August 2014

Thesis for the Degree of Masters of Science

Kalman Filter Approach for Mobile Robot Localization Fusing Ultrasonic Sensor and Laser Range Finder Measurements

Graduate School, Chosun University

Department of Control and Instrumentation Engineering

Sairah Naveed

August 2014

Thesis for the Degree of Masters of Science

Kalman Filter Approach for Mobile Robot Localization Fusing Ultrasonic Sensor and Laser Range Finder Measurements

Graduate School, Chosun University

Department of Control and Instrumentation Engineering

Sairah Naveed

초음파 센서와 레이저 거리 센서 측정 융합을 이용한 이동로봇 위치 추정 칼만필터 방법

Kalman Filter Approach for Mobile Robot Localization Fusing Ultrasonic Sensor and Laser Range Finder Measurements

2014년 8 월 25 일

조선대학교 대학원

제어계측공학과

사이라흐 나비드

초음파 센서와 레이저 거리 센서 측정 융합을 이용한 이동로봇 위치 추정 칼만필터 방법

지도교수 고낙용

이 논문을 석사학위신청 논문으로 제출함

2014년 4 월

조선대학교 대학원

제어계측공학과

사이라흐 나비드

사이라흐 나비드
석사학위논문을 인준함

위원장 조선대학교 교수 이진이 인

위 원 조선대학교 교수 곽근창 인

위 원 조선대학교 교수 고낙용 인

2014년 5월

조선대학교 대학원

Acknowledgement

I would like to express the deepest gratitude to my advisor, Professor, Dr. Nak Yong Ko for providing me an excellent scholarship support and highly professional atmosphere for doing research in Korea. Without his guidance and persistent help this thesis would not have been possible.

I am particularly grateful to my lab members Dr. Sung Woo Noh, and Mr. Seokki Jeong for being my helping hand. Their cooperation in carrying out my research experiments is really worthy.

I would like to pay my gratitude to my parents, their insightful opinions were an enormous help to me. I am also thankful to my siblings who were always there to encourage me with their best wishes. I am deeply thankful to my loving husband Muhammad Naveed Iqbal Qureshi. Without his constructive suggestions, generous support, and fragile care this work would not have been accomplished on time.

Contents

List of Figures

List of Tables

초록

초음파 센서와 레이저 거리 센서 측정 융합을 이용한 이동로봇 위치 추정 칼만필터 방법

나비드 사이라흐
지도교수 : 고낙용 교수
학과 : 제어계측공학과
학교 : 조선대학교

지난 몇십년 동안 이동로봇 위치추정은 연구자나 로봇 엔지니어들로부터 많은 관심을 받아왔다. 이동로봇이 어떤 환경 안에서 신뢰성 있는 일을 하기 위해서는 로봇은 자신의 위치를 알수 있어야한다. 행성탐사, 구출, 그리고 감시등과 같은 작업에서는 미지의 환경에서 로봇이 자신의 위치를 효율적으로 알아야만 한다. 이러한 작업을 위해서는 작업공간에서의 지도 작성과 함께 로봇은 작성된 지도안에서의 자신의 위치를 알 수 있어야한다. 아주 복잡하고 다양한 환경들이 많이 있는데 예를 들면, 많은 물건들이 놓여있고 여러 가지 제한적인 상황에서 로봇이 운행을 해야 하는데 이러한 경우에는 로봇의 운행이 아주 어려운 문제가 된다. 게다가 로봇 플랫폼과 센서는 시스템마다 서로가 아주 많이 다르다. 그래서 그런 만큼 감지와 조종 하는 게 굉장히 어려워진다. 미지의 환경에서 로봇의 주행정확도를 높이기 위

해서 다양한 센서들로 부터의 정보를 융합하여야 할 필요가 있다. 물건들을 탐지하는 다양한 센서들의 정보를 융합해서 맵을 정확히 만들어내고 로봇의 위치를 정확하게 추정해내는 것들이 필요하다.

본 논문에서는 무향칼만필터 초음파센서 그리고 레이저 거리 센서들을 사용한 이동로봇의 위치추정방법을 제안한다. 제안된 무향 칼만 필터 위치추정 알고리즘은 특징점 대응 값이 주어지지 않은 상황에서 동작한다. 제안된 알고리즘은 로봇의 모션모델, 센서 측정모델을 필요로 하고 아울러 실내에서의 특징점이나 랜드 마크들에 대한 특징점 대응 값을 계산할 것을 필요로 한다. 특징점 대응 값이라는 것은 측정된 센서 정보가 어떤 특징점에 대응되는지를 알아내는 것을 이야기 한다. 측정된 센서정보의 특징점 대응 값은 유사성이 있는 최대화된 추정들을 사용해서 계산된다. 유사성이 있는 최대화된 추정이라는 것은 센서에 의해 검출된 특징점과 지도에 나타난 특징점의 유사성을 가장 최대화 시키도록 하는 특징점 대응 값을 구한다. 제안된 방법은 초음파센서와 레이저거리센서정보를 융합하여서 실내에서의 로봇 위치추정 성능을 향상시켰다.

ABSTRACT

Kalman Filter Approach for Mobile Robot Localization Fusing Ultrasonic Sensor and Laser Range Finder Measurements

Sairah Naveed
Advisor: Prof. Nak Yong Ko, Ph. D.
Dept. of Control and Instrumentation Engg.,
Graduate School of Chosun University

Mobile robot localization has attracted significant attention from the researchers and robotic engineers in the past few decades. To reliably navigate in an environment, a mobile robot must know where it is. Many robotic applications, such as planetary exploration, rescue, and surveillance etc., requires efficient robot localization within unknown environments. This issue involves building a map of the work space and localizing itself with in such map. There are varieties of complex environment where robot need to navigate. One of the issue of localization is to find the location of the robot in partially or completely unknown environment, which makes the problem more difficult to solve. Moreover, physical platforms and sensor suites vary significantly from system to system and complicating the task of sensing and control. In order to increase the accuracy when robot navigates into partially or completely unknown environment, it is necessary to integrate sensing information from distinct sensors for detecting objects and translate the different sensory inputs into reliable robot position estimates with respect to the environment.

We propose the approach for mobile robot localization in a partially unknown indoor environment. The propose approach use Unscented Kalman filter, ultrasonic sensor system and laser range finder. We implement UKF localization algorithm fusing the measurements of ultrasonic sensor and laser range finder. This algorithm relies on calculating correspondence of data association for beacons. The data association is computed through maximum likelihood estimator which determines the most likely value of the correspondence variable and the consider it as a true value. While the sensor fusion in this context is for integrating the data from ultrasonic sensor and laser range finder to improve the robot localization.

1. Introduction

Frequently, robots find themselves asking this question “Where Am I?”. Knowing your location, and being able to navigate to other locations, is extremely important for mobile robots. Robot localization addresses the pre requisite issue in field of robotics. It is an act of determining the pose of the robot relative to a map of the environment. The reliable localization is necessary for most robotic applications. It is imperative for a mobile robot to perform the given missions successfully. The most common form of localization and navigation is to use a map, combined with sensor readings and some form of motion feedback [1]. As localization is always a precursor to navigation. The issues associated to localization of mobile robot break down into three main conditions:

1. Known map and known localization: This is the easiest type of problem. The robot knows the map, and its current location, so it can easily navigate to a new location.

2. Known map and unknown localization: The robot knows what the world looks like, but it does not know it current location. Once it finds out where it is, it just really does the condition 1 above.

3. Unknown map and unknown localization: This is the hardest of all problems. This is known as “Simultaneous localization and mapping”. After decent amount of time, robot should have the pretty good map, putting robot in condition 2 above.

Generally, localization and navigation of mobile robots are done in an iterative way. The iteration procedure is underlying structure from the fields of recursive state estimation and in particular markov chains. A robot will typically do the followings, while maintaining its knowledge of its current state in memory:

1. Move some distance, which typically introduces some error in its known

position. This is error termed as motion uncertainty.

2. Take some sensor measurements, which should help robot to reduce the error in its known position. Frequently sensors introduce errors during its measurement these errors known as measurement uncertainties or measurement noise.

3. Localize itself against a map using its knowledge of how far it might have moved and its sensor readings.

Motion uncertainty and measurement noise are considered as the control parameters. These control parameters mainly effects location estimation with respect to the odometery of the robot and intrinsic behavior of the sensor systems. Not every localization condition which discussed above is equally hard. But they associated with localization problems which characterized by the type of knowledge that is available initially and at run time. There are three types of problem encounters in mobile robot localization shown in Table 1.

Type	**Definition**
Position tracking	Assumes that initial pose of the robot is known
Global localization	Assumes the initial pose of the robot in unknown
kidnapped robot	During its operation, the robot kidnapped and teleport to some other location

Table 1.1 Taxonomy of localization problems

A variety of Localization methods for mobile robots in indoor environment have been researched. Dead-reckoning method has long been used for localization. This method estimates robot pose by using robot motion information received from the wheel encoders over time form its starting position. Other method used beacons [2] placed at known position in the environment. The range sensors used ultrasonic signal or radio signal to determine the distance between the robot and the beacons. Laser range finders also served for mobile robot position estimation methods [3, 4]. They worked as active range sensors and the most popular sensors in mobile robotics. They worked on the laser beams which measures the range from robot to objects. Least square methods or filtering methods uses the range data to estimate the position of the mobile robot.

1.1 Brief View of Probabilistic Approaches

The probabilistic approaches are the most promising candidate which provides the comprehensive and real-time solution to robot localization problems. Kalman filter (KF) [5] is one of the most popular approaches to solve localization problem. All the Kalman filters uses Gaussian distribution for estimation. There are several variants of Kalman filter among them most popular are extended Kalman filter (EKF) [6, 7], and unscented Kalman filter (UKF) [8, 9]. Pure Kalman filters only apply to linear Gaussian problems. To deal with non-linear and non-Gaussian problems EKF and UKF are employed. EKF approximates the non-linearity with first-order Taylor series. EKF uses analytical linearization approach involving Jacobian matrices. Until now extensive researches have been reported employing EKF to address several aspects of robot localization. EKF suffers considerable hurdles and lack to localize the robot, in case of real data, which is typically involving elements of non-Gaussianity and higher order non-linearities. Unscented Kalman filter (UKF) approximates the non-linearity up to more degrees than EKF. UKF uses statistical approach called the unscented transform which uses set of samples called as sigma points for estimation. UKF is favorable when the system is highly non-linear and the derivation of Jacobian is not feasible. The particle filter (PF) [10, 11] is an alternative implementation for non-linear and non-Gaussian problems. This filter is not bound to Gaussian distribution it can approximate any kind of probabilistic distribution of practical importance. The Monte Carlo method is a PF based method uses a set of samples called as particles to depict the probabilistic aspect of robot location. In other words rather than approximating the probabilistic distribution of in parametric forms as is the case of Kfs, it describes the probabilistic distribution as it is using particles. But the number of particles trade-off the efficiency of the computation and the computational resources which is necessary to run MCL.

UKF and PF have some resemblance with each other in some aspects. Both the filter uses samples to represent the distribution of estimation. In case of UKF the samples called as sigma points, while in case of PF the samples are known as particles. However the UKF selects the sigma points deterministically while the PF selects particles stochatically. PF can deal with any probabilistic distribution, but UKF uses only Gaussian distribution which is one of the essential feature of KF approach. In some case where fast computations require UKF considered being more feasible than PF. Beacause in case of PF increasing in number of particles decreases the computational time. Besides favorably the computational efficiency of UKF is comparable to EKF but better than PF.

1.2 Related Work

Although much progress has been made which focused on mobile robot localization using probabilistic approaches which are capable of solving localization problems and make use of several sensors such as range-bearing sensor, ultrasonic sensor, vision sensor, laser range finder, and RFID sensor etc.

Giuseppe Cotugno *et al*. compares the two algorithms based on Extended Kalman filter (EKF) and Unscented Kalman filter (UKF) for the mobile robot localization environment reconstruction. Their proposed algorithm has no assumption on the robot working space. The robot drives only by the measurements of ultrasonic sensors system. The basic idea behind the algorithm is only estimates the current position and orientation of robot. They implement the sensor switching activation policy which allows energy saving and still achieving the efficient tracking of the workspace [12]. Leonardo Romero *et al.* have proposed the approach for an indoor mobile robot using data fusion of ultrasonic sensor, laser range finder to learn probabilistic grid maps of an environment. [13]. Byoung-Suk *et al.* suggest ta novel scheme to solve the problem in localization using the fusion of RFID, ultrasonic sensor and wheel encoder data to reduce the uncertainty of localization system. In this scheme certain data combination set to fuse are selected according to environmental factor. [14]. Byoung-Suk *et al.* developed the RFID based mobile localization system which adopts RFID tags distributed in space. The proposed algorithm improves the localization by fusing an RFID system with an ultrasonic sensor system. This system removed the uncertainties of RFID systems by using distance data obtained from ultrasonic sensors. This system mainly based on hierarchical to estimate the position of robot using global position estimation (GPE) and local environment cognition (LEC) [15]. Ashley W. Stroupe *et al.* have proposed the method for representing, communicating and fusing distributed, noisy and uncertain observation of an object by multiple robots. The approach of this method based on Bayes's rule and Kalman filter (KF) theory enables two or more observers to achieve greater effective sensor coverage of the environment and improved the accuracy in object position estimation [16]. Muhammad. L. Anjum *et al.* implement sensor data fusion method using an Unscented Kalman filter (UKF) for mobile robot localization. They use accelerometer, gyroscope, and encoders to obtain robot motion information and UKF used as an efficient sensor fusion algorithm. This method able to compensate slip error by switching between two different UKF models built for slip or no-slip cases. This system is able to track the robot motion in various scenarios including the scenario when robot's encoder data is not reliable due to the slip occurrence [17].

Some of the recent sensor fusion approaches used in robot localization like the work of W. Haijun et al. [18] they worked on the rough set theory. On the basis of this theory infrared sensor data fusion system was built and improved attribute reduction algorithm was proposed. Although the rough set theory is a powerful tool in processing much uncertain and incomplete information systems. But fused infrared sensors data is not enough in case of complex tasks for mobile robot especially in unknown environments. L. D. Alfosono et al. [19] present the sensor fusion surrounding environment mapping for a mobile robot using mixed extended Kalman filter. In their research the workspace is totally unknown. The local information is used to give a good measurement model to complement filter. However, the lack of communication between transmission channel and the mixed filter approach encountered loss of information especially in case of more complex environment. G. G. Rigatos [20] worked on the estimation of robot position using sensor by implementing EKF and PF filter approaches and compared their performance. He showed in case of sensor fusion PF has better performance than EKF at the cost of demanding computation.

