are made using an imaging sonar that scans the horizontal plane around the vehicle, as shown in Figure 6. Point features are extracted from the sonar scans and are matched against existing features in the map. The feature extraction algorithm is described in more detail in Section 4.5. The observation consists of a relative distance and orientation from the vehicle to the feature. The terrain feature observations are assumed to be corrupted by zero-mean, temporally uncorrelated white noise with variance σ_R and σ_θ, respectively. Given the current vehicle position $\mathbf{x}_v(k)$ and the position of an observed feature $\mathbf{x}_i(k)$, the observation, consisting of range, $z_R(k)$ and bearing, $z_\theta(k)$, can be modeled as

$$\begin{aligned}\mathbf{z}(k) &= \begin{bmatrix} z_R(k) \\ z_\theta(k) \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{(x_v(k)-x_i(k))^2 + (y_v(k)-y_i(k))^2} \\ \arctan(\frac{(y_v(k)-y_i(k))}{(x_v(k)-x_i(k))}) - \psi_v(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} w_r(k) \\ w_\theta(k) \end{bmatrix}.\end{aligned} \tag{57}$$

4.4. The Estimation Process

For the undersea work reported here, an EKF is used to estimate the pose of the vehicle $\hat{\mathbf{x}}_v^+(k)$ along with the positions of the n_f observed features $\hat{\mathbf{x}}_i^+(k)$, $i = 1, ..., n_f$. The augmented state vector therefore consists of all states associated with the vehicle as well as those associated with the observed features

$$\hat{\mathbf{x}}^+(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^+(k) \\ \hat{\mathbf{x}}_1^+(k) \\ \vdots \\ \hat{\mathbf{x}}_{n_f}^+(k) \end{bmatrix}. \tag{58}$$

4.4.1. Prediction

Given the discretized vehicle model shown in eq. (54), the prediction stage for the filter results in

$$\begin{bmatrix} \hat{x}_v^-(k) \\ y_v^-(k) \\ \hat{\psi}_v^-(k) \\ \hat{V}_v^-(k) \\ \hat{\gamma}_v^-(k) \\ \hat{\dot{\psi}}_{bias}^-(k) \end{bmatrix} = \tag{59}$$

$$\begin{bmatrix} \hat{x}_v^+(k-1) + \Delta t(k)\hat{V}_v^+(k-1)\cos(\hat{\psi}_v^+(k-1) \\ \quad +\hat{\gamma}_v^+(k-1)) \\ \hat{y}_v^+(k-1) + \Delta t(k)\hat{V}_v^+(k-1)\sin(\hat{\psi}_v^+(k-1) \\ \quad +\hat{\gamma}_v^+(k-1)) \\ \hat{\psi}_v^+(k-1) + \Delta t(k)(\dot{\hat{\psi}}_{gyro}(k-1) \\ \quad -\hat{\dot{\psi}}_{bias}^+(k-1)) \\ \hat{V}_v^+(k-1) \\ \hat{\gamma}_v^+(k-1) \\ \hat{\dot{\psi}}_{bias}^+(k-1) \end{bmatrix}.$$

As shown in Section 2, the covariance matrix must also be predicted by linearizing about the current state estimates

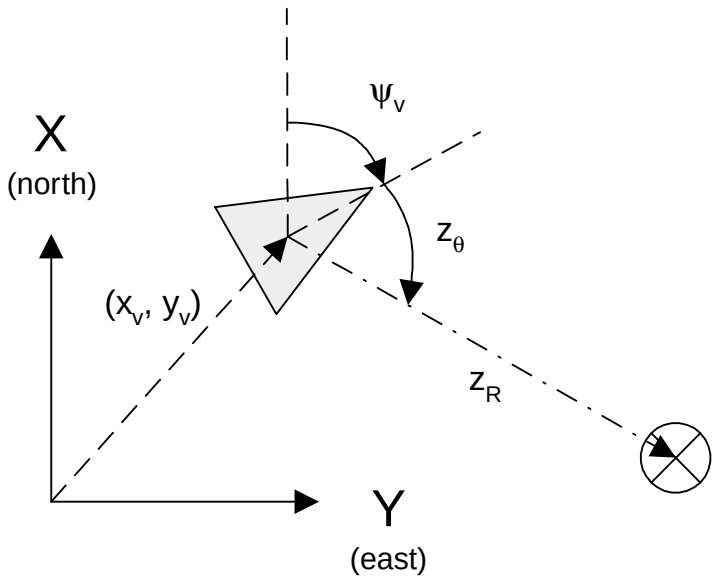


Fig. 6. The vehicle observation model currently employed with the submersible vehicle. The scanning sonar is used to identify point features in the environment. These point features are reported as a range, z_R, and bearing, z_θ, between the vehicle and the observed feature.

$$\mathbf{P}^-_{vv}(k) = \nabla_{\mathbf{x}}\mathbf{f}(k)\mathbf{P}^+_{vv}(k-1)\nabla_{\mathbf{x}}\mathbf{f}(k)^{\mathrm{T}} + \nabla_{\mathbf{u}}\mathbf{f}(k)\mathbf{U}(k)\nabla_{\mathbf{u}}\mathbf{f}(k)^{\mathrm{T}} + \mathbf{Q}(k) \tag{60}$$

where

$$\nabla_{\mathbf{x}}\mathbf{f}(k) = \begin{bmatrix} 1 & 0 & -\Delta t\hat{V}_v \sin(\hat{\psi}_v + \hat{\gamma}_v) & \Delta t \cos(\hat{\psi}_v + \hat{\gamma}_v) \\ 1 & 0 & \Delta t\hat{V}_v \cos(\hat{\psi}_v + \hat{\gamma}_v) & \Delta t \sin(\hat{\psi}_v + \hat{\gamma}_v) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -\Delta t\hat{V}_v \sin(\hat{\psi}_v + \hat{\gamma}_v) & 0 \\ -\Delta t\hat{V}_v \sin(\hat{\psi}_v + \hat{\gamma}_v) & 0 \\ 0 & -\Delta t \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{61}$$

and

$$\nabla_{\mathbf{u}}\mathbf{f}(k) = \begin{bmatrix} 0 \\ 0 \\ \Delta t \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{62}$$

with

$$\mathbf{U}(k) = \mathrm{diag}\left[\sigma^2_{gyro}\right] \tag{63}$$

and

$$\mathbf{Q}(k) = \mathrm{diag}\left[\sigma_x^2 \quad \sigma_y^2 \quad \sigma_\psi^2 \quad \sigma_V^2 \quad \sigma_\gamma^2 \quad \sigma^2_{bias}\right]. \tag{64}$$

The time indices have been omitted from the Jacobians for conciseness.

