Reliable Mobile Robot Localization

PhD Dissertation Proposal
submitted by

Stergios I. Roumeliotis
email: stergios@robotics.usc.edu

Guidance Committee

George A. Bekey[1]
Jerry M. Mendel
Elias B. Kosmatopoulos
Gaurav S. Sukhatme
Stefan K. Schaal[2]

University of Southern California
Los Angeles, California
May 4, 1999

[1]Chairman
[2]Outside Member

Contents

Abstract

Reliable localization is the problem of determining the position of a mobile with respect to a global or local frame of reference in the presence of sensor noise, uncertainties and potential failures. The basic idea behind many mobile robot localization techniques is to combine sensor data with a priori knowledge about the specifications of these sensors, the structure of the mobile platform, and the environment the vehicle travels in. For example, it is often assumed that a detailed map of the area is known. In this case, the problem of identifying the position of the robot is the problem of finding an area within the map such that the expected sensor values are at all times in accordance with the actual readings.

Many mobile robot applications depend on the ability of the robot to localize using sparse and many times uncertain information concerning its position. For example, fusion of odometric and/or inertial sensors with exteroceptive sensors has received significant attention in the past two decades. The techniques mostly used to process sensor data from different sources are Kalman filtering and Bayesian estimation. When an initial position estimate is known, a Kalman filter is capable of continuously updating this estimate combining noisy data from a variety of sensors. Bayesian multiple hypothesis testing is suitable for resolving the ambiguity associated with the identification of detected landmarks. The proposed study intends to exploit the strengths of both methods. There have been few attempts to combine these two techniques but require strict assumptions. For example it is often assumed that either the motion tracking sensors are perfect or the exteroceptive sensors are capable of extracting a detailed and precise layout of the surroundings. One of the objectives of the proposed research is to introduce a general framework that subsumes both approaches in a single architecture and exploits noisy kinetic information in order to reduce the uncertainty associated with the presence or absence of coarse, sparse, reappearing features of the environment. Restructuring of the exact framework leads to the multiple model adaptive estimation applied here to fault detection and identification.

It is a common assumption for current localization systems that all sensors provide information at the same rate. Reality dictates that in most

cases noisy data from kinetic sensors arrive at high rates while absolute attitude and/or position data tend to be scarce. The intermittent nature of the externally provided or extracted absolute information implies that the system is observable only at certain time instants. This observability limitation restricts the potential autonomy of a robot and causes the loss of precious information whenever real-time filtering is involved. Smoothing of the attitude estimates reduces the overall uncertainty and allows for longer traverses before a new orientation measurement is required. Similar post-processing of the position estimates supports reconstruction of a robot's trajectory and increases the accuracy level during mapping tasks.

A simple scenario of an office environment with only one type of features is presented to testify the validity of the general approach for combining Bayesian estimation with Kalman filtering in a multiple hypothesis framework. Similarly the simple problem of smoothing the attitude estimates of a robot moving in a 2-D environment is formulated to demonstrate the improvement in the quality of the orientation estimates.

It is the objective of this research to ratify the ability of the multiple hypothesis framework to deal with (i) more complicated environments containing different types of features with different probabilities, (ii) the absence of expected features, (iii) the detection and identification of different types of failures. In addition, this research will focus on the re-formulation of the smoothing problem in order to be applied in the following cases: (i) smoothing of the attitude estimates in the 3-D case for enhanced localization, (ii) smoothing of the position estimates for enhanced mapping.

Chapter 1

Introduction

1.1 Motivation

Mobile robots are fast becoming one of the most prominent applications of robotics. They have moved from the industry floor and are now being sent on missions to other planets [32], remote areas [66], or dangerous radioactive sites [16]. The potential applications for mobile robots do not include only special missions. They are also being used as guides in museums [61] and for entertainment purposes [29]. In order for a mobile robot to travel from one location to another it has to know its position at any given time. In most of the cases today this is achieved by either having a human in the navigation loop [15] who directs the vehicle remotely or by constraining the robot to operate in a certain area, precisely mapped [19], [61] or suitably engineered; i.e. marked with beacons [40] or other artificial landmarks [46].[1] The level of autonomy depends predominantly on the ability of the robot to know its exact location using minimal previous information for the environment, represented in a simple way.

[1] From now on the words landmark and feature will be used interchangeably.

1.2 General statement of the problem and review of previous approaches

The localization problem for a mobile robot consists of its ability to answer the question *"Where am I?"*, with respect to a given set of reference coordinates. This problem could be solved if the area in which a robot moves could be marked in a unique separable way. That means each place should contain information that will make it recognizable and distinguishable from all other places in the area. A first approach to the problem would require each location to have a unique "signature". This signature may be provided directly or calculated with the aid of an external source. This is already available for some areas where the Global Positioning System (GPS) signals are available. The triangulation of the distances from the satellites in view assigns a unique pair of numbers for each location on the surface of the planet. Longitude and latitude are the geographical coordinates that are used for long distance navigation of airplanes, ships and even cars. This attractive method suffers from a wide variety of problems. The first and most important one is the accuracy of the signals. Even some of the most expensive commercial GPS devices provide an accuracy of a few tens of centimeters which is more than enough when a car drives from one side of a city to the other but it is not acceptable for robots that move in cluttered environments or handle material located within small distances. Another serious drawback is that the GPS signals are not available indoors or in the vicinity of tall buildings, bridges and structures that occlude the view of the necessary number of satellites. Finally, the cost of a GPS receiver is too high for many robotic applications.

In many scenarios where GPS is not available, the signature of a location has to be determined based on the special characteristics of the area itself. In this case the dissimilarity between signatures of different locations has to be identifiable by the sensors used for this reason. There are many attributes of the environment that can be exploited to gain signature diversity. These can be dynamic or static. Temperature, chemical consistency of the air, light and sound conditions, belong in the first category and though they can be measured accurately, are not appropriate for marking an area. This is because neighboring locations tend to have similar values for the same quantities. In addition, these values can change with respect to time in an unpredictable or difficult to model fashion. More static characteristics need to be detected

in order to provide distinctive "signatures". The color and the shape of a location are strong candidates for discriminating different areas and tend to remain unchanged over long periods of time. Visual memorization of each place may be sufficient for distinguishing between them. Although the homing capabilities of many types of insects [10], [64] and other animals [25], [20] depend on this type of localization, there are serious implementation issues that have to be resolved before this technique can be applied to mobile robot localization. The requirements for storing and processing the numerous collected images are probably the most difficult to overcome.

Another way to achieve the desired distinguishability of a location of concern is by preparing a detailed volumetric map of the area and then matching the information from exteroceptive sensors with the information stored in the map. This approach suffers from the following drawbacks: (i) The preparation of the map adds significant overhead to the process. It has to be full, exhaustive, and precise enough to exclude any ambiguities and avoid misinterpretation, (ii) The exteroceptive sensors have to be able to supply the robot with the same level of detail as that imprinted on the map. A laser scanner is a good example of such a device. The cost of such sensors is prohibitive for many mobile robot applications, (iii) The processing requirements to obtain the necessary information and match it to the corresponding locations on the map call for a powerful processor on-board the robot if real-time operation is demanded. This adds further expenses to the cost of this approach.

Instead of building elaborate detailed maps of almost every bit of the environment, an alternative would be to compose a topological map of the area where separate locations are described in terms of the actions or directions required for a robot to follow in order to transfer from a known initial location to the desired one. This can be seen as creating a graph of the area where the nodes are locations of interest and the arcs contain descriptions for the transitions. These directions are usually described in terms of the capabilities of the particular vehicle. A more unifying description of the topology would be to use the metric distances between two locations of interest. Although this approach allows one to abstract from the robot currently utilized, it introduces extra problems such as path planning and motion control. In addition, a mobile robot is capable of distinguishing between adjacent areas only if it can precisely track its own position. Regardless of how the map was derived, the vehicle is considered capable of following designated paths. This

is conditioned on the robot's ability to estimate at all times its displacement with respect to some initial or intermediate locations.

Most of the current localization efforts have focused on supporting high quality displacement estimates. Different sensing devices and odometric techniques have been exploited for this purpose. The common characteristic of these approaches is that they rely on the integration of some kinetic quantity. The main drawbacks of any form of odometry are: 1) Every sensor monitoring the motion of the vehicle has a certain type and level of noise contaminating its signal. Integration of the noisy components causes gradual error accumulation and makes the estimates untrustworthy. 2) The kinematic model of the vehicle is never accurate. For example, we do not know with infinite precision the distance between the wheel axes of the vehicle. 3) The sensor models also suffer from inaccuracies and can become very complicated. For example, the use of complicated models to describe the gyroscope drift. 4) The motion of the vehicle involves external sources of error that are not observable by the sensors used. For example, slippage in the direction of motion or in the perpendicular direction is many times not detected by the motion sensors. Externally provided or extracted information is necessary from time to time if we wish to keep the error bounded. This group of approaches is also referred to as "dead-reckoning"[2] and we will revisit them in more detail in the following chapters.

After briefly reviewing some of the marking strategies that could be exploited by a variety of localization algorithms, it is obvious that determining the position of the robot is not an easy task for any of these approaches. In the proposed thesis, a schema that relies on a *combination of different types of "signatures"* in order to distinguish between different places, results in a flexible, more robust and easier to implement solution to the localization problem. For instance, instead of reproducing on a detailed map all the surroundings of a robot for each area of interest, it may be desirable to attempt to create a more compact representation, i.e. to extract the minimal information that makes a location unique *with respect to other locations*. This is not necessary restricted to the particular attributes that a certain spot has. It can be generalized to include information for this location with respect to its

[2]Sensors used for measuring kinetic quantities such as velocity, acceleration, etc will be referred here as *kinetic* or *proprioceptive* sensors. Amongst these are odometric sensors such as wheel encoders or inertial sensors such as gyroscopes and accelerometers

displacement from other known or previously visited sites. More specifically the discrimination requirement can be approached by considering not only "What does this location look like" but also "How do I get to this location".

If, for example, a robot is capable of distinguishing two fairly similar doors just by looking at them (as humans do by reading a sign on a door) then the problem is essentially solved. This is one of the prominent ways that humans use to recognize their location. Certain aspects of different locations are compiled in a compact form of mnemonic representation. Not all the information is preserved, which sometime makes it unreliable. Our ability to distinguish between one place or another is not unlimited. There are certain cases where the memory is not developed enough to support an unambiguous decision or prevent a mistake. This problem is even more prevalent in artificial machines such as robots equipped with sensing devices like a camera, for example. The state of the art of pattern recognition techniques is far from being able to mirror the perceptual and processing capabilities of the human eye.

This compact definition of "What does this location look like" is in essence the definition of a landmark used in a localization algorithm. Landmark based localization differs from visual memorization of the environment in that only certain spots in an area are marked this way while the rest of the vicinity is marked as "How do I get to this location". Before going into a detailed examination of the landmark based localization we reiterate the meaning of "How do I get to this location" for the specific case of a two dimensional (2-D), global, metric map. The marking of the area in this case is with respect to some known world frame and consists of the coordinates (x, y) of each location. The fact that these coordinates are calculated with respect to some other location makes them actually $(\Delta x, \Delta y)$, i.e. this set of numbers marks the final location by its transition from the initial location, thereby following a meaningful pre-specified way. In this case it is moved by Δx on the x axis and by Δy on the y axis. The initial location can be arbitrarily chosen as well as the coordinate system or the units used.

Chapter 2

Literature Review

The following review, contains representative examples of different approaches to mobile robot localization.

2.1 Position Tracking

The errors associated with position tracking can be divided into two main categories: (i) systematic and (ii) stochastic. Stochastic errors are due to sensor noise while systematic errors are primarily due to errors in the values of the parameters used to describe the system. For example, systematic errors occur when the radii of the wheels of a robot are not known precisely and they are assumed to be equal to a nominal value.

In order to deal with systematic errors which are particularly notable in indoor applications, J. Borenstein and L. Feng present in [8] a calibration technique called the UMBmark test. The dominant systematic error sources are identified as the difference in wheel diameter and the uncertainty about the effective wheel base. By measuring the individual contribution of these vehicle specific, dominant error sources, they counteract their effect in software.

2.1.1 Position Tracking and Kalman filtering

Stochastic errors cannot be predicted or compensated off-line. Instead filtering is introduced to estimate the values of the kinetic quantities of interest in the presence of noise and combine their values in order to produce an estimate of the position of the robot. Kalman filtering offers a powerful method for low-level fusion in mobile robots, provided that the filter's modeling assumptions can be satisfied, and that the uncertainty models that it requires are available and reliable. The data integration process executed by the Kalman filter is modular and can accommodate a large variety of sensory measurements as long as the error covariance matrices are available. The sequential processing of sensor data makes the Kalman filter a very flexible fusing mechanism that can process data from different sensors arriving at different frequencies [56]. There exist a well-developed body of literature on performance, stability and consistency of the filter, as well as tests for data validation and performance consistency. These tests provide means to reject spurious data on-line, and to check that state estimate errors satisfy certain conditions on their statistics in spite of the use of approximations [3]. There also exist different techniques for error compensation in linearized filters such as the use of pseudo-noise covariance and compensation for bias in errors.

What follows is a representative group of efforts that focus on improving the quality of the estimates when filtering is applied to position tracking. In [9], the authors use measurements from odometric sensors (wheel encoders) and INS (gyroscope) during different time intervals. Their method, gyrodometry, uses odometry data for the majority of time, while substituting gyro data only during brief instants (e.g. when the vehicle goes over a bump) during which gyro and odometry data differ drastically. This way the system is kept largely free of the drift associated with the gyroscope and compensates for the non-systematic errors introduced by odometry. In [23], Y. Fuke and E. Krotkov use a complementary Kalman filter introduced in [11] to estimate the vehicle's attitude from the accelerometer signal in low frequency motion and the gyro signal in high frequency motion. The attitude information is then used to calculate the rotational transformation matrix from the vehicle coordinate system to the ground coordinate system. This matrix transforms the velocity information to a position increment. In [4], the authors use a low cost INS system (3 gyroscopes, a triaxial accelerometer) and 2 tilt sensors. Their approach is to incorporate in the system a priori information about

the error characteristics of the inertial sensors and to use this directly in an Extended Kalman Filter (EKF) to estimate position before supplementing the INS with absolute sensing mechanisms. The main drawbacks of this work is that the statistical assumptions for the accelerations are not justified and the orientation information is missing in the calculation of the position.

2.1.2 Position Tracking and Landmarks

Landmark Methods rely on the recognition of landmarks to keep the robot localized geometrically. Landmarks may be given a priori (for example passive or active beacons) or learned by the robot as it maps the environment. While landmark methods can achieve impressive geometric localization, they require either engineering of the environment to provide a set of adequate landmarks, or efficient recognition of features to use as landmarks.

If a robot carries only proprioceptive sensors that monitor the motion of the vehicle, then the system is not observable. This means that the Kalman filter will drift away from the real position estimate unless absolute position information is available to the robot at frequent intervals. Most of the current research efforts, incorporate some form of exteroceptive information in order to update the position estimate and thus keep the uncertainty bounded. This usually requires engineering of the environment with artificial landmarks or beacons. Work related to absolute localization is presented by J. J. Leonard and H. F. Durrant-Whyte. In [40], they develop a system in which the basic localization algorithm is formalized as a vehicle-tracking problem, employing an *Extended Kalman filter* (EKF) to match beacon observations (environment features) to a navigation map to maintain an estimate of the position of the mobile robot. In [7], the authors also use an EKF in order to fuse odometry and angular measurements of known landmarks (light sources detected using a CCD camera). In [5], E. T. Baumgartner and S. B. Skaar estimate a vehicle's position and orientation based on the observation of visual cues located at discrete positions within structured environment. An EKF is used to combine these visual observations with sensed wheel rotations in order to continuously produce optimal estimates.

In another approach to the position tracking problem supported by maps of the environment, instead of Kalman filtering, the *SPfilter* is introduced [47]. In this paper the authors use a version of the *symmetries and per-*

turbation model method to process uncertain geometric information from a monocular camera and a laser range-finder. Information provided by the laser sensor is used both to determine the location of walls in front of the robot, and by processing the intensity image, to determine the location of vertical edges that can correspond to corners, as well as door and window frames. Odometry is used to predict the location of walls and edges. Matching of these elements with an a priori map of the environment provides the required correction to the degrading odometry. Initial estimate of the position of the robot is necessary for the initialization of the algorithm. The correspondence between elements on the map and elements that the robot senses is based on vicinity criteria. When there is more than one candidate for this pairing, the measurement is discarded.

Most of the previous methods focus on optimizing *position tracking*. These methods fail when the initial estimate of the position is not available. In addition, the common assumption made is that a landmark appears before the position tracking estimate degrades to the level that the robot cannot identify the landmark in sight. These algorithms cannot deal with cases where the environmental features appear to be identical.[1] The decision for the identity of the landmark has to be made immediately and it is based on the current position estimate of the robot and the topology of the landmarks. The nearest neighbor is assumed to be the feature detected. If more than one identical features are in the same area - the size of which is determined according to the uncertainty of the current position estimate of the vehicle - this information is discarded by the robot.

2.2 Absolute Localization

In this family of approaches the initial position of the robot is assumed to be unknown and it has to be determined by the robot based on information from maps of the environment. In some cases landmarks are being used and in others a grid-based description of the area is involved.

[1] Two landmarks appear to be identical if the feature extraction module is not able to distinguish them based on the information provided from the exteroceptive sensors available to the robot.

2.2.1 Absolute Localization and Landmarks

One category of landmark based mobile robot localization approaches, relies on angular measurements of known but indistinguishable landmarks [31]. The assumption is that 3 landmarks have to be in sight at all times. For example, in [57], Sugihara addresses a version of the localization problem where the points in the environment look identical. The robot measures the direction of rays each of which emanates from the robots position and goes through an observed point. Sugihara gives an $O(n^3 \lg(n))$ algorithm for finding the robot's position and orientation such that each ray pierces at least one of the n points in the environment. Avis and Imai [2] and Krotkov [37] extend Sugihara's results to the case that angle measurements include errors. In [37], the robot is equipped with a CCD camera and it is assumed to move on a flat surface. A map of the environment is available to the robot. The camera extracts vertical edges from the environment. These edges do not contain enough information to locally determine which vertical edge corresponds to which point in the map. The problem is to identify the position and orientation of the robot by establishing the correspondence between the directions of the vertical edges and points on the map. This approach avoids the 3D reconstruction and the feature identification problem. It is formulated as a search in a tree of interpretations [27] (pairings of landmark directions and landmark points). An algorithm to search the tree efficiently is developed taking into account errors in the landmark directions extracted by image processing. The main drawbacks of this algorithm is the time requirements $O(n^4)$ and its inability to deal with spurious measurements. Only the worst case in the accuracy of the estimate is calculated and it lacks the ability to incorporate information provided by the odometry.

In [6], the landmarks are considered to be *distinguishable*. This paper deals with the problem of how to relate robot-centered information with global centered information in an environment with landmarks. The assumption is that the robot can detect these landmarks and measure their bearings relative to each other. The map of the environment is used to determine the location of the robot. This is another triangulation approach. At least three landmarks have to be in sight in order to estimate the position of the robot. The algorithm presented assumes that more than 3 landmarks are available and thus the system is overdetermined. Least squares estimation is applied and the major contribution of the suggested method is that the derived al-

gorithm is linear in the number of landmarks. Similarly, in [58], the authors are also using distinguishable landmarks of the environment. The focus of this paper is to show that for a certain error in the angle measurement, the localization error varies depending on the configuration of the landmarks. In [59], a *Neural Network*-based approach is presented to the problem of learning the most suitable, distinguishable landmarks for mobile robot localization. The choice of a landmark as useful depends on the improvement to the belief level for the position of the robot in an office environment. The presented algorithm during each step of the training assumes an initially uniform belief function centered around the position of the landmark under investigation. The results presented are for the best case when the initial belief function for the robot's position is the same one that was used during the training phase. Another key limitation of this approach is that it does not learn the locations of the landmarks ((x, y) coordinates). It only learns to associate sensor readings with locations that the robot visits.

A more qualitative approach is presented in [42]. Long surfaces such as walls and corridors are used as topological primitives for localization and mapping. Each of the landmarks detected is identified based on its type (left wall, right wall, corridor), rough size and approximate position and orientation. All these criteria used to compare landmarks are either qualitative or approximate. There is no probabilistic interpretation of their similarities and a decision has to be made immediately after a feature is detected. It is either decided that it is the same with another landmark previously detected and stored in a map, or it is assumed to be a new feature of the environment. There is no notion of the confidence level related with each decision made. Another drawback of this approach is that it does not allow for more than one hypotheses for the location of the robot to exist and thus the decision cannot be postponed for a latter time when more information is available. Finally, the graph connectivity used for the landmark disambiguation depends heavily on the precise knowledge of the robots orientation and it cannot be used if this transition has not been done before.

2.2.2 Absolute Localization and Dense Sensor Maps

Dense Sensor methods [28] attempt to use whatever sensor information is available to update the robot's position. They do this by matching dense

sensor scans against a surface map of the environment, without extracting landmark features. Thus, dense sensor matching can take advantage of whatever surface features are present, without having to explicitly decide what constitutes a landmark.