In this thesis we proposed the modified UKF localization algorithm fusing ultrasonic sensor and laser range finder measurements for indoor mobile robot localization in a partially unknown environment. The later section will describe the motivation of the proposed approach.

1.2. Motivation

Accurate and efficient mobile robot localization in an unknown environment is long awaited requirement of many robotic engineers and researchers. One of the requirements of localization is to localize the robot in a partially unknown indoor environment. Aiming at this issue we propose the approach for mobile robot localization in a partially unknown environment. We implement UKF localization algorithm fusing the measurements of ultrasonic sensor and laser range finder. The purpose of fusing two different sensor measurements is to efficiently translate the sensory inputs into reliable estimates that can be used by the robot to sense its whereabouts. The proposed approach also deals with the problem of data association in case of partially unknown environment. The robot is given with partial map information of its workspace consisting of tables, chairs on the floor, and ultrasonic beacons attached on the ceiling of the room. The environement of the floow is modeled with occupancy grid map while the ultrasonic beacons are considered as landmarks. The goal of robot is to determine its position relative to its workspace, and calculate the correspondence of range data to ultrasonic beacon while the robot position is estimated. Figure. 1 shows the method of robot localization using data association.

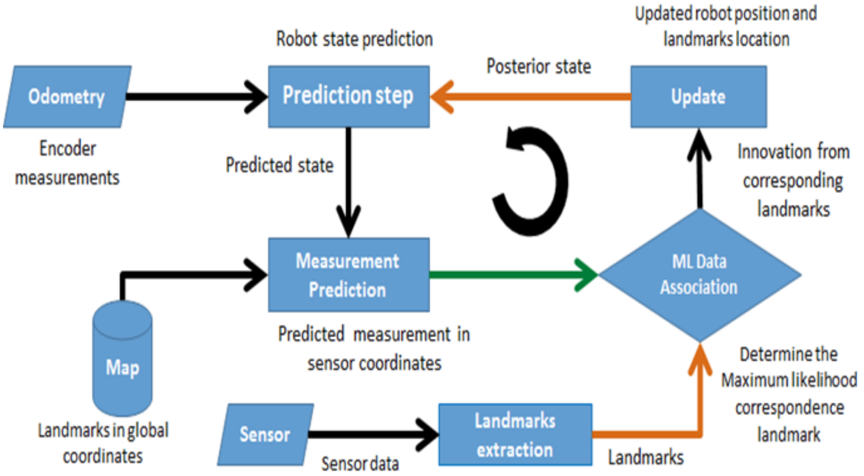

Updated robot position and landmarks location
Robot state prediction
Posterior state
Odometry
Prediction step
Update
Encoder measurements
Predicted state
Innovation from corresponding landmarks
Measurement Prediction
ML Data Association
Predicted measurement in sensor coordinates
Map
Determine the Maximum likelihood correspondence landmark
Landmarks in global coordinates
Sensor
Sensor data
Landmarks extraction
Landmarks

Figure 1.1. Flow chart of mobile robot localization with unknown correspondence black arrows shows the information, orange arrows show the procedure, and green arrow shows procedure + information

2. Approaches for Robot Localization

Mobile robot localization is the most perceptual problem in robotics. Nearly all the robotics tasks require information of the location of objects that are being manipulated. Mobile robot localization considered as an issue of coordinate transformation. Maps are described in a global coordinate system, which is independent of a robot's pose. Localization method establishes the correspondence between map coordinate and the robot's local coordinate system. Knowing the robot pose (x, y, θ) is sufficient to determine this coordinate transformation, assuming that the pose is expressed in the same coordinate as the map. Mostly robot does not possess the noise-free sensors for measuring the pose. A key difficulty arises from the fact that a single sensor measurement is usually sufficient to determine the robot pose [21]. The robot has to integrate the sensor data over time't' to determine its pose. Figure 2.1 show the graphical model for the mobile robot localization problem. The map information and sensor data are given to robot and it has determined its position relative to the map and its movements.

The two fundamental issues that have to deal with when designing a localization system is how to represent the uncertain information about the environment and the robot pose. Localization is thus to a large extent about representing to uncertainties or noise [22]. This chapter represents the different approaches to represent the environment, dedicated methods to represent the robot pose and the methods deal with robot perception i.e. how robot sense its whereabouts when it navigates into an environment.

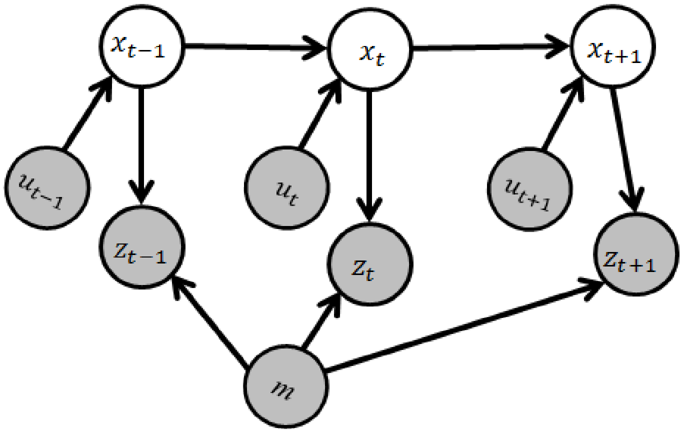


Figure 2.1. Graphical model of mobile robot localization. The values of shaded nodes are known: the map m, measurements z and the controls u. The goal of localization is to infer the robot pose *x* [21].

2.1. Environment representation

There are many ways to represent the knowledge about an environment. As it state early in Chapter 1, the robot localization deals with indoor environment. Typical examples of indoor environment are domestic office, classroom, laboratory, and office settings. The assumptions and decision made in this work do not necessarily apply to other types of environments, for example an outdoor scene.

Localization techniques have been developed for a broad set of environment representations. Two major paradigms which are identified for mapping of indoor environments are: grid-based, and landmark-based.

2.1.1. Grid-based

The key idea behind the grid based map is to represent the map of an environment as evenly spaced grids, and each grid in the map represent the presence of obstacle at that location in environment.

To create grid maps Occupancy grid algorithm are use to compute the posterior estimates for these grids in the map. These algorithms require range measurements in a number of different directions creates a map of the environment by integrating data collected over time. Such measurements readily obtained by sweeping the sensor. As the robot explores its environment, information from the range sensors sweeps combined with the information about the robot’s location to update the occupancy values for the global grid map. Thus the occupancy values for each grid cell in the grid indicate whether the cell is occupied, empty or unexplored.In V. Vareropoulos et al. they develop the method of localization and a map construction using sonar data from a sonar-based range sensor [23]. No prior knowledge about the environment is assumed. The map is constructed autonomously by the robot. S. Fheng et. al proposed the grid-based improved maximum likelihood estimation (GIMLE) method for localization in the complex indoor Wireless sensor network (WSN) environment. In this method the grid area is dynamically changed based on the position of mobile robot in each sampling time [24]. In order to make a grid-based map responsive to the changes in the environment but at the same time reduce the noise of false range sensor readings (due to multiple reflection and transducer accuracy); a weight is given to each occupancy update. There are two modes of mapping: exploration and localization. In exploration mode any change in the occupancy value will have the maximum effect, but during the localization mode each will have the reduced effect in the global map.

The occupancy grid map is chosen for this research to model the geometry of the experiement work space. We assume the information of each cell in the gird map and the grid occupancy is already known.

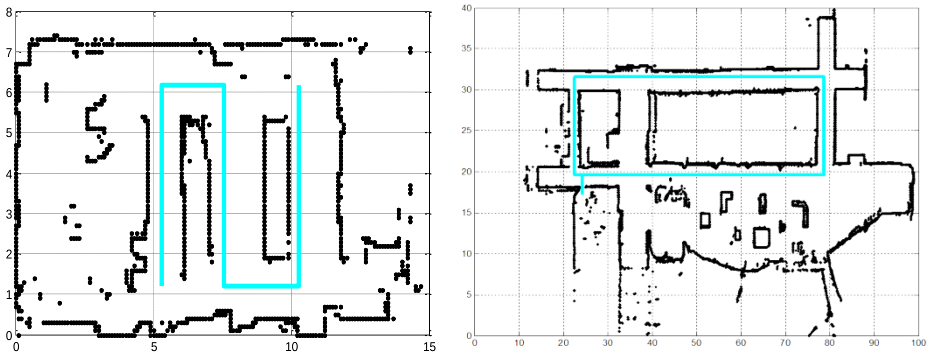


Figure 2.2. Examples of 2D gridbased map, in the above figures the blue line represent the real robot path (The left figure represent the grid based map of a work space which we use for navigation experiement in this research

2.1.2. Landmark based

The idea of trying to extract features from the environment is quiet natural; this is for example how most city maps are constructed. Landmarks are the features in the environment that robot can detect. Sensor readings from a robot are analyzed for the existence of landmark in it. Once landmarks are detected, they are matched with the priori known information of the environment of the environment to determine the position of the robot. Landmarks can be divided into active and passive landmarks [25, 26].

Active landmarks, are also called beacons, are landmarks that actively send out the location information. Active landmarks can take on the form of satellites or other radio transmitting object. A robot sense the signal sent out by the landmark to determine its position. The two closely related methods are used to determine the absolute position of the robot using active landmarks: triangulation and trilateration. Triangulation techniques are use distance and angles to three or more landmarks, while trilateration technique only uses distance. The angles and distance are then used to calculate the position and orientation of the robot.

If the landmarks do not transmit signals, the landmarks are called as passive landmarks. The robot has to actively look at those landmarks to acquire position measurements. Techniques using passive landmarks in determining the position of the robot rely on detection of those landmarks from sensor readings. The detection of the landmarks depends upon the type of senor used. For example, in detecting landmarks in images from the vision systems, image processing techniques are used. When three or more landmarks are detected it uses triangulation or trilateration techniques to detect the landmark location. Passive landmarks can either be artificial or natural and the choice of which kind of landmark to use can play the significant role in the performance of localization system.

2.2. Robot Motion

When we talk about location, pose, or position we mean the x and y coordinates and θ(heading) of the robot in global coordinate system. The robot configuration of a mobile robot is commonly describing the three variables (x, y, θ), its three-dimensional Cartesian coordinates relative to the global coordinate frame. Figure 2.3 shows the robot pose in Cartesian coordinate frame. The robot pose is the fundamental knowledge to estimate the robot's location in an environment.

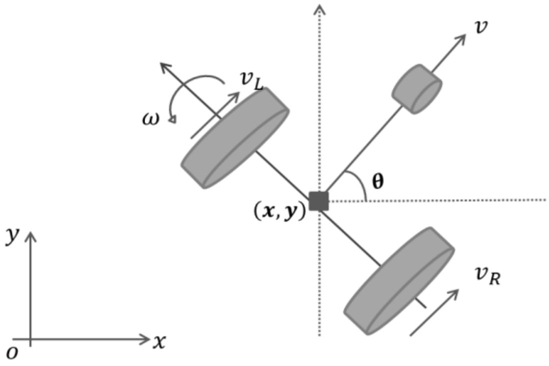


Figure 2.3. The robot poses in Cartesian coordinate frame

2.2.1. Dead reckoning

Dead reckoning is the estimation of a robot's location based on its estimated speed, direction and time of travel with respect to a previous estimate [27]. If the robot has a good initial pose estimate $X_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$ at time $t-1$, odometric information can be used to track of the pose of the robot.

The odometer sensor is used to measure distance travelled, typically by measuring the angular rotation of the wheels. The direction of travel can be measured using an electronic compass or the change in heading can be measured using gyroscope or differential odometry. These sensors are imperfect due to systematic errors such as incorrect wheel radius or gyroscope bias, and random errors such as slip between the wheels and the ground, or the effect of turning [28].

Let's consider the kinematics for an ideal noise free robot. Let $u_t = (v, \omega)^T$ denotes the control at time't'. Both velocities are kept at fixed value for the entire time interval $(t-1, t]$. Mathematically, the robot's configuration at the time step t, including the odometry is expressed as:

$$\begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}sin\theta + \frac{v}{\omega}(sin\theta + \omega\Delta t) \\ \frac{v}{\omega}cos\theta - \frac{v}{\omega}cos\theta + \omega\Delta t) \\ \omega\Delta t \end{pmatrix} \tag{2.1}$$

The Equation 2.4 [29] follows the simple trigonometry. After Δt units of time, the noise free robot has progressed $v.\Delta t$ along the circle which caused its heading direction to turn by $\omega.\Delta t$. Of course the robots cannot jump for one velocity to another, and keep velocity constant in each time interval. To compute robot motion with non-constant velocities, it is therefore common practice to use small values for Δt and to approximate the actual velocity by a constant within each time interval. The approximate final pose is then obtained by concatenating the corresponding trajectories using equation 2.1.

2.2.1.1 Motion Model

In actual, robot motion is subject to noise. The actual velocities differ from the commanded ones or measured ones. The difference between commanded and actual velocities usually model by a zero- centered variable with finite variance. More precisely, the actual velocities are given by:

$$\begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix} = \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} \varepsilon_{\alpha_1 v^2 + \alpha_2 \omega^2} \\ \varepsilon_{\alpha_3 v^2 + \alpha_4 \omega^2} \end{pmatrix} \tag{2.2}$$

Here ε_{a^2} is the zero mean error with variance a^2. Thus, the true velocity equals the commanded velocity plus some small additive error (noise). The parameters α_1 through α_4 are robot specific motion error parameters. They model the accuracy of the robot. The less accurate motion of a robot, the larger these parameters.

The motion model assumes that the robot control through two velocities, a translational velocity ‘v’ and a rotational velocity ‘ω’. Many control robots offer control interfaces where the programmer specifies velocities. Differential drives, Ackerman drives and synchro drives are controlled in this way. The transitional velocity at time‘t’ by v_{t}, and the rotational velocity by ω_{t}. Hence we have,

$$u_t = \binom{v_t}{\omega_t} \tag{2.3}$$

In this thesis we consider positive rotational velocity ω_t induce the counter clockwise turn (left turn). Positive translational velocity v_t corresponds to forward motion.