4.4.2. Observation

The filter generates an estimate of the current yaw of the vehicle by fusing the predicted yaw estimate with the compass output. A shaping state that estimates the yaw rate bias of the gyroscope is also generated. The yaw measurements are incorporated into the SLAM filter using the yaw observation estimate:

$$\hat{\mathbf{z}}^-_{\psi}(k) = \hat{x}^-_{\psi}(k). \tag{65}$$

The predicted terrain feature observation, $\hat{\mathbf{z}}^-_i(k)$, when observing landmark i located at $\mathbf{x}_i(k)$ can be computed using the non-linear observation model $\mathbf{h}_i(\hat{\mathbf{x}}^-_v(k), \hat{\mathbf{x}}^-_i(k))$.

$$\hat{\mathbf{z}}^-_i(k) = \mathbf{h}_i(\hat{\mathbf{x}}^-_v(k), \hat{\mathbf{x}}^-_i(k)) \tag{66}$$

where $\mathbf{h}_i$ is defined by

$$\hat{\mathbf{z}}^-_i(k) = \begin{bmatrix} \sqrt{(\hat{x}^-_v(k) - \hat{x}^-_i(k))^2 + (\hat{y}^-_v(k) - \hat{y}^-_i(k))^2} \\ \arctan\left(\frac{(\hat{y}^-_v(k) - \hat{y}^-_i(k))}{(\hat{x}^-_v(k) - \hat{x}^-_i(k))}\right) - \hat{\psi}^-_v(k) \end{bmatrix}.$$

4.4.3. Update

The estimated states are updated using the usual EKF update equations shown previously in eq. (26).

4.5. Feature Extraction

The development of autonomous map based navigation relies on the ability of the system to extract appropriate and reliable features with which to build maps. Point features are identified from the sonar scans returned by the imaging sonar and are used to build up a map of the environment.

The extraction of point features from the sonar data is essentially a three stage process. The range to the principal return must first be identified in individual pings. This represents the range to the object that has produced the return. The principal returns must then be grouped into clusters. Small, distinct clusters can be identified as point features and the range and bearing to the landmark estimated. Finally, the range and bearing information must be matched against existing features in the map. In this section we provide more details of the feature identification algorithms used to provide observations for the filter.

4.5.1. Sonar Targets

In the current implementation, sonar targets are introduced into the environment in which the vehicle will operate (see Figure 7). These act as identifiable and stable features. Prominent portions of the reef wall and rocky outcrops can also be classified as a point feature. If the naturally occurring point features are stable, they will also be incorporated into the map. The development of techniques to extract terrain aiding information from more complex natural features, such as coral reefs and the natural variations on the sea floor, is an area of active research (Majumder, Scheding, and Durrant-Whyte 2000). The ability to use natural features would allow a submersible to be deployed in a larger range of environments without the need to introduce artificial targets.

The sonar targets produce strong sonar returns that can be characterized as point features for the purposes of mapping (see Figure 9(a)) . The lighter sections in the scan indicate stronger intensity returns. In the scan of Figure 9, two sonar targets are clearly visible. The features extracted by the algorithm are shown in Figure 9(b). More details of the feature extraction algorithms are presented in the following subsections.

4.5.2. Principal Returns

The data returned by the imaging sonar consist of the complete time history of each sonar ping in a discrete set of bins scaled over the desired range. The first task in extracting reliable features is to identify the principal return from the ping data. The principal return is considered to be the start of the maximum energy component of the signal above a certain noise threshold. Figure 8(a) shows a single ping taken from a scan in the field. This return is a reflection from one of the sonar targets and the principal return is clearly visible. The return exhibits very good signal-to-noise ratio, making the extraction of the principal returns relatively straightforward.

At present, the vehicle relies on the sonar targets as its primary source of navigation information. It is therefore paramount for the vehicle to reliably identify returns originating from the sonar targets. Examination of the returns generated by the targets shows that they typically have a large magnitude return concentrated over a very short section of the ping. This differs from returns from other objects in the environment, such as rocks and the reef walls that tend have high-energy returns spread over a much wider section of the ping, as seen in Figure 8(b).

4.5.3. Identification of Point Features

Following the extraction of the principal return from individual pings, these returns are then processed to find regions of constant depth within the scan that can be classified as point features. Sections of the scan are examined to find consecutive pings from which consistent principal return ranges are located. The principal returns are classified as a point feature if the width of the cluster is small enough to be characterized as a point feature and the region is spatially distinct with respect to other returns in the scan (Newman 1999). The bearing to the feature is computed using the center of the distribution of principal returns. The range is taken to be the median range of the selected principal returns.

A scan taken in the field is shown in Figure 9(a). Two targets are clearly visible in the scan along with a section of the reef wall. Figure 9(b) shows the principal returns selected from the scan along with the point features extracted by the algorithm. Both targets are correctly classified as point features while the returns originating from the reef are ignored. Future work concentrates on using the information available from the unstructured natural terrain to aid in navigation.

4.6. Subsea Deployment

The SLAM algorithms have been tested in a natural environment off the coast of Sydney, Australia. The submersible was deployed in a natural inlet with the sonar targets positioned in a straight line at intervals of 10 m. The vehicle controls were set to maintain a constant heading and altitude during the run. Once the vehicle had reached the end of its tether (approximately 50 m) it was turned around and returned along the line of targets. The slope of the inlet in which the vehicle was deployed meant that the depth of the vehicle varied between approximately 1 and 5 m over the course of the run.

4.6.1. Delayed Initialization Filter

Figure 10(a) shows a plot of the final map obtained by the SLAM algorithm. The positions of the sonar features are

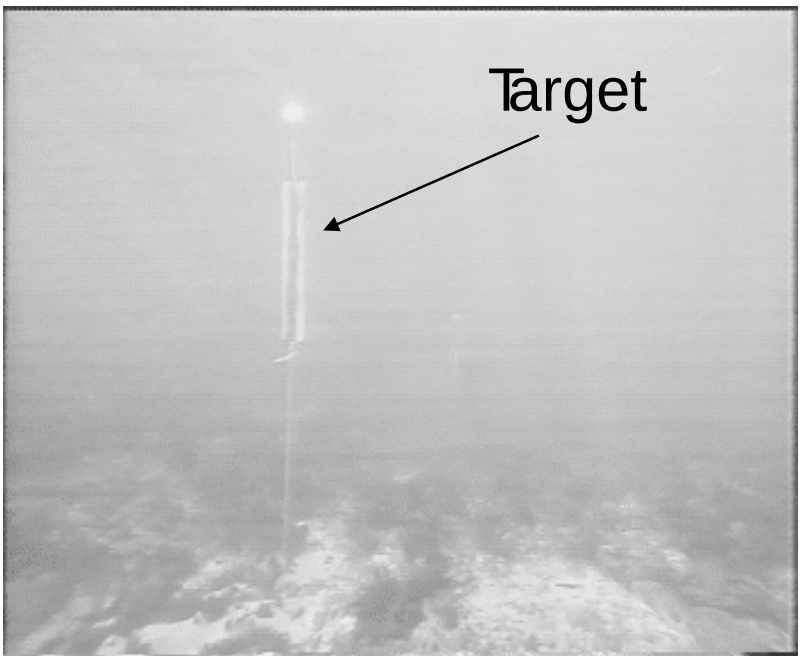


Fig. 7. An image captured from the submersible of one of the sonar targets deployed at the field test site.

clearly visible along with a number of tentative targets that are still not confirmed as sufficiently reliable. Some of the tentative targets are from the reef wall while others come from returns off the tether. These returns are typically not very stable and therefore are not incorporated into the SLAM map. The absolute locations of all the potential point targets identified based on the sonar principal returns are also shown in this map. These locations were computed using the estimated vehicle location at the instant of the corresponding sonar return. The returns seen near the top and bottom of the map are from the reef walls. As can be seen, large clusters of returns have been successfully identified as landmarks.

