

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Robot Collaboration for Simultaneous Map Building and Localization

M. Oussalah and D. Wilson

*University of Birmingham, Electronics, Electrical and Computer Engineering
Edgbaston, Birmingham B15 2TT, UK*

1. Introduction

Collaboration has been acknowledged as an excellent tool to compensate for limitations and shortcoming of individuals in order to achieve complex tasks. Yet, robotics collaboration has been recognized as an independent entity on its own within robotics community as suggested by the emerging literature and the growing applications like RoboCup, FIRA competitions (Kitanev, 1997), autonomous vehicles for space/submarine exploration (Todd and Pomerleau, 1996). This promises a leading future for this field in a medium term. However, the development of effective collaboration schemes is subject to several challenges. This concerns aspects related to robot localization (absolute and/or relative localization), environment map building, sensor modelling and fusion, game-theoretic scenarios, collaboration/cooperation modes, user's interface and control modes, among others, see, for instance, (Mataric, 1998). This chapter aims to contribute at least to the first two aspects of the aforementioned challenges where the issue of dynamic localization and map building using two miniature Khepera® robots is tackled. An extended-Kalman filter based approach is developed and implemented in order to model the state of the robot and various observations as well as to determine and update the positioning estimates of both robots together with the identified landmarks in the environment. A virtual representation of the map and robots is also put forward using OpenGL for 3D representation. While the developed interface uses enhanced help capabilities in case of unsafe or non-tolerated manipulations by the user.

The issue of mobile localization and map building has been a challenging issue that faced the robotics community since the eighties due the debatable issues related to the state and observation modelling, map initialization and building, and convergence of the estimation process, among others. This led to the development of several techniques to overcome the above challenges. Since the pioneering work of Smith and Cheesman (1986), a bridge from geometrical features and stochastic models has been established, which led to a variety of algorithms, mainly using Kalman filter (Gib, 1986) or its variants, whose feasibility and satisfactory performances have been demonstrated both from theoretical and practical perspectives through the convergence properties of the algorithms and the successful applications.

The concept of robot localization and map building is often referred to as SLAM (Simultaneous Localization And Mapping), in which both the environment map represented as a set of landmarks and the robot states are estimated simultaneously by augmenting the state vector to

include both robot's state and landmark states (Leonard and Durrant-White, 1991a; Dissanayake et al., 2000, 2001; Thrun et al., 1998; Bailey and Durrant-White, 2006). This trivially increases the autonomy of the robot(s) as it permits consistent robot navigation without requiring a priori map. Besides, the study and experiments carried out in the above citations, among others, demonstrated the feasibility of SLAM both from theoretical and practical viewpoints despite the challenging issues related to complexity, map building and data association, etc. In contrast to the stochastic models which govern the construction of the majority of the proposed SLAM models, one shall mention the increasing literature in soft-computing based approaches like fuzzy/possibility-based approach (Oussalah et al., 2003), neural network (Nonato et al., 2006), genetic algorithms (Higuchi, 1996), inference systems (Begum et al., 2005). Reference (Murphy, 2000) provides a good overview of such methodologies.

The vast majority of SLAM implementations utilise expensive sensor equipment that has good range detection and high accuracy, typically laser range finder, ultrasonic sensors and/or vision systems. Although the use of ultrasonic sensors causes specular reflection while the ultrasonic beam deteriorates the measurement accuracy. On the other hand, the vision system induces high computational complexity, which opens new areas for research and investigations in order to achieve high balance in terms of cost-effectiveness ratio. Besides, almost all the implementations so far restricted to a single robotic system (Bailey and Durrant-White, 2006; Durrant-White and Bailey, 2006). Consequently the use of a group of Khepera robots together with the limited sensor capabilities and restricted range of infrared sensors makes the SLAM problematic even more challenging. For this purpose, similarly to (Dissanayake et al., 2000, 2001) a stochastic SLAM based approach was developed to account for the multi-robotic systems. The underlying SLAM model depends on the robot collaboration mode. For instance, in case where the vision robot restricts its movement to rotation to identify possible objects, the state vector includes both the state of both robots as well as that of landmarks. While in case where both robots achieve non-negligible movement, the state vector includes state of each robot together with each environment map constituted by its set of landmarks. The latter is made of Cartesian points, which are transformed into feature landmarks - mainly segment lines and corners-. Performances of the proposal will be illustrated through experimental setup. Section 2 of this chapter describes the overall system setup providing an overview of the system, concept of robots' collaboration, mapping and user's interface. Section 3 recalls the basis of stochastic SLAM model and develops the proposed filter estimation algorithm. Section 4 examines the experimental performances of the developed algorithm. Next overall conclusion is provided.

2. System Setup

2.1 System Overview

Two Khepera robots were used in our experiment. Both robots are equipped with encoders and infrared sensors. Besides, one of the robots is equipped with a linear vision turret, which allows detection of far away objects. One refers Vision Robot (VR) to the robot equipped with vision turret and Blind Robot (BR) to the other one. The two robots BR and VR are controlled via the two serial connections COM1 and COM2, respectively, of a regular windows-based PC platform.

Each of the robots only has 512k of memory on-board. This small amount of memory and the limited processing powers of the robots entail that almost all of the map construction and robot control are accomplished on the PC platform while a

behavioural safety oriented navigation scheme is embedded in robot platform executed by local micro-controller in order to prevent robots to crash with obstacles or objects using the infrared sensors. The user interacts with the system through the control program interface. Figure 1 shows a typical system setup using a PC with two serial extension cables and a custom made testing environment.

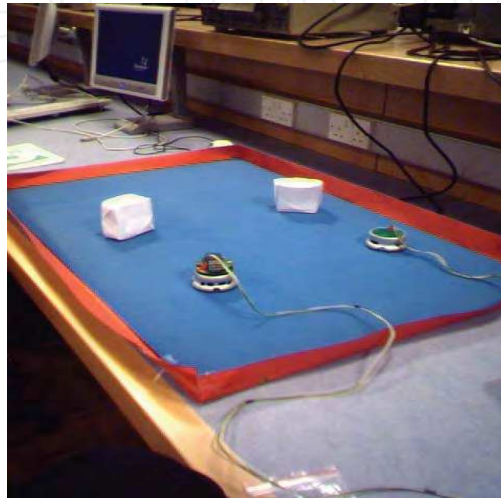


Fig. 1. Instance system setup.

2.2 Robot Collaboration: Concept

Building on the ability of the two robots to individually add objects to the map based on their tracked location, the robots are able to collaborate their efforts to achieve the shared goal of mapping the environment. This ability also helps make use of the vision turret, which one of the robots is equipped with. For example the VR can send a BR to a typical location viewed through its vision turret in order to explore the neighbourhood of the underlying object using the infrared sensor the BR is equipped with. This is especially useful when, for example, the two robots are on opposite sides of an object. So, by calling the other robot to the location of the detected object, if the other robot comes from the other side of the object, it will encounter the other edge of the object perimeter in its travels, and add that to the map accordingly. On the other hand as the range of the infrared sensors is very limited (only few centimeters), the use of vision turret whose range may go up to 1 meter allows us to compensate for such limitation. The diagram in Figure 2 below illustrates this point where the arrows show direction of robot travel.

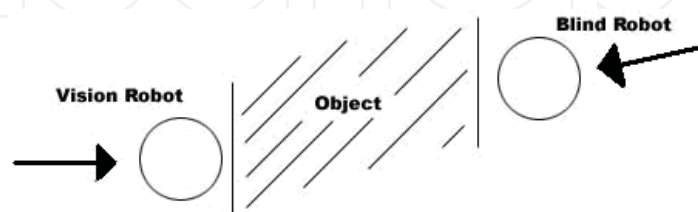


Fig. 2. Robots collaborating to map opposing edges of same object.

Especially, when the action to search for an object is sent to the VR, the latter starts turning via fixed angular increments until the vision turret detects an object, then either the robot moves toward it or sends a command to the BR to go to object location according to the determined azimuth and explore it via its infra-red sensors. The process of detecting an object with the turret is based on: i) the number of pixels (out of 64 maximum) that detected the object; ii) calculus of the target pixel, which defined the edge of the object; iii) turning the VR until the turret detects an object at the target pixel; vi) storing the orientation angle. The above process effectively ensures that of all the pixels that do not detect an object, half are on the left and half are on the right, leaving the object-detected pixels in the centre of the field of view.

2.3 Mapping

Due to low resolution of sensors equipping the robots, the SLAM approach of representing the object as virtual points in the x-y coordinates sounds appealing. However, these virtual objects, if detected by vision turret, will get further explored using infrared sensors in which the virtual points are, under some geometrical constraint, linearly fitted together, which will form the environment map. These points will act as landmarks for the robots, which will then be added to the vector state model containing the x-y and pose of the robot. As special interest, one notices the fact that the second robot will not be used as an extra object in the environment but rather will use the information of 2nd robot's positioning to update the current robot location. This builds a bridge towards the issue of relative positioning which has focused much of interest in recent years. From a geometrical perspective, as far as range sensors were used, the detected object is modelled as a rectangular object centred around the virtual point identified by the sensor and whose direction is perpendicular to sensor beam as illustrated in Figure 3.

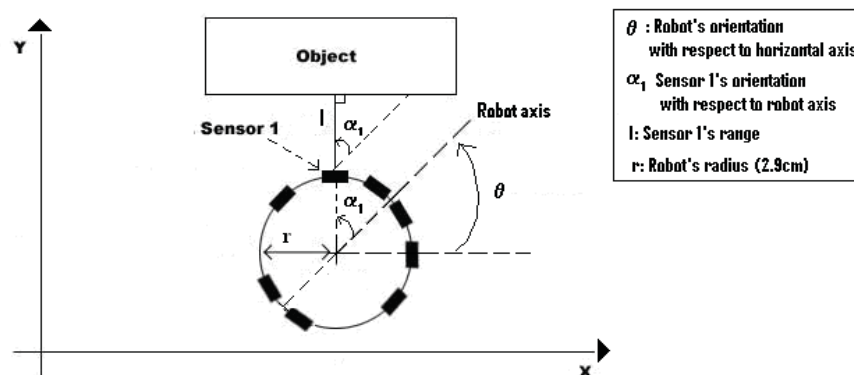


Fig. 3. Robot configuration and object modelling.

