Feature Extraction for Autonomous Navigation

Elizeth G. Araujo and Roderic A. Grupen

Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts, Amherst MA 01003

Technical Report 97-55

This technical report presents techniques for environmental feature detection and identification using sonar sensors. By detecting common features in indoor environments and using them as landmarks, a robot can navigate reliably, recovering its pose when necessary. Results using a multiple hypothesis testing procedure for feature localization and identification show that accurate feature information can be acquired with adequate sonar models and configurations. In addition, a method that associates sonar configuration with the precision of feature extraction is discussed, as well as its utility for guiding an active sonar sensor.

Future goals are to make the pose recovery procedure dependent upon navigation constraints, and to study the use of the navigational knowledge acquired to optimize the path generated by an incremental motion planner.

ACKNOWLEDGEMENTS

The research reported here was conducted in the Laboratory for Perceptual Robotics at UMASS and supported in part by NSF IRI-9503687, IRI-9704530, and CDA-9703217.

TABLE OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Sound-based navigation has been shown to be effective, not only in man-made systems, but primarily in nature. Bats master echolocation [22], suggesting that sonars can extract high level information from the environment. This work focuses on the extraction of specific information from the environment to reduce pose uncertainty in robot navigation tasks. Our objective is to identify feature sets which actively resolve localization errors.

This chapter gives an introduction to sonar sensors, sonar-based modeling, and sonar configuration, followed by the description of the sensor system selected. Chapter 2 together with Appendices A and B present the feature localization models, the method used to estimate feature errors, and the multiple hypothesis testing methods employed in feature localization and identification. A number of experiments are discussed in Chapter 3, and Chapter 4 presents conclusions and future work. Finally, Appendix C describes the 2D sonar simulator developed to test different sonar configurations, and Appendix D presents our mobile robot and active sonar system.

1.1 Sonar Sensor

Sensors are usually described by their characteristics, such as range, accuracy, precision, and response time. These characteristics are measured under specific conditions in controlled environments that are, for some sensors, too restrictive and normally violated in practice. The quality of the information extracted by a sensor depends primarily on the accuracy of the model used to translate raw sensor readings into measurements (perceptions). Accurate models are difficult to derive because the physics of the transducers is usually complex and context dependent. A new and alternative approach is to employ a model that accounts for a sufficient part of the data, ignoring the contexts it cannot model. The selection of which data to ignore can be achieved by exploring the correlation between multiple readings from the same sensor, or from multiple sensors and sensor modalities. Examples of such methods are presented in the following chapters.

In this work, sonar represents airborne ultrasonic range sensing based uniquely in time-of-flight (TOF). The main advantages of using sonars as range sensors in a mobile robot are their low price, range of actuation, simple interface, and typically highly accurate readings over the entire range. However, sonars have a slow response time (limited by the velocity of sound in air); multiple sonars firing at the same time may generate cross-talk; their large beam angle makes the use of simple ray-trace models impossible; and typically only some of the readings (in some cases less then 50%) are consistent with the model used. These limitations make the use of sonar a challenge to sensor modeling, data fusion, and sensor management, creating a fertile testbed for testing new methods that address these issues.

1.2 Sonar-Based Modeling

Two sonar-based modeling approaches have been described in the literature: grid-based probabilistic models that avoid direct modeling of the environment [7, 4, 9], and feature-based models that exploit the interaction between sonar beam and frequently encountered environmental features [6, 13, 14, 12]. Complementing these methods, sensor fusion approaches and data pre-filtering algorithms are widely used, not only to reduce uncertainty, but also to identify contexts consistent with the model employed.

1.2.1 Chronology

One of the first sonar response models presented was a feature-based model [6], where surface information was first extracted from the raw sonar data, creating a *logical sensor*, and then applied to map building. Limitations of this approach led to the use of grid-based probabilistic model approaches, such as occupancy grid and vector field. The argument used in favor of a probabilistic approach to sonar modeling is that raw sonar data is subject to several, difficult to model, environmentally dependent effects such as specular reflections and sensor cross-talk, and therefore geometrical reasoning purely on the basis of raw data is not appropriate.

Subsequent feature-based models disagreed on how the ultrasonic signal interacts with the objects in the environment in general. Previous models assume a specular reflection when the difference between wavefront incident angle and the normal to a smooth surface is too large, causing no return signal. In this case, objects are assumed to be detected mainly by the effect of diffusion [6, 7, 4]. Recent feature-based approaches argue that indoor environments consist mainly of specular surfaces or *"mirror-like"* reflectors. This assumption is based on the the significantly different acoustic impedances of air and solids, and the wavelength of ultrasonic frequencies compared to object surface roughness [14, 15, 12]. Specular world assumptions have proven to be more general and at the same time enabled a more detailed and fruitful geometric analysis of the interaction between sonar beam and general office environment features. This fact gave a new spin to the use of feature-based sonar models, showing that even very simple sonar devices could produce better quality information with the use of an adequate model and sensor configuration [14, 20, 15, 11].

1.2.2 Grid-based Probabilistic Models

Grid-based methods discretize the environment and update the occupancy hypothesis at correspondent grid cells with each sonar reading based on sonar model and the data fusion technique employed. Examples of this approach include: occupancy grid [7, 8, 19], inference grids [9], and vector field histogram [4].

The vector field technique models the sonar as a ray-trace sensor, and employ a histogram to accumulate occupancy statistics. It has been tested in specific obstacle avoidance tasks with relative success, but robustness with respect to different environments and tasks is low, partially due to the limitations of the sonar model used.

Occupancy grid techniques associate a probability of occupancy to each of the cells in the sector swept out by the sonar beam. This information is fused into a map by using Bayesian updates [7, 8] or Dempster-Shafer influence rules [19]. Inference grid technique is a generalization of the occupancy grid approach that estimate other properties in addition to occupancy, such as reachability, color, reflectance, etc. Both approaches have been tested on robot perception and navigation tasks with relative success, highlighting the superiority of such representations in multi-modal sensor fusion tasks [16].

1.2.3 Feature-based Models

The feature-based models transform raw sonar data, sometimes only time-of-flight (TOF), into information about the environmental geometry. The success of these methods is highly dependent on how the interaction between sonar beam and the environmental features were modeled, and how common these features are in the environment. Examples of feature-based approaches are the composite local model applied to a navigation task [6], the corner-edge-wall-transducer model (CEWT) used on feature detection and identification tasks [13], regions of constant depth (RCDs) used on map building and pose localization tasks [14], and the tri-aural approach applied to the localization and identification of corners, edges, and planes in the environment [20].

In the composite local model approach [6], line segments are extracted from data acquired from a rotating sonar device. This work assumes diffuse reflections in which sonar returns are considered to be the minimum beam distance to the reflecting surface.

Most feature-based methods argue that office environments are in general composed by specular surfaces to the sonar acoustic signal (*"mirror-like surfaces"*) [5]. By modeling the environment as mainly specular, several known techniques employing detailed models of the interaction between the sonar beam and the common feature in the environment can be exploited.

The CEWT model presents closed-form solutions for the reflections from corner, edges, and wall features by considering only specular surfaces and by employing two collaborating sonars. These solutions have been tested in simulation and validated by comparing them with actual sonar maps of simple real world environments, obtained with off-the-shelf ultrasonic transducers.

The method described above inspired the subsequent feature-based models. In the RCD model, multiple adjacent returns with similar range were grouped forming RCDs, reducing the uncertainty produced by the sonar large beam angle since adjacent returns from a target restricts the target's bearing. The RCDs were subsequently matched to target models of common environmental features: corner, edge, plane, and cylinder, to address the problem of pose localization given a map, and map building given the robot's pose.

The tri-aural approach uses an array of three ultrasonic transducers aligned and evenly spaced with the middle transducer acting both as transmitter and receiver and the others acting only as receivers. The use of a transmitter and multiple receivers configuration facilitates feature localization and identification by increasing the quality of information obtained per sonar firing and over different poses.

1.3 Sonar Configurations

Methods used to extract information from sonar are highly dependent on sonar configuration. Sonar configuration encompasses not only the spatial configuration of transducers over the robot, but also how their functionality is distributed – e.g. the geometric relationship between transmitter, world, and receiver is important, and the kind of signal processing utilized is also important (analysis in: time, amplitude, frequency, and phase). Robots are usually equipped with fixed transducers evenly distributed around their periphery on a plane parallel to the floor; on round robots, rings with 8 to 24 transducers are common. This configuration facilitates obstacle avoidance, since multiple sonars can be fired at the same time, quickly observing the robot's surroundings. Since the overlap between sonar beams is minimal in sonar ring configurations (sparsely sampled data), tasks such as tracking and robot localization are jeopardized. These tasks need densely sampled data obtained, for example, with a rotating transducer, or array of of transducers [14, 15].

Only recently have different sonar configurations been explored to better fulfill task requirements, with the introduction of sonar arrays and virtual sonar sensors [2, 15, 11, 12]. A sonar array is a group of transducers that collaborate on a measurement, organized in a certain configuration. A virtual sensor is an abstraction where a set of observations, over space and time, is transformed into a measurement.

1.4 Our Sonar System

In this work, sonar-based modeling aims at the extraction of specific information from the environment with the objective of reducing the uncertainty associated with odometry errors, and subsequently with navigation. A feature-based model on a sonar array configuration was selected for this task. The idea is to detect a set of common features in the environment, and to use these features as landmarks (or soft beacons), allowing the robot to navigate without loosing its pose, and thus, to keep collecting consistent spatial information for navigation purpose. The generality and applicability of this method depends not only on the accuracy of the sonar model, but also on how frequently, how constant, and how easily detectable the features selected are. In this respect, the environmental features selected are those produced by walls – plane (line), convex edge (edge), concave edge (corner). These environmental features have the characteristics required above, and closed-form solutions exist for properly configured sonar arrays [20, 15, 11].

Feature detection is based on geometrical relations between two sonars; one of which transmits and receives (T), and the other only receives the return signal (R). It is assumed that the world geometry is approximately plane-, edge-, or corner-like and that it produces a reproducible pattern of responses on these feature detectors. Figures 1.1, 1.2, and 1.3 depict the geometric analysis involved on the translation from sonar ranging (r_1 from the transducer T and r_2 from the transducer R) to feature coordinates: edges and corners are recovered as (x,y) (Cartesian coordinate system), and lines are expressed using (r,θ) (Polar coordinate system).

The sonar model procedure, as shown on Figure 1.4, receives as input a reading pair ($r_1 \pm \Delta r_1$,$r_2 \pm \Delta r_2$) and outputs, when appropriate, the position evidence ($x \pm$

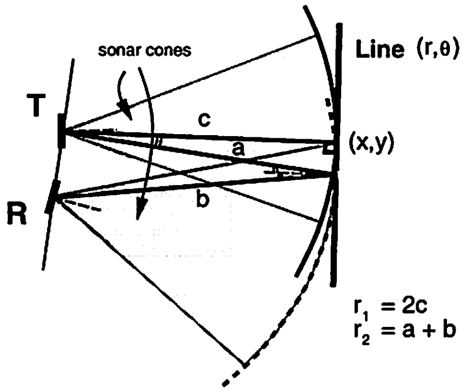


Figure 1.1. Line feature model

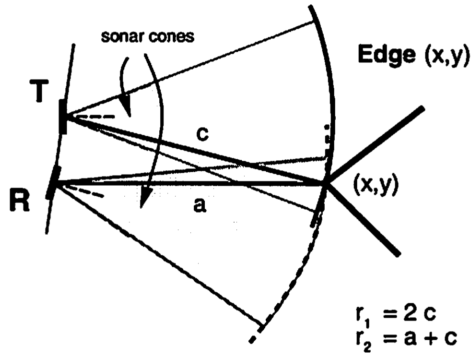


Figure 1.2. Edge feature model

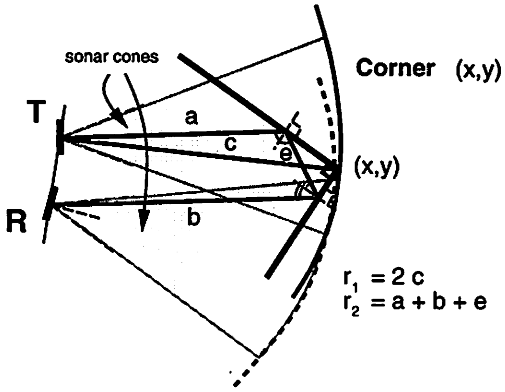
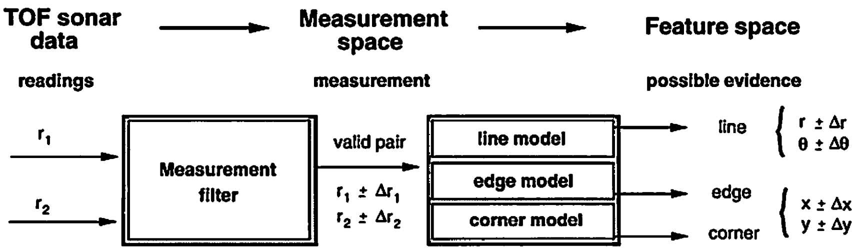


Figure 1.3. Corner feature model



Sonar-based Modeling Procedure:

1. *Discard the reading if the reflections (r_1,r_2) have no evidence of coming from the same feature;*
2. *Translate the sonar measurement (r_1,r_2) into 3 feature position evidence in feature space, (x,y) for edge and corner features, and (r,θ) for line feature;*
3. *Compute the error associated to each feature position evidence by translating the uncertainty from measurement space (Δr_1,Δr_2) to feature space – (Δx,Δy) and (Δr,$\Delta\theta$).*

Figure 1.4. Sonar model, from raw sonar data to feature evidence.

Δx,$y \pm \Delta y$) and ($r \pm \Delta r$,$\theta \pm \Delta\theta$) for the three features. Details about this procedure are presented on Chapter 2 and Appendix A.

Several spatial configurations were considered and are shown on Figure 1.5: (1) a sonar ring with 24 transducers that can execute 3 simultaneous transmissions from equidistant transducers, and all transducers can listen for return signals. (2) a two sonar array with one transducer transmitting and both receiving (Figure 1.5, middle and right frames). The impact of active sonar configuration in feature detection and identification tasks are analyzed in Chapter 2.

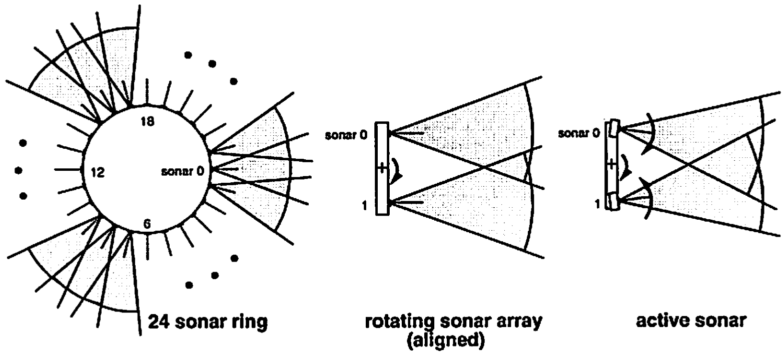


Figure 1.5. Sonar spatial configurations.

CHAPTER 2

FEATURE DETECTION AND IDENTIFICATION

The goal of extracting features from the environment is to use them as landmarks, or beacons, in navigation tasks. Therefore, features should be selected that are easy to detect and are abundant enough in the environment to permit their utilization as navigational feedback. This work uses lines, corners, and edges extracted from common indoor room features such as vertical planes (walls), corners and convex corners (edges), respectively. These features are not only common in an indoor environment, but they also allow closed-form estimates from time-of-flight (TOF) sonar information.

To extract features using sonar, models of the interaction between the ultrasound signal and the environment are required. The physics of these interactions are extremely complex and dependent on the environment, making the derivation of a precise model impossible. The alternative is to create simpler models by relying on some assumptions and introduce restrictions on the applicability of the model. In particular, we assume that the environment is specular to ultrasound. This assumption is imperative to the feature model derivation, and it was shown elsewhere to be not restrictive in practice [13].

2.1 Feature Localization Models

All the feature models described here use only range information from a pair of sonars. The range is easily obtained from *TOF*, since the velocity of sound in air is approximately constant in indoor environments ($\approx 343m/s$). One of the sonar pair operates as a transmitter and receiver (T), returning the range r_1, and the other operates only as a receiver (R), producing r_2. The range pair (r_1, r_2) is then used to compute a position estimate of each possible feature.

2.1.1 Line Feature

The line feature hypothesis uses the pair of ranges (r_1, r_2) and the position and orientation estimate of the transducer T (x_T,y_T,ϕ_T) to create a line position estimate (r, θ). This transformation is based on two assumptions: the line belongs to a specular surface, thus the angle of incidence is equal to the angle of reflection; and both signals received by the transducers T and R come from the same planar surface.

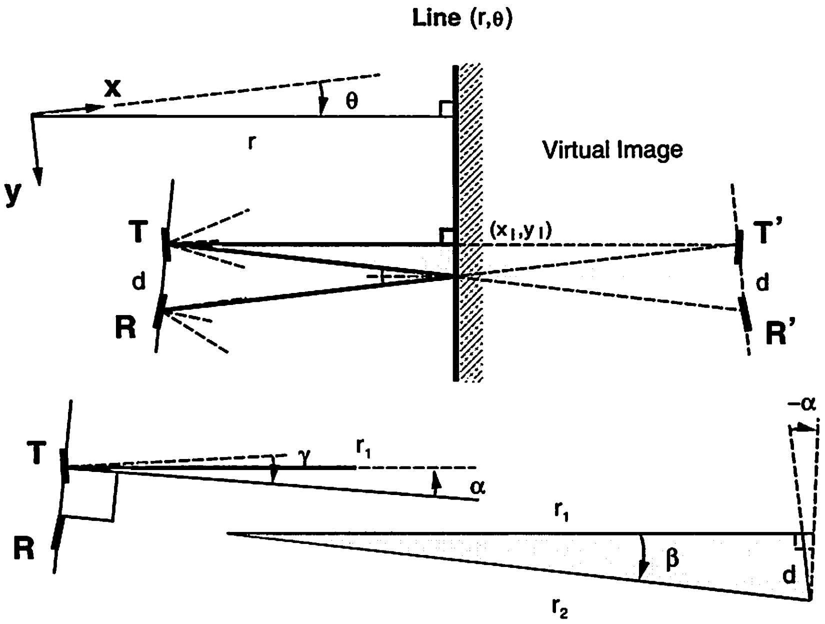


Figure 2.1. Line reflection

Figure 2.1 depicts the reflections generated by the ultrasonic signal on a planar reflector where T' and R' are virtual images of T and R, respectively. Under these circumstances, (r_1, r_2) must satisfy the following relations:

$$\begin{aligned} r_2 &= \sqrt{(r_1 - d\sin(\alpha))^2 + (d\cos(\alpha))^2} \\ &= \sqrt{{r_1}^2 - 2r_1 d\sin(\alpha) + d^2} \end{aligned}$$

$$\alpha = \arcsin\left(\frac{d^2 + {r_1}^2 - {r_2}^2}{2dr_1}\right) \tag{2.1}$$

$$\beta = \arctan\left(\frac{d\cos(\alpha)}{r_1 - d\sin(\alpha)}\right) \tag{2.2}$$

where d is the distance between transducers, α is the angle between the line that connects the transducers and the line feature, β is the angle between sonar bearings corresponding to r_1 and r_2.