Table 2.1 shows the motion model which generates random values of v_t, and ω_t for a fixed control u_t and robot pose X_{t-1}. This algorithm accepts X_{t-1}, u_t and *motionpara* as input and predict the pose of the robot X_t at time t. u_t is the control commanded or proprioceptive motion sensing $(v,\omega)^T$. Δt is the sampling time step. The variable *motionpara* consisting of motion noise parameters $(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$. The function *sample* generates the random sample from a zero centered distribution with variance $\alpha_i v^2+\alpha_j\omega^2$ [29, 30].

```
----------------------------------------------------------
Algorithm Motion model $(u_t, X_{t-1}, motionpara)$
----------------------------------------------------------
$\hat{v} = v + sample(\alpha_1 v^2 + \alpha_2 \omega^2)$
$\hat{\omega} = \omega + sample(\alpha_3 v^2 + \alpha_4 \omega^2)$
$x_t = x_{t-1} - \frac{\hat{v}}{\hat{\omega}}(\sin\theta + \sin(\theta + \hat{\omega}\Delta t))$
$y_t = y_{t-1} + \frac{\hat{v}}{\hat{\omega}}(\cos\theta - \cos(\theta + \hat{\omega}\Delta t))$
$\theta_t = \theta_{t-1} + \hat{\omega}\Delta t$
return $X_t = (x_t, y_t, \theta_t)^T$
----------------------------------------------------------
```

Table 2.1. Algorithm of motion model

2.3. Robot Perception

Robot perception has traditionally been addressed as a passive and incidental activity, rather than active as task-directed activity. The robot has to get the idea of the environment surrounding it. The robot senses the environement by means of its exteroceptive sensors. These sensors give momentary situation information called as observation or measurments [25]. However, what robot thinks it senses is not necessary what actually the case is. Uncertainties in sensor readings make difficult in obtaining an accurate idea of the environment. To enable the reason about the perceptual capabilities of robot system, mathematical models are needed to describe the various sensors of robot.

2.3.1. Sensor Model

Sensor or measurement models describe the formation process by which sensor measurements are generated in the physical world. The specifics of the sensor model depend upon the type of sensor. Imaging sensors are best modeled by the projective geometry, sonar sensors are best modeled by describing the sound wave and its reflection on surfaces in the environment, and ultrasonic senosrs are best modeled by calculating the distance between beacons to object through ultrasonic signal.Whereas Laser range finders are modeled through raycasting technique. Probabilistic distribution methods are used to model the inherent uncertinaites in sensors. In this research we use two different range sensors and we model the noise in their measurements by Gaussian noise on the range. Table 2.2 shows the algorithm of sensor model.

```
Sensor Model ( $f_t, X_t, m$ )

$\hat{r}_t = \sqrt{(m_x - x_t)^2 + (m_y - y_t)^2}$
$q = prob(r_t - \hat{r}_t, \sigma_r)$
return $q$
```

Table 2.2. Algorithm of Sensor model

This algorithm requires an observed feature $f_t = (r_t)$, the robot pose $X_t = (x_t, y_t, \theta_t)^T$, and map m as inputs. The output q returns the weight value of noise in received range information with zero mean and and standard deviation σ_r .

2.3.2. Ultrasonic Sensor

Ultrasonic sensor system (also known as ultrasonic transceivers) is among the most popular sensors in robotics. Ultrasonic sensors have various applications in robotics i.e. to detect the movement of robot, to measure the distance between target and object, process plants, and unmanned aerial vehicle (UAV) navigation etc. They primarily work on the principle similar to radar or sonar which evaluates the attributes of a target by interpreting the echoes from radio or sound waves respectively. Because the ultrasonic sensors use high frequency sound waves rather than light for detection, they work in applications where photoelectric sensors may not. Ultrasonic sensor calculates the time interval between sending and receiving the echo to determine the distance to an object. In this research we have used USAT A105 ultrasonic sensor system developed by Korea LPS Company shown in Figure 2.4.

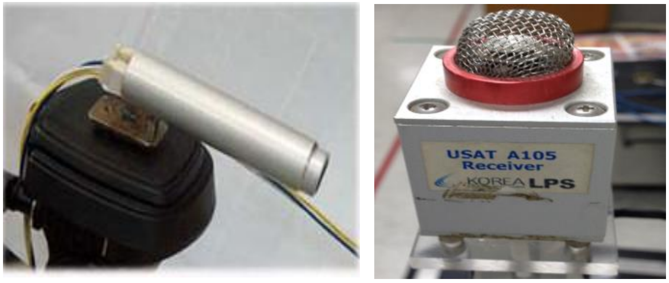


Figure 2.4. USAT A105 Ultrasonic sensor system. The left figure shows ultrasonic beacon (or transmitter) and the figure in the right side shows ultrasonic receiver

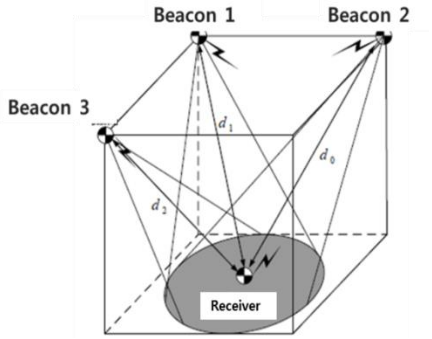


Figure 2.5. Ultrasonic sensor system environment [31]

Figure 2.5 shows the usable area for the existent beacons methods. Each beacon has fixed angle, and it usually aim at the center of its corresponding section. Also dotted lines from each beacons show the beacon’s bean angle resulting in the recognizable areas (dark parts) while there are un- recognized areas beyond the beam angles [31].

2.3.3. Laser Range Finder

Laser range finders have been in extensive use for many years for autonomous robots. They have multiple applications in mobile robotics. They can be used for object tracking, obstacle avoidance, feature extraction, map building or self-localization. Their primary usage is to measure the range to nearby objects. Range measured along a beam, considered good working model of laser range finders [32]. We have three different kinds of laser range finders available in the market. As an example three different kinds of laser range finder are shown in Figure 2.6.

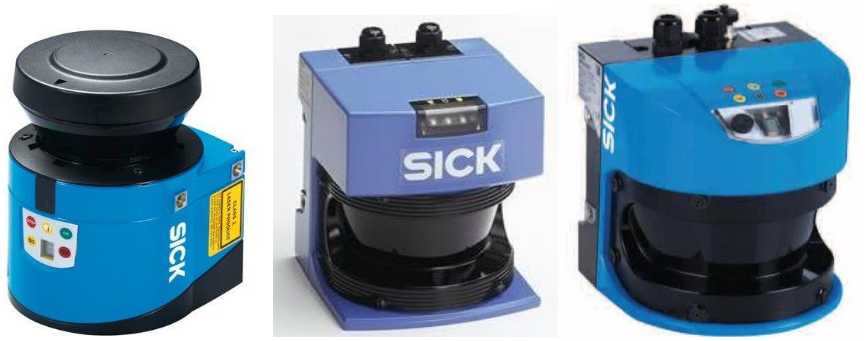


Figure 2.6. A number of laser range finders from the SICK Company (LMS Tech, 2004) The right-most scanner LMS-511 is specifically used in the reported experiments.

A laser range finder is sending out the series of M laser rays in a plane as illustrated in Figure 2.7 illustrates the principle. The left most pulse represents the riginal pulse and the time of flight is denoted by Δt. Δt is directly proportional to the distance r from the scanner to the object, r can be computed as:

$$r = c\frac{\Delta t}{2} \tag{2.4}$$

Where, *c* is the speed of light.

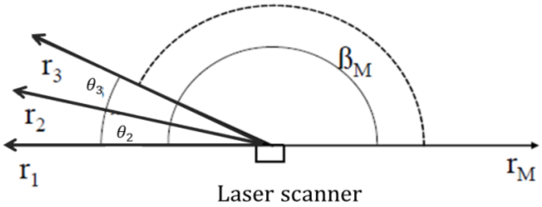


Figure 2.7. Top view of laser scanner consisting of M laser rays emitted in a plane [31]

The operates on the principle of "time of flight" between sending laser pulse in narrow beam towards the object and measure the time taken by the pulse to be reflected off the target and returned to the sensor.

2.3.2.1 Ray Casting

The robot used in this research features the laser range finder that is mounted on the robot and it sweep about 180o and computes range from its laser beams. The number of laser beams is computed mathematically using Equation 2.5

$$\frac{\phi_{LRF}}{\Delta\phi} + 1 = N_{LRF} \tag{2.5}$$

Where, ϕ_{LRF} shows total sweep, $\Delta\phi$ is the difference angle between the beams of laser, and N_{LRF} represent the total number of laser beams. In this research we use 19 laser beams in our experiment to calculate the distance from obstacles to robot. In order to calculate 19 laser beams we keep ϕ_{LRF} =180^{o} and $\Delta\phi$ = 10^{o}. The range computed from the laser range finder is computed through ray casting operation.

Ray casting determines the heading and distance to an obstacle and to discover obstacle free routes for robot to navigate. It works by tracing the path of a series of rays originating from the laser range finder current position on a map of objects. When one of these rays hits an object (such as wall) the distance is registered and the object is rendered at a size proportional to its distance. Raycasting can be applied to a robot by using the robot's global position to the place the robot on a map and then casting the rays from robot's position until they reach an obstacle on the map. This will result the distance between the robot and the object [33]. Figure 2.9 shows the example of ray casting with 360^{o} sweep.

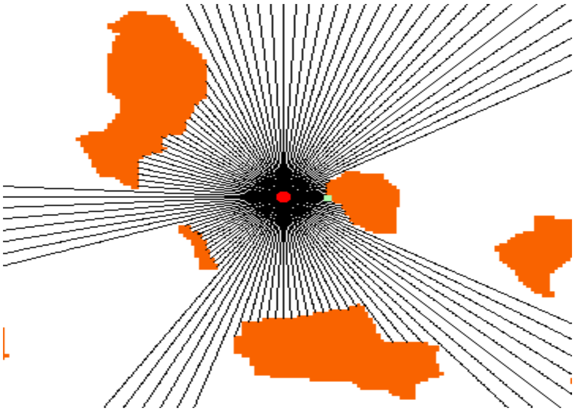


Figure 2.8. An example of ray casting with 360^{o} sweep. The ray can be seem emerging from the robot which is represented by the red circular dot, The green square just right to this dot indicates the closest occupied grid cell to the robot [33].

Table 2.3 shows the ray casting algorithm using grid map information. This algorithm requires grid map information m_{grid} robot pose X_t , laser range finder maximum range r_{Max} , grid resolution $grid\_resolution$, and the grid occupancy $grid\_occ$ as an input. The output of the raycasting algorithm will be the range computed from the laser range finder r_{LRF} . We have implemented the same raycasting algorithm in this research to compute the distance from obstacles to robot through laser range finders.

```
-----------------------------------------------------------------------------------
$\textbf{Ray Casting}\ (m_{grid}, X_t, r_{Max}, grid\_resolution, grid\_occ)$
-----------------------------------------------------------------------------------
$\Delta\phi = \frac{1}{resolution}$
$\textbf{for}\ N_{LRF} = 0\ \ to\ \ 18$
    $\phi_{raycast} = -\frac{\pi}{2} + (N_{LRF} \times \Delta\phi \times \frac{\pi}{180})$
    $\textbf{for}\ k = 0\ \ to\ \ (\frac{r_{Max}}{\Delta\phi})$
        $r_x = x_t + (\Delta\phi \times r_{LRF}\ \cos(\theta_t + \phi_{raycast}))$
        $r_y = y_t + (\Delta\phi \times r_{LRF}\ \cos(\theta_t + \phi_{raycast}))$
        $r_{x,index} = r_x \times grid\_resolution$
        $r_{y,index} = r_y \times grid\_resoltuion$
        $\textbf{if}\ (r_{x,index}\ \&\&\ r_{y,index} <= 0)$
            $r_{x,index} = r_{y,index} = 0$
        $\textbf{elseif}\ (r_{x,index}\ \&\&\ r_{y,index} >= m_{grid} \times grid\_resolution)$
            $r_{x,index} = r_{y,index} = (m_{grid} - 1) \times grid\_resolution$
        $\textbf{endif}$
        $m_{gird_x} = floor(r_{x,index} + 1)$
        $m_{gird_y} = floor(r_{y,index} + 1)$
        $\textbf{if}\ m_{grid}\ (m_{gird_x}, m_{gird_y}) > 128$
            $r_{LRF}^{[k]} = \sqrt{(x_t - r_x)^2 + (y_t - r_y)^2}$
        $\textbf{endif}$
    $\textbf{endfor}$
$\textbf{endfor}$
$\textbf{return}\ r_{LRF}$
-----------------------------------------------------------------------------------
```

Table 2.3. Ray Casting algorithm using grid map

2.4 Sensor Fusion

The methods that solve robot localization problem combine initial information, absolute and relative measurements to form estimates of the location of the robot at a certain time. If the measurements are considered to be readings from different sensors, the process used to combine the readings from different sensors to forma a combined representation of the environment. The term used in robotics for this process is called as sensor fusion or multi-sensor fusion.

Fusion of information from multiple sensors is important, since combined representation information from multiple sensors can be more accurate. In particular, when all sensors are not able to sense the same. Some features may occlude for some sensors, while visible to others. Together the sensor can provide a more complete picture of scene at a certain time. Multi-sensor fusion is also important since it can reduce the effect of errors in measurements [34].

Multi-sensor fusion can rely on a probabilistic approach, where notion of uncertainty and belief are common terminology. Kalman filter (KF) approach is the most popular tools for sensor fusion in mobile robot navigation. Sensor data fusion through Kalman filter significantly improves on-line estimation reducing the effect of sensor noise and bias. Figure 2.10 shows the basic sensor fusion scheme using Kalman filter.

In Kalman filter formulation, the observation $z(k) \in \mathcal{R}^n$ is approximated by the linear model

$$z(t+1) = H(t+1)x(t+1) + Q(t+1 \tag{2.9}$$

Where $x \in \mathcal{R}^m$ is a state vector, $H \in \mathcal{R}^{n \times m}$ is an observation model, and $\upsilon \in \mathcal{R}^n$ is the observation noise.

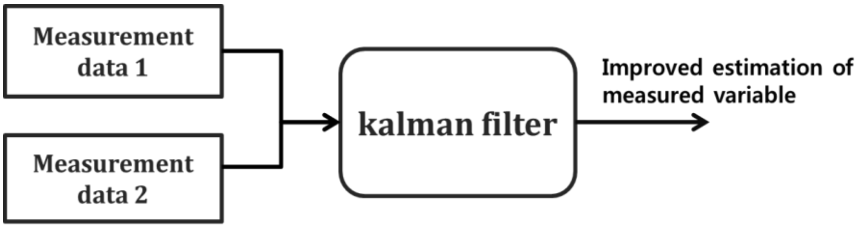


Figure 2.10. Sensor fusion scheme using Kalman filter approach

The most common use of Kalman filter in sensor fusion for navigation is to construct and maintain the a model of the mobile robot environment, and to monitor the position of the moving robot in that environment, the filter estimates simultaneously the parameters of the features or landmarks needed in the environment model and the state of the moving robot.

As stated in the previous chapter, in our proposed approach we use the fusion of ultrasonic sensor and laser range finder measurements for mobile robot localization. The next chapters will explain in detail how we fuse the measurements from both sensors in UKF localization algorithm.

3. Unscented Kalman Filter for Mobile Robot Localization

Filtering methods play vital role in mobile for estimating the states of robotic system. A good filtering algorithm can reduce the noise from the control and measurement signals while retaining the useful information. During the last few decades Kalman filter (KF) and extended Kalman filter (EKF) have been widely used for state estimation. The Unscented Kalman filter (UKF) is one of the recent variations of Kalman filter. UKF have been developed to overcome the inherent flaws of Extended Kalman filter (EKF) using statistical approach for calculating the mean and covariance which undergoes a non-linear transform upto higher order.

3.1 Preliminary Knowledge of UKF parameters

The UKF is a versatile engineering tool that provides good non-linear estimation results for many practical problems. UKF performs the stochastic linearization through the use of weighted linear regression process called as unscented transform. Unscented transform is the central technique of UKF which is used to handle the non-linearity in a non-linear transform by deterministiacally selects the set of samples called as sigma points [35]. Figure 3.1 shows the basic figure of unscented transformation.

UKF compute the transformation which approximates the mean and covariance of a random vector of length L when it is transformed by the linear map. This is done by computing a set of $2L+1$ sigma points, provides the basis of mean μ and covariance Σ of the original vector, transforming these points by the non-linear mapping and then approximate the mean μ and covariance Σ of transformed vector from the transformed sigma points.

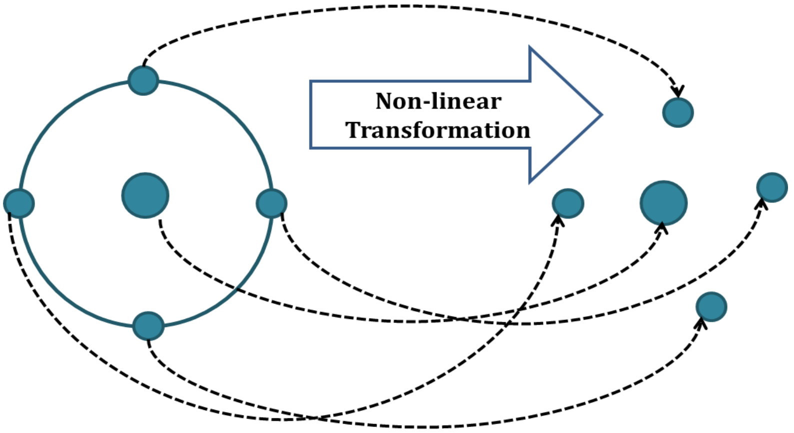


Figure 3.1. Unscented transformation

For L dimensional Gaussian with mean μ and covariance Σ the resulting sigma points are chosen according to the following rule:

$$\begin{aligned}
&\chi^{[0]} = \mu \\
&\chi^{[i]} = \mu + (\sqrt{(L+\lambda)\Sigma})_i \quad for \;\; i = 1,\ldots,L \\
&\chi^{[i]} = \mu + (\sqrt{(L+\lambda)\Sigma})_{i-L} \quad for \;\; i = L+1,\ldots,2L
\end{aligned} \tag{3.1}$$

Here $\lambda = \alpha^2(L+\kappa) - L$ with α and κ being scaling parameters that determines how far the sigma points are spread from the mean. Each sigma point $\chi^{[i]}$ has two weights associated with it. One weight, $w_m^{[i]}$ is used when computing the mean, the other weight, $w_c^{[i]}$ is used when recovering the covariance of the Gaussian.

$$\begin{aligned}
&w_m^{[0]} = \frac{\lambda}{L+\lambda} \\
&w_c^{[0]} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\
&w_m^{[i]} = w_c^{[i]} = \frac{1}{2(L+\lambda)} \quad for \; i = 1,\ldots,2L
\end{aligned} \tag{3.2}$$

The parameter β can be chosen to encode additional (higher order) knowledge about the distribution underlying the Gaussian representation. If the distribution is exact Gaussian then β =2 is the optimal choice.

3.1.1 Augmented Unscented Kalman Filter

In this thesis we implement the augmented unscented Kalman filter (UKF) strategy in robot localization. The key trick is to augment the state with the additional components representing the control and the measurement noise. The dimensionality of the augmented state is given by the sum of state, control and measurement dimensions, which 3+2+N (where N is the number of measurements). We assume zero-mean Gaussian noise, throughout in this algorithm. The mean μ_{t-1}^{a} of the augmented state estimate is given by the mean of location estimate $\mu_{X,t-1} = (\mu_{x,t-1}, \mu_{y,t-1}, \mu_{\theta,t-1})$, and zero mean values for the control z_t and measurement μ_{z_t} noise.

$$\mu_{t-1}^{a} = \begin{bmatrix} \mu_{X,t-1} \\ \mu_{u_t} \\ \mu_{z_t} \end{bmatrix} \quad (\because \mu_{u_t} = \mu_{z_t} = 0) \tag{3.3}$$

The covariance Σ_{t-1}^{a} of the augmented state estimate is given by combining the location position covariance Σ_{t-1}, the control noise M_t and measurement noise covariance Q_t

$$\Sigma_{t-1}^{a} = \begin{bmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{bmatrix} \tag{3.4}$$

The Equation 3.3 and 3.4 are used to generate the augmented sigma points using unscented transform (Equation 3.1).

$$\chi_{t-1}^{a} = \begin{pmatrix} \chi_{t-1}^{x\ T} \\ \chi_{t}^{uT} \\ \chi_{t}^{zT} \end{pmatrix} = [\mu_{t-1}^{a} \quad \mu_{t-1}^{a} + \gamma\sqrt{\Sigma_{t-1}^{a}} \quad \mu_{t-1}^{a} - \gamma\sqrt{\Sigma_{t-1}^{a}}] \tag{3.5}$$

Where, $\gamma = \sqrt{L+\lambda}$ and the matrix of augmented covariance matrix Σ_{t-1}^{a} can be done in block as, in

$$\sqrt{\Sigma_{t-1}^{a}} = \begin{bmatrix} \sqrt{\Sigma_{t-1}} & 0 & 0 \\ 0 & \sqrt{M_t} & 0 \\ 0 & 0 & \sqrt{Q_t} \end{bmatrix} \tag{3.6}$$

The augmented covariance calculation is done prior to the prediction step. In prediction the state and process noise sigma points passes through the f motion model function.

$$\chi_{t|t-1}^{x(i)} = f(\chi_{t-1}^{x(i)}, u_{t-1}, \chi_{t}^{u(i)}) \quad for\ i = 1, \ldots \tag{3.7}$$

The mean $\mu_{X,t|t-1}$ and covariance $\Sigma_{t|t-1}$ of the state calculated as:

$$\begin{aligned} \mu_{X,t|t-1} &= \sum_{i=0}^{2L} w_m^{[i]} \chi_{t|t-1}^{x(i)} \\ \Sigma_{t|t-1} &= \sum_{i=0}^{2L} w_c^{[i]} (\chi_{t|t-1}^{x(i)} - \mu_{X,t|t-1})(\chi_{t|t-1}^{x(i)} - \mu_{X,t|t-1})^T \end{aligned} \tag{3.8}$$

In measurement prediction the predicted state sigma points $\chi_{t|t-1}^{x(i)}$ and the measurement noise sigma point passes through the h measurement model function.

$$z^{(i)}_{t|t-1} = h(\chi^{x(i)}_{t|t-1}, \chi^{z(i)}_{t}) \quad for\ i = 1,.......2L \tag{3.9}$$

Then estimated measurement $\hat{z}_{t|t-1}$, measurement covariance S_t and the cross covariance $\Sigma^{X,z}_t$ between state and measurement are calculated

$$\begin{aligned}
\hat{z}_{t|t-1} &= \sum_{i=0}^{2L} w^{[i]}_m \ z^{(i)}_{t|t-1} \\
S_t &= \sum_{i=0}^{2L} w^{[i]}_c (z^{(i)}_{t|t-1} - \hat{z}_{t|t-1})(z^{(i)}_{t|t-1} - \hat{z}_{t|t-1})^T \\
\Sigma^{X,z}_t &= \sum_{i=0}^{2L} w^{[i]}_c (\chi^{x(i)}_{t|t-1} - \mu_{X,t|t-1})(z^{(i)}_{t|t-1} - \hat{z}_{t|t-1})^T
\end{aligned} \tag{3.10}$$

The correction step of augmented UKF is virtually identical to EKF correction step. All the values computed form Equations 3.3 to 3.10 are propagated to the next time step where they need to re-augment with the noise states (all zeros) and corresponding covariance matrices. Repeat the computation of each equation at each time step of UKF for the desired number of time steps. This section provided the theoretical explanation of UKF. In the later section 3.3 of this chapter will covers the details of each equation with respect to proposed approach used to process UKF algorithm.

3.2 Maximum Likelihood Estimation

The principle of maximum likelihood estimation (MLE) originally developed by R.A fisher in 1920s, states that the desired probability distribution is the one that makes the observed data "most likely" which means that one must seek the values of the parameter vector that maximizes the likelihood function [35]. In this section we discuss the maximum likelihood estimation (MLE) method used to cope with the data association problem to calculate the correspondence in a partially unknown evnironment. The maximum likelihood estimator determines the correspondence that maximizes the data likelihood.

$$\hat{c}_t = \underset{c_t}{argmax}\, p(z_t \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) \tag{3.11}$$

Here c_t is the correspondence vector at time t. As before, the vector $z_t = \{z_t^1, z_t^1, ...\}$ is the measurement vector that contains the list of features, or landmarks, z_t^k observed at time t [21].

The *argmax* operator in Equation 3.11 selects the correspondence vector $\hat{c}_t$ that maximizes the likelihood of the measurement. Note that this expression is conditioned on the prior correspondences $c_{1:t-1}$. While those have been estimated in previous update steps, the maximum likelihood approach treats them as if they are always correct. This has two important ramifications:

1. It makes it possible to update the filter incrementally

2. It also introduces brittleness in the filter, which tends to diverge when correspondence estimates are erroneous.

Even under the assumptions of known prior correspondences, there are exponentially many terms of maximization in Equation 3.13. When the number of detected landmarks per measurement is large, the number of possible correspondences may grow too large for practical implementations. The most common technique is to avoid such an exponential complexity is to performs the maximization separately for each individual feature, z_t^k in the measurement vector z_t. Equation 3.14 shows the correspondence of each feature.

$$\begin{aligned} \hat{c}_t^k &= \underset{c_t^i}{argmax}\, p(z_t^k \mid c_{1:t}, m, z_{1:t-1}, u_{1:t}) \\ &\approx \underset{c_t^i}{argmax}\, p(z_t^k \mid h\,(\mu_t, c_t^k, m), z_{1:t-1}, u_{1:t}) \end{aligned} \tag{3.12}$$

We have implemented Equation 3.12 in the later section of this chapter for calculating the correspondence associated with ultrasonic beacons. This component wise optimization is justified only when we know that individual feature vectors are conditionally independent (an assumption that is usually adopted for convenience)

3.3 UKF Localization Algorithm

In this section we discuss the augmented UKF localization algorithm for the case of unknown correspondence. This section describes the detail of this algorithm based on the robot navigation experiment which is performed for this thesis. Basically the UKF algorithm undergoes the two main steps shown in Table 3.1.

```
$\textbf{\textit{UKF Localization}}\ (\mu_{X,t-1}, \Sigma_{t-1}, u_t, z_t, m)$
1. $(\bar{\mu}_{X,t}, \bar{\Sigma}_t) = \textbf{\textit{Prediction step}}\,(\mu_{X,t-1}, \Sigma_{t-1}, u_t)$
2. $(\mu_{X,t}, \Sigma_t) = \textbf{\textit{Correction step}}(\bar{\mu}_{X,t}, \bar{\Sigma}_t, z_t, m)$
3. $\textbf{\textit{return}}(\mu_{X,t}, \Sigma_t)$
```

Table 3.1. Pseudo code of UKF algorithm

Line1 shows the Prediction step in which state is predicted at time t using the previous state information $t-1$ and control u_t as input. Then mean $\bar{\mu}_{X,t}$ and covariance $\bar{\Sigma}_t$ of predicted state is computed. The Correction step corrects the predicted mean and covariance using the measurement z_t , the map m information and generates the mean $\mu_{X,t}$ and covariance Σ_t of the updated state. On the basis of experiment we illustrate the prediction step and correction step of augmented UKF algorithm in detail below.

3.3.1 Illustration of Prediction Step

In prediction step UKF localization algorithm considers control input u_t as translational velocity v_t and rotational velocity ω_t from the wheel encoders, then determined the motion uncertainty covariance matrix M_t in control space. The covariance matrix in Equation 3.13 follows directly from the motion model in Table 2.1.

$$M_t = \begin{bmatrix} \alpha_{vv} v_t^2 + \alpha_{v\omega}\omega_t^2 & 0 \\ 0 & \alpha_{\omega v} v_t^2 + \alpha_{\omega\omega}\omega_t^2 \end{bmatrix}_{(2\times 2)} \tag{3.13}$$

Where $\alpha_{vv} \sim \alpha_{\omega\omega}$ in Equation 3.13 show the motion uncertainty parameters. Equation 3.14 shows the measurement noise covariance matrix Q_t with respect to range data.

$$Q_t = \begin{bmatrix}
\sigma^2_{r_{u,1}} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma^2_{r_{u,2}} & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \sigma^2_{r_{u,3}} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma^2_{r_{u,4}} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma^2_{r_{LRF,1}} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{LRF,2}} & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{LRF,3}} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \sigma^2_{r_{LRF,14}} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \sigma^2_{r_{LRF,15}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \sigma^2_{r_{LRF,16}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \sigma^2_{r_{LRF,17}} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \sigma^2_{r_{LRF,18}} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \sigma^2_{r_{LRF,19}}
\end{bmatrix}_{(23\times 23)} \tag{3.14}$$

The variable σ_r^{usat} denotes the measurement noise for ultrasonic sensor, and σ_r^{LRF} denotes the measurement noise for laser range finder. For sensor measurements we only consider sensor noise with respect to range. Because we have use the fusion of ultrasonic sensor and laser range finder measurements. Altogether we have $N = N_{usat} + N_{LRF} = 23$ range data. Where N_{usat} = 4 denotes the number of ultrasonic sensor data, and N_{LRF} =19 denotes the number of laser range finder data.

Then Equation 3.15 the augmented means state vector μ_{t-1}^a consists of mean of robot position $\mu_{X,t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})$, and zero mean vectors for the control uncertainty and measurement values (as discussed in earlier we assume zero mean noise) at time $t-1$. The dimensionality of this vector should be 3+2+23.

$$\mu_{t-1}^{a} = \begin{bmatrix} \mu_{x,t-1} \\ \mu_{y,t-1} \\ \mu_{\theta,t-1} \\ \mu_{v,t-1} \\ \mu_{\omega,t-1} \\ \mu_{r_{usat,1},t-1} \\ \mu_{r_{usat,2},t-1} \\ \vdots \\ \mu_{r_{usat,4},t-1} \\ \mu_{r_{LRF,1},t-1} \\ \mu_{r_{LRF,2},t-1} \\ \vdots \\ \mu_{r_{LRF,19},t-1} \end{bmatrix}_{28\times 1} \quad \because \left(\mu_{v,t-1}, \mu_{\omega,t-1}, \mu_{r_{usat,1},t-1}, \ldots, \mu_{r_{usat,4},t-1}, \mu_{r_{LRF,1},t-1}, \ldots, \mu^{a}_{r_{LRF,19},t-1} = 0 \right) \tag{3.15}$$

Equation 3.16 shows the augmented covariance matrix Σ_{t-1}^{a} which consists of robot position covariance Σ_{t-1} motion uncertainty covariance M_t , and measurement noise covariance Q_t at time $t-1$.

$$\Sigma_{t-1}^{a} = \begin{bmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{bmatrix}$$

$$\Sigma_{t-1}^{a}=\begin{bmatrix} \Sigma_{xx,t-1} & \Sigma_{xy,t-1} & \Sigma_{x\theta,t-1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \Sigma_{yx,t-1} & \Sigma_{yy,t-1} & \Sigma_{y\theta,t-1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \Sigma_{\theta x,t-1} & \Sigma_{\theta y,t-1} & \Sigma_{\theta\theta,t-1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{t,(1,1)} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{t,(2,2)} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{r_1^{usat}}^2 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{r_2^{usat}}^2 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \sigma_{r_4^{usat}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \sigma_{r_1^{LRF}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{r_2^{LRF}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \sigma_{r_{18}^{LRF}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \sigma_{r_{19}^{LRF}}^2 \end{bmatrix}_{(28\times 28)} \tag{3.16}$$

Now the augmented mean state vector and the augmented state covariance are used to generate the sigma points. The Equation 3.17 illustrates the sigma point matrix χ_{t-1}^{a}. The dimension of sigma point matrix is $L\times(2L+1)$ and each column of this matrix represents a sigma point. Where L denotes the number of states in augmented state vector. In this case L =28, so we have 28×57 dimension of sigma point matrix and 57 sigma points.

$$\begin{aligned} \chi_{t-1}^{a} &= \left[\mu_{t-1}^{a} \quad \mu_{t-1}^{a}+\gamma\sqrt{\Sigma_{t-1}^{a}} \quad \mu_{t-1}^{a}-\gamma\sqrt{\Sigma_{t-1}^{a}}\right] \\ &= \begin{bmatrix} \mu_{x,t-1} & \mu_{x,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(1,1)} & \cdots & \mu_{x,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(1,28)} & \mu_{x,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(1,1)} & \cdots & \mu_{x,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(1,28)} \\ \mu_{y,t-1} & \mu_{y,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(2,1)} & \cdots & \mu_{y,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(2,28)} & \mu_{y,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(2,1)} & \cdots & \mu_{y,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(2,28)} \\ \mu_{\theta,t-1} & \mu_{\theta,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(3,1)} & \cdots & \mu_{\theta,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(3,28)} & \mu_{\theta,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(3,1)} & \cdots & \mu_{\theta,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(3,28)} \\ \mu_{v,t-1} & \mu_{v,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(4,1)} & \cdots & \mu_{v,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(4,28)} & \mu_{v,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(4,1)} & \cdots & \mu_{v,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(4,28)} \\ \mu_{\omega,t-1} & \mu_{\omega,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(5,1)} & \cdots & \mu_{\omega,t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(5,28)} & \mu_{\omega,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(5,1)} & \cdots & \mu_{\omega,t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(5,28)} \\ \mu_{r_1^{usat},t-1} & \mu_{r_1^{usat},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(6,1)} & \cdots & \mu_{r_1^{usat},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(6,28)} & \mu_{r_1^{usat},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(6,1)} & \cdots & \mu_{r_1^{usat},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(6,28)} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \mu_{r_4^{usat},t-1} & \mu_{r_4^{usat},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(9,1)} & \cdots & \mu_{r_4^{usat},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(9,28)} & \mu_{r_4^{usat},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(9,1)} & \cdots & \mu_{r_4^{usat},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(9,28)} \\ \mu_{r_1^{LRF},t-1} & \mu_{r_1^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(10,1)} & \cdots & \mu_{r_1^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(10,28)} & \mu_{r_1^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(10,1)} & \cdots & \mu_{r_1^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(10,28)} \\ \mu_{r_2^{LRF},t-1} & \mu_{r_2^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(11,1)} & \cdots & \mu_{r_2^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(11,28)} & \mu_{r_2^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(11,1)} & \cdots & \mu_{r_2^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(11,28)} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \mu_{r_{18}^{LRF},t-1} & \mu_{r_{18}^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(27,1)} & \cdots & \mu_{r_{18}^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(27,28)} & \mu_{r_{18}^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(27,1)} & \cdots & \mu_{r_{18}^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(27,23)} \\ \mu_{r_{19}^{LRF},t-1} & \mu_{r_{19}^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(28,1)} & \cdots & \mu_{r_{19}^{LRF},t-1}+\gamma\sqrt{\Sigma_{t-1}^{a}}_{(28,28)} & \mu_{r_{19}^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(28,1)} & \cdots & \mu_{r_{19}^{LRF},t-1}-\gamma\sqrt{\Sigma_{t-1}^{a}}_{(28,28)} \end{bmatrix}_{(28\times 57)} \end{aligned} \tag{3.17}$$

The γ, λ parameters are already defined in previous section 3.1. Equation 3.18 shows velocity motion model g. The g requires the location components χ_{t-1}^{x} of the sigma points, control input u_t with added control noise component $\chi_{i,t-1}^{u}$ of each sigma point at time $t-1$.

$$\bar{\chi}_t^x = g(u_t + \chi_t^u, \chi_{t-1}^x) \tag{3.18}$$

$$= \begin{bmatrix} g(u_t + \chi_{t-1,(:,1)}^u, \chi_{t-1(1,1)}^x) & g(u_t + \chi_{t-1,(:,2)}^u, \chi_{t-1(1,2)}^x) & \cdots & g(u_t + \chi_{t-1,(:,57)}^u, \chi_{t-1(1,57)}^x) \\ g(u_t + \chi_{t-1,(:,1)}^u, \chi_{t-1(2,1)}^x) & g(u_t + \chi_{t-1,(:,2)}^u, \chi_{t-1(2,2)}^x) & \cdots & g(u_t + \chi_{t-1,(:,57)}^u, \chi_{t-1(2,57)}^x) \\ g(u_t + \chi_{t-1,(:,1)}^u, \chi_{t-1(2,1)}^x) & g(u_t + \chi_{t-1,(:,2)}^u, \chi_{t-1(3,2)}^x) & \cdots & g(u_t + \chi_{t-1,(:,57)}^u, \chi_{t-1(3,57)}^x) \end{bmatrix}_{(3\times57)}$$

The Equation 3.18 actually performs the prediction step. In this case the prediction position components of sigma points χ_{t-1}^{x} are predicted at time t. Equation 3.19 computes the predicted state mean $\bar{\mu}_t$ by adding up the location component of sigma point χ_{t-1}^{x} with the mean weight $w_i^{(m)}$ value. The $\bar{\mu}_t$ vector is the predicted robot position $X_t = (x_t, y_t, \theta_t)$ at time t.

$$\bar{\mu}_t = \sum_{i=0}^{2L} w_i^{(m)} + \bar{\chi}_{i,t}^x \tag{3.19}$$

$$= w_0^{(m)} \begin{bmatrix} \bar{\chi}_{t,(1,1)}^x \\ \bar{\chi}_{t,(2,1)}^x \\ \bar{\chi}_{t,(3,1)}^x \end{bmatrix} + w_1^{(m)} \begin{bmatrix} \bar{\chi}_{t,(1,2)}^x \\ \bar{\chi}_{t,(2,2)}^x \\ \bar{\chi}_{t,(3,2)}^x \end{bmatrix} + \cdots + w_{54}^{(m)} \begin{bmatrix} \bar{\chi}_{t,(1,55)}^x \\ \bar{\chi}_{t,(2,55)}^x \\ \bar{\chi}_{t,(3,55)}^x \end{bmatrix} + w_{55}^{(m)} \begin{bmatrix} \bar{\chi}_{t,(1,56)}^x \\ \bar{\chi}_{t,(2,56)}^x \\ \bar{\chi}_{t,(3,56)}^x \end{bmatrix} + w_{56}^{(m)} \begin{bmatrix} \bar{\chi}_{t,(1,57)}^x \\ \bar{\chi}_{t,(2,57)}^x \\ \bar{\chi}_{,t(3,57)}^x \end{bmatrix} = \begin{bmatrix} \bar{\mu}_{x,t} \\ \bar{\mu}_{y,t} \\ \bar{\mu}_{\theta,t} \end{bmatrix}_{(3\times1)}$$

Equation 3.20 computes the predicted state covariance $\bar{\Sigma}_t$ at time t .

$$
\begin{aligned}
\bar{\Sigma}_t &= \sum\nolimits_{i=0}^{2L} w_i^{(\mathrm{c})} \left(\bar{\chi}_{i,t}^{x} - \bar{\mu}_t\right)\left(\bar{\chi}_{i,t}^{x} - \bar{\mu}_t\right)^T \\
&= w_0^{(c)} \left(\begin{bmatrix} \bar{\chi}_{t,(1,1)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,1)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,1)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right) \left(\begin{bmatrix} \bar{\chi}_{t,(1,1)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,1)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,1)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right)^T + w_1^{(c)} \left(\begin{bmatrix} \bar{\chi}_{t,(1,2)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,2)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,2)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right) \left(\begin{bmatrix} \bar{\chi}_{t,(1,2)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,2)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,2)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right)^T + \cdots \\
&+ w_{55}^{(c)} \left(\begin{bmatrix} \bar{\chi}_{t,(1,56)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,56)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,56)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right) \left(\begin{bmatrix} \bar{\chi}_{t,(1,56)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,56)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,56)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right)^T + w_{56}^{(c)} \left(\begin{bmatrix} \bar{\chi}_{t,(1,57)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,57)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,57)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right) \left(\begin{bmatrix} \bar{\chi}_{t,(1,57)}^{x} - \bar{\mu}_{x,t} \\ \bar{\chi}_{t,(2,57)}^{x} - \bar{\mu}_{y,t} \\ \bar{\chi}_{t,(3,57)}^{x} - \bar{\mu}_{\theta,t} \end{bmatrix} \right)^T \\
&= \begin{bmatrix} \bar{\Sigma}_{xx,t} & \bar{\Sigma}_{xy,t} & \bar{\Sigma}_{x\theta,t} \\ \bar{\Sigma}_{yx,t} & \bar{\Sigma}_{yy,t} & \bar{\Sigma}_{y\theta,t} \\ \bar{\Sigma}_{\theta x,t} & \bar{\Sigma}_{\theta y,t} & \bar{\Sigma}_{\theta\theta t} \end{bmatrix}_{(3\times 3)}
\end{aligned}
\tag{3.20}
$$

The complete pseudo code of UKF prediction step is shown in Table 3.2.

```
Prediction step $(\mu_{t-1}, \Sigma_{t-1}, u_t)$
1. $M_t = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{bmatrix}$
2. $Q_t = \begin{bmatrix} \sigma_{r_1^{usat}}^2 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & \sigma_{r_4^{usat}}^2 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \sigma_{r_1^{LRF}}^2 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \sigma_{r_{18}^{LRF}}^2 & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & \sigma_{r_{19}^{LRF}}^2 \end{bmatrix}$
3. $\mu_{t-1}^a = \begin{bmatrix} \mu_{x,t-1} & \mu_{y,t-1} & \mu_{\theta,t-1} & \mu_{v,t-1} & \mu_{\omega,t-1} & \mu_{r_1,t-1} & \mu_{r_1,t-1} & \cdots & \mu_{r_{22},t-1} & \mu_{r_{23},t-1} \end{bmatrix}^T$
4. $\Sigma_{t-1}^a = \begin{bmatrix} \Sigma_{t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{bmatrix}$
5. $\chi_{t-1}^a = \begin{bmatrix} \mu_{t-1}^a & \mu_{t-1}^a + \gamma\sqrt{\Sigma_{t-1}^a} & \mu_{t-1}^a - \gamma\sqrt{\Sigma_{t-1}^a} \end{bmatrix}$
6. $\bar{\chi}_t^x = g(u_t + \chi_t^u, \chi_{t-1}^x)$
7. $\bar{\mu}_t = \sum_{i=0}^{2L} w_i^{(m)} + \bar{\chi}_{i,t}^x$
8. $\bar{\Sigma}_t = \sum_{i=0}^{2L} w_i^{(\mathrm{c})} \left(\bar{\chi}_{i,t}^x - \bar{\mu}_t\right)\left(\bar{\chi}_{i,t}^x - \bar{\mu}_t\right)$
```

Table 3.2. Pseudo code of UKF prediction step

3.3.2 Illustration of Correction Step

Correction step of UKF algorithm undergoes the measurement prediction step and the filter update step. The measurement prediction step requires measurement model with additive measurement noise sigma points. As we discussed earlier in this section this algorithm works for unknown correspondence. Therefore, the additional block for calculation of correspondence is accumulated in the UKF correction step. For the calculation of correspondence we only consider the ultrasonic sensor measurement data and the position of the ultrasonic beacons in the experiment work space.

3.3.2.1 Calculation of Correspondence

To determine the correspondence, we have $N_{usat} = 4$ sensor data z_t^{usat} from the ultrasonic sensor, the variable k denotes the number of actual ultrasonic sensor data. The c_t^k is correspondence variable consisting of hypothesis which associates range data to beacon by selecting the maximum likelihood of the measurement $z_{t,k}^{usat}$ of any possible ultrasonic beacon in the map m^{usat} . The variable m^{usat} denotes the map which specifies every j^{th} ultrasonic beacon's position in work space. In this case the correspondence variable c_t^k contains 4 different most likely correspondence values correspond to each ultrasonic beacon position in the experiment work space.

Equation 3.21 shows that the predicted robot position components $\bar{\chi}_t^x$ passes through the ultrasonic measurement model h_{usat} and added up with measurement noise components $\bar{\chi}_t^{z_{usat}}$; generates the measurement sigma points $\bar{Z}_t^{usat}$ for ultrasonic sensor measurement the superscript $usat$ denotes ultrasonic sensor measurement.

$$\bar{Z}^{usat}_{t,j} = (h_{usat}(\bar{\chi}^{x}_{t}) + \bar{\chi}^{z_{usat}}_{t})$$

$$\begin{aligned}
\bar{Z}^{usat}_{t,1} &= \left[h_{usat,1}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,1)} \quad h_{usat,1}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,2)} \quad \cdots \quad h_{usat,1}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,56)} \quad h_{usat,1}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,57)} \right] \\
\bar{Z}^{usat}_{t,2} &= \left[h_{usat,2}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,1)} \quad h_{usat,2}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,2)} \quad \cdots \quad h_{usat,2}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,56)} \quad h_{usat,2}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,57)} \right] \\
\bar{Z}^{usat}_{t,3} &= \left[h_{usat,3}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,1)} \quad h_{usat,3}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,2)} \quad \cdots \quad h_{usat,3}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,56)} \quad h_{usat,3}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,57)} \right] \\
\bar{Z}^{usat}_{t,4} &= \left[h_{usat,4}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,1)} \quad h_{usat,4}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,2)} \quad \cdots \quad h_{usat,4}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,56)} \quad h_{usat,4}\begin{pmatrix} \bar{\chi}^{x}_{t,(1,1)} \\ \bar{\chi}^{x}_{t,(2,1)} \\ \bar{\chi}^{x}_{t,(3,1)} \end{pmatrix} + \bar{\chi}^{z_{usat}}_{t\,(1,57)} \right]
\end{aligned} \qquad (3.21)$$

Equation 3.22 shows the predicted measurement $\hat{z}^{usat}_{t}$ by adding up the measurement sigma points $\bar{Z}^{usat}_{t}$ with the weighted mean $w^{(m)}_{i}$.

$$\hat{z}^{usat}_{t,j} = (\sum_{i=0}^{2L} w^{(m)}_{i} \bar{Z}^{usat,i}_{t,j})$$

$$\begin{aligned}
\hat{z}^{usat}_{t,1} &= w^{(m)}_{0}(\bar{Z}^{usat}_{t,(1,1)}) + w^{(m)}_{1}(\bar{Z}^{usat}_{t,(1,2)}) + \cdots + w^{(m)}_{55}(\bar{Z}^{usat}_{t,(1,17)}) + w^{(m)}_{56}(\bar{Z}^{usat}_{t,(1,18)}) \\
\hat{z}^{usat}_{t,2} &= w^{(m)}_{0}(\bar{Z}^{usat}_{t,(1,1)}) + w^{(m)}_{1}(\bar{Z}^{usat}_{t,(1,2)}) + \cdots + w^{(m)}_{55}(\bar{Z}^{usat}_{t,(1,17)}) + w^{(m)}_{56}(\bar{Z}^{usat}_{t,(1,18)}) \\
\hat{z}^{usat}_{t,3} &= w^{(m)}_{0}(\bar{Z}^{usat}_{t,(1,1)}) + w^{(m)}_{1}(\bar{Z}^{usat}_{t,(1,2)}) + \cdots + w^{(m)}_{55}(\bar{Z}^{usat}_{t,(1,17)}) + w^{(m)}_{56}(\bar{Z}^{usat}_{t,(1,18)}) \\
\hat{z}^{usat}_{t,4} &= w^{(m)}_{0}(\bar{Z}^{usat}_{t,(1,1)}) + w^{(m)}_{1}(\bar{Z}^{usat}_{t,(1,2)}) + \cdots + w^{(m)}_{55}(\bar{Z}^{usat}_{t,(1,17)}) + w^{(m)}_{56}(\bar{Z}^{usat}_{t,(1,18)})
\end{aligned} \qquad (3.22)$$

Equation 3.23 illustrates the measurement covariance S^{usat}_{t} for each j^{th} ultrasonic beacon.

$$S_{t,j}^{usat} = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{t,j}^{usat} - \hat{z}_{t,j}^{usat})(\bar{Z}_{t,j}^{usat} - \hat{z}_{t,j}^{usat})^T)$$

$$\begin{aligned}
S_{t,1}^{usat} = {} & w_0^{(c)} \left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)^T + w_1^{(c)} \left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)^T + \cdots \\
& w_{55}^{(c)} \left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)^T + w_{56}^{(c)} \left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)^T
\end{aligned}$$

$$\begin{aligned}
S_{t,2}^{usat} = {} & w_0^{(c)} \left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)\left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)^T + w_1^{(c)} \left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)\left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)^T + \cdots \\
& w_{55}^{(c)} \left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)\left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)^T + w_{56}^{(c)} \left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)\left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(2)}^{usat}\right)^T
\end{aligned}$$

$$\begin{aligned}
S_{t,3}^{usat} = {} & w_0^{(c)} \left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)\left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)^T + w_1^{(c)} \left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)\left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)^T + \cdots \\
& w_{55}^{(c)} \left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)\left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)^T + w_{56}^{(c)} \left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(3)}^{usat}\right)^T
\end{aligned}$$

$$\begin{aligned}
S_{t,4}^{usat} = {} & w_0^{(c)} \left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)\left(\bar{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)^T + w_1^{(c)} \left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)\left(\bar{Z}_{t,(1,2)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)^T + \cdots \\
& w_{55}^{(c)} \left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)\left(\bar{Z}_{t,(1,55)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)^T + w_{56}^{(c)} \left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(1)}^{usat}\right)\left(\bar{Z}_{t,(1,56)}^{usat} - \hat{z}_{t,(4)}^{usat}\right)^T
\end{aligned} \tag{3.23}$$

Equation 3.24 illustrates the maximum likelihood this calculation requires predicted measurements $\hat{z}_t^{usat}$, measurement covariance S_t^{usat} , and the actual ultrasonic sensor data z_t^{usat} . The variable ML in this equation shows the maximum likelihood measurement of every k^{th} actual ultrasonic sensor data.

$$\begin{aligned}
ML_j^k &= \det(2\pi S_t^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})^T \left[S_t^{usat}\right]^{-1} (z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})\} \\
ML_1^k &= \det(2\pi S_{t,1}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat})^T \left[S_{t,1}^{usat}\right]^{-1} (z_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat})\} \\
ML_2^k &= \det(2\pi S_{t,2}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat})^T \left[S_{t,2}^{usat}\right]^{-1} (z_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat})\} \\
ML_3^k &= \det(2\pi S_{t,3}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,(3,1)}^{usat} - \hat{z}_{t,(3,1)}^{usat})^T \left[S_{t,3}^{usat}\right]^{-1} (z_{t,(3,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat})\} \\
ML_4^k &= \det(2\pi S_{t,4}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat})^T \left[S_{t,4}^{usat}\right]^{-1} (z_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat})\}
\end{aligned} \tag{3.24}$$

Equation 3.25 determines the correspondence. The *argmax* operator selects the correspondence vector $\hat{c}_t$ that maximizes the likelihood of every k^{th} ultrasonic sensor measurement z_t^{usat} .

$$\begin{aligned} c_t^k &= \underset{k}{argmax}\ (ML) \\ c_t^k &= \underset{k}{argmax}\ (ML_{(1)}, ML_{(2)}, ML_{(3)}, ML_{(4)}) \end{aligned} \tag{3.25}$$

Table 3.3 shows the complete pseudo code for the calculation of correspondence of ultrasonic beacons.

For the case of unknown correspondence

```
for all actual sensor data $z_{t,k}^{usat} = \left( r_{t,k}^{usat} \right)_{(4\times1)}$
        for all USAT beacons $j$ in the map $'m'$
            $\bar{Z}_{t,j}^{usat} = (h_{usat}(\bar{\chi}_t^x) + \bar{\chi}_t^{z_{usat}})_{(1\times57)}$
            $\hat{z}_{t,j}^{usat} = (\sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{t,j}^{usat})_{(1\times1)}$
            $S_{t,j}^{usat} = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{t,j}^{usat} - \hat{z}_{t,j}^{usat})(\bar{Z}_{t,j}^{usat} - \hat{z}_{t,j}^{usat})^T)_{(1\times1)}$
            $ML_j^k = (\det(2\pi S_{t,j}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})^T \left[ S_{t,j}^{usat} \right]^{-1} (z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})\})_{1\times1}$
        end
    $c_t^k = \underset{k}{argmax}\ (ML)$
end
```

Table 3.3. Pseudo code of calculation of correspondence

3.3.2.2 Measurement Prediction and filter update

In measurement prediction steps the robot position components $\bar{\chi}_t^x$ passes through the measurement model h shown in Equation 3.26 then added with measurement noise sigma points $\bar{\chi}_t^z$.

$\bar{Z}_t = h\ (\bar{\chi}_t^x)\ +\ \bar{\chi}_t^z$

$$
= \begin{bmatrix}
h_{usat,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(1,1)} & h_{usat,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(1,2)} & \cdots & h_{usat,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(1,56)} & h_{usat,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(1,57)} \\
h_{usat,2}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(2,1)} & h_{usat,2}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(2,2)} & \cdots & h_{usat,2}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(2,56)} & h_{usat,2}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(2,57)} \\
h_{usat,3}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(3,1)} & h_{usat,3}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(3,2)} & \cdots & h_{usat,3}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(3,56)} & h_{usat,3}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(3,57)} \\
h_{usat,4}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(4,1)} & h_{usat,4}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t,(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(4,2)} & \cdots & h_{usat,4}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t,(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(4,56)} & h_{usat,4}\begin{pmatrix}\bar{\chi}^x_{t,(1,2)}\\ \bar{\chi}^x_{t,(2,2)}\\ \bar{\chi}^x_{t,(3,2)}\end{pmatrix}+\bar{\chi}^z_{t(4,57)} \\
h_{LRF,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(5,1)} & h_{LRF,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(5,2)} & \cdots & h_{LRF,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(5,56)} & h_{LRF,1}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(5,57)} \\
\vdots & \vdots & \cdots & \vdots & \vdots \\
h_{LRF,19}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(23,1)} & h_{LRF,19}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(23,2)} & \cdots & h_{LRF,19}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(23,56)} & h_{LRF,19}\begin{pmatrix}\bar{\chi}^x_{t,(1,1)}\\ \bar{\chi}^x_{t,(2,1)}\\ \bar{\chi}^x_{t,(3,1)}\end{pmatrix}+\bar{\chi}^z_{t(23,57)}
\end{bmatrix}_{(23\times57)} \tag{3.26}
$$

The Equation 3.26 generates the measurement sigma points $\bar{Z}_t$. The measurement model h incorporates the ultrasonic measurement model h_{usat} and laser range finder measurement model h_{LRF} . Basically this equation fuses the ultrasonic sensor and laser range finder measurements. Equation 3.27 shows the calculation of predicted measurement $\hat{z}_t$ which requires the sum of measurement sigma points $\bar{Z}_t$ with weighted mean $w_i^{(m)}$.

$$\hat{z}_t = \sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{i,t}$$

$$= w_0^{(m)} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} \end{bmatrix} + w_1^{(m)} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,2)} \\ \bar{Z}^{usat}_{t,(2,2)} \\ \bar{Z}^{usat}_{t,(3,2)} \\ \bar{Z}^{usat}_{t,(4,2)} \\ \bar{Z}^{LRF}_{t,(1,2)} \\ \bar{Z}^{LRF}_{t,(2,2)} \\ \bar{Z}^{LRF}_{t,(3,2)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,2)} \\ \bar{Z}^{LRF}_{t,(18,2)} \\ \bar{Z}^{LRF}_{t,(19,2)} \end{bmatrix} + \cdots + w_{55}^{(m)} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,55)} \\ \bar{Z}^{usat}_{t,(2,55)} \\ \bar{Z}^{usat}_{t,(3,55)} \\ \bar{Z}^{usat}_{t,(4,55)} \\ \bar{Z}^{LRF}_{t,(1,55)} \\ \bar{Z}^{LRF}_{t,(2,55)} \\ \bar{Z}^{LRF}_{t,(3,55)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,55)} \\ \bar{Z}^{LRF}_{t,(18,55)} \\ \bar{Z}^{LRF}_{t,(19,55)} \end{bmatrix} + w_{56}^{(m)} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,56)} \\ \bar{Z}^{usat}_{t,(2,56)} \\ \bar{Z}^{usat}_{t,(3,56)} \\ \bar{Z}^{usat}_{t,(4,56)} \\ \bar{Z}^{LRF}_{t,(1,56)} \\ \bar{Z}^{LRF}_{t,(2,56)} \\ \bar{Z}^{LRF}_{t,(3,56)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,56)} \\ \bar{Z}^{LRF}_{t,(18,56)} \\ \bar{Z}^{LRF}_{t,(19,56)} \end{bmatrix} = \begin{bmatrix} \hat{z}^{usat}_{t,(1,1)} \\ \hat{z}^{usat}_{t,(2,1)} \\ \hat{z}^{usat}_{t,(3,1)} \\ \hat{z}^{usat}_{t,(4,1)} \\ \hat{z}^{LRF}_{t,(1,1)} \\ \hat{z}^{LRF}_{t,(2,1)} \\ \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \hat{z}^{LRF}_{t,(17,1)} \\ \hat{z}^{LRF}_{t,(18,1)} \\ \hat{z}^{LRF}_{t,(19,1)} \end{bmatrix}_{(23\times1)} \qquad (3.27)$$

The variable $\hat{z}_t$ describes the predicted ultrasonic sensor and laser range finder measurements. Equation 3.28 illustrates the measurement covariance matrix which calculates through measurement sigma points $\bar{Z}_t$, predicted measurement $\hat{z}_t$ and the weighted sigma point covariance $w_i^{(c)}$.

$$S_t = \sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T$$

$$w_0^{(c)} \left(\begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix}^T \right) + w_1^{(c)} \left(\begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix}^T \right) + \cdots$$

$$+ w_{55}^{(c)} \left(\begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix}^T \right) + w_{56}^{(c)} \left(\begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix} \begin{bmatrix} \bar{Z}^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ \bar{Z}^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ \bar{Z}^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ \bar{Z}^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ \bar{Z}^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ \bar{Z}^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ \bar{Z}^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ \bar{Z}^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(21,1)} \\ \bar{Z}^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(22,1)} \\ \bar{Z}^{LRF}_{t,(19,1)} - \hat{z}^{LRF}_{t,(23,1)} \end{bmatrix}^T \right)$$

$$\begin{bmatrix} S_{t,(1,1)} & S_{t,(1,2)} & S_{t,(1,3)} & S_{t,(1,4)} & S_{t,(1,5)} & S_{t,(1,6)} & S_{t,(1,7)} & \cdots & S_{t,(1,17)} & S_{t,(1,18)} & S_{t,(1,19)} & S_{t,(1,20)} & S_{t,(1,21)} & S_{t,(1,22)} & S_{t,(1,23)} \\ S_{t,(2,1)} & S_{t,(2,2)} & S_{t,(2,3)} & S_{t,(2,4)} & S_{t,(2,5)} & S_{t,(1,6)} & S_{t,(2,7)} & \cdots & S_{t,(2,17)} & S_{t,(2,18)} & S_{t,(2,19)} & S_{t,(2,20)} & S_{t,(2,21)} & S_{t,(2,22)} & S_{t,(2,23)} \\ S_{t,(3,1)} & S_{t,(3,2)} & S_{t,(3,3)} & S_{t,(3,4)} & S_{t,(3,5)} & S_{t,(1,6)} & S_{t,(3,7)} & \cdots & S_{t,(3,1)} & S_{t,(3,18)} & S_{t,(3,19)} & S_{t,(3,20)} & S_{t,(3,21)} & S_{t,(3,22)} & S_{t,(3,23)} \\ S_{t,(4,1)} & S_{t,(4,2)} & S_{t,(4,3)} & S_{t,(4,4)} & S_{t,(4,5)} & S_{t,(1,6)} & S_{t,(4,7)} & \cdots & S_{t,(4,17)} & S_{t,(4,18)} & S_{t,(4,19)} & S_{t,(4,20)} & S_{t,(4,21)} & S_{t,(4,22)} & S_{t,(4,23)} \\ S_{t,(5,1)} & S_{t,(5,2)} & S_{t,(5,3)} & S_{t,(5,4)} & S_{t,(5,5)} & S_{t,(1,6)} & S_{t,(5,7)} & \cdots & S_{t,(5,17)} & S_{t,(5,18)} & S_{t,(5,19)} & S_{t,(5,20)} & S_{t,(5,21)} & S_{t,(5,22)} & S_{t,(5,23)} \\ S_{t,(6,1)} & S_{t,(6,2)} & S_{t,(6,3)} & S_{t,(6,4)} & S_{t,(6,5)} & S_{t,(6,6)} & S_{t,(6,7)} & \cdots & S_{t,(6,17)} & S_{t,(6,18)} & S_{t,(6,19)} & S_{t,(6,20)} & S_{t,(6,21)} & S_{t,(6,22)} & S_{t,(6,23)} \\ S_{t,(7,1)} & S_{t,(6,2)} & S_{t,(7,3)} & S_{t,(7,4)} & S_{t,(7,5)} & S_{t,(1,6)} & S_{t,(7,7)} & \cdots & S_{t,(7,17)} & S_{t,(7,18)} & S_{t,(7,19)} & S_{t,(7,20)} & S_{t,(7,21)} & S_{t,(7,22)} & S_{t,(7,23)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{t,(17,1)} & S_{t,(17,2)} & S_{t,(16,3)} & S_{t,(17,4)} & S_{t,(17,5)} & S_{t,(1,6)} & S_{t,(17,7)} & \cdots & S_{t,(17,17)} & S_{t,(17,18)} & S_{t,(17,19)} & S_{t,(17,20)} & S_{t,(17,21)} & S_{t,(17,22)} & S_{t,(17,23)} \\ S_{t,(18,1)} & S_{t,(18,2)} & S_{t,(17,3)} & S_{t,(18,4)} & S_{t,(18,5)} & S_{t,(1,6)} & S_{t,(18,7)} & \cdots & S_{t,(18,17)} & S_{t,(18,18)} & S_{t,(18,19)} & S_{t,(18,20)} & S_{t,(18,21)} & S_{t,(18,22)} & S_{t,(18,23)} \\ S_{t,(19,1)} & S_{t,(19,2)} & S_{t,(18,3)} & S_{t,(19,4)} & S_{t,(19,5)} & S_{t,(1,6)} & S_{t,(19,7)} & \cdots & S_{t,(19,17)} & S_{t,(19,18)} & S_{t,(19,19)} & S_{t,(19,20)} & S_{t,(19,21)} & S_{t,(19,22)} & S_{t,(19,23)} \\ S_{t,(20,1)} & S_{t,(20,2)} & S_{t,(19,3)} & S_{t,(20,4)} & S_{t,(20,5)} & S_{t,(1,6)} & S_{t,(20,7)} & \cdots & S_{t,(20,17)} & S_{t,(20,18)} & S_{t,(20,19)} & S_{t,(20,20)} & S_{t,(20,21)} & S_{t,(20,22)} & S_{t,(20,23)} \\ S_{t,(21,1)} & S_{t,(21,2)} & S_{t,(21,3)} & S_{t,(21,4)} & S_{t,(21,5)} & S_{t,(1,6)} & S_{t,(21,7)} & \cdots & S_{t,(1,17)} & S_{t,(21,18)} & S_{t,(21,19)} & S_{t,(21,20)} & S_{t,(21,21)} & S_{t,(21,22)} & S_{t,(21,23)} \\ S_{t,(22,1)} & S_{t,(22,2)} & S_{t,(22,3)} & S_{t,(22,4)} & S_{t,(22,5)} & S_{t,(1,6)} & S_{t,(22,7)} & \cdots & S_{t,(1,17)} & S_{t,(22,18)} & S_{t,(22,19)} & S_{t,(22,20)} & S_{t,(22,21)} & S_{t,(22,22)} & S_{t,(22,23)} \\ S_{t,(23,1)} & S_{t,(23,2)} & S_{t,(23,3)} & S_{t,(23,4)} & S_{t,(23,5)} & S_{t,(23,6)} & S_{t,(23,7)} & \cdots & S_{t,(23,17)} & S_{t,(23,18)} & S_{t,(23,19)} & S_{t,(23,20)} & S_{t,(23,21)} & S_{t,(23,22)} & S_{t,(23,23)} \end{bmatrix}_{(23\times23)} \tag{3.28}$$

Equation 3.29 illustrates the cross covariance, which describes the covariance between the position components sigma points $\overline{\chi}_t^x$ and the measurement sigma points $\overline{Z}_t$.

$$\Sigma_t^{x,z} = \sum_{i=0}^{2N} w_i^{(c)} (\overline{\chi}_{i,t}^x - \overline{\mu}_t)(\overline{Z}_{i,t} - \hat{z}_t)^T$$

$$= w_0^{(c)} \begin{bmatrix} \overline{\chi}_{t,(1,1)}^x - \overline{\mu}_{x,t} \\ \overline{\chi}_{t,(2,1)}^x - \overline{\mu}_{y,t} \\ \overline{\chi}_{t,(3,1)}^x - \overline{\mu}_{\theta,t} \end{bmatrix} \left(\begin{bmatrix} \overline{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat} \\ \overline{Z}_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat} \\ \overline{Z}_{t,(3,1)}^{usat} - \hat{z}_{t,(3,1)}^{usat} \\ \overline{Z}_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat} \\ \overline{Z}_{t,(1,1)}^{LRF} - \hat{z}_{t,(1,1)}^{LRF} \\ \overline{Z}_{t,(2,1)}^{LRF} - \hat{z}_{t,(2,1)}^{LRF} \\ \overline{Z}_{t,(3,1)}^{LRF} - \hat{z}_{t,(3,1)}^{LRF} \\ \vdots \\ \overline{Z}_{t,(17,1)}^{LRF} - \hat{z}_{t,(21,1)}^{LRF} \\ \overline{Z}_{t,(18,1)}^{LRF} - \hat{z}_{t,(22,1)}^{LRF} \\ \overline{Z}_{t,(19,1)}^{LRF} - \hat{z}_{t,(23,1)}^{LRF} \end{bmatrix} \right)^T + w_1^{(c)} \begin{bmatrix} \overline{\chi}_{t,(1,1)}^x - \overline{\mu}_{x,t} \\ \overline{\chi}_{t,(2,1)}^x - \overline{\mu}_{y,t} \\ \overline{\chi}_{t,(3,1)}^x - \overline{\mu}_{\theta,t} \end{bmatrix} \left(\begin{bmatrix} \overline{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat} \\ \overline{Z}_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat} \\ \overline{Z}_{t,(3,1)}^{usat} - \hat{z}_{t,(3,1)}^{usat} \\ \overline{Z}_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat} \\ \overline{Z}_{t,(1,1)}^{LRF} - \hat{z}_{t,(1,1)}^{LRF} \\ \overline{Z}_{t,(2,1)}^{LRF} - \hat{z}_{t,(2,1)}^{LRF} \\ \overline{Z}_{t,(3,1)}^{LRF} - \hat{z}_{t,(3,1)}^{LRF} \\ \vdots \\ \overline{Z}_{t,(17,1)}^{LRF} - \hat{z}_{t,(21,1)}^{LRF} \\ \overline{Z}_{t,(18,1)}^{LRF} - \hat{z}_{t,(22,1)}^{LRF} \\ \overline{Z}_{t,(19,1)}^{LRF} - \hat{z}_{t,(23,1)}^{LRF} \end{bmatrix} \right)^T + \cdots$$

$$+ w_{55}^{(c)} \begin{bmatrix} \overline{\chi}_{t,(1,1)}^x - \overline{\mu}_{x,t} \\ \overline{\chi}_{t,(2,1)}^x - \overline{\mu}_{y,t} \\ \overline{\chi}_{t,(3,1)}^x - \overline{\mu}_{\theta,t} \end{bmatrix} \left(\begin{bmatrix} \overline{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat} \\ \overline{Z}_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat} \\ \overline{Z}_{t,(3,1)}^{usat} - \hat{z}_{t,(3,1)}^{usat} \\ \overline{Z}_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat} \\ \overline{Z}_{t,(1,1)}^{LRF} - \hat{z}_{t,(1,1)}^{LRF} \\ \overline{Z}_{t,(2,1)}^{LRF} - \hat{z}_{t,(2,1)}^{LRF} \\ \overline{Z}_{t,(3,1)}^{LRF} - \hat{z}_{t,(3,1)}^{LRF} \\ \vdots \\ \overline{Z}_{t,(17,1)}^{LRF} - \hat{z}_{t,(21,1)}^{LRF} \\ \overline{Z}_{t,(18,1)}^{LRF} - \hat{z}_{t,(22,1)}^{LRF} \\ \overline{Z}_{t,(19,1)}^{LRF} - \hat{z}_{t,(23,1)}^{LRF} \end{bmatrix} \right)^T + w_{57}^{(c)} \begin{bmatrix} \overline{\chi}_{t,(1,1)}^x - \overline{\mu}_{x,t} \\ \overline{\chi}_{t,(2,1)}^x - \overline{\mu}_{y,t} \\ \overline{\chi}_{t,(3,1)}^x - \overline{\mu}_{\theta,t} \end{bmatrix} \left(\begin{bmatrix} \overline{Z}_{t,(1,1)}^{usat} - \hat{z}_{t,(1,1)}^{usat} \\ \overline{Z}_{t,(2,1)}^{usat} - \hat{z}_{t,(2,1)}^{usat} \\ \overline{Z}_{t,(3,1)}^{usat} - \hat{z}_{t,(3,1)}^{usat} \\ \overline{Z}_{t,(4,1)}^{usat} - \hat{z}_{t,(4,1)}^{usat} \\ \overline{Z}_{t,(1,1)}^{LRF} - \hat{z}_{t,(1,1)}^{LRF} \\ \overline{Z}_{t,(2,1)}^{LRF} - \hat{z}_{t,(2,1)}^{LRF} \\ \overline{Z}_{t,(3,1)}^{LRF} - \hat{z}_{t,(3,1)}^{LRF} \\ \vdots \\ \overline{Z}_{t,(17,1)}^{LRF} - \hat{z}_{t,(21,1)}^{LRF} \\ \overline{Z}_{t,(18,1)}^{LRF} - \hat{z}_{t,(22,1)}^{LRF} \\ \overline{Z}_{t,(19,1)}^{LRF} - \hat{z}_{t,(23,1)}^{LRF} \end{bmatrix} \right)^T$$

$$
= \begin{bmatrix}
\Sigma^{x,z}_{t,(1,1)} & \Sigma^{x,z}_{t,(1,2)} & \Sigma^{x,z}_{t,(1,3)} & \Sigma^{x,z}_{t,(1,4)} & \cdots & \Sigma^{x,z}_{t,(1,20)} & \Sigma^{x,z}_{t,(1,21)} & \Sigma^{x,z}_{t,(1,22)} & \Sigma^{x,z}_{t,(1,23)} \\
\Sigma^{x,z}_{t,(2,1)} & \Sigma^{x,z}_{t,(2,2)} & \Sigma^{x,z}_{t,(2,3)} & \Sigma^{x,z}_{t,(2,4)} & \cdots & \Sigma^{x,z}_{t,(2,20)} & \Sigma^{x,z}_{t,(2,21)} & \Sigma^{x,z}_{t,(3,22)} & \Sigma^{x,z}_{t,(2,23)} \\
\Sigma^{x,z}_{t,(3,1)} & \Sigma^{x,z}_{t,(3,2)} & \Sigma^{x,z}_{t,(3,3)} & \Sigma^{x,z}_{t,(3,4)} & \cdots & \Sigma^{x,z}_{t,(3,20)} & \Sigma^{x,z}_{t,(3,21)} & \Sigma^{x,z}_{t,(3,22)} & \Sigma^{x,z}_{t,(3,23)}
\end{bmatrix}_{(3\times 23)} \tag{3.29}
$$