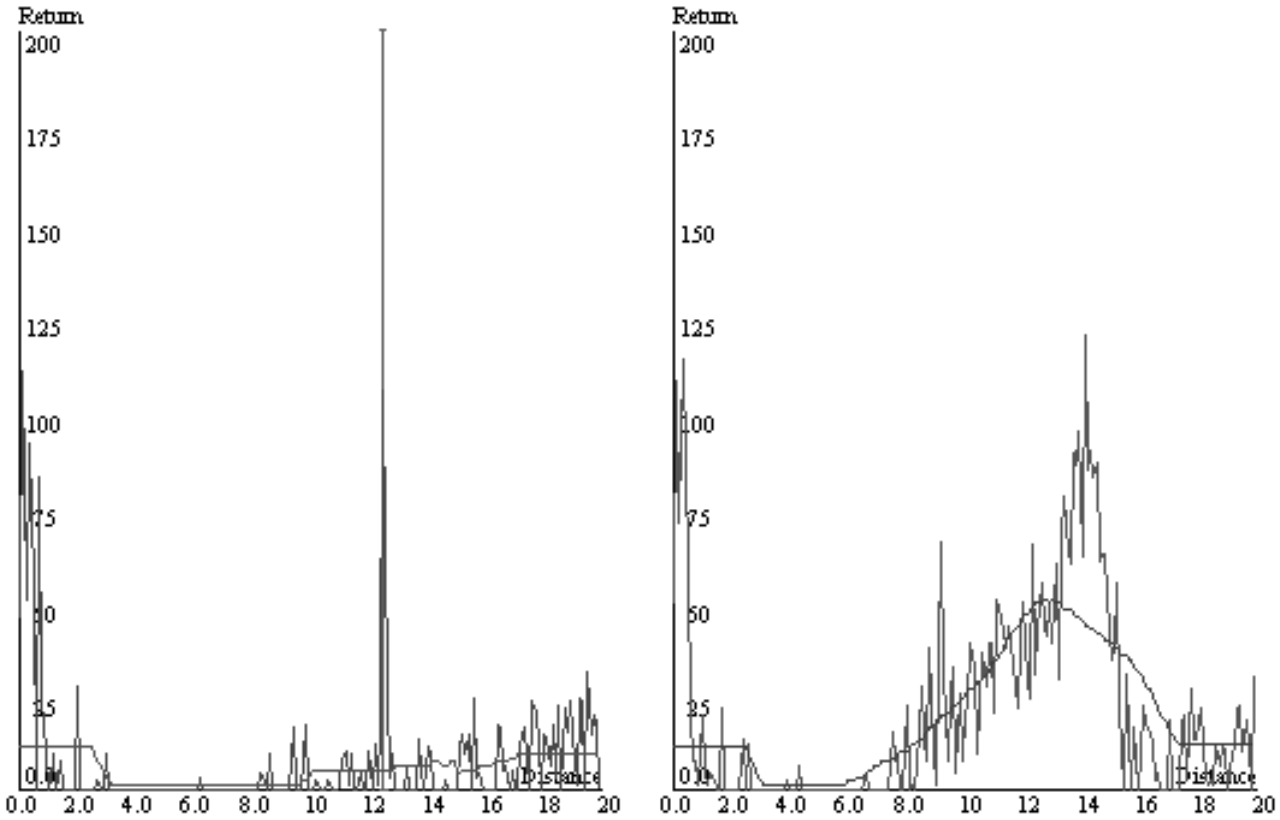
Since there is currently no absolute position sensor on the vehicle, the performance of the positioning filter cannot be measured against ground truth at this time. In previous work, it was shown that the estimator yields consistent results in the controlled environment of the swimming pool at the University of Sydney (Williams et al. 2000). To verify the performance of the filter, the innovation sequence can be monitored to check the consistency of the estimates (Williams, Dissanayake, and Durrant-Whyte 2001).

4.6.2. Constrained Initialization

The novel initialization technique described in Section 3 was used on the same subsea data set presented in Section 4.6.1. The plot of the final map obtained by the SLAM algorithm is reproduced here along with the new map generated using the constrained initialization (see Figure 10(b)). Both the estimated feature locations and the estimated vehicle locations are characterized by smaller covariances than in the original plot. This is due to the rapid rise in uncertainty during the initialization phase of the algorithm. With a poor dead-reckoning model and no sensor to estimate the vehicle velocity, the vehicle covariance grows large prior to the incorporation of the first feature in the map. As mentioned previously, this large uncertainty affects the lower bound achievable for the landmark covariances.

The comparative vehicle position covariance estimates can be examined to verify that the constrained initialization yields tighter covariance bounds. Figure 11 shows the final covariance estimates generated by the constrained initialization algorithm. Notice that the vehicle y position variance is considerably smaller for the constrained initialization case. This is due to the high initial uncertainty associated with the vehicle velocity. Since the vehicle does not have a velocity sensor at this time, it must rely exclusively on observations of the target positions in order to produce an estimate of vehicle velocity. This requires a large initial uncertainty in this parameter in order to allow the filter to converge to an estimate of vehicle velocity. This large uncertainty in turn translates into a large growth of uncertainty in the vehicle position in the early part of the run.

It is also interesting to examine the landmark covariances. These are shown in Figure 12. Once again, the large initial vehicle velocity uncertainty has affected the steady-state covariance of the landmark estimates more prominently in the y-direction. The constrained initialization overcomes this to



(a) Feature Ping (b) Terrain Ping

Fig. 8. (a) A single sonar ping showing the raw ping, the moving average and the computed principal return. This ping is a reflection from one of the sonar targets and shows very good signal-to-noise ratio. The dashed line marks the principal return. (b) A single sonar ping reflected from the reef surrounding the vehicle showing the raw ping and the moving average. The terrain returns are distinguishable from the target returns by the fact that the high-energy returns are spread over a much wider section of the ping. The large amplitude return at low range in this ping results from the interface between the oil-filled sonar transducer housing and the surrounding sea water. Large amplitude returns are ignored if they are below 2.0 m from the vehicle.

some extent by using the early feature observations to generate the initial estimate of vehicle position and velocity.

5. Conclusions

In this paper we have proposed a novel feature initialization scheme designed to improve the performance of the SLAM algorithm. Rather than discarding observations of tentative features in the environment, this initialization scheme incorporates them into the filter. When a feature is confirmed through multiple sightings, the information is consolidated into a single estimate of the feature through the application of appropriately formulated constraints. We present results of the application of this technique to data collected using a submersible vehicle.