Given the angle α from Equation 2.1, the position and orientation of the transducer T (x_T,y_T,ϕ_T), and the angle (γ) between the transducer T orientation and the normal to the line that connects the transducers, it is possible to compute the line parameters (r, θ), referring again to Figure 2.1:

$$r = x_l\cos(\theta) + y_l\sin(\theta) \tag{2.3}$$

$$\theta = \phi_T + \xi \tag{2.4}$$

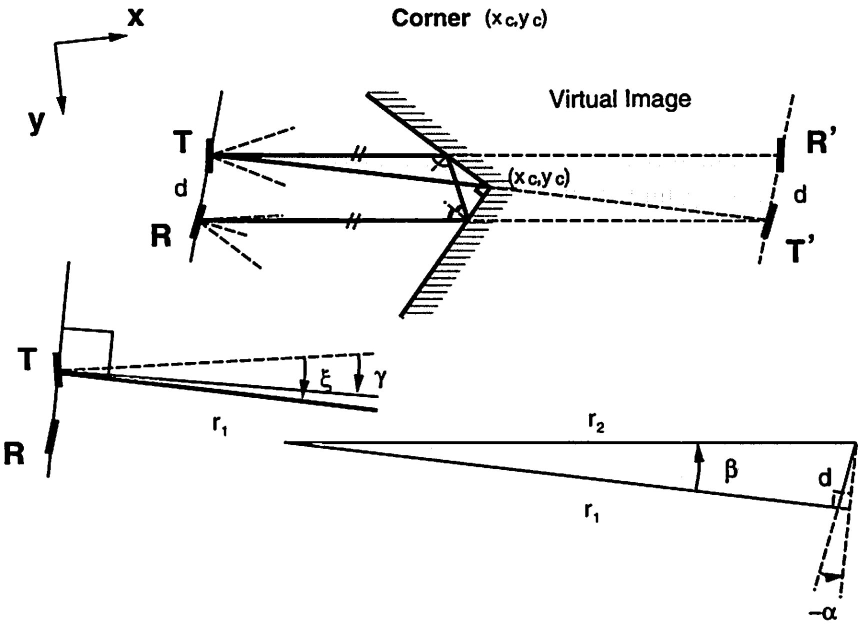


Figure 2.2. Corner reflection

where:

$$x_l = x_T + \frac{r_1}{2}\cos(\phi_T + \xi) \tag{2.5}$$

$$y_l = y_T + \frac{r_1}{2}\sin(\phi_T + \xi) \tag{2.6}$$

$$\xi = \gamma + \alpha \tag{2.7}$$

2.1.2 Corner Feature

The corner feature uses the range pair and the current position and orientation estimate of the transducer T to create the corner position estimate (x_c, y_c). A corner is composed of two intersecting, orthogonal planar surfaces. It is assumed that all the reflections come from the same corner feature.

The reflections of the ultrasonic signal on two planar reflectors forming a right angle corner are shown on Figure 2.2. The relations between sonars ranges are:

$$\begin{aligned} r_2 &= \sqrt{(r_1 - d\sin(\alpha))^2 + (d\cos(\alpha))^2} \\ &= \sqrt{{r_1}^2 - 2r_1 d\sin(\alpha) + d^2} \end{aligned}$$

$$\alpha = \arcsin\left(\frac{d^2 + {r_1}^2 - {r_2}^2}{2dr_1}\right) \tag{2.8}$$

$$\beta = -\arctan\left(\frac{d\cos(\alpha)}{r_1 - d\sin(\alpha)}\right) \tag{2.9}$$

where d is the distance between transducers, α is the angle between the line that connects the transducers and the corner feature, β is the angle between sonar bearings corresponding to r_1 and r_2, and T' and R' are virtual images of T and R, respectively. As shown, the corner feature ranges share the same relation as the line feature, except by an inverse sign on the angle β.

The corner parameters are calculated based on the angle α from Equation 2.8, the position and orientation of the transducer T (x_T,y_T,ϕ_T), and the angle (γ) between the transducer T orientation and the normal to the line connecting the transducers. Notice that they are identical to the line feature parameters.

$$x_c = x_T + \frac{r_1}{2}\cos(\phi_T + \xi) \tag{2.10}$$

$$y_c = y_T + \frac{r_1}{2}\sin(\phi_T + \xi) \tag{2.11}$$

$$\xi = \gamma + \alpha \tag{2.12}$$

Also notice that the orientation of the corner cannot be extracted since r_1 and r_2 depend only on the corner position (x_c, y_c) and the transducers' relative position, as shown by the shaded triangle on Figure 2.2.

2.1.3 Edge Feature

The edge feature model also uses the pair of sonar ranges and the current position and orientation estimate of the transducer T to create the edge position estimate (x_e, y_e). The assumptions required for modeling an edge feature are distinct from the previous assumptions. For edge feature modeling, the point of reflection is assumed to be relatively independent of the sonars position, and the edge must be a high curvature convex corner [1]. The physical phenomenon that causes the reflections on a sharp edge is modeled as pure diffusion, and not specular reflection as on the other features.

Figure 2.3 shows typical reflections on a high curvature edge surface, where the following relations are extracted:

$$\begin{aligned} r_2 &= \frac{r_1}{2} + \sqrt{(\frac{r_1}{2})^2 + d^2 - r_1 d \sin(\alpha)} \\ \alpha &= \arcsin\left(\frac{d^2 + r_1 r_2 - {r_2}^2}{d r_1}\right) \end{aligned} \tag{2.13}$$

where d is the distance between transducers, α is the angle between the line that connects the transducers and the edge feature, and T' and R' are virtual images of T and R, respectively. Notice that β, the angle between sonar bearings corresponding to r_1 and r_2, is zero.

The following derivation of the edge parameters, similarly to the previous features, uses the angle α from Equation 2.13, the position and orientation of the

---

[1] High curvature in this situation are curvatures with radius ψ smaller than the wavelength of the ultrasonic signal transmitted, $\psi < 7mm$.

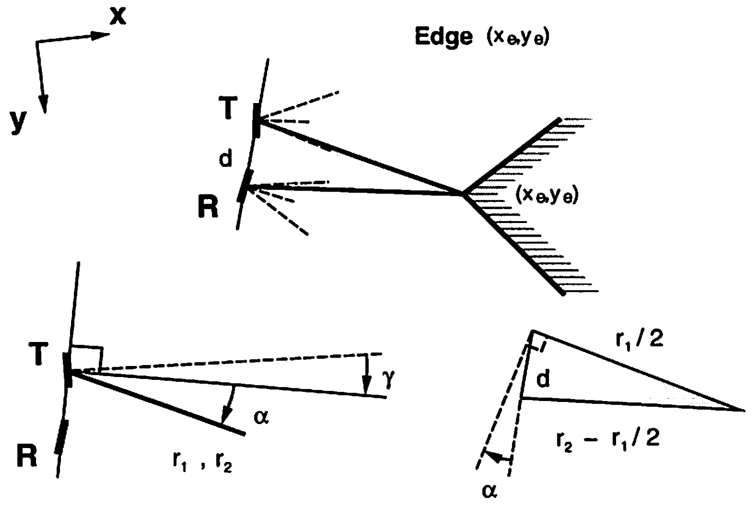


Figure 2.3. Edge reflection

transducer T (x_T,y_T,ϕ_T), and the angle (γ) between the transducer T orientation and the normal to the line that connects the transducers.

$$x_e = x_T + \frac{r_1}{2}\cos(\phi_T + \xi) \tag{2.14}$$

$$y_e = y_T + \frac{r_1}{2}\sin(\phi_T + \xi) \tag{2.15}$$

$$\xi = \gamma + \alpha \tag{2.16}$$

2.1.4 Feature Localization Error

To complete the derivation of the feature localization models, it is necessary to estimate the uncertainty associated with each feature parameter; such uncertainty is a function of the uncertainty in the sonar measurements r_1 and r_2, given by Δr_1 and Δr_2 respectively, the non-linear transformations from measurement space to feature space, and the transducers spatial configuration together with beam angles.

Two different approaches were tested; an *analytic* approach and a *geometric* approach. Given the error in sonar measurements, the analytic approach is based on the computation of the maximum possible error generated by the transformations from measurement to feature space. The geometric approach uses the configuration of the transducers, their beam angles, and the uncertainty on their readings to calculate the region in space that might contain the feature correct position. The uncertainty in the feature localization can then be calculated indirectly from the resulting estimate. A brief description of both methods used to estimate the feature localization error follows; a more detailed description can be found in Appendix A.

Both approaches are conservative in their error estimate. The analytic method is very conservative since it uses the maximum absolute error of the non-linear feature

Feature Error Calculation (Analytic Method):

Compute the maximum error of a feature parameter $u : u \in \{r, \theta, x_e, y_e, x_c, y_c, \xi\}$ *by:*

1. *Calculating the extrema of the feature parameter function* f_u *within the range* $r_1 \pm \Delta r_1$*, and* $r_2 \pm \Delta r_2$.
2. *Returning the maximum absolute error of the feature parameter* (Δu)*, if there was no constraint violation on the measurement range tested; otherwise, ignore this measurement.*

Feature Error Calculation (Geometric Method):

1. *Calculate the region where the sonar reflections could occur, given the sonar measurements range* ($r_1 \pm \Delta r_1$, $r_2 \pm \Delta r_2$)*, the sonar configuration, the beam angles, and the type of feature under analysis.*
2. *Return the error in the feature localization* (Δr,$\Delta \theta$,Δx_e,Δy_e, Δx_c*, or* Δy_c) *given the above region, if such region exist; otherwise, ignore the measurement.*

localization formulas. The geometric method is as accurate as the sonar beam angle selected. The main advantage of the geometric over the analytic approach is the direct association between sensor configuration and precision of the measurement. In principle, this permits the process of computing the feature localization error to be inverted, leading to techniques for selecting the sonar configuration for more precise measurements.

2.2 Extracting Features from Sonar Information

As features are extracted from the environment, a filtering process is used to overcome measurement noise. For normally distributed error processes, filtering normally produces better feature hypotheses by fusing measurements over multiple observations [17, 1, 10].

Figure 2.4 depicts the process of extracting feature localization hypotheses from sonar ranging. The ranging information is filtered (*Measurement Filter*) creating consistent sonar measurement pairs – consistency in this case means that both reflections come from the same feature. Then each pair can be transformed into feature evidence using the feature models. Over time, feature localization evidence from independent observations are fused in a recursive filter. The first method (method #1 in Figure 2.4) uses Extended Kalman Filters based on the feature models to convert from measurement space to feature-hypotheses automatically. The second method (method #2) initially transforms a measurement into supporting evidence for all three features (line, edge, and corner) using the feature models and then uses Linear Kalman Filters to incorporate the feature evidence into the feature hypothesis

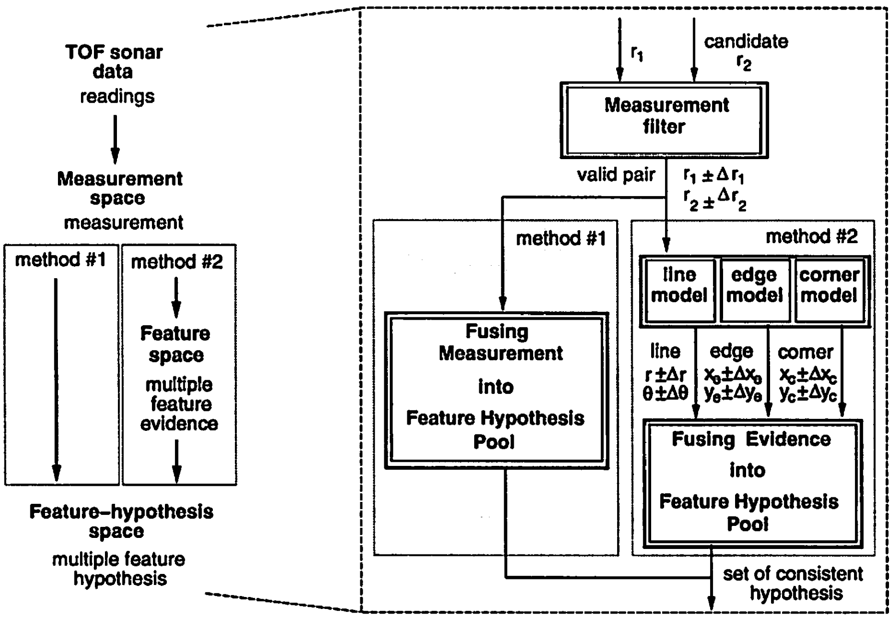


Figure 2.4. Feature localization procedure.

pool. The derivation of the extended and Linear Kalman Filters used by the line, edge, and corner features are the subject of Appendix B.

2.2.1 Measurement Space

A measurement in this space is composed of two sensor readings (r_1 and r_2) from the same sonar firing event, where the sonars are in a T/R functional configuration. Moreover, these readings come from reflections on the same feature in the environment. To avoid readings from multiple features, a filter based on a relation derived from the feature models is employed. Equations 2.1, 2.8, and 2.13 express the relation between the sonar reflections (r_1 and r_2), and the distance between the transducers (d) that cannot be violated by any of the features. The formulas below show the constraints imposed on r_1 and r_2.

$$\left| \frac{d^2 + {r_1}^2 - {r_2}^2}{2dr_1} \right| \leq 1 \tag{2.17}$$

$$\left| \frac{d^2 + r_1 r_2 - {r_2}^2}{dr_1} \right| \leq 1 \tag{2.18}$$

These equations evaluate to the same relation since r_1, r_2, and d are positive numbers and $r_1 > d$:

$$|r_2 - r_1| \leq d \tag{2.19}$$

Therefore, all readings that violate this relation are disregarded as measurements, eliminating, to some extent, readings that are not consistent with the feature models.

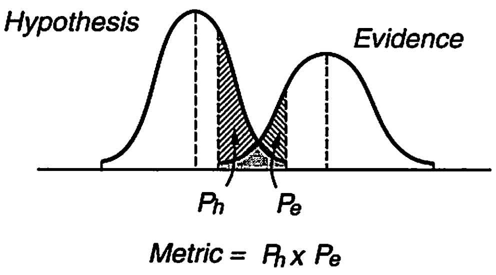


Figure 2.5. Feature-hypothesis update metric – one dimensional Gaussians were used here for illustration purposes.

2.2.2 Feature Space and Feature-hypothesis Space

In feature space the information is expressed as multiple evidence, composed by line feature evidence ($r \pm \Delta r$, $\theta \pm \Delta\theta$), edge evidence ($x_e \pm \Delta x_e$, $y_e \pm \Delta y_e$), and corner evidence ($x_c \pm \Delta x_c$, $y_c \pm \Delta y_c$). Multiple hypotheses are generated by transforming each sonar measurement pair (r_1, r_2) using the respective feature models.

Each feature hypothesis represents a feature in the environment, and is obtained by fusing evidence over time and from different robot poses. Each hypothesis is interpreted by three Kalman Filters, one for each sonar feature type, creating a feature-hypothesis triple.

2.2.3 Feature-hypothesis Update

In the process of updating the feature-hypothesis triple, the fusion process must determine whether a new observation belongs to an existing feature-hypothesis in the pool. The metric used computes the probability that samples drawn from both distributions simultaneously lie in the overlapping volumes, expressing the similarity between the hypothesis and the measurement. This probability is obtained by calculating the product of the overlapping volumes of two 2D-Gaussian distributions ($N((x,y),((\frac{\Delta x}{3})^2,(\frac{\Delta y}{3})^2))$ for edges and corners, or $N((r,\theta),((\frac{\Delta r}{3})^2,(\frac{\Delta \theta}{3})^2))$ for lines) generated by the Kalman Filter estimate and the feature measurement evidence, as shown in Figure 2.5. The metric assumes values between 0 and 1, and is equal to 1 when the distributions are identical, and zero when there is no overlap between them[2]. A more detailed description on how to select the hypothesis to fuse the new feature evidence follows.

---

[2] Both Gaussian distributions were truncated on the $\pm 3\sigma$ points. And, in the case of one distribution being contained in the other, a factor ($thr = 0.3$) is added to the metric, biasing the system to avoid the creation of multiple similar hypotheses for the same feature.

Selecting in which hypothesis to fuse the new observation:

1. *Compute the similarity metric for each feature-hypothesis in the pool of hypotheses, and for each feature type;*
2. *Search for the hypothesis that has the higher metric value;*
3. *Return the hypothesis' id if its metric value exceeded a threshold ($thr = 0.3$); otherwise create a new hypothesis.*

2.3 Feature Identification

The degree of belief in a feature hypothesis is determined by the diagonal terms of the covariance matrix of the corresponding filter ($\sigma^2 < 0.001$). When the belief in a feature hypothesis is sufficiently high, the next step is to identify the feature-type which best accounts for the data. This process is called feature identification, and is the last step in feature extraction. The approach used calculates a confidence measurement for each feature in a hypothesis, and selects the feature with a higher relative confidence value. This confidence discrimination measure is calculated as follows:

Feature confidence measurement:

1. *On the last n measurements ($n \leq 20$) fused into the hypothesis, use their corresponding sonar configurations to:*
    (a) *Compute which sonar measurements the current filter feature estimate generates;*
    (b) *Compare the above measurements with the original measurements fused, by using the same metric used to select in which hypothesis to fuse a new evidence;*
2. *Compute each feature confidence by taking the average of all the metric values over the n measurements;*
3. *Select the feature with the best confidence value, higher (0.2) than the second best on this hypothesis.*

The success of this approach depends on the quality of the information fused. The feature identification process is computational expensive; it requires information from multiple sonar configurations and sometimes even different robot poses to correctly distinguish the three possible features. Therefore, any passive sensing procedure is in principle inappropriate for this task and the acquisition of information should be active based on previous knowledge of how and where the relevant information can be acquired. This knowledge can be obtained in part by analyzing the task requirements and can be augmented by reasoning on the basis of past experiences.

CHAPTER 3

EXPERIMENTS

3.1 Experiments in Feature Localization and Identification

All the experiments reported here were obtained using a 2D simulator of sonar reflections on a specular environment composed of lines, edges, and corners features. The simulator uses the feature models, a simple model of the ultrasonic sensor that considers range and beam angle, and a Gaussian noise process to corrupt the sonar returns (Appendix C).

This simulator was developed for testing the feature localization approaches based on linear and Extended Kalman Filters and also to characterize the importance of sonar configuration in feature extraction.

The experiments are divided in three parts. The first part presents results of feature extraction at different locations of the same environment using the Extended Kalman Filter and the Linear Kalman Filter approach, followed by a comparison of both methods. The second experiment explores the detection of features by fusing information from multiple robot poses and possible active sensing strategies. And the last experiment addresses the impact of sonar configuration on feature extraction. All the experiments were done using the analytical method in the feature error calculation.

3.2 Part 1: Experiments on Feature Extraction and Filtering Approaches

In this experimental part, we selected 3 robot poses in the same environment, composed of 12 features (lines, corners, and an edge), approximately 6 by 3 m. All simulator snapshots were taken after 3 full scans ($\approx$ 300 firings – 4 firings every 15 degrees, interleaving TR and RT configurations) of a rotating 2-sonar array. In the figures, the robot is represented by the "T" shaped bar, and the sonars, localized in the ends of the bar ($d = 0.6m$), are aligned in the direction of the robot. The absolute localization error of the features identified at the end of each full scan is reported on the corresponding tables (a dash on the table represents a feature not identified at that moment). No feature was misclassified on the following examples.

3.2.1 Extended Kalman Filter Approach

An Extended Kalman Filter (EKF) is employed when the system cannot be adequately described by a linear model. In our case we used a first-order EKF to directly estimate the position of a 2D feature (line, edge, or corner) in a static

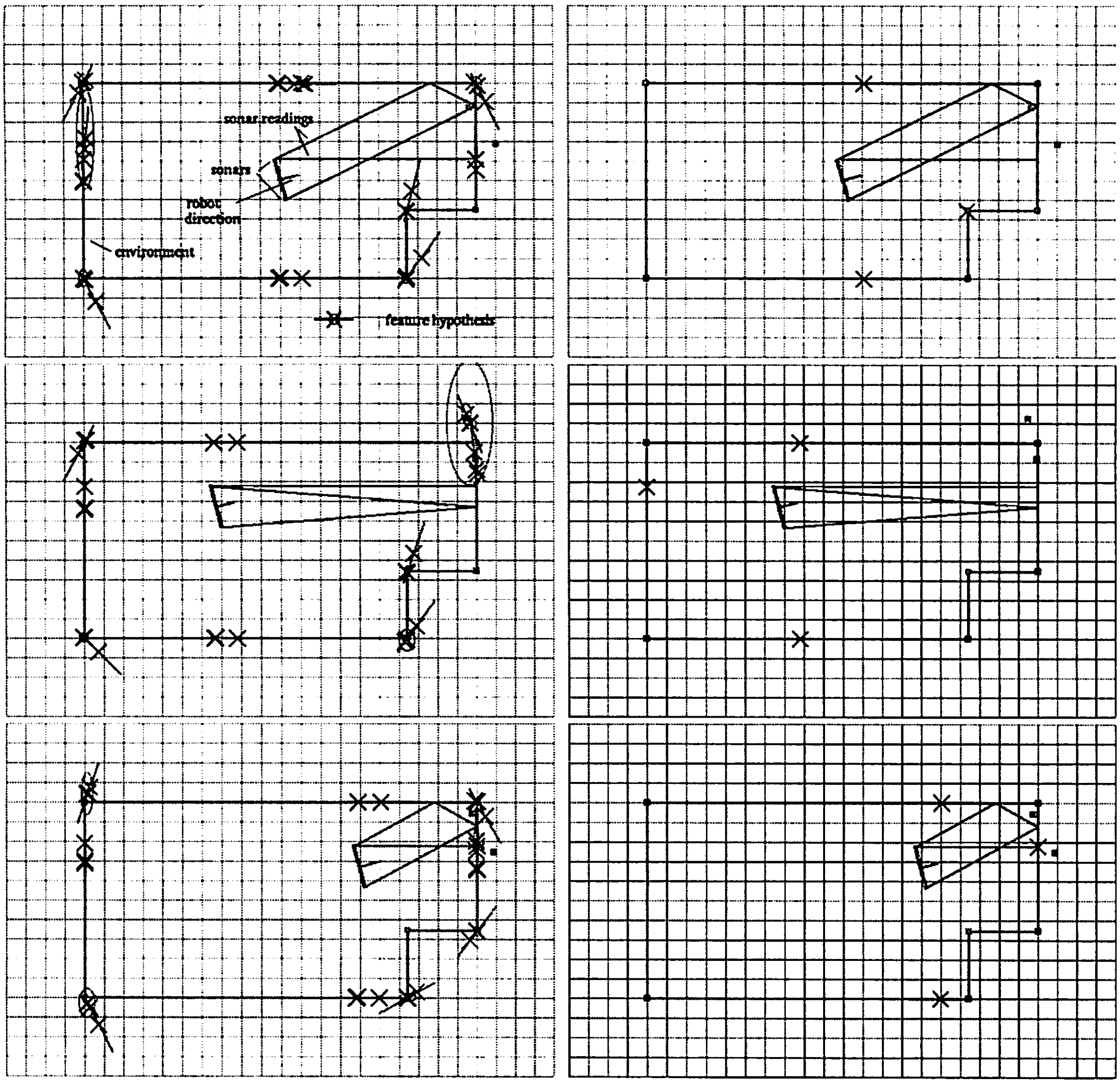


Figure 3.1. Simulator snapshots of a rotating 2-sonar array using the EKF method.

environment, using the discrete-time measurements obtained from the two sonar sensors (r_1, r_2), as described in Chapter 2 and Appendix B.

Figure 3.1 presents snapshots from 3 robot poses after 3 full scans as previously described. The left column presents all the hypothesis gathered during the scans and the right column shows the features identified. As shown, the system was able to classify correctly several nearby features ($< 2\ m$ from the robot). The absolute localization error of the features extracted on each sonar scan is reported on Table 3.1, where the localization error is on average one order of magnitude better than the uncertainty of the raw data (1% of the readings). These tables directly correspond to the previous snapshots.

Scan #			1	2	3	4
Total # of features identified			3	3	3	3
Total # of hypothesis created			9	9	9	9
Absolute feature error r, x, y in m θ in rad	Right-lower edge	x	< 0.001	0.001	0.002	0.003
		y	0.006	0.004	0.004	0.008
	Bottom wall	r	0.036	0.002	0.011	0.002
		θ	0.011	< 0.001	0.004	0.001
	Top wall	r	0.008	0.010	0.010	0.007
		θ	0.003	0.003	0.004	0.003

Scan #			1	2	3	4
Total # of features identified			3	3	3	3
Total # of hypothesis created			8	8	9	9
Absolute feature error r, x, y in m θ in rad	Bottom wall	r	0.013	0.008	0.008	0.002
		θ	0.007	0.003	0.003	0.001
	Left-side wall	r	0.003	0.004	0.001	0.007
		θ	0.002	0.003	0.002	0.006
	Top wall	r	0.004	0.003	0.003	< 0.001
		θ	0.002	0.002	0.002	< 0.001

Scan #			1	2	3	4
Total # of features identified			3	3	3	3
Total # of hypothesis created			10	10	10	10
Absolute feature error r, x, y in m θ in rad	Right-side wall	r	0.006	0.005	< 0.001	< 0.001
		θ	0.005	0.004	< 0.001	0.002
	Bottom wall	r	0.013	0.027	0.033	0.032
		θ	0.003	0.006	0.008	0.008
	Top wall	r	0.005	0.009	0.006	0.009
		θ	0.001	0.002	0.001	0.002

Table 3.1. Feature extraction results from 2-sonar array using the EKF method.

3.2.2 Linear Kalman Filter Approach

In this section, we present the same case examples as before, but with the Linear Kalman Filter (KF) approach. As described in Chapter 2 and Appendix B, a Linear Kalman Filter was employed after computing the feature position and error associated with each pair of measurements (r_1, r_2), simplifying the data fusion by executing directly in feature space (linear data fusion).

Figure 3.2 presents snapshots from the same 3 robot poses as before. The left column presents all the hypothesis gathered during the scans and the right column shows the features identified. Again, the system was able to classify correctly almost all nearby features ($< 2\, m$ from the robot). Table 3.2 reports the absolute localization error of the features extracted on each sonar scan, and the localization error is on average in the order of millimeters.

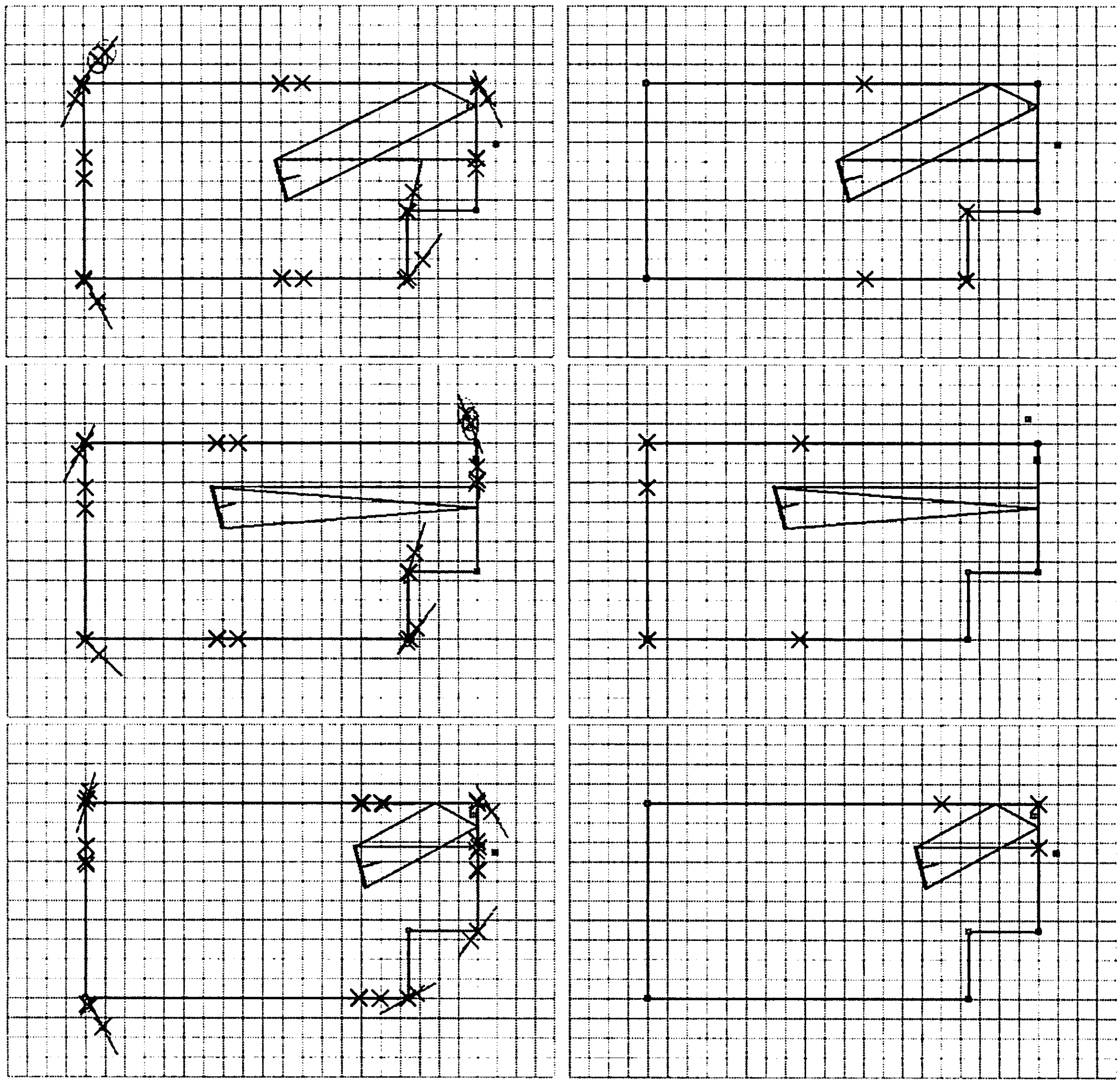


Figure 3.2. Simulator snapshots of a rotating 2-sonar array using the KF approach.

3.2.3 Comparison Between Filtering Approaches

Both filtering methods, the EKF and the KF, demonstrated to be effective in localizing the features given the sonar readings. They were equally able to localize line, edge, and corner features with an average absolute error in the order of millimeters.

The only discrepancy was in the number of features identified, and in the number of evidence required by the filtering process before an identification; more features where identified using the KF approach, however, the EKF approach was in some cases faster identifying the features.

The EKF method was better identifying line and edge features, but it identified less features by failing to identify the corner features. This happened because the variance of the corner feature was higher than the threshold used in identification

Scan #			1	2	3	4
Total # of features identified			0	2	4	4
Total # of hypothesis created			9	10	10	10
Absolute feature error r, x, y in m θ in rad	Right-lower edge	x	–	0.002	0.002	0.003
		y	–	0.006	0.006	0.010
	Right-lower corner	x	–	0.010	0.008	0.017
		y	–	0.019	0.017	0.027
	Bottom wall	r	–	–	0.004	0.021
		θ	–	–	0.002	0.006
	Top wall	r	–	–	0.023	0.013
		θ	–	–	0.007	0.004

Scan #			1	2	3	4
Total # of features identified			2	4	5	5
Total # of hypothesis created			8	8	9	9
Absolute feature error r, x, y in m θ in rad θ in rad	Left-side wall	r	0.001	0.002	0.002	0.007
		θ	0.006	0.001	< 0.001	0.010
	Top wall	r	0.001	0.005	0.001	0.002
		θ	0.001	0.002	< 0.001	< 0.001
	Bottom wall	r	–	0.014	0.011	0.011
		θ	–	0.006	0.004	0.005
	Left-upper corner	x	–	0.001	0.002	< 0.001
		y	–	0.005	0.004	0.006
	Left-lower corner	x	–	–	0.006	0.004
		y	–	–	0.017	0.008

Scan #			1	2	3	4
Total # of features identified			1	2	3	3
Total # of hypothesis created			10	10	11	10
Absolute feature error r, x, y in m θ in rad	Right-side wall	r	0.013	0.010	0.003	0.011
		θ	0.011	0.009	0.001	0.008
	Right-upper corner	x	–	0.001	0.001	0.003
		y	–	0.006	0.005	0.005
	Top wall	r	–	–	0.004	0.003
		θ	–	–	< 0.001	< 0.001

Table 3.2. Feature extraction results from 2-sonar array using the KF approach.

($\sigma^2 < 0.001$). And, since the data fused were similar in both approaches, the higher variance may be explained by the first-order EKF been unable to cope with the nonlinearities of the system. Therefore, the implementation of a second-order EKF may be necessary for the identification of corner features.

The main advantage of using an extended Kalman Filter is to have a direct implementation, where the transformation from measurement space to feature space is automatically done by the filter (direct approach). However, the filter derivation is usually difficult, especially when a high-order filter is necessary, and in several cases it is preferable to separate the space transformation from the filtering process (indirect approach). After testing both methods, we conclude that, in our application, the direct approach was easier to implement than the indirect approach; however the analyze of the feature error associated with each measurement pair and sensors' configuration gave us new insights on possible active sensing strategies to maximize the information extracted from the measurements.

3.3 Part 2: Feature Extraction over Multiple Poses

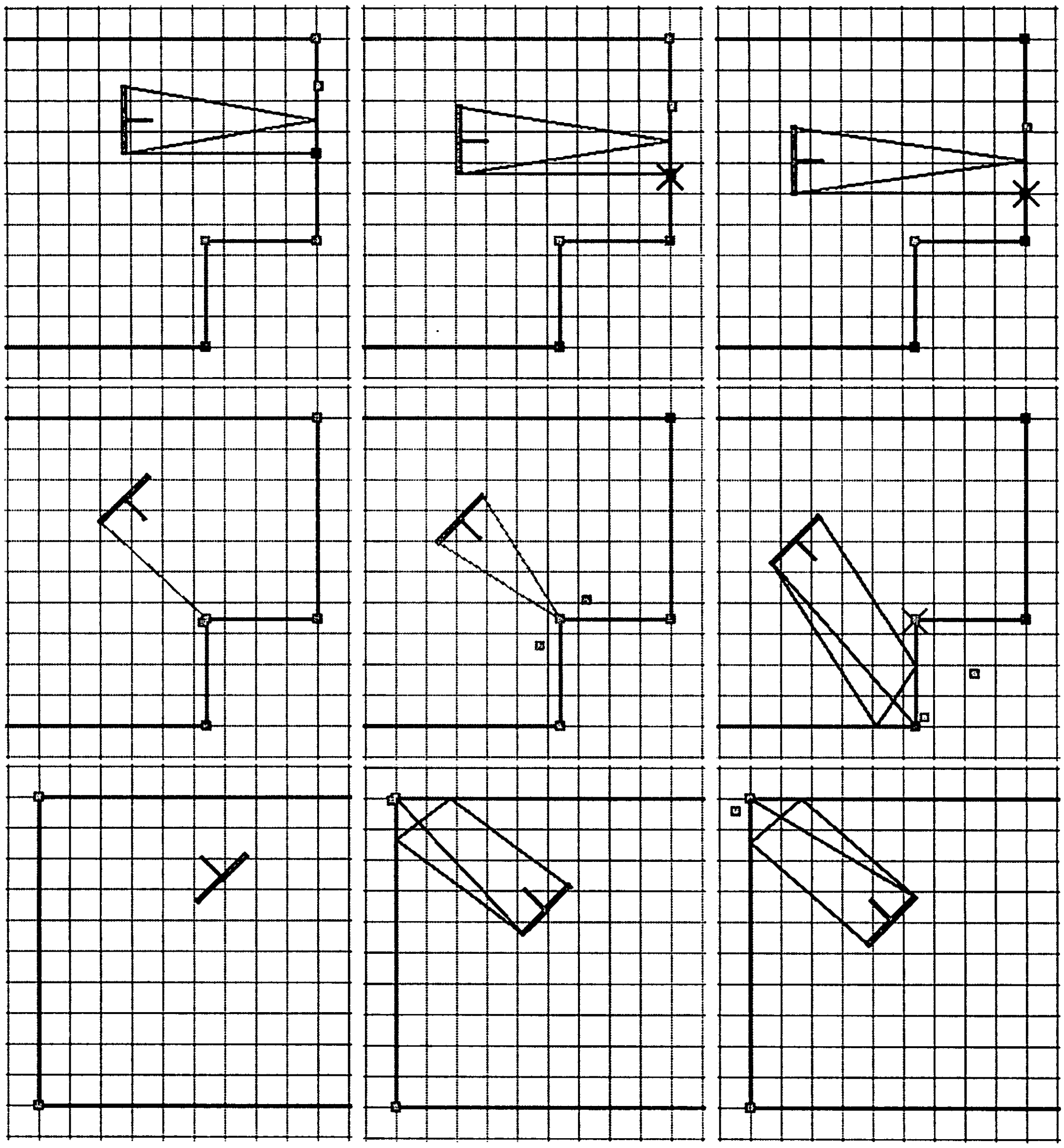


Figure 3.3. Simulator snapshots of a rotating 2-sonar array over multiple poses, using the KF approach.

The objective of this experiment was to show how an active sensing strategy could improve and accelerate the feature extraction process. The experiment is divided in 3 parts, corresponding to the line, edge, and corner feature extraction procedures. The data were collected over 5 aligned, equally spaced robot poses, and the sensors' orientation was kept constant during the experiment. Figure 3.3 presents 3 simulator snapshots of each experimental part at the extrema and middle robot poses. The rightmost snapshots corresponds to the first sonar firing, the middle snapshots to third firing event, and the leftmost to the fifth one.

In this experiment, the line feature was identified after 3 sonar firings (or evidence fused) with absolute feature error in the order of millimeters ($\Delta r = 0.003$, $\Delta\theta = 0.002$). Notice that this feature was not identified in the previous experiments from a single robot pose (see Figure 3.2 and Table 3.2, first experiment). Similarly, the edge feature was identified after 4 evidence being fused with absolute feature error of ($\Delta x < 0.001$, $\Delta y = 0.001$); and, as before, the system was unable to identify this feature from a single pose (see Figure 3.2 and Table 3.2, last experiment). The corner feature required 6 evidence, contrasting with 17 evidence needed from a fixed, near ideal robot pose, with absolute feature error of ($\Delta x = 0.001$, $\Delta y = 0.003$).

3.4 Part 3: Comparison Between Sonar Array Configurations

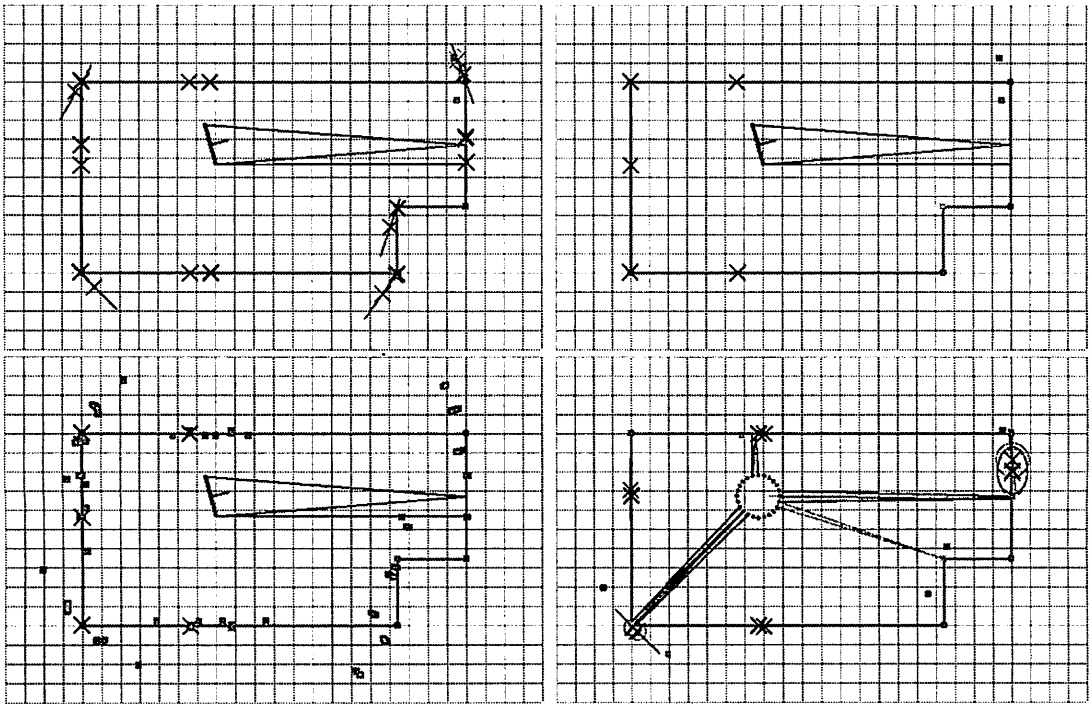


Figure 3.4. Simulator snapshots of a rotating 2-sonar array (first three snapshots) and a 24 sonar ring (last snapshot).

Following the experimental procedure of the first experiment (Section 3.2), Figure 3.4 compares the performance of a rotating 2-sonar array and a 24 sonar ring.

The first snapshot shows all the features localized, and the second (left on the figure) presents the features identified (all 5 correctly identified). In the bottom row, the left snapshot presents the raw data used on the previous examples (dots). The last snapshot (right on the picture) presents all the hypotheses created by a ring after the same amount of firings, but, in this case, no feature was identified.

Both sonar configurations have advantages and drawbacks. As demonstrated, the 2-sonar array not only detected more features than the ring, but also correctly identified all the nearby features. On the other hand, the ring was able to localize 4 out of 5 features faster than the 2-sonar array, and it is an effective configuration for obstacle avoidance. In a navigation task a synergetic relation can be created where a sonar ring can contribute by directing a rotating sonar array to places with high probability of finding a landmark, navigating reliably and avoiding obstacles.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

This technical report presented a procedure based on multiple hypothesis testing for localizing and identifying indoor features using sonar data, demonstrating that accurate feature information can be acquired with the use of an adequate sonar model and configuration. The results presented shows a direct association between sensor configuration and localization precision, suggesting the possibility of creating sonar controllers capable of extracting better information by actively exploiting sensor configuration.

4.1 Active Sonar System

An active sonar sensor is being developed to further validate the results presented in here. Figure 4.1 shows our mobile robot and Figure 4.2 shows the functional diagram of the two-sonar array (left), where both transducers are mounted on an axis that can pan and tilt, and each transducer has an extra pan degree of freedom. The pan and tilt of the axis affords the localization of features at any direction, and at any height respectively. In addition, the extra verge degree of freedom (DOF) on each transducer allows the system to adjust the sonar configuration to feature distance and feature type, thereby producing better quality measurements. In the right side of Figure 4.2, a TRC stereo head system that will serve as testbed for the 2-sonar active array system is presented. More details about our mobile robot (Isaac) and the active sonar system can be found in Appendix D.

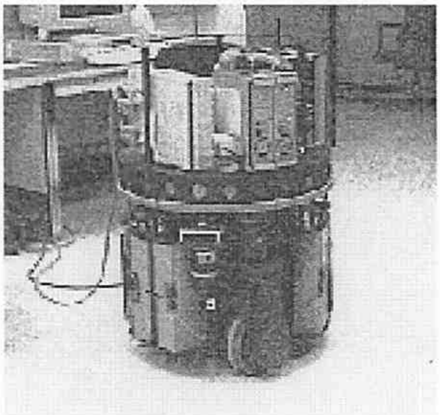


Figure 4.1. Mobile robot and proposed active sonar.

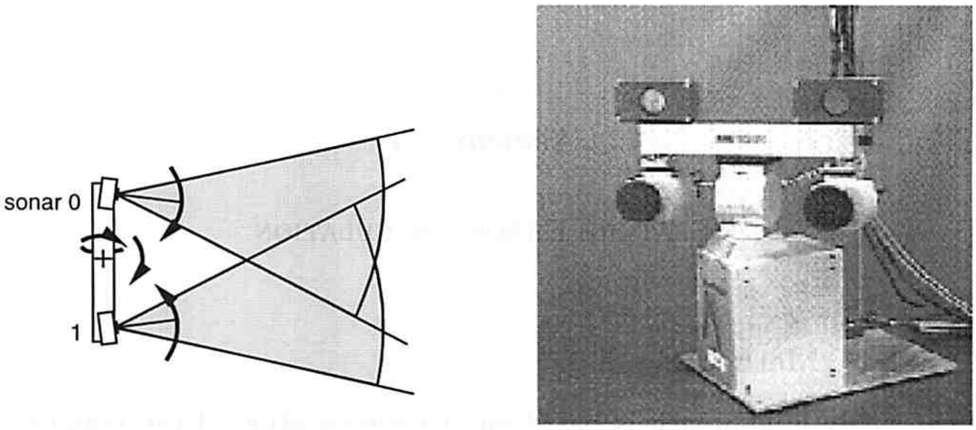


Figure 4.2. Active sonar system

This work supports robot navigation, by allowing the identification of landmarks even in presence of measurement noise. By detecting common features in indoor environments and using these features as landmarks, a robot can navigate reliably, recovering its pose when necessary.

APPENDIX A

FEATURE ERROR CALCULATION

A.1 Analytic Method

The analytic approach is based on the computation of the maximum error on feature space, given the measurement pair (r_1, r_2), their correspondent error $(\Delta r_1, \Delta r_2)$, and the non-linear transformations from measurement to feature space. The algorithm below summarizes the procedure used to transform errors in measurement space to feature space.

Feature Error Calculation (Analytic Method):

Compute the maximum error of a feature parameter $u : u \in \{r, \theta, x_e, y_e, x_c, y_c, \xi\}$ *by:*

1. *Calculating the extrema* $(e_{r_1 i}, e_{r_2 i}) : i \in E$, *where* E *is the set of all possible maxima and minima of the feature parameter function* f_u *within the range* $r_1 \pm \Delta r_1$, *and* $r_2 \pm \Delta r_2$.
2. *Returning the maximum absolute error of the feature parameter* (Δu), *if there was no constraint violation on the measurement range tested; otherwise, ignore this measurement.*

$$\Delta u = \max_{i \in E} |f_u(r_1, r_2) - f_u(e_{r_1 i}, e_{r_2 i})|$$

The non-linear transformations from measurement space to each feature (previously described in Chapter 2), their partial derivatives with respect to the measurement, and the tests used to identify violation of constraints on the range of possible measurements $(u \pm \Delta u)$ are presented in the following sections. Notice that r_1 and r_2 represent variables now and not a measurement.

Mathematical software packages were employed on the calculation of the extrema. *Mathematica*[1] was the software package used, except for the extrema calculation relative to the line feature ($\frac{\partial r}{\partial r_1} = 0$) that was solved using *Maple*[2], because *Mathematica* was unable to solve it.

The solutions obtained present zeros that are not zeros of the initial function; the extra roots are introduced by the presence of square roots in the derivatives. Therefore, an online check of the solutions to determine the true extrema of the

---

[1] Mathematica 2.2 - ©Wolfram Research, Inc.

[2] Maple V release 2 - ©Waterloo Maple, Inc.

initial function is necessary. Moreover, there are physical factors that restrict the extrema, such as r_1 and r_2 being positive measurements limited by the sonar range, d being limited by the diameter of the mobile robot, the origin frame being conveniently selected to reside out of the robots space, and ξ being limited to the transmitter cone angle.

A.1.1 Edge Feature

- The feature parameter functions are:

$$\xi = \arcsin\left(\frac{d^2 + r_1 r_2 - {r_2}^2}{d r_1}\right) + \gamma \ ,$$

$$x_e = x_T + \frac{r_1}{2}\cos(\phi_T + \xi) \ , \tag{A.1}$$

$$y_e = y_T + \frac{r_1}{2}\sin(\phi_T + \xi) \tag{A.2}$$

  where (x_T,y_T,ϕ_T) are the position and orientation of the transducer T, γ is the angle between the transducer T orientation and the normal to the line that connects the transducers, and d is the distance between transducers.

- The corresponding partial derivatives are:

$$\frac{\partial \xi}{\partial r_1} = \frac{-d^2 + {r_2}^2}{d\,{r_1}^2\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}} \ , \qquad \frac{\partial \xi}{\partial r_2} = \frac{r_1 - 2\,r_2}{d\,r_1\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}} \ ,$$

$$\frac{\partial x_e}{\partial r_1} = \frac{r_1\left(\frac{\left(d^2 - {r_2}^2\right)\left(d^2 + r_1\,r_2 - {r_2}^2\right)\cos(\phi_T + \gamma)}{d^2\,{r_1}^3\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}} - \frac{r_2\sin(\phi_T + \gamma)}{d\,r_1}\right)}{2} + \frac{\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}\cos(\phi_T + \gamma)}{2} \ ,$$

$$\frac{\partial x_e}{\partial r_2} = \frac{r_1\left(-\frac{\left(r_1 - 2\,r_2\right)\left(d^2 + r_1\,r_2 - {r_2}^2\right)\cos(\phi_T + \gamma)}{d^2\,{r_1}^2\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}} - \frac{\left(r_1 - 2\,r_2\right)\sin(\phi_T + \gamma)}{d\,r_1}\right)}{2} \ ,$$

$$\frac{\partial y_e}{\partial r_1} = \frac{r_1\left(\frac{\left(d^2 - {r_2}^2\right)\left(d^2 + r_1\,r_2 - {r_2}^2\right)\sin(\phi_T + \gamma)}{d^2\,{r_1}^3\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 + {r_2}^2\right)^2}{d^2\,{r_1}^2}}} - \frac{r_2\cos(\phi_T + \gamma)}{d\,r_1}\right)}{2} + \frac{\sqrt{1 - \frac{\left(d^2 + r_1\,r_2 - {r_2}^2\right)^2}{d^2\,{r_1}^2}}\sin(\phi_T + \gamma)}{2} \ ,$$

$$\frac{\partial y_e}{\partial r_2} = \frac{r_1 \left(-\frac{(r_1-2\,r_2)\left(d^2+r_1\,r_2-{r_2}^2\right)\,\sin(\phi_T+\gamma)}{d^2\,{r_1}^2\,\sqrt{1-\frac{\left(d^2+r_1\,r_2-{r_2}^2\right)^2}{d^2\,{r_1}^2}}} + \frac{(r_1-2\,r_2)\,\cos(\phi_T+\gamma)}{d\,r_1} \right)}{2}$$

- Extrema calculation:
  The extrema of a bounded 2D function could be on the boundaries or on the interior. On the boundaries, the extrema could be at the four corners, or at the edges of the region defined by ($r_1 \pm \Delta r_1$ and $r_2 \pm \Delta r_2$). All the possible extrema points within the measurement range are used to compute the maximum absolute error of the feature parameter.

  - Finding extrema on the borders:
    - Solving $\frac{\partial \xi}{\partial r_1} = 0$

      The function ξ is strictly monotonic with respect to r_1.

    - Solving for $\frac{\partial \xi}{\partial r_2} = 0$

      $$r_2 = \frac{r_1}{2}$$

    - Solving $\frac{\partial x_e}{\partial r_1} = 0$

      $$r_1 = r_2 \pm \frac{\sqrt{2\,d^2\,{r_2}^2\,\left(-d^2+2\,{r_2}^2-d^2\,\cos(2\,(\phi_T+\gamma))\right)\,\sin(\phi_T+\gamma)^2}}{d^2-2\,{r_2}^2+d^2\,\cos(2\,(\phi_T+\gamma))}$$

    - Solving $\frac{\partial x_e}{\partial r_2} = 0$

      $$\begin{aligned} r_2 &= \frac{r_1}{2}, \\ r_2 &= \frac{r_1 \pm \sqrt{4\,d^2+{r_1}^2 \pm 4\,\sqrt{d^2\,{r_1}^2\,\sin(\phi_T+\gamma)^2}}}{2} \end{aligned}$$

    - Solving $\frac{\partial y_e}{\partial r_1} = 0$

      $$r_1 = r_2 \pm \frac{\sqrt{2\,d^2\,{r_2}^2\,\left(-d^2+2\,{r_2}^2+d^2\,\cos(2\,(\phi_T+\gamma))\right)\,\cos(\phi_T+\gamma)^2}}{d^2-2\,{r_2}^2-d^2\,\cos(2\,(\phi_T+\gamma))}$$

    - Solving $\frac{\partial y_e}{\partial r_2} = 0$

      $$\begin{aligned} r_2 &= \frac{r_1}{2}, \\ r_2 &= \frac{r_1 \pm \sqrt{4\,d^2+{r_1}^2 \pm 4\,\sqrt{d^2\,{r_1}^2\,\cos(\phi_T+\gamma)^2}}}{2} \end{aligned}$$

  - Find interior extrema:
    - Solving $\frac{\partial\xi}{\partial r_1} = 0$ and $\frac{\partial\xi}{\partial r_2} = 0$;
    - Solving $\frac{\partial x_e}{\partial r_1} = 0$ and $\frac{\partial x_e}{\partial r_2} = 0$;
    - Solving $\frac{\partial y_e}{\partial r_1} = 0$ and $\frac{\partial y_e}{\partial r_2} = 0$.

    The functions ξ, x_e, and y_e are strictly monotonic with respect to r_1 and r_2, thus presenting no solution.

- Measurements' constraints:
  Derived from the previous equations, the constraints imposed are:

$$d,\ r_1 \neq 0 \quad , \qquad |r_2 - r_1| < d \quad , \qquad r_2 > d$$

A.1.2 Corner Feature

- The feature parameter functions are:

$$\xi = \arcsin\left(\frac{d^2 + r_1^2 - {r_2}^2}{2dr_1}\right) + \gamma \tag{A.3}$$

The other parameters functions x_c and y_c are the same as in the case of an edge feature (Equations A.1 and A.2).

- The corresponding partial derivatives are:

$$\frac{\partial\xi}{\partial r_1} = \frac{-d^2 + {r_1}^2 + {r_2}^2}{2\,d\,{r_1}^2\sqrt{1 - \frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}} \quad , \qquad \frac{\partial\xi}{\partial r_2} = -\frac{r_2}{d\,r_1\sqrt{1 - \frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}} \ ,$$

$$\frac{\partial x_c}{\partial r_1} = \frac{r_1\left(\frac{\left(d^4-{r_1}^4-2\,d^2\,{r_2}^2+{r_2}^4\right)\cos(\phi_T+\gamma)}{4\,d^2\,{r_1}^3\sqrt{1-\frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}} - \frac{\sin(\phi_T+\gamma)}{d}\right)}{2} + \frac{\sqrt{1 - \frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}\,\cos(\phi_T+\gamma)}{2} \ ,$$

$$\frac{\partial x_c}{\partial r_2} = \frac{r_1\left(\frac{r_2\left(d^2+{r_1}^2-{r_2}^2\right)\cos(\phi_T+\gamma)}{2\,d^2\,{r_1}^2\sqrt{1-\frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}} + \frac{r_2\,\sin(\phi_T+\gamma)}{d\,r_1}\right)}{2} \ ,$$

$$\frac{\partial y_c}{\partial r_1} = \frac{r_1\left(\frac{\left(d^4-{r_1}^4-2\,d^2\,{r_2}^2+{r_2}^4\right)\sin(\phi_T+\gamma)}{4\,d^2\,{r_1}^3\sqrt{1-\frac{(d^2+{r_1}^2-{r_2}^2)^2}{4\,d^2\,{r_1}^2}}} + \frac{\cos(\phi_T+\gamma)}{d}\right)}{2} +$$

$$\frac{\sqrt{1-\frac{(d^2+r_1{}^2-r_2{}^2)^2}{4\,d^2\,r_1{}^2}}\,\sin(\phi_T+\gamma)}{2},$$

$$\frac{\partial y_c}{\partial r_2} = \frac{r_1\left(\frac{r_2\left(d^2+r_1{}^2-r_2{}^2\right)\sin(\phi_T+\gamma)}{2\,d^2\,r_1{}^2\sqrt{1-\frac{(d^2+r_1{}^2-r_2{}^2)^2}{4\,d^2\,r_1{}^2}}} - \frac{r_2\cos(\phi_T+\gamma)}{d\,r_1}\right)}{2}$$

- Extrema calculation:
  - Finding extrema on the borders:
    - Solving $\frac{\partial\xi}{\partial r_1}=0$

      $$r_1 = \pm\sqrt{d^2-r_2{}^2}$$

    - Solving for $\frac{\partial\xi}{\partial r_2}=0$

      $$r_2 = 0$$

    - Solving $\frac{\partial x_c}{\partial r_1}=0$

      $$r_1 = \pm\sqrt{d^2+r_2{}^2\pm 2\sqrt{d^2\,r_2{}^2\sin(\phi_T+\gamma)^2}}$$

    - Solving $\frac{\partial x_c}{\partial r_2}=0$

      $$\begin{aligned} r_2 &= 0\,, \\ r_2 &= \pm\sqrt{d^2+r_1{}^2\pm 2\sqrt{d^2\,r_1{}^2\sin(\phi_T+\gamma)^2}} \end{aligned}$$

    - Solving $\frac{\partial y_c}{\partial r_1}=0$

      $$r_1 = \pm\sqrt{d^2+r_2{}^2\pm 2\sqrt{d^2\,r_2{}^2\cos(\phi_T+\gamma)^2}}$$

    - Solving $\frac{\partial y_c}{\partial r_2}=0$

      $$\begin{aligned} r_2 &= 0\,, \\ r_2 &= \pm\sqrt{d^2+r_1{}^2\pm 2\sqrt{d^2\,r_1{}^2\cos(\phi_T+\gamma)^2}} \end{aligned}$$

  - Find interior extrema:
    - Solving $\frac{\partial\xi}{\partial r_1}=0$ and $\frac{\partial\xi}{\partial r_2}=0$;
    - Solving $\frac{\partial x_c}{\partial r_1}=0$ and $\frac{\partial x_c}{\partial r_2}=0$;
    - Solving $\frac{\partial y_c}{\partial r_1}=0$ and $\frac{\partial y_c}{\partial r_2}=0$.

    The functions ξ, x_c, and y_c are strictly monotonic with respect to r_1 and r_2, thus presenting no solution.

- Measurements' constraints:

  Derived from the previous equations, the constraints imposed are:

$$d,\ r_1 \neq 0\ , \quad |r_2-r_1| < d$$

A.1.3 Line Feature

- The feature parameter functions are:

$$\begin{aligned} \theta &= \phi_T + \xi \ , \\ r &= x_l cos(\theta) + y_l sin(\theta) \end{aligned}$$

The parameters functions x_l, y_l, and ξ are the same as in the case of a corner feature (Equations A.1, A.2, and A.3).

- The corresponding partial derivatives are:

The partial derivatives of x_l, y_l, and ξ are similar to the ones presented on the corner feature (Section A.1.2), and the partial derivatives of θ are also similar to the ones for ξ, because the θ and ξ functions only differ by a constant ϕ_T.

$$\begin{aligned} \frac{\partial r}{\partial r_1} &= \frac{1}{2} + \left(\frac{d^2 - {r_1}^2 - {r_2}^2}{2\, d\, {r_1}^2} \right) (x_T \sin(\phi_T + \gamma) - y_T \cos(\phi_T + \gamma)) + \\ & \frac{(d^2 - {r_1}^2 - {r_2}^2)\, (d^2 + {r_1}^2 - {r_2}^2)\, (x_T \cos(\phi_T + \gamma) + y_T \sin(\phi_T + \gamma))}{4\, d^2\, {r_1}^3 \sqrt{1 - \frac{(d^2 + {r_1}^2 - {r_2}^2)^2}{4\, d^2\, {r_1}^2}}} , \\ \frac{\partial r}{\partial r_2} &= \frac{r_2\, (d^2 + {r_1}^2 - {r_2}^2)\, (x_T \cos(\phi_T + \gamma) + y_T \sin(\phi_T + \gamma))}{2\, d^2\, {r_1}^2 \sqrt{1 - \frac{(d^2 + {r_1}^2 - {r_2}^2)^2}{4\, d^2\, {r_1}^2}}} + \\ & \frac{r_2\, (x_T \sin(\phi_T + \gamma) - y_T \cos(\phi_T + \gamma))}{d\, r_1} \end{aligned}$$

- Extrema calculation:
  - Finding extrema on the borders:
    - * Solving $\frac{\partial r}{\partial r_1} = 0$:

In this case, it is equivalent to solve the polynomial:

$$a_1\, {r_1}^8 + a_2\, {r_1}^6 + a_3\, {r_1}^4 + a_4\, {r_1}^2 + a_5 = 0 \ ,$$

where the coefficients are:

$$\begin{aligned} a_1 &= -2\, d\, (x_T \sin(\phi_T + \gamma) - y_T \cos(\phi_T + \gamma)) + d^2 + {x_T}^2 + {y_T}^2 \ , \\ a_2 &= -4\, d^2 \left(x_T^2 \sin(\phi_T + \gamma)^2 + y_T^2 \cos(\phi_T + \gamma)^2 \right) + \\ & 8\, d^2 x_T\, y_T \sin(\phi_T + \gamma) \cos(\phi_T + \gamma) - 2\, d^2 \left(d^2 + r_2^2 \right) + \\ & 2\, d \left(3\, d^2 + r_2^2 \right) (x_T \sin(\phi_T + \gamma) - y_T \cos(\phi_T + \gamma)) \ , \\ a_3 &= 2 \left(d^2 - r_2^2 \right) \left(4\, d^2 \left(x_T^2 \sin(\phi_T + \gamma)^2 + y_T^2 \cos(\phi_T + \gamma)^2 \right) - \right. \\ & 8\, d^2 x_T\, y_T \sin(\phi_T + \gamma) \cos(\phi_T + \gamma) - \left(d^2 - r_2^2 \right) \left(x_T^2 + y_T^2 - \frac{d^2}{2} \right) - \end{aligned}$$

$$
\begin{aligned}
& d\left(3\,d^2+r_2^2\right)\left(x_T\,\sin(\phi_T+\gamma)-y_T\,\cos(\phi_T+\gamma)\right)\Big)\ , \\
a_4 \;=\; & (d-r_2)^2\,(d+r_2)^2\,\Big(-4\,d^2\left(x_T^2\sin(\phi_T+\gamma)^2+y_T^2\cos(\phi_T+\gamma)^2\right)+ \\
& 8\,d^2x_T\,y_T\,\sin(\phi_T+\gamma)\,\cos(\phi_T+\gamma)+ \\
& 2\,d\left(d^2-r_2^2\right)\left(x_T\,\sin(\phi_T+\gamma)-y_T\,\cos(\phi_T+\gamma)\right)\Big)\ , \\
a_5 \;=\; & (d-r_2)^4\,(d+r_2)^4\,\left(x_T^2+y_T^2\right)
\end{aligned}
$$

The solution is:

$$
\begin{aligned}
r_1 \;=\; & \pm\Big(-9\,a_2\,\sqrt[6]{aux_1}\sqrt[4]{aux_2}+{aux_2}^{3/4}\sqrt{3}\pm\Big(27\,{a_2}^2\sqrt[3]{aux_1}\sqrt{aux_2}- \\
& 72\,a_3\,a_1\,\sqrt[3]{aux_1}\sqrt{aux_2}-\sqrt{aux_2}\,54^{2/3}{aux_1}^{2/3}a_1+ \\
& 18\,\sqrt{aux_2}\,a_1\sqrt[3]{54}\,a_2\,a_4-72\,\sqrt{aux_2}\,{a_1}^2\sqrt[3]{54}\,a_5- \\
& 6\,\sqrt{aux_2}\,a_1\,\sqrt[3]{54}\,{a_3}^2+324\,\sqrt{3}\sqrt{aux_1}\,a_2\,a_1\,a_3- \\
& 648\,\sqrt{3}\sqrt{aux_1}\,{a_1}^2a_4\pm 81\,\sqrt{3}\sqrt{aux_1}{a_2}^3\Big)^{1/2}\sqrt{6}\Big)^{1/2}\Big/ \\
& \left(6\,\sqrt{a_1}\,{aux_1}^{1/12}\sqrt[8]{aux_2}\right)\ ,
\end{aligned}
$$

where the auxiliary variables are:

$$
\begin{aligned}
aux_1 \;=\; & -9\,a_3\,a_2\,a_4-72\,a_5\,a_3\,a_1+27\,{a_4}^2a_1+27\,a_5\,{a_2}^2+2\,{a_3}^3+ \\
& 3\,\sqrt{3}\,\Big(27\,{a_4}^4{a_1}^2+6\,{a_2}^2{a_4}^2a_5\,a_1+192\,a_2\,a_4\,{a_5}^2{a_1}^2+ \\
& 80\,a_2\,a_4\,a_5\,a_1\,{a_3}^2-18\,a_3\,a_2\,{a_4}^3a_1-18\,a_3\,{a_2}^3a_4\,a_5- \\
& 144\,a_5\,a_3\,{a_1}^2{a_4}^2-144\,{a_5}^2a_3\,a_1\,{a_2}^2+4\,{a_2}^3{a_4}^3-256\,{a_5}^3{a_1}^3+ \\
& 27\,{a_5}^2{a_2}^4-{a_2}^2{a_4}^2{a_3}^2+128\,{a_5}^2{a_1}^2{a_3}^2-16\,a_5\,a_1\,{a_3}^4+ \\
& 4\,{a_4}^2a_1\,{a_3}^3+4\,a_5\,{a_2}^2{a_3}^3\Big)^{1/2}\ , \\
aux_2 \;=\; & 27\,{a_2}^2\sqrt[3]{aux_1}-72\,a_3\,a_1\,\sqrt[3]{aux_1}+2\,54^{2/3}{aux_1}^{2/3}a_1- \\
& 36\,a_1\,\sqrt[3]{54}a_2\,a_4+144\,{a_1}^2\sqrt[3]{54}a_5+12\,a_1\,\sqrt[3]{54}{a_3}^2
\end{aligned}
$$

- Solving for $\frac{\partial r}{\partial r_2}=0$

$$
\begin{aligned}
r_2 \;=\; & 0\ , \\
r_2 \;=\; & \pm\Big(\left(d^2+{r_1}^2\right)\left({x_T}^2+{y_T}^2\right)\pm 2\,\Big(d^2\,{r_1}^2\,\left({x_T}^2+{y_T}^2\right) \\
& \left(x_T\,\sin(\phi_T+\gamma)-y_T\,\cos(\phi_T+\gamma)\right)^2\Big)^{1/2}\Big)^{1/2}\Big/\left(\sqrt{{x_T}^2+{y_T}^2}\right)
\end{aligned}
$$

- Find interior extrema:
  - Solving $\frac{\partial \xi}{\partial r_1}=0$ and $\frac{\partial \xi}{\partial r_2}=0$;
  - Solving $\frac{\partial x_l}{\partial r_1}=0$ and $\frac{\partial x_l}{\partial r_2}=0$;
  - Solving $\frac{\partial y_l}{\partial r_1}=0$ and $\frac{\partial y_l}{\partial r_2}=0$;
  - Solving $\frac{\partial \theta}{\partial r_1}=0$ and $\frac{\partial \theta}{\partial r_2}=0$;

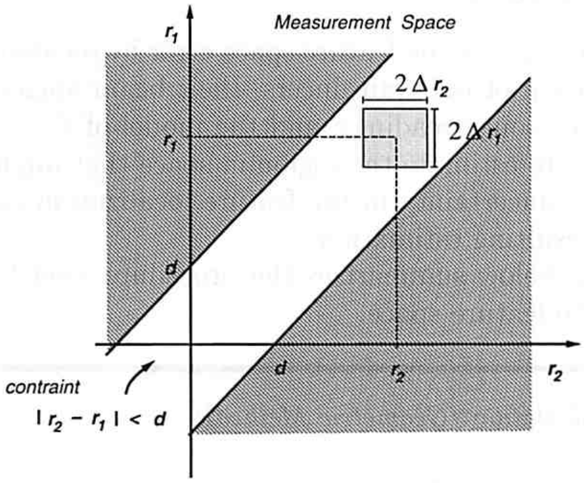


Figure A.1. Constraint on the Sonar Measurements

- $*$ Solving $\frac{\partial r}{\partial r_1} = 0$ and $\frac{\partial r}{\partial r_2} = 0$.

  The functions ξ, x_l, y_l, and θ are strictly monotonic with respect to r_1 and r_2. The function r is also strictly monotonic with respect to r_1 and r_2, as determined by a geometrical analysis presented on Section A.2.3.

- Measurements' constraints:

  Derived from the previous equations, the constraints imposed are:

$$x_T,\ y_T \neq 0\ , \qquad d,\ r_1 \neq 0\ , \qquad |r_2 - r_1| < d$$

A.1.4 Characteristics of the Analytic Approach

The three main characteristics of this method are: its simplicity (the extrema for the error functions have closed-form solutions); the use of the maximum absolute error, making it a conservative method; and the ease of detecting measurements that violate model constraints. To illustrate the last characteristic, Figure A.1 depicts the limitation imposed by the constraint $|r_2 - r_1| < d$. As shown, the distance between transducers (d) and the error in the measurements $(\Delta r_1, \Delta r_2)$ are related limiting factors. The larger the distance between transducers, the more variance in the measurements can be supported, thus, causing measurements from more distant features not to be discarded.

A.2 Geometric Method

This method computes the feature space error by means of geometric analysis. It uses the configuration of the transducers, their beam angles, the sonar readings, the uncertainty on the sonar readings, and the model of the feature under analysis (line, edge, or corner) to estimate the region in space that might contain the feature correct position. The uncertainty in the feature localization can then be calculated indirectly from the resulting estimation.

The algorithm below summarizes the procedure used to transform errors in measurement space to feature space.

Feature Error Calculation (Geometric Method):

Compute the maximum error of a feature parameter $u : u \in \{r, \theta, x_e, y_e, x_c, y_c\}$ by:

1. *Calculating the region where the sonar reflections could occur, using the sonar measurements range ($r_1 \pm \Delta r_1$, $r_2 \pm \Delta r_2$), the transducers configuration, their beam angles, and the type of feature under analysis.*
2. *Returning the maximum absolute error in the feature parameter (Δu) given the above region, if such region exist; otherwise, ignore this measurement.*

The region where the sonar reflection could occur, given a measurement pair $(r_1 \pm \Delta r_1, r_2 \pm \Delta r_2)$, changes depending on the type of feature under analysis. The geometrical analysis used on the calculation of this region is described on the following sections.

A.2.1 Edge Feature

In the case of an edge feature, as shown in Figure A.2, both r_1 and r_2 reflections occur exactly on the edge. In this situation, the error region is limited by the intersection of the transmitter cone with the receiver cone, and by the error region imposed by the measurements (r_1, r_2). Considering the measurement r_1, the error region is delimited by two circular arcs generated by

$$x^2 + y^2 = r^2 \quad , \quad r = \frac{1}{2}\left(r_1 \pm \Delta r_1\right) \ .$$

In the case of the measurement r_2, the error region is bounded by two elliptical arcs given by:

$$\frac{x^2}{a} + \frac{y^2}{b} = 1 \quad , \quad a = \frac{1}{2}\left(r_2 \pm \Delta r_2\right) \quad , \quad b = \frac{1}{2}\sqrt{\left(r_2 \pm \Delta r_2\right)^2 - d^2} \ ,$$

where d is the distance between transducers.

The error in feature space $(\Delta x_e, \Delta y_e)$ is obtained by computing the maximum absolute error in x and y directions, respectively. For an edge feature, the possible extrema are on the intersection between circles, ellipses, and sonar cone boundaries, or on the external circle or ellipse, as shown on the left diagram of Figure A.2.

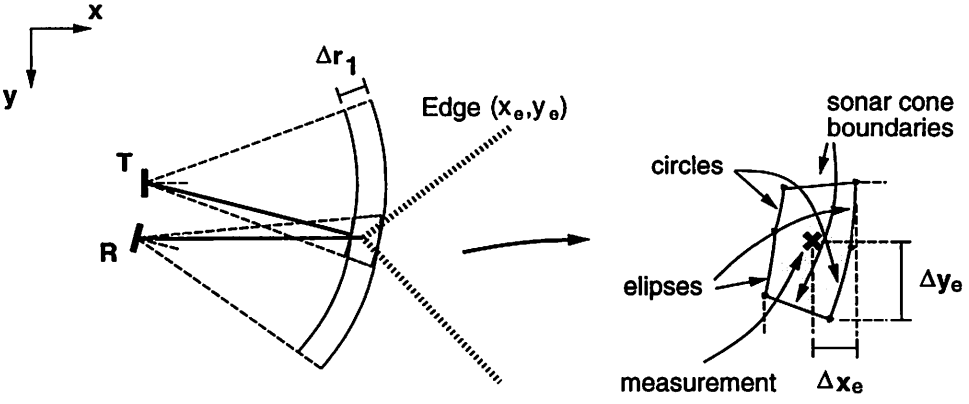


Figure A.2. Edge error

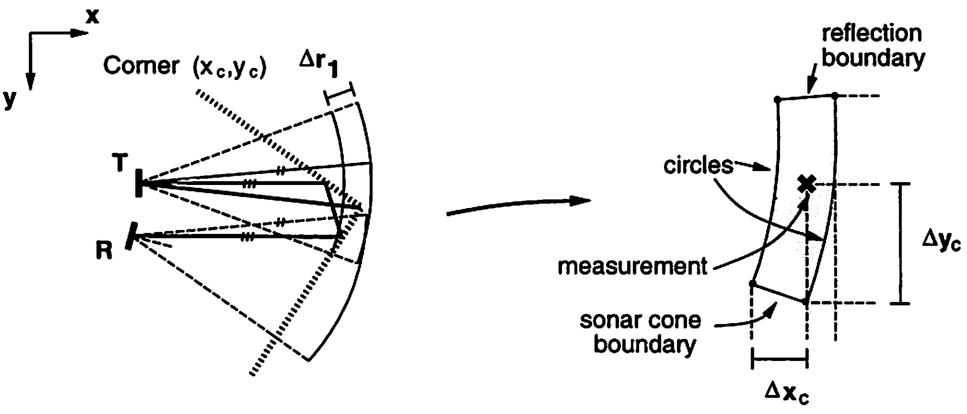


Figure A.3. Corner error

A.2.2 Corner Feature

As shown in Figure A.3, only the r_1 reflection occurs on the corner. Therefore, the x and y position of the corner is contained in the transmitter cone, on the region limited by the uncertainty in the measurement r_1 (Δr_1). Moreover, because the corner feature is a right-angle corner, the r_2 reflection has parallel segments, and the above region can be further limited by the receiver's cone boundaries.

Similarly to the edge feature, the corner error $(\Delta x_c, \Delta y_c)$ is the maximum absolute error in the x and y directions, respectively. In this case, the possible extrema are on the intersection between the circles, the transmitter boundaries, and the reflection boundary, or on the external circle, as depicted on Figure A.3.

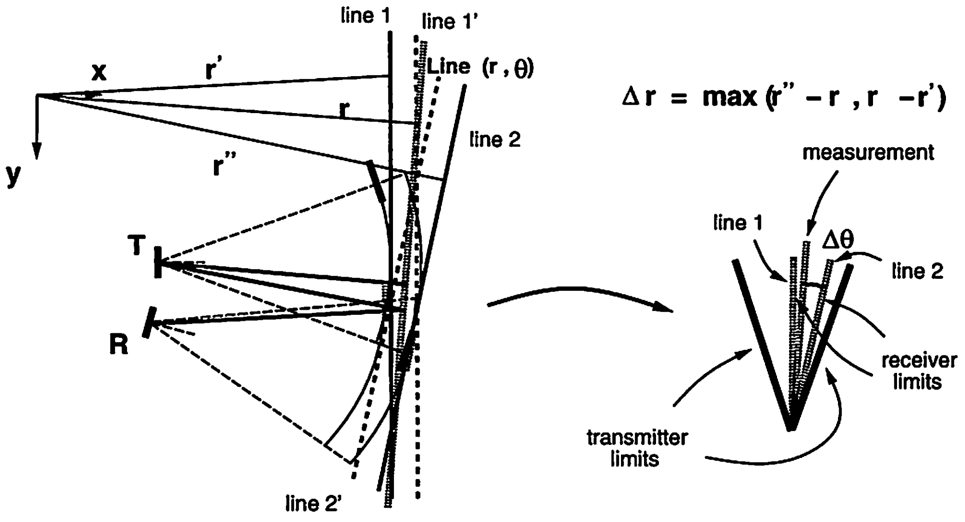


Figure A.4. Line error

A.2.3 Line Feature

On the line feature, the r_1 reflection occurs on the transmitter cone and the r_2 reflection occurs on the intersection of the transmitter and receiver cones. Both reflections are limited by the error region imposed by the measurements (r_1, r_2), as described before is Section A.2.1. The error region is delimited by the transmitter and receiver limit lines, as presented in Figure A.4, corresponding to possible r_1 and r_2 reflections, respectively.

As before, the line error $(\Delta r, \Delta\theta)$ is calculated based on the maximum absolute error in r and θ, respectively. Given that a possible line has to be tangent to the circles produced by the r_1 reflection, and also tangent to the ellipses produced by the r_2 reflection, both errors $(\Delta r, \Delta\theta)$ can be obtained from the limit lines, as demonstrated by the diagram on the left of Figure A.4 where the possible extreme lines are *line* 1, *line* 2, *line* 1′, and *line* 2′. Notice that depending on the position of the referential, a combination of lines different than the pair $(line\ 1, line\ 2)$ will generate the maximum Δr, but limit lines have always r_1 or r_2 at an extreme value.

A.2.4 Error Minimization

In measurement space, the error associated with the measurements r_1 and r_2 is directly proportional to the measurements' value,

$$\Delta r_1 = pr_1 \ , \quad \Delta r_2 = pr_2 \ , \quad p \leq 0.01$$

Therefore, to obtain a more precise measurement, the sensors should be closer to the object being measured.

As shown in the previous sections, in feature space, not only the sensor distance to a feature but also the configuration of the sensors, and the feature type play an important role on feature error minimization and on feature characterization. Some techniques used in radar systems that exploits the configuration of the antennas to improve measurement quality can also be applied here to the sonar system.

In the case of an edge or line feature, a more precise measurement is obtained when the overlap between the receiver and transmitter cones is minimized by rotating the transducers, or by increasing their distance (d). This is also true for the corner feature case, except that the receiver and transmitter cones do not necessarily need to overlap, because the error region is produced by the reflection boundary, and not by the receiver cone boundary. All these facts can be used on the design of active sonar sensor controllers.

A.3 Comparison

Both approaches are conservative in their error estimate, since they use the maximum absolute error. The analytic method uses the maximum absolute error derived from the extrema values of the non-linear feature localization formulas, and its application is limited to the constraints imposed to the measurements. The geometric method, in the other hand, uses the sonar configuration and the sonar beam angle estimate to calculate the maximum absolute error, thus being as conservative and accurate as the sonar beam angle selected. The main advantage of the geometric over the analytic approach is the direct association between sensor configuration and precision of the measurement, allowing for the possibility of inverting the problem and selecting the sonar configuration to obtain a more precise measurement.

APPENDIX B

STATE ESTIMATION - KALMAN FILTER

A Kalman filter is defined by Maybeck in [17] as an *optimal recursive data processing algorithm.* It produces an optimal estimate of the state variables of the system by recursively combining all available measurement data. Its applicability is constrained to situations where the dynamics of the system and measurement devices, together with the statistical description of the noises and uncertainties associated with them, and the initial state of the system are known.

The Kalman filter's main importance as a data processing algorithm relies on its optimality and recursive characteristic. A Kalman filter is optimal not only because it uses all the information available, even the least precise measurements, but because it produces the unique best estimate of the state variables when the system dynamics can be described by a linear model, and the system and measurement noises are white and Gaussian. The recursive characteristic of the filter has practical importance, because it does not require all previous data to be saved and reprocessed at each new measurement.

In practice, the Kalman filter can be used even when some of the constraints presented above are violated. Extra filtering can be used to change a system with time or frequency correlated noise into a linear system with white noise. Gaussian noise is a physically plausible and practical assumption. And, in case of lack of statistical information about the noise process, there is no better option than to assume Gaussian noise [17]. The Kalman filter can be also compensated to overcome inadequacies on the system and measurement devices models, and extended to address systems that are better described by nonlinear models [18, 1].

The nomenclature used in this paper to describe both the Kalman filter (linear system) and the Extended Kalman filter (nonlinear system) follows the one used by Bar-Shalom in [1]. Figure B.1 presents the nomenclature used on a Kalman filter state estimation cycle, where the system, the control, and the discrete-time Kalman filter estimation algorithm (shaded area) are depicted. For a formal mathematical description of Kalman filters see [17, 1, 10, 18], and, in special, as an introductory reading refer to Maybeck [17].

In this paper, the Kalman filter was used to estimate the position of a 2D feature (line, edge, or corner) in a static environment, using the discrete-time measurements obtained from two sonar sensors, directly (nonlinear case), or indirectly (linear case), as described in Chapter 2. In the following filter implementations, the system has no dynamics and no controller, because the features are static and their positions are computed in global coordinates, reducing the Kalman filter algorithm to a recursive form of a least-square estimator [1].