Markov Localization uses a discrete representation for the a priori and a posteriori probabilities of the position of the robot. A grid map or topological graph has to cover the space of robot poses and a probability is assigned to each element of this space. The key idea is to compute the discrete approximation of a probability distribution over all possible poses[2] in the environment. The area around the robot has to be updated at every step to provide the new probabilities for each cell for the robot to be there. Computationally this technique depends on the dimensionality of the grid and the size of the cells. The main advantage of this technique is that it is capable of localizing the robot even when its initial position is unknown. This is essential for an autonomous robot since its initial position need not be given when the robot is switched on or it gets lost due to an accident or a failure. They can be categorized according to the type of discretization they rely on. In most cases[3], a fine-grained grid-based approximation of the environment is used to calculate the distribution of the robot's position. In these cases metric maps of the environment are being used. These can be CAD maps that consist of line segments or occupancy grid maps. In all cases the map is used to compute what the sensor readings should be from a given cell in the state space. In [60] for example, a probabilistic grid-based algorithm is used for localization while a topological map is derived for the path planning task. The main drawback of this method, namely large memory and processing requirements, are due to the grid-based representation of the area. Another disadvantage is that the accuracy of the method depends on the size of the grid used. The method lacks the ability to optimally fuse different sources of proprioceptive sensor information and capitalize in the existing framework of Kalman filtering based position tracking approaches.

Kalman filtering can be applied to the case of Dense Sensor methods [28]. Scan matching (model matching) is performed by rotating and translating every range scan in such a way that a maximum overlap between the sensor

[2] *Pose* denotes position and orientation.

[3] In some cases a topological discretization of the environment is used. Landmarks are detected to localize the robot and the assumption commonly made is that these areas (landmarks) are distinguishable.

readings and the a priori map emerges. The advantage of this method is that it can globally localize the robot precisely given accurate inputs. The main drawback of the approach is the number of computations required for the model matching. A reasonable solution would be to incorporate odometry and based on the odometric estimate of the position of the robot, to limit the search to small perturbations of the sensor scans of the area that is visible from the current position of the robot and discard the non-visible areas. Though this approach reduces the required number of computations, it sacrifices the ability to globally localize the robot, i.e. when its initial position is unknown.

2.3 Rule Based Sensor Fusion

The methods previously reported, rely on well defined uncertainty models, and execute variants of maximum likelihood or maximum a posteriori estimation. Such models are difficult to develop when the navigation depends on patterns and signatures on the map (e.g. landmarks). Often they are also difficult to obtain for data that are combined from sensors that use significantly different principles and processing techniques. When uncertainty models are either unavailable or are meaningless, ad hoc techniques have often been used within the domain of sensory data. In almost all cases, the advantage of a sound formal basis is lost, though several attempts were made to provide alternative "unified" frameworks in these situations. It is instructive to point out that in spite of their seemingly different procedures, most of these frameworks still execute the same prediction, observation, validation, and updating cycle used by the Kalman filter.

Rule-Based Sensor Fusion: To avoid the difficulty in modeling sensor readings under a unified statistical model, several robotic applications use rule-based algorithms. Often these algorithms do not require an explicit analytical model of the environment. Expert knowledge of the sensors' characteristics and prior knowledge about the environment are used in the design of feature extraction, mapmaking, and self localization techniques. The resulting rules, after some experimentation, are usually simple and robust. The obvious disadvantage is limited domain of applicability - the insights to create rules for a specific environment cannot be easily exported to other en-

vironments. Changes in the sensor suite and in the environment may require re-evaluation and re-compilation of the rule set. In [22], Flynn developed a simple set of heuristic rules for a robot that uses an ultrasonic sensor and a near-infrared proximity sensor. Ultrasonic sensors have wide angles but give relative accurate range measurements. Infrared sensors have excellent angular resolution but poor distance measurements - they are therefore well-suited for detection of edges and large depth discontinuities such as doorways. Fusion of measurements from both ultrasonics and infrareds has the potential to provide robots with good readings in both range and angular position. A set of rules has been used by Flynn to validate the data from the sensors and to determine which sensor to rely upon when conflicting information is obtained. Using these simple rules, the original sonar boundary was redrawn to take into account features that were found by the near-infrared sensor. Significant improvement in the mapping of the laboratory environment was demonstrated in [22].

2.4 Multiple Hypothesis Testing

2.4.1 Previous Approaches

To the best of our knowledge the only applications of a Multiple Hypothesis approach in mobile robots are presented in [13] and [41]. In both cases the *Multiple Hypotheses Testing* framework is employed in order to deduce the origin of sonar data. Two different formulations of a Kalman filter are involved. One tuned to reflections from a corner and the second tuned to reflections from a planar surface. In the focus of both these approaches is the data association problem [38]. A different Kalman filter (one of the two types mentioned) is used for each new feature. As a result, the number of potential hypotheses is growing fast in time and there is a need to provide active and effective management of candidate hypotheses in order to limit the computation involved in developing identity and location estimates. Thresholding of the innovation (residual) is used to decide if a measurement originated from a certain target. The selection of the threshold is add hoc. Another drawback is that odometric information is not considered at all. At the same time, the location and orientation of the vehicle is supposed to be precisely known

at any time. The procedure involved to construct the hypothesis matrix required in this algorithm is based on the assumption that each measurement has only one source and each geometric feature has at most one associated measurement per scan. In other words, geometric features that produce more than one measurement in a single data set are excluded. This causes loss of useful information. The main limitation of this approach is that by applying multiple hypothesis estimation on low level data, such as single scans from sonar sensors, limits its application to the case of corners and flat surfaces only. It is not straightforward how other types of sensors or features can be incorporated in the same framework.

2.4.2 Proposed Extensions to Mobile Robot Localization

Following a different trail of thought, we intend to reformulate the *Multiple Hypotheses* approach to develop a robust and reliable framework for mobile robot localization. In this framework, the data to be combined are: (i) *relative position estimates* as these result from a Kalman filter that optimally fuses the information from the proprioceptive sensors to keep track of the position of the robot and (ii) *absolute position measurements* that result by processing the raw exteroceptive sensor data in a variety of feature extracting modules and using previous knowledge for the environment stored into maps of the area. Consequently, our method is **modular** and **expandable** to any number and type of sensors and features. This adds flexibility to the system since different robotic systems are equipped with different sets of sensors and different types of environments are populated by different types of features.

Another advantage of the proposed method is that it can deal with cases of landmarks that look similar. Each possible landmark initiates a new hypothesis. The number of hypotheses is kept small by incorporating the odometric information in the decision process. Relative position information is sufficient to resolve the ambiguity about the identity of a landmark when sufficient time is allowed. The proposed *Multiple Hypotheses Tracking* algorithm produces estimates of the position of the robot at all times starting from an initial uniform position distribution. Each of the hypotheses created contains a separate **history of the topological transitions** of the robot

through different landmark sites.

One of the drawbacks of the Kalman filter estimation is that it can only represent unimodal distributions approximated by a single Gaussian. The proposed Multiple Hypotheses Tracking framework is capable of sustaining more than one hypotheses for the position of the robot. Each of these hypotheses is represented as a Gaussian distribution. Thus, **multi-modal** distributions can be represented and approximated by an arbitrary collection of Gaussians.[4] Another advantage of the proposed method is that for any number of hypotheses, **only one Kalman filter** is required to propagate the position estimate and update it when a new landmark appears.

Our approach can rely solely on the detection of structural features of the environment such as doors, windows, corners, poles, corridors etc. Compared to the grid-based approaches it is **robust to dynamic environments** since it does not depend on a detailed map of the area that should include objects of small size that usually are part of the ever changing clutter. Instead we prefer to use a coarse scale map composed of reliably detectable features and their position with respect to some global frame.

Kalman filtering is more accurate when sufficient sensor information is available from the sensors. On the other hand Markov localization (that uses Bayesian Estimation) is more robust and it is capable of localizing a robot in an arbitrary probabilistic configuration (when the initial position of the robot is not known). Kalman filtering uses a more compact representation for the uncertainty of the position of the robot and thus the propagation of this uncertainty is very efficient. In Section 3.2 we propose a method that **combines the robustness of the Markov localization and the efficiency and accuracy of Kalman filtering**. In Section 3.3 instead of many two only hypotheses are combined within the *Kalman Smoothing* framework and finally in Section 4.1.3 the multiple hypotheses approach is formulated as a *Multiple Model Adaptive Estimation* solution applied to the fault detection and identification problem.

[4] In the limit, **any distribution** can be approximated by a collection of Gaussians. The quality of the approximation depends mainly on the number of the Gaussians used for the approximation.

Chapter 3

Statement of the Problem, Proposed Methodology and Initial Results

In order to navigate effectively, a robot must be able to quickly and accurately determine its location. Fairly accurate position estimates can be obtained by integrating kinetic information from the robot's proprioceptive sensors. The error accumulation in these estimates when traveling over long distances can lead to unacceptable performance. An effective way of observing the surroundings when the robot is uncertain of its position estimates, is by focusing on landmarks. A landmark is a feature of the environment that the robot's exteroceptive sensors are capable of detecting. When a landmark is being sensed the robot can estimate its own position[1] by invoking a map that contains the landmark absolute positions. The common assumption is that the world is populated with distinct landmarks that can be sensed at all times. Finally, it is almost always the case that the sensors on-board of a robot are considered to be fault free.

[1] A generalization of this would be that stellar objects (e.g. the sun or stars [14]), the magnetic pole of the earth, the gravitational center of the earth, the horizon, or objects in the horizon (e.g. mountain peaks) can be used as landmarks that provide attitude information.

3.1 Precise Statement of the Problem

Here we propose to study the case of a mobile robot that carries a variety of proprioceptive sensors that monitor the motion of the vehicle. These sensors can be wheel-shaft encoders, accelerometers, gyroscopes, optical flow odometric systems, Doppler radars, kinesthetic sensors etc. These devices are commonly found as parts of dead-reckoning systems where the position of the robot is tracked by integrating the kinetic information. The signals from these sensors contain components of noise. In our case, the assumption made is that the noise is Gaussian but it does not have to be zero-mean white. Shaping filters are implemented to estimate on-line possible biases.

The robot is also equipped with exteroceptive sensors that monitor the environment for features. These sensors usually are sonars, laser scanners, cameras (in singe or stereo formation), compasses, sun sensors, star trackers etc. There are two kinds of features that we plan to exploit for localization purposes and each of them provides a different level of localization information.

- *Landmarks for position estimation:* In this case, the exteroceptive sensors first provide their data to the feature extracting module(s). Depending on the identification capabilities of the feature recognition algorithm, one or more possible matches with landmarks represented on an a priori known map are returned along with their probabilities. The positions of the candidate landmarks are processed within the proposed *Multiple Hypothesis Tracking* framework. Most of the time, the resulting position estimate for the robot is a collection of Gaussians produced using a Kalman filter that tracks the position of the robot through a number of possible paths each of them containing a different set of landmarks. The assumption for perfect landmark identification is raised.

- *Features for attitude estimation:* In this case, the exteroceptive sensors extract orientation information from the environment and this is fed directly into the Kalman filter that tracks the position of the robot. In many cases, the attitude information is only available intermittently. The objects that are being used as orientation features may not be in view or their signals can be occluded or distorted for long periods of

time.[2] When the absolute orientation is not known the robot has to rely on integrating the rotational velocity of the vehicle. The lack of observability for long periods of time is the main reason for introducing *Smoother based Attitude Estimation*. The assumption of continuous attitude measurements is removed.

A direct extension of smoothing of the attitude estimates is smoothing of the position estimates. One of the objectives of the proposed study is to apply smoother based estimation in cases where the absolute position of the robot is only known intermittently. This will be beneficial for mapping tasks in partially known environments. When the position of the robot is given for certain locations only, the smoothing of the in-between position estimates will increase the accuracy of the mapping process.

At this point we will present initial results that we have obtained until now with respect to the proposed course of study. In Section 3.2, we introduce the *Multiple Hypotheses Tracking* approach and we formulate this for the case of an environment populated with a single type of features. Preliminary results are shown as this methodology is applied in the case of a robot moving within the corridors of an office environment. In Section 3.3 we present the *Smoother based Attitude Estimation* for the 2-D case. The formulation of the required Indirect Kalman filter is examined in detail. The validity of the theoretical results is demonstrated in simulation using the noise model of the gyroscope that is currently on-board Rocky 7, one of the experimental rovers for the Mars exploration program, currently under development at the NASA Jet Propulsion Laboratory (JPL).

[2]For example, a compass is very sensitive to the presence of metallic objects in the vicinity.

3.2 Bayesian Estimation and Kalman Filtering: Multiple Hypothesis Tracking

In this section we present a general architecture that combines the benefits of both Kalman filtering and Bayesian estimation. As shown in Figure 3.1, *Maps* of the environment are available to the robot.[3] These maps contain a compact representation of the environment in terms of landmarks. The landmarks can be any features of the environment detectable by the robot's exteroceptive sensors. For example, they can be doors in a hallway detected by a camera that looks at the color differences on the walls. In an outdoors scenario, the trunk of a tree detected using a sonar sensor is a possible feature. The maps contain the correspondences between types of landmarks and possible locations. In the case of the 2-D office environment depicted in Figure 3.3a, the feature *door* corresponds to locations $(5,4)$, $(9,4)$, $(13,4)$, $(7,6)$, $(11,6)$, $(4,10)$, or $(14,10)$. Each of these locations is represented using the coordinates of the doors number 1 to number 7 with respect to the coordinate frame with origin the bottom left corner of the floor.

The *Feature Extracting* module can be any collection of data processing algorithms that extract features using data from the exteroceptive sensors. As described in the algorithm presented in Figure 3.2, when a new feature is detected, the feature extracting module identifies its type and then consults the maps of the environment in order to generate a new set of potential hypotheses. These new hypotheses are presented to the *Bayesian Estimation* module where they are combined with the previously existing hypotheses as these are updated using the displacement information provided by the *Kalman Filter*. The resulting new hypotheses are then stored in the *Hypotheses Database*. While the hypotheses fusion takes place only when a new feature is detected by the exteroceptive sensors, the Kalman filter is fusing the proprioceptive sensor information at all times. By tracking the position displacement continuously, each of the hypotheses is updated continuously as well.

[3] For now, we will assume that these maps have been composed by an entity outside the robot before it visits the area of concern. In the future, we intend to extend the proposed methodology to the case where the map was constructed by this robot or a similar one, at some previous time, following a set of prespecified rules.

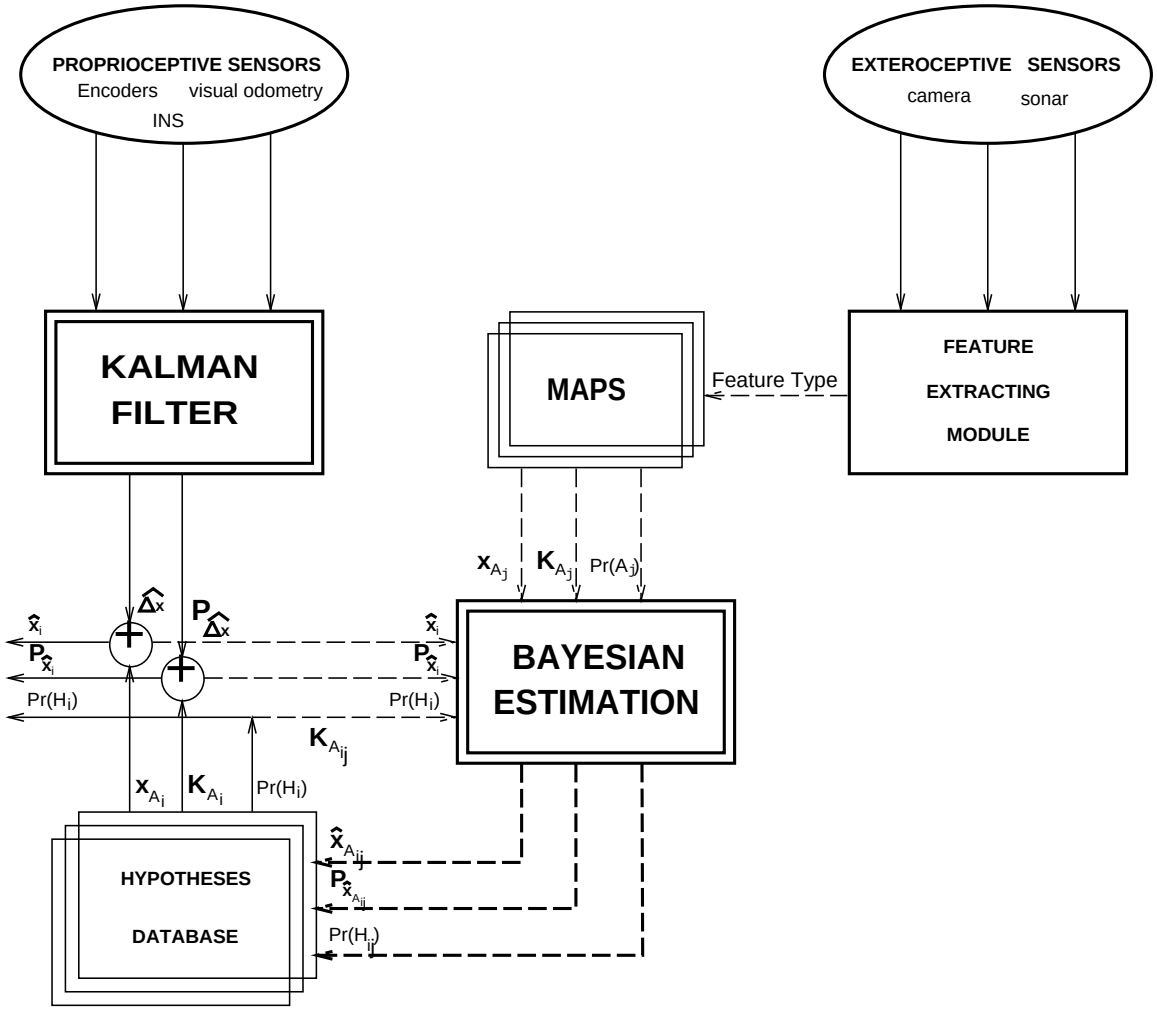

Figure 3.1: *Multiple Hypotheses Tracking Architecture*

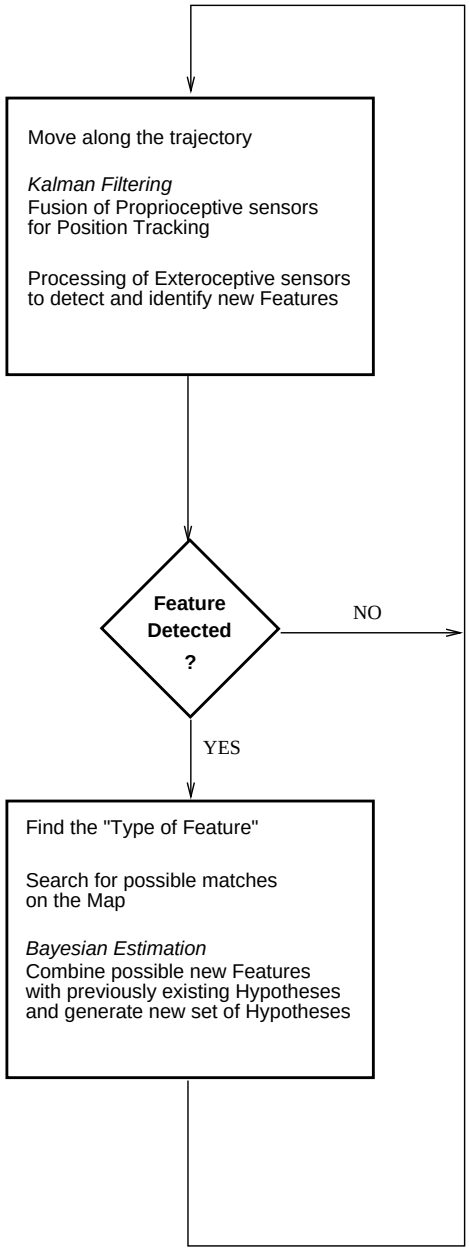

Figure 3.2: *Multiple Hypotheses Tracking Algorithm*

3.2.1 Absolute Localization: The "kidnapped robot" Problem, Position Tracking, and Landmark Detection

At this point we will define the *"kidnapped robot"* problem. A robot suddenly finds itself in an unknown location as if it has been kidnapped and dropped there. In order to proceed to its goal destination, it has to identify its new position. Maps of the environment are available to the robot. The challenge is to design a localization schema that will enable the robot to determine its location in the map by searching around and exploiting its sensing capabilities. The ability of the robot to recognize its location is very important because it allows a robot to recover from an accident and thus increases its reliability. For example, if a robot while moving along its path is hit by something and pushed to a new location, it should be able to determine its new position and continue with its tasks. In addition, the first time that a robot is brought into a new area, it is desirable that it finds its initial position without external help.

The absolute localization problem can be defined as threefold:

1. The robot determines its initial position ("kidnapped robot" problem),
2. The robot continuously tracks its position, and
3. The robot updates its deteriorating position tracking estimates by detecting new features in the environment.

Before presenting the proposed approach to the general localization problem, we first study the ideal case of a robot equipped with exteroceptive sensors capable of distinguishing between almost identical landmarks and restricted to motion in a certain area. Though this scenario is fictional, it will ease the transition to the more realistic case of imperfect landmark recognition and imperfect odometry.