Appendix A: Constraints

A.1. Linear Constraints

A set of m linear constraints on a random vector $\mathbf{x}(k)$ can be written as

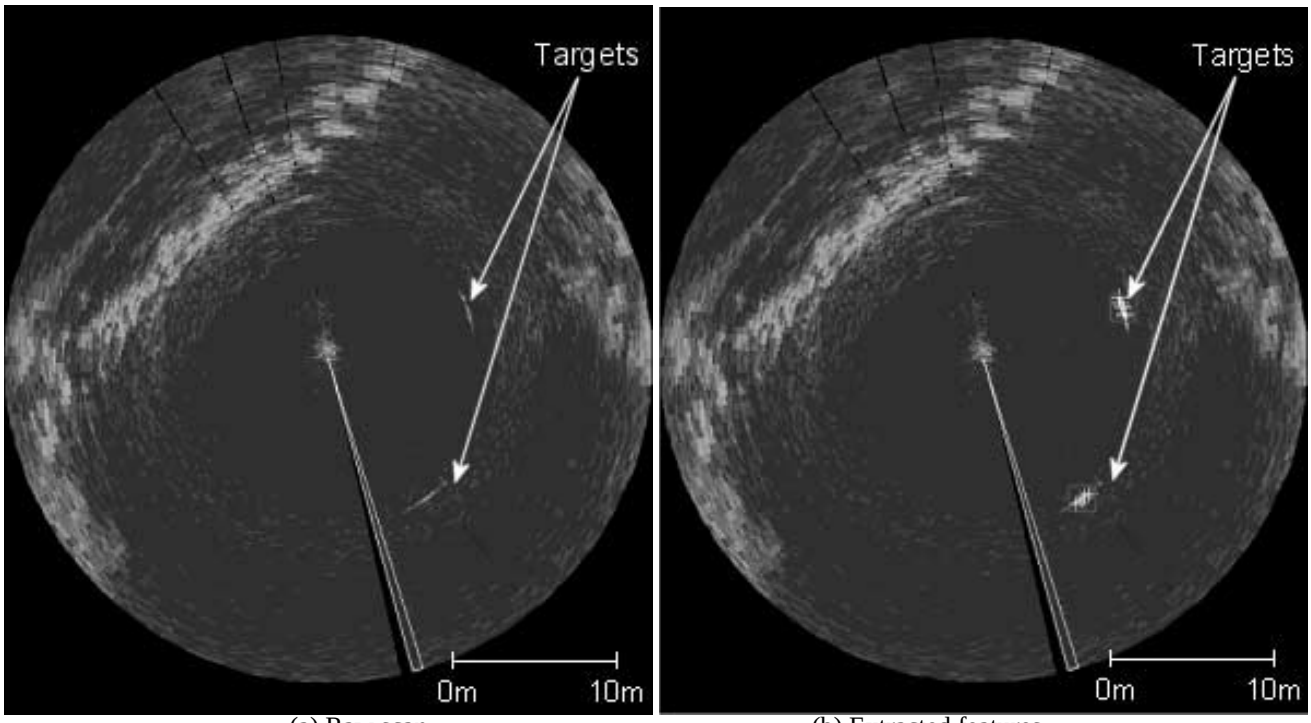
$$\mathbf{C}\mathbf{x}(k) = \mathbf{b} \tag{67}$$

where $\mathbf{C}$ is an $m \times n$ constraint matrix and $\mathbf{b}$ is a vector of dimension m.

As shown in Newman (1999), given an estimate $\hat{\mathbf{x}}^+(k)$ with covariance $\mathbf{P}^+(k)$, constraint matrix $\mathbf{C}$ and solution vector $\mathbf{b}$ it is possible to generate the constrained posterior $\hat{\mathbf{x}}_c^+(k)$ with covariance $\mathbf{P}_c^+(k)$ as follows:

$$\hat{\mathbf{x}}_c^+(k) = \hat{\mathbf{x}}^+(k) + \mathbf{W}_c(k)[\mathbf{b} - \mathbf{C}\hat{\mathbf{x}}^+(k)] \tag{68}$$

$$\mathbf{P}_c^+(k) = \mathbf{P}^+(k) - \mathbf{W}_c(k)\mathbf{S}_c(k)\mathbf{W}_c^{\mathrm{T}}(k) \tag{69}$$



(a) Raw scan (b) Extracted features

Fig. 9. Extracting features from sonar scans: (a) a scan in the field showing sonar targets; (b) the principal returns (+) and the extracted point features (□) from the scan in (a).

with

$$\mathbf{W}_c(k) = \mathbf{P}^+(k)\mathbf{C}^{\mathrm{T}}\mathbf{S}_c^{-1}(k) \tag{70}$$

and

$$\mathbf{S}_c(k) = \mathbf{C}\mathbf{P}^+(k)\mathbf{C}^{\mathrm{T}}. \tag{71}$$

This operation can be considered a weighted projection of the estimates onto the space spanned by the constraints. The weighting factors are functions of the variance of the prior estimates.

A.2. Non-linear Constraints

For the case of non-linear constraints, the m constraint equations become

$$\mathbf{C}(\hat{\mathbf{x}}^+(k)) = \mathbf{b}. \tag{72}$$

A first-order approximation to the solution of this system of constraints can be derived in a similar manner to that used for the EKF. This results in the following non-linear constrained estimate:

$$\hat{\mathbf{x}}_c^+(k) = \hat{\mathbf{x}}^+(k) + \mathbf{W}_c[\mathbf{b} - \mathbf{C}(\hat{\mathbf{x}}^+(k))] \tag{73}$$

$$\mathbf{P}_c^+(k) = \mathbf{P}^+(k) - \mathbf{W}_c(k)\mathbf{S}_c(k)\mathbf{W}_c^{\mathrm{T}}(k) \tag{74}$$

with

$$\mathbf{W}_c(k) = \mathbf{P}^+(k)\nabla\mathbf{C}^{\mathrm{T}}\mathbf{S}_c^{-1}(k) \tag{75}$$

and

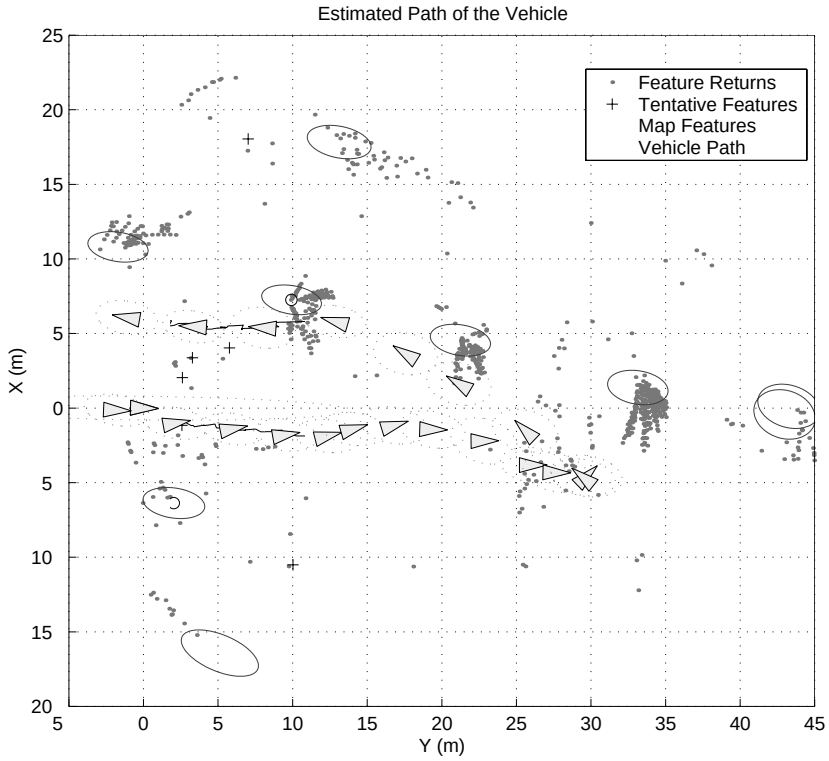
$$\mathbf{S}_c(k) = \nabla\mathbf{C}\mathbf{P}^+(k)\nabla\mathbf{C}^{\mathrm{T}}. \tag{76}$$

Figure 13 shows an example of the projection operation. The original estimate, $\hat{\mathbf{x}}^+(k)$, is projected onto the constraint surface $\mathbf{C}(\hat{\mathbf{x}}^+(k)) = \mathbf{b}$. The dimensionality of the resulting estimate, $\hat{\mathbf{x}}_c^+(k)$ is therefore reduced.