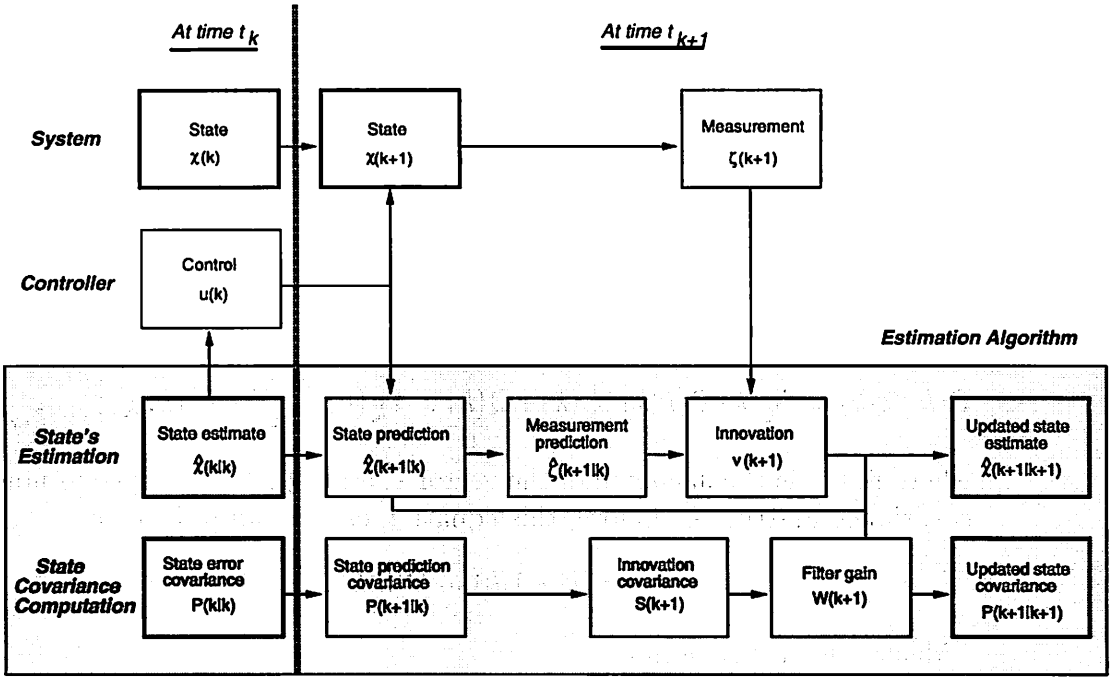


Figure B.1. Kalman filter - state estimation cycle

B.1 Extended Kalman Filter

The Extended Kalman filter (EKF) is used when the system cannot be adequately described by a linear model. The idea behind the method is to better follow the system's reference state trajectory, allowing for the use of linear perturbation techniques; therefore, the EKF linearizes about each new state estimate to produce a new and better state trajectory for the estimation process [18, 1].

The following are the general equations of the first-order Extended Kalman filter algorithm, and the simplified EKF version implemented:

- System:

$$\chi(k+1) = \mathbf{f}[k, \chi(k)] + \mathbf{g}[k, \mathbf{u}(k)] + \mathbf{v}(k)$$

  Taking into account that the process under study is static ($\mathbf{f}[\cdot] = \chi(k)$), does not have a controller ($\mathbf{g}[\cdot]$), and consequently does not present process noise ($\mathbf{v}(k)$), the equation above becomes:

$$\chi(k+1) = \chi(k)$$

- Measurement:

$$\zeta(k+1) = \mathbf{h}[k+1, \chi(k+1)] + \mathbf{w}(k+1)$$

  The nonlinear function **h** transforms state space variables of a feature (line, edge, or corner) to measurement space variables (sonar readings, (r_1, r_2)). The measurement noise **w** represents an additive, zero-mean, and white noise.

- State estimation and covariance computation:
  - State prediction:

$$\hat{\chi}(k+1|k) = \mathbf{f}[k, \hat{\chi}(k|k)] + \mathbf{g}[k, \mathbf{u}(k)]$$

  Similarly to the system's state equation above, ($\mathbf{f}[\cdot] = \hat{\chi}(k|k)$), and ($\mathbf{g}[\cdot] = 0$), resulting in:

$$\hat{\chi}(k+1|k) = \hat{\chi}(k|k)$$

  - State prediction covariance:

$$\mathbf{P}(k+1|k) = \mathbf{f}_{\chi}(k)\,\mathbf{P}(k|k)\,\mathbf{f}_{\chi}^{T}(k) + \mathbf{Q}(k)\,, \quad \mathbf{f}_{\chi}(k) \triangleq \left[\nabla_{\chi}\mathbf{f}^{T}(k,\chi)\right]^{T}_{\chi=\hat{\chi}(k|k)}$$

  where $\mathbf{f}_{\chi}(k)$ is the Jacobian of the vector $\mathbf{f}$, and $\mathbf{Q}(k)$ is the system noise covariance matrix. As before, this equation can be simplified to:

$$\mathbf{P}(k+1|k) = \mathbf{P}(k|k)$$

  - Measurement prediction:

$$\hat{\zeta}(k+1|k) = \mathbf{h}[k+1, \hat{\chi}(k+1|k)]$$

  - Innovation:

$$\nu(k+1) = \zeta(k+1) - \hat{\zeta}(k+1|k)$$

  - Innovation covariance:

$$\mathbf{S}(k+1) = \mathbf{h}_{\chi}(k+1)\,\mathbf{P}(k+1|k)\,\mathbf{h}_{\chi}^{T}(k+1) + \mathbf{R}(k+1)\quad ,$$

$$\mathbf{h}_{\chi}(k+1) = \left[\nabla_{\chi}\mathbf{h}^{T}(k+1,\chi)\right]^{T}_{\chi=\hat{\chi}(k+1|k)}$$

  where $\mathbf{h}_{\chi}(k+1)$ is the Jacobian of the vector $\mathbf{h}$, and $\mathbf{R}(k)$ is the covariance matrix of the noise associated with the measurements.
  - Filter gain:

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\,\mathbf{h}_{\chi}^{T}(k+1)\,\mathbf{S}^{-1}(k+1)$$

  - Updated state estimate:

$$\hat{\chi}(k+1|k+1) = \hat{\chi}(k+1|k) + \mathbf{W}(k+1)\,\nu(k+1)$$

  - Updated state covariance:

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\,\mathbf{S}(k+1)\,\mathbf{W}^{T}(k+1)$$

Each feature type uses a different set of nonlinear equations and state variables. In the following sections, the derivation of each feature EKF function is reported.

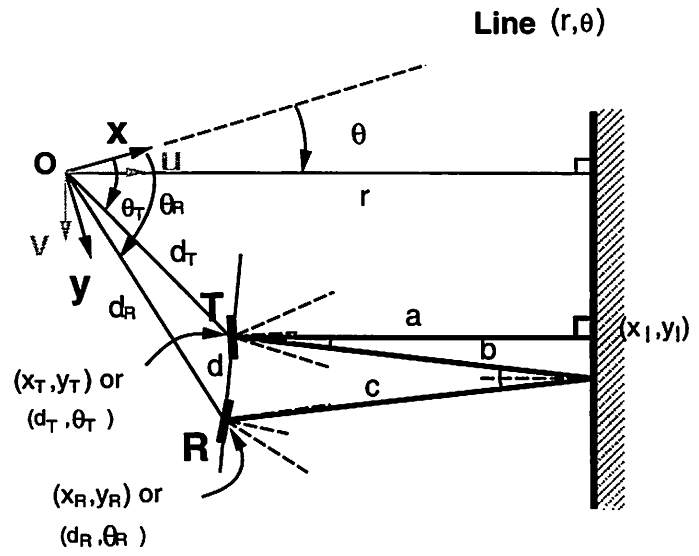


Figure B.2. Line feature

B.1.1 Line Feature Localization Filter

In the case that the feature is a line, the process state variables and measurement variables are:

$$\chi(k) = \begin{bmatrix} r(k) \\ \theta(k) \end{bmatrix} \quad , \qquad \zeta(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$

where (r_1, r_2) are the sonar TOF measurements, and (r, θ) are the line position in global Cartesian coordinate with origin at O, as shown in Figure B.2. Furthermore, the nonlinear functions $\mathbf{h}$ and its Jacobian $\mathbf{h}_\chi$ are:

$$\mathbf{h}(k, \chi(k)) = \begin{bmatrix} h_{r_1}(k, \chi(k)) \\ h_{r_2}(k, \chi(k)) \end{bmatrix} \; , \quad \mathbf{h}_\chi(k{+}1) = \begin{bmatrix} \frac{\partial h_{r_1}(k+1|\chi)}{\partial r} & \frac{\partial h_{r_1}(k+1|\chi)}{\partial \theta} \\ \frac{\partial h_{r_2}(k+1|\chi)}{\partial r} & \frac{\partial h_{r_2}(k+1|\chi)}{\partial \theta} \end{bmatrix}_{\chi = \hat{\chi}(k+1|k)}$$

The following nonlinear equations were derived using the line feature localization model in Section 2.1.1, and simple trigonometric relations depicted in Figure B.3. The reference to time (k) was omitted in the following equations for the sake of clarity.

$$\begin{aligned} h_{r_1} &= 2\,a &= \overline{TT'} &= 2\,(r - u_T)\ , \\ h_{r_2} &= b + c &= \overline{RT'} &= \sqrt{4\,(r - u_R)\,(r - u_T) + d^2} \end{aligned}$$

where u_T, v_T, u_R, and v_R correspond to the sonars T and R localization, respectively, in $u \times v$ coordinate.

$$u_T = d_T\,cos(\theta - \theta_T)\ , \qquad v_T = -\,d_T\,sin(\theta - \theta_T)$$

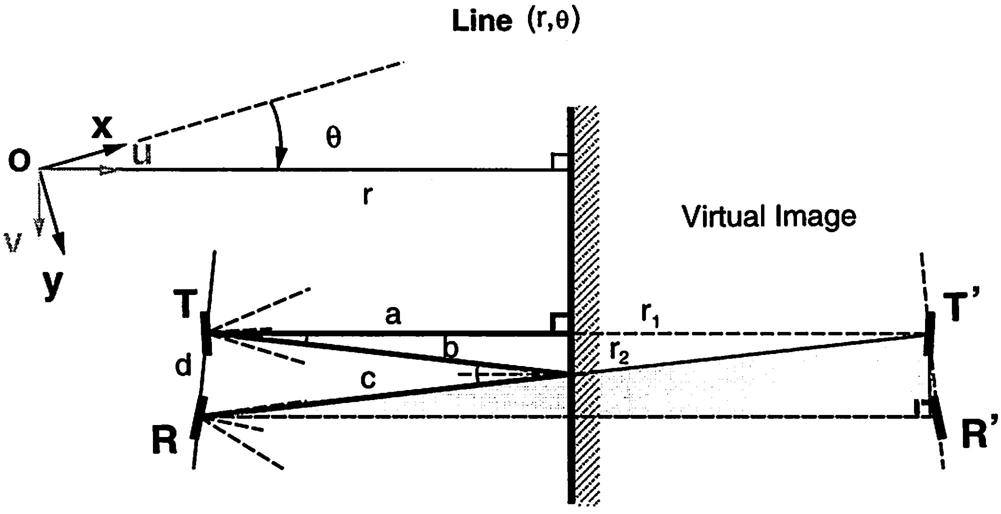


Figure B.3. Line feature derivation

$$u_R = d_R\, cos(\theta - \theta_R) \;, \qquad v_R = -\, d_R\, sin(\theta - \theta_R)$$

$$\frac{\partial h_{r_1}}{\partial r} = 2 \;, \qquad \frac{\partial h_{r_2}}{\partial r} = \frac{(2\, r - u_R - u_T)}{\sqrt{4\; (r - u_R)\; (r - u_T) + d^2}}$$

$$\frac{\partial h_{r_1}}{\partial \theta} = -2\, v_T \;, \qquad \frac{\partial h_{r_2}}{\partial \theta} = \frac{(-r\, v_R - r\, v_T + v_R\, u_T + v_T\, u_R)}{\sqrt{4\; (r - u_R)\; (r - u_T) + d^2}}$$

B.1.2 Edge Feature Localization Filter

For the edge feature, the process state variables and measurement variables are:

$$\chi(k) = \begin{bmatrix} x_e(k) \\ y_e(k) \end{bmatrix} \;, \qquad \zeta(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$

where (r_1, r_2) are the sonar TOF measurements, and (x_e, y_e) is the edge Cartesian position in global coordinate with origin at **O**, as shown in Figure B.4. Furthermore, the nonlinear functions **h** and its Jacobian $\mathbf{h}_\chi$ are:

$$\mathbf{h}(k, \chi(k)) = \begin{bmatrix} h_{r_1}(k, \chi(k)) \\ h_{r_2}(k, \chi(k)) \end{bmatrix} \;, \quad \mathbf{h}_\chi(k{+}1) = \begin{bmatrix} \frac{\partial h_{r_1}(k+1|\chi)}{\partial x_e} & \frac{\partial h_{r_1}(k+1|\chi)}{\partial y_e} \\ \frac{\partial h_{r_2}(k+1|\chi)}{\partial x_e} & \frac{\partial h_{r_2}(k+1|\chi)}{\partial y_e} \end{bmatrix}_{\chi = \hat{\chi}(k+1|k)}$$

The following nonlinear equations were derived using the edge feature localization model in Section 2.1.3, and simple trigonometric relations depicted in Figure

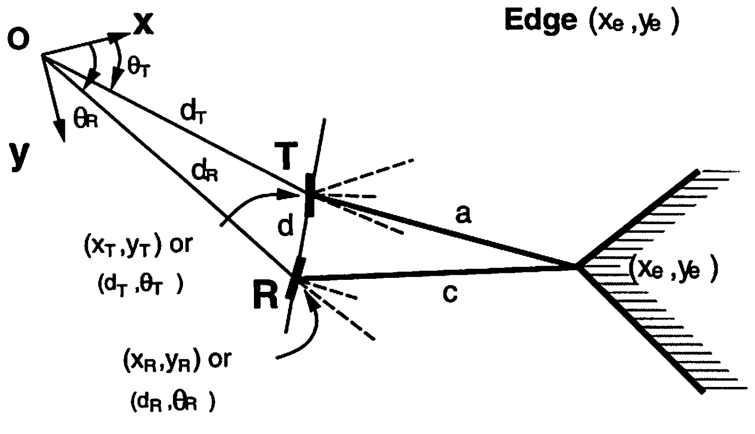


Figure B.4. Edge feature

B.4. The reference to time (k) was again omitted in the following equations for the sake of clarity.

$$\begin{aligned} h_{r_1} &= 2a &&= 2\sqrt{(x_e - x_T)^2 + (y_e - y_T)^2} \\ h_{r_2} &= a + c &&= \sqrt{(x_e - x_R)^2 + (y_e - y_R)^2} + \sqrt{(x_e - x_T)^2 + (y_e - y_T)^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial h_{r_1}}{\partial x_e} &= \frac{2\ (x_e - x_T)}{\sqrt{(x_e - x_T)^2 + (y_e - y_T)^2}} \\ \frac{\partial h_{r_2}}{\partial x_e} &= \frac{x_e - x_R}{\sqrt{(x_e - x_R)^2 + (y_e - y_R)^2}} + \frac{x_e - x_T}{\sqrt{(x_e - x_T)^2 + (y_e - y_T)^2}} \\ \frac{\partial h_{r_1}}{\partial y_e} &= \frac{2\ (y_e - y_T)}{\sqrt{(x_e - x_T)^2 + (y_e - y_T)^2}} \\ \frac{\partial h_{r_2}}{\partial y_e} &= \frac{y_e - y_R}{\sqrt{(x_e - x_R)^2 + (y_e - y_R)^2}} + \frac{y_e - y_T}{\sqrt{(x_e - x_T)^2 + (y_e - y_T)^2}} \end{aligned}$$

B.1.3 Corner Feature Localization Filter

In the case that the feature is a corner, the process state variables and measurement variables are:

$$\chi(k) = \begin{bmatrix} x_c(k) \\ y_c(k) \end{bmatrix} , \qquad \zeta(k) = \begin{bmatrix} r_1(k) \\ r_2(k) \end{bmatrix}$$

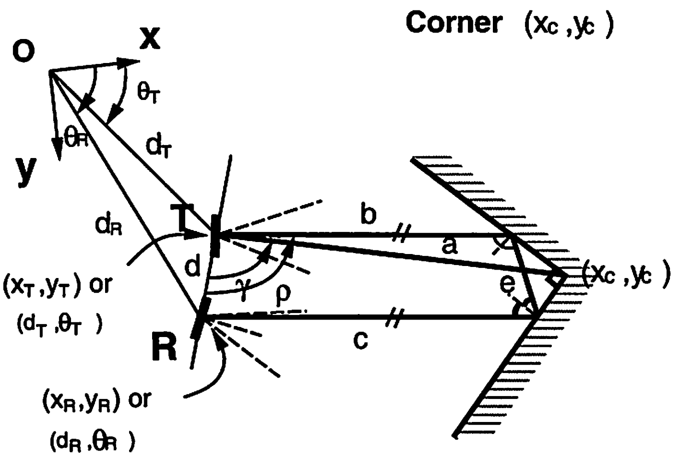


Figure B.5. Corner feature

where (r_1, r_2) are the sonar TOF measurements, and (x_c, y_c) is the corner Cartesian position in global coordinate with origin at **O**, as shown in Figure B.5. Moreover, the nonlinear functions **h** and its Jacobian $\mathbf{h}_\chi$ are:

$$\mathbf{h}(k,\chi(k)) = \begin{bmatrix} h_{r_1}(k,\chi(k)) \\ h_{r_2}(k,\chi(k)) \end{bmatrix}, \quad \mathbf{h}_\chi(k{+}1) = \begin{bmatrix} \frac{\partial h_{r_1}(k+1|\chi)}{\partial x_c} & \frac{\partial h_{r_1}(k+1|\chi)}{\partial y_c} \\ \frac{\partial h_{r_2}(k+1|\chi)}{\partial x_c} & \frac{\partial h_{r_2}(k+1|\chi)}{\partial y_c} \end{bmatrix}_{\chi=\hat{\chi}(k+1|k)}$$

The following nonlinear equations were derived using the corner feature localization model in Section 2.1.2, and simple trigonometric relations depicted in Figure B.6. In particular, h_{r_2} was derived using the sum of the equations of the two triangles shown, where $r_3 = 2f = \overline{RR'}$. The reference to time (k) was again omitted.

$$\begin{aligned} h_{r_1} &= 2a = \overline{TT'} = 2\sqrt{(x_c - x_T)^2 + (y_c - y_T)^2}\,, \\ h_{r_2} &= b + c + e = \overline{TR'} \\ &= \sqrt{2\left((x_c - x_R)^2 + (y_c - y_R)^2\right) + 2\left((x_c - x_T)^2 + (y_c - y_T)^2\right) - d^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial h_{r_1}}{\partial x_c} &= \frac{2\,(x - x_T)}{\sqrt{(x - x_T)^2 + (y - y_T)^2}} \\ \frac{\partial h_{r_2}}{\partial x_c} &= \frac{2\,(x - x_R) + 2\,(x - x_T)}{\sqrt{2\,\left((x - x_R)^2 + (y - y_R)^2\right) + 2\,\left((x - x_T)^2 + (y - y_T)^2\right) - d^2}} \\ \frac{\partial h_{r_1}}{\partial y_c} &= \frac{2\,(y - y_T)}{\sqrt{(x - x_T)^2 + (y - y_T)^2}} \end{aligned}$$

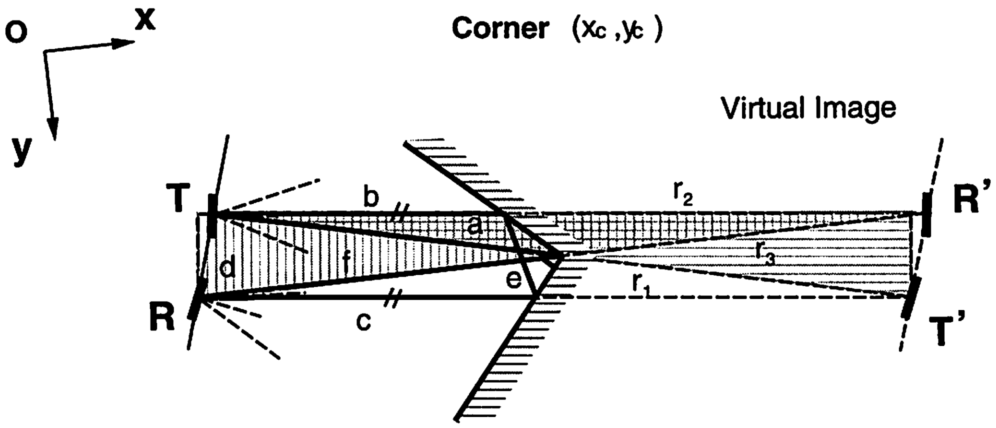


Figure B.6. Corner feature derivation

$$\frac{\partial h_{r_2}}{\partial y_c} = \frac{2\,(y-y_R)+2\,(y-y_T)}{\sqrt{2\left((x-x_R)^2+(y-y_R)^2\right)+2\left((x-x_T)^2+(y-y_T)^2\right)-d^2}}$$

B.2 Kalman Filter

A linear Kalman filter was also tested. The idea employed was to compute the feature position and error associated with each pair of measurements, as described in Chapter 2 and Appendix A, simplifying the data fusion by doing it directly in feature space (linear data fusion).