Perfect Landmark Recognition, Imperfect Odometry

In the absence of any previous knowledge, the robot's initial position distribution can be assumed to be uniform and thus the probability density function (pdf) is:

$$f_0(\mathbf{x}) = \frac{1}{S}\,, \tag{3.1}$$

where S is the surface of the area that the robot is allowed to move in and $\mathbf{x} = [x, y]^T$ is the position of the robot. In the case of perfect landmark recognition, there is no ambiguity. The decision is binary for each of the possibilities encountered:

$$P(z_k = A_i) = 1\,,\ P(z_k = A_j) = 0\,,\ j = 1, ..., N,\ j \neq i\,, \tag{3.2}$$

where A_i is the encountered landmark and z_k is the kth measurement provided from the exteroceptive sensors.[4] Every time the robot detects and identifies a landmark whose exact position $\mathbf{x_{A_i}}$ is known, the updated pdf becomes:

$$f(\mathbf{x}/\mathbf{z_k}) = f(\mathbf{x}/z_k = A_i) = \delta(\mathbf{x} - \mathbf{x_{A_i}})\,, \tag{3.3}$$

where δ is the Dirac delta, or, in the case that the position of the landmark is known to follow a Gaussian distribution with mean value $\mathbf{x_{A_i}}$ and covariance $\mathbf{K_{A_i}}$:[5]

$$f(\mathbf{x}/\mathbf{z_k}) = f(\mathbf{x}/z_k = A_i) =$$

$$\frac{1}{(2\pi)^{n/2} det(\mathbf{K_{A_i}})^{1/2}} \exp[-\frac{1}{2}(\mathbf{x} - \mathbf{x_{A_i}})^T \mathbf{K_{A_i}}^{-1}(\mathbf{x} - \mathbf{x_{A_i}})] \tag{3.4}$$

where

$$\mathbf{K_{A_i}} = E[(\mathbf{x} - \mathbf{x_{A_i}})\ (\mathbf{x} - \mathbf{x_{A_i}})^T] \tag{3.5}$$

The Gaussian assumption for the distribution of the position of the landmark is often invoked to represent the spatial uncertainty of the measurements provided from the exteroceptive sensing devices. The sensors used to locate the position of a landmark have limited precision and their signals are in general noisy. Therefore, the calculation of the position of the landmark with respect to the current position of the robot has its own inaccuracies. A Gaussian model can be used to describe the uncertainty of the calculated transformation from the egocentric frame to the landmark centered frame.

Every time a landmark is left behind, the robot knows its location precisely. The pdf of its position is a Gaussian function of its coordinates in the

[4] Actually z_k is the result of processing the data from the exteroceptive sensors within the feature extracting module.

[5] For the 2D case $\mathbf{K_{A_i}} = [\sigma^2_{x_{A_i}}\ 0\ ;\ 0\ \sigma^2_{y_{A_i}}]$

area. This distribution is **unimodal** since we have assumed a 1-to-1 match between encountered landmarks and locations on the map. From then on the robot has to rely on its odometry until it encounters another landmark. In the absence of any external positioning information, the uncertainty of the position estimate will increase continuously as the robot moves. The system is not observable[6] and thus any estimation technique will eventually fail. The accuracy of the position estimate based on the imprecise odometry will decrease and it will become untrustworthy. The rate of deterioration depends on the number and type of sensors used as well as their noise characteristics. Any form of absolute orientation measurement will reduce the uncertainty while low levels of noise will reduce the rate of increase of the variance of the position estimate. Substantial improvement is possible by applying Kalman filtering techniques [26], [43]. These techniques have been used successfully in position estimation problems such as missile tracking and ship navigation for the last four decades [3] and their field of application has been extended to mobile robots. Relying on a simple kinematic model of the robot [52] or on sensor modeling [53] a Kalman filter can be formulated to estimate the displacement of the robot

$$\widehat{\mathbf{\Delta X}} = [\widehat{\Delta x} \quad \widehat{\Delta y}]^T \tag{3.6}$$

and the associated covariance:

$$\mathbf{P}_{\widehat{\mathbf{\Delta x}}} = E[\widehat{\mathbf{\Delta X}}\ \widehat{\mathbf{\Delta X}}^T] \tag{3.7}$$

where the " ^ " is used to indicate estimated values. The robot will have to drive towards the position of another known landmark in order to reduce its uncertainty. When it does so, the Kalman filter framework is suitable for fusing the new landmark information (measurement $\mathbf{x_{B_j}}$) with the current position estimate $\hat{\mathbf{x}}_{\mathbf{robot}} = \mathbf{x_{A_i}} + \widehat{\mathbf{\Delta X}}$ as this is calculated based on the previous landmark information $\mathbf{x_{A_i}}$ and the processed odometric data $\widehat{\mathbf{\Delta X}}$.

[6]Observability is concerned with the effect of states $\mathbf{x}(t)$ of a model upon outputs $\mathbf{z}(t)$. A system model is observable if *any* state $x_i(t)$ can be determined exactly for any time t from knowledge of only previous (i) inputs $\mathbf{u}(t)$ and (ii) outputs $\mathbf{z}(t)$. To be completely observable, the representation structure must be such that the output $\mathbf{z}$(t) is affected in some manner by the change of any single state variable. Moreover, the effect of any one state variable on the output must be distinguishable from the effect of any other state variable.

Examples of treatment of similar cases can be found in the related literature [40], [5].

Environments with low landmark density are more difficult to navigate in and require that the path planing module consider the locations of the landmarks when computing a path in order to facilitate the localization process. It is very possible that a trade off would occur between the total length of the path between two locations and the required level of localization accuracy which depends on the frequency of landmarks within the path.

The robot relies on the position estimates when it follows a commanded path between an initial and a final position. Failure of the localization module will make it impossible for the robot to find its way in its environment. It will have to wander either randomly or by following some search strategy, until it arrives at the desired location. This is possible only if the goal is in the vicinity of a known landmark.

Within this scenario of very precise landmark recognition where odometry suffers from inaccuracies there are some interesting issues related to fault detection. The absence of a landmark where it was expected to be (within some limits due to uncertainties of the odometry) or the appearance of a landmark in a "wrong" position can reveal failures of the localization module.

Imperfect Landmark Recognition, Imperfect Odometry

This section deals with a more realistic and challenging scenario. We now relax the assumption that each landmark is uniquely identifiable. In most of the current approaches where both these sources of information are considered, either the odometry (as in [13]) or the landmark recognition module (as in [40] or [59]) is assumed perfect. Many times the landmark recognition module is capable of recognizing certain types of landmarks but it cannot distinguish between members of the same type. For example, it can distinguish a door from a corner but it is not capable of identifying which door on the map it is facing now, or around which corner it is about to turn. The inability of the exteroceptive sensing to determine reliably the identity of a landmark can cost the robot its ability to globally localize itself in an environment represented on a given map. To make this more clear, consider the simple case where the position calculated using the odometry is accurate to a few meters on each direction of motion and the map contains two similar landmarks within this area of a few square meters. The appearance of these

two landmarks will have no effect on the accuracy of the localization. There is no extra confidence that the encountered landmark is one or another. The uncertainty remains the same as before, being able only to categorize the appeared landmark to be of a certain type.

As we will see in Section 4.1.1, similar problems occur when the landmark recognition module is capable of identifying possible types of landmarks along with the associated probabilities that portray the level of confidence. For example a landmark recognition system based on vision that looks for circular or square patches in the environment has a certain hit ratio when categorizing its visual cues. The information provided does not have to be either "square" or "disk". It can be "square" with probability 70% and "disk" with probability 30%. Although a decision about the robot's location cannot be made precisely because the new information is inconclusive, instead of disregarding it, it is better to "store" it and use it later.

A framework that is capable of incorporating uncertainty in the landmark recognition as well as in the odometry is presented here. The seemingly untrustworthy information from the landmark recognition module is compiled in a number of different hypotheses. Each hypothesis is based on certain assumptions for the current location of the robot as well as the possible identity of the encountered landmark and has a level of confidence assigned to it. The uncertainty is portrayed as a probability $P(H_i)$ associated with the particular hypothesis H_i. In the algorithm presented here, both the total position pdf and the probabilities for each of the hypotheses are adjusted continuously while the robot is in motion.

As mentioned before, a Kalman filter can optimally fuse information from a variety of kinetic sensors on-board a robot, estimate the position displacement $\widehat{\mathbf{\Delta x}}$, and provide the level of belief to this estimate, i.e. the covariance $\mathbf{P}_{\widehat{\mathbf{\Delta x}}}$. It can also optimally combine the positioning information (from a possible landmark match) with the position estimate calculated using the odometric information. The uncertainty in the landmark recognition is the main reason for introducing the multiple hypothesis approach. Multiple hypotheses testing for target tracking was first presented in [51], and applied to mobile robot localization by [13]. The main drawback of this application was that a Kalman filter was required for every hypothesis encountered. With

the number of hypotheses growing rapidly,[7] this methodology could not be used in real time. Another major drawback was that the belief levels for each hypothesis were updated only when a landmark was encountered. Though it is less obvious, as we will see in Section 4.1.2 the absence of a landmark also contains information which can eliminate a weak hypothesis and thus keep their number considerably smaller. It is also worth mentioning that in [13], the authors assume perfect odometry while our algorithm takes under consideration the odometric uncertainties.

In the approach presented here, only one Kalman filter is used regardless of the number of existing hypotheses. This filter calculates the quantities in Equations (3.6) and (3.7) which are the same for each hypothesis and thus reduces considerably the computations required. The confidence levels of the hypotheses change when a new landmark appears. New hypotheses can appear and old ones can vanish. The competition between the hypotheses can continue even when there is no landmark in sight. The absence of a landmark carries enough information to terminate some of the hypotheses and thus strengthen others (the total probability of all the hypotheses adds up to one at any given time). The Kalman filter is based on a unimodal distribution and thus it cannot directly be applied to the multiple hypotheses case. The Bayesian formulation of the problem dictates the competition between the possible localization scenarios and allows for a **multi-modal** pdf. Hereafter we describe the proposed algorithm in more detail for the case of an environment with one type of features.

Step 1*[Kidnapped robot localization]* First the robot finds itself at an unspecified location somewhere in the mapped area. This is the kidnapped robot localization problem. The assumption is that the initial pdf is uniform across the space as in Equation (3.1). The robot starts to move and after a while, it encounters the first landmark A_i. The feature extraction module assigns probabilities $P(z_1 = A_i) = 1/N, i = 1..N$ to the different choices,[8] where N is the number of possible matches on the map. Due to the uncertainty related to the landmark recognition module, the hypotheses assuming that different landmarks have been detected are not mutually exclusive. Each

[7] Every time a new set of measurements is received, there is a new hypothesis for almost every new piece of data. That is, for a single scan of a sonar range finder, tens of new hypotheses are created.

[8] For example, if there exist seven similar doors in the area then the $P(z_1 = A_i) = 1/7$ for each door location.

of the probabilities associated with a different landmark A_i receives non zero values. Therefore the new pdf is:

$$f(\mathbf{x}/z_1) = \sum_i P(z_1 = A_i)\ f(\mathbf{x}/z_1 = A_i) \tag{3.8}$$

where $i = 1..N$, $\mathbf{x} = [x, y]^T$, and z_1 is the first feature sensed by the exteroceptive sensors. Here the pdf's $f(\mathbf{x}/z_1 = A_i)$ are the same ones as in Equation (3.4) that describe the distribution of the position for each of the known landmarks. Each of the non zero probabilities defines a new hypothesis H_i with assigned probability $P(H_i) = P(z_1 = A_i)$, that considers the position of the robot to follow a different Gaussian distribution: $\mathbf{x} \sim \mathbf{N}(\mathbf{x_{A_i}}, \mathbf{K_{A_i}}), i = 1..N$, where $\mathbf{K_{A_i}}$ is the covariance given in Equation (3.5). The weighted average of all the pdf's linked to one of the existing hypotheses is the total pdf given in Equation (3.8) and repeated here for the general case of the $(k-1)th$ landmark:

$$f(\mathbf{x}/\mathbf{z_{k-1}}) = \sum_i P(H_i)\ f(\mathbf{x}/H_i) \tag{3.9}$$

Step L-1*[Kalman filtering]* As the robot leaves the previous landmark $(k-1)$, it has to rely on its odometry to track its position. Only *one* Kalman filter is required to process the information from the odometric sensors and calculate the quantities in Equations (3.6) and (3.7).[9] The updated position estimate for the robot at each time t is given by the following set of equations:

$$\hat{\mathbf{x}}_{\mathbf{robot}}(t, i) = \mathbf{x_{A_i}} + \widehat{\mathbf{\Delta x}}(t)\,, i = 1..N \tag{3.10}$$

Each equation corresponds to a hypothesis H_i with probability $P(H_i)$. $\mathbf{x_{A_i}}$ is the position of the last visited landmark and $\widehat{\mathbf{\Delta x}}(t)$ is the position displacement estimated in the Kalman filter. The position covariance related to each hypothesis H_i is calculated by adding the covariance of the position of the corresponding landmark A_i to the covariance of the displacement $\widehat{\mathbf{\Delta x}}(t)$ which is computed at each time step in the Kalman filter:

$$\mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}}(t, i) = \mathbf{K_{A_i}} + \mathbf{P}_{\widehat{\mathbf{\Delta x}}}(t) \tag{3.11}$$

[9]The formulation of the Kalman filter is omitted due to space limitations. The interested reader is referred to [45] and [43] for a detailed description of the propagation and update equations. Some simple implementations can be found in [34] and [17]. An example of an Extended Kalman filter applied to the case of motion on a planar surface is presented in Appendix A.

As the robot travels in-between landmarks the uncertainty of its position displacement grows. Thus, the position covariance for each hypothesis is constantly increasing. The new total pdf at time t is the weighted average of N pdf's each described by a Gaussian $\mathbf{N}(\hat{\mathbf{x}}_{\mathbf{robot}}(t,i), \mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}}(t,i))$:

$$f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}(t)) = \sum_i P(H_i)\ f(\mathbf{x}/H_i) =$$

$$\sum_i P(H_i)\ \frac{1}{(2\pi)^{n/2} det(\mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}}(t,i))^{1/2}}\ \times$$

$$\exp[-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{\mathbf{robot}}(t,i))^T \mathbf{P}^{-\mathbf{1}}_{\hat{\mathbf{x}}_{\mathbf{robot}}}(t,i)(\mathbf{x} - \hat{\mathbf{x}}_{\mathbf{robot}}(t,i))] \qquad (3.12)$$

in accordance with Equation (3.9).

Step L*[Bayesian estimation]* After the robot encounters the *kth* landmark, the distribution of its position is described by the following pdf:

$$f(\mathbf{x}/\mathbf{z_k}) = f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k) =$$

$$\sum_{j=1}^{N} f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(z_k = B_j) =$$

$$\sum_{j=1}^{N} P(z_k = B_j) \sum_{i=1}^{N} f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) =$$

$$\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) \qquad (3.13)$$

where N is the number of hypotheses in the previous step and the number of possible matches on the map for the *kth* landmark. $P(z_k = B_j)$ is the probability that the *kth* landmark is B_j. The first quantity in Equation (3.13) is the updated pdf for the position estimate after taking into consideration the new *kth* landmark. This quantity is calculated in the Kalman filter as the

latest estimates (Equations (3.10) and (3.11)) are combined with the new information for the position of the new landmark $B_j, j = 1..N$. Each *new* hypothesis H_j assumes that the position of B_j follows a Gaussian distribution with pdf as in Equation (3.4). The fusion of the latest Kalman filter estimates $\hat{\mathbf{x}}_{\mathbf{robot}}(t,i)$ and $\mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}}(t,i)$ with the position information $\mathbf{x}_{B_j}$, $\mathbf{K}_{B_j}$ for each of the possible landmarks B_j is a *one step* process and thus it can be implemented in parallel. The resulting $f(\mathbf{x}/\mathbf{z_k})$ is a multi-modal Gaussian distribution composed of 1 to $N \times N$ functionals. The second quantity in Equation (3.13) is the result of the Bayesian estimation step (Multiple Hypothesis Testing) and is calculated for each hypothesis H_i as follows:

$$P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) =$$

$$\frac{P(z_k = B_j/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})}{\sum_{i=1}^{N} P(z_k = B_j/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})} \tag{3.14}$$

where $P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})$ is the a priori probability for each hypothesis H_i available from the previous step and $P(z_k = B_j/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)$ is the probability to detect landmark B_j given that the current location of the robot is $\hat{\mathbf{x}}_{\mathbf{robot}}(t,i)$:

$$P(z_k = B_j/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i) = \frac{1}{(2\pi)^{n/2} det(\mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}} + \mathbf{K}_{B_j})^{1/2}} \times$$

$$\exp[-\frac{1}{2}(\mathbf{x}_{B_j} - \hat{\mathbf{x}}_{\mathbf{robot}}(t,i))^T (\mathbf{P}_{\hat{\mathbf{x}}_{\mathbf{robot}}} + \mathbf{K}_{B_j})^{-1} (\mathbf{x}_{B_j} - \hat{\mathbf{x}}_{\mathbf{robot}}(t,i))] \tag{3.15}$$

for $j = 1..N$. Each of the N previously existing hypotheses H_i are now tested in light of the new evidence. The hypotheses whose assumption for the estimated position of the robot (given in Equation (3.10)) is closer to the location of any of the new possibly encountered landmarks $B_j, j = 1..N$ will be strengthened. The rest will be weakened. The total probability calculated by summing the probabilities in Equation (3.13) equals to 1. Due to the additional constraints imposed from the information associated with the position tracking, a few only hypotheses will sustain probabilities over a preset threshold and continue to the next round, i.e. when the next landmark is encountered.

After a sufficient number of repetitions of **Step L-1** and **Step L** of this algorithm the robot will be left with one only choice for possible location. This is demonstrated in detail in the next section.

3.2.2 Experimental Results

In this section we demonstrate the efficiency of the presented algorithm in the test case of an office environment shown in Figure 3.3a. The position of the robot is marked with a small triangle.

In order to support dead-reckoning based position estimation, the vehicle carries shaft encoders and a gyroscope. These signals are fed to a Kalman filter implemented for this set of sensors. The computed position displacement estimate $\widehat{\mathbf{\Delta x}}(t)$ is available to the localization module at any given time. In addition the robot is equipped with exteroceptive sensors. These can be either cameras capable of detecting color or sonars that measure distances from obstacles. A feature extraction module capable of detecting doors is fed with these signals. This can be done by either checking the color contrast on the sides of the corridor or by monitoring the distance from the wall. Although a door can be detected, it can not be identified. There are seven doors facing this **U** shaped corridor. A number is assigned to each of them. The robot has stored a map of the floor containing the positions of each of the doors in some coordinate frame.

In the beginning, the robot is brought and set at an arbitrary location in the corridor. In the absence of any initial positioning information, the localization module assumes a uniform position pdf along the corridor and sets the robot in motion until the first landmark is in sight. When the robot encounters the first door on its right, say number 1, it does not know which door this is. There are in fact 7 equally likely hypotheses of where it is. These are depicted in Figure 3.3a. The pdf's associated with each of these hypotheses are fairly sharp and are shown in Figure 3.3b.

As the robot begins to move again following the wall on its right, **Step L** is applied. That is, the new information from the odometric sensors is processed in the Kalman filter and an estimate of its position is computed along with the uncertainty associated with it. After the robot has covered 2 meters the position uncertainty has increased and thus each pdf related to each of the 7 hypotheses is spread around a larger area as shown in Figure 3.3c. As the robot covers another 2 meters this phenomenon becomes more prominent. This can be seen in Figure 3.3d.

After the robot has covered a total of 4 meters, it detects another door on its right (number 2). Application of **Step L** of the algorithm immediately reduces the number of hypotheses to 3. Out of the 7 previous hypotheses only

3 are possible. These are the ones that assumed that the previous landmark was door number 1, door number 2, or door number 5. For these hypotheses there is a new door on the right side of the robot after the vehicle has moved for 4 meters. In Figure 3.4a we see again that the precise localization information related to a potential landmark makes the pdf's for each of the surviving hypotheses sharp again.

The robot has to travel a longer distance before it is able to determine its position uniquely. **Step L-1** *[Kalman filtering]* is applied again and thus the position estimates related to the 3 hypotheses deteriorate. As the robot moves for another $2 + 2$ meters, the uncertainty increase is shown in Figures 3.4b and 3.4c. Finally the robot senses a third door on its right (number 3) and by applying **Step L** of the algorithm again, it concludes with the position estimate described by the pdf of Figure 3.4d.