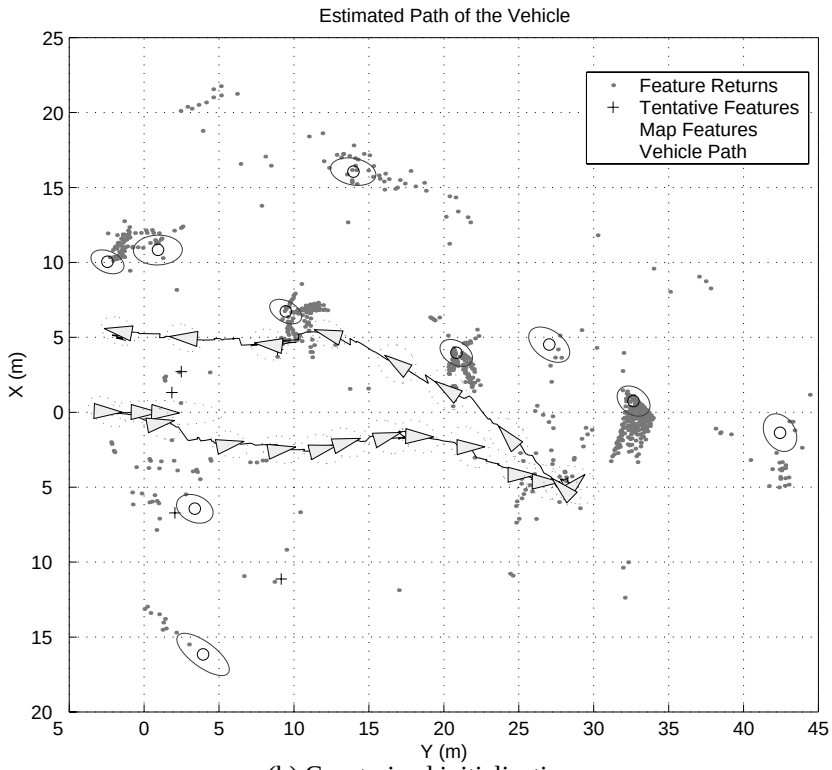
Appendix B: Constrained Initialization

The fact that the constrained update methods presented in this paper are consistent and recover the full state estimate can be verified as follows. Consider a simplified linear case similar to that developed previously. By verifying the consistency of a single step observation in the estimation process, the result for the general case can be inferred.

Consider an estimate of the vehicle and a feature $\mathbf{x}_i$ in some global frame of reference, $\mathcal{F}_G$. For this demonstration, the rest of the map features will not be considered. It is straightforward to extend the demonstration to the case of extra map estimates



(a) Unconstrained initialization



(b) Constrained initialization

Fig. 10. Path of the robot shown against the final map of the environment: (a) the path of the robot and map generated by the tentative feature initialization technique; (b) the path of the robot and resulting map when the constrained initialization is used. It is clear that both the landmark and vehicle estimates have smaller covariances when the constrained initialization routine is used.

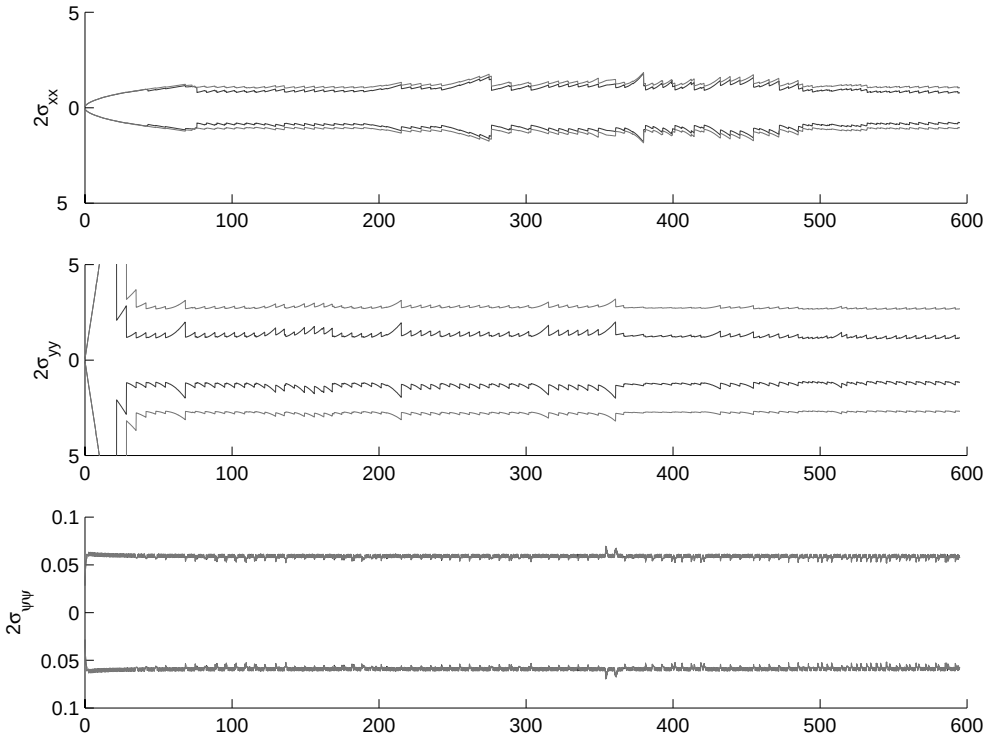


Fig. 11. Comparative vehicle covariances. Notice that the original initialization routine (light) yields a much poorer covariance estimate than the constrained initialization (dark), especially in the case of the vehicle y position. This is due to the large initial uncertainty in velocity that results in a rapid growth in position uncertainty along this axis.

but this results in numerous additional terms that are difficult to show here.

Now the predicted state and covariance estimates are

$$\hat{\mathbf{x}}^-(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_i^-(k) \end{bmatrix} \tag{77}$$

with covariance

$$\mathbf{P}^-(k) = \begin{bmatrix} \mathbf{P}_{vv}^-(k) & \mathbf{P}_{ii}^-(k) \\ \mathbf{P}_{vi}^{-\mathrm{T}}(k) & \mathbf{P}_{ii}^-(k) \end{bmatrix}. \tag{78}$$

Assume that an observation of the feature, $\mathbf{x}_i$, is received. Under normal circumstances, this observation would be fused into the state estimate using the standard Kalman update equations. The linear observation model for this case, $\mathbf{H}(k)$, can be written as

$$\mathbf{H}(k) = \begin{bmatrix} -\mathbf{H}_v(k) & \mathbf{H}_i(k) \end{bmatrix} \tag{79}$$

which reflects the fact that this is a relative observation between the vehicle and the feature.

