

# **Mobile Robot Map Building from an Advanced Sonar Array and Accurate Odometry<sup>1</sup>**

Kok Seng CHONG

Institute of Microelectronics

11 Science Park Road

Singapore Science Park II, Singapore 117685

Lindsay KLEEMAN

Intelligent Robotics Research Centre (IRRC)

Dept of Electrical and Computer System Eng.

Monash University, Victoria 3168 Australia

## ***Abstract***

*This paper describes a mobile robot equipped with a sonar sensor array in a guided feature based map building task in an indoor environment. The landmarks common to indoor environments are planes, corners and edges, and these are located and classified with the sonar sensor array. The map building process makes use of accurate odometry information that is derived from a pair of knife edged unloaded encoder wheels. Discrete sonar observations are incrementally merged into partial planes to produce a realistic representation of environment that is amenable to sonar localisation. Collinearity constraints among features are exploited to enhance both the map feature estimation and robot localisation. The map update employs an Iterated Extended Kalman Filter (IEKF) in the first implementation and subsequently a comparison is made with the Julier-Uhlmann-Durrant-Whyte Kalman Filter (JUDKF) which improves the accuracy of covariance propagation when non-linear equations are involved. The map accounts for correlation among features and robot positions. Partial planes are also used to eliminate phantom targets caused by specular reflection of the sonar. Unclassifiable sonar targets are integrated into the map for the purpose of obstacle avoidance. The paper presents simulated and experimental data.*

---

<sup>1</sup> This work was performed whilst K S Chong was at Monash University as a PhD student.

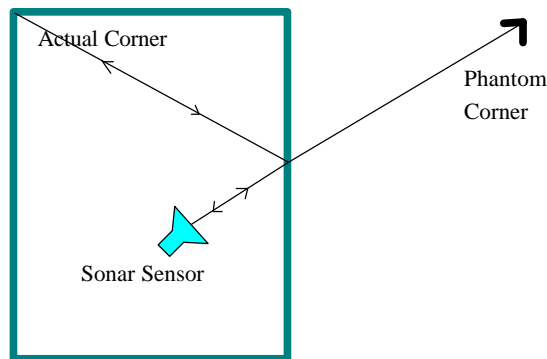
## **1. Introduction**

The objective of this work is to implement an autonomous mobile robot capable of navigating in an *a priori* unknown indoor environment using a sonar sensor. To this end, the robot requires the capability to build a map of the environment, which is a cyclic process of moving to a new position, sensing the environment, updating the map and planning subsequent motion. Map building and navigation is a complex problem because map integrity cannot be sustained by odometry alone due to errors introduced by wheel slippage and distortion. Exteroceptive sensing, such as sonar sensing as employed in this paper, is necessary, but any sensing is also subject to random errors. Hence, neither odometry nor matching sensory data to the map gives flawless estimation of the robot's position, yet this position estimate becomes a reference for the integration of new features in the map. Consequently, with time, errors in robot position influence errors in the map and map errors influence the position estimation.

This paper employs sonar sensing in the map building process for many reasons. Sonar has the property that the data is sparse and naturally selects useful landmarks, such as walls, wall mouldings and corners. This alleviates the data processing compared to dense ranging devices such as laser range finders and stereo vision systems. Sonar also offers a high degree of ranging and bearing accuracy in an array configuration as deployed in this paper. Since most robots today employ some form of sonar due to the cost and power consumption advantages, there is considerable interest in its application.

Sonar sensing has some important properties that need to be carefully understood in order to properly exploit the sensing data. Firstly, sonar transducers have a significant angular spread of energy known as the beamwidth. In many systems, the beamwidth

gives rise to large angular uncertainty in measurement. Some researchers have attempted to deal with this uncertainty by employing grid based maps and repetitive measurements, as in the work by (Moravec and Elfes 1985). Grid map update schemes range from Bayesian (Lim and Cho 1993), evidential (Pagac, Nebot, and Durrant-Whyte 1996) to fuzzy (Oriolo, Vendittelli and Ulivi 1995) and rely on viewing targets from many locations. Localisation with a grid map can be complex. A grid map based localisation scheme has been developed in (Gonzalez 1992), but this work is suitable for laser rangefinder systems only. Other researchers do not consider localisation necessary in their applications (Dudek, Freedman and Rekleitis 1996; Ohya, Nagashima, and Yuta 1995). Feature-based mapping schemes have become more commonplace (Leonard and Durrant-Whyte 1991, Rencken 1993) after Kuc and Siegel (Kuc and Siegel 1987) presented a method for discriminating planes, corners and edges using sonar data gathered at two positions. Two significant follow-ups to (Kuc and Siegel 1987) include the work by Barshan and Kuc (Barshan and Kuc 1990), which differentiates planes and corners with multiple transducers at a single position, and the work by Bozma and Kuc (Bozma and Kuc 1991), which differentiates planes and corners with one transducer at two positions. Later (Kleeman and Kuc 1995) developed a sonar sensor which allows target discrimination at one position and target localisation with high precision. (Hong and Kleeman 1995) have successfully demonstrated the localisation capability of a mobile robot with a sonar array in a known environment using 3D features. Data fusion methods associated with feature based mapping include the Kalman Filter (Ayache and Faugeras 1989; Moutarlier and Chatila 1989; Smith and Cheeseman 1986), maximum likelihood estimation (Lu and Milios 1995), evidential reasoning (Pagac, Nebot and Durrant-Whyte 1996), and heuristic rules (Dudek, Freedman and Rekleitis 1996).



**Figure 1 : Phantom Target Example**

The second important property of sonar systems is the appearance of phantom targets that are due to multiple specular reflections. For example, a sonar sensor will see a virtual image of a corner due to the reflection from the wall in the outwards and return paths in Figure 1. A credibility count (Leonard and Durrant-Whyte 1991) has been used to identify these phantom targets, however this approach fails when the phantom target appears consistently from different positions as is the case in the example of Figure 1. A physically based solution is presented in this paper. The third important property of sonar systems is that, when sensing a planar wall, the sensor can only see the part of the wall which is orthogonal to the line of sight - like phantom targets, this property results from specular reflection. Therefore, if the robot navigates along a wall, the robot sees the wall not as an entity but as a set of discrete, approximately collinear planar elements. Postulates need to be made about the relationships between various sonar features during map matching. Furthermore, to reduce the risk of wrongly associating two features, the robot has to be refrained from moving a long distance between successive scanning points during map building.

In the authors' opinion, prolonged navigation can best be achieved when map feature errors are systematically generated from the sensor and odometry errors. It is convenient, and usually justifiable, to assume that errors are approximately Gaussian in their distribution, and to represent the errors with covariance matrices, since robust noise filtering tools use this form of representation. The authors employ the Kalman Filter because the basis of the Kalman Filter is the Bayesian formula and the principle of minimum mean square error (Bar-Shalom and Li 1993) that are well understood and physically acceptable. The Kalman Filter reduces the uncertainties of the parameters of interest by weighting the initial estimation of errors with the errors associated with the new information (known as *observations*) about the parameters. In the context of sonar map building, the parameters to be estimated are the robot's position and the features in the map. The observations are the postulates about the relational constraints among the new features and the existing features. The Kalman Filter makes available knowledge about the uncertainties of map features that is important for path planning that avoids obstacles and localisation within the map.

The Kalman Filter is based on a linear system model. To overcome this limitation, the Extended Kalman Filter (EKF) can be employed and is founded on the assumption that for small noise, first order linearisation of the system model is sufficient for propagating the noise covariance. The problem with discarding higher order terms is that bias can accumulate after repeated estimation. Two approaches have been proposed to deal with this bias: The first approach, called the Iterated Extended Kalman Filter (IEKF), iteratively estimates the parameters of interest by repetitively linearising the system equations about the new estimates until little change results. The second approach, which will be referred to as the Julier-Uhlmann-Durrant-Whyte method (JUDKF) (Julier,

Uhlmann and Durrant-Whyte 1995), is based on generating a set of data points using the error covariance of the input parameters, propagating the data points and computing the resulting error covariance, thus obviating the need to manually evaluate various Jacobian matrices. This paper compares the accuracy of the two methods.

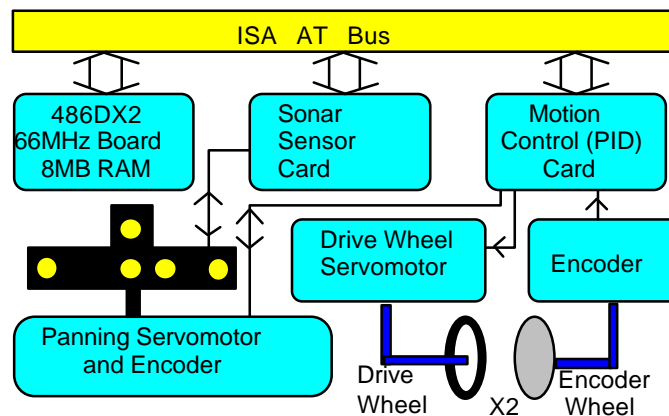
The mapping strategy presented here is feature based and has the following attributes:

1. All three types of primitive features recognisable by our advanced sonar sensor are processed to become part of a map: Discrete planar and corner elements gathered by the sonar sensor at various stages are merged incrementally to form *partial planes*. Planar elements are only merged to the adjacent partial planes to avoid falsely closing a gap, such as a doorway. Discrete edge elements do not partake in the process of forming partial planes, but they are still used to enhance localisation accuracy and map integrity.
2. Not only does ‘plane to plane’, ‘corner to corner’ and ‘edge to edge’ matching occur as in other approaches (Leonard and Durrant-Whyte 1991; Kuc and Siegel 1987), but the relational constraint between a corner and two intersecting planes is exploited to further improve the fidelity of map. Relational constraints are described in (Ayache and Faugeras 1989) and are used by (Neira et al 1996) for a known environment.
3. The partial planes are used to distinguish and subsequently eliminate phantom corner targets and edge targets caused by specular reflection.
4. Two implementations based on the two filters, JUDKF and IEKF, are used to evaluate state transition equations, generate state-measurement cross covariance and propagate error covariance matrices. The two approaches are compared.

This paper is organised as follows. The robot processing, locomotion, odometry and

sonar sensor are described in Section 2. Section 3 presents a summary of the IEKF and JUDKF filters. In section 4 the map environmental model is presented and formulated as a statistical optimisation problem that is solvable with a Kalman Filter. Two sets of equations, one for the IEKF and one for the JUDKF, are derived. The results of four experiments are shown in Section 5. Finally the conclusion summarises the mapping technique and the findings of the comparison between the IEKF and JUDKF filters and also presents future directions for the research.