The filter update step consists of unscented Kalman gain K_t calculation, updates the predicted mean and covariance. Equation 3.30, 3.31 and 3.32 illustrates the Kalman gain, updates mean and updates covariance calculations.

$$
\begin{aligned}
K_t &= \Sigma^{x,z}_t S^{-1}_t \\
&= \left(\begin{bmatrix}
\Sigma^{x,z}_{t,(1,1)} & \Sigma^{x,z}_{t,(1,2)} & \Sigma^{x,z}_{t,(1,3)} & \Sigma^{x,z}_{t,(1,4)} & \cdots & \Sigma^{x,z}_{t,(1,20)} & \Sigma^{x,z}_{t,(1,21)} & \Sigma^{x,z}_{t,(1,22)} & \Sigma^{x,z}_{t,(1,23)} \\
\Sigma^{x,z}_{t,(2,1)} & \Sigma^{x,z}_{t,(2,2)} & \Sigma^{x,z}_{t,(2,3)} & \Sigma^{x,z}_{t,(2,4)} & \cdots & \Sigma^{x,z}_{t,(2,20)} & \Sigma^{x,z}_{t,(2,21)} & \Sigma^{x,z}_{t,(3,22)} & \Sigma^{x,z}_{t,(2,23)} \\
\Sigma^{x,z}_{t,(3,1)} & \Sigma^{x,z}_{t,(3,2)} & \Sigma^{x,z}_{t,(3,3)} & \Sigma^{x,z}_{t,(3,4)} & \cdots & \Sigma^{x,z}_{t,(3,20)} & \Sigma^{x,z}_{t,(3,21)} & \Sigma^{x,z}_{t,(3,22)} & \Sigma^{x,z}_{t,(3,23)}
\end{bmatrix} \right) \\
&\left(\begin{bmatrix}
S_{t,(1,1)} & S_{t,(1,2)} & S_{t,(1,3)} & S_{t,(1,4)} & S_{t,(1,5)} & S_{t,(1,6)} & S_{t,(1,7)} & \cdots & S_{t,(1,17)} & S_{t,(1,18)} & S_{t,(1,19)} & S_{t,(1,20)} & S_{t,(1,21)} & S_{t,(1,22)} & S_{t,(1,23)} \\
S_{t,(2,1)} & S_{t,(2,2)} & S_{t,(2,3)} & S_{t,(2,4)} & S_{t,(2,5)} & S_{t,(1,6)} & S_{t,(2,7)} & \cdots & S_{t,(2,17)} & S_{t,(2,18)} & S_{t,(2,19)} & S_{t,(2,20)} & S_{t,(2,21)} & S_{t,(2,22)} & S_{t,(2,23)} \\
S_{t,(3,1)} & S_{t,(3,2)} & S_{t,(3,3)} & S_{t,(3,4)} & S_{t,(3,5)} & S_{t,(1,6)} & S_{t,(3,7)} & \cdots & S_{t,(3,1)} & S_{t,(3,18)} & S_{t,(3,19)} & S_{t,(3,20)} & S_{t,(3,21)} & S_{t,(3,22)} & S_{t,(3,23)} \\
S_{t,(4,1)} & S_{t,(4,2)} & S_{t,(4,3)} & S_{t,(4,4)} & S_{t,(4,5)} & S_{t,(1,6)} & S_{t,(4,7)} & \cdots & S_{t,(4,17)} & S_{t,(4,18)} & S_{t,(4,19)} & S_{t,(4,20)} & S_{t,(4,21)} & S_{t,(4,22)} & S_{t,(4,23)} \\
S_{t,(5,1)} & S_{t,(5,2)} & S_{t,(5,3)} & S_{t,(5,4)} & S_{t,(5,5)} & S_{t,(1,6)} & S_{t,(5,7)} & \cdots & S_{t,(5,17)} & S_{t,(5,18)} & S_{t,(5,19)} & S_{t,(5,20)} & S_{t,(5,21)} & S_{t,(5,22)} & S_{t,(5,23)} \\
S_{t,(6,1)} & S_{t,(6,2)} & S_{t,(6,3)} & S_{t,(6,4)} & S_{t,(6,5)} & S_{t,(6,6)} & S_{t,(6,7)} & \cdots & S_{t,(6,17)} & S_{t,(6,18)} & S_{t,(6,19)} & S_{t,(6,20)} & S_{t,(6,21)} & S_{t,(6,22)} & S_{t,(6,23)} \\
S_{t,(7,1)} & S_{t,(6,2)} & S_{t,(7,3)} & S_{t,(7,4)} & S_{t,(7,5)} & S_{t,(1,6)} & S_{t,(7,7)} & \cdots & S_{t,(7,17)} & S_{t,(7,18)} & S_{t,(7,19)} & S_{t,(7,20)} & S_{t,(7,21)} & S_{t,(7,22)} & S_{t,(7,23)} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
S_{t,(17,1)} & S_{t,(17,2)} & S_{t,(16,3)} & S_{t,(17,4)} & S_{t,(17,5)} & S_{t,(1,6)} & S_{t,(17,7)} & \cdots & S_{t,(17,17)} & S_{t,(17,18)} & S_{t,(17,19)} & S_{t,(17,20)} & S_{t,(17,21)} & S_{t,(17,22)} & S_{t,(17,23)} \\
S_{t,(18,1)} & S_{t,(18,2)} & S_{t,(17,3)} & S_{t,(18,4)} & S_{t,(18,5)} & S_{t,(1,6)} & S_{t,(18,7)} & \cdots & S_{t,(18,17)} & S_{t,(18,18)} & S_{t,(18,19)} & S_{t,(18,20)} & S_{t,(18,21)} & S_{t,(18,22)} & S_{t,(18,23)} \\
S_{t,(19,1)} & S_{t,(19,2)} & S_{t,(18,3)} & S_{t,(19,4)} & S_{t,(19,5)} & S_{t,(1,6)} & S_{t,(19,7)} & \cdots & S_{t,(19,17)} & S_{t,(19,18)} & S_{t,(19,19)} & S_{t,(19,20)} & S_{t,(19,21)} & S_{t,(19,22)} & S_{t,(19,23)} \\
S_{t,(20,1)} & S_{t,(20,2)} & S_{t,(19,3)} & S_{t,(20,4)} & S_{t,(20,5)} & S_{t,(1,6)} & S_{t,(20,7)} & \cdots & S_{t,(20,17)} & S_{t,(20,18)} & S_{t,(20,19)} & S_{t,(20,20)} & S_{t,(20,21)} & S_{t,(20,22)} & S_{t,(20,23)} \\
S_{t,(21,1)} & S_{t,(21,2)} & S_{t,(21,3)} & S_{t,(21,4)} & S_{t,(21,5)} & S_{t,(1,6)} & S_{t,(21,7)} & \cdots & S_{t,(1,17)} & S_{t,(21,18)} & S_{t,(21,19)} & S_{t,(21,20)} & S_{t,(21,21)} & S_{t,(21,22)} & S_{t,(21,23)} \\
S_{t,(22,1)} & S_{t,(22,2)} & S_{t,(22,3)} & S_{t,(22,4)} & S_{t,(22,5)} & S_{t,(1,6)} & S_{t,(22,7)} & \cdots & S_{t,(1,17)} & S_{t,(22,18)} & S_{t,(22,19)} & S_{t,(22,20)} & S_{t,(22,21)} & S_{t,(22,22)} & S_{t,(22,23)} \\
S_{t,(23,1)} & S_{t,(23,2)} & S_{t,(23,3)} & S_{t,(23,4)} & S_{t,(23,5)} & S_{t,(23,6)} & S_{t,(23,7)} & \cdots & S_{t,(23,17)} & S_{t,(23,18)} & S_{t,(23,19)} & S_{t,(23,20)} & S_{t,(23,21)} & S_{t,(23,22)} & S_{t,(23,23)}
\end{bmatrix} \right)^{-1} \\
&= \left[\begin{bmatrix}
K_{t,(1,1)} & K_{t,(1,2)} & K_{t,(1,3)} & K_{t,(1,4)} & \cdots & K_{t,(1,20)} & K_{t,(1,21)} & K_{t,(1,22)} & K_{t,(1,23)} \\
K_{t,(2,1)} & K_{t,(2,2)} & K_{t,(2,3)} & K_{t,(2,4)} & \cdots & K_{t,(2,20)} & K_{t,(2,21)} & K_{t,(2,22)} & K_{t,(2,23)} \\
K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & \cdots & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(1,23)}
\end{bmatrix} \right]_{(3\times 23)}
\end{aligned} \tag{3.30}
$$

Equation 3.31 and 3.32 illustrates the updated mean and covariance matrices.

$$
\begin{aligned}
\mu_t &= \bar{\mu}_t + K_t(z_t - \hat{z}_t) \\
&= \begin{bmatrix} \bar{\mu}_{x,t} \\ \bar{\mu}_{y,t} \\ \bar{\mu}_{\theta,t} \end{bmatrix} + \left(\begin{bmatrix} K_{t,(1,1)} & K_{t,(1,2)} & K_{t,(1,3)} & K_{t,(1,4)} & \cdots & K_{t,(1,20)} & K_{t,(1,21)} & K_{t,(1,22)} & K_{t,(1,23)} \\ K_{t,(2,1)} & K_{t,(2,2)} & K_{t,(2,3)} & K_{t,(2,4)} & \cdots & K_{t,(2,20)} & K_{t,(2,21)} & K_{t,(2,22)} & K_{t,(2,23)} \\ K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & \cdots & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(1,23)} \end{bmatrix} \right) \left(\begin{bmatrix} z^{usat}_{t,(1,1)} - \hat{z}^{usat}_{t,(1,1)} \\ z^{usat}_{t,(2,1)} - \hat{z}^{usat}_{t,(2,1)} \\ z^{usat}_{t,(3,1)} - \hat{z}^{usat}_{t,(3,1)} \\ z^{usat}_{t,(4,1)} - \hat{z}^{usat}_{t,(4,1)} \\ z^{LRF}_{t,(1,1)} - \hat{z}^{LRF}_{t,(1,1)} \\ z^{LRF}_{t,(2,1)} - \hat{z}^{LRF}_{t,(2,1)} \\ z^{LRF}_{t,(3,1)} - \hat{z}^{LRF}_{t,(3,1)} \\ \vdots \\ z^{LRF}_{t,(17,1)} - \hat{z}^{LRF}_{t,(17,1)} \\ z^{LRF}_{t,(18,1)} - \hat{z}^{LRF}_{t,(18,1)} \\ z_{t,(19,1)} - \hat{z}_{t,(19,1)} \end{bmatrix} \right) = \begin{bmatrix} \bar{\mu}_{x,t} \\ \bar{\mu}_{y,t} \\ \bar{\mu}_{\theta,t} \end{bmatrix}_{(3\times1)}
\end{aligned} \tag{3.31}
$$

$$
\begin{aligned}
\Sigma_t &= \bar{\Sigma}_t - K_t S_t K_t^T \\
&= \left(\begin{bmatrix} \bar{\Sigma}_{xx,t} & \bar{\Sigma}_{xy,t} & \bar{\Sigma}_{x\theta,t} \\ \bar{\Sigma}_{yx,t} & \bar{\Sigma}_{yy,t} & \bar{\Sigma}_{y\theta,t} \\ \bar{\Sigma}_{\theta x,t} & \bar{\Sigma}_{\theta y,t} & \bar{\Sigma}_{\theta\theta t} \end{bmatrix} \right) \left(\begin{bmatrix} K_{t,(1,1)} & K_{t,(1,2)} & K_{t,(1,3)} & K_{t,(1,4)} & \cdots & K_{t,(1,20)} & K_{t,(1,21)} & K_{t,(1,22)} & K_{t,(1,23)} \\ K_{t,(2,1)} & K_{t,(2,2)} & K_{t,(2,3)} & K_{t,(2,4)} & \cdots & K_{t,(2,20)} & K_{t,(2,21)} & K_{t,(2,22)} & K_{t,(2,23)} \\ K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & \cdots & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(1,23)} \end{bmatrix} \right) \\
&\quad \left(\begin{bmatrix} S_{t,(1,1)} & S_{t,(1,2)} & S_{t,(1,3)} & S_{t,(1,4)} & \cdots & S_{t,(1,20)} & S_{t,(1,21)} & S_{t,(1,22)} & S_{t,(1,23)} \\ S_{t,(2,1)} & S_{t,(2,2)} & S_{t,(2,3)} & S_{t,(2,4)} & \cdots & S_{t,(2,20)} & S_{t,(2,21)} & S_{t,(2,22)} & S_{t,(2,23)} \\ S_{t,(3,1)} & S_{t,(3,2)} & S_{t,(3,3)} & S_{t,(3,4)} & \cdots & S_{t,(3,20)} & S_{t,(3,21)} & S_{t,(3,22)} & S_{t,(3,23)} \\ S_{t,(4,1)} & S_{t,(4,2)} & S_{t,(4,3)} & S_{t,(4,4)} & \cdots & S_{t,(4,20)} & S_{t,(4,21)} & S_{t,(4,22)} & S_{t,(4,23)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ S_{t,(20,1)} & S_{t,(20,2)} & S_{t,(20,3)} & S_{t,(20,4)} & \cdots & S_{t,(20,20)} & S_{t,(20,21)} & S_{t,(20,22)} & S_{t,(20,23)} \\ S_{t,(21,1)} & S_{t,(21,2)} & S_{t,(21,3)} & S_{t,(21,4)} & \cdots & S_{t,(21,20)} & S_{t,(21,21)} & S_{t,(21,22)} & S_{t,(21,23)} \\ S_{t,(22,1)} & S_{t,(22,2)} & S_{t,(22,3)} & S_{t,(22,4)} & \cdots & S_{t,(22,20)} & S_{t,(22,21)} & S_{t,(22,22)} & S_{t,(22,23)} \\ S_{t,(23,1)} & S_{t,(23,2)} & S_{t,(23,3)} & S_{t,(23,4)} & \cdots & S_{t,(23,21)} & S_{t,(23,22)} & S_{t,(23,22)} & S_{t,(23,23)} \end{bmatrix} \right) \\
&\quad \left(\begin{bmatrix} K_{t,(1,1)} & K_{t,(1,2)} & K_{t,(1,3)} & K_{t,(1,4)} & \cdots & K_{t,(1,20)} & K_{t,(1,21)} & K_{t,(1,22)} & K_{