The systems mapping ability is combined with the robot localisation, which provides further evidence on whether the object needs to be added to the map or is already an observed landmark or just a false alarm that needs to be discarded. Besides, the object once added to the map will be used again in the observation model to correct the robot positioning as will be detailed later on. On the other hand, since the map contains several objects, the mapping-localization mechanism should be endowed with retrieval capabilities to search the closest

objects in the map to the current robot positioning in order to avoid using the whole set of objects, which increases substantially the complexity of the algorithm. This leads to a local map of the environment, used for updating and data association purposes. This helps in the decision making process of adding a new object to the map or not. Indeed, if, for example, the sensors are observing an object at much closer range than the closest mapped object, then the observed object is added to the map as described earlier.

With the ability to know the robots location and correctly add objects to the map around that location the system can map out the environment the robots are in. To map the perimeter of the environment a robot will travel forwards constantly checking its sensors. When a sensor detects an object any necessary adjustments are made to the map as described above, then the robot turns to align itself with the object and then continues to travel forward. At set intervals (set by the required resolution of the map and robot speed) the robots location is updated and the object detected is added to the map.

2.4 User's Interface

The interface consists of three parts: the console, the display and the menu. These can be seen in Figure 4. The console is the most diverse aspect of the interface, in that it has the most uses. Firstly the internal workings of the system can be displayed in text format using the console. This can range from simply displaying the current state of the system (such as robot coordinates and orientation), to the most recent recorded values (such as sensor, turret and odometer readings), to the actual values being calculated and used in a process. The console also allows the user to enter custom data into the system, such as providing a filename to save a map as. Aside from the need to enter filenames of maps to load or save the console can be mainly ignored for general system usage.

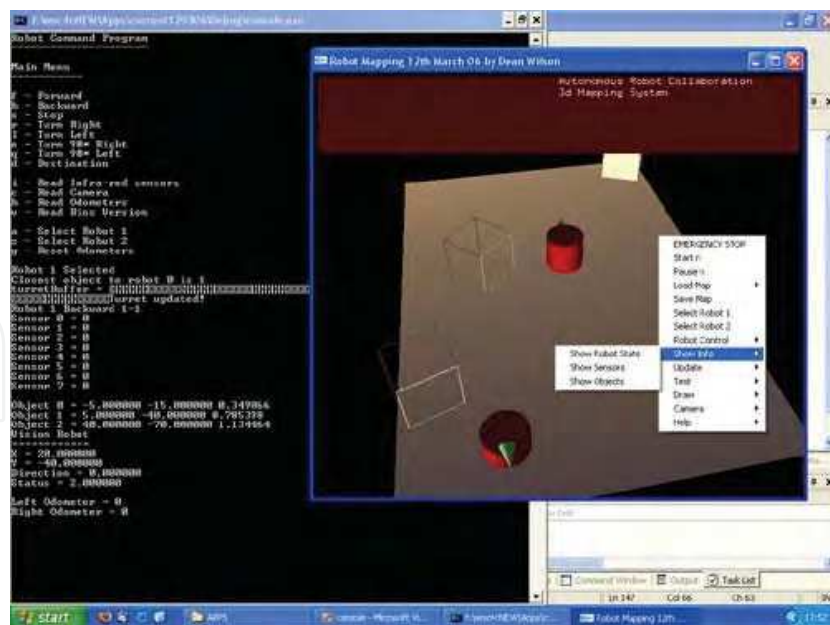


Fig. 4. Example of screenshot showing the three components of the user interface.

The display shows a 3D map which represents the environment as it is known by the system. The map includes the ground, objects and the two robots. On the other hand, the interface also includes some help functionalities in the event the user was unsure how to start using the system. It is fairly brief and only covers the most common mishaps. The user can also display the system credits, which states the program author and completion date.

2.5 Optimal Resource Allocation

Due to discrepancy between processing capability of PC and robot's controller, it was necessary to take this into account when designing the interaction between host PC and robots. This can be ensured by using a delay function to ensure the sensors are being queried at reasonable time. Besides to avoid inconsistency in requesting information from different robot's sensors, another small delay of around 0.05 seconds between sending a request for a sensor or odometer update and reading in characters from the receive buffer (reading the response) is inserted. The turret returns many more characters, so it was necessary to use such delay, anything less and some of the characters do not get received.

To allocate the best use of available resources, the 'reading' process was split into 'update' and 'show'. Rather than have the program poll the robot continually every time it wanted to make a decision, the readings are updated once and then stored on the PC. The program can then access these stored readings as many times as it wants, as fast as it wants without putting further strain on the robots. 'Show' refers to accessing these stored readings, whereas 'update' refers to polling the robots to update the stored readings with current data. Obviously the update process needs to be called periodically before the stored readings get too out of date. This design improved system efficiency greatly. It also allows the system to fully analyse a specific time index before moving onto the next. For example when checking the sensors for object detection the stored sensor readings can be updated once. An individual analysis of the reading of each of the sensors at that time index can then be made, and any necessary processing done. A separate update process eliminates the need to poll the robot once for each of the 8 sensors, the polling of which would incur a 0.05second delay for each sensor.

3. Kalman filter and SLAM models

The aim of this section is to investigate the stochastic models underlying the SLAM or simultaneous robot localization and map building. First let us describe the standard Kalman filter approach without recourse to SLAM.

3.1 State model

Using the incremental moving l_k^r and l_k^l of the right and left wheel, respectively, obtained by reading the encoder sensor of the robot, one can estimate the pose of the robot given in term of x-y coordinate of a reference point in the robot, usually taken as the centre of the robot and the orientation of the robot with respect to horizontal axis as it can be seen in Figure 3. The prediction model giving the state of the robot $(x_k, y_k, \theta_k)^T$ based on previous state $(x_{k-1}, y_{k-1}, \theta_{k-1})^T$ and the incremental encoder readings is given by the expression:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \frac{l_k^r + l_k^l}{2} \cos \theta_k \\ y_k + \frac{l_k^r + l_k^l}{2} \sin \theta_k \\ \theta_k + \frac{l_k^r - l_k^l}{E} \end{bmatrix} + \eta_k \quad (1)$$

where η_k stands for Gaussian zero-mean noise pervading the state components x , y and θ ; that is, $\eta_k \mapsto N([0 \ 0 \ 0]^T, Q)$, where Q is a 3x3 noise variance-covariance matrix, usually taken as a fixed symmetric definite matrix. E is the distance between the wheels (left and right wheels) of the robot. Expression (1) assumes that the robot trajectory is linear between two consecutive time increments k and $k+1$, while the incremental moving of θ_k is assimilated to an arc of circle.

One designates $X_R(k) = [x_k \ y_k \ \theta_k]^T$ the state vector of the robot positioning. So, (1) can be rewritten as

$$X_R(k+1|k) = F_k(X_R(k|k)) + \eta_k \quad (2)$$

The quantity $F_k(X_R(k|k))$ represents the prediction of the estimate on X_R denoted $\hat{X}_R(k+1|k)$. Due to randomness pervading the estimation of X_R expressed in the form of additive Gaussian noise with known statistics (zero mean and Q variance-covariance matrix), the entity $X_R(k+1|k)$ is attached a Gaussian distribution probability with mean $\hat{X}_R(k+1|k)$ and variance-covariance matrix

$$P_{k+1|k} = \nabla F \cdot P_{k|k} \cdot \nabla F^T + Q. \quad (3)$$

Where ∇F indicates the Jacobian (with respect to x_k , y_k and θ_k) of the state transition function F , i.e.,

$$\nabla F = \begin{bmatrix} \frac{\partial x_{k+1}}{\partial x_k} & \frac{\partial x_{k+1}}{\partial y_k} & \frac{\partial x_{k+1}}{\partial \theta_k} \\ \frac{\partial y_{k+1}}{\partial x_k} & \frac{\partial y_{k+1}}{\partial y_k} & \frac{\partial y_{k+1}}{\partial \theta_k} \\ \frac{\partial \theta_{k+1}}{\partial x_k} & \frac{\partial \theta_{k+1}}{\partial y_k} & \frac{\partial \theta_{k+1}}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{l_k^d + l_k^g}{2} \sin \theta_k \\ 0 & 1 & \frac{l_k^d + l_k^g}{2} \cos \theta_k \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

So, $X_R(k+1|k) \mapsto N(\hat{X}_R(k+1|k), P_{k+1|k})$

Observation Model

The exteroceptive sensors of the robot consist of infrared range sensors and vision turret (for only one of the robots). Therefore, the observation consists of the range -distance d_i from the sensor location within the robot platform to the i^{th} object (whose x - y coordinates are (x_{B_i}, y_{B_i})) while the information issued from the vision sensor can be translated into the azimuth β_i indicating the pose of the object with respect to the horizontal axis. Notice that the distance d_i can also be measured from the centre of robot as suggested by Figure 3 due to knowledge of radius r of the robot. Now relating the state variables to the observation

leads to the following expression of the observation model

$$d_i(k) = \sqrt{(x_k - x_{B_i})^2 + (y_k - y_{B_i})^2} + r + v_1(k) \quad (5)$$

$$\beta_i = a \tan\left(\frac{y_{B_i} - y_k}{x_{B_i} - x_k}\right) - \phi_k + v_2(k) \quad (6)$$

Or in matrix formulation

$$\begin{bmatrix} d_i(k) \\ \beta_i(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_k - x_{B_i})^2 + (y_k - y_{B_i})^2} + r \\ a \tan\left(\frac{y_{B_i} - y_k}{x_{B_i} - x_k}\right) - \phi_k \end{bmatrix} + v_k \quad (7)$$

Or, more generally,

$$z_i(k+1) = H_k(\hat{X}_R(k+1|k)) + v_k \quad (8)$$

The set of all measurements available at current time $k+1$ is denoted by $Z(k+1) = (z_1(k+1) \dots z_n(k+1))$, where n stands for the total number of observations at time $(k+1)$.

Similarly, $v_k \mapsto N([0 \ 0]^T, R)$, where R is a 2×2 noise variance-covariance matrix, usually taken as symmetric definite matrix.