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{W}(k)\nu(k) \tag{80}$$

with

$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^{\mathrm{T}}(k) \tag{81}$$

where the gain matrix

$$\mathbf{W}(k) = \mathbf{P}^-(k)\mathbf{H}^{\mathrm{T}}(k)\mathbf{S}^{-1}(k), \tag{82}$$

innovation,

$$\nu(k) = \mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}^-(k) \tag{83}$$

and innovation covariance,

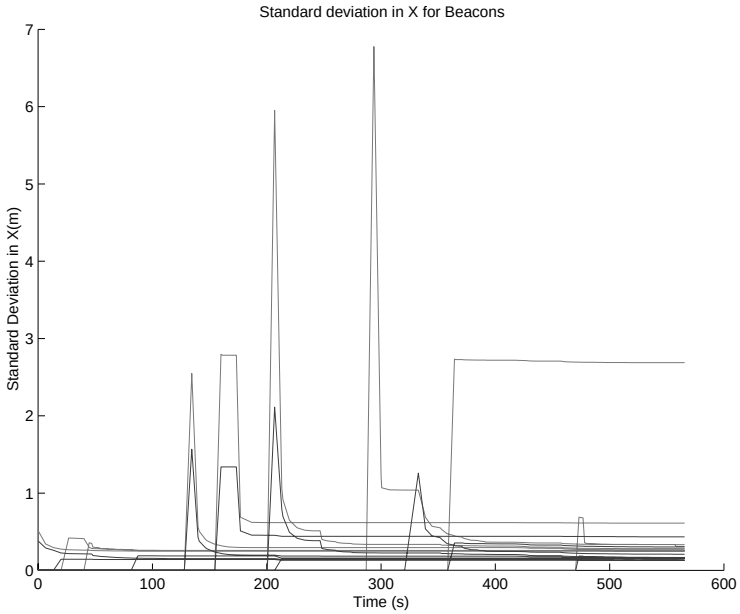
$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}^-(k)\mathbf{H}^{\mathrm{T}}(k) + \mathbf{R}(k), \tag{84}$$

have their usual meaning. This results in the following update of the covariance matrix

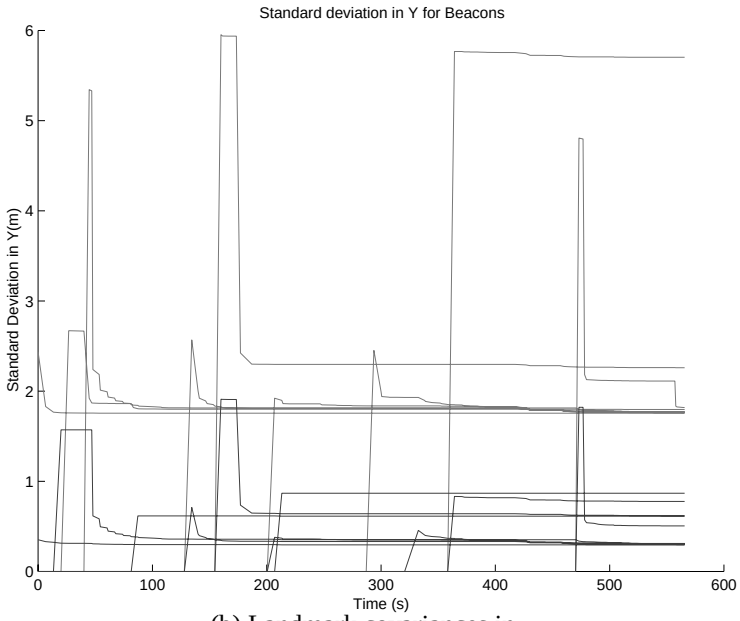
$$\mathbf{P}^+(k) = \mathbf{P}^-(k) - \begin{bmatrix} \mathbf{M}_1\mathbf{S}^{-1}(k)\mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_1\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} \\ \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} \end{bmatrix} \tag{85}$$

with

$$\begin{aligned} \mathbf{M}_1 &= -\mathbf{P}_{vv}^-(k)\mathbf{H}_v^{\mathrm{T}}(k) + \mathbf{P}_{vi}^-(k)\mathbf{H}_i^{\mathrm{T}}(k) \\ \mathbf{M}_2 &= -\mathbf{P}_{vi}^-(k)\mathbf{H}_v^{\mathrm{T}}(k) + \mathbf{P}_{ii}^-(k)\mathbf{H}_i^{\mathrm{T}}(k). \end{aligned}$$



(a) Landmark covariances in x



(b) Landmark covariances in y

Fig. 12. Comparative landmark covariances. Notice that the original initialization routine (light) yields a much poorer covariance estimate than the constrained initialization (dark), especially in the case of the landmark y position estimates. This is due to the large initial uncertainty in vehicle velocity that results in a rapid growth in position uncertainty along this axis. The uncertainty present in the vehicle estimate when the features are initialized affects their steady-state value.

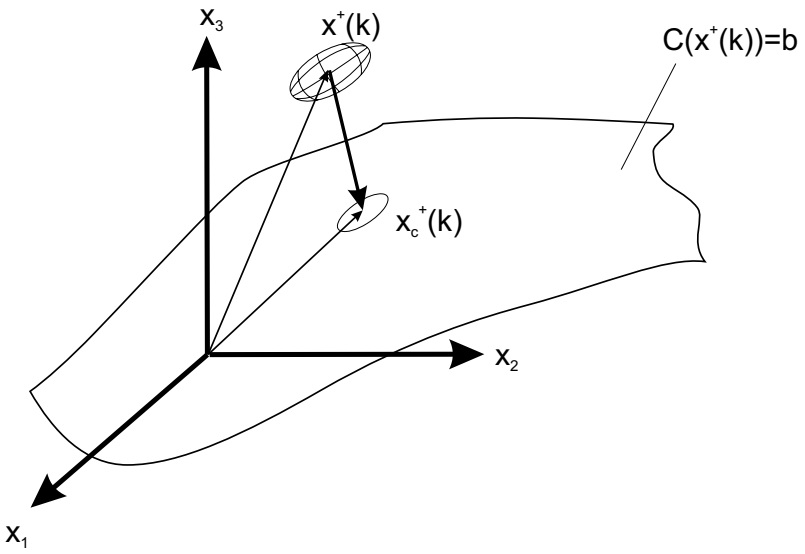


Fig. 13. The application of constraints as a projection operation. The initial estimate is projected onto the constraint surface, thereby reducing its dimensionality while meeting the constraints.

A one-step implementation of the constrained initialization algorithm developed in Section 3 is now shown for comparison. In the case of constrained initialization, the observation of the feature, $\mathbf{x}_i$, will be used to initialize a new estimate of the state. The state vector is first augmented with the observation and the linear observation initialization model is applied to yield the new estimate

$$\hat{\mathbf{x}}^{*+}(k) = \mathbf{G}\hat{\mathbf{x}}^{*-}(k) \tag{86}$$

with

$$\hat{\mathbf{x}}^{*-}(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_i^-(k) \\ \mathbf{z}(k) \end{bmatrix} \tag{87}$$

and

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{H}_i^\dagger \mathbf{H}_v & 0 & \mathbf{H}_i^\dagger \end{bmatrix} \tag{88}$$

where $\mathbf{H}_i^\dagger$ represents the generalized inverse of the landmark observation model.

The covariance matrix is also updated through a similar mechanism

$$\mathbf{P}^{*+}(k) = \mathbf{G}\mathbf{P}^{*-}(k)\mathbf{G}^{\mathrm{T}} \tag{89}$$

with

$$\mathbf{P}^{*-}(k) = \begin{bmatrix} \mathbf{P}_{vv}^-(k) & \mathbf{P}_{vi}^-(k) & 0 \\ \mathbf{P}_{vi}^{-\mathrm{T}}(k) & \mathbf{P}_{ii}^-(k) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix} \tag{90}$$

resulting in the updated augmented covariance matrix

$$\mathbf{P}^{*+}(k) = \begin{bmatrix} \mathbf{P}_{vv}^-(k) & \mathbf{P}_{vi}^-(k) & \mathbf{M}_3^{\mathrm{T}} \\ \mathbf{P}_{vi}^{-\mathrm{T}}(k) & \mathbf{P}_{ii}^-(k) & \mathbf{M}_4^{\mathrm{T}} \\ \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5 \end{bmatrix} \tag{91}$$

with

$$\begin{aligned} \mathbf{M}_3 &= \mathbf{H}_i^\dagger \mathbf{H}_v \mathbf{P}_{vv}^-(k) \\ \mathbf{M}_4 &= \mathbf{H}_i^\dagger \mathbf{H}_v \mathbf{P}_{vi}^-(k) \\ \mathbf{M}_5 &= \mathbf{H}_i^\dagger \mathbf{H}_v \mathbf{P}_{vv}^-(k) \mathbf{H}_v^{\mathrm{T}} \mathbf{H}_i^{\dagger\mathrm{T}} + \mathbf{H}_i^\dagger \mathbf{R}(k) \mathbf{H}_i^{\dagger\mathrm{T}}. \end{aligned}$$

The application of the constraint that the two estimates of the common feature, $\mathbf{x}_i$, are identical can be shown to recover all of the information available to the filter and results in an identical updated covariance matrix to that of the AMF. Using the constraint

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \tag{92}$$

enforces this condition. This results in the constrained covariance update

$$\mathbf{P}_c^+(k) = \mathbf{P}^{*+}(k) - \mathbf{W}_c(k)\mathbf{S}_c(k)\mathbf{W}_c^{\mathrm{T}}(k) \tag{93}$$

with the constraint gain

$$\begin{aligned} \mathbf{W}_c(k) &= \mathbf{P}^{*+}(k)\mathbf{C}^{\mathrm{T}}\mathbf{S}_c^{-1}(k) \\ &= \begin{bmatrix} \mathbf{P}_{vi}^-(k) - \mathbf{M}_3^{\mathrm{T}} \\ \mathbf{P}_{ii}^-(k) - \mathbf{M}_4^{\mathrm{T}} \\ \mathbf{M}_4^{\mathrm{T}} - \mathbf{M}_5^{\mathrm{T}} \end{bmatrix} \mathbf{S}_c^{-1}(k) \end{aligned}$$

and constraint innovation covariance

$$\begin{aligned}\mathbf{S}_c(k) &= \mathbf{C}\mathbf{P}^{*+}(k)\mathbf{C}^{\mathrm{T}} \\ &= \mathbf{P}^{-}_{ii}(k) - \mathbf{M}_4 - \mathbf{M}_4^{\mathrm{T}} + \mathbf{M}_5 \\ &= \mathbf{H}_i^{\dagger}(\mathbf{H}_i\mathbf{P}^{-}_{ii}(k)\mathbf{H}_i^{\mathrm{T}} - \mathbf{H}_i\mathbf{P}^{-}_{vi}(k)\mathbf{H}_v^{\mathrm{T}} - \mathbf{H}_v\mathbf{P}^{-\mathrm{T}}_{vi}(k)\mathbf{H}_i^{\mathrm{T}} \\ &\quad + \mathbf{H}_v\mathbf{P}^{-}_{vv}(k)\mathbf{H}_v^{\mathrm{T}} + \mathbf{R}(k))\mathbf{H}_i^{\dagger\mathrm{T}} \\ &= \mathbf{H}_i^{\dagger}(\mathbf{H}\mathbf{P}^{*+}(k)\mathbf{H}^{\mathrm{T}} + \mathbf{R}(k))\mathbf{H}_i^{\dagger\mathrm{T}}. \end{aligned} \tag{94}$$

Expanding the update term yields

$$\begin{aligned}\mathbf{P}_c^{+}(k) &= \mathbf{P}^{*+}(k) - \mathbf{P}^{*+}(k)\mathbf{C}^{\mathrm{T}}\mathbf{S}_c^{-1}(k)\mathbf{C}\mathbf{P}^{*+}(k) \\ &= \mathbf{P}^{*+}(k) - \begin{bmatrix}\mathbf{P}^{-}_{vi}(k) - \mathbf{M}_3^{\mathrm{T}} \\ \mathbf{P}^{-}_{ii}(k) - \mathbf{M}_4^{\mathrm{T}} \\ \mathbf{M}_4^{\mathrm{T}} - \mathbf{M}_5^{\mathrm{T}}\end{bmatrix}(\mathbf{H}_i^{\dagger}(\mathbf{H}\mathbf{P}^{*+}(k)\mathbf{H}^{\mathrm{T}} \\ &\quad + \mathbf{R}(k))\mathbf{H}_i^{\dagger\mathrm{T}})^{-1}\mathbf{C}\mathbf{P}^{*+}(k) \\ &= \mathbf{P}^{*+}(k) - \begin{bmatrix}\mathbf{P}^{-}_{vi}(k) - \mathbf{P}^{-\mathrm{T}}_{vv}(k)\mathbf{H}_v^{\mathrm{T}}\mathbf{H}_i^{\dagger\mathrm{T}} \\ \mathbf{P}^{-}_{ii}(k) - \mathbf{P}^{-\mathrm{T}}_{vi}(k)\mathbf{H}_v^{\mathrm{T}}\mathbf{H}_i^{\dagger\mathrm{T}} \\ \mathbf{P}^{-\mathrm{T}}_{vi}(k)\mathbf{H}_v^{\mathrm{T}}\mathbf{H}_i^{\dagger\mathrm{T}} - \mathbf{M}_5^{\mathrm{T}}\end{bmatrix}\mathbf{H}_i^{\mathrm{T}} \\ &\quad ((\mathbf{H}\mathbf{P}^{*+}(k)\mathbf{H}^{\mathrm{T}} + \mathbf{R}(k)))^{-1}\mathbf{H}_i\mathbf{C}\mathbf{P}^{*+}(k) \\ &= \mathbf{P}^{*+}(k) - \begin{bmatrix}\mathbf{P}^{-}_{vi}(k)\mathbf{H}_i^{\mathrm{T}} - \mathbf{P}^{-}_{vv}(k)\mathbf{H}_v^{\mathrm{T}} \\ \mathbf{P}^{-}_{ii}(k)\mathbf{H}_i^{\mathrm{T}} - \mathbf{P}^{-}_{vi}(k)\mathbf{H}_v^{\mathrm{T}} \\ \mathbf{P}^{-}_{ii}(k)\mathbf{H}_i^{\mathrm{T}} - \mathbf{P}^{-}_{vi}(k)\mathbf{H}_v^{\mathrm{T}}\end{bmatrix} \\ &\quad ((\mathbf{H}\mathbf{P}^{*+}(k)\mathbf{H}^{\mathrm{T}} + \mathbf{R}(k)))^{-1}\mathbf{C}\mathbf{P}^{*+}(k) \\ &= \mathbf{P}^{*+}(k) \\ &\quad - \begin{bmatrix}\mathbf{M}_1\mathbf{S}^{-1}(k)\mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_1\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} & \mathbf{M}_1\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} \\ \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} & \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} \\ \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_1^{\mathrm{T}} & \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}} & \mathbf{M}_2\mathbf{S}^{-1}(k)\mathbf{M}_2^{\mathrm{T}}\end{bmatrix}. \end{aligned}$$