## 2. Robot Architecture

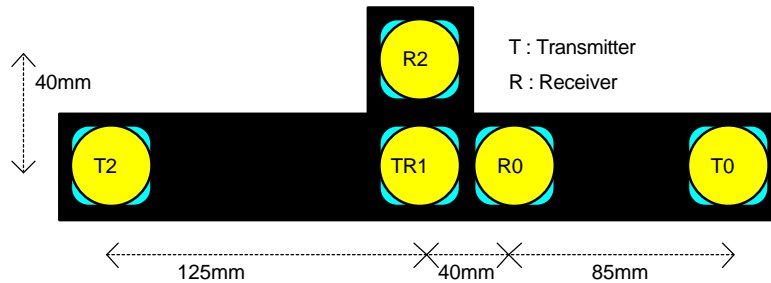


**Figure 2 : The robot system architecture**

As shown in Figure 2, the communication backbone of the robot is an ISA AT Bus with a 486DX2-66MHz processor board controlling a custom sonar sensor card and a custom servo motion control card. The sensor control card sends transmit pulses and captures entire echoes from three receiving transducers. The transmit pulse is generated from a  $10\ \mu\text{s}$  300 to 0 to 300 Volt pulse and the echo waveform is sampled with a 12 bit ADC at 1 MHz. The motion control card contains a MC1401 chip which provides PID control to the four DC motors, two in the pan tilt mechanism and two for the drive wheels. For

every motor, an encoder is mounted on the actuation shaft (*ie* after a gear box) to generate feedback information that is not corrupted by backlash in the gearbox.

## 2.1. The Sonar Array



**Figure 3: The sonar array configuration**

The sonar array illustrated in

Figure 3 has a multiple transducer configuration which makes it possible to classify common indoor features into planes,  $90^\circ$  concave corners and edges. The sonar array accurately estimates specular target ranges to within 0.2 mm and elevation and azimuth angles to within  $0.02^\circ$  for ranges to 5 meters within the sensor beamwidth (Kleeman and Kuc 1995). At every scanning point, the sensor first simultaneously fires TR1 while scouting anticlockwise at  $90^\circ$  per second to locate the directions of potential targets from the echoes on the three receivers. Then, it pans clockwise at the same speed, only slowing down at the directions of the potential targets found earlier and fires T0 followed by T2. If classification is unsuccessful, the target is tagged as unknown but range and bearing are still recorded to unknown objects.



## 2.2. The Locomotion and Odometry System

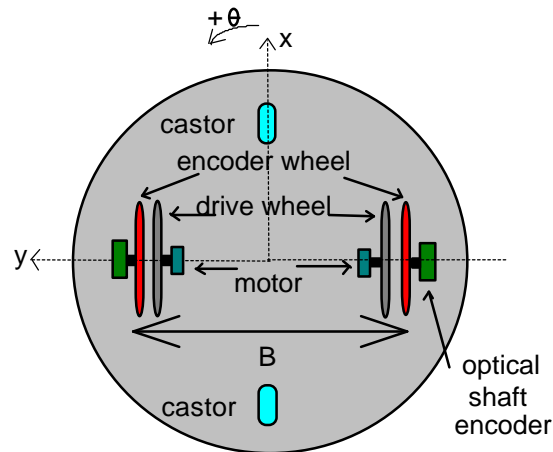


Figure 4 : The odometry system

The locomotion and odometry system shown in Figure 4 consists of drive wheels and separate encoder wheels that generate odometry measurements from optical shaft encoders. The encoder wheels are made with O-rings contacting the floor so as to be as sharp-edged as practically possible to reduce wheelbase uncertainty, and are independently mounted on linear bearings to allow vertical motion, and hence minimise problems of wheel distortion and slippage. This design greatly improves the reliability of odometry measurements. The odometry error model used to propagate error covariance and odometry benchmarking can be found in (Chong and Kleeman 1997).

### 3. Summary of the Iterated Extended Kalman Filter (IEKF) and the Julier-Uhlmann-Durrant-Whyte Kalman Filter (JUDKF)

The section begins by introducing Kalman Filter in a general context. Before proceeding, the notation used will be explained. A circumflex above a random variable,  $\hat{\mathbf{S}}(k+1)$ , is used to indicate the *estimator* of the random variable, whereas a bar over a random

variable,  $\overline{\mathbf{S}(k+1)}$ , is used to indicate the *mean* of the random variable. The partial derivative operator is denoted by  $\nabla$  and is defined by (1).

$$\nabla_{\mathbf{x}} = \frac{\mathcal{I}}{\mathcal{I}\mathbf{x}} = \left[ \begin{array}{cccc} \frac{\mathcal{I}}{\mathcal{I}x_1} & \frac{\mathcal{I}}{\mathcal{I}x_2} & \cdots & \frac{\mathcal{I}}{\mathcal{I}x_n} \end{array} \right] \text{ where } \mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n] \quad (1)$$

Suppose the state vector  $\mathbf{S}(k)$  contains all the randomly distributed parameters of interest which evolve with discrete time according to the state transition equation

$$\mathbf{S}(k+1) = \mathbf{F}(\mathbf{S}(k), \mathbf{U}(k+1)) \quad (2)$$

where  $\mathbf{U}(k)$  is the input vector. At stage  $k+1$ , these random parameters can be *observed* with a set of measurements  $\mathbf{M}(k)$  via the observation model

$$\mathbf{G}(\mathbf{S}(k+1), \mathbf{M}(k+1)) = \mathbf{0} \quad (3)$$

The estimation of  $\mathbf{S}(k+1)$  with equation (2) and (3) is inherently imperfect because of the noise in  $\mathbf{S}(k)$ ,  $\mathbf{U}(k+1)$  and  $\mathbf{M}(k+1)$ . The goal of optimisation is to generate a new state estimate  $\hat{\mathbf{S}}(k+1)$  that minimises the *mean square error* of the parameters  $\mathbf{S}(k+1)$  conditioned on all the past observations which is equal to the mean of the parameters conditioned on all the past observations (Bar-Shalom and Li 1993). Let  $\mathbf{Z}^j$  be all the observations gathered up to stage  $j$ , and  $\hat{\mathbf{S}}(i|j)$  be the minimum mean square error estimate of  $\mathbf{S}(i)$  conditioned on  $\mathbf{Z}^j$ , then

$$\hat{\mathbf{S}}(k+1|k) = E\{\mathbf{F}(\mathbf{S}(k), \mathbf{U}(k+1)|\mathbf{Z}^k)\} \quad (4)$$

and

$$\mathbf{P}_{ss}(k+1|k) = E\left\{\left(\mathbf{S}(k+1) - \hat{\mathbf{S}}(k+1|k)\right)\left(\mathbf{S}(k+1) - \hat{\mathbf{S}}(k+1|k)\right)^T \mid \mathbf{Z}^k\right\} \quad (5)$$

where  $E\{\cdot\}$  is the expectation of a random variable.

A set of measurements  $\mathbf{M}(k+1)$  about  $\mathbf{S}(k+1)$  can also be gathered at stage  $k+1$ . Due to the noise in both  $\mathbf{S}(k+1)$  and  $\mathbf{M}(k+1)$ , equation (3) does not hold exactly. A *residual* vector can be defined as

$$\mathbf{z}(k+1) = -E\{\mathbf{G}(\mathbf{S}(k+1), \mathbf{M}(k+1)|\mathbf{Z}^k)\} \quad (6)$$

With the residual vector, the Kalman Filter can be invoked to generate a better estimate of  $\mathbf{S}(k+1)$ , namely  $\hat{\mathbf{S}}(k+1|k+1)$  based on (Bar-Shalom and Li 1993),

$$\hat{\mathbf{S}}(k+1|k+1) = \hat{\mathbf{S}}(k+1|k) + \mathbf{P}_{sz}(k+1|k)\mathbf{P}_{zz}^{-1}(k+1|k)\mathbf{z}(k+1) \quad (7)$$

and the error covariance is also reduced to

$$\mathbf{P}_{ss}(k+1|k+1) = \mathbf{P}_{ss}(k+1|k) - \mathbf{P}_{sz}(k+1|k)\mathbf{P}_{zz}^{-1}(k+1|k)\mathbf{P}_{sz}^T(k+1|k) \quad (8)$$

Where  $\mathbf{P}_{sz}(k+1|k)$  is the cross-covariance between  $\hat{\mathbf{S}}(k+1|k)$  and  $\mathbf{z}(k+1)$ , and

$\mathbf{P}_{zz}(k+1/k)$  is the covariance of  $\mathbf{z}(k+1)$ , defined in a similar fashion.

In practice, the state transition equation and the observation equation are non-linear. Some methods are required to estimate the covariance and cross-covariance matrices required by Kalman Filter. The IEKF filter and the JUDKF are introduced for this purpose.

### 3.1. The IEKF Method

The IEKF method is an extension of the Extended Kalman Filter (EKF) which is discussed first. With the EKF method,

$$\hat{\mathbf{S}}^{IEKF}(k+1|k) \approx \mathbf{F}(\hat{\mathbf{S}}(k|k), \hat{\mathbf{U}}(k+1)) \quad (9)$$

$$\begin{aligned} \mathbf{z}^{IEKF}(k+1) &\approx -\mathbf{G}(\hat{\mathbf{S}}(k+1|k), \hat{\mathbf{M}}(k+1)) \\ &\approx \nabla_{\mathbf{S}}\mathbf{G}\{\mathbf{S}(k+1|k+1) - \hat{\mathbf{S}}(k+1|k)\} + \nabla_{\mathbf{M}}\mathbf{G}\{\mathbf{M}(k+1) - \hat{\mathbf{M}}(k+1)\} \end{aligned} \quad (10)$$

The noise associated with all random vectors is assumed small, so that applying a first order Taylor's expansion about the estimator is reasonable for propagating the error covariance through non-linear equations. Suppose the error of  $\hat{\mathbf{S}}(k+1|k)$  is not correlated with  $\mathbf{U}(k)$ , then

$$\mathbf{P}_{ss}^{IEKF}(k+1|k) \approx \nabla_{\mathbf{S}}\mathbf{F}\mathbf{P}_{ss}(k|k)\nabla_{\mathbf{S}}\mathbf{F}^T + \nabla_{\mathbf{U}}\mathbf{F}\mathbf{Cov}(\mathbf{U}(k+1))\nabla_{\mathbf{U}}\mathbf{F}^T \quad (11)$$

where  $\nabla\mathbf{F}$  is the Jacobian matrix of  $\mathbf{F}()$  evaluated around  $\mathbf{S}(k|k)$  or  $\mathbf{U}(k+1)$ , as indicated by the subscript.  $\nabla\mathbf{F}$  is also known as the *state transition matrix*.  $\mathbf{Cov}(\mathbf{U}(k+1))$  is the

error covariance of the input vector  $\mathbf{U}(k+1)$ . In a similar manner,

$$\mathbf{P}_{zz}^{IEKF}(k+1|k) \approx \nabla_{\mathbf{S}} \mathbf{G} \mathbf{P}_{ss}(k+1|k) \nabla_{\mathbf{S}} \mathbf{G}^T + \nabla_{\mathbf{M}} \mathbf{G} \mathbf{Cov}(\mathbf{M}(k+1)) \nabla_{\mathbf{M}} \mathbf{G}^T \quad (12)$$

$$\mathbf{P}_{sz}^{IEKF}(k+1|k) \approx \mathbf{P}_{ss}^{IEKF}(k+1|k) \nabla_{\mathbf{S}} \mathbf{G}^T \quad (13)$$

The IEKF method improves the performance of the Extended Kalman Filter by linearising the measurement equation about the new estimate  $\hat{\mathbf{S}}(k+1|k+1)$ , and attempts to iteratively draw  $\hat{\mathbf{S}}(k+1|k+1)$  closer to the true mean, in a way similar to solving a non-linear algebraic equation using Newton Raphson algorithm. Let  $\mathbf{h}_i = \hat{\mathbf{S}}^{IEKF,i}(k+1|k+1)$ , with  $\mathbf{h}_0 = \hat{\mathbf{S}}^{IEKF}(k+1|k)$ , the pseudo code is as follows,

set  $i \leftarrow 0$

repeat {

$$\mathbf{h}_{i+1} = \hat{\mathbf{S}}(k+1|k) + \mathbf{P}_{sz}(k+1|k+1; \mathbf{h}_i) \mathbf{P}_{zz}^{-1}(k+1|k+1; \mathbf{h}_i) \left[ \mathbf{z}(k+1; \mathbf{h}_i) - \nabla_{\mathbf{S}} \mathbf{G}(k+1|k+1; \mathbf{h}_i) (\hat{\mathbf{S}}(k+1|k) - \mathbf{h}_i) \right]$$

$i \leftarrow i+1$

} while  $(|\mathbf{h}_i - \mathbf{h}_{i-1}| > \mathbf{e}_h)$

$$\mathbf{P}_{ss}^{IEKF}(k+1|k+1) = \mathbf{P}_{ss}(k+1|k) - \mathbf{P}_{sz}(k+1|k; \mathbf{h}_i) \mathbf{P}_{zz}^{-1}(k+1|k; \mathbf{h}_i) \mathbf{P}_{sz}^T(k+1|k; \mathbf{h}_i)$$

where the notation ‘ $;\mathbf{h}_i$ ’ means ‘evaluated at the new estimator  $\mathbf{h}_i$ ’, and  $\mathbf{e}_h$  is a threshold vector. Further details on the IEKF can be found in (Bar-Shalom and Li 1993; Jazwinski 1970).