It should be noticed that not both measurement equations are used necessarily simultaneously due to possible non-availability of either distance reading or camera reading. In such case, one only uses either v_1 or v_2 noise expressions, which are one-dimensional entities.

Kalman filter or extended Kalman filter (in case of nonlinear state or measurement equation) aims at finding the estimation $\hat{X}_R(k+1|k+1)$ of the robot's state $X_R(k+1|k+1)$ of the current state of the vehicle given the set of measurements. This is typically given as the expectation given the set of observation Z , i.e., $\hat{X}_R(k+1|k+1) = E[X_R(k+1|k+1) | Z]$. The uncertainty on such estimation is provided by the state variance-covariance matrix $P_{k+1|k+1}$, given as covariance on error of estimate:

$$P_R(k+1|k+1) = E[(X(k+1|k+1) - \hat{X}_R(k+1|k+1))^T (X(k+1|k+1) - \hat{X}_R(k+1|k+1)) | Z].$$

These entities are determined using Kalkan filter equations, which proceeds recursively in two stages –prediction and update- whose expressions are given below:

$$P_R(k+1|k) = \nabla F.P_R(k|k).\nabla F^T + Q_k \quad (\text{prediction of state covariance matrix}) \quad (9)$$

$$\hat{X}_R(k+1|k) = F(\hat{X}_R(k|k)) \quad (\text{prediction of state vector}) \quad (10)$$

$$S_R(k+1) = \nabla H.P_R(k+1|k).\nabla H^T + R \quad (\text{variance-covariance of innovation matrix}) \quad (11)$$

$$K_R(k+1) = P_R(k+1|k).\nabla H^T.S_R^{-1}(k+1) \quad \text{Gain matrix} \quad (12)$$

$$P_R(k+1|k+1) = P_R(k+1|k) - K_R(k+1).\nabla H.P_R(k+1|k) \quad (\text{state covariance update}) \quad (13)$$

$$\hat{X}_R(k+1|k+1) = \hat{X}_R(k|k) + K(k+1).(Z(k+1) - H(\hat{X}_R(k+1|k))) \quad (\text{state update}) \quad (14)$$

Where ∇H represents the Jacobian of the measurement equation H , which in case that both distance and landmark location were used, is given by

$$\nabla H = \begin{bmatrix} \frac{\partial d_i}{\partial x_k} & \frac{\partial d_i}{\partial y_k} & \frac{\partial d_i}{\partial \theta_k} \\ \frac{\partial \beta_i}{\partial x_k} & \frac{\partial \beta_i}{\partial y_k} & \frac{\partial \beta_i}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} -\frac{x_{B_i} - x_k}{\sqrt{(x_{B_i} - x_k)^2 + (y_{B_i} - y_k)^2}} & -\frac{y_{B_i} - y_k}{\sqrt{(x_{B_i} - x_k)^2 + (y_{B_i} - y_k)^2}} & 0 \\ \frac{y_{B_i} - y_k}{(x_{B_i} - x_k)^2 + (y_{B_i} - y_k)^2} & -\frac{1}{(x_{B_i} - x_k)^2 + (y_{B_i} - y_k)^2} & -1 \end{bmatrix} \quad (15)$$

In case where only one single type of observation is available, then one uses on a single row of matrix ∇H .

So, the estimation of $X_R(k+1|k+1)$ follows the Gaussian probability distribution $N(\hat{X}_R(k+1|k+1), P_R(k+1|k+1))$.

Kalman equations (9-14) are recursive, so they only depend on previous state. At time 0 where initially no observations were made, one requires an initial guess of the state vector and variance-covariance matrix, $X_R(0|0)$ and $P_R(0|0)$, respectively, this allows us to determine the new estimate $X_R(1|1)$ and $P_R(1|1)$ given the observation vector $Z(1)$.

Notice that since the measurements are usually sampled at lower rate than encoders (almost five to 10 times less), the prediction equations (9) and (10) are applied several times before calling up for update stage using expressions (11-14).

3.2 SLAM mode

The preceding development of Kalman filter model assumes that the landmarks (observed objects) and robot positioning are independent. For instance, if the absolute locations of landmarks are fully known, then the previous Kalman filter approach does make sense. However, in reality, as far as the construction of global map of environment is concerned and no absolute knowledge of the landmark location is priori given, the estimations of landmarks positioning are correlated and strongly influenced by the uncertainty pervading the robot's location. Indeed, as the robot moves forth and back through the environment, the uncertainty pervading the landmarks' locations will be influenced and since the overall set of landmarks are linked through geometrical entities like wall, corners, etc, such uncertainty would propagate through overall set of landmarks. On the other hand, as all the observations (landmarks) are implicitly linked to robot state such uncertainty would also affect the robot state estimate X_R . This has given rise to the idea of simultaneous mapping and localization using estimation-theoretic methods known as SLAM. Work by Smith and Cheesman (1986) and Durrant-White (1988) established a statistical basis for describing relationships between landmarks and manipulating geometric uncertainty. Smith et al. (1990) established that if a mobile robot is moving through unknown environment and taking relative observations of landmarks, then the estimates of these landmarks are all necessarily correlated with each others because of common error in estimated robot location. As a result of this, a consistent estimation would require a joint state composed of both robot state and each landmark position leading to an augmented state vector. However as a result of increasing number of landmarks, the dimension of such state vector increases accordingly, which often induces further challenges in terms of computational complexity, convergence behaviour, conflict resolution, among others (Durrant-White and Bailey, 2006; Martinili et al., 2003).

More specifically, let $X_{L_i} = (x_{L_i}, y_{L_i})^T$ be the state of the i^{th} feature or landmark given in terms of x-y Cartesian coordinates. First, one assumes the environment be static. This assumption is very common and trivial if the objects are not dynamic. Indeed, tracking moving objects is not considered of much value for the navigation purpose. So, the dynamic model that includes both landmark and robot's state becomes

$$\begin{cases} X_R(k+1|k) = F(X_R(k|k)) + \eta_k \\ X_L(k+1|k) = X_L(k|k) \end{cases} \quad (16)$$

Where $X_L = ((x_{L_1} \ y_{L_1}) \ (x_{L_2} \ y_{L_2}) \dots (x_{L_N} \ y_{L_N}))^T \in \mathfrak{R}^{2N}$ represents the set of all N landmarks identified up to current time. Loosely speaking, in some literature N is set as an arbitrary total number of landmarks that may exist in the environment, while it is common that the value of N varies within time due to update of the environment and addition of new landmarks to the map. So, the new state vector will be $X = (X_R, X_L)^T \in \mathfrak{R}^{2N+3}$.

The augmented state transition model for the complete system can now be written as

$$\begin{bmatrix} X_R(k+1|k) \\ x_{L_1} \\ y_{L_1} \\ \vdots \\ x_{L_N} \\ y_{L_N} \end{bmatrix} = \begin{bmatrix} F_k & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} X_R(k|k) \\ x_{L_1} \\ y_{L_1} \\ \vdots \\ x_{L_N} \\ y_{L_N} \end{bmatrix} + \begin{bmatrix} \eta_k \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (17)$$

Accordingly, the new state Jacobian matrix ∇F^e (one denotes F^e for extended state transition F) will be

$$\nabla F^e = \begin{bmatrix} \nabla F & 0_{3 \times N} \\ 0_{N \times 3} & I_{2N \times 2N} \end{bmatrix} \quad (18)$$

Where $0_{3 \times N}$ stands for $3 \times N$ zero matrix, similar definitions hold for $0_{N \times 3}$ and $I_{2N \times 2N}$.

The new observation model can be written

$$z_i(k+1) = H_k(\hat{X}_R(k+1|k)) + 0_{N \times 1} X_L + v_k \quad (19)$$

Similarly, the new Jacobian ∇H^e of the observation model reads as (assuming that only a single landmark is observed at a given time):

$$\nabla H^e = \begin{bmatrix} \frac{\partial H}{\partial X_R} & 0_{2 \times 1} \dots 0_{2 \times 1} & \frac{\partial H}{\partial (x_{L_i}, y_{L_i})} & 0_{2 \times 1} \dots 0_{2 \times 1} \end{bmatrix}, \quad (20)$$

For example in the case of both (5-6) were used, we have

$$\nabla H^e = \begin{bmatrix} -\frac{x_{B_i} - x_k}{\Delta_i} & -\frac{y_{B_i} - y_k}{\Delta_i} & 0 & 0 \dots 0 & \frac{x_{B_i} - x_k}{\Delta_i} & -\frac{y_{B_i} - y_k}{\Delta_i} & 0 \dots 0 \\ \frac{y_{B_i} - y_k}{\Delta_i^2} & -\frac{1}{\Delta_i^2} & -1 & 0 \dots 0 & -\frac{y_{B_i} - y_k}{\Delta_i^2} & \frac{1}{\Delta_i^2} & -1 & 0 \dots 0 \end{bmatrix} \quad (21)$$

With $\Delta_i = \sqrt{(x_{B_i} - x_k)^2 + (y_{B_i} - y_k)^2}$

(21) can also be rewritten as

$$\nabla H^e = [H_1 \ \emptyset_1 \ H_2 \ \emptyset_2] \quad (22)$$

Where \emptyset_1 and \emptyset_2 stand for all null elements located in (20) or (21). Notice that most elements of both ∇F^e and ∇H^e are null elements.

From implementation perspective of the (extended) Kalman filter in the sense of expressions (9-13), a naïve implementation consists to compute the predicted state variance-covariance:

$$P_{k+1|k} = \nabla F^e P_{k|k} \nabla F^{eT} + Q \quad (23)$$

Strictly speaking the above operation induces a cubic complexity in the number of landmarks. However, intuitively since only the robot state variables are involved in the observation, the covariance should be simplified accordingly. For this purpose, by distinguishing parts related to robot state and those linked to landmark state in matrix P as

$$P = \begin{bmatrix} P_R & P_{RL} \\ P_{RL}^T & P_L \end{bmatrix}, \quad (24)$$

so the prediction stage (23) boils down to

$$\begin{aligned} P_{k+1|k} &= \begin{bmatrix} \nabla F & \emptyset \\ \emptyset^T & 1 \end{bmatrix} \begin{bmatrix} P_R(k) & P_{RL}(k) \\ P_{RL}^T(k) & P_L(k) \end{bmatrix} \begin{bmatrix} \nabla F^T & \emptyset^T \\ \emptyset & 1^T \end{bmatrix} + \begin{bmatrix} Q & \emptyset \\ \emptyset & \emptyset_1 \end{bmatrix} \\ &= \begin{bmatrix} \nabla F P_R(k) \nabla F^T & \nabla F P_{RL}(k) \\ (\nabla F P_R(k))^T & P_L(k) \end{bmatrix} + \begin{bmatrix} Q & \emptyset \\ \emptyset & \emptyset_1 \end{bmatrix} \end{aligned} \quad (25)$$

It has been shown that the evaluation of this matrix requires approximately $9(N+3)$ multiplications (Guivant and Neboit, 2001).