The following are the equations of the Kalman filter algorithm (linear case):

- System:

  $$\chi(k+1) = \mathbf{F}(k)\,\chi(k) + \mathbf{G}(k)\,\mathbf{u}(k) + \mathbf{v}(k)$$

  By the same reasons presented before (static state space with no control), **F** becomes the identity matrix **I**, and the two last terms are dropped. Thus, the state variables do not change value:

  $$\chi(k+1) = \chi(k)$$

- Measurement:

  $$\zeta(k+1) = \mathbf{H}(k+1)\,\chi(k+1) + \mathbf{w}(k+1)$$

  Transformations from sonar measurement space to feature space are done before the fusing process, as presented in Appendix A. The measurements in this case are already in feature space, and consequently **H** is the identity matrix, and $\mathbf{w}(k+1)$ is a sequence of zero-mean, white, Gaussian measurement noise in feature space corresponding to the noise of the respective sonar measurements.

$$\zeta(k+1) = \chi(k+1) + \mathbf{w}(k+1)$$

- State estimation and covariance computation:
  - State prediction:

    $$\hat{\chi}(k+1|k) = \mathbf{F}(k)\,\hat{\chi}(k|k) + \mathbf{G}(k)\,\mathbf{u}(k)$$

    Consequently, by the reasons explained before:

    $$\hat{\chi}(k+1|k) = \hat{\chi}(k|k)$$

  - State prediction covariance:

    $$\mathbf{P}(k+1|k) = \mathbf{F}(k)\,\mathbf{P}(k|k)\,\mathbf{F}^{\mathbf{T}}(k) + \mathbf{Q}(k)$$

    where $\mathbf{Q}(k)$ is the covariance matrix associated to $\mathbf{v}(k)$, and therefore null, resulting in the following expression:

    $$\mathbf{P}(k+1|k) = \mathbf{P}(k|k)$$

  - Measurement prediction:

    $$\hat{\zeta}(k+1|k) = \mathbf{H}(k+1)\,\hat{\chi}(k+1|k)$$

    In this implementation, the measurement prediction is identical to the state prediction,

    $$\hat{\zeta}(k+1|k) = \hat{\chi}(k+1|k)$$

  - Innovation:

    $$\nu(k+1) = \zeta(k+1) - \hat{\zeta}(k+1|k)$$

  - Innovation covariance:

    $$\mathbf{S}(k+1) = \mathbf{H}(k+1)\,\mathbf{P}(k+1|k)\,\mathbf{H}^{\mathbf{T}}(k+1) + \mathbf{R}(k+1)$$

    where $\mathbf{R}(k)$ is the covariance matrix associated to $\mathbf{w}(k)$, resulting in the following expression:

    $$\mathbf{S}(k+1) = \mathbf{P}(k+1|k) + \mathbf{R}(k+1)$$

  - Filter gain:

    $$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\,\mathbf{H}^{\mathbf{T}}(k+1)\mathbf{S}^{-1}(k+1)$$

    Consequently:

    $$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\,\mathbf{S}^{-1}(k+1)$$

  - Updated state estimate:

    $$\hat{\chi}(k+1|k+1) = \hat{\chi}(k+1|k) + \mathbf{W}(k+1)\,\nu(k+1)$$

  - Updated state covariance:

    $$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\,\mathbf{S}(k+1)\,\mathbf{W}^{\mathbf{T}}(k+1)$$

B.3 Filter Initialization and Uncertainties Associated

Both filter approaches were initialized with the following state and measurement covariances:

$$\mathbf{P}(0|0) = \begin{bmatrix} 100.0 & 0.0 \\ 0.0 & 100.0 \end{bmatrix}, \qquad \mathbf{R}(k) = \begin{bmatrix} \sigma_{o_1}(k)^2 & 0.0 \\ 0.0 & \sigma_{o_2}(k)^2 \end{bmatrix}$$

For lack of information, the initial state covariance $\mathbf{P}(0|0)$ was simply initialized with a high constant. Also, the states were assumed independent causing the matrix to be diagonal. The covariance matrix of the noise associated with the measurements $\mathbf{R}(k)$ was calculated using the variance on the measurements $(\sigma_o(k)^2)$. Again, the matrix is diagonal because it was assumed that the measurements were not cross-correlated. The variances were calculated considering that the maximum error on a measurement is equivalent to $3\,\sigma$.

In the EKF case, the maximum error on a measurement is given by the precision of the sonar sensor. The sonar sensors used have an error equivalent to 1% of the distance measured, therefore,

$$\mathbf{w} = \begin{bmatrix} \Delta r_1 & 0.0 \\ 0.0 & \Delta r_2 \end{bmatrix}, \qquad \begin{array}{l} \Delta r_1 = 0.01\, r_1 \\ \Delta r_2 = 0.01\, r_2 \end{array}, \qquad \mathbf{R} = \begin{bmatrix} \left(\frac{\Delta r_1}{3}\right)^2 & 0.0 \\ 0.0 & \left(\frac{\Delta r_2}{3}\right)^2 \end{bmatrix}$$

where the discrete time reference k was omitted.

In the linear case, the same precision of the sonar sensor is used to compute the maximum error on a measurement in feature space by applying one of the approaches presented in Appendix A.

APPENDIX C

SONAR SIMULATOR

The 2D simulator developed tests sonar configurations in specular environments composed of lines, edges, and corners. The information and processes used by the simulator to calculate the sonar measurements r_1 and r_2 are: the sonar configuration (position and heading of transmitters and receivers); the location of the visible features and the features' reflectance models; a sonar model that considers range and beam angle; a Gaussian process noise to corrupt the sonar returns.

Some simplifying assumptions, commonly used in sonar-based modeling, were employed to reduce the simulator's complexity: the sonar is modeled as a 2D sensor, and all the reflections occur on the transducers' plane; valid sonar returns are limited to one-point or two-point reflections and a single return (single echo); the transducer's beam pattern is considered strong and constant on the sonar beam angle (40°) and zero otherwise; the intensity of the ultrasonic signal does not decay with distance traveled, type of reflection, and number of reflections on the signal path, if the signal is in range it is detected (0.3 to 10 m), otherwise it is ignored; the only error in the measurements is given by an additive Gaussian noise where 3σ corresponds to a 1% error in measurement.

This simulator was created not to test the robustness of the method with respect to uncertainty in the feature and sonar models, but to identify configurations that facilitate the extraction of features to be used as navigational landmarks.

C.1 Sonar Returns

Figures C.1, C.2, and C.3 exemplifies the geometrical relations involved on the calculation of the sonar returns (r_1 and r_2). As shown in Figure C.1, r_1 is derived from c, the distance between the transducer T and the line feature; β is obtained by computing α from φ and γ and applying Equation 2.2; then, to compute b, a, r_2, and the point (x_1, y_1) is straightforward. The derivation of r_1 and r_2 in the edge feature case is clear from Figure C.2. In the case of the corner feature, Figure C.3, the derivation of the variables are extremely simplified by considering only right-angle corners, causing the line segment a to be parallel to the line segment b.

C.2 Simulator Procedure

The simulator calculates the sonar measurements r_1 and r_2 as described in the following procedure:

As previously mentioned, legal reflections are those that do not intersect other features or the robot on their way from the transmitter to the receiver, and do not violate the transmitter and receiver beam angle cones. In addition, the reflections

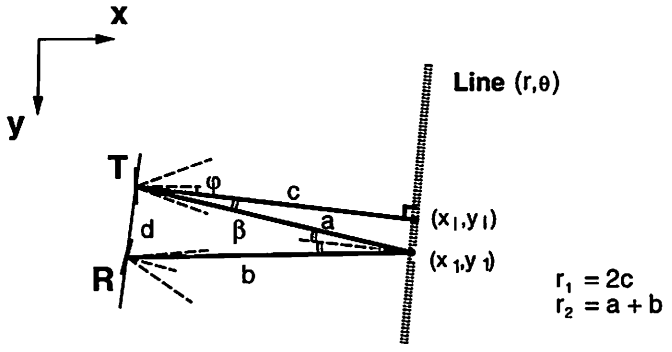


Figure C.1. Readings generated by a line feature

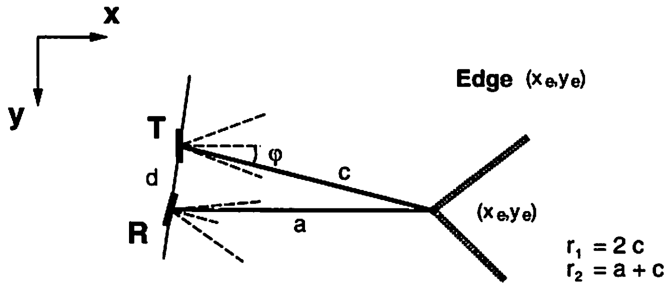


Figure C.2. Readings generated by an edge feature

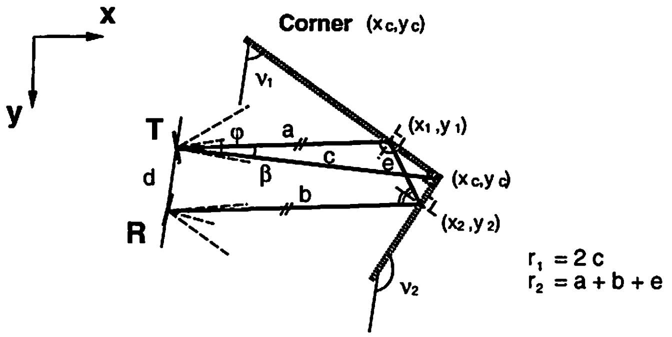


Figure C.3. Readings generated by a corner feature

Simulator Procedure:

1. *For each sonar transmitting (mode T) at time t, check which features are captured by the transmitter's cone, saving these features for analysis.*
   - (a) *For all edge and line features inside the transmitter's cone, check for a one-point reflection to all the receivers;*
   - (b) *For all corner features inside the transmitter's cone, check for a two-point reflection to all the receivers;*
   - (c) *For all virtual corner features with at least one of its line segments inside the transmitter's cone, check for a two-point refection to all the receivers.*
2. *For all legal one-, two-point reflections, save the minimum distance correspondent to time of flight (D-TOF) for each receiver as a measurement (r_1 for a transducer in T mode, or r_2 for a transducer in R mode), and draw those reflections.*
3. *Add Gaussian noise to each measurement r_1 and r_2 and plot a marker at the corresponding half distance from each receiver's heading.*

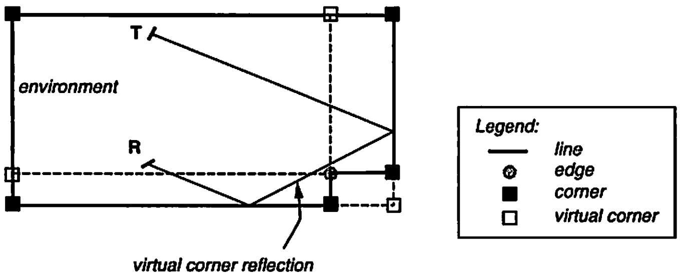


Figure C.4. Example of virtual corner

must be in range (from 0.6 to 20 m), and the reflection points must belong to the part of the feature inside the beam angle cones. Virtual corners are phantom corners originated by two-point reflections on two line segments that do not compose a real corner. An example of virtual corner reflection is depicted in Figure C.4.

APPENDIX D

MOBILE ROBOT: ISAAC

Isaac is a fully autonomous three wheeled robot. Its chassis is a MRV-3 model from Denning Mobile Robotics, Inc.[1], upgraded at our laboratory to support all the processing on-board. The MRV-3 is a synchro-drive chassis with a drive motor and a steer motor. All three wheels are driven in forward and reverse, and steer by maintaining parallel axis through a turn angle, allowing a zero turning radius without rotating the chassis on axis (omnidirectional design).

Several modifications were made on the original MRV-3 model to allow all the processing to be executed on-board. Primarily, it was changed to a VME architecture, causing all previous computational systems to be discarded, and the low-level controllers to be reimplemented on VME processor boards. The VME architecture was selected because of its distributed processing capability, and its widespread use in robotics. Isaac is depicted in Figure D.1 on its tethered configuration (physically connected to power and Ethernet), and a detailed description of the VME architecture is presented on Section D.1.

Figure D.1. Mobile Robot - Isaac

The introduction of a VME cage required extra power causing the number of batteries to double (from 3 to 6, 12VDC batteries), and the addition of two UPS power supplies. Space constraints limited the number of batteries to 6, and the vehicle motors that initially operated on 36VDC had to be changed to 24VDC to

---

[1] Currently Denning Branch International Robotics (DBI).

Chassis Specifications	
Size	0.69m diameter, 1 to 1.3 m height
Weight	≈ 180 kg
Payload	≈ 70 kg
Ground Clearance	0.06 m
Max. Speed	≈ 1.5 m/s
Climbing Ability	≈ 0.02 m threshold
10% Ramp Speed	≈ 0.3 m/s
Untethered Max. Oper.	≈ 2 hours
Drive Amplifier	600W Continuous, 1200 Peak
Steer Amplifier	200W Continuous, 300 Peak
DC-DC Converters	Auto Power-down Dual Supply, DC-DC converters
Power Cage	for the DC-DC converter and Motor Servo Amplifiers
Power Supply (UPS)	2 330W JRS(N+1) Modules, Joule Power Inc.
Battery Power (Motor)	24 VDC, 2 gel type rechargeable 12VDC - 40 Ah
Battery Power (System)	48 VDC, 4 gel type rechargeable 12VDC - 40 Ah

Table D.1. Isaac's chassis specifications

accommodate the power needs of the VME cage (48VDC). Table D.1 presents the specifications of our mobile; part of this information was taken from the Denning MRV-3 manual, and some are only approximations given the modifications made to the original vehicle.

D.1 Computational System

The system consists of a VME cage with 5 CPU boards, a 2 Mbytes global memory board, a 100 Mbytes hard drive, and a custom made IO board. Communication with a outside host computer is done via a wireless Ethernet connection. Figure D.2 depicts the main components of the robot's computational architecture, as well as the final assignment of control processes to processors.

D.1.1 Low-level Controllers

The low-level controllers are responsible for driving the motors (drive and steer) and for the interface with the sensors (encoders, sonars, bumpers). In our case the control was divided in two procedures, each one running on a processor board.

In one procedure, a model-based PD position controller is used to control the drive and steer motors using feedback from the sensors. At each control loop, odometry is computed using encoder information. Bumpers and emergency stop buttons are checked in each servo loop – bringing the robot to a full stop if any of them is activated, and the presence of a joystick is tested – if detected, the system changes to a velocity mode controlled by joystick commands. Moreover, a routine that uses the Hall effect index switch to home the robot to its initial steering angle configuration is provided.

Hardware
Standard VME Card Cage, 12 slots, 6U x 160mm card cage - Zero Corp.
1 VME Sparc board - SPARC CPU-2CE, Force Computers, Inc.
1 68030 board - HK68/V30XE, Heurikon Corp.
3 68030s boards - HK68/V3E, Heurikon Corp.
Global Memory board - MVME 204-2F, Motorola, Inc.
100 Mbytes Hard drive, Digital Inc.
IO board - Multi-function Interface, custom made
Wireless Ethernet - 2 RangeLan2/Access Point, Proxim, Inc. (one on-board)
Operating Systems
Real-time OS - VxWorks v5.0, Wind River Systems, Inc.
Unix - SunOS v4.1.4

Table D.2. VME Computational System

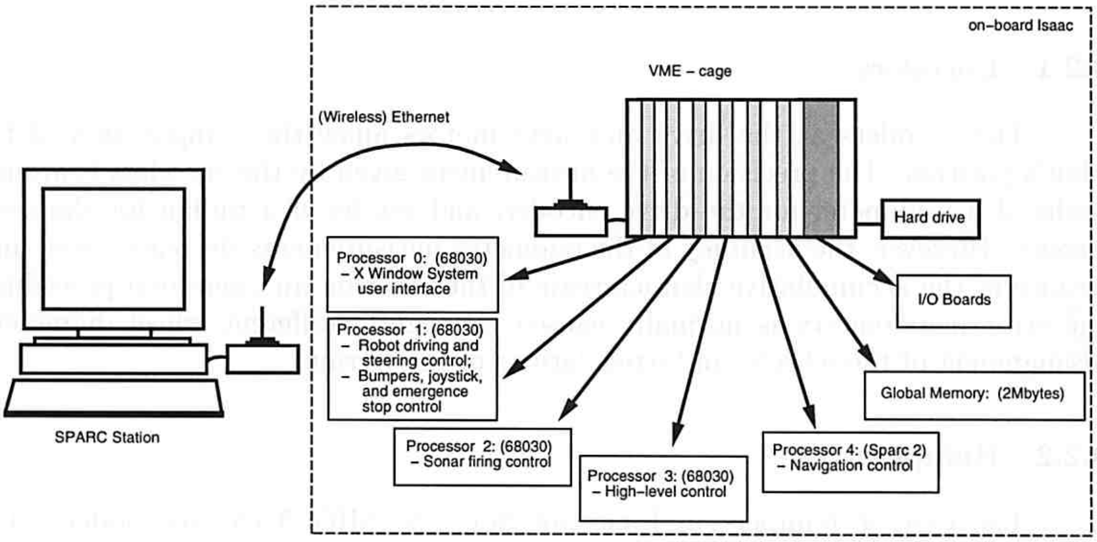


Figure D.2. Isaac's control processes architecture

The other procedure controls the 24 sonar sensors by interfacing with their sonar ranging modules. Currently this routine selects which sensors to activate based on the mode selected, and computes the distance to obstacles using time-of-flight (TOF) computed on the processor board. In future, the sonar ranging modules will be modified to accept two modes of operation (transmit-receive, or only receive) and to compute the TOF in hardware, with a better precision, in the order of millimeters.

D.2 Sensors

Table D.3 presents the sensors previously installed in Isaac. Currently, a two-sensor active sonar array is being designed to test the feature extraction methods presented here. And, in the near future, the upgrade of the sonar ring and the bumpers, in addition to the introduction of CCD cameras and a compass is being considered.

Sensors and Accessories
100 line shaft encoders on steer and drive motors
3 Directional Bumper Switches
Ultrasonic ring with 24 transducers (Polaroid 600 Series), each controlled by a Polaroid 6500 Series Sonar Ranging Module
Hall Effect index switch on steer axis
3 Emergency Stop Buttons
Velocity mode joystick for manually moving the robot

Table D.3. Isaac's sensors and accessories

D.2.1 Encoders

The encoders at the drive and steer motors allow the computation of the robot's position. The precision of the measurement given by the encoders is around tenths of a millimeter for the drive encoder, and tenths of a radian for the steer encoder. However, the accuracy of the odometry measurements decreases over time because of the accumulative characteristic of the error on an open-loop procedure. The error in odometry is normally caused by slightly different wheel diameters, misalignment of the wheels, and irregularities of the terrain.

D.2.2 Bumpers

The current bumpers in Isaac are from the MRV-3 chassis model. The bumpers are 3 directional switches that take the robot to an emergency stop when a certain pressure is exerted against it. This bumper system is only appropriate to avoid further damage after a collision. Another system will be needed for short range detection to facilitate navigation through narrow regions, such as doors.

D.2.3 Sonar Ring

The sonar ring mounting and transducers are from the MRV-3 chassis. Each one of the sonar sensors has its own sonar ranging module, allowing for the independent activation of any subset of the 24 sonars [21]. All the sonars operate at the same frequency (49.4 kHz), causing some activation procedures to produce better results than others because of sonar cross-talk. Moreover, the number of sensors activated simultaneously is limited by the power supply peak current capacity, since each sonar ranging module requires a peak-current of 2 amperes at the moment of firing.

D.3 Odometry Error

The method used here to calculate the odometry error is based on the procedure created by Borenstein and Feng to measure and correct dead-reckoning errors [3]. Our main interest is to show the dynamics of Isaac's odometry error on a common path. The first experiment, in Figure D.3, depicts the desired paths (a counter-clockwise and clockwise rectangular path on the same flat floor), the odometry estimate of the executed paths, and the executed paths, always starting at position $(0, 0)$, showing its tendency to deviate to the right on forward movements, and to undo this error in backward movements.