### 3.2. The JUDKF Method

Julier, Uhlmann and Durrant-Whyte (Julier, Uhlmann and Durrant-Whyte 1995) have developed a method for accurately propagating a covariance matrix through non-linear equations while reducing the bias associated with the result. This section summarises and generalises the JUD method in the context of this paper. Examples of how this method can be used with the Kalman Filter (hence the JUDKF) are given at the end of this section.

Suppose that  $\hat{\mathbf{S}}(k|k)$  of size  $n_s \times 1$  is an estimate of a particular random vector  $\mathbf{S}(k)$ , and associated with the estimate is an error covariance matrix  $\mathbf{P}_{ss}(k/k)$  of size  $n_s \times n_s$ , then a set of *sigma points*  $\mathbf{s}_j$  are generated from the  $2n_s$  columns of

$$\pm \sqrt{n_s \mathbf{P}_{ss}(k|k)} = \pm [\mathbf{s}_0 \quad \mathbf{s}_1 \quad \cdots \quad \mathbf{s}_{n_s-1}] = \pm \mathbf{L} \quad (14)$$

where  $\mathbf{L}$  is the lower triangular matrix of the Cholesky Decomposition of  $n_s \mathbf{P}_{ss}(k|k) = \mathbf{L}\mathbf{L}^T$ .

A set of  $2n_s$  data points can be formed,

$$\mathbf{S}_i(k|k) \in \bigcup_{i=0}^{n_s-1} \{ \hat{\mathbf{S}}(k|k) + \mathbf{s}_i, \hat{\mathbf{S}}(k|k) - \mathbf{s}_i \} \quad (15)$$

Let  $\mathbf{X}$  and  $\mathbf{Y}$  be non-linear  $n_x \times 1$  and  $n_y \times 1$  functions of  $\mathbf{S}$ ,

$$\mathbf{X}(k+1) = \mathbf{X}(\mathbf{S}(k)) \quad \mathbf{Y}(k+1) = \mathbf{Y}(\mathbf{S}(k)) \quad (16)$$

The following quantities can be calculated with  $\mathbf{S}_i(k|k)$

$$\hat{\mathbf{X}}(k+1|k) = \frac{1}{2n_s} \sum_{i=0}^{2n_s-1} \mathbf{X}(\mathbf{S}_i(k|k)) \quad (17)$$

$$\hat{\mathbf{Y}}(k+1|k) = \frac{1}{2n_s} \sum_{i=0}^{2n_s-1} \mathbf{Y}(\mathbf{S}_i(k|k)) \quad (18)$$

$$\mathbf{P}_{XX}(k+1|k) = \frac{1}{2n_s} \sum_{i=0}^{2n_s-1} \left[ \mathbf{X}(\mathbf{S}_i(k|k)) - \hat{\mathbf{X}}(k+1|k) \right] \left[ \mathbf{X}(\mathbf{S}_i(k|k)) - \hat{\mathbf{X}}(k+1|k) \right]^T \quad (19)$$

$$\mathbf{P}_{YY}(k+1|k) = \frac{1}{2n_s} \sum_{i=0}^{2n_s-1} \left[ \mathbf{Y}(\mathbf{S}_i(k|k)) - \hat{\mathbf{Y}}(k+1|k) \right] \left[ \mathbf{Y}(\mathbf{S}_i(k|k)) - \hat{\mathbf{Y}}(k+1|k) \right]^T \quad (20)$$

$$\mathbf{P}_{XY}(k+1|k) = \frac{1}{2n_s} \sum_{i=0}^{2n_s-1} \left[ \mathbf{X}(\mathbf{S}_i(k|k)) - \hat{\mathbf{X}}(k+1|k) \right] \left[ \mathbf{Y}(\mathbf{S}_i(k|k)) - \hat{\mathbf{Y}}(k+1|k) \right]^T \quad (21)$$

The equations (19) to (21) for obtaining the covariance and cross-covariance are considered suboptimal (Julier, Uhlmann and Durrant-Whyte 1995) at the expense of ensuring positive (semi)definiteness. To simplify subsequent discussion, the computational details are encapsulated into the following functions:

1.  $\Omega(k, \mathbf{X}(\mathbf{S}), \mathbf{P}_{SS})$  takes the transformation equations,  $\mathbf{X}(\mathbf{S})$  and the covariance matrix  $\mathbf{P}_{SS}$  of the random vector  $\mathbf{S}$  and generates the means of  $\mathbf{X}(\mathbf{S})$ .
2.  $\Lambda(k, \mathbf{X}(\mathbf{S}), \mathbf{Y}(\mathbf{S}), \mathbf{P}_{SS})$  takes the transformation equations,  $\mathbf{X}(\mathbf{S})$  and  $\mathbf{Y}(\mathbf{S})$  and the covariance matrix  $\mathbf{P}_{SS}$  of the random vector  $\mathbf{S}$  and generates the cross-covariance between  $\mathbf{X}(\mathbf{S})$  and  $\mathbf{Y}(\mathbf{S})$ .  $\Lambda(k, \mathbf{X}(\mathbf{S}), \mathbf{P}_{SS})$  returns the covariance of  $\mathbf{X}(\mathbf{S})$ .

In both functions,  $k$  is the stage specifier for all the independent parameters.

The computation of the square root of a matrix is computationally expensive and simplification is desirable. For example, if  $\mathbf{P}_{SS}$  has a diagonal structure, that is,  $\mathbf{P}_{SS} = \text{diag}(\mathbf{P}_i)$ , then

$$\pm \sqrt{n_s \mathbf{P}_{SS}} = \pm \sqrt{n_s} \sqrt{\mathbf{P}_{SS}} = \pm \sqrt{n_s} \text{diag}(\sqrt{\mathbf{P}_i}) \quad (22)$$

The JUD method can now be applied to a Kalman Filtering problem:

$$\hat{\mathbf{S}}^{JU}(k+1|k) \approx \Omega(k|k, \mathbf{F}(\hat{\mathbf{S}}, \hat{\mathbf{U}}), \text{diag}(\mathbf{P}_{ss}, \mathbf{Cov}(\mathbf{U}))) \quad (23)$$

$$\mathbf{z}^{JU}(k+1) \approx -\Omega(k+1|k, \mathbf{G}(\hat{\mathbf{S}}, \hat{\mathbf{U}}), \text{diag}(\mathbf{P}_{ss}, \mathbf{Cov}(\mathbf{M}))) \quad (24)$$

$$\mathbf{P}_{ss}^{JU} \approx \Lambda(k|k, \mathbf{F}(\hat{\mathbf{S}}, \hat{\mathbf{U}}), \text{diag}(\mathbf{P}_{ss}, \mathbf{Cov}(\mathbf{U}))) \quad (25)$$

$$\mathbf{P}_{zz}^{JU} \approx \Lambda(k+1|k, \mathbf{G}(\hat{\mathbf{S}}, \hat{\mathbf{M}}), \text{diag}(\mathbf{P}_{ss}, \mathbf{Cov}(\mathbf{M}))) \quad (26)$$

$$\mathbf{P}_{sz}^{JU} \approx \Lambda(k+1|k, \mathbf{S}, \mathbf{G}(\hat{\mathbf{S}}, \hat{\mathbf{M}}), \text{diag}(\mathbf{P}_{ss}, \mathbf{Cov}(\mathbf{M}))) \quad (27)$$

where  $\text{diag}(\cdot)$  here is the matrix formed by the argument matrices on the diagonal.

#### **4. Map Building Formalism**

This section is further subdivided into nine parts. Section 4.1 contains a discussion of the environment model and pinpoints the parameters to be optimised. Section 4.2 details all mapping scenarios that must be considered in order to grow the map primitives. Section 4.3 presents map building as a statistical optimisation problem and formulates solutions in the context of Kalman Filter. Two formulations are presented: The classical Global approach (Leonard, Durrant-Whyte and Ingemar 1992) and the Relocation-Fusion approach (Moutarlier and Chatila 1989). The authors' formulations bear resemblance to these precursors, but are more general in the sense that they extend beyond feature-to-feature matching in order to tackle the more complex scenarios faced by sonar mapping. Section 4.4 explains why a corner should be merged to two intersecting partial planes, not one. Section 4.5 describes how a collinearity constraint should be validated. Section 4.6 to section 4.9 focus on the discussion for other important map management details, namely,

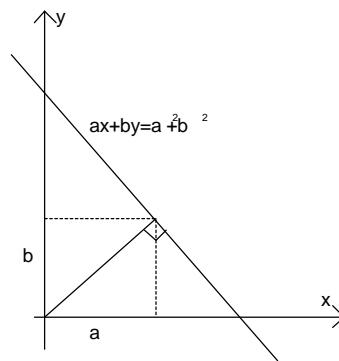


discrimination of phantom targets, incorporation of new measurement, as well as merge and removal of existing primitives.

#### 4.1. Map Primitives

The environmental model comprises two types of primitives:

**Partial Plane** is characterised by its state parameters  $x_i(k) = [a_i(k) \quad b_i(k)]^T$  from the line equation  $ax + by = a^2 + b^2$ , the Cartesian coordinates of its approximate endpoints, and a status associated with each endpoint, indicating whether it is terminated with another partial plane to form a corner. When a wall is first detected, it is registered as a partial plane with only one endpoint. It is then grown to have two endpoints and extended as the robot moves along the wall.



**Figure 5 : Parameterisation of a partial plane**

**Corner** is characterised by its Cartesian coordinates  $x_i(k) = [x_i(k) \quad y_i(k)]^T$  only. The sonar sensor provides no indication of its orientation.