Eliminating the duplicate estimate of the feature $\mathbf{x}_i$ results in an identical update to the full covariance update.

$$\mathbf{P}_c^{+}(k) = \mathbf{P}^{+}(k). \tag{95}$$

References

Ayache, N., and Faugeras, O. 1989. Maintaining a representation of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation* 5(6):804–819.

Brooks, R. A. 1986. A robust, layered control system for a mobile robot. *IEEE Transactions on Robotics and Automation* 2(1):14–23.

Castellanos, J. A., Montiel, J. M. M., Neira, J., and Tardos, J. D. 1999. The SpMap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation* 15(5):948–952.

Castellanos, J. A., Montiel, J. M. M., Neira, J., and Tardos, J. D. 2000. Sensor influence in the performance of simultaneous mobile robot localization and map building. In P. Corke and J. Trevelyan, eds., *Experimental Robotics IV*, Springer-Verlag, Berlin, pp. 287–296.

Csorba, M. 1997. Simultaneous Localization and Map Building. PhD Thesis, University of Oxford.

Deans, M. C. 2002. Maximally informative statistics for localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1824–1829.

Dissanayake, M. W. M. G., Newman, P., Durrant-Whyte, H. F., Clark, S., and Csorba, M. 2000. An experimental and theoretical investigation into simultaneous localization and map building. *Experimental Robotics IV*, pp. 265–274.

Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csobra, M. June 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation* 17(3):229–241.

Durrant-Whyte, H. F. 2001. *Introduction to Estimation and the Kalman Filter*, Australian Centre for Field Robotics.

Feder, H. J. S. 1999. Simultaneous Stochastic Mapping and Localization. PhD Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering.

Feder, H. J. S., Leonard, J. J., and Smith, C. M. 1999. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research* (Special Issue on Field and Service Robotics) 18(7):650–668.

Gelb, A. 1996. *Applied Optimal Estimation*, 14th edn, MIT Press.

Guivant, J., Nebot, E. M., and Durrant-Whyte, H. F. 2000. Simultaneous localization and map building using natural features in outdoor environments. In *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*, Vol. 1, pp. 581–588.

Gutmann, J. S., and Konolige, K. 2000. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 318–325.

Kuipers, B. J., and Byun, Y. T. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems* 8(1–2):47–63.

Leonard, J. J., and Durrant-Whyte, H. F. 1991. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Vol. 3, pp. 1442–1447.

Leonard, J. J., and Durrant-Whyte, H. F. 1992. *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic, Dordrecht.

Leonard, J. J., and Feder, H. J. S. 1999. A computationally efficient method for large-scale concurrent mapping and localization. In *Proceedings of the 9th International Symposium on Robotics Research*, International Foundation of Robotics Research, pp. 169–176.

Levitt, T. S., and Lawton, D. T. 1990.Qualitative navigation for mobile robots. *Artificial Intelligence Journal* 44(3):305–360.

Lu, F., and Milios, E. 1997. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems* 18(3):249–275.

Majumder, S. 2001. Sensor Fusion and Feature Based Navigation for Subsea Robotics. PhD Thesis, University of Sydney, Australian Centre for Field Robotics.

Majumder, S., Scheding, S., and Durrant-Whyte, H. F. 2000. Sensor fusion and map building for underwater navigation. In *Proceedings of the Australian Conference on Robotics and Automation*, Australian Robotics Association, pp. 25–30.

Maybeck, P. 1982. *Stochastic Models Estimation and Control*, Vol. 1, Academic, New York.

Miller, K. S., and Leskiw, D. M. 1987. *An Introduction to Kalman Filtering with Applications*, 1st edn., Krieger Publishing Company.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada.

Neira, J., and Tardos, J. 2001. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation* 17(6):890–897.

Newman, P. 1999. On The Structure and Solution of the Simultaneous Localization and Map Building Problem. PhD Thesis, University of Sydney, Australian Centre for Field Robotics.

Newman, P., and Durrant-Whyte, H. F. 1997. Toward terrain-aided navigation of a subsea vehicle. In A. Zelinsky, ed., *Field and Service Robotics*, Springer-Verlag, Berlin, pp. 231–236.

Rencken, W. D. 1993. Concurrent localization and map building for mobile robots using ultrasonic sensors. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Vol. 3, pp. 2192–2197.

Rikoski, R. J., Leonard, J. J., and Newman, P. M. 2002. Stochastic mapping frameworks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 426–433.

Smith, R., Self, M., and Cheeseman, P. 1987. A stochastic map for uncertain spatial relationships. *Autonomous Mobile Robots: Perception, Mapping and Navigation* 1:323–330.

Smith, R., Self, M., and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. In G. T. Wilfong and I. J. Cox, eds., *Autonomous Robot Vehicles*, Springer-Verlag, Berlin, pp. 167–193.

Smith, C., Feder, H., and Leonard, J. 1998. Multiple target tracking with navigation uncertainty. In *Proceedings of the 37th IEEE International Conference on Decision and Control*, Vol. 1, pp. 760–761.

Thrun, S., Fox, D., and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31(1–3):29–53 and *Autonomous Robots* 5(3–4):253–271 (joint issue).

Williams, S. B. 2001. Efficient Solutions to Autonomous Mapping and Navigation Problems. PhD Thesis, University of Sydney, Australian Centre for Field Robotics.

Williams, S. B., Newman, P., Majumder, S., Rosenblatt, J., and Durrant-Whyte, H. F. 1999. Autonomous transect surveying of the great barrier reef. In *Proceedings of the Australian Conference on Robotics and Automation*, Australian Robotics Association, pp. 16–20.

Williams, S. B., Newman, P., Dissanayake, M. W. M. G., and Durrant-Whyte, H. F. 2000. Autonomous underwater simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1793–1798.

Williams, S. B., Dissanayake, M. W. M. G., and Durrant-Whyte, H. F. 2001. Towards terrain-aided navigation for underwater robotics. *Advanced Robotics* 15(5):533–550.

Williams, S. B., Dissanayake, M. W. M. G., and Durrant-Whyte, H. F. 2002. An efficient approach to the simultaneous localization and mapping problem. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 406–411.