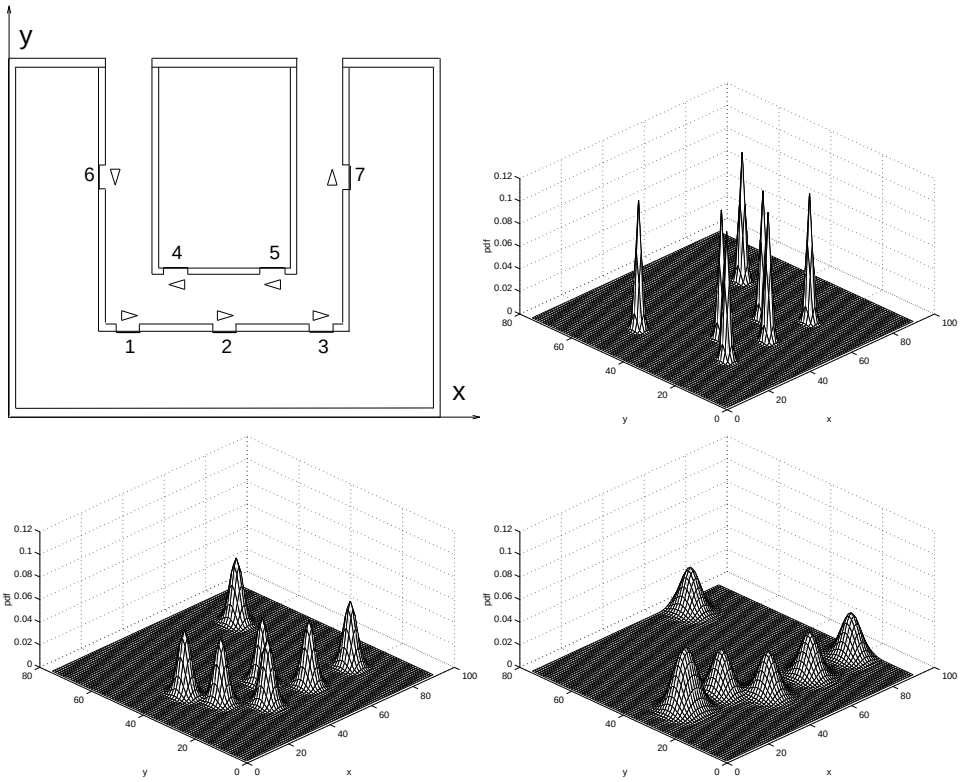

Figure 3.3: a. *The environment of the robot, b. The initial pdf for the position of the robot when it senses one door on its right, c. The pdf after the robot moves for 2 meters, d. The pdf after the robot moves for another 2 meters and before sensing the second door on its right. Notice that 6 instead of 7 Gaussians are shown in this figure. The explanation is that 2 of the 7 hypotheses point to the same location and thus their pdf's are combined to a single Gaussian which is taller that any of the other 5. The units on these plots are 0.2m/div.*

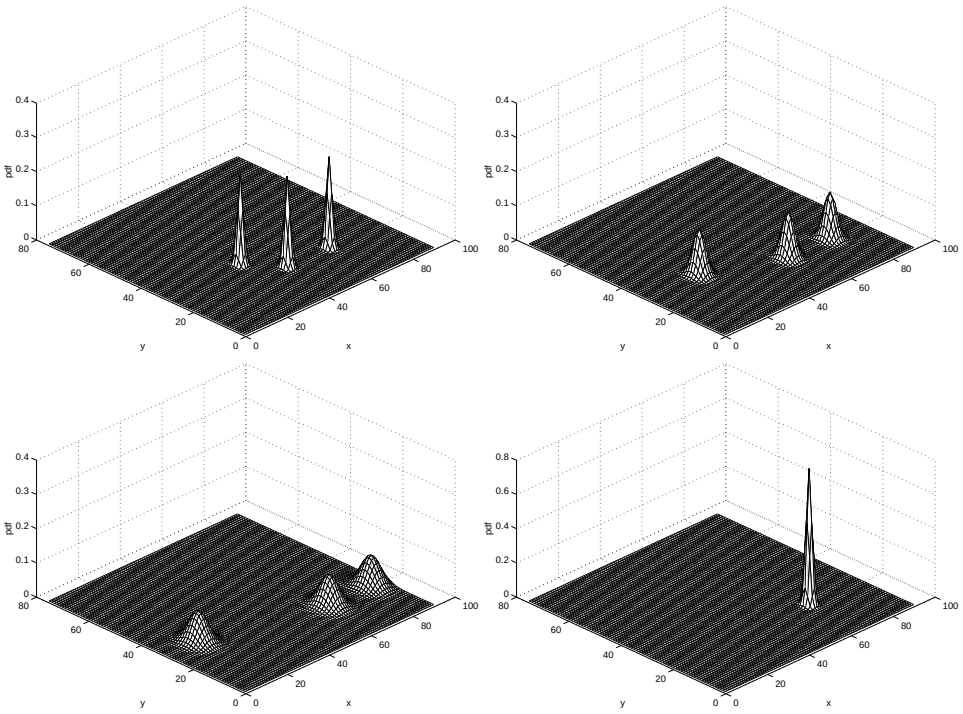

Figure 3.4: a. *The pdf right after the robot has sensed the second door on its right, b. The pdf after the robot moves for another 2 meters, c. The pdf after the robot moves for another 2 meters and before sensing the third door on its right, d. The pdf right after the robot has sensed the second door on its right. The units on these plots are 0.2m/div.*

3.3 Smoother Based Attitude Estimation: 2-D Enhanced Localization

3.3.1 Localization and Attitude estimation

The sensor suite of Rocky 7 includes wheel encoders, gyros, accelerometers and a sun sensor. Since there is no device measuring the absolute position of the rover (GPS signals are not currently available on the surface of Mars), the position can be estimated through the integration of the accelerometer signal and/or the signals from the encoders. The quality of this estimate depends heavily on the noise profile of the inertial and the odometric sensors. Bias and noise in the accelerometer signal and imprecise encoders can cause serious drift of the position estimate. The task at hand is even more difficult because the propagation of the position relies upon the attitude estimate. Even small errors in the orientation can fast become large errors in position. Formally speaking, the position is not observable and thus the uncertainty of the position estimate will grow without bounds. The most promising course of action with the current set of sensors would be to focus on gaining a very precise attitude estimate. As a result the position uncertainty will grow at a slower rate and will allow for increased autonomy. The estimator will maintain accurate position estimates during longer traverses before other localization techniques (such as visual tracking) are required.
The attitude estimate is used twice during position estimation:

i) The accelerometer measures both the vehicle's acceleration and the projection of the gravitational acceleration on the accelerometer local frame. The relation between these is described by the following equation:

$$\ddot{\vec{p}}(t) = \vec{f}(t)/m = \vec{a}_{accelerometer}(t) - A(q(t))\vec{g} \tag{3.16}$$

where $\ddot{\vec{p}}$ is the vehicle's (non-gravitational) acceleration, $\vec{a}_{accelerometer}$ is the measurement from the 3-axis accelerometer and $\vec{g}$ is the gravitational acceleration. Precise knowledge of the orientation matrix $A(q)$ is mandatory to extract $\ddot{\vec{p}}$ accurately.

ii) The next step requires integration of $\ddot{\vec{p}}$ to derive the position. The information of $\ddot{\vec{p}}$ is local (i.e. expressed in the local coordinate frame) and in order to calculate the position in global coordinates the attitude information is once again required:

$$\vec{p}(t) = \int_0^t dt' \int_0^{t'} A(q(t''))\vec{\ddot{p}}(t'')dt'' \tag{3.17}$$

Attitude Measuring Devices

Amongst the sensors on-board the rover, the gyroscopes can be used to calculate the attitude of the vehicle by simply integrating their output signal (the angular rate). On the other hand, the sun sensor measures directly the values of the two components of a two-dimensional vector. This vector is the projection of the unit vector towards the sun on the sun sensor plane. Another sensory input of the same nature is required in order to satisfy attitude observability requirements. It is evident from Equations (3.16) and (3.17) that the accelerometer is used to advance the position estimate. It can also be used in an alternative way. An accelerometer can measure the gravitational acceleration, a three-dimensional vector parallel to the local vertical. This provides another orientation fix independent from the sun and thus makes the vehicle's attitude observable.

This method fails when the rover is in motion. The gravity vector is then "contaminated" by the non-gravitational acceleration of the vehicle seen in Equation (3.16). The gravity vector could be extracted while the vehicle is moving if an independent measurement of its own acceleration was available. Research efforts have tried to address this problem. In [63], the authors attempt to remove the non-gravitational acceleration by approximating it. Odometry information from the encoders is used in this approximation. Each encoder signal has to undergo two differentiations to provide acceleration estimates along the local x and y directions. The main claim reported is that in the long term, the roll and pitch estimates from the accelerometers have good mean values while in the short term the gyroscopes are mostly responsible for the attitude estimation. It is also clear that instantaneous attitude estimates provided by the accelerometer can be completely wrong. As the authors mention, robot vibrations are wrongfully converted into angles and numerical differentiations affect the quality of the result. In their work, a Kalman filter that estimates the drift of the gyroscopes based on the accelerometer information is developed. The algorithm was tested in the case of a mobile robot that moves on flat ground except when one of the wheels climbs over a 2 cm plank using a small inclined plane. After that the robot

again moves on flat ground. We believe that this approach is sufficient for indoor applications and can deal with cases of motion over small objects but it is not accurate enough for outdoor environments, mainly because of the limited accuracy of the estimates of the non-gravitational acceleration. This is due to numerical differentiation, limited precision of encoders, longitudinal wheel slippage and lateral skidding during turning. The approach of Fuke and Krotkov [23] is similar. They concentrate on motion in a straight line over inclined surfaces (sandy craters). Their method suffers from the same drawbacks as the previously discussed method.

A more thorough consideration of the problem would require dynamic modeling of the vehicle. An estimator that incorporates a dynamic model of the vehicle could estimate its non-gravitational accelerations. An example of an effort along these lines comes from the Aerobots domain. In [54], an estimator for a telerobotic balloon is described. While the balloon is floating in the air, the "swaying" of its gondola can be modeled as a pendulum. That allows for modeling of its dynamics. The result is an estimator that uses, in effect, the accelerometer as another source of attitude information.

Dynamic Model Replacement

In our approach we avoid dynamic modeling and we restrict ourselves to using the accelerometer only when the rover is stopped. The reasoning behind this is as follows:

i) The most elementary reason is that every time there is a modification on the rover (i.e. a mass changes, or a part is relocated, or dimensions are altered) the dynamic modeling has to be redone. The produced estimator is tailored for a specific structure. A slightly different vehicle would require a new estimator.

ii) A more practical reason is that dynamic modeling would require a very large number of states (consider that the vehicle has 6 wheels, 2 steering joints, 3 bogey joints). An estimator has to be practical as far as its computational needs are concerned. The size of the estimated state can have large computational demands with very little gain in precision.

iii) Dynamic modeling and the added complexity caused do not always produce the expected results. One example is an attempt by Lefferts and Markley [39] to model the attitude dynamics of the NIMBUS-6 spacecraft,

which indicated that dynamic modeling with elaborate torque models could still not give acceptable attitude determination accuracy. For this reason most attitude estimation applications in the aerospace domain use gyros in a dynamic model replacement mode [65].

iv) Modeling a mobile robot moving on rough terrain is more complicated than that of a spacecraft because of the interaction between the wheeled vehicle and the ground.[10] The external forces on a spacecraft traveling on a ballistic trajectory in space, are precisely described by the law of gravitational forces. The interaction of a rover with the ground depends on many difficult to measure parameters and requires simplifying assumptions to be made (the point contact assumption is frequently used). The modeling of the vehicle-terrain dynamic effects such as wheel impact, wheel slippage, wheel sinkage, all require prior knowledge of the ground parameters (i.e. friction coefficients as a function of wheel slippage ratio, soil shear strength, elastic modulus etc). In addition, the lateral slippage is not observable and can not easily be accounted for in the dynamic model. Precise modeling of the motors is also required to obtain the real values of the torques acting on each of the wheels. At this point we also have to mention that the applicability of Kalman filtering techniques rests on the availability of an accurate dynamic model. An inaccurate model can cause the estimate to drift away from the real value.

v) Looking at the problem from another perspective, the vehicle has a suspension system (bogey joints) that allows it to climb rocks 1.5 times its wheel diameter [32]. One of the purposes of every suspension system is to decouple the vehicle's motion from the terrain morphology (Figure 3.5). The "ideal" suspension system would support a motion of the vehicle that would imitate to some extent, the motion of a hovercraft. The rover then could move in a very smooth fashion. A motion like this is prone to be effectively estimated following methods that rely on the use of inertial navigation systems as a dynamic model replacement, and are independent of the particular terrain.

[10] Another example is that of legged locomotion. Modeling the interaction with the ground is easier for legged locomotion compared to wheeled locomotion. For example, when considering displacements on rugged terrains, perturbations resulting from the systems interaction with the ground are intermittent and discretized in space in the case of legged robots, while their effects on a vehicle are permanent (non-stop) in the case of wheeled locomotion.

As previously mentioned, we use the accelerometer as an attitude sensor, only when the vehicle is stopped. The stopping is justifiable for a variety of reasons. To begin with, the rover is expected to stop to periodically deploy instruments against rocks (e.g. a spectrometer or a drill). In other instances it is expected to stop to save energy, or just to wait for commands from the operator on earth. In every case, speed is not the main requirement and thus the slow motion can include some small intervals of stopping. The gain in accuracy and simplicity are two more reasons to exploit the stop to obtain a better attitude estimate. When the vehicle is stopped the accelerometer measures only the gravitational acceleration:

$$\vec{a}_{accelerometer} = A(q)\vec{g} \tag{3.18}$$

The roll and pitch of the vehicle can be precisely calculated from this equation. The sun sensor then provides the yaw measurement and thus the matrix $A(q)$ is known precisely.

3.3.2 Forms of the Kalman filter

As mentioned before, Kalman filtering has been widely used for localization purposes. In most of the mobile robot applications, the direct (full state) form of either the linear Kalman filter or the Extended (linearized at each point) Kalman filter (EKF) is found. In this work we take advantage of the indirect-feedback formulation of the Kalman filter. The following sections contain the derivations for the equations needed for this. We begin however, with the reasons for selecting this formulation over others.

Indirect versus Direct Kalman filter

A very important aspect of the implementation of a Kalman filter in conjunction with inertial navigation systems (INS) is the use of the indirect instead of the direct form, also referred to as the error state and the total state formulation respectively [43]. As the name indicates, in the total state (direct) formulation, total states such as orientation are amongst the variables in the filter, and the measurements are INS outputs, such as from a gyro, and external source signals. In the error state (indirect) formulation, the *errors* in

orientation are amongst the estimated variables, and each measurement presented to the filter is the difference between the INS and the external source data.

There are some serious drawbacks inherent to the direct realization of the Kalman filter. Being in the INS loop and using the total state representation, the filter has to maintain explicit, accurate awareness of the vehicle's angular motion and at the same time attempt to suppress noisy and erroneous data. Sampled data require a sampling rate of at least twice the highest frequency signal (in practice a factor of 5-10 is used) for adequate reconstruction of the continuous time system behavior. The filter would have to perform all the required computations within a short sampling period. Moreover, in most cases, the estimation algorithm and thus the Kalman filter is allocated only a small portion of the processor's clock cycles. Frequently, it runs in the background at a lower priority than more critical algorithms, such as real-time vision, obstacle avoidance and fault detection.

In addition, the dynamics involved in the total state description of the filter include high frequency components and are well described only by a non-linear model. The development of a Kalman filter is predicated upon an adequate linear system model, and such a total state model does not exist.

Another disadvantage of the direct filter design is that if the filter fails (as by a temporary computer failure) the entire navigation algorithm will fail. The INS is useless without the filter. From the reliability point of view it would be desirable to provide an emergency degraded performance mode in such a case of failure[11]. A direct filter would be ideal for fault detection and identification purposes.

The Indirect (error state) Kalman filter estimates the errors in the navigation and attitude information using the difference between the INS and external sources of data (e.g. GPS, radar). The INS itself is able to follow the high frequency motions of the vehicle very accurately, and there is no need to model these dynamics explicitly in the filter. Instead, the Indirect filter is based on a set of inertial system error propagation equations which are low frequency and adequately represented as linear. Because the filter is out of the INS loop and is based on low frequency dynamics, its sampling rate can be much lower than that of the direct filter. In fact, an effective Indirect

[11] As we will see in the next section, in the event of a failure the Indirect filter is capable of providing estimates by acting as an integrator on the INS data

filter can be developed with a sample period (of the external source) of the order of minutes [43]. This is very practical with respect to the amount of computer resources required. For these reasons, the error state formulation is used in essentially all terrestrial aided inertial navigation systems [55].

Feedforward versus feedback Indirect Kalman filter

The basic difference between the feedforward and feedback Indirect Kalman filter involves how they handle the updated error estimate. In the feedforward case, the updated error estimate is used to correct the current orientation estimate without updating the INS. In the feedback formulation, the correction is actually fed back to the INS to correct its "new" starting point, i.e. the state that the integration for the new time step will start from. In a sense, the difference between the feedforward and feedback forms is equivalent to the difference between the Linearized Kalman filter and the Extended Kalman filter. In the EKF case the state propagation starts from the corrected (updated) state right after a measurement while in the linearized filter the propagation continues at the state that the propagation reached when the measurement appeared, thus ignoring the correction just computed. The linearized Kalman filter and the feedforward Indirect Kalman filter are free to drift with unbounded errors. The basic assumption of Kalman filtering that the estimation errors stay small becomes invalid and thus the performance of the estimator will degrade.

3.3.3 Planar Motion. One Gyro Case - Bias compensation

A common difficulty found in all attitude estimation approaches that use gyros is the low frequency noise component (also referred to as bias or drift) that violates the white noise assumption required for standard Kalman filtering. This problem has attracted the interest of many researchers since the early days of the space program. The first papers describing statistical models of gyro drift were by Newton [48] and Hammon [30]. Newton considered additive white noise in the gyro drift, while Hammon assumed that the gyro drift rates are exponentially correlated random processes. Dushman [18] considered a drift-rate model obtained by adding a random walk component to Hammon's autocorrelation function. Inclusion of the gyro noise model in

a Kalman filter by suitably augmenting the state vector has the potential to provide estimates of the sensor bias when the observability requirement is satisfied. Early implementations of gyro noise models in Kalman filters can be found in: [50], [49], [62], if not for navigation.

This section contains: (i) a summary of the testing results for the Systron Donner gyro, (ii) a presentation of the gyro noise model used throughout this proposal, (iii) derivation of the equations for the Indirect Kalman filter and an examination of its effect on the different sources of sensor noise, (iv) observability considerations, (v) derivation of the equations for the backward filter and the smoother. Simulation results demonstrating the achieved improvement in estimation accuracy are also included in this section.

Systron Donner Quartz Gyro

In the case of Rocky 7 (Figure 3.6), the gyro device is the Systron Donner Quartz Gyro (QRS11-100-420). In the information provided in [1], it is obvious that this gyro does not have a stable bias. From page 1-4: *"Low Rate Application - These gyros showed reasonable performance for rate scale factor stability but would not be useful for applications where bias stability was of high importance to meet mission requirements. The bias changed significantly as the input rate was changing making predictable bias compensation very difficult".*

Long term bias stability data were gathered after the initial testing experiments to create a stochastic model useful for attitude estimator performance prediction. The noise model used was identical to [35] and [36]. This model does not capture the fact that in reality the bias uncertainty will not grow to infinity after infinite time. It is a useful and simple construct for attitude estimator design and simulation, under the assumption that the bias will be calibrated at intervals which are short compared to intervals when the noise model predicts grossly large bias variance. The model assumes that the gyro noise is composed of 3 elements, namely: rate noise $n_r(t)$ (additive white noise), rate flicker noise $n_f(t)$ (generated when white noise passes through a filter with transfer function $1/\sqrt{s}$) and a rate random walk $n_w(t)$ (generated when white noise passes through a filter with transfer function $1/s$). The Power Spectral Density (PSD) of the noise was measured experimentally and the logarithmic plots of the PSD with respect to frequency were used to fit the described model. The intensities calculated (ignoring the flicker

noise) were: $\sigma_r = \sqrt{N_r} = 0.009\ (deg/sec)/\sqrt{Hz}$ and $\sigma_w = \sqrt{N_w} = 0.0005012$ $(deg/sec)\sqrt{Hz}$.

Gyro Noise Model

In our approach we use the simple and realistic model of [21]. In this model the angular velocity $\omega = \dot{\theta}$ is related to the gyro output ω_m according to the equation:

$$\dot{\theta} = \omega_m + b + n_r \tag{3.19}$$

where b is the drift-rate bias and n_r is the drift-rate noise. n_r is assumed to be a Gaussian white-noise process:

$$E[n_r(t)] = 0, \quad E[n_r(t)n_r'(t')] = N_r\delta(t - t') \tag{3.20}$$

The drift-rate bias is not a static quantity but is driven by a second Gaussian white-noise process, the gyro drift-rate ramp noise:

$$\dot{b} = n_w \tag{3.21}$$

with

$$E[n_w(t)] = 0, \quad E[n_w(t)n_w'(t')] = N_w\delta(t - t') \tag{3.22}$$

The two noise processes are assumed to be uncorrelated

$$E[n_w(t)n_r'(t')] = 0 \tag{3.23}$$

We study the simple case of attitude estimation when the vehicle moves in a plane. It is assumed that only one gyro is used and that absolute attitude information (the yaw) is provided directly by another sensor[12] at periodic intervals. The purpose of the current section is to reveal how the gyro bias can be estimated using an absolute measurement of the orientation. The two-dimensional case has the advantage that the system equations are linear; this allows us to study observability analytically. At this point we derive the equations required for the formulation of an Indirect Kalman filter that estimates the orientation of a robot that moves in a 2D space.

[12]This external sensor can be a magnetic compass or a sun sensor in this two dimensional example. From now on we will assume that this sensor measures the absolute orientation of the vehicle.