Similarly, in the updating stage, by rewriting $P_k \nabla H^{eT} = P_R H_1^T + P_L H_2^T$ leads to a cost, which is proportional to $(N+3)$, so the evaluation of the covariance update is $\sim O(N^2)$.

Moreover, it has been shown that it is not necessary to perform full SLAM update when dealing with a local area. So, the complexity can even get substantially reduced accordingly. More formally, assuming the state vector is divided as $X = [X_A \ X_B]^T$ with $X_A \in \mathfrak{R}^{N_A+3}$ and $X_B \in \mathfrak{R}^{N_B+3}$, $N = N_A + N_B$ (Guivant and Nebo, 2001). The states X_A can be initially selected as the state of all landmarks located in the neighborhood of the vehicle in addition to the three states of the vehicle, while X_B corresponds to the states of all remaining landmarks. The hint is that at a give time, the observations are only related to X_A . Accordingly,

$$\nabla H^e = \begin{bmatrix} \frac{\partial H}{\partial X_A} & \frac{\partial H}{\partial X_B} \end{bmatrix} = [H_A \ \emptyset] \quad (26)$$

Consequently, given $P = \begin{bmatrix} P_A & P_{AB} \\ P_{AB}^T & P_B \end{bmatrix}$, one induces

$$S = H P H^T + R = H_A P_{AA} H_A^T + R \quad (27)$$

And the filter gain

$$W = \begin{bmatrix} P_{AA} H_A^T S^{-1} \\ P_{AB}^T H_A^T S^{-1} \end{bmatrix} = \begin{bmatrix} W_A \\ W_B \end{bmatrix} \quad (28)$$

In other words, the innovation matrix and matrix gain W_A are independent of remaining landmarks X_B . When the vehicle departs from this local area the information will be propagated to global landmark. So, the entities X_B , P_{AB} and P_{BB} will only be determined when the vehicle moves away from the local area. It has been shown that the complexity of update in such case is of order $O(N_A^2)$ and since N_A is in general much smaller than N_B , the gain in terms of complexity becomes significant. This reduction method is known as compressed (extended) Kalman filter in (Guivant and Nebo, 2001). Williams (2001) has put forward the Constrained Local Submap Filter approach in which both the relative state of each landmark with respect to local map as well as its global coordinate with respect to the global map are carried out. The method maintains an independent, local submap estimate of the features in the immediate vicinity of the vehicle.

An ultimate problem which arises from the above submapping is the selection of the local area. Several approaches have been investigated for such purpose. One conventional approach consists of dividing the global map into rectangular regions with size at least equal to the range of the external sensor. So, at each position, one may consider for instance the eight or twenty fourth neighbouring cells as suggested in (Guivant and Nebo, 2001).

In the context of our work, we rather adopted an approach close to that developed by Dissanayake et al. (2001). In this course, given a time interval h_T , a two-stage selection process is carried out:

- First, one maintains all landmarks that have been seen by the vehicle within the time interval h_T . Alternatively, authors in (Dissanayake et al., 2000) used a predefined distance travelled by the vehicle.
- Next, among the above set of landmarks, one selects only those, which are the most informative in the sense of landmark variance-covariance matrix. For this purpose, the reciprocal of the trace of such variance-covariance matrix was used as a tool to evaluate the extent of the information content. Consequently, from the set of landmarks, only those landmarks whose information content in the above sense is beyond some threshold are considered. The value of the threshold is here taken to be a function of the information content associated to the fully defined prior landmarks concerning the border of the environment as will be pointed in the map initialization section.

3.3 Convergence properties

As far as the construction of the submap is concerned, the aspect of convergence becomes crucial. From theoretical perspective, some appealing results have been reported by Dissanayake et al. (2001). Especially given the decomposition (24), it has been proven that

- i) The determinant of any submatrix of the map covariance matrix P_L decreases monotonically as successive observations are made
- ii) In the limit case (at time infinity), the determinant of P_R tends towards zero, so the landmark estimates become fully correlated.
- iii) In the limit case, the lower bound on the covariance matrix associated with any single landmark is determined only by the initial covariance of the vehicle estimate P_R .

The above testifies on the steady state behavior of the convergence of the landmark estimates. Especially, it stresses that as the vehicle moves on the environment the

uncertainty pervading the landmark estimations reduces monotonically. The estimation of any pair of landmarks becomes fully correlated in the sense that if one landmark is known with full certainty, the others can be known with full certainty too. The individual landmark variances converge toward a lower bound determined by initial uncertainties in vehicle position and observations as indicated by matrices $P(0|0)$ and R .

On the other hand, Julier (2003) has investigated the effect of adding noise to the long term behaviors of SLAM and has shown that:

- i) If the steady state covariance will not be degraded, the computational and storage cost increase linearly with the number of landmarks in the map;
- ii) Even if the steady state covariance is preserved, local performance can be unacceptably high;
- iii) If the solution causes the steady state covariance to degrade, the addition can only be a finite number of times.

This entails that it is more appropriate to maintain the full correlation structure of all landmarks within the operational area of the vehicle.

On the other hand, from the observability perspective, it has been shown that the Riccati equation in P (that follows from update expression (13)), e.g., (Andrade-Cetto and Sanfeliu, 2004), which can be rewritten as:

$$P = \nabla F.(P - P.\nabla H^T (\nabla H.P.\nabla H^T + R)^{-1}.\nabla H.P).\nabla F^T + Q$$

converges to a steady state covariance only if the pair $(\nabla F, \nabla H)$ is fully observable. In addition if the pair $(\nabla F, I)$ is fully controllable, then the steady state covariance is a unique positive definite matrix, independent of the initial covariance $P(0|0)$.

3.4 Map Initialization

Initialization is required to infer the number of landmarks N as well as their x-y coordinates, which will be used in the SLAM model. Several studies have explored the initialization of the map through sensor scan, using, for instance, sonar-like measurements (Chong and Kleeman, 1999; Ip and Rad, 2004), which an initial value of landmarks. While other studies assumed the initial map is initially empty, and as soon as an observation gets reinforced by other observations, it will be promoted to a landmark (Dissanayake et al., 2001). Both approaches can be used in our study. Indeed, the use of initial mapping using a single sensor can be accomplished using the vision sensor. So, in the light of the emerging works in the bearing-only SLAM, one can think of the robot using a single rotation at discrete sample intervals, repeated at two different robot's locations, would allow us in theory to determine initial set of landmarks. However, the data association problem in such case becomes difficult. While the second approach is trivially straightforward where the initial state vector reduces to robot state vector. In our study, in order to make use of the geometrical environment constraints at one hand, and on the other hand, avoid the nontrivial data association problem due to the limited sensory perception, one assumes that the boundary of the environment is fully known. Consequently, the four corners of the rectangular environment are taken as fully determined landmarks. This also allows us to set up a geometrical consistency test in the sense that as soon as the perceived landmark is located beyond the border limit, it is systematically discarded.

Therefore, initially, the state vector is $X = [X_R \ x_{L_1} \ y_{L_1} \ x_{L_2} \ y_{L_2} \ x_{L_3} \ y_{L_3} \ x_{L_4} \ y_{L_4}]^T$

3.5 Data Association and Map building

Data association has always been a critical and crucial issue in practical SLAM implementations. That is because it governs the validation of the new landmarks and the matching of the observation(s) with the previously validated landmarks. On the other hand, an incorrect association of the observation to the map can cause the filter to diverge. Given the knowledge of the geometrical boundary of the environment, two validation tests are carried out:

- *Geometrical validation test*: This is a basic check to test whether the location of the observation is within the environment boundary. This is mainly meant to remove possible outliers and noise measurement observations.
- *Statistical validation test*: This uses the statistical properties of the observations as well as landmarks as a tool to achieve the matching. Especially, the nearest neighbour association is taken as the closest association in statistical sense. For this purpose, one first needs to translate the range/bearing observation into landmark locations. In case where measurement coincides with range measurement, e.g., $z(k) = r_L$, we have

$$\begin{cases} x_{L_z} = x_1(k) + r_L \cdot \cos(\theta_k + \alpha_j) \\ y_{L_z} = y_1(k) + r_L \cdot \sin(\theta_k + \alpha_j) \end{cases} \quad (29)$$

With $X_R(k) = [x_1(k) \ x_2(k) \ \theta_k]^T$ and α_j stands for the azimuth of the j th robot's sensor that detected the underlying landmark, with respect to robot axis. Putting (29) in matrix formulation as $X_{L_z} = T(X_R(k), r_L)$, the variance-covariance of the landmark estimate is given by

$$P_{L_z} = \nabla T_{x_1 y_1 \theta_k} \cdot P_R \cdot \nabla T_{x_1 y_1 \theta_k}^T + \nabla T_{r_L} \cdot R \cdot \nabla T_{r_L}^T \quad (30)$$

Where R stands for the $z(k)$'s covariance, and P_R for the (updated) robot state vector variance-covariance matrix as determined by the filter.

Now given the vector X_{L_i} symbolized by (29) and given a set of confirmed landmarks $(L_1, P_{L_1}), \dots, (L_m, P_{L_m})$, where (L_i, P_{L_i}) stands for the first and second statistics of the i th landmark, the measurement $z(k)$ is associated with the j th landmark if:

$$(X_{L_z} - L_j)^T (P_{L_z} + P_{L_j})^{-1} (X_{L_z} - L_j) \leq d_{\min} \quad (31)$$

where d_{\min} is some validation gate. The threshold d_{\min} can be determined by noticing that the left hand side part of the inequality in (31) is χ^2 distributed, so by choosing the null hypothesis and the confidence level, the d_{\min} value is straightforward.