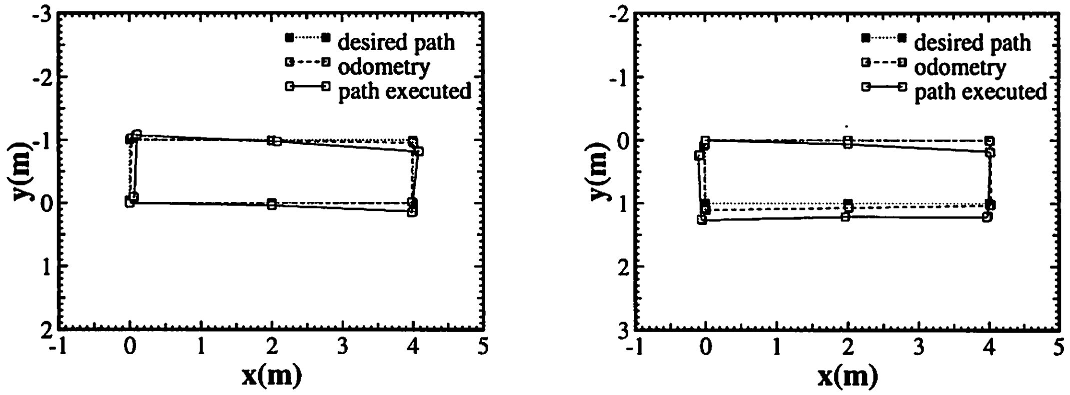


Figure D.3. Isaac's performance on a counter-clockwise path (ccw) - left, and on a clockwise path (cw) - right

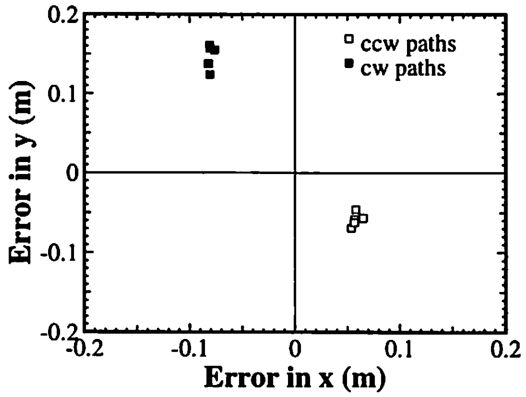


Figure D.4. Isaac's odometry error at the end of path

Figure D.4 presents the final error after five clockwise and five counter-clockwise executions of the same path in the same environment. As shown, the accumulated odometry error is significantly larger in the clockwise path.

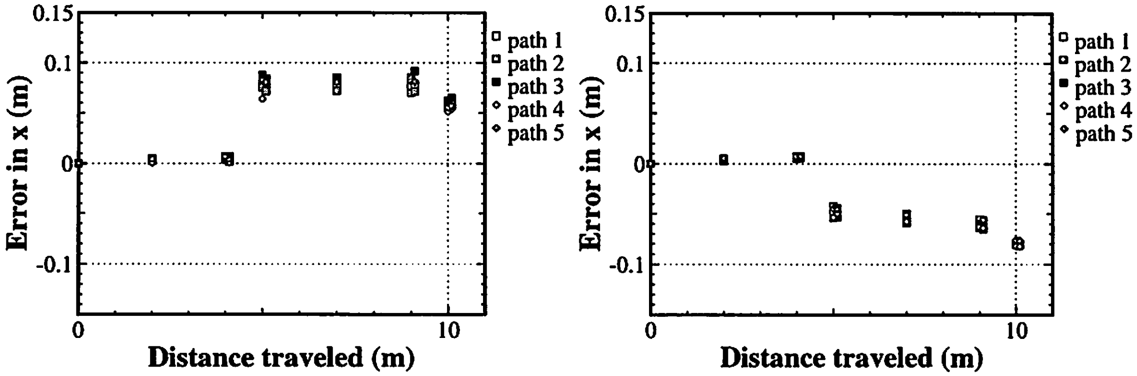


Figure D.5. Odometry error in x direction for ccw (left) and cw (right) paths

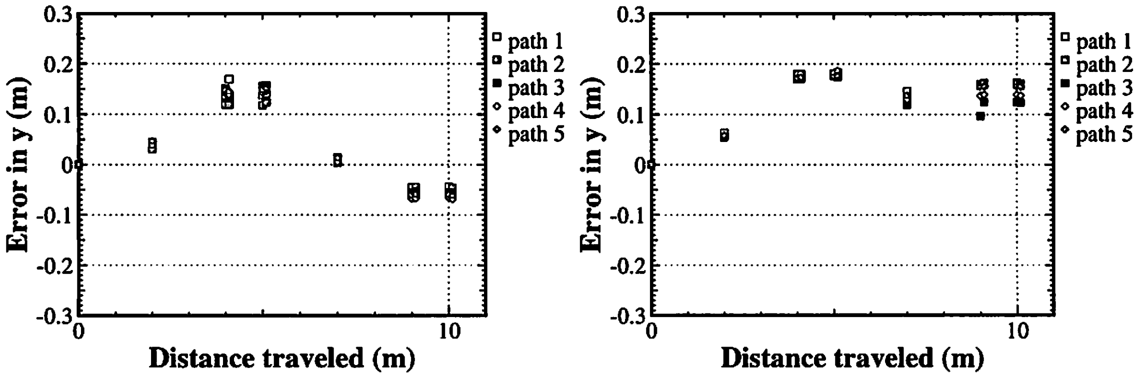


Figure D.6. Odometry error in y direction for ccw (left) and cw (right) paths

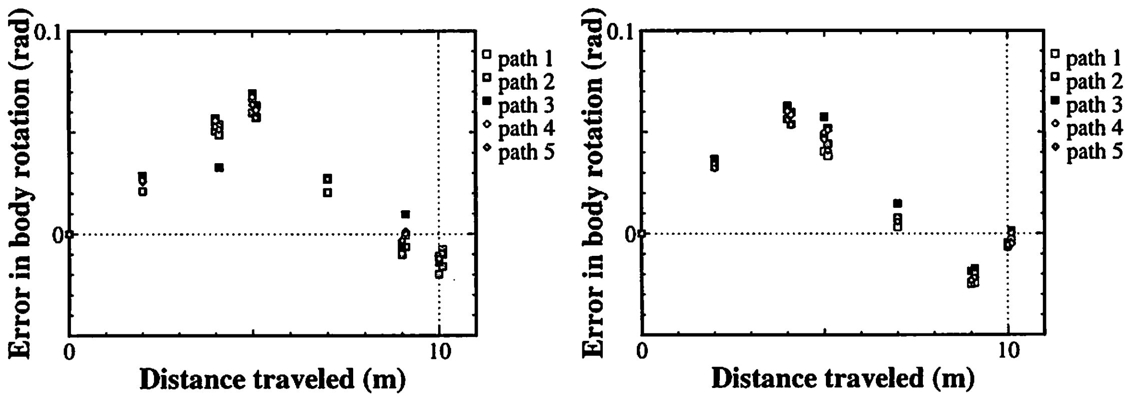
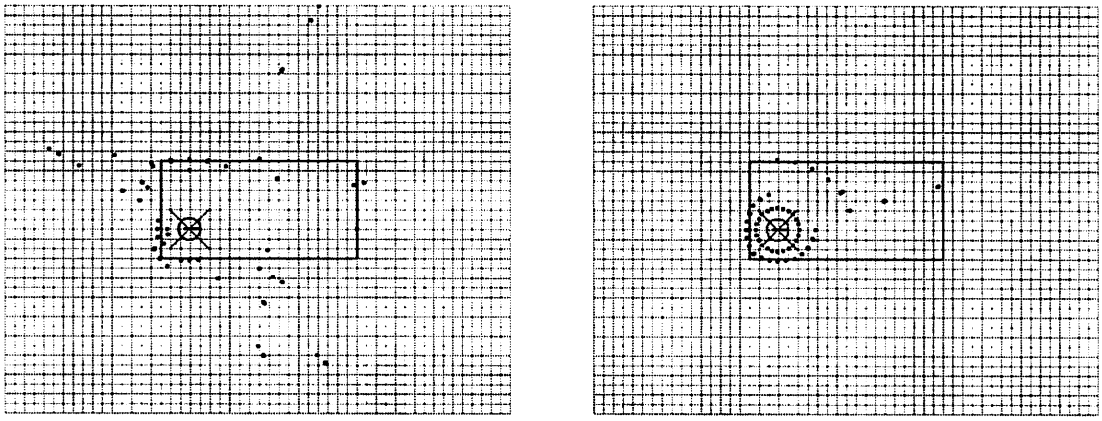
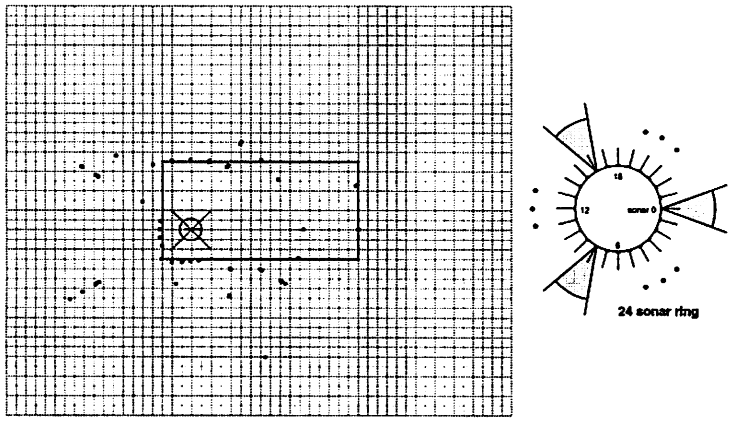


Figure D.7. Error in body rotation for ccw (left) and cw (right) paths

The dynamics of the odometry error is depicted in Figures D.5, D.6, and D.7. These experiments show that Isaac has a small error in the direction of the movement but a consistently larger deviation in the direction orthogonal to the movement, suggesting that it has a systematic error related to a difference in wheels radii or misalignment of the wheels. It is also important to notice that on some paths, for example closed-loop paths, the odometry error does not necessarily increase monotonically. And thus, a correct analysis of the contribution of other sensors, such as sonar and laser, to pose localization should take this fact into account.

D.4 Simultaneous Sonar Firings





Firing Patterns	
Sequential (sonar #)	Interleaved (sonar #)
0 8 16	0 8 16
1 9 17	4 12 20
2 10 18	7 15 23
3 11 19	3 11 19
4 12 20	6 14 22
5 13 21	2 10 18
6 14 22	5 13 21
7 15 23	1 9 17

Figure D.8. Simultaneous sonar firings, three sequential, all, and three interleaved, respectively

Simultaneous firings are of interest because of the low bandwidth of sonars (12 Hz). Figure D.8 depicts experiments with different activation patterns, where each sensor was activated 10 times from the same position on a rectangular room of dimensions 3 by 6 meters, approximately. The distances to obstacles obtained from the sensors are represented by dark points, assuming a ray-trace model, and the robot is represented by a circle. Notice that, when all sensors are activated at the same time (top right figure), cross-talk and probably problems with the power supply corrupted completely the data. Figure D.8 also shows that it is not sufficient to activate equidistant sensors (top left figure, with 38% consistently correct results), but from one firing set to the next, the sensors selected should also be equidistant (interleaving - bottom figure, with 54% consistently correct results), as reported in the table.

D.5 Active Sonar System

As previously discussed in Chapter 1, the sonar configuration is an important factor on the precision of the measurement obtained. Here we present a sonar apparatus that allows the system to change the sonar configuration to improve feature detection and identification.

Figure D.9 shows the functional diagram of the two-sonar array. The pan and tilt of the axis affords the localization of features at any direction, and at any height respectively. In addition, the extra pan on each transducer allows the system to adjust the sonar configuration to feature distance and feature type, thereby producing better quality measurements.

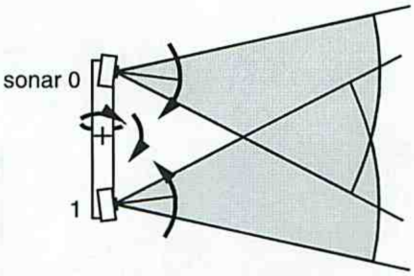


Figure D.9. Active sonar system

The sonar ranging modules from Polaroid have to be modified to accept the two modes of operation (transmit-receive, and only receive) [21]. Figure D.10 presents a way of modifying the sonar ranging module to permit only reception by bypassing the transmission signal. In this circuit, currently under development, the shaded circuit parts belong to the original Polaroid 6500 Series module, the transducer device connects to NEW E1 and NEW E2, and the switch SW1 selects hardware control timing (1), or software control timing (2).

Timing diagrams depicting how the circuit works in reception mode are presented in Figure D.11; in transmission-receive mode the circuit behaves like the original sonar ranging module. The timing diagram on the right depicts the hardware control timing (SW1-1). Notice that the original transmission pulses on XMIT are suppressed, and a single pulse is used to energize the transducer (output of 123-1),

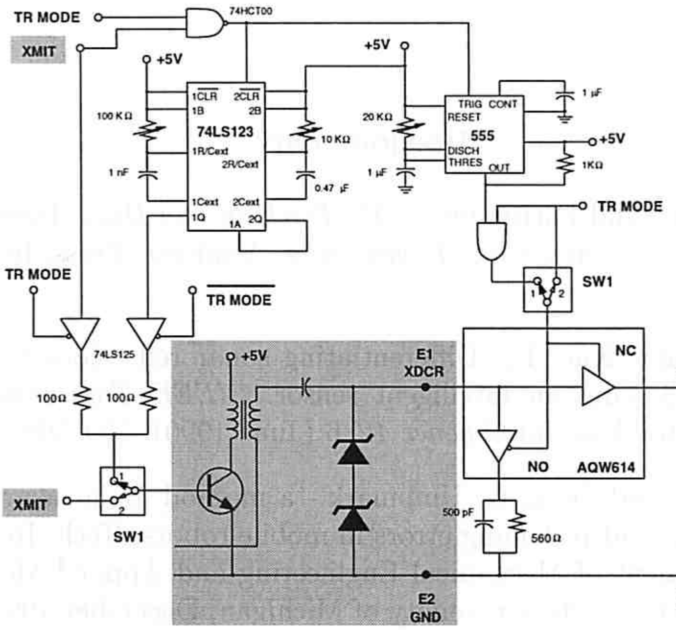


Figure D.10. Additional circuit to allow reception without transmission

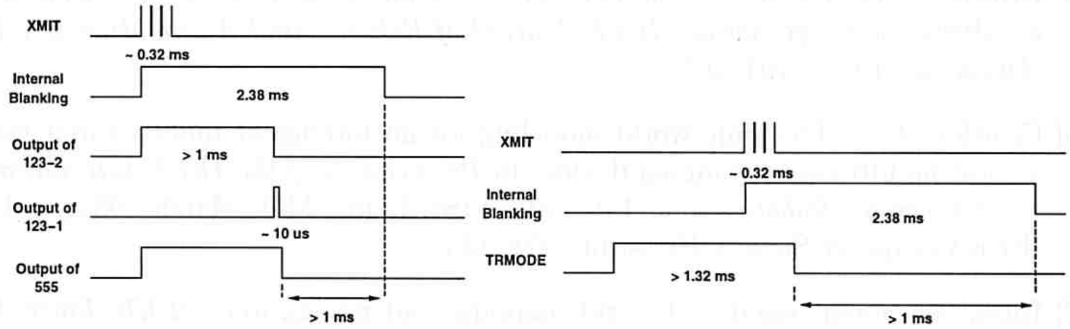


Figure D.11. Circuit timing in reception mode for both switch configurations

followed by the switching to the original reception configuration (output of the 555 turning off) at the end of the internal blanking period. The software control timing (SW1-2) is presented in the diagram on the left of Figure D.11. The main difference between the software and hardware timing procedure is that the former uses the TRMODE signal to time the circuit, reducing the extra hardware needed. On the other hand, the software timing procedure requires a processor board capable of producing pulses on the order of milliseconds.

BIBLIOGRAPHY

[1] Bar-Shalom, Y., and Fortmann, T. E. *Tracking and Data Association*, vol. 179 of *Mathematics in Science and Engineering.* Academic Press, Inc., Orlando, FL, 1989.

[2] Barshan, B., and Kuc, R. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 6 (June 1990), 560–569.

[3] Borenstein, J., and Feng, L. Umbmark - a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. Tech. Rep. UM-MEAM-94-22, Department of Mechanical Engineering and Applied Mechanics, Mobile Robots Laboratory, The University of Michigan, December 1994.

[4] Borenstein, J., and Koren, Y. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation 7*, 3 (June 1991), 278–288.

[5] Brown, M. K. Feature extraction techniques for recognizing solid objects with an ultrasonic range sensor. *IEEE Journal of Robotics and Automation RA-1*, 4 (December 1985), 191–205.

[6] Crowley, J. L. Dynamic world modeling for an intelligent mobile robot using a rotating ultra-sonic ranging device. In *Proceedings of the IEEE International Conference on Robotics and Automation* (St. Louis, MO, March 1985), vol. 1, IEEE Computer Society Press, pp. 128–135.

[7] Elfes, A. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation RA-3*, 3 (June 1987), 249–265.

[8] Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer Magazine* (June 1989), 46–57.

[9] Elfes, A. Dynamic control of robot perception using multi-property inference grids. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Nice, France, May 1992), vol. 3, IEEE Computer Society Press, pp. 2561- 2567.

[10] Gelb, A., Ed. *Applied Optimal Estimation.* M.I.T. Press, Cambridge, MA, 1974.

[11] Kleeman, L., and Kuc, R. An optimal sonar array for target localization and classification. In *Proceedings of the IEEE International Conference on Robotics and Automation* (San Diego, CA, May 1994), vol. 4, IEEE Computer Society Press, pp. 3130–3135.

[12] Kleeman, L., and Kuc, R. Mobile robot sonar for target localization and classification. *International Journal of Robotics Research 14*, 4 (August 1995), 295–318.

[13] Kuc, R., and Siegel, M. W. Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 6 (November 1987), 766–778.

[14] Leonard, J. J., and Durrant-Whyte, H. F. *Direct Sonar Sensing for Mobile Robot Navigation*, vol. SECS 175 of *The Kluwer International Series in Engineering and Computer Science.* Kluwer Academic Publishers, Norwell, MA, 1992.

[15] Manyika, J. M., and Durrant-Whyte, H. F. A tracking sonar sensor for vehicle guidance. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Atlanta, GA, May 1993), vol. 3, IEEE Computer Society Press, pp. 424–429.

[16] Matthies, L., and Elfes, A. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Philadelphia, PA, April 1988), vol. 2, IEEE Computer Society Press, pp. 727–733.

[17] Maybeck, P. S. *Stochastic Models, Estimation, and Control*, vol. 141-1 of *Mathematics in Science and Engineering.* Academic Press, Inc., New York, NY, 1979.

[18] Maybeck, P. S. *Stochastic Models, Estimation, and Control*, vol. 141-2 of *Mathematics in Science and Engineering.* Academic Press, Inc., New York, NY, 1982.

[19] Pagac, D., Nebot, E. M., and Durrant-Whyte, H. An evidential approach to probabilistic map-building. In *Reasoning with Uncertainty in Robotics: International Workshop; Proceedings / RUR'95* (Amsterdam, The Netherlands, December 1996), L. Dorst, M. van Lambalgen, and F. Voorbraak, Eds., Springer, pp. 164–170. (Lecture notes in Computer Science; Vol. 1093).

[20] Peremans, H., and Van Campenhout, J. Tri-aural perception on a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Atlanta, GA, May 1993), vol. 1, IEEE Computer Society Press, pp. 265–270.

[21] Polaroid Corporation, OEM Components Group. *Ultrasonic Components.* Cambridge, MA, October 1994.

[22] Simmons, J. A., and Grinnell, A. D. The performance of echolocation: Acoustic images perceived by echolocating bats. In *Animal Sonar: processes and Performance* (1988), P. E. Nachtigall and P. W. B. Moore, Eds., vol. 156 of *NATO ASI Series. Series A. Life Sciences*, Plenum Press, pp. 353–385. Proceedings of a NATO ASI on Animal Sonar Systems, held September 10-19, 1986, in Helsingor, Denmark.