Equations of the feedback Indirect Kalman filter

As previously mentioned in Equation (3.19) the real angular velocity can be expressed as:

$$\dot{\theta}_{true} = \omega_{measured} + b_{true} + n_r \tag{3.24}$$

where

$$\dot{b}_{true} = n_w \tag{3.25}$$

These equations can be rearranged in a matrix form as:

$$\frac{d}{dt}\begin{bmatrix} \theta_{true} \\ b_{true} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \theta_{true} \\ b_{true} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\omega_{measured} + \begin{bmatrix} n_r \\ n_w \end{bmatrix} \tag{3.26}$$

The absolute orientation measurement is:

$$z = \theta_{measured} = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \theta_{true} \\ b_{true} \end{bmatrix} + n_\theta \tag{3.27}$$

where n_θ is the noise for the sensor measuring the absolute orientation. The assumption is that n_θ is a Gaussian white-noise process with:

$$E[n_\theta(t)] = 0, \quad E[n_\theta(t)n'_\theta(t')] = N_\theta\delta(t-t') \tag{3.28}$$

The orientation estimate obtained by integrating the gyro signal is given by:

$$\dot{\theta}_i = \omega_{measured} + b_i \tag{3.29}$$

while the rate of the estimated bias is given by:

$$\dot{b}_i = 0 \tag{3.30}$$

The state equations for the integrator can be written in a matrix form as:

$$\frac{d}{dt}\begin{bmatrix} \theta_i \\ b_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \theta_i \\ b_i \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\omega_{measured} \tag{3.31}$$

Subtracting Equations (3.24) and (3.29) the error in orientation can be written as:

$$\Delta\dot{\theta} = \Delta b + n_r \tag{3.32}$$

where $\Delta\theta$ is the error in orientation and Δb is the bias error. Subtracting Equations (3.25) and (3.30) the bias rate error can be written as:

$$\Delta\dot{b} = n_w \tag{3.33}$$

The error propagation equations for the Indirect (error state) Kalman filter can be rearranged in a matrix form as:

$$\frac{d}{dt}\begin{bmatrix} \Delta\theta \\ \Delta b \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \Delta\theta \\ \Delta b \end{bmatrix} + \begin{bmatrix} n_r \\ n_w \end{bmatrix} \tag{3.34}$$

or in a more compact form as:

$$\frac{d}{dt}\Delta x = F\Delta x + n \tag{3.35}$$

We will assume that the measurement provided to the Indirect Kalman filter is:

$$\Delta z = \theta_{measured} - \theta_i = \theta_{true} + n_\theta - \theta_i = \Delta\theta + n_\theta \tag{3.36}$$

where θ_i is available through the gyro signal integration and $\theta_{measured}$ is the absolute orientation measurement. This equation in matrix form becomes:

$$\Delta z = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \Delta\theta \\ \Delta b \end{bmatrix} + n_\theta \tag{3.37}$$

or in a more compact form:

$$\Delta z = H\Delta x + n_\theta \tag{3.38}$$

The continuous Kalman filter equation for the covariance update is:

$$\dot{P} = FP + PF^T + Q - PH^T R^{-1} HP \tag{3.39}$$

where P is the covariance matrix, F is the system matrix, H is the measurement matrix, Q is the system noise covariance matrix and R is the measurement noise covariance matrix. Consider this equation in the steady state case where $lim_{t\to\infty}(\dot{P}) = 0$ and we have

$$0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} N_r & 0 \\ 0 & N_w \end{bmatrix} -$$

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} N_\theta \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{3.40}$$

Solving this equation for the elements of matrix P we find:

$$p_{11} = \sqrt{N_\theta}\sqrt{N_r + 2\sqrt{N_w N_\theta}} \tag{3.41}$$

$$p_{12} = \sqrt{N_w N_\theta} \tag{3.42}$$

$$p_{22} = \sqrt{N_w}\sqrt{N_r + 2\sqrt{N_w N_\theta}} \tag{3.43}$$

Now the steady state Kalman gain can be calculated from:

$$K = PH^T\ R^{-1} = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} N_\theta \end{bmatrix}^{-1} = \begin{bmatrix} \sqrt{\frac{N_r + 2\sqrt{N_w N_\theta}}{N_\theta}} \\ \sqrt{\frac{N_w}{N_\theta}} \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{3.44}$$

The Indirect Kalman filter estimates the error states. Starting from Equation (3.34), the estimate propagation equation with the added correction is:

$$\frac{d}{dt} \begin{bmatrix} \widehat{\Delta\theta} \\ \widehat{\Delta b} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\Delta\theta} \\ \widehat{\Delta b} \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \left(\Delta z - \widehat{\Delta\theta} \right) \tag{3.45}$$

Substituting the error state estimates according to

$$\widehat{\Delta\theta} = \hat{\theta} - \theta_i \tag{3.46}$$

$$\widehat{\Delta b} = \hat{b} - b_i \tag{3.47}$$

in Equation (3.45) we have

$$\frac{d}{dt} \begin{bmatrix} \hat{\theta} - \theta_i \\ \hat{b} - b_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\theta} - \theta_i \\ \hat{b} - b_i \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \left(\Delta z - \widehat{\Delta\theta} \right) \tag{3.48}$$

separating the estimated from the integrated quantities we have:

$$\frac{d}{dt} \begin{bmatrix} \hat{\theta} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\theta} \\ \hat{b} \end{bmatrix} + \left(\frac{d}{dt} \begin{bmatrix} \theta_i \\ b_i \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_i \\ b_i \end{bmatrix} \right) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \left(\Delta z - \widehat{\Delta\theta} \right) \tag{3.49}$$

Notice that from Equations (3.36) and (3.46):

$$\Delta z - \widehat{\Delta\theta} = (\theta_{measured} - \theta_i) - (\hat{\theta} - \theta_i) = \theta_{measured} - \hat{\theta} \tag{3.50}$$

Now substituting in Equation (3.49) from Equations (3.31) and (3.50) we have

$$\frac{d}{dt}\begin{bmatrix} \hat{\theta} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \hat{\theta} \\ \hat{b} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\omega_{measured} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}(\theta_{measured} - \hat{\theta}) \tag{3.51}$$

Deriving the Laplace transform of this equation we have

$$s\begin{bmatrix} \hat{\Theta}(s) \\ \hat{B}(s) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \hat{\Theta}(s) \\ \hat{B}(s) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\Omega_{measured}(s) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}(\Theta_{measured}(s) - \hat{\Theta}(s)) \tag{3.52}$$

which becomes

$$\left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ k_2 & 0 \end{bmatrix}\right)\begin{bmatrix} \hat{\Theta}(s) \\ \hat{B}(s) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\Omega_{measured}(s) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}\Theta_{measured}(s) \tag{3.53}$$

solving for $\hat{\Theta}(s)$ and $\hat{B}(s)$

$$\begin{bmatrix} \hat{\Theta}(s) \\ \hat{B}(s) \end{bmatrix} = \begin{bmatrix} s + k_1 & -1 \\ k_2 & s \end{bmatrix}^{-1}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\Omega_{measured}(s) + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}\Theta_{measured}(s)\right) \tag{3.54}$$

Finally, the quantity of interest in the frequency domain is expressed as:

$$\hat{\Theta}(s) = \frac{s}{s^2 + k_1 s + k_2}\Omega_{measured}(s) + \frac{k_1 s + k_2}{s^2 + k_1 s + k_2}\Theta_{measured}(s) \tag{3.55}$$

or

$$\hat{\Theta}(s) = \frac{s^2}{s^2 + k_1 s + k_2}\frac{\Omega_{measured}(s)}{s} + \frac{k_1 s + k_2}{s^2 + k_1 s + k_2}\Theta_{measured}(s) \tag{3.56}$$

or

$$\hat{\Theta}(s) = F(s)\frac{\Omega_{measured}(s)}{s} + (1 - F(s))\Theta_{measured}(s) \tag{3.57}$$

which reveals that the Indirect Kalman filter weighs the two different sources of information (i.e. the integrated gyro signal and the absolute orientation

measurement) in a complementary fashion according to their noise characteristics. To acquire more intuition on how the Indirect Kalman filter deals with the noise, we should examine the case where the sensor inputs are only noise, (i.e. the sensor signals do not contain any useful information at any frequency).

In Figure 3.8, the first plot shows the effect of the integration process on the gyro noise. The already strong noise components in the lower frequencies (below $f = 10^{-2.5} = 0.0032Hz$) are amplified because of the integration process. Thus, useful low frequency information will be contaminated by this noise. The obvious conclusion is that we can not rely on the gyroscope integration to estimate low frequency motion. On the other hand, in higher frequencies the effect of the integrator is to suppress the gyro noise and therefore the gyro becomes reliable for high frequency motion. The ideal situation would be to fuse the integrated gyro information with a sensor (an absolute orientation sensor such as a sun sensor, for example) that has a complementary noise profile (i.e. the noise is small at lower frequencies and becomes larger at higher frequencies) or at least has a constant noise profile, the same for all frequencies. The second case has been assumed and the noise profiles of both the integrated gyro and the absolute orientation sensor are depicted in the second plot of Figure 3.8.

At this point we should clarify how the Indirect Kalman filter optimally combines the information from the two different sources by examining it as a signal processing entity. The explanation comes from the third plot of Figure 3.8. Function $F(s)$ which filters the integrated gyro signal works as a high pass filter. It suppresses the low frequency noise content and allows the higher frequencies to pass undistorted. To the contrary, function $1 - F(s)$ filters the absolute orientation sensor and acts as a low pass filter. It allows the critical low frequency information from the absolute orientation sensor to pass undistorted while reduces its strength in the higher frequencies. This is desired since the estimator performs better if it relies on the gyro in that range.

Observability

Measuring Orientation Continuously
The observability matrix of the system is:

$$\left[\begin{array}{c:cc} H^T & F^T & H^T \end{array}\right] = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right] \tag{3.58}$$

Clearly it has rank 2. Thus the system is observable and the uncertainty for both states (orientation and bias) is bounded. This can be seen in Figure 3.9. Both covariances reach a steady state.[13] It is worth noticing that though the bias is not directly measured it can be estimated. This is shown in Figure 3.10.

Measuring Orientation Intermittently
In this case the observability matrix

$$\left[\begin{array}{c:cc} H^T & F^T & H^T \end{array}\right] = \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \end{array}\right] \tag{3.59}$$

looses its rank for those time intervals for which we do not have an absolute measurement of the orientation. This can be seen graphically in Figure 3.11. The covariances (which capture the uncertainty) of both the orientation and the bias grow without bounds until a new measurement from the absolute orientation sensor is received. This causes a decrease in the uncertainty which then begins to grow as before. Following this strategy (i.e. taking an absolute orientation measurement only at certain instants) produces the "saw tooth" pattern in the covariance plot.[14]

The formal explanation is provided by the discrete Kalman filter formulation for the covariance:

$$P_{k,k-1} = \Phi(k, k-1) P_{k-1,k-1} \Phi(k, k-1)^T + Q_{k-1} \tag{3.60}$$

[13]The covariance matrix P calculated in the Kalman filter is defined as:

$$P = E\{\left[\begin{array}{c} \hat{\theta} \\ \hat{b} \end{array}\right] \left[\begin{array}{cc} \hat{\theta} & \hat{b} \end{array}\right]\} = \left[\begin{array}{cc} \sigma^2_{\hat{\theta}} & \sigma_{\hat{\theta},\hat{b}} \\ \sigma_{\hat{b},\hat{\theta}} & \sigma^2_{\hat{b}} \end{array}\right]$$

The covariance of the orientation estimate $\hat{\theta}$ and of the bias estimate $\hat{b}$ are given by the first and the second diagonal element of this matrix, respectively.

[14]This observation was first reported in [52].

$$P_{k,k} = P_{k,k-1} - P_{k,k-1}H_k^T\ [H_k P_{k,k-1} H_k^T + R_k]^{-1} H_k P_{k,k-1} \qquad (3.61)$$

where $P_{k-1,k-1}$ is the covariance matrix from the previous step (time t_{k-1}, measurements up to $k-1$), $P_{k,k-1}$ is the propagated covariance (time t_k, measurements up to $k-1$), $P_{k,k}$ is the updated covariance after the new absolute orientation measurement (time t_k, measurements up to k), $\Phi(k,k-1)$ is the state transition matrix from time step t_{k-1} to time step t_k, H_k is the measurement matrix at time t_k, Q_k is the system noise covariance matrix at time t_k and R_k is the measurement noise covariance matrix at time t_k.

Equation (3.60) represents the increase in the system uncertainty at every time step due to the propagated previous system uncertainty and the newly added system noise. This equation is valid at every time step and causes the covariance to grow without bounds. Equation (3.61) is applied only when a new measurement arrives. It reduces the value of the propagated covariance matrix $P_{k,k-1}$ since it subtracts from it a positive definite matrix ($P_{k,k-1}H_k^T\ [H_k P_{k,k-1} H_k^T + R_k]^{-1} H_k P_{k,k-1}$). The smaller the matrix R_k (i.e. the better the quality of the absolute orientation measurement), the larger the decrease.

Here we focus on how the Indirect Kalman filter tracks the bias in the case that the system is observable only at certain time instants. In Figure 3.12 we see that in the absence of any other information, the filter assumes that the bias is constant. When a new absolute orientation measurement is introduced, the filter updates the bias estimate according to the cumulative effect of the bias on the error in orientation during the preceding time interval. After that the bias is assumed again to be constant, until the next measurement.

It is obvious that in the long run, uncertainty in orientation is bounded. It will not exceed a certain limit depending on the frequency of the absolute orientation updates and on the accuracy of this sensor along with the noise profile of the gyro and its bias. Thus, the frequency of the absolute measurements is a design parameter that the system engineer can select based on mission specific constraints.

Backward filter

At this point we should revisit the initial problem at hand. We built an Indirect Kalman filter that estimates the vehicle orientation mainly because

we need the attitude information to calculate the position update of the rover. The orientation information is required at every time step (Equations (3.16) and (3.17)) and its accuracy is crucial for precise position estimation. As we can see from the orientation covariance plot in Figure 3.11 the covariance of the estimate reaches its maximum right before a new absolute orientation measurement. During the preceding time instants, the orientation information becomes increasingly untrustworthy and the validity of the position updates that depend on it, decays.

In the proposed novel approach we take advantage of the fact that the uncertainty decreases considerably right after the absolute orientation measurement at some time t_k. We apply a backward propagation of the Indirect Kalman filter every time we receive a good estimate of the orientation (i.e. when a new absolute orientation measurement is introduced). In the backward propagation the initial condition for the state is the updated (corrected) state estimate at time t_k. Then a backward integration using the previously acquired (stored) gyro values is performed to calculate the orientation at each previous time instant up to the last absolute orientation measurement at time t_{k-M}. The effect is to obtain more accurate estimates for the interval preceding time t_k. These estimates will have the same behavior when they move away (in the backward time direction) from t_k as the forward filter estimates have when they move away from time t_{k-M} (previous absolute orientation measurement). As the backward filter estimates get closer to t_{k-M} their covariance increases and the quality of the estimate decreases. This is clearly shown in Figure 3.13. To better understand the functionality of the backward filter we examine the equations that govern it as presented in [24]. If the system equation is

$$\dot{x} = Fx + n \tag{3.62}$$

then by substituting

$$\tau = T - t \tag{3.63}$$

where τ is the backward time variable and

$$T = t_k - t_{k-M} \tag{3.64}$$

is the time interval of smoothing, the backward system equation can be derived from:

$$\frac{dx}{d\tau} = \frac{dx}{dt}\frac{dt}{d\tau} = -\dot{x} \tag{3.65}$$

and is:

$$\frac{dx_b}{d\tau} = -Fx_b - n \tag{3.66}$$

The continuous time covariance propagation equation for the forward filter is:

$$\dot{P} = FP + PF^T + Q \tag{3.67}$$

and correspondingly the covariance backwards propagation equation would be:

$$\dot{P}_b = -FP_b - P_bF^T + Q \tag{3.68}$$

In the discrete time domain the previous equation becomes:

$$P_{b(k-1,k)} = \Phi(k-1,k)P_{b(k,k)}\Phi(k-1,k)^T + Q_k \tag{3.69}$$

which is of the same form as Equation (3.60) and thus the covariance of the backward filter also increases during propagation in the same manner as the forward filter's covariance. The main difference is that P_b starts from a small value at time t_k and grows as it moves towards t_{k-M}. These results are depicted in Figure 3.13 for the orientation covariance. In Figure 3.14 we can see the performance of the backward filter for the variable of interest, the orientation. The orientation estimate for the backward filter is close to its true value for $t = 400sec$ and deviates from it as it reaches close to $t = 200sec$. The next step is to optimally combine the forward and the backward filter estimates in the smoother.

Smoother

The smoother combines the estimates of the forward and the backward Kalman filter to provide estimates of the state at every time as if all the measurements during the entire smoothing interval were available. The total (smoothed) result outperforms both filters. This is evident in Figure 3.13 where the total covariance (or smoothed covariance) is at all times smaller than the smallest of the two filters. This can be explained by examining the equation for the smoothed covariance [44]:

$$P_{total}^{-1} = P_f^{-1} + P_b^{-1} \tag{3.70}$$

The equation for the total (smoothed) state is[15]

$$P_{total}^{-1}\hat{x}_{total} = P_f^{-1}\hat{x}_f + P_b^{-1}\hat{x}_b \tag{3.71}$$

Equation (3.71) demonstrates how the states are optimally combined to produce the smoothed (total) estimate. Each of the covariance matrices represents the uncertainty of the estimate. The inverse of it represents the certainty or the information contained in this estimate. The higher the uncertainty, the larger the covariance matrix, or the smaller the information matrix, or the smaller the contribution of this filter to the total estimate. In other words the previous equation weighs each of the available estimates (from the forward and the backward filter) according to their certainty. The result is the optimal estimate given that all the measurements of the time interval of smoothing are available at once. The achieved improvement is obvious in Figures 3.13, 3.14, and 3.15. The accuracy of the absolute orientation sensor was 3 degrees for all the simulation results presented here.

At this point we should make clear that the smoothed estimate is not available in real-time. It requires the forward Kalman filter to propagate the estimate from t_{k-M} to t_k before the backward filter starts. Thus the smoothed estimation is performed off-line and so is the position estimation using the smoothed orientation estimates.

[15]It should be clarified that the quantities with subscript f (forward) are the same as those previously used in the Indirect Kalman filter without any subscript, i.e. $P_f = P$ and $\hat{x}_f = \hat{x}$. The new notation is used to distinguish these quantities from the corresponding ones of the backward filter, where the subscript b is used to denote backward, and the ones of the smoother where the subscript *total* is used for the smoothed (total) estimate.

Figure 3.5: *The rover is capable of climbing over small rocks without significant change in attitude. To a certain extent, the bogey joints decouple the ground morphology from the vehicle's motion.*

Figure 3.6: *The Rocky 7 Mars rover prototype.*

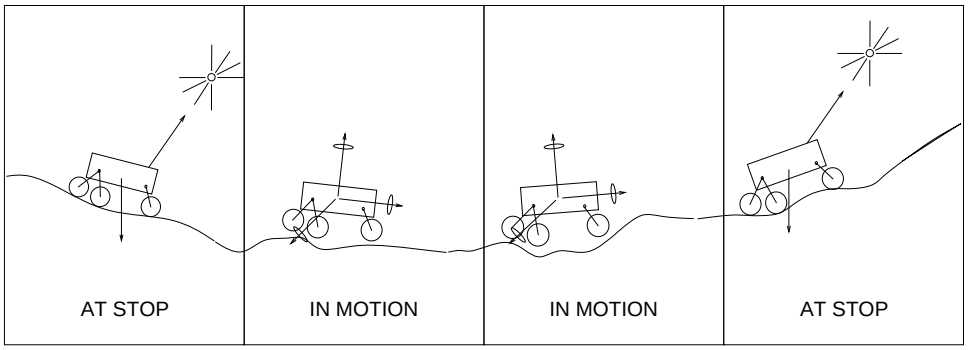

Figure 3.7: *In the first and the last frame the vehicle is stopped and uses the sun sensor and the 3-axis accelerometer to determine its absolute orientation. The second and the third frame are two instances of the vehicle while in motion. The motion starts at the position shown in the first frame and it ends at the position shown in the last frame. The attitude estimation throughout the in between time interval relies solely on the gyros.*

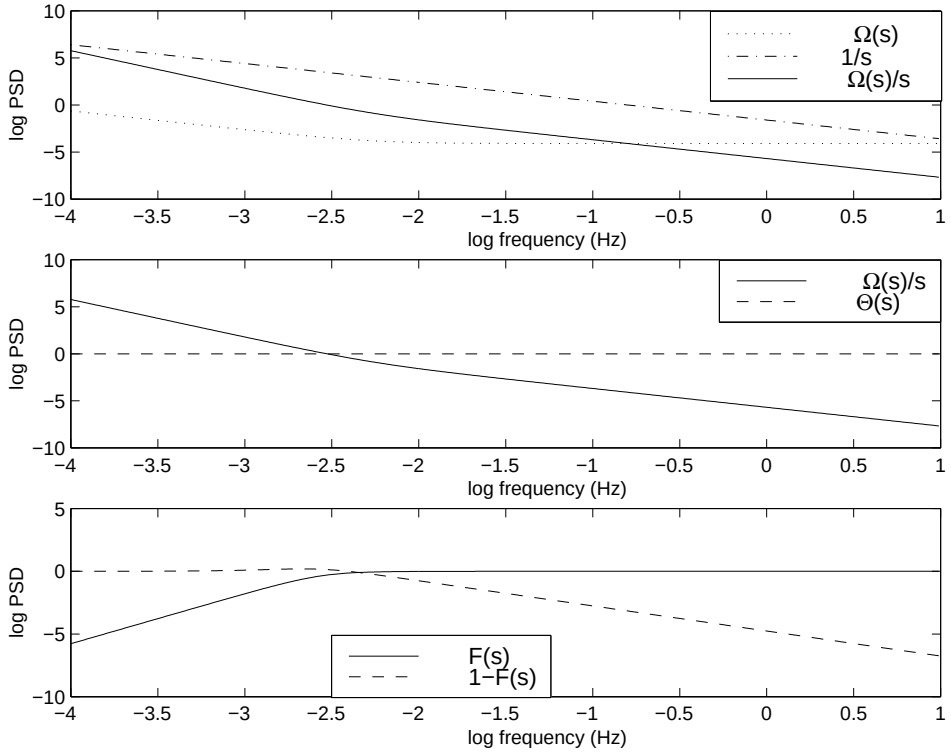

Figure 3.8: *In the first plot the power spectral density of the gyro noise is displayed before* $(\Omega(s))$ *and after* $(\Omega(s)/s)$ *integration* $(1/s)$*. In the second plot the power spectral densities of the integrated gyro noise and of the absolute orientation sensor noise are presented. The third plot displays the power spectral densities of function* $F(s)$ *that filters the integrated gyro noise and of function* $1 - F(s)$ *that filters the absolute orientation sensor noise.*