Therefore, if the above condition holds only for one single landmark, then the underlying observation $z(k)$ is associated with that landmark. Otherwise, if the inequality holds for more than one landmark, the observation is then omitted, meaning that under the current level of confidence, the statistical test cannot lead to a matching. Obviously, it is still possible to narrow the confidence level such that the validation gate d_{\min} decreases, which may result in resolving the conflict among the possible candidates.

On the other hand, if the above inequality cannot hold indicating that there is no landmark that may match the current observation, then such observation can be considered as a new landmark. Once the validated landmarks are constituted of a set of Cartesian points, a

geometrical fitting allows us to group these features into high level geometrical entities constituted of two main feature landmarks: line segment, if the underlying points are sufficiently aligned up, and corner. The former is modelled by the extreme points of the segment while the corner by the x-y coordinates of the underlying Cartesian point.

In other studies, e.g., (Guivant and Neboit, 2001), a non-matched observation will be treated as a potential landmark (so maintaining at each time a set of confirmed landmarks and a set of tentative landmarks), and will not be promoted to a confirmed landmark until sufficient number of observations are found matching this potential landmark in the above statistical sense. Strictly speaking such reasoning cannot be applied in our experiment due to the lack of redundant data and limited navigation tasks. Therefore, as soon as the statistical test fails for all set of landmarks, the new observation is automatically promoted to a confirmed new landmark, unless the geometrical validation test fails as well in which case, the observation is fully ignored. Adding a new landmark to the new set of already confirmed landmarks will obviously result in an augmented state vector.

3.6 Discussions

- It should be noted that the above data association reasoning relies heavily on the range of sensors because the information about the landmark location can be directly inferred as according to (29). However, the use of bearing sensor would be beneficial if the two robots were equipped with vision turret. In this course, the use of bearing information from two different robot locations would allow us to infer the x-y coordinate of the associated landmark, assuming the data association is very simple in the sense that at each robot location the sensory information identified the same object, which in reality is not always the case.
- The representation of the landmark in this study is made easy by choosing a Cartesian point as a geometric primitive, which are later combined to form more generic feature like segment line and corner. However, such choice, even if it is motivated by the limited sensory modality of Khepera, can also be questioned. Dufourd and Chatila, (2004) provide a comparison of space-based, grid-based and feature based map formats. Lisien et al. (2003) suggested to combine topological and feature based mapping where topological methods are used for planning feature based mapping. This leads to what is referred to as hierarchy SLAM.
- The restriction concerning the validation of new landmark using only geometrical and statistical tests is also shown to be limited. Indeed, it can make sense for more point-based-landmarks but it is difficult to be justified for more realistic geometric patterns. Several studies using SLAM with range-only sensors (Leonard et al., 2003) and bearing-only sensors (Lemaire et al., 2005; Deans and Hebert, 2000) proved that a single measurement is insufficient to constrain landmark location, instead several observations are necessary to confirm or delete the tentative landmark.
- The use of Mahalanobis distance as in (31), even it has proven to be successful, can also be questioned. Alternatives include Multiple hypotheses tree using Bayes' theorem, but, this raises the complexity of the algorithm due to the cost of maintaining separate map estimates for each hypothesis, and the pruning decision. Montemerlo and Thrun (2003) suggested a fast SLAM algorithm based on the idea of exact factorization of posterior distribution into a product of conditional

landmark distributions and a distribution over robot paths. Instead of geometrical feature landmarks, Nieto et al. (2005) has suggested a methodology to deal with features of arbitrary shapes, where each landmark is defined by a shape model incorporating an embedded coordinate frame, so, the map is constituted of a set of landmark frame locations. Eliazar and Parr (2004) advocated the use of grid cell representation in conjunction with particle filter. Nieto et al., (2004) also used occupancy grid structure where each grid-cell is determined by a set of local landmarks in the overall SLAM map. The data association problem in such cases boils down to ambiguity in cell allocation. The latter can be solved by Bayes' like approach. However, the suggested algorithmic representation sounds very context dependent. Also, the choice of grid cell posterior as well as adapting the number of cells is very debatable in the literature. Wijesoma et al. (2006) advocated the use of optimization problem-based approach where the data association is formulated as a generalized discrete optimization problem where the cost function is constructed from joint likelihood of measurements in multiple frames and features. The minimization is subject to some environmental and rational constraints.

- The issue of landmark selection in suboptimal filtering as detailed is very debatable as well. Indeed, this boils down to the difficult trade-off of maintaining sufficient representation of the map which allows good estimates of robot pose versus reducing the map size to its nominal representation in order to reduce the computational complexity. Indeed, the crucial question is how much should we be looking back into the past such that all the visited landmarks will be maintained? Typically, there is no exact answer to this question as it is very much context dependent; that is, it requires knowledge of how often the vehicle visits the already perceived landmarks. The aspect of information content discussed in previous section requires also further analysis. Indeed, we adopted, similarly to Dissanayake et al. (2001), the reciprocal of the trace of the covariance matrix. However, other alternatives are also possible. This includes, for instance, Shannon entropy, Fisher entropy, among others.
- The map initialization adopted in this study contrasts with alternatives approaches in which either no prior knowledge is assumed leading to zero initial landmarks and the full scanning of the environment where the obtained landmark states will be updated as far as further observations reinforce or delete the initial knowledge. This study by assuming fully known boundary landmarks offers on one hand, an appealing opportunity for geometrical validation test in data association stage, and, on the other hands, allows more accurate association and filter update estimation as soon as one of these landmarks is part of the suboptimal map, which is part of the state vector. This is due to the fact that the variance-covariance matrices associated to these landmarks are close to null evaluation, which, in turn, affects, the estimation process of the filter. Kwork and Dissanayake (2004) used a multiple hypothesis filter to initialise landmarks based on a number of hypotheses.
- The issue of when the filter will be updated is also debatable. Indeed, while the computational complexity requirement tends to postpone the update as late as possible (Knight et al., 2001), the requirement of building a complete and a consistent map tends to prevent such postponement. However, this aspect is rather very context dependent.
- The extended Kalman filter has often been criticised in case of high nonlinearity of either the state or measurement equations, which led to the rapidly developing

Monte-Carlos based approaches (Montemerlo et al., 2003). However, the computational complexity get increased, while in case of relatively small robot speed, the performance of the extended Kalman filter still are acceptable.

3.7 Multiple Robot Localization

Important feature of our robotic system is the use of two different robots. Consequently, this offers the possibility of mutual collaboration in updating their current states. Intuitively, the basic scenario consists of making the vision robot turning around until the second robot is identified, and next the coordinates of the robots are matched, and updated accordingly. The identification of the robot by the vision turret is made possible through appropriate choice of black/white cover, which was selected different from objects in the environment. More formally, let $X_R^i = (x_i \ y_i \ \theta_i)^T$ be the state vector attached to the i^{th} robot. Similarly let P_R^i designate the associated variance-covariance matrix. So, for the two robots, the dynamic models can be rewritten in the light of (2) as:

$$X_R^1(k+1|k) = F_k^1(X_R^1(k|k)) + \eta_k^1 \quad (32)$$

$$X_R^2(k+1|k) = F_k^2(X_R^2(k|k)) + \eta_k^2 \quad (33)$$

With

$\eta_k^1 \mapsto N([0 \ 0 \ 0]^T, Q_1)$ and $\eta_k^2 \mapsto N([0 \ 0 \ 0]^T, Q_2)$. F_k^1 and F_k^2 correspond to odometric model of robot 1 and 2, respectively, similarly to (1).

Let $X = [X_R^1 \ X_R^2]^T$ be the state vector constituted of the two robot states. Then the associated variance-covariance matrix P can be rewritten as in (Martinelli et al., 2005):

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \text{ with } P_{ii} \text{ stands for } P_R^i \text{ -variance-covariance of the state of } i^{\text{th}} \text{ robot.}$$

The predicted variance-covariance and predicted state are evaluated as

$$P_{ii}(k+1|k) = \nabla F_k^i \cdot P_{ij}(k|k) \cdot \nabla F_k^{iT} + Q_i \quad (i=1,2) \quad (34)$$

$$P_{ij}(k+1|k) = \nabla F_k^i \cdot P_{ij}(k|k) \cdot \nabla F_k^{jT} \quad (i,j=1,2 \text{ and } i \neq j). \quad (35)$$

$$\hat{X}(k+1|k) = [F_k^1(X_R^1(k|k)) \ F_k^2(X_R^2(k|k))]^T \quad (36)$$

Within the framework constituted of the single system of the two robots, assume, for instance, at a given time, the Robot 1 observes the Robot 2, this relative observation can be modelled as

$$z = h(X) + w, \text{ with } w \mapsto N(0, R_w) \quad (37)$$

Where h corresponds to the model of the predicted observation, which, in case of relative bearing in terms of the two robot configurations, is given by:

$$h(X) = a \tan \left(\frac{-\sin \theta_1 \cdot (x_2 - x_1) + \cos \theta_1 \cdot (y_2 - y_1)}{\cos \theta_1 \cdot (x_2 - x_1) + \sin \theta_1 \cdot (y_2 - y_1)} \right) \quad (38)$$

Therefore, the update estimations given the (relative) observations are determined using the

standard (extended) Kalman filter equations by:

$$\hat{X}(k+1|k+1) = X(k+1|k) + P(k+1|k) \cdot (\nabla h \cdot P(k+1|k) \cdot \nabla h^T)^{-1} [z - h(X)] \quad (39)$$

$$P(k+1|k+1) = P(k+1|k) - P \cdot \nabla h^T (\nabla h \cdot P(k+1|k) \cdot \nabla h^T)^{-1} \nabla h \cdot P(k+1) \quad (40)$$

With,

$$\nabla h = [\nabla_{x_R^1} h \quad \nabla_{x_R^2} h] \quad (41)$$

$$\nabla_{x_R^1} h = \begin{bmatrix} \frac{y_2 - y_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & -\frac{x_2 - x_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & -1 \end{bmatrix} \quad (42)$$

$$\nabla_{x_R^2} h = \begin{bmatrix} -\frac{y_2 - y_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & \frac{x_2 - x_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & 0 \end{bmatrix} \quad (43)$$

The above estimation provides an estimation of the states of both robots 1 and 2 in terms of state vector estimation as well as the variance-covariance matrix associated to each one. Moreover, the estimation also provides an indication concerning the correlation among the estimations of both robots as quantified by the quantity P_{12} and P_{21} of the matrix P .