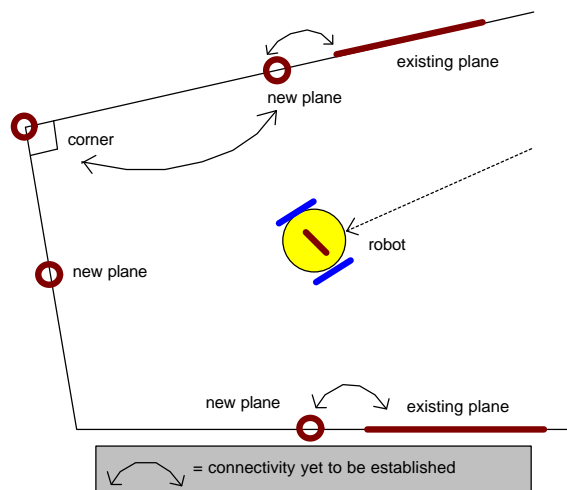
**Edge** is similarly characterised by its Cartesian coordinates  $x_i(k) = [x_i(k) \quad y_i(k)]^T$  only.

The sonar sensor provides no indication of its orientation.

In addition, the covariance and cross-covariance between these features are also kept (Leonard, Durrant-Whyte and Ingemar 1992, Moutarlier and Chatila 1989). Each time a new primitive is added, it will expand the number of system state parameters by two. The current strategy also records the unclassifiable features as unknown.

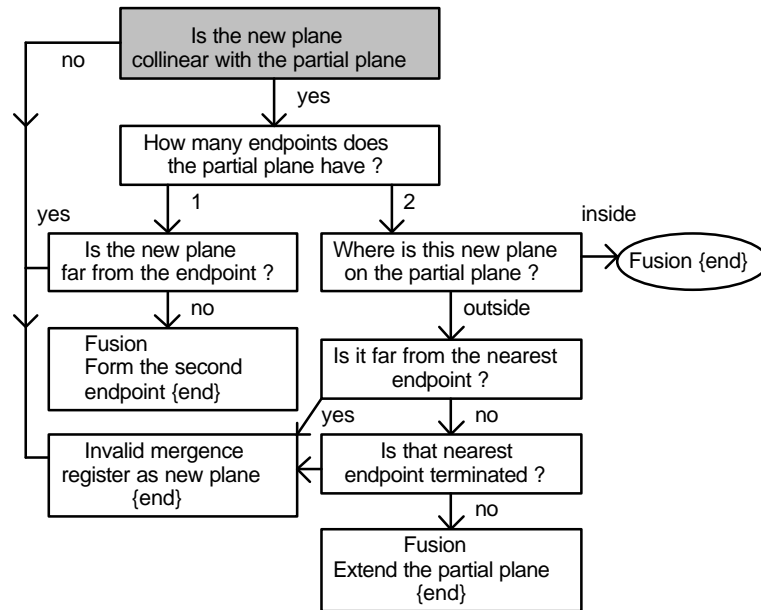
#### 4.2. Growing Map Primitives

A snapshot of the map building scenario at stage  $k+1$  is depicted in Figure 6. The robot has just moved to a new position and sensed a few new features. It is now ready to use some features for localisation, and add the remaining features to the map.

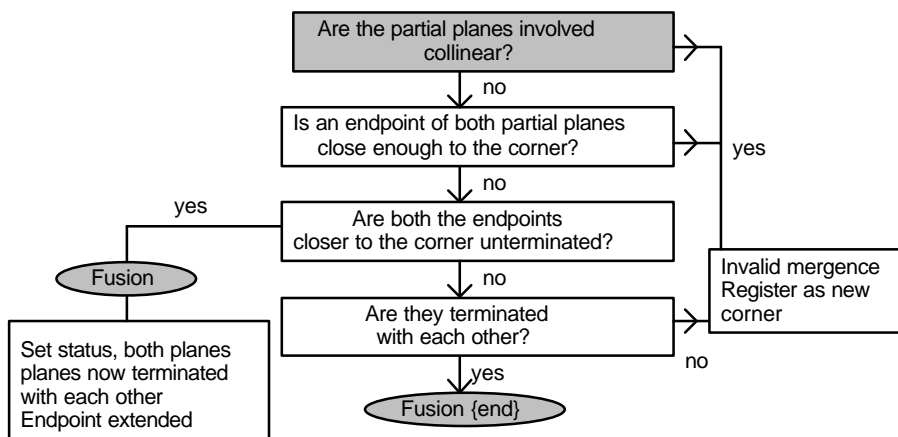


**Figure 6 : Status of map and data fusion process at stage  $k+1$**

Since the robot is operating indoors, discrete feature elements are assumed to come from a few planes, so that they can be merged using some *collinearity constraint* to give a more realistic representation of the environment.



**Figure 7 : Conditions for growing map primitives with a plane measurement**



**Figure 8 : Conditions for growing map primitives with a corner measurement**

A planar measurement would be fused to a partial plane if it satisfies the conditions depicted in Figure 7. A corner measurement would be fused to an existing corner feature if it is close enough to it, otherwise it would be fused to two existing intersecting planes if it satisfies the conditions depicted in Figure 8. In a typical real environment, edges are produced by the artefacts on the walls such as mouldings. While being excellent

stationary landmarks for map building and localisation, they cannot be considered as collinear with the nearby walls. Therefore an edge is only fused to an existing edge if they are in the proximity of each other. For all greyed condition boxes in the figures,  $\chi^2$  tests (to be described later) are applied. Every time a re-observation of a feature/relation occurs, the state of every map feature would be updated because of their correlation. The unterminated endpoints of partial planes are projected to the new gradient determined by the new state parameters, whereas the terminated endpoints are re-calculated from the intersections of all pairs of partial planes marked as terminated with each other.

### **4.3. The Kalman Filter Formulation of Map Building Problem**

Under this section, the map building problem is first formulated according to the classical global approach. A few equations are then highlighted and modified according to the concept of the Relocation-Fusion approach introduced by (Moutarlier and Chatila 1989). All these are done in the specific context of the sonar mapping. After embedding IEKF or JUDKF, the result is more general than the original formulation.

The two dimensional coordinates and orientation (collectively known as the *state*) of the robot, as well as the speed of sound, at stage  $k$  is denoted by the random vector  $\mathbf{x}_0(k) = [x(k) \quad y(k) \quad \mathbf{q}(k) \quad c_s(k)]^T$  with respect to a global reference frame. Further assume that a partial map already exists, and the random parameter vectors of the existing features  $\mathbf{x}_i(k)$  are concatenated with  $\mathbf{x}_0(k)$  to form the *global state vector*  $\mathbf{S}(k)$ .  $\mathbf{S}(k)$  contains all the parameters to be optimised, and  $S$  is the set of all state vectors to be optimised.

$$\mathbf{S}(k) = [\mathbf{x}_0(k) \quad \mathbf{x}_1(k) \quad \mathbf{x}_2(k) \quad \dots \quad \mathbf{x}_n(k)]^T \quad (28)$$

$$S = \bigcup_{i=0}^n \{\mathbf{x}_i\} \quad (29)$$

At stage  $k+1$ , the robot travels to a new destination. The intermediate state of the robot  $\hat{\mathbf{S}}(k+1|k)$  can be predicted as a function of its preceding state  $\hat{\mathbf{S}}(k|k)$  and the input vector  $\mathbf{U}(k+1)$  using the state transition equation (9) or (23). In this case  $\mathbf{U}(k+1)$  is specified by the distance travelled by the left wheel and right wheel. Strictly speaking, the time histories of the left and right wheel rotations,  $L(t)$  and  $R(t)$ , are required to compute the intermediate state. In this experiment, the motion types are confined to linear translation and on the spot rotation only. If the motion is a translation,  $L$  and  $R$  should have equal sign; Likewise, if the motion is a rotation,  $L$  and  $R$  should have opposite sign.

$$\mathbf{U} = \mathbf{U}(k+1) = \begin{bmatrix} L(k+1) \\ R(k+1) \end{bmatrix} \quad (30)$$

$$\mathbf{Cov}(\mathbf{U}(k+1)) = \begin{bmatrix} k_L^2 |\hat{L}(k+1)| & 0 \\ 0 & k_R^2 |\hat{R}(k+1)| \end{bmatrix} \quad (31)$$

Since a new model has been developed in (Chong and Kleeman 1997) for propagating random odometry errors, and motion will only transform  $\mathbf{x}_0(k/k)$ , equation (9), (23), (11) and (25) can be simplified further. For the IEKF method,

$$\hat{\mathbf{x}}_0^{IEKF}(k+1|k) = \mathbf{F}(\hat{\mathbf{x}}_0(k|k), \hat{\mathbf{U}}(k+1)) \quad (32)$$

$$\mathbf{P}_{00}^{IEKF}(k+1|k) = \nabla_{\mathbf{x}_0} \mathbf{F} \mathbf{P}_{00}(k|k) \nabla_{\mathbf{x}_0} \mathbf{F}^T + \mathbf{Odom}(\hat{\mathbf{U}}, \mathbf{Cov}(\mathbf{U})) \quad (33)$$

$$\mathbf{P}_{j0}^{IEKF}(k+1|k) = \nabla_{\mathbf{x}_0} \mathbf{F} \mathbf{P}_{00}(k|k) \quad \forall j > 0 \quad (34)$$

and for the JUDKF method,

$$\hat{\mathbf{x}}_0^{JU}(k+1|k) = \Omega(k|k, \mathbf{F}(\hat{\mathbf{x}}_0, \hat{\mathbf{U}}), \text{diag}(\mathbf{P}_{00}, \mathbf{Cov}(\mathbf{U}))) \quad (35)$$

$$\mathbf{P}_{00}^{JU}(k+1|k) = \Lambda(k|k, \mathbf{F}(\hat{\mathbf{x}}_0, \hat{\mathbf{U}}), \mathbf{P}_{00}) + \mathbf{Odom}(\hat{\mathbf{U}}, \mathbf{Cov}(\mathbf{U})) \quad (36)$$

$$\mathbf{P}_{0j}^{JU}(k+1|k) = \Lambda\left(k|k, \mathbf{F}(\hat{\mathbf{x}}_0, \hat{\mathbf{U}}), \hat{\mathbf{x}}_j, \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{0j} \\ \mathbf{P}_{j0} & \mathbf{P}_{jj} \end{bmatrix}\right) \quad \forall j > 0 \quad (37)$$

where  $\mathbf{Odom}()$  represents the new odometry error model developed in (Chong and Kleeman 1997) that takes in the robot's wheel covariance matrix  $\mathbf{Cov}(\mathbf{U}(k+1))$  and wheel turns  $\hat{\mathbf{U}}$  and outputs the propagated covariance matrix.

A measurement vector consists of a time of flight  $r_i$  and a direction  $\mathbf{Y}_i$  to a target, and is denoted by

$$\mathbf{M} = \mathbf{M}_i(k+1) = [r_i(k+1) \quad \mathbf{y}_i(k+1)]^T \quad (38)$$

Every new measurement is tested against all the possible collinearity constraints set out in section 4.2, in order to grow the map primitives. A typical constraint would take the form of (3). Its residual vector (also known as innovation) and its covariance can be computed for each measurement, using (10) and (11) for IEKF and (24) and (26) for JUDKF respectively.

Since the noise incurred on these residuals are not correlated, block processing is not necessary (Bar-Shalom and Li 1993) (that is, they can be processed one at a time). Each

residual vector  $\mathbf{z}_i(k+1)$  is just a function of  $\hat{\mathbf{x}}_0(k|k)$ , the measurement  $\hat{\mathbf{M}}_i(k+1)$  and the corresponding ‘matched’ map features, therefore there are significant zero submatrices in the Jacobian matrix on which simplification can be made. The following simplifying formulation involves only one feature,  $\hat{\mathbf{x}}_i(k|k)$ . Formulation involving two states (for example, fusing a new corner measurement to two existing intersecting partial planes) is similar so will not be detailed. The modified residual vector looks like

$$\mathbf{z}_i^{IEKF}(k+1; \mathbf{h}_r) = -\mathbf{G}(\hat{\mathbf{x}}_0(k+1|k), \hat{\mathbf{x}}_i(k+1|k), \hat{\mathbf{M}}_i(k+1)) \quad (39)$$

$$\mathbf{z}_i^{JU}(k+1) = -\Omega \left( k+1|k, \mathbf{G}(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_i, \hat{\mathbf{M}}_i), \text{diag} \left( \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{0i} \\ \mathbf{P}_{i0} & \mathbf{P}_{ii} \end{bmatrix}, \mathbf{Cov}(\mathbf{M}_i) \right) \right) \quad (40)$$

and its error covariance,

$$\begin{aligned} \mathbf{P}_{zz}^{IEKF}(k+1|k; \mathbf{h}_r) = & \begin{bmatrix} \nabla_{\mathbf{x}_0} \mathbf{G} & \nabla_{\mathbf{x}_i} \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{00}(k|k) & \mathbf{P}_{0i}(k|k) \\ \mathbf{P}_{i0}(k|k) & \mathbf{P}_{ii}(k|k) \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{x}_0} \mathbf{G}^T \\ \nabla_{\mathbf{x}_i} \mathbf{G}^T \end{bmatrix} \\ & + \nabla_{\mathbf{M}_i} \mathbf{G} \mathbf{Cov}(\mathbf{M}_i(k+1)) \nabla_{\mathbf{M}_i} \mathbf{G}^T \end{aligned} \quad (41)$$

$$\mathbf{P}_{zz}^{JU}(k+1|k) = \Lambda \left( k+1|k, \mathbf{G}(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_i, \hat{\mathbf{M}}_i), \text{diag} \left( \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{0i} \\ \mathbf{P}_{i0} & \mathbf{P}_{ii} \end{bmatrix}, \mathbf{Cov}(\mathbf{M}_i) \right) \right) \quad (42)$$

where  $\mathbf{h}_r$  is the  $r^{\text{th}}$   $\hat{\mathbf{S}}(k+1|k+1)$  generated by IEKF, triggered with  $\mathbf{h}_0 = \hat{\mathbf{S}}(k+1|k)$ . Here  $\hat{\mathbf{S}}(k+1|k)$  includes  $\hat{\mathbf{x}}_0(k+1|k)$  and  $\hat{\mathbf{x}}_i(k+1|k)$  only.

The covariance of the measurement should account for the imperfect polygonal world assumption. For example, not all walls are strictly flat. It has a form depicted by

equation (43) but more about the matrix values is presented later.

$$\mathbf{Cov}(\mathbf{M}_i(k+1)) = \begin{bmatrix} \mathbf{S}_{r_i}^2 & \mathbf{S}_{ry_i} \\ \mathbf{S}_{ry_i} & \mathbf{S}_{y_i}^2 \end{bmatrix} \quad (43)$$

Kalman Filter equations require that the cross-covariance between the observation matrix and the state matrix be evaluated with equation (13) and (27). After that, the state and the error covariance matrix of the map features together with the robot's position can be updated with equation (7) and (8). Once again, efficiency can be improved by processing the covariance matrix in disparate blocks. To update the state of  $x_j \in \mathcal{S}$ ,

$$\mathbf{P}_{jz}^{IEKF}(k+1|k; \mathbf{h}_r) = \mathbf{P}_{j0}(k+1|k) \nabla_{\mathbf{x}_0} \mathbf{G}^T + \mathbf{P}_{ji}(k+1|k) \nabla_{\mathbf{x}_i} \mathbf{G}^T \quad (44)$$

$$\mathbf{P}_{jz}^{JU}(k+1|k) = \Lambda \left( k+1|k, \hat{\mathbf{x}}_j, \mathbf{G}(\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_i, \hat{\mathbf{M}}_i), \text{diag} \left( \begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{0i} & \mathbf{P}_{0j} \\ \mathbf{P}_{i0} & \mathbf{P}_{ii} & \mathbf{P}_{ij} \\ \mathbf{P}_{j0} & \mathbf{P}_{ji} & \mathbf{P}_{jj} \end{bmatrix}, \mathbf{Cov}(\mathbf{M}_i) \right) \right) \quad (45)$$

For IEKF, let  $\mathbf{h}_{j,r}$  be the iterator for  $\mathbf{x}_j$  only,  $\hat{\mathbf{x}}_j(k+1|k+1)$ , after the  $r^{\text{th}}$  iteration, starting with the initial value  $\mathbf{h}_{j,0} = \hat{\mathbf{x}}_j(k+1|k)$

$$\mathbf{h}_{j,r+1} = \mathbf{h}_{j,0} + \mathbf{P}_{jz}(k+1|k; \mathbf{h}_r) \mathbf{P}_{zz}^{-1}(k+1|k; \mathbf{h}_r) [\mathbf{z}_i(k+1; \mathbf{h}_r) - \nabla_{\mathbf{S}} G(\mathbf{h}_0 - \mathbf{h}_r)] \quad (46)$$

For JUDKF,  $\hat{\mathbf{x}}_j(k+1|k+1)$  is simply found with the following classical equation,

$$\hat{\mathbf{x}}_j(k+1|k+1) = \hat{\mathbf{x}}_j(k+1|k) + \mathbf{P}_{jz}(k+1|k) \mathbf{P}_{zz}^{-1}(k+1|k) \mathbf{z}_i(k+1) \quad (47)$$



Both cases share the same covariance update formula. For all combinations of state  $m$  and  $n$ ,

$$\mathbf{P}_{mn}(k+1|k+1) = \mathbf{P}_{mn}(k+1|k) - \mathbf{P}_{mz}(k+1|k; \mathbf{h}_r) \mathbf{P}_{zz}^{-1}(k+1|k; \mathbf{h}_r) \mathbf{P}_{zn}(k+1|k; \mathbf{h}_r) \quad (48)$$

Extra care has been taken when forming the covariance matrices required by JUDKF. For example, in equation (4444), if  $j=0$  or  $j=i$ , then the composite covariance matrix passed into the JUDKF function should only include  $\mathbf{P}_{00}$ ,  $\mathbf{P}_{0i}$ ,  $\mathbf{P}_{i0}$  and  $\mathbf{P}_{ii}$  only. Otherwise, a redundant (hence singular with zero determinant) covariance would be formed which triggers a fatal computer run-time error.

The process is then repeated until all observations have been processed. Since IEKF is also an extremely computationally demanding implementation, simplification becomes essential. The original algorithm is modified such that it terminates after exactly three iterations. Under this simplification, the iterator should only contain the states which affect all the matrix terms appearing in the IEKF algorithm, namely  $\mathbf{x}_0$  and  $\mathbf{x}_i$ .

After the fusion of all measurements associated with the reobserved features, the remaining features are considered new and are simply incorporated into the global state. More information about fusing new observations is contained in section 4.7.

The Relocation-Fusion approach formulated by (Moutarlier and Chatila 1989) makes a minor variation on the Global approach. The measurement error is first used to update the robot's state  $\mathbf{x}_0$  ONLY. The improved  $\mathbf{x}_0$  is then used to re-calculate the residual vector and all the related Jacobian matrices, which are then used to update the remaining map features. Stepwise, after  $\mathbf{z}_i$  is computed,  $\mathbf{x}_0$  and  $\mathbf{P}_{00}$  and the cross-

covariance between  $\mathbf{x}_0$  and all other map features  $\mathbf{x}_n$  can be found using (47) by setting  $j=0$  and (48) by setting  $m=0$ , respectively.

With the improved  $\mathbf{x}_0$ ,  $\mathbf{z}_i$ , ( $\nabla_{\mathbf{x}_0} \mathbf{G}$  and  $\nabla_{\mathbf{x}_i} \mathbf{G}$  in case of IEKF),  $\mathbf{P}_{zz}$ , and  $\mathbf{P}_{jz}$  can be re-generated, in this particular order, by applying equation (39) to (42). This is followed by the update of the states of all other map features, all covariance and cross-covariance, excluding  $\mathbf{x}_0$  and  $\mathbf{P}_{00}$ , using equation (47) to (48). To embed the IEKF algorithm, iteration is first performed on  $\mathbf{x}_0$ . The matched feature  $\mathbf{x}_i$  is included in the iteration after three runs. After this, the remaining states follow.

```

set r←0

set  $\mathbf{h}_{0,r} = \mathbf{x}_0(k+1|k)$ ,  $\mathbf{h}_{i,r} = \mathbf{x}_i(k+1|k)$ 

repeat {      /* Relocation with IEKF */

    evaluate  $\mathbf{z}_i, \nabla_{\mathbf{x}_0} \mathbf{G}, \nabla_{\mathbf{x}_i} \mathbf{G}, \mathbf{P}_{zz}, \mathbf{P}_{0z}$  at  $\mathbf{h}_{0,r}$  and  $\mathbf{h}_{i,0}$ 

     $\mathbf{h}_{0,r+1} = \mathbf{h}_{0,0} + \mathbf{P}_{0z} \mathbf{P}_{zz}^{-1} (\mathbf{z}_i - \nabla_{\mathbf{x}_0} \mathbf{G}(\mathbf{h}_{0,0} - \mathbf{h}_{0,r}))$ 

    r←r+1

} while (r<3)

 $\forall \mathbf{x}_j \in S \setminus \{\mathbf{x}_0\}$ , evaluate  $\mathbf{P}_{jz}$ 

 $\forall \mathbf{x}_j \in S$ , update  $\mathbf{P}_{0j}$  and  $\mathbf{P}_{j0}$ 

set r←0

repeat {      /* Fusion with IEKF */

    evaluate  $\mathbf{z}_i, \nabla_{\mathbf{x}_0} \mathbf{G}, \nabla_{\mathbf{x}_i} \mathbf{G}, \mathbf{P}_{zz}, \mathbf{P}_{iz}, \mathbf{P}_{0z}$  at  $\mathbf{h}_{0,2}$  and  $\mathbf{h}_{i,r}$ 

     $\mathbf{h}_{i,r+1} = \mathbf{h}_{i,0} + \mathbf{P}_{iz} \mathbf{P}_{zz}^{-1} (\mathbf{z}_i - \nabla_{\mathbf{x}_i} \mathbf{G}(\mathbf{h}_{i,0} - \mathbf{h}_{i,r}))$ 

    r←r+1

```

```

} while (r<3)

 $\forall \mathbf{x}_j \in S \setminus \{\mathbf{x}_i\}$ , evaluate  $\mathbf{P}_{jz}$ 

 $\forall \mathbf{x}_j \in S \setminus \{\mathbf{x}_0, \mathbf{x}_i\}$ , update  $\mathbf{x}_j$ 

 $\forall \{\mathbf{x}_m, \mathbf{x}_n\} \subset S$ , update  $\mathbf{P}_{mn}$  (exclude  $\mathbf{P}_{00}$ )

```

The Relocation-Fusion (Moutarlier and Chatila 1989) approach has been shown to be less sensitive to position bias introduced by non-linearities and non-ideal odometry model, at the expense of optimality caused by the removal of position information from map features update. In the implementation here, this approach is applied hand in hand with the IEKF and JUDKF to achieve the maximal effect.