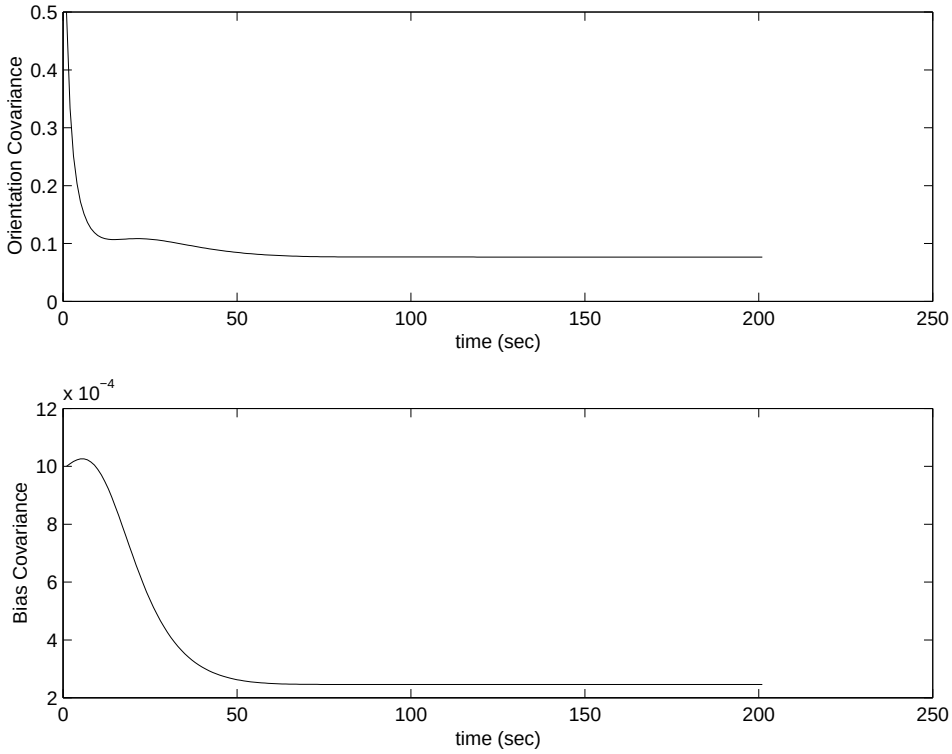

Figure 3.9: *Measuring orientation continuously: In the first plot the covariance of the orientation is displayed. In the second plot the covariance of the bias is shown. Both of these covariances reach a low steady state. This is expected since the system is observable.*

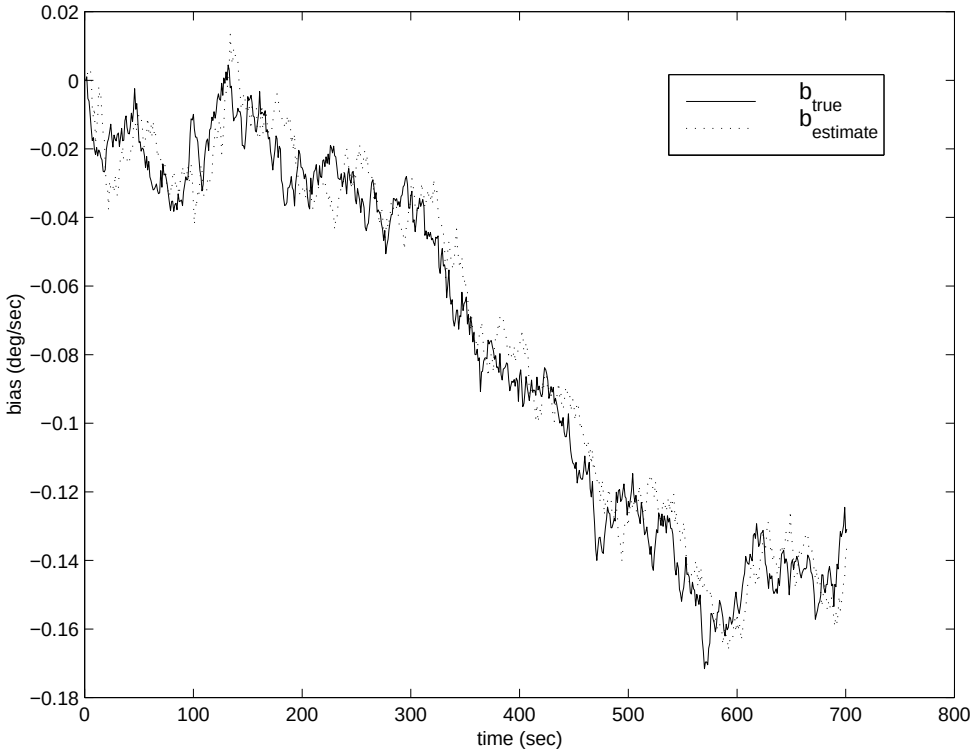

Figure 3.10: *The solid line represents the true value of the gyro bias. The dotted line is the estimate of the bias from the Indirect Kalman filter. Though the estimate follows the true value it is obvious that there is a lag. This is because the absolute orientation sensor does not measure the bias directly. It only measures its effect on the orientation error.*

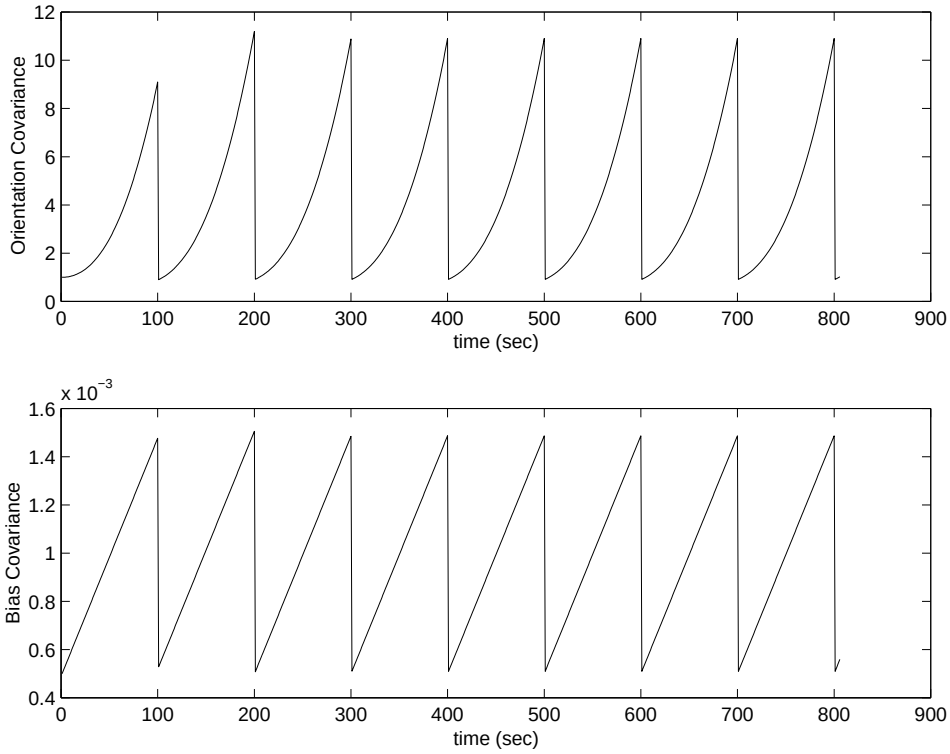

Figure 3.11: *Measuring orientation intermittently: In the first plot the covariance of the orientation is presented. The sudden decreases take place at the instants when an absolute orientation measurement is provided. The second plot portrays the covariance of the bias for the same case.*

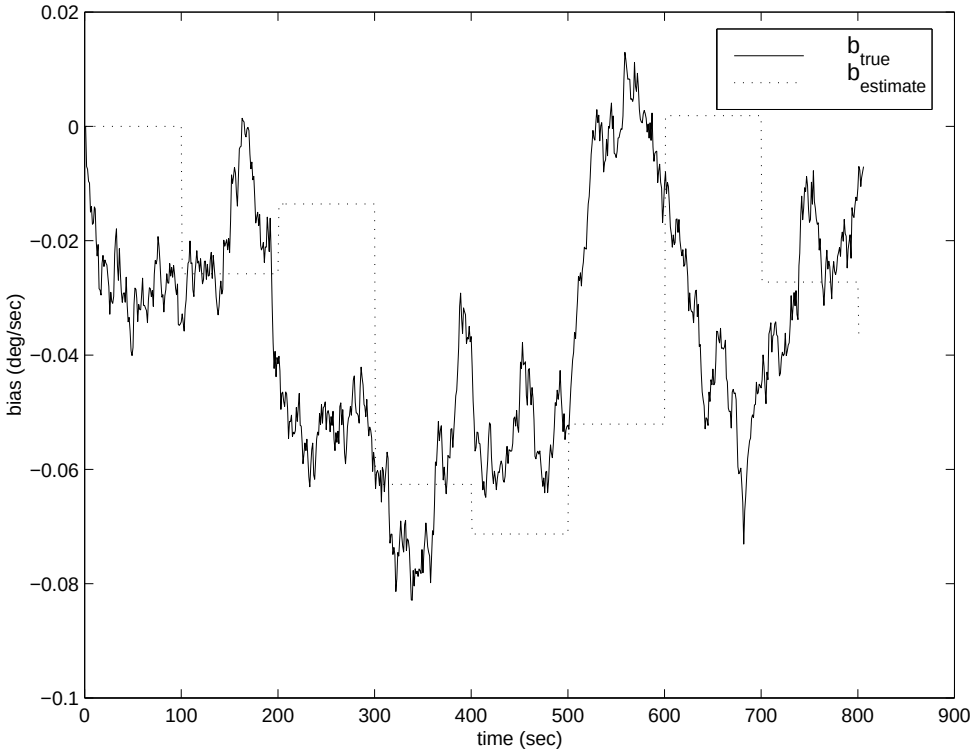

Figure 3.12: *The flat parts of the estimate depict the constant bias assumption in the integrator. The sharp step changes occur when absolute orientation measurements are provided.*

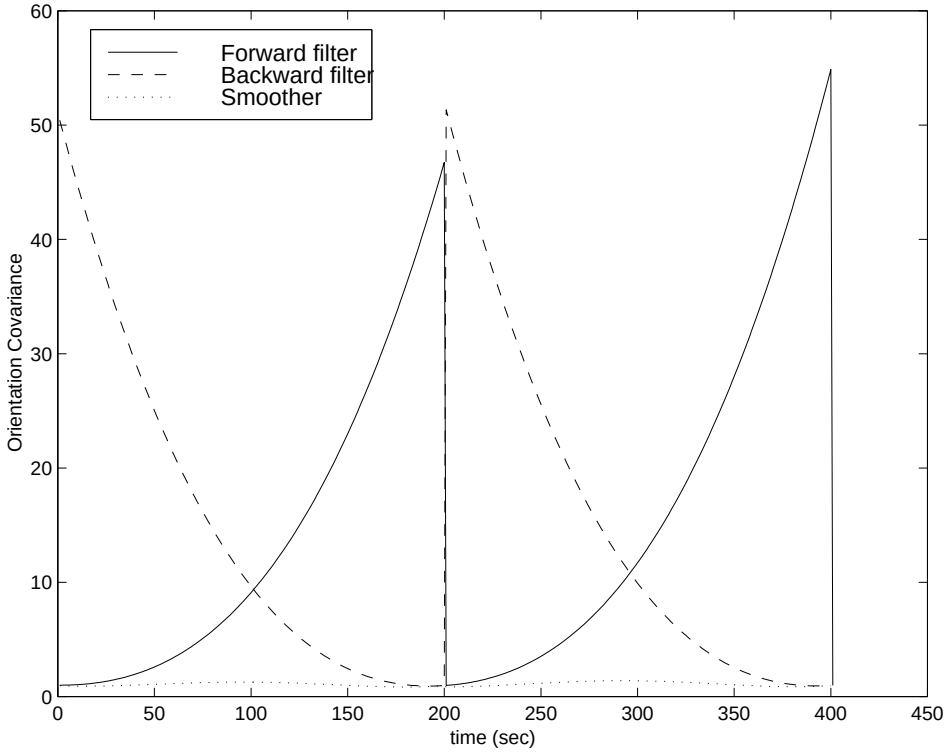

Figure 3.13: *The solid line represents the orientation covariance of the forward filter, the dashed the orientation covariance of the backward filter as they propagate during the intervals between three absolute orientation measurements at 0, 100 and 400 seconds respectively. The dotted line shows the orientation covariance resulting from the combination of the forward and the backward filter.*

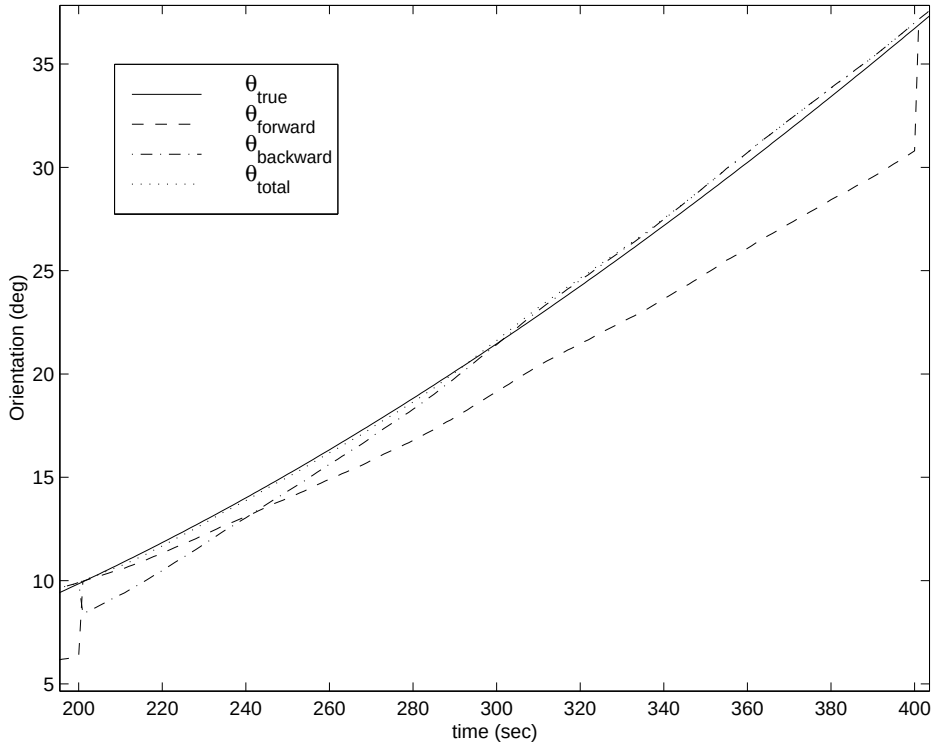

Figure 3.14: *The solid line shows the true orientation of the vehicle. The dashed line shows the forward filter's estimate and the dashed-dotted the backward filter's estimate. The dotted line is the resulting orientation estimate from the smoother. The smoothed (total) estimate stays close to the backward filter estimate for the second half of the smoothing interval while for the first part it relies on both filters and slowly abandons the backward filter estimates which drift away from the true values.*

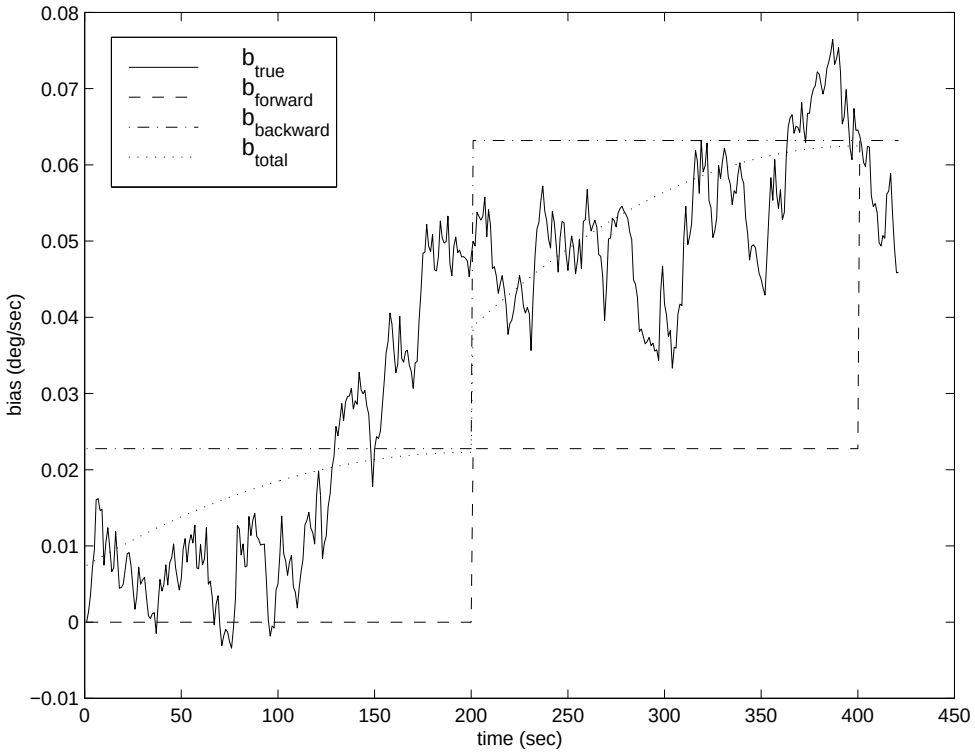

Figure 3.15: *The solid line shows the true gyro bias, the dashed the forward filter's estimate and the dashed-dotted the backward filter's estimate. The dotted line is the resulting bias estimate from the smoother. The smoothed (total) estimate stays close to the backward filter estimate for the second half of the smoothing interval while for the first part it depends on both the forward filter's estimate and the backward filter's estimate. This is due to the fact that the initial bias value of the backward filter is more trustworthy for this time interval than the initial value for the forward filter. The explanation is that the initial bias value for the backward filter is the one obtained at the end of the smoothing time interval; it is provided from the forward filter and is based on the bias values during this particular interval. The forward filter's initial bias value is based on the bias values during the previous time interval and thus is a less accurate predictor of the current interval.*

Chapter 4

Proposed Future Work

4.1 Multiple Hypothesis Tracking

4.1.1 Environments with many types of features

We propose to apply the *Multiple Hypothesis Tracking* method to the localization problem when more than one feature type is considered. In addition to doors, interior and exterior corners can also be used as different types of potential landmarks for localization in indoor environments. The following formulation shows that the proposed approach builds naturally on the methodology introduced in Section 3.2.1. The suggested algorithm is similar to the algorithm presented in Figure 3.2. Equation (3.13) has to be modified in order to accommodate the existence of different types of features. After the robot encounters the kth landmark, the distribution of its position is described by the following pdf:

$$f(\mathbf{x}/\mathbf{z_k}) = f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k) =$$

$$\sum_{j=1}^{M} f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(z_k = B_j/Type(z_k) = B) P(Type(z_k) = B) =$$

$$\sum_{j=1}^{M} P(z_k = B_j/Type(z_k) = B) P(Type(z_k) = B) \times$$

$$\sum_{i=1}^{N} f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) =$$

$$\sum_{i=1}^{N} \sum_{j=1}^{M} P(z_k = B_j/Type(z_k) = B) P(Type(z_k) = B) \times$$

$$f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j) \tag{4.1}$$

where N is the number of hypotheses in the previous step and the M is the number of possible matches on the map for the kth landmark. $P(z_k = B_j)$ is the probability that the kth landmark is B_j and it is calculated as the product of $P(z_k = B_j/Type(z_k) = B)$ which is the probability that the landmark in sight is B_j given that its type is B, and $P(Type(z_k) = B)$ which is the probability that the feature type is B. This last probability is provided by the feature extraction module. We assume that the feature extraction module can detect landmarks of different types and provide the level of confidence associated with the decision about the type.

The quantity $f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j)$ in Equation (4.1) is the updated pdf for the position estimate after taking into consideration the new kth landmark. This quantity is calculated in the Kalman filter as the latest estimates are combined with the new information for the position of the new landmark B_j. Each *new* hypothesis H_j assumes that the position of B_j follows a Gaussian distribution with pdf as in Equation (3.4). The fusion of the latest Kalman filter estimates $\widehat{\mathbf{x}}_{\mathbf{robot}}(t,i)$ and $\mathbf{P}_{\widehat{\mathbf{x}}_{\mathbf{robot}}}(t,i)$ with the position information $\mathbf{x}_{B_j}$, $\mathbf{K}_{B_j}$ for each of the possible landmarks B_j is a *one step* process and thus it can be implemented in parallel. The resulting $f(\mathbf{x}/\mathbf{z_k})$ is a multi-modal Gaussian distribution composed of 1 to $N \times M$ functionals. The quantity $P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = B_j)$ in Equation (4.1) is the result of the Bayesian estimation step (Multiple Hypothesis Testing) and is calculated for each hypothesis H_i using Equations (3.14) and (3.15).

4.1.2 Absence of expected features

The focus of the proposed study will be in cases where more than one hypotheses for the position of the robot are valid and according to one or more of these there should be a specific feature appearing when the robot has covered a certain distance. The absence of the expected landmark carries valuable information. We intend to involve the *Multiple Hypothesis Tracking* framework in order to exploit this additional source of information.

The algorithm employed here will be the one shown in Figure 4.1. There is an extra step added to this when compared to the algorithm presented in Figure 3.2. At every time instant, along with the position tracking, the localization module checks for the existence of features in the environment. Thus, the competition between the hypotheses continues even when there is no landmark in sight. The absence of a landmark carries enough information to terminate some of the hypotheses and thus strengthen others (the total probability of all the hypotheses adds up to one at any given time). By detecting what is *not* in the vicinity of the robot we will be able to prune down the number of existing hypotheses.

The following formulation shows that the proposed approach is an extension of the methodology introduced in Section 3.2.1. Starting from Equation (3.13) the distribution of the position of the robot is described by the following pdf:

$$f(\mathbf{x}/\mathbf{z_k}) = f(\mathbf{x}/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = \emptyset) =$$

$$\sum_{i=1}^{N} f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = \emptyset) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = \emptyset) =$$

$$\sum_{i=1}^{N} f(\mathbf{x}, H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}) P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = \emptyset) \tag{4.2}$$

where N is the number of hypotheses in the previous step. $z_k = \emptyset$ denotes that there is no feature detected at this time instant. The first quantity in Equation (4.2) is the current pdf for the position estimate which is independent of the absence of a landmark. This quantity is updated by the Kalman filter continuously by adding the latest Kalman filter estimates $\widehat{\mathbf{\Delta x}}(t)$ and

$\mathbf{P}_{\widehat{\mathbf{\Delta x}}}(t)$ to the position information $\mathbf{x}_{A_i}$, $\mathbf{K}_{A_i}$ for each of the N existing hypotheses.

The second quantity in Equation (4.2) is the result of the Bayesian estimation step (Multiple Hypothesis Testing) and is calculated for each hypothesis H_i as follows:

$$P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, z_k = \emptyset) =$$

$$\frac{P(z_k = \emptyset/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})}{\sum_{i=1}^{N} P(z_k = \emptyset/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})} \tag{4.3}$$

where $P(H_i/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}})$ is the a priori probability for each hypothesis H_i available from the previous step and $P(z_k = \emptyset/\mathbf{z_{k-1}}, \widehat{\mathbf{\Delta x}}, H_i)$ is the probability that there is no detectable feature given that the current location of the robot is $\widehat{\mathbf{x}}_{\mathbf{robot}}(t, i)$. This probability is close to zero for those hypotheses whose assumption for the estimated position of the robot is closer to the location of any of the existing landmarks. These hypotheses will then be weakened. The remaining will be strengthened. The total probability calculated by summing the probabilities in Equation (4.2) has to equal to 1.

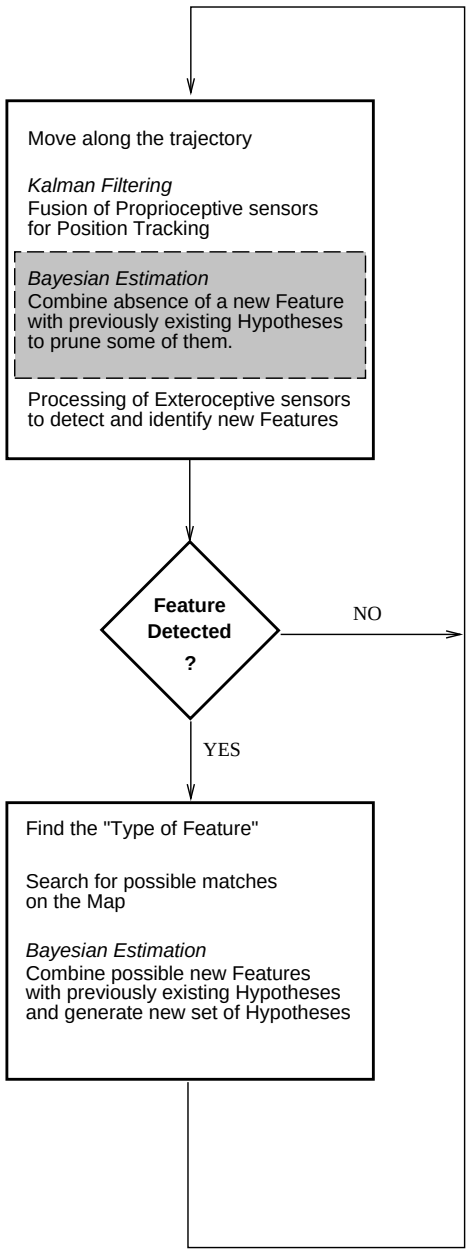

Figure 4.1: *Multiple Hypotheses Tracking Algorithm. Case: Absence of Expected Features.*

4.1.3 Fault detection and identification

Mobile robots are susceptible to sensor and mechanical failures. We propose to investigate the applicability of the *Multiple Hypothesis Tracking* approach to the task of fault detection and identification. The basic idea is to construct a bank of Kalman filters each of them tuned to a specific failure of the system. The proposed architecture is depicted in Figure 4.2.

Failure detection and identification is accomplished using hypothesis testing on the residuals. Each filter in the bank of Kalman filters, produces a residual r_i. The nominal model results in a residual of r_0. All the residual vectors are fed into the fault detection and identification module. The decision if and which failure has occurred depends on the selection of the state estimator with the minimal residual. This estimator is the one which assumes currently sound knowledge of the system description, i.e. it has incorporated the failure effect in its structure.

At each time step the estimated state of the system is given by:

$$\hat{\mathbf{x}}_k = \sum_{i=0}^{N} \hat{\mathbf{x}}_k(\alpha_\mathbf{i})\ P(\alpha_\mathbf{i}/\mathbf{z_k}), \tag{4.4}$$

where each of the estimates $\hat{\mathbf{x}}_k(\alpha_\mathbf{i}), i = 0..N$ is provided by the corresponding estimator in the bank of Kalman filters and the probability $P(\alpha_\mathbf{i}/\mathbf{z_k})$ is calculated as

$$P(\alpha_\mathbf{i}/\mathbf{z_k}) = \frac{P(z_k/\mathbf{z_{k-1}},\ \alpha_\mathbf{i})P(\alpha_\mathbf{i}/\mathbf{z_{k-1}})}{\sum_{i=0}^{N} P(z_k/\mathbf{z_{k-1}},\ \alpha_\mathbf{i})P(\alpha_\mathbf{i}/\mathbf{z_{k-1}})}. \tag{4.5}$$

This probability expresses the confidence to the estimator i. Each estimator is a Kalman filter which assumes that the parameter vector is $\alpha_\mathbf{i}$ and thus it is tuned to the modeled failure i for $i = 1..N$, or to the nominal model of the system for $i = 0$. $P(\alpha_\mathbf{i}/\mathbf{z_{k-1}})$ is the same probability provided from the previous step and $P(z_k/\mathbf{z_{k-1}},\ \alpha_\mathbf{i})$ is the probability that the received measurement at this time instant is z_k provided that the previous measurements were $\mathbf{z_{k-1}}$ and the parameter vector is $\alpha_\mathbf{i}$. This conditional probability is calculated as:

$$P(z_k/\mathbf{z_{k-1}},\ \alpha_\mathbf{i}) = \frac{1}{(2\pi)^{m/2}|\mathbf{S}_k|^{1/2}} \exp[-\frac{1}{2}\mathbf{r}_k{}^T \mathbf{S}_k{}^{-1}\mathbf{r}_k], \tag{4.6}$$

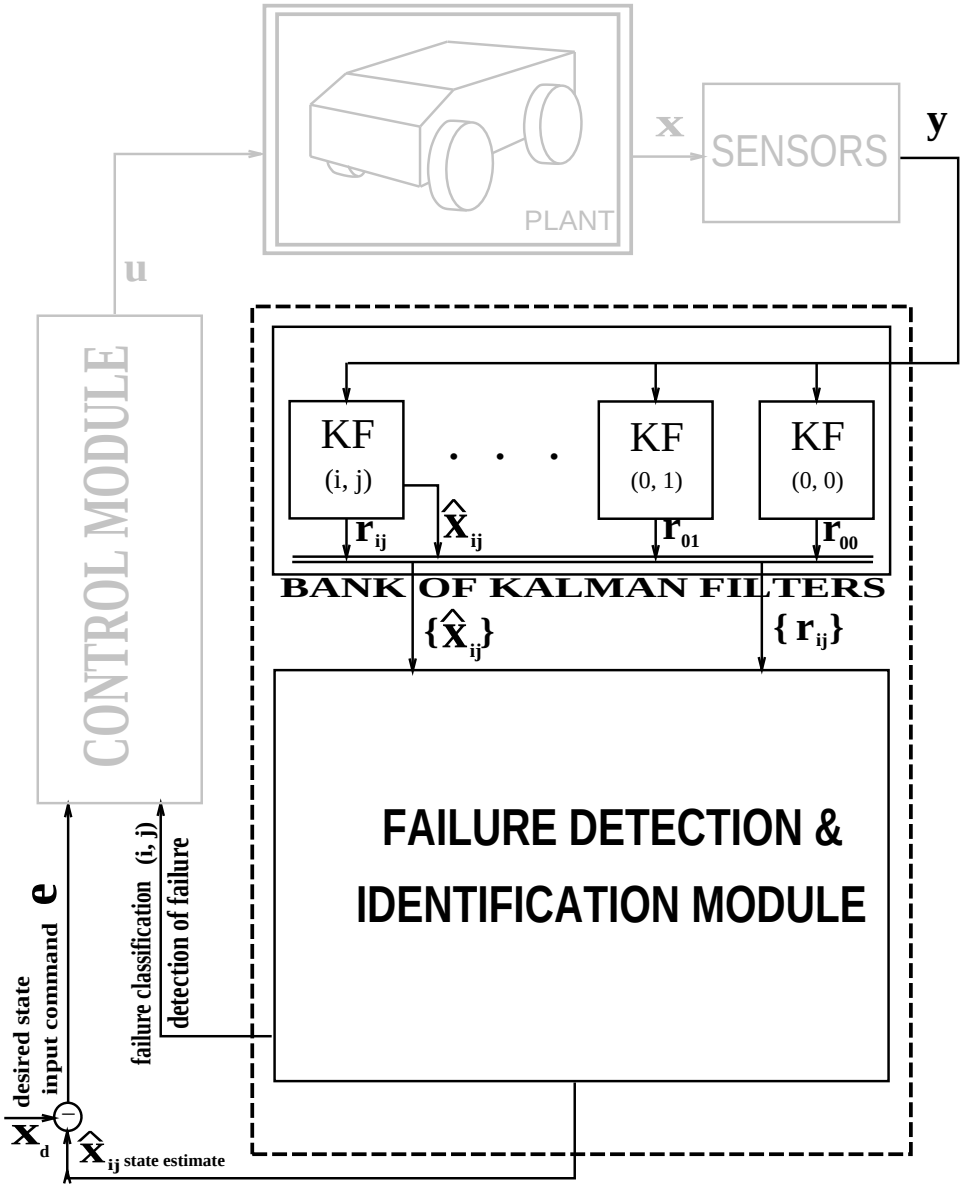

Figure 4.2: *Failure Detection and Identification*

where m is the dimension of the measurement vector z_k, and $\mathbf{S}_k$ is the covariance of the residual $\mathbf{r}_k$ which is calculated as:

$$\mathbf{r}_k = z_k - \hat{z}_k, \tag{4.7}$$

the difference between the actual measurement z_k and the estimated (expected) measurement $\hat{z}_k$.

4.2 Smoothing

We propose to extend the smoother based attitude estimation to the case of robots that perform tasks outdoors and attitude measurements are available to them intermittently. In a non structured environment, the flat terrain assumption fails and thus the localization module has to monitor the motion of the vehicle in all 3 dimensions. For the attitude estimation part of the algorithm, we intend to use quaternion algebra for the formulation of the equations of the Indirect Kalman filter.

4.2.1 Smoother Based Attitude Estimation: 3-D Enhanced Localization

In the previous chapter we demonstrated the proposed method for the case of planar motion. The linear form of the system and measurement equations allowed us to study observability analytically and examine the role of the Kalman filter as a signal processing unit that weighs the sensory information according to their noise power spectral density profile. We also demonstrated the improvement in the overall quality of the attitude estimate when the smoother was involved. The lack of observability for the largest part of the trajectory forces the covariance of the estimate to steadily increase after each absolute measurement. The increase of the uncertainty is due to the gyro noise and bias-drift. During the instants of observability, the accumulated effect of the gyro noise is canceled and precise attitude information is provided to the filter. The estimate becomes more trustworthy for a limited time. As the vehicle moves the quality of the attitude estimate decreases. Any algorithm that computes the position of the robot using these attitude estimates will drift. This anomaly manifests itself as a saw-tooth pattern in the time profile of the covariance of the estimate. A smoother can be used when the system is observable only during certain time instants. The smoother computes the best estimates of the attitude during a certain time interval provided all the measurements collected during this time. These measurements include the initial absolute orientation, all the intermediate (inertial) readings from the gyros, and the final absolute orientation. This last measurement marks the end of this time interval and the beginning of the next one. The improvement is greater for those parts of the trajectory during which the estimate is of the lowest quality, i.e. for the instants right

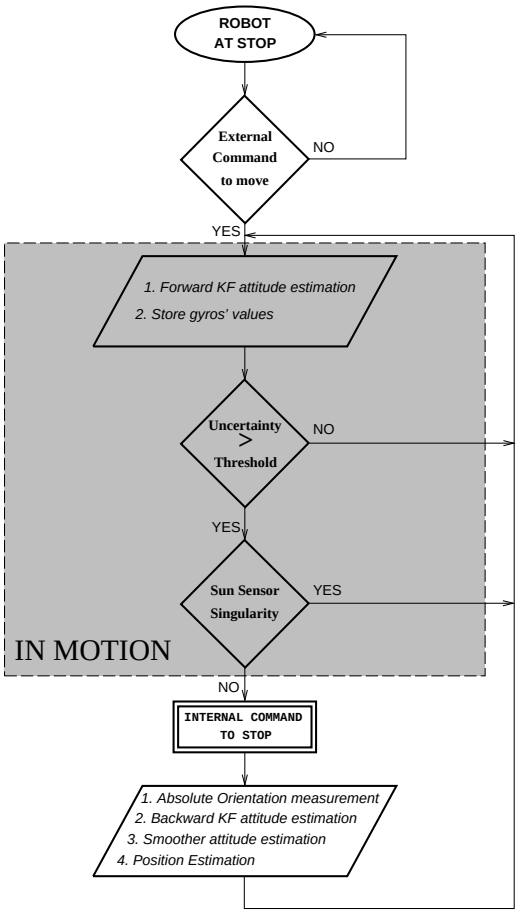

Figure 4.3: *Algorithm Flow Chart: The robot is idle until it is commanded to move (by a task planner or externally). While it is in motion the forward Kalman filter processes the information provided by the gyros and produces (real-time) a first approximation of the attitude estimate. The covariance of this estimate is calculated within the filter in every cycle of it and when it exceeds a preset threshold the robot is forced to stop. Then an absolute orientation measurement is acquired using the sun sensor and the triaxial accelerometer. A backward estimation is performed (off-line). Its results are combined with the ones from the forward filter within the smoother (off-line). Finally the position is being estimated (off-line) using the new (smoothed) attitude estimate for each instance of the trajectory.*

before an absolute measurement. In this way, we are able to sustain uniformly lower levels of uncertainty throughout the duration of the motion.

We propose to apply these ideas to a rover on the surface of Mars. As in almost every outdoors environment 3D localization is required for the robot to position itself accurately with respect to a known initial position (i.e. landing site). The assumption of flat and horizontal ground, frequently used for indoor applications, fails dramatically. When the rover moves on slopes (inclined planes) precise orientation is needed to correct the position estimates. Neglecting the roll and pitch information, not only degrades the localization accuracy but can also endanger the completion of the mission. All terrain navigation requires estimates of the roll and pitch angles to ensure vehicle safety. In addition, proper interpretation of external sensor data (e.g. vision) needs attitude information in order to construct an accurate model of the environment. The potential sensors on-board the rover are three gyroscopes, a sun sensor and an accelerometer. As aforementioned, when the vehicle moves on the cratered, rocky slopes of the planet the gyros are the only source of attitude information. When it stops, the accelerometer measures solely the gravitational acceleration and thus determines the unit vector of the local vertical (roll and pitch information). Simultaneously, the sun sensor locates the position of the sun in the sky and thus specifies the heading of the vehicle (yaw information). This is the absolute orientation of the vehicle and it is available only at stop.

The proposed method is summarized in the following algorithm (Figure 4.3): The vehicle is commanded to move for some time. A (forward) Indirect Kalman filter processes the incoming gyro values (angular velocities) and provides an initial real-time attitude estimate. The gyro values are stored for later processing. The covariance of the estimate is monitored and when it becomes unreliable the vehicle stops. Then, another absolute orientation measurement is collected and the smoother is activated. The backward filter uses the last (absolute) measurement and propagates the orientation estimate backwards using the stored values of angular velocity measurements. Finally, the resulting estimates are combined with the previous ones (from the forward filter) and the smoothed attitude estimate is passed to the position estimating module.

This new method is not limited to robots used for extra-terrestrial exploration. It can also be applied to other autonomous vehicles equipped with an equivalent set of sensors.

4.2.2 Smoother Based Position Estimation: Enhanced Mapping

The focus here is in the case of robots that receive information for their location intermittently. We propose to apply Kalman smoothing of the position estimates in order to reconstruct more precisely the trajectory of the vehicle between two locations of known position.

We intend to follow a similar approach to that presented in Section 3.3.3. Two Kalman filters have to be involved, one of them processing data forward in time and the other one backward. The first will provide position estimates starting from the absolute position measurement at the beginning of some time interval. The second will provide position estimates for the same time interval but will start from the next absolute position measurement (at the end of the time interval). A smoother will combine the position estimates of the forward and the backward Kalman filter to provide estimates of the state at every time as if all the measurements during the entire smoothing interval were available. The covariance of the estimate from the smoother $P_{smoother}$ is given by:

$$P_{smoother}^{-1} = P_f^{-1} + P_b^{-1} \tag{4.8}$$

where P_f is the covariance of the position estimate from the forward Kalman filter and P_b is the covariance of the position estimate from the backward Kalman filter. The equation for the smoothed position estimate $\hat{x}_{smoother}$ is

$$P_{smoother}^{-1}\hat{x}_{smoother} = P_f^{-1}\hat{x}_f + P_b^{-1}\hat{x}_b, \tag{4.9}$$

where $\hat{x}_f$ is the position estimate from the forward filter and $\hat{x}_b$ is the position estimate from the backward filter.

Equation (4.9) demonstrates how the states are optimally combined to produce the smoothed position estimate. Each of the covariance matrices represents the uncertainty of the estimate. The inverse of it represents the certainty or the information contained in this estimate. The higher the uncertainty, the larger the covariance matrix, the smaller the information matrix, the smaller the contribution of this filter to the total estimate. In other words the previous equation weighs each of the available estimates (from the forward and the backward filter) according to their certainty. The result is the optimal estimate given that all the measurements of the time interval of

smoothing are available at once.

This capability of a robot to reconstruct parts of its trajectory based on sparse position measurements will be beneficial for mapping tasks in partially known environments. When the position of the robot is given for certain locations only, the smoothing of the in-between position estimates will increase the accuracy of the mapping process.

Appendix A

Kalman filter formulation

Here it is assumed that the robot moves on a plane. The quantities of interest are the first two moments (mean and variance) of the x, y position and ϕ, the orientation of the vehicle.

A.1 Kinematic model

The kinematic model used is that of a three wheeled planar vehicle which has the steering at the rear wheel shown in Figure A.1. Under the Ackermann steering assumption the equations describing the kinematic model in the discrete time domain are:

$$x_{k+1} = x_k + V_k \ \Delta T \ \cos(\dot{\phi}_k \ \Delta T + \phi_k) \tag{A.1}$$

$$y_{k+1} = y_k + V_k \ \Delta T \ \sin(\dot{\phi}_k \ \Delta T + \phi_k) \tag{A.2}$$

$$\phi_{k+1} = \phi_k + \dot{\phi}_k \ \Delta T \tag{A.3}$$

where (x_{k+1}, y_{k+1}) and (x_k, y_k) is the current and previous position of the vehicle, ϕ_{k+1} and ϕ_k is the current and previous orientation, v_k is the linear velocity, $\dot{\phi}_k$ is the angular velocity, and ΔT is the time increment. Figures A.1 and A.2 show the various quantities of interest and the associated reference frames.

The robot is assumed to move on the plane with constant linear and rotational velocity:

$$\dot{V} = 0, \quad \ddot{\phi} = 0 \tag{A.4}$$

where $\dot{V}$ is the linear acceleration and $\ddot{\phi}$ is the rotational acceleration of the vehicle.