Notice that in case where the two robots are within range sensor reach, the observation also includes the relative distance, in this case, the observation model h in (37) boils down to:

$$h(X) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (44)$$

This leads to a Jacobian measurement matrix $\nabla h = [\nabla_{x_R^1} h \quad \nabla_{x_R^2} h]$ with,

$$\nabla_{x_R^1} h = \begin{bmatrix} -\frac{x_2 - x_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & -\frac{y_2 - y_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & 0 \end{bmatrix} \quad (45)$$

$$\nabla_{x_R^2} h = \begin{bmatrix} \frac{x_2 - x_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & \frac{y_2 - y_1}{(y_2 - y_1)^2 + (x_2 - x_1)^2} & 0 \end{bmatrix} \quad (46)$$

The (update) estimation process is still made of expressions (39-40).

Note that the update equations can be used either via one of the measurement equation (38) or (44) or both of them depending on availability of measurements as the filter can be used recursively with respect to measurements as well.

Remark that the state vector in this situation does not include landmark states due to the fact that the process of multi-localization is not used as often as mapping. It is only employed when the process of robot collaboration is called upon, or when the quality of the estimation as quantified by the matrix P is too poor. This is because the process of looking for Robot 2 by Robot 1 using the vision turret is relatively costly.

Alternatively, one may think of using both the mapping and multi-robot simultaneously. In this case, the rational is to leave the state of landmarks with only one robot state vector, e.g., $X = [X_R^1 \quad X_L \quad X_R^2]$. However, if both robots navigate within the environment for sufficient time interval, then a rational is to maintain two different maps; that is, each robot will maintain its own map of the environment, leading to an augmented state vector. The estimation process is somehow similar to that already developed in this section, where

observations might be either relative observation with respect to a second robot where one robot identified another robot, or relative observation with respect to landmark (s).

3.8 Overview of general approach

Figure 5 summarizes the general scheme of our SLAM approach involving multi-robot collaboration and localization. The vector U mentioned in Figure 5 corresponds to the command vector in terms of incremental moving of the right and the left wheel of the robot used by the encoder.

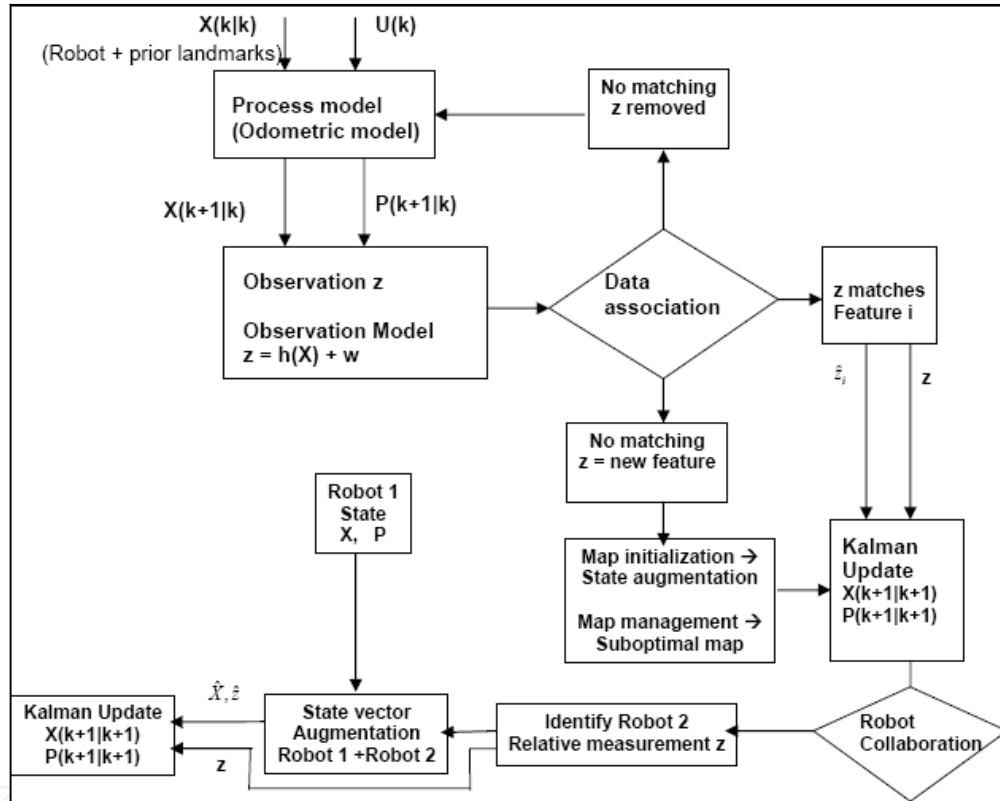


Fig. 5. General scheme of the SLAM-collaboration process.

Note that the possible movement of Robot 1 is not represented, but can obviously carry out the same reasoning as Robot 2 in terms of stages: i) prediction using encoder model; ii) measurements & measurement model; iii) data association based on Mahalanobis distance that determines the validity of the measurement and whether it corresponds to an already identified landmark or to a new landmark, which, in turn, induces either state augmentation, or just a false alarm that needs to be discarded; iv) mapping and suboptimal map construction based on the viewing field of the sensor and the timing frame; v) update with respect to (extended) Kalman filter equations.

4. Testing and results

Figure 6 shows an example of mapping the entire perimeter of the environment using one single infrared sensor. On the left hand side of Figure 6 is shown the virtual representation of the perceived environment in terms of a set of landmarks. Remark that the number of landmarks depends on the robot speed and sampling frequency of infrared sensors. Also by increasing the resolution to 'High' the walls have a more solid and continuous look.

The prior landmarks consisting of the four corners of the environment are used here for geometrical validation purpose of all represented landmarks but they are not plotted in figure 6 because they are not perceived by the robot sensors.

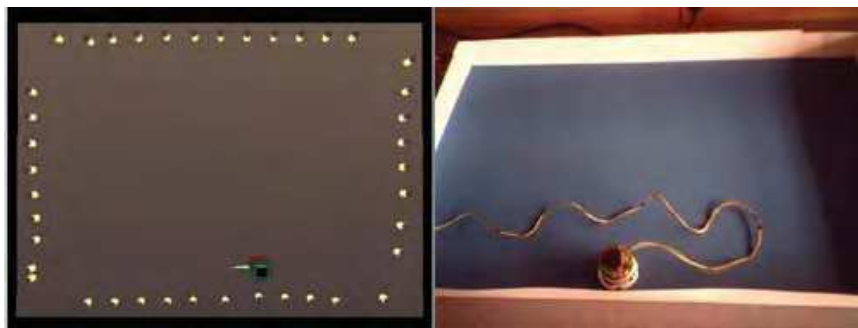


Fig. 6. Mapping the perimeter of the environment by robot.

Figure 7 shows another example of robot configuration and map building using SLAM algorithm while using curved object as a set of bounded plans in the environment. On the left hand side of Figure 7, the objects are modelled as a successive set of landmarks (virtual Cartesian points). The feature landmarks can therefore be extracted from these Cartesian points. Intuitively, one can identify at least eight segment lines together with eight corners.

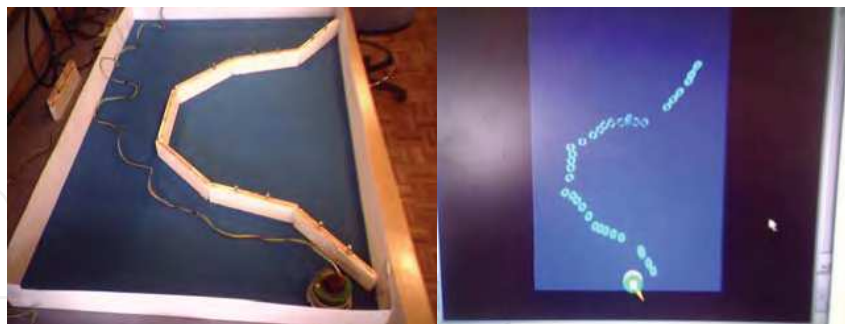


Fig. 7. Example of environment mapping.

Ultimately linked to the preceding is the issue of loading and saving environment map. This is carried out by the user's interface. Indeed, to ensure the maintaining of previous maps even if the system switched off, the map is saved in file format and loaded upon request. To recreate the map from the interface, one requires: i) select "Load Map" previously saved

map in the interface; ii) click the console window to make it the active window; iii) type in the name of the map to load, in this case “completeperimeter” and press enter. The console will then display object and robot information that is being loaded into the map, the result of which will be displayed in the map window. An example is shown in Figure 8 that retrieves the perimeter map, which is kept along all experiments carried out in this study. Now in order to quantify the quality of the estimation, Figure 9 displays the error in vehicle localization. The errors are quantified for the environment shown in Figure 7. In the latter the true positioning are measured with respect to Cartesian coordinate chosen at the left hand corner of the environment.



Fig. 8. Console output showing map objects being successfully loaded.

Figure 9 shows the actual error in estimated vehicle location in both x and y coordinates (solid line) as a function of time increments, which summarizes the vehicle movement as in Figure 7. The Figure also displays the 95% confidence limits, or two-times standard deviation around estimates (represented in dashed line), which is driven from the state covariance P by selecting the relevant component of P pertaining to $P_{R_{xx}}$ and $P_{R_{yy}}$ and taking the square root. As it can be noticed from the plot, the vehicle error is clearly bounded by the confidence limits of estimated vehicle error, which shows the consistency and convergence of the underlying SLAM algorithm. This also demonstrates that the algorithm clearly yields consistent and bounded errors.