#### 4.4. Why Should a New Corner be Fused to Two Intersecting Partial Planes ?

If a corner measurement fails to be fused to one of the standalone corners, an attempt will be made to fuse it to two intersecting planes. The reason behind fusing a corner to two intersecting planes is that, a corner measurement vector  $[a_i(\mathbf{x}_0, \mathbf{M}_i) \ b_i(\mathbf{x}_0, \mathbf{M}_i)]^T$  has a size  $2 \times 1$ . If it sets up a collinearity constraint with one partial plane  $[a \ b]^T$  only, then the residual vector formed would have a size  $1 \times 1$ , that is,  $\mathbf{z}_i = [a, a + b; b - a^2 - b^2]$ . After fusion, three options for dealing with the corner are available:

- Discard the corner measurement. This wastes some information.
- Parameterise again the partial plane to create some kind of ‘plane-corner’ entity, which should have  $\text{size}(\text{partial plane vector}) + \text{size}(\text{corner vector}) - \text{size}(\text{residual vector}) = 3$  state parameters. This leads to a series of avalanche effects. For example, it would later lead to some ‘plane-corner-plane’ entity and so forth. This complicates the map

management process many folds.

- Register the corner as a new map feature too. However, since it has been used to fuse a partial plane before, any composite covariance matrix involving both of these features, like the state covariance matrix, would carry redundancy (not full rank), hence suffer the risk of singularity (zero determinant). Once again, this choice calls for complex map management scheme, as instances like this increase.

On the contrary, if it is not used at all, then there wouldn't be redundancy to consider. If two intersecting partial planes could be found, the constraint would be 'the corner should be collinear with two partial planes', so the residual vector would have a size of  $2 \times 1$ . This means the corner measurement can be discarded after fusion without wasting any information. On top of that, the two partial planes would be marked as terminated with each other, so the corner position could always be generated if required by subsequent path planning. The test to ensure that the two partial planes are not collinear is vital because if not, later if the two partial planes are merged into one, the resultant partial plane would have a 'hanging' terminated endpoint.

#### **4.5. Validity of Collinearity Constraint**

To access the validity of a constraint relationship, Mahalabonis distance test (or also known as  $\mathbf{c}^2$  test) is applied to the residual vector :

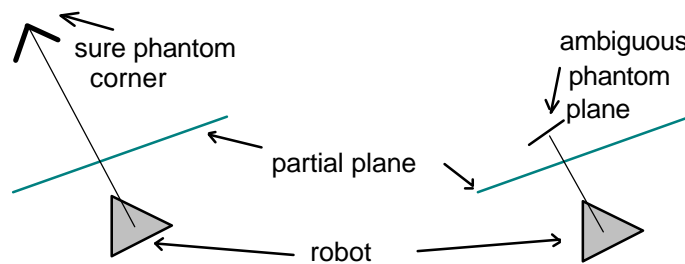
$$\|\mathbf{z}_i\|_{\mathbf{P}_{zz}}^2 = \mathbf{z}_i^T(k+1)\mathbf{P}_{zz}^{-1}(k+1|k)\mathbf{z}_i(k+1) \quad (49)$$

where  $\|\mathbf{z}_i\|_{\mathbf{P}_{zz}}^2$  is the normalised sum of square of all the vector components. If the residual

vector is assumed to be jointly Gaussian, then the expression will have a  $\chi^2$  distribution with degree of freedom determined by the rank of  $\mathbf{P}_{zz}$ . A one-sided acceptance interval is chosen to establish a 90% probability concentration ellipsoid in the distribution. A new measurement whose  $\|\mathbf{z}_i\|_{\mathbf{P}_{zz}}^2$  falls in this acceptance interval is assumed to have satisfied the collinearity constraint set up with the existing feature(s). In this work, all residual vectors have a size of  $2 \times 1$ , so the degree of freedom is 2, and the acceptance interval is  $< 5.991$ .

To improve computational efficiency,  $\mathbf{z}_i$  which is considerably different from  $\mathbf{0}$  is rejected without going through the test, to avoid the series of matrix operations. At this stage, the issue of features falling into more than one validation gates has been temporarily put aside. This problem can arise either when the position covariance is too large, or when two existing map features are very close together but are not yet merged. This difficult issue will be investigated in the future.

#### 4.6. Distinguishing Phantom Targets



**Figure 9 : Example of treatment of phantom targets**

Local maps are preserved. Each feature in the local map has a parameter indicating which state it has been fused to. Therefore, the knowledge of where a particular map primitive was observed, is available. When the map is sufficiently complete, many phantom targets caused by specular reflection can be eliminated by checking whether the line of sights

from the positions they were observed are blocked by some partial planes. If the phantom targets are too close to some partial planes they are considered ambiguous and would not be eliminated. The strategies stem from the experimental observation that in a cramped indoor environment, such as a narrow corridor, corners and edges are more likely to cause phantom targets than planes. This is a result of the property that corners (and edges) return (some) acoustic energy in the opposite direction to its arrival, while planes reflect energy away from the arrival direction except for normal incidence.

#### 4.7. Fusion of the Remaining New Features

After localisation, the fusion of the remaining features will make use of the estimated robot position. Each new feature  $\mathbf{x}_i$  is a function  $\mathbf{H}()$  of the robot's position  $\mathbf{x}_0$  and a measurement vector  $\mathbf{M}_i$ . For each new feature, the error covariance can be calculated :

$$\hat{\mathbf{x}}_i^{IEKF}(k+1|k+1) = \mathbf{H}\left(\hat{\mathbf{x}}_0(k+1|k+1), \hat{\mathbf{M}}_i(k+1)\right) \quad (50)$$

$$\mathbf{P}_{ii}^{IEKF}(k+1|k+1) = \nabla_{\mathbf{x}_0} \mathbf{H} \mathbf{P}_{00}(k+1|k+1) \nabla_{\mathbf{x}_0} \mathbf{H}^T + \nabla_{\mathbf{M}_i} \mathbf{H} \mathbf{Cov}(\mathbf{M}_i(k+1)) \nabla_{\mathbf{M}_i} \mathbf{H}^T \quad (51)$$

$$\hat{\mathbf{x}}_i^{JU}(k+1|k+1) = \Omega\left(k+1|k+1, \mathbf{H}(\hat{\mathbf{x}}_0, \hat{\mathbf{M}}_i), \text{diag}(\mathbf{P}_{00}, \mathbf{Cov}(\mathbf{M}_i))\right) \quad (52)$$

$$\mathbf{P}_{ii}^{JU}(k+1|k+1) = \Lambda\left(k+1|k+1, \mathbf{H}(\hat{\mathbf{x}}_0, \hat{\mathbf{M}}_i), \text{diag}(\mathbf{P}_{00}, \mathbf{Cov}(\mathbf{M}_i))\right) \quad (53)$$

and all the cross-covariance among the new features and the existing features are also generated. Let  $j$  denote the objects already in the map,

$$\mathbf{P}_{ij}^{IEKF}(k+1|k+1) = \nabla_{\mathbf{x}_0} \mathbf{H} \mathbf{P}_{0j}(k+1|k+1) \quad \forall j \neq i \quad (54)$$

$$\mathbf{P}_{ij}^{JU}(k+1|k+1) = \Lambda\left(k+1|k+1, \mathbf{H}(\hat{\mathbf{x}}_0, \hat{\mathbf{M}}_i), \hat{\mathbf{x}}_j, \text{diag}\left(\begin{bmatrix} \mathbf{P}_{00} & \mathbf{P}_{0j} \\ \mathbf{P}_{j0} & \mathbf{P}_{jj} \end{bmatrix}, \mathbf{Cov}(\mathbf{M}_i)\right)\right) \quad (55)$$

As a reminder, if  $j=0$ , the composite matrix passed to the JUDKF function should comprise  $\mathbf{P}_{00}$  and  $\mathbf{Cov}(\mathbf{M}_i)$  only.

#### **4.8. Simultaneous Encounter of Collinear Features**

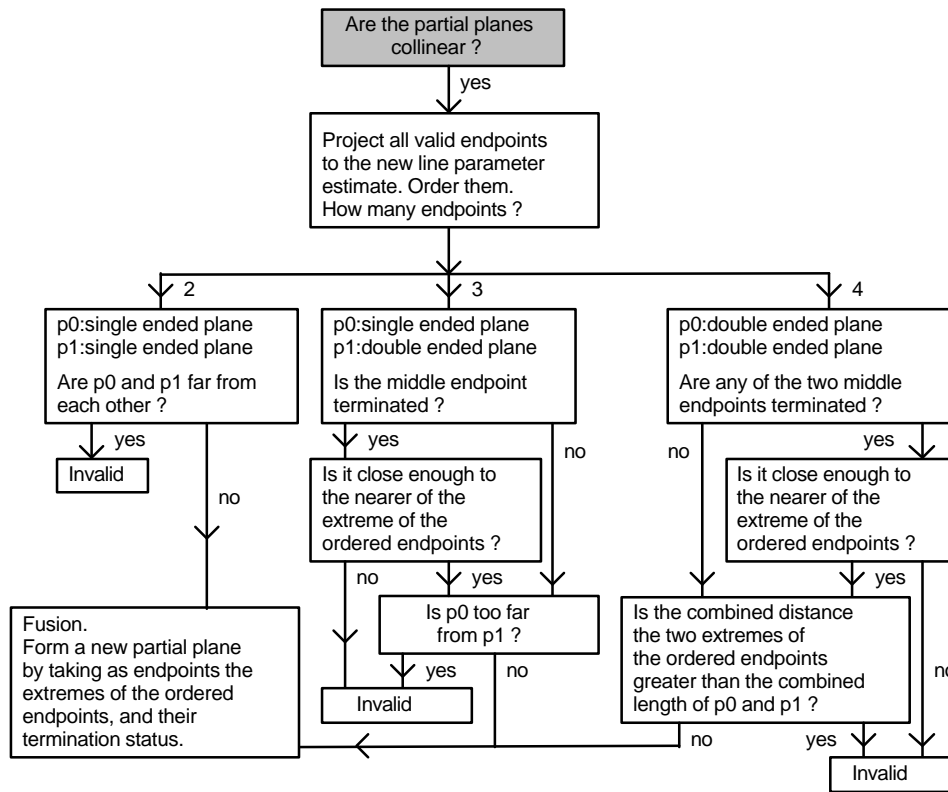
There would be occasions when two or more collinear features are encountered at the same stage. For instance, this situation would occur if the robot reaches a corner and observes the corner and the walls that form the corner for the first time. When this situation arises, the planar feature is first incorporated into the global state vector as a new feature. The corner feature is then regarded as the observation for that new feature.

#### **4.9. Removal of Redundant Primitives**

Removal of redundant primitives would occur when

1. Two existing partial planes are actually collinear and adjacent to each other.
2. Two existing corners are the same.
3. Two existing edges are the same.
4. An existing corner appears to be located at the intersection of two existing partial planes.

In such cases, internal fusion is performed by forming residual vectors in a manner similar to section 5. In terms of equations, all terms involving  $\mathbf{M}_i$  are set to nil because no new measurements are involved. For the last three cases listed above, the map primitive growth assessment/procedure is similar to that depicted in Figure 8, except that when invalidity occurs, integration of the new feature need not be carried out. To merge two partial planes, the new procedure depicted in Figure 10 is followed.



**Figure 10 : Conditions for merging two existing partial planes in map**

As a brief illustration, when two existing features, with states  $\mathbf{x}_i(k)$  and  $\mathbf{x}_j(k)$  respectively, are to form a collinearity constraint, redundancy can be removed by enhancing the estimation of  $\mathbf{x}_i(k)$  with  $\mathbf{x}_j(k)$ , and discarding  $\mathbf{x}_j(k)$  afterwards. After filtering, all  $\mathbf{x}_j$  related covariance and cross covariances can be removed. The three-run IEKF algorithm is applied in a similar fashion so it will not be elaborated further. If a constraint involves three states, as in the case of fusing a corner to two intersecting partial planes, the formulation is once again similar.

It is also possible to exploit the orthogonality constraint among partial planes. However, not all intersecting walls in today's indoor environments are strictly perpendicular, so the idea has not been implemented even though it can be accommodated. If implemented, a  $\chi^2$  test would also be applied to assess orthogonality.



## **5. Experimental Results**

Experiments have been carried out in four artificial environments erected with cardboard boxes but here two are shown from Figure 11(a) to Figure 12(a). The odometry of the robot has been calibrated to reduce systematic errors, and the parameters required by the non-systematic error model have been obtained in (Chong and Kleeman 1997) prior to experiment.

Since the cardboard boxes were being lined up manually taking the gridlines on the parquetry floor as reference, the variance associated with the time of flight measurement and angular measurement were set larger than that achievable by the sonar sensor (Kleeman and Kuc 1995) in order for the collinearity constraints to hold. The initial value of speed of sound was set to 342.5 m/s, which is the mean value at the time. In fact, for all four experiments, the following tentative values were used:

$$\text{standard deviation of time of flight} = 1.6 \times 10^{-5} \text{ s}$$

$$\text{standard deviation of direction} = 2.4^\circ$$

$$\text{initial standard deviation of } c_s = 0.18 \text{ m/s}$$

The resultant maps are shown in Figure 11(c)(d) to Figure 12(c)(d). The (b) subfigures show the raw sonar measurements detected at various positions (before position correction) being superimposed onto the same diagrams, and the ‘scan lines’ from one of the position indicate the typical number of features the sonar sensor can capture at any one time. The grid spacing is 1 metre. It has been noticed that the sensor detects the gaps between the cardboard boxes as edges. They should not be regarded as

some artificial aids to the mapping process as in a real environment, wall mouldings are often found to give rise to the same phenomenon.

In the first environment, the robot was programmed to repetitively enter, make a 180° turn, exit an enclosure four times to investigate the long term performance of both filters. The maps generated with JUDKF and IEKF are very similar but the speed of JUDKF is significantly slower. Both filters have remained consistent throughout the navigation. JUDKF produces a map with all features correctly merged. IEKF's performance closely matches that of JUDKF, with only one corner not fused to two partial planes and two edge features not identified as belonged to the same physical edge. Their fusions are found to be hindered by the  $\chi^2$  tests. A comparison of the covariance generated by IEKF and JUDKF for a few features indicates that JUDKF in general tends to generate larger covariance. As a result, the error ellipses for 'related' features are more likely to overlap and more mergence can be observed. Despite the minor imperfection, both post-filtering maps show that only one partial plane is generated for each wall, and most of the corners have been successfully fused to two intersecting partial planes, hence well defined intersections can be observed. Also, most repetitive observations of the same edge are successfully merged into one edge map feature. All unterminated endpoints of partial planes have also been properly projected to the line parameters. Three phantom corners are retained in the raw data map, but are subsequently eliminated in the post-filtering maps by the partial planes blocking their lines of sight.

In the second experiment, the robot was programmed to follow a rectangular path four times. One side of the 'wall' was indented by about 0.5 metre. The observations made about this experiment are virtually the same as those made in the third experiment, so no repetition is necessary. At first glance, it might seem wasteful not to extend the

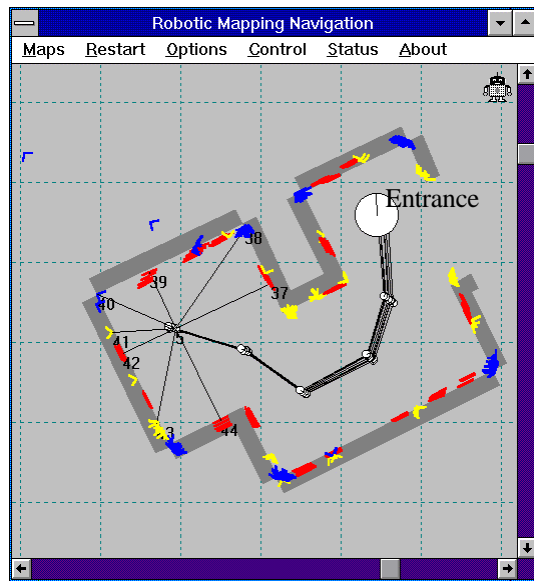
partial planes with the edge artefacts by exploiting another collinearity constraint. This idea is found to be impractical in real world for two important reasons:

- Most edge reflections are generated by artefacts which are not necessarily collinear with the wall, such as wall mouldings.
- Even if the edges are really collinear with the partial planes, once they have been merged into the partial planes, they have to be eliminated, much like merging a corner to two partial planes. Unlike the corner, the fused edge cannot be regenerated when required. Simply put, a corner can be regenerated by solving for the point of intersection of two partial planes, but no similar approach is available for a fused edge. Therefore, fusing an edge reduces the number of useful landmarks for localisation.

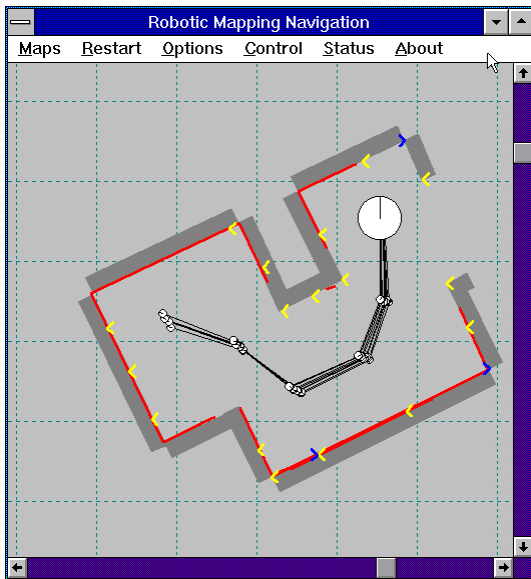
Figure 13(a)-(b) compare the time taken to fuse all measurements into the map at all sensing points for the two experiments. The experiments were conducted on a 486DX-33 MHz computer with 16 MB RAM. The time taken to move the robot and to process sonar echoes by template matching have been excluded from the comparison. All graphs show that JUDKF requires about 5 times as much processing time as IEKF. It has been observed that the speed discrepancy is primarily caused by that fact that, for IEKF, pre-computed matrices such as  $\nabla F$  and  $\nabla G$  can be used in several cross-covariance and state updates in each mapping cycle. Whereas for JUDKF, a square root of a matrix with Cholesky Decomposition must be performed for virtually every cross-covariance (except pairs with transpose relationship) and state update.



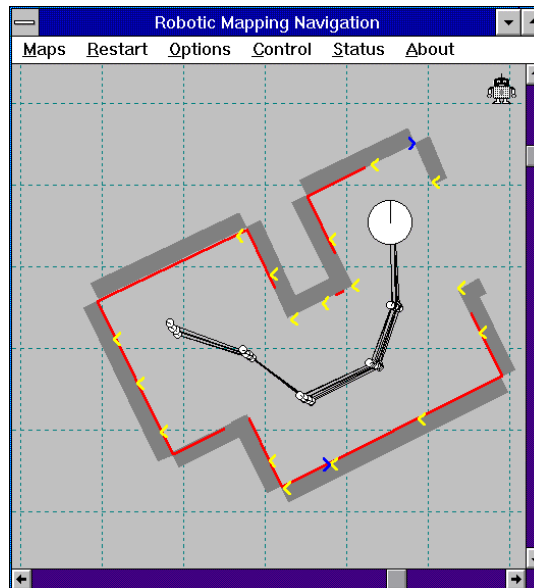
(a) Actual environment



(b) Pre-filtering Perception



(c) IEKF

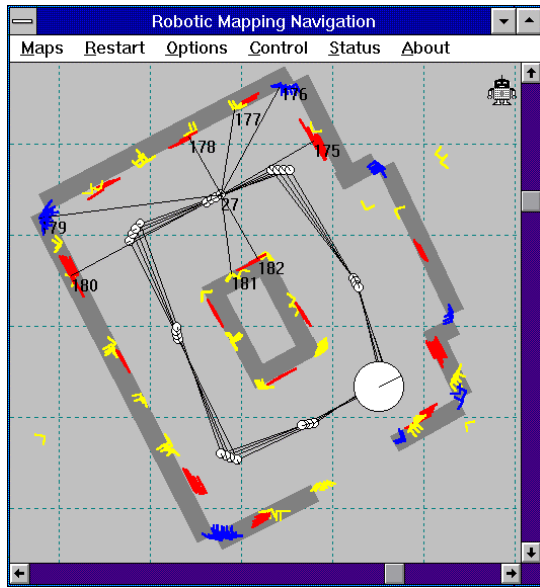


(d) JUDKF

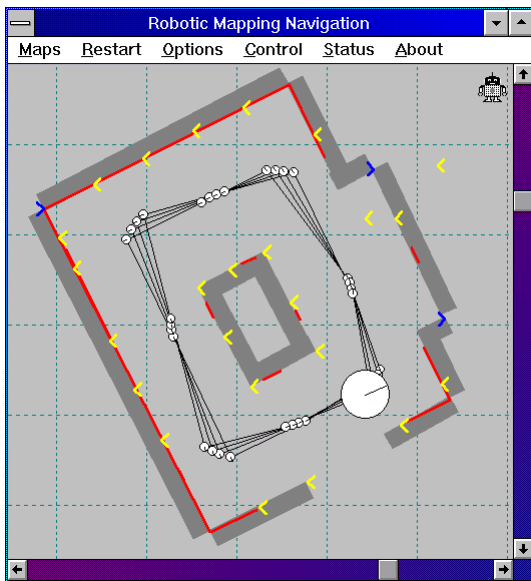
Figure 11 : The first environment, with the robot navigating into and out of the enclosure four times



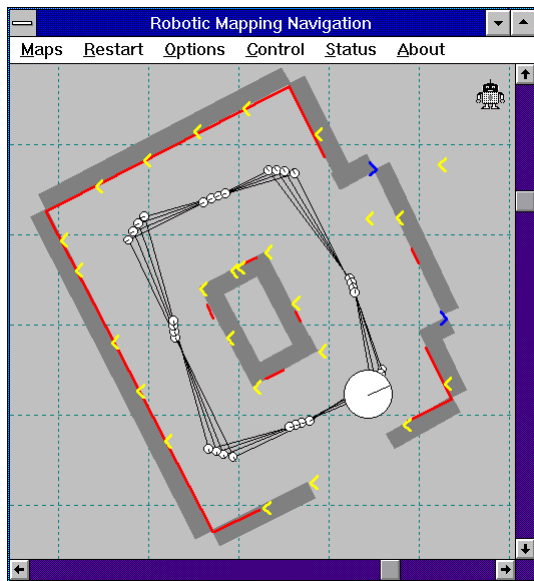
(a) Actual environment



(b) Pre-filtering Perception

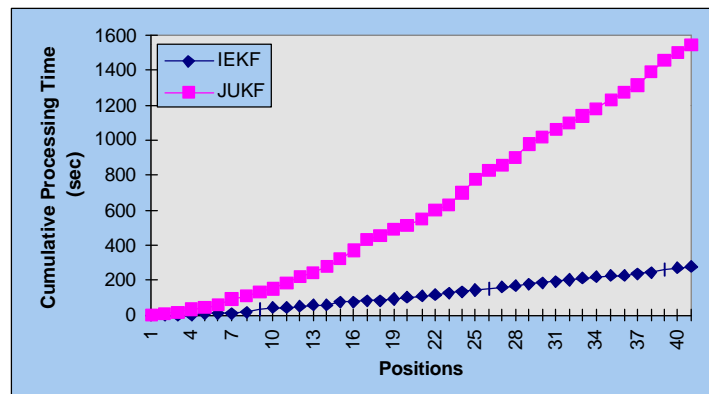


(c) IEKF

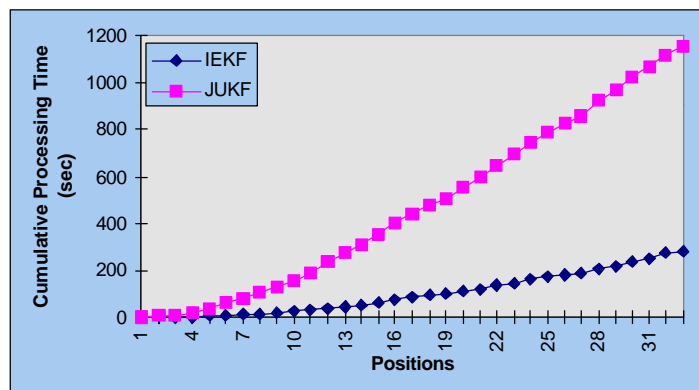


(d) JUDKF

Figure 12 : The second environment, with the robot repeating a rectangular path four times



(c) The first experiment



(d) The second experiment

Figure 13 : Cumulative processing time at all sensing positions for the two experiments.

## **6. Conclusion**

The capability of autonomous navigation by mapping of our mobile robot system in some simple environments has been demonstrated. IEKF and JUDKF have been employed to deal with the problem of covariance propagation through nonlinear transformation, and their strengths and weaknesses with regards to accuracy and speed have been compared with simulated and real data. It has been shown that the accuracy demonstrated by IEKF is comparable to that by JUDKF and is in fact sufficient in practice. While eliminating the tedium of deriving Jacobian matrices, JUDKF is less efficient compared to IEKF. The algorithm is now being intensively upgraded to enhance its robustness and efficiency. Current research focal points include the elimination of the storage and update of the covariance between two features if it is found to be small, in order to improve the speed and memory requirement of the algorithm. Also under investigation is a map matching strategy to re-establish robot's position when its uncertainty is too large or when the accumulation of position bias becomes significant.

## **7. Acknowledgment**

Mr. Greg Curmi's assistance in the design of the robot and sonar implementation, the funding of a large ARC grant, the financial assistance provided by an MGS scholarship and an OPRS Scholarship are all gratefully acknowledged.

### **Appendix: Some Implementation Examples**

This appendix evaluates some of the equations given in the main body of this paper. Let  $B$  denote the effective wheelbase, after moving, the robot's new position can be computed from the following state update equations:

$$\hat{\mathbf{x}}_0(k+1|k) = \begin{bmatrix} \hat{x}(k|k) + \hat{r}(k+1) \left( \sin \hat{\mathbf{q}}(k|k) - \sin \left( \hat{\mathbf{q}}(k|k) + \frac{\hat{R}(k+1) - \hat{L}(k+1)}{B} \right) \right) \\ \hat{y}(k|k) + \hat{r}(k+1) \left( \cos \left( \hat{\mathbf{q}}(k|k) + \frac{\hat{R}(k+1) - \hat{L}(k+1)}{B} \right) - \cos \hat{\mathbf{q}}(k|k) \right) \\ \hat{\mathbf{q}}(k|k) + \frac{\hat{R}(k+1) - \hat{L}(k+1)}{B} \\ c_s(k|k) \end{bmatrix} \quad (56)$$

where

$$\hat{r}(k+1) = \frac{B}{2} \left( \frac{\hat{L}(k+1) + \hat{R}(k+1)}{\hat{L}(k+1) - \hat{R}(k+1)} \right) \quad (57)$$

The Jacobian matrices with respect to  $\hat{\mathbf{x}}_0(k|k)$  and  $\hat{\mathbf{U}}(k+1)$  can be found in (Chong and Kleeman 1997) hence will not be reproduced here.

To match a corner feature to two partial planes  $\mathbf{x}_i = [a_i \ b_i]^T$  and  $\mathbf{x}_j = [a_j \ b_j]^T$ , the residual vector and various Jacobian matrices are

$$\mathbf{z} = - \begin{bmatrix} a_i(x + r_i c_s \cos \mathbf{a}_i) + b_i(y + r_i c_s \sin \mathbf{a}_i) - a_i^2 - b_i^2 \\ a_j(x + r_j c_s \cos \mathbf{a}_j) + b_j(y + r_j c_s \sin \mathbf{a}_j) - a_j^2 - b_j^2 \end{bmatrix} \quad (58)$$

$$\nabla_{\mathbf{M}_i} \mathbf{G} = \begin{bmatrix} c_s(a_i \cos \mathbf{a}_i + b_i \sin \mathbf{a}_i) & r_i c_s(-a_i \sin \mathbf{a}_i + b_i \cos \mathbf{a}_i) \\ c_s(a_j \cos \mathbf{a}_j + b_j \sin \mathbf{a}_j) & r_j c_s(-a_j \sin \mathbf{a}_j + b_j \cos \mathbf{a}_j) \end{bmatrix} \quad (59)$$



$$\nabla_{\mathbf{x}_0} \mathbf{G} = \begin{bmatrix} a_i & b_i & r_i c_s (-a_i \sin \mathbf{a}_i + b_i \cos \mathbf{a}_i) & r_i (a_i \cos \mathbf{a}_i + b_i \sin \mathbf{a}_i) \\ a_j & b_j & r_j c_s (-a_j \sin \mathbf{a}_j + b_j \cos \mathbf{a}_j) & r_j (a_j \cos \mathbf{a}_j + b_j \sin \mathbf{a}_j) \end{bmatrix} \quad (60)$$

$$\nabla_{x_i} \mathbf{G} = \begin{bmatrix} x + r_i c_s \cos \mathbf{a}_i - 2a_i & y + r_i c_s \sin \mathbf{a}_i - 2b_i \\ 0 & 0 \end{bmatrix} \quad (61)$$

$$\nabla_{y_i} \mathbf{G} = \begin{bmatrix} 0 & 0 \\ x + r_i c_s \cos \mathbf{a}_i - 2a_j & y + r_i c_s \sin \mathbf{a}_i - 2b_j \end{bmatrix} \quad (62)$$

More constraint equations and Jacobian matrices can be found in (Chong and Kleeman 1997).

## 9. References

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

Bar-Shalom, Y. and Li, X.R. 1993. *Estimation and tracking: principles, techniques and software*. Boston, London: Artech House Inc.

Barshan, B and Kuc, R. 1990. Differentiating sonar reflections from corners and planes employing an intelligent sensor, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(6):560-569.

Bozma, O. and Kuc, R. 1991, Building a sonar map in a specular environment using a single mobile sensor, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12):1260-1269.

Chong, K.S. and Kleeman, L. 1997. Accurate odometry and error modelling for a mobile

robot. *IEEE International Conference on Robotics and Automation*, Albuquerque, pp. 2783-2788.

Chong, K.S. and Kleeman, L. 1996. Mobile robot map building from an advanced sonar array and accurate odometry. Technical Report MECSE-1996-10, Melbourne. Monash University. Department of Electrical and Computer Systems Engineering.

Dudek, G., Freedman, P. and Rekleitis, I. M. 1996. Just-in-time sensing: efficiently combining sonar and laser range data for exploring unknown worlds, *IEEE International Conference on Robotics and Automation*, Minneapolis, pp. 667-672.

Gonzalez, J. 1992. An iconic position estimator for a 2d laser rangefinder. *IEEE International Conference on Robotics and Automation*. pp. 2646-2651.

Hong, M.L. and Kleeman, L. 1995. A low sample rate 3d sonar sensor for mobile robots. *IEEE International Conference on Robotics and Automation*. Nagoya. pp. 3015-3020.

Jazwinski, A.H. 1970. *Stochastic processes and filtering theory*, New York: Academic Press.

Julier, S., Uhlmann, J. and Durrant-Whyte, H. 1995 A new approach for filtering nonlinear systems, *Proceedings of the 1995 American Control Conference*, Seattle, Washington, pp. 1628-1632.

- Kleeman, L. and Kuc, R. 1995. Mobile robot sonar for target localization and classification. *International Journal of Robotics Research*, 14(4):295-318.
- Kuc, R. and Siegel, M.W. 1987. Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 9(6):766-778.
- Leonard, J.J. and Durrant-Whyte, H.F. 1991. Simultaneous map building and localisation for an autonomous robot. *IEEE/RSJ International Workshop on Intelligent Robots and Systems*. pp. 1442-1447.
- Leonard, J.J., Durrant-Whyte H.F. and Ingemar J.C. 1992. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*. 11(4):286-298.
- Lim, J.H. and Cho, D.W. 1993. Experimental investigation of mapping and navigation based on certainty grid using sonar sensors. *Robotica* 11(1):7-17.
- Lu, F. and Miliotis, E.E. 1995. Optimal global pose estimation for consistent sensor data registration. *IEEE International Conference on Robotics and Automation*. pp. 93-100.
- Moutarlier, P. and Chatila, R. 1989. Stochastic multisensory data fusion for mobile robot location and environment modeling. *5th International Symposium on Robotics Research*. Tokyo. pp. 85-94.

Moravec, H.P. and Elfes, A. 1985. High resolution map from wide-angle sonar. *IEEE International Conference on Robotics and Automation*. pp. 116-121.

Neira, J. et al. 1996 Multisensor mobile robot localisation. *IEEE International Conference on Robotics and Automation*. pp. 673-679.

Ohya, A., Nagashima, Y, and Yuta, S. 1994. Exploring unknown environment and map construction using ultrasonic sensing of normal direction of walls, *IEEE International Conference on Robotics and Automation*. pp. 485-492.

Oriolo, G., Vendittelli M. and Ulivi, G. 1995. On-line map building and navigation for autonomous mobile robots. *IEEE International Conference on Robotics and Automation*. pp. 2900-2906.

Pagac, D., Nebot, E. M. and Durrant-Whyte, H. 1996. An evidential approach to probabilistic map-building. *IEEE International Conference on Robotics and Automation*. pp. 745-750.

Rencken, W.D. 1993. Concurrent localisation and map building for mobile robots using ultrasonic sensors. *IEEE International Conference on Robotics and Automation*. pp. 2192-2197.

Smith, R.C. and Cheeseman, P. 1986. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*. 5(4):.56-68.