A.2 Sensors

We assume that all the measurements are corrupted by additive Gaussian noise and they are provided by:

1) *Two encoders* mounted on the two front wheels which measure the corresponding velocities (z_{V1}, z_{V2}). The associated noise n_{V_1} and n_{V_2} for each of the two encoders, is assumed to be zero-mean, white, Gaussian with covariance $\sigma^2_{V_1}$ and $\sigma^2_{V_2}$:

$$E[n_{V_1}] = 0, \quad E[n_{V_1}\ n_{V_1}] = \sigma^2_{V_1} \tag{A.5}$$

$$E[n_{V_2}] = 0, \quad E[n_{V_2}\ n_{V_2}] = \sigma^2_{V_1} \tag{A.6}$$

Instead of using the encoder measurements directly, we incorporate their information in *two* pseudo-measurements of the (total) linear velocity z_V and angular velocity $z_{\dot{\phi}}$:

$$z_V = \frac{z_{V_1} + z_{V_2}}{2}, \quad z_{\dot{\phi}} = \frac{z_{V_2} - z_{V_1}}{a} \tag{A.7}$$

where a is the distance between the two wheels. The noise associated with the two pseudo-measurements is:

$$n_V = \frac{n_{V_1} + n_{V_2}}{2}, \quad n_{\dot{\phi}} = \frac{n_{V_2} - n_{V_1}}{a} \tag{A.8}$$

with mean value:

$$E[n_V] = 0, \quad E\left[n_{\dot{\phi}}\right] = 0 \tag{A.9}$$

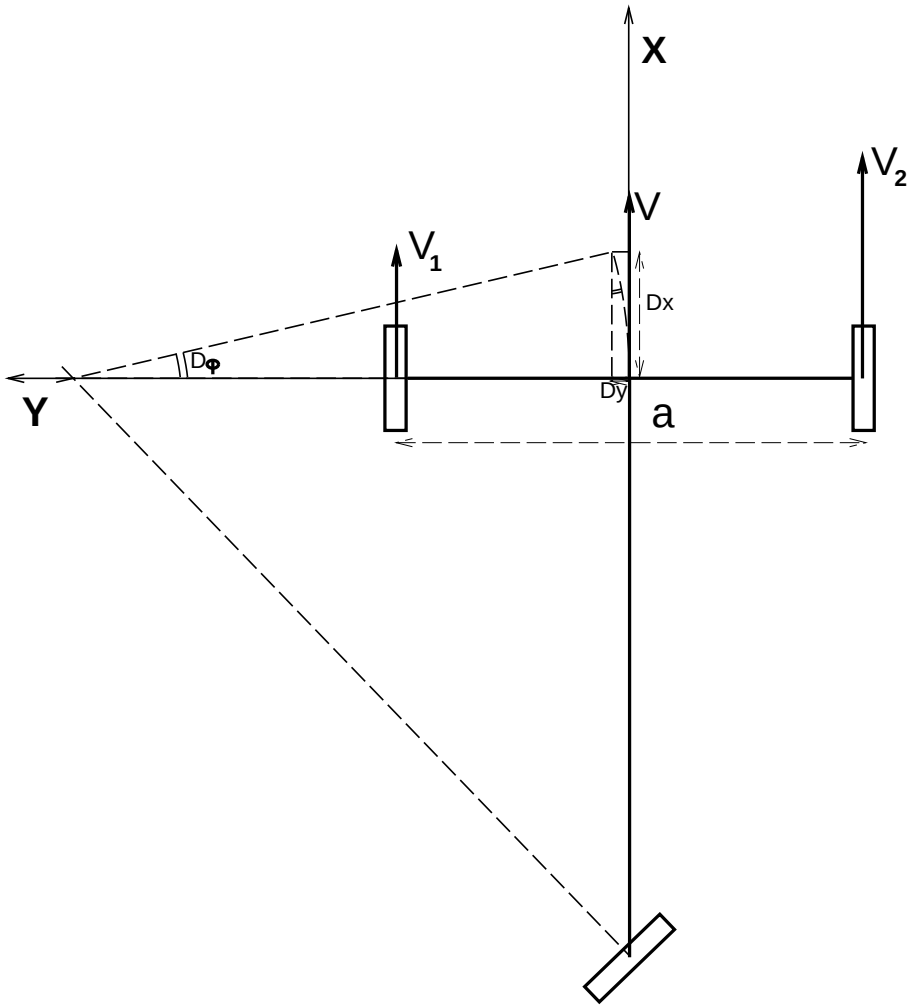

Figure A.1: *Planar model of the vehicle.*

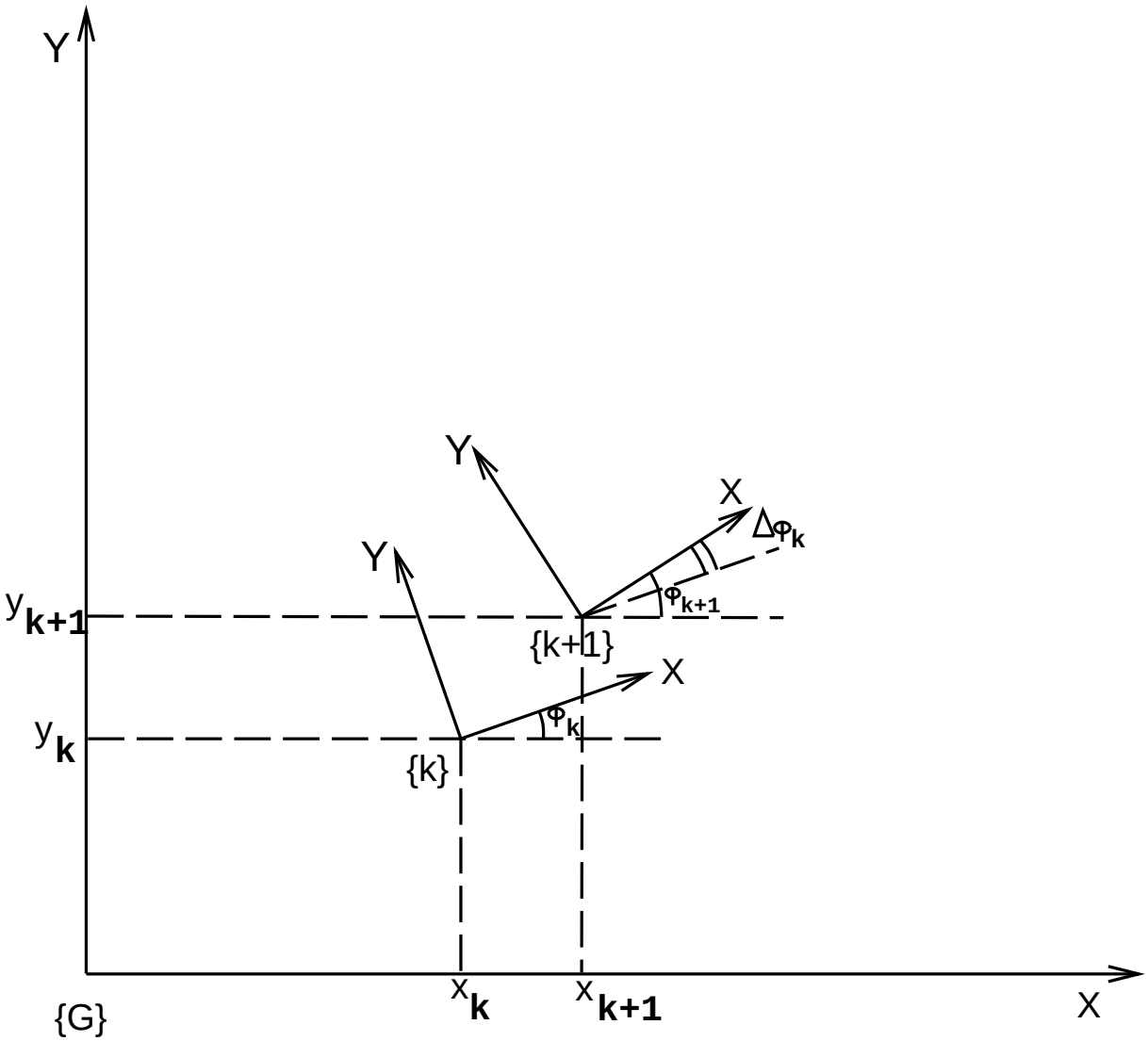

Figure A.2: *Consecutive translations and rotations of the vehicle.*

and covariance:[1]

$$E\left[n_V \; n_V\right] = \frac{\sigma_{V_1}^2 + \sigma_{V_2}^2}{2}, \quad E\left[n_{\dot{\phi}} \; n_{\dot{\phi}}\right] = \frac{\sigma_{V_1}^2 + \sigma_{V_2}^2}{a} \tag{A.10}$$

2) *A gyroscope* which gives the angular velocity z_ω of the vehicle on the axis perpendicular to the plane of motion. The associated noise n_ω is assumed zero-mean, white, Gaussian with covariance σ_ω^2 :

$$E\left[n_\omega\right] = 0, \quad E\left[n_\omega \; n_\omega\right] = \sigma_\omega^2 \tag{A.11}$$

A.3 Discrete Extended Kalman Filter

The kinematic model of the vehicle is non-linear, hence we use the Extended Kalman Filter. The EKF uses the system (kinematic) model and the measurement model. The system model describes how the vehicle's state $\mathbf{x}(k) = \left[\begin{array}{ccccc} x_k & y_k & V_k & \phi_k & \dot{\phi}_k \end{array}\right]^T$ changes with time in the presence of noise $\mathbf{v}(k)$:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{v}(k), \tag{A.12}$$

where $\mathbf{f}$ is the non-linear transition function and $\mathbf{v}(k) \sim N(\mathbf{0}, \mathbf{Q}(k))$. The noise, as we have already mentioned, is assumed to be white, zero-mean Gaussian with variance $\mathbf{Q}(k)$.

The measurement model relates the sensor observations $\mathbf{z}(k) = \left[\begin{array}{ccc} z_V & z_{\dot{\phi}} & z_\omega \end{array}\right]^T$ to the state of the system and has the following form:

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{w}(k), \tag{A.13}$$

where $\mathbf{h}$ is, in general, a non-linear function and $\mathbf{w}(k) \sim N(\mathbf{0}, \mathbf{R}(k))$. We assume that the measurements are corrupted by additive, white, zero-mean Gaussian noise with variance $\mathbf{R}(k)$.

[1] Although the linear velocity pseudo-measurement and the angular velocity pseudo-measurement are derived using the independent measurements of the velocities of the two wheels, the noise associated with them is not independent:

$$E\left[n_V \; n_{\dot{\phi}}\right] = E\left[n_{\dot{\phi}} \; n_V\right] = \frac{\sigma_{V_2}^2 - \sigma_{V_1}^2}{2a}$$

The estimation algorithm produces an estimate of the state $\hat{\mathbf{x}}(k+1,k+1)$ of the system at time step $k+1$ based on the previously updated estimate of the state $\hat{\mathbf{x}}(k,k)$ and the observations $\mathbf{z}(k+1)$. The basic steps of the algorithm are: 1) Prediction, 2) Measurement, and 3) Estimation update. We state the Kalman filter equations without derivation and we refer to [43], [45] for proofs and details.

A.3.1 Prediction

We predict the new state $\hat{\mathbf{x}}(k+1,k)$ of the system using the system model at time step $k+1$:

$$\hat{\mathbf{x}}(k+1,k) = \mathbf{f}(\hat{\mathbf{x}}(k,k)). \tag{A.14}$$

and the covariance $\mathbf{P}(k+1,k)$ associated with this prediction:

$$\mathbf{P}(k+1,k) = \mathbf{F}(k)\mathbf{P}(k,k)\mathbf{F}^T(k) + \mathbf{Q}(k), \tag{A.15}$$

where $\mathbf{F}(k)$ is the Jacobian of $\mathbf{f}$ obtained by linearizing about the updated state estimate $\hat{\mathbf{x}}(k,k)$:

$$\mathbf{F}(k) = \nabla\mathbf{f}(\hat{\mathbf{x}}(k,k)). \tag{A.16}$$

In our case using Equations (A.1) to (A.4) we have:

$$\mathbf{F}(k) = \begin{bmatrix} 1 & 0 & \Delta T\ cos(\dot{\phi}_k\ \Delta T + \phi_k) & -V_k\ \Delta T\ sin(\dot{\phi}_k\ \Delta T + \phi_k) & -V_k\ \Delta T^2\ sin(\dot{\phi}_k\ \Delta T + \phi_k) \\ 0 & 1 & \Delta T\ sin(\dot{\phi}_k\ \Delta T + \phi_k) & V_k\ \Delta T\ cos(\dot{\phi}_k\ \Delta T + \phi_k) & V_k\ \Delta T^2\ cos(\dot{\phi}_k\ \Delta T + \phi_k) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.17}$$

Finally we compute the predicted measurements using the updated state estimate $\hat{\mathbf{x}}(k,k)$:

$$\hat{\mathbf{z}}(k+1) = \mathbf{h}(\hat{\mathbf{x}}(k+1,k)). \tag{A.18}$$

A.3.2 Measurements

In this step we get the actual measurements $\mathbf{z}(k+1)$ and we compare them to the predicted ones $\hat{\mathbf{z}}(k+1)$. The difference between them is the measurement residual (or innovation):

$$\mathbf{r}(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1). \tag{A.19}$$

At this point we calculate the covariance of the residual:

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1,k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1), \tag{A.20}$$

where $\mathbf{H}(k+1)$ is the Jacobian of $\mathbf{h}$ obtained by linearizing about the state estimate $\hat{\mathbf{x}}(k+1,k)$:

$$\mathbf{H}(k+1) = \nabla \mathbf{h}(\hat{\mathbf{x}}(k+1,k)) \tag{A.21}$$

and which for our case is:

$$\mathbf{H}(k+1) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.22}$$

A.3.3 Estimation update

In the last step of the estimation algorithm we use the measurement residual $\mathbf{r}(k+1)$ to correct the state prediction $\hat{\mathbf{x}}(k+1,k)$ and thus compute the updated state estimate $\hat{\mathbf{x}}(k+1,k+1)$. In order to do that we calculate the Kalman gain:

$$\mathbf{W}(k+1) = \mathbf{P}(k+1,k)\mathbf{H}^T(k+1)\mathbf{S}^{-1}(k+1), \tag{A.23}$$

and then we update the state prediction:

$$\hat{\mathbf{x}}(k+1,k+1) = \hat{\mathbf{x}}(k+1,k) + \mathbf{W}(k+1)\mathbf{r}(k+1). \tag{A.24}$$

Finally we compute the associated covariance:

$$\mathbf{P}(k+1,k+1) = \mathbf{P}(k+1,k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1). \tag{A.25}$$

This will be used as the new state covariance in the next iteration of the estimation algorithm.

Bibliography

[1] R. O. Allen and D. H. Chang. Performance testing of the systron donner quartz gyro. JPL Engineering Memorandum, EM #343-1297, January 5, 1993.

[2] D. Avis and H. Imai. Locating a robot with angle measurements. *Journal of Symbolic Computation*, 10(3-4):311–326, Sept.-Oct. 1990.

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

[4] B. Barshan and H. F. Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342, June 1995.

[5] E. T. Baumgartner and S. B. Skaar. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–501, March 1994.

[6] M. Betke and L. Gurvits. Mobile robot localization using landmarks. *IEEE Transactions on Robotics and Automation*, 13(2):251–263, April 1997.

[7] Ph. Bonnifait and G. Garcia. A multisensor localization algorithm for mobile robots and it's real-time experimental validation. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 1395–1400, 1996.

[8] J. Borenstein and L. Feng. Correction of systematic odometry errors in mobile robots. In *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, pages 569–574, 1995.

[9] J. Borenstein and L. Feng. Gyrodometry: A new method for combining data from gyros and odometry in mobile robots. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 423–428, 1996.

[10] M. Collett, T. S. Collett, S. Bisch, and R. Wehner. Local and global vectors in desert ant navigation. *Nature*, 394(6690):269–272, 1998.

[11] S. Cooper and H. F. Durrant-Whyte. A frequency response method for multi-sensor high-speed navigation systems. In *Proceedings of the 1994 IEEE Conference on Multisensor Fusion and Integration Systems*, 1994.

[12] I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, April 1991.

[13] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

[14] F. Cozman and E. Krotkov. Robot localization using a computer vision sextant. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 106–111, Nagoya, Japan, 21-27 May 1995.

[15] F. Cozman and E. Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, volume 3, pages 2452–2457, New York, NY, April 1997.

[16] T. Denmeade. A pioneer's journey into the sarcophagus. *Nuclear Engineering International*, 43(524):18–20, March 1998.

[17] H. F. Durrant-Whyte. Where am i? a tutorial on mobile vehicle localization. *Industrial Robot*, 21(2):11–16, 1994.

[18] A. Dushman. On gyro drift models and their evaluation. *IRE Transactions on Aerospace and Navigational Electronics*, ANE-9:230–234, Dec. 1962.

[19] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989.

[20] R. Epstein and N. Kanwisher. A cortical representation of the local visual environment. *Nature*, 392(6676):598–601, 1998.

[21] R. L. Farrenkopf. Analytic steady-state accuracy solutions for two common spacecraft estimators. *Journal of Guidance and Control*, 1:282–284, July-Aug. 1978.

[22] A.M. Flynn. Combining sonar and infrared sensors for mobile robot navigation. *Internation Journal of Robotics Research*, 7(6):5–14, Dec. 1988.

[23] Y. Fuke and E. Krotkov. Dead reckoning for a lunar rover on uneven terrain. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 411–416, 1996.

[24] A. Gelb, editor. *Applied Optimal Estimation*, chapter 5. M.I.T. Press, 1994.

[25] C. H. Greene and R. G. Cook. Landmark geometry and identity controls spatial navigation in rats. *Animal Learning & Behavior*, 25(3):312–323, 1997.

[26] M. S. Grewal and A. P. Andrews. *Kalman Filtering, Theory and Practice*. Prentice Hall, 1993.

[27] W. E. L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):469–482, July 1987.

[28] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *In Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 736–743, Victoria, BC, Canada, Oct. 13-17 1998.

[29] N. Hager, A. Cypher, and D. C. Smith. Cocoa at the visual programming challenge 1997. *Computing*, 9(2):151–169, April 1998.

[30] R. L. Hammon. An application of random process theory to gyro drift analysis. *IRE Transactions on Aeronautical and Navigational Electronics*, ANE-7:230–234, Dec. 1962.

[31] U. D. Hanebeck and G. Schmidt. Set theoretic localization of fast mobilerobots using an angle measurement. In *n Proceedings of the 1996 International Conference on Robotics and Automation*, pages 1387–1394, Minneapolis, MN, April, 22-28 1996.

[32] S. Hayati, R. Volpe, P. Backes, J. Balaram, and R. Welch. Microrover research for exploration of mars. In *Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics*, University of Wisconsin, Madison, August 1-2 1996.

[33] M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1):110–119, Jan. 1997.

[34] A. Kelly. A 3d state space formulation of a navigation kalman filter for autonomous vehicles. Technical report, CMU, CMU-RI-TR-94-19.

[35] A. D. King. Characterization of gyro in run drift. In *Proceedings of the 1984 Symposium on Gyro Technology*, Sept. 1984.

[36] C. R. Kochakian. Time-domain uncertainty charts (green charts): A tool for validating the design of imu/instrument interfaces. Technical Report P-1154, The Charles Stark Draper Laboratory, Aug. 1980.

[37] E. Krotkov. Mobile robot localization using a single image. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, volume 2, pages 978–983, Washington, DC, 14-19 May 1989.

[38] R. Kuc and Y.-D. Di. Intelligent sensor approach to differentiating sonar reflections from corners and planes. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, pages 329–333, Amsterdam, 8-11 Dec. 1987.

[39] E. J. Lefferts and F. L. Markley. Dynamics modeling for attitude determination. *AIAA Paper 76-1910*, Aug. 1976.

[40] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[41] D. Maksarov and H. Durrant-Whyte. Mobile vehicle navigation in unknown environments: a multiple hypothesis approach. *IEE Proceedings-Control Theory and Applications*, 142(4):385–400, July 1995.

[42] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, 1992.

[43] P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 141-1 of *Mathematics in Science and Engineering*, chapter 6. Academic Press, 1979.

[44] P. S. Maybeck. *Stochastic Models, Estimation and Control*, volume 141-2 of *Mathematics in Science and Engineering*. Academic Press, 1982.

[45] J. M. Mendel. *Lessons in Digital Estimation Theory*, chapter 17. Prentice-Hall, 1987.

[46] R. R. Murphy, D. Hershberger, and G. R. Blauvelt. Learning landmark triples by experimentation. *Robotics and Autonomous Systems*, 22(3-4):377–392, Dec. 1997.

[47] J. Neira, J. Tardos, J. Horn, and G. Schmidt. Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1):76–84, Feb. 1999.

[48] G. C. Newton. Inertial guidance limitations imposed by fluctuation phenomena in gyroscopes. In *Proceedings of the IRE*, volume 48, pages 520–527, April 1960.

[49] D. C. Pauling, D. B. Jackson, and C. D. Brown. Spars algorithms and simulation results. In *Proceedings of the Symposium on Spacecraft Attitude Determination, Aerospace Corp. Rept. TR-0066 (6306)-12*, volume 1, pages 293–317, Sept-Oct 1969.

[50] J. E. Potter and W. E. Vander Velde. Optimum mixing of gyroscope and star tracker data. *Journal of Spacecraft and Rockets*, 5:536–540, May 1968.

[51] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6):843–854, Dec. 1979.

[52] S. I. Roumeliotis and G. A. Bekey. An extended kalman filter for frequent local and infrequent global sensor data fusion. In *Proceedings of the SPIE (Sensor Fusion and Decentralized Control in Autonomous Robotic Systems)*, pages 11–22, Pittsburgh, PA, Oct. 1997.

[53] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15 1999. to appear.

[54] R. E. Scheid, D. S. Bayard, J. (Bob) Balaram, and D. B. Gennery. Onboard state estimation for planetary aerobots. In *12th Lighter-Than-Air Systems Technology Conference, AIAA International Balloon Technology Conference, 14th Aerodynamic Decelerator Systems Technology Conference and Seminar*, San Francisco, CA, June 3-5 1997. AIAA.

[55] G. M. Siouris. *Aerospace Avionics Systems*, chapter 5. Academic Press, Inc., 1993.

[56] H. W. Sorenson. *Advances in Control Systems*, volume 3, chapter Kalman Filtering Techniques. Academic Press, 1966.

[57] K. Sugihara. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing*, 42(1):112–129, April 1988.

[58] K. T. Sutherland and W. B. Thompson. Inexact navigation. In *In Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 1–7, Atlanta, GA, May 2-6 1993.

[59] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, Oct. 1998.

[60] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1), Feb. 1998.

[61] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hhnel, C. Rosenberg, N. Roy, J. Schulte, and D. Shulz. Minerva: A second-generation museum tour-guide robot. In *Proceedings of the 1999 IEEE International Conference in Robotics and Automation*, Detroit, MI, 1999. to appear.

[62] N. F. Toda, J. L. Heiss, and F. H. Schlee. Spars: The system, algorithms, and test results. In *Proceedings of the Symposium on Spacecraft Attitude Determination, Aerospace Corp. Rept. TR-0066 (6306)-12*, volume 1, pages 361–370, Sept-Oct 1969.

[63] J. Vaganay, M. J. Aldon, and A. Fournier. Mobile robot attitude estimation by fusion of inertial data. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, pages 277–282, 1993.

[64] R. Wehner, B. Michel, and P. Antonsen. Visual navigation in insects: Coupling of gocentric and geocentric information. *Journal of Experimental Biology*, 199(1):129–140, 1996.

[65] J. R. Wertz, editor. *Spacecraft Attitude Determination and Control*, volume 73 of *Astrophysics and Space Science Library*. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1978.

[66] D. Wettergreen, H. Pangels, and J. Bares. Behavior-based gait execution for the dante ii walking robot. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 274–279, Pittsburgh, PA, 5-9 Aug. 1995.