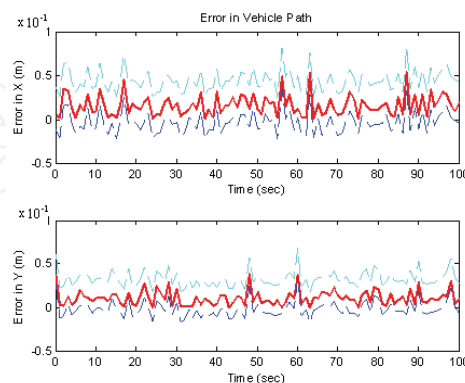


Fig. 9. Error in vehicle location estimate in x and y coordinates and the associated 95%

confidence bounds (dotted line) obtained from state estimate covariance matrix.

Figure 10 shows the evolution of the innovation of range measurement over the time. The bounded limit curves corresponding to 95% confidence limit drawn in the same plot indicate the consistency of the filter outcomes and estimation process. Note that only the range innovation is displayed here because only range measurements were used in the map initialization and management. The bearing measurements are only used for robot collaboration.

Figure 11 provides an estimation of the quality of landmark estimations in terms of standard deviation with respect to x and y coordinates as quantified by landmark variance-covariance matrices P_{L_i} . We restricted to three chosen landmarks consisting of the first landmark encountered and two others.

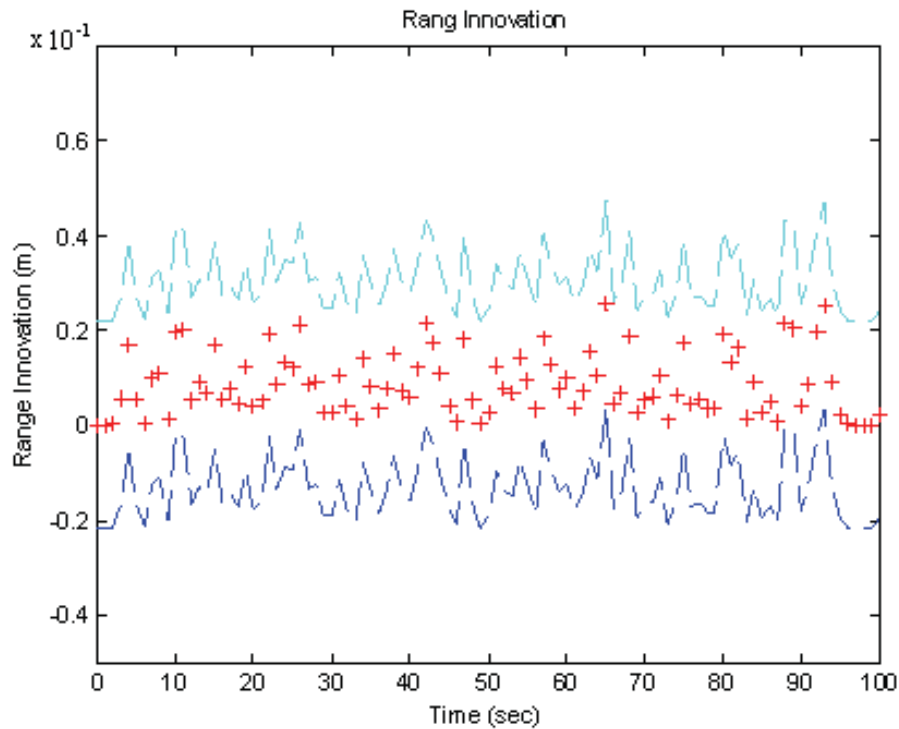


Fig. 10. Range innovation with associated 95 confidence bounds.

This explains why the estimation process does not start at time $t=0$ for all landmarks as landmark 2 and 3 are not identified yet at the beginning. The plots were obtained after using least square fitting in standard deviation results in order to obtain a smooth representation. The graph shows clearly a decreasing tendency in the standard deviation over time, which indicates consistent and convergence of the estimation process. On the other hand, as far as the local map is concerned, the display shows that despite the use of local map while maintaining all information with computational effective algorithm whose complexity is proportional to the number of landmarks, the performances of the filter in terms of variance-covariance matrix still behave consistently. This demonstrates that the developed cost effective SLAM algorithm does not suffer from divergence. Notice that in Figure 11, the initial

uncertainty ascribed to landmarks 2 and 3 is always less than that of landmark 1. This is because while the vehicle is moving around, its associated variance-covariance matrix tends to decrease, which, in turn, reduces the uncertainty of the identified landmarks.

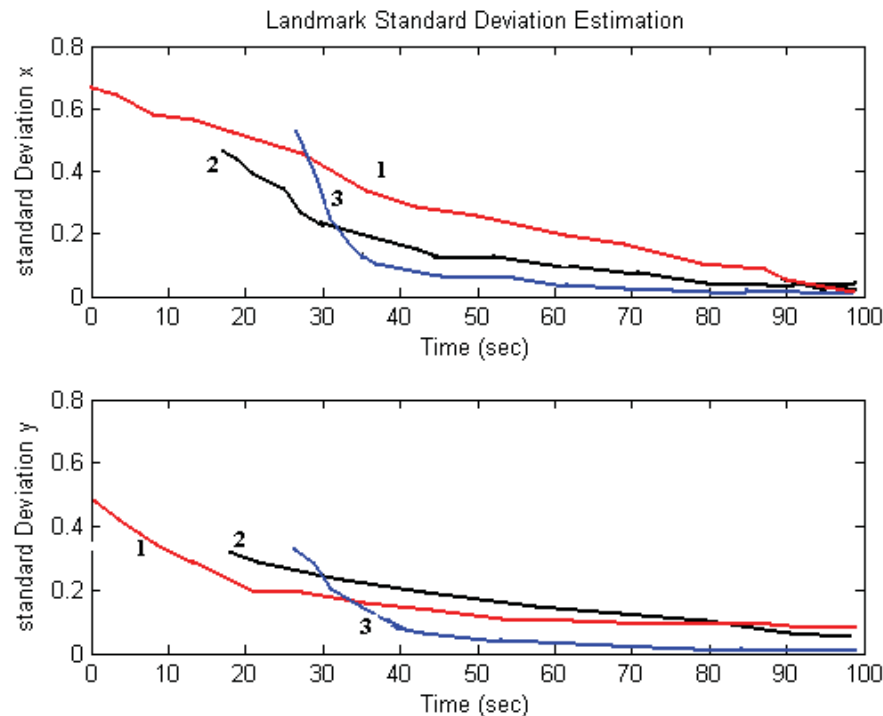


Fig. 11. Landmark Estimation In terms of Standard Deviations in x and y coordinates.

Now considering the situation in which two robots were used to accomplish a task, in order to quantify the performance of the algorithm, one considers the scenario shown in Figure 1 of this chapter. In the latter Robot 1 (vision robot) identifies an object (white box) in the environment and sends a command to Robot 2 to go to its location. The figure displays the performance of the estimation of the Robot 2 localization in terms of x and y standard deviation together with the associated 95% confidence bounds. Clearly, the decreasing behaviour together with the conservative bounds testify on the consistent and convergent estimation process along the collaboration scheme. Notice that the figure shows a local region where standard deviation does increase. This, in fact, corresponds to the region of the environment where the robot goes straight to the object and no observations were taken, so the filter only relies on prediction part of the filter, which trivially tends to increase the variance-covariance estimate. Intuitively by exchanging relative pose information, the states of the robots are updated in a centralized fashion. Note that Robot 1 once the image of the object after a single revolution, the robot becomes static. Consequently, the possibility of using Robot 2 to update the state of Robot 1 is very limited in this scenario.

Figure 13 shows the trajectory of Robot 2 while moving toward the target object and exploring the underlying object. In the same plot is displayed the true position of the vehicle in terms of x-y coordinates of the reference point in the robot platform. The estimated trajectory is represented in circles (o) while the true trajectory is drawn in start (*).

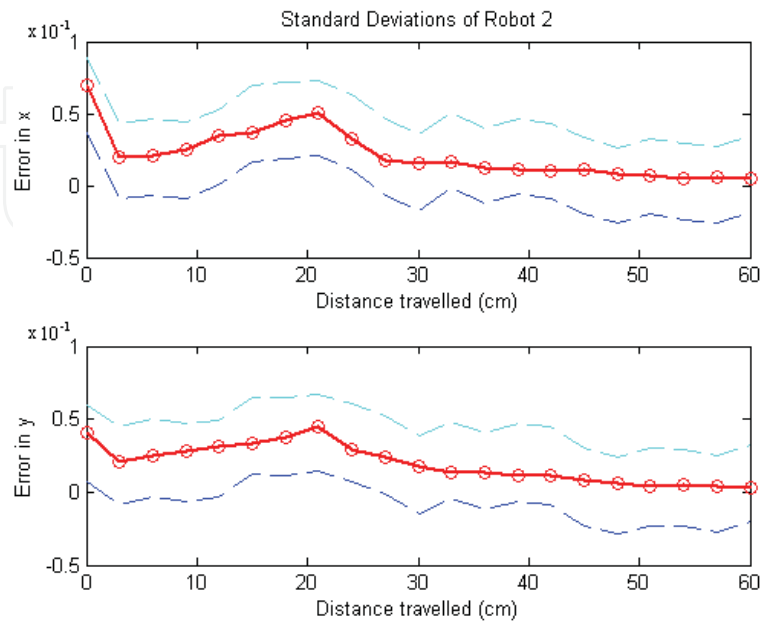


Fig. 12. Performances of Robot 2 in terms of standard deviation in x and y in case of robot-collaboration scenario.

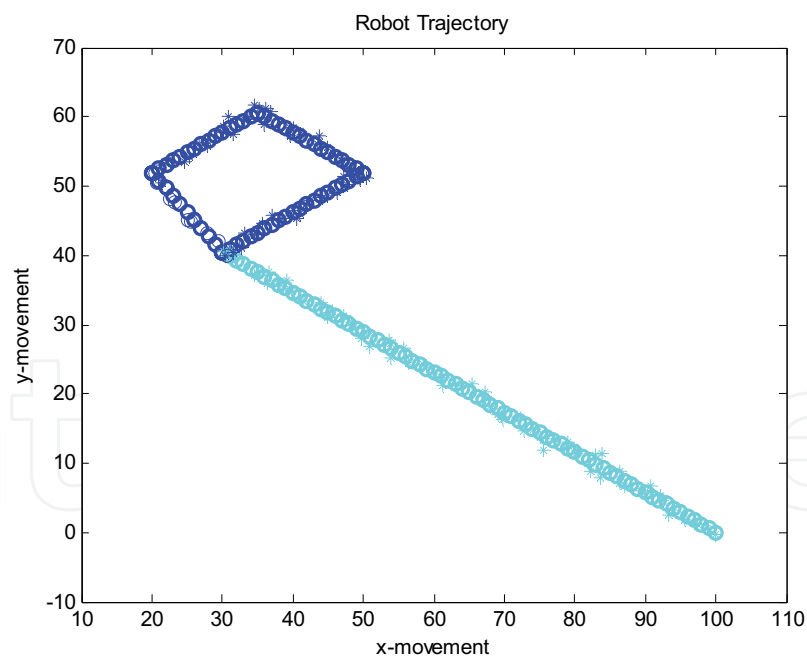


Fig. 13. Estimated Robot trajectory versus True trajectory.

5. Conclusion

This chapter investigated the state of the art of SLAM system whose aim is to simultaneously map the environment and provide accurate positioning of the vehicle where the state of both vehicle and landmarks are brought into a single estimation process. The estimation process supplies estimation of vehicle and landmarks in terms of mean and variance-covariance estimate of the state vector conditional on the whole set of measurements. The construction of the map involves appropriate initialization step in which some prior knowledge regarding the environment is reported, which allows us to ensure geometrical validation test later on, and an alignment stage in which the observations are turned into landmark Cartesian coordinates. Next, a data association stage is required to map the observations to the already identified landmark or initiate new landmarks. For this purpose, one uses Mahalanobis distance to match the observation to possible landmarks. If none of the landmarks matches the current observation and both the geometrical and statistical tests were positive, then a new landmark is initiated and the state vector is therefore augmented. Note that, in order to balance the cost effectiveness and optimality requirement, the reasoning is carried out only within a submap of the environment, where it is most likely to find the matching given the sensor limitations. At later stage, the obtained landmarks are also used to extract feature landmark consisting of segments and corners. The methodology has been validated in a platform using two Khepera robots, one of which is equipped with vision turret while both are equipped with range infrared sensors and encoders. A virtual interface showing the robot trajectory as well as environment is developed using OpenGL platform for 3D visualization. The use of both robots also allowed us to test and validate collaboration scenarios in which multi-robot localization technique is used in conjunction with SLAM algorithm. The tests carried out demonstrated the validation of the developed algorithm and the consistency of the outcomes when looking at the 95% confidence bound limits.

This opens a new area of research where more advanced collaboration scenarios can be used in more complex environments where the features can be constituted of either geometrical or non-geometrical features. On the other hand, inspired by the overall intelligent behaviour of large biological insect communities, together with the rapid development of the field of distributed artificial intelligence, through, for instance, the concrete RoboCup robot soccer initiative, this offers new motivation grounds for further developments of multiple robot systems at different research communities.

6. Acknowledgment.

This work is sponsored in part by a grant from Nuffield Foundation.

7. References

- J. Andrade-Cetto and A. Sanfeliu (2004). The Effect of partial observability in SLAM, in *Proceedings of IEEE International Conference on Robotics and Automation*, p. 397-402
- T. Bailey and H. Durrant-White (2006). Simultaneous localization and Mapping (SLAM): Part II. State of the Art, *IEEE Robotics and Automation Magazine* 13(3), 108-117
- M. Begum, G.K.I. Mann and R.G. Gosine (2005), Concurrent mapping and localization for mobile robot using soft computing techniques, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, p. 2266-2271
- V. Bonato, M.M. Fernández and E. Marques (2006). A smart camera with gesture recognition and SLAM capabilities for mobile robots, *International Journal of Electronics*, 93(6), 385-401

- K. Chong and L. Kleeman (1999). Mobile robot map building from an advanced sonar array and accurate odometry. *International Journal of Robotics Research*, 18(1):20–36.
- M. Deans and M. Hebert (2000). Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In: *Proceedings of the International Symposium on Experimental Robotics*.
- M.W.M.G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba (2000). An experimental and theoretical investigation into simultaneous localisation and map building. *Experimental Robotics IV*, p. 265–274.
- M.W.M.G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csorba (2001). A solution to the simultaneous and mapping (SLAM) problem, *IEEE Transactions On Robotics and Automations*, 17(3), 229–241
- M.W.M.G. Dissanayake, H.F. Durrant-Whyte, and T. Bailey (2000). A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, p. 1009–1014.
- D. Dufourd and R. Chatila (2004), Combinatorial maps for Simultaneous Localization and Map building (SLAM), In: *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p.1047-1052
- H.F. Durrant-White (1988), Uncertain geometry in robotics, *IEEE Transactions on Robotics and Automation*, 4(1), 23–31.
- H. Durrant White and T. Bailey, Simultaneous localization and Mapping: Part I (2006), *IEEE Robotics and Automation Magazine*, 13(2), 99–108
- A.I. Eliazar and R. Parr, DP-SLAM 2.0 (2004), in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, p.1324-1320
- A. Gelb (1996). *Applied Optimal Estimation*. MIT Press, 14th edition.
- J. Guivant, E.M. Nebo (2001) Optimization of the simultaneous localization and map building algorithm for real time implementation, *IEEE Transaction on Robotics and Automation*, 17(3), 242–257.
- T. Higuchi (1996). Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, 59(1):1–23.
- Y.L. IP and A.B. Rad (2004), Incorporation of Feature Tracking into Simultaneous Localization and Map Building via Sonar Data, *Journal of Intelligent and Robotic Systems*, 39(2), 149–172
- S. J. Julier (2003), The stability of covariance inflation methods for SLAM, in: *Proceedings of the IEEE/RSJ International Conference on Robots and Systems*, 2749–2754
- S. J. Julier and J. K. Uhlmann (2001), Simultaneous Localization and Map Building Using Split Covariance Intersection, In: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, p.1257-1262
- H. Kitano (1997), *RoboCup-97: Robot Soccer World Cup I*, *Lecturer Notes in Computer Sciences / Lecturer Notes in Artificial Intelligence* Vol. 1375, Springer-Verlag, Berlin, NY.
- J. Knight, A. Davison and I. Reid (2001), Towards Constant Time SLAM using Postponement, in: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, 406–412
- N. M. Kwok and G. Dissanayake (2004). An Efficient Multiple Hypothesis Filter for Bearing-Only SLAM, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, p.736–741
- T. Lemaire, S. Lacroix and J. Sola (2005), A practical 3D Bearing-Only SLAM algorithm, in: *Proceedings of Intelligent Robotics and Systems, IEEE/RSJ International Conference*, p.2449–2454.

- J. J. Leonard and H.F. Durrant-Whyte (1991). Simultaneous map building and localization for an autonomous mobile robot. In: *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Vol. 3, p. 1442-1447.
- J. J. Leonard, R.J. Rikoski, P.M. Newman, and M.C. Bosse (2002). Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research*, 21(10-11):943-975.
- B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset (2003). Hierarchical simultaneous localization and mapping. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, p 448-453.
- A. Martinelli, N. Tomatis and R. Siegward (2003), Open Challenges in SLAM: an open solution based on shift and rotation invariants, in: *Proceedings of IEEE International Conference on Robotics and Automation*, p.1327-1332
- A. Martinelli, F. Pont and R. Siegward (2005). Multi-robot localization using relative observations, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, p. 2808-2813
- M. Matarik (1998), Coordination and learning in multi-robot systems, *IEEE Intelligent Systems*, 13(2), 6-8
- M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0 (2003): An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *Proceedings of the IJCAI-03*, Morgan Kaufmann,. 1151-1156.
- M. Montemerlo and S. Thrun (2003). Simultaneous localization and mapping with unknown data association using FastSLAM. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, p. 1985-1991.
- R. Murphy (2000), *An introduction to AI Robotics*, MIT Press
- J. Nieto, J. Guivant, and E. Nebot (2004). The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, p. 391-396.
- J. Nieto, T. Bailey, and E. Nebot. Scan-SLAM (2005). Combining EKF SLAM and scan correlation. In: *Proceedings of the International Conference on Field and Service Robotics*.
- M. Oussalah, H. Maaref and C. Barret (2003), Application of a possibilistic-based approach to mobile robotics, *Journal of Intelligent and Robotics Systems*, 38(2), 175-195.
- R. Smith and P. Cheeseman (1986). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56-68.
- R. Smith, M. Self, and P. Cheeseman (1987). Estimating uncertain spatial relationship in robotics, in: *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfon (Editor), *A stochastic map for uncertain spatial relationships. Autonomous Mobile Robots: Perception, Mapping and Navigation*, 323-330.
- J. Sola, A. Monin, M. Devy and T. Lemaire (2005), Undelayed Initialization in Bearing Only SLAM, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2499-2504
- J. Todd and D. Pomerleau (1996). Life in the fast lane: The evolution of an adaptive vehicle control system. *Artificial Intelligence Magazine* 17 (2) 11-50.
- S. Thrun, D. Fox and W. Burgard (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31:29-53. Also appeared in: *Autonomous Robots* 5:253-271
- W. S. Wijesoma , L. D. L. Perera and M.D. Adams (2006). Towards Multidimensional Assignment Data Association in Robot Localization and Mapping, *IEEE*

Transactions on Robotics, 22(2), 350-365.

S.B. Williams (2001), *Efficient Solutions to Autonomous Mapping and Navigation Problems*, PhD thesis, University of Sydney.

IntechOpen

IntechOpen



Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

ISBN 3-86611-283-1

Hard cover, 704 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, February, 2007

Published in print edition February, 2007

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well as those handling global navigation aspects of a single robot and even for a group of robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

M. Oussalah and D. Wilson (2007). Robot Collaboration for Simultaneous Map Building and Localization, Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), ISBN: 3-86611-283-1, InTech, Available from: http://www.intechopen.com/books/mobile_robots_perception_navigation/robot_collaboration_for_simultaneous_map_building_and_localization

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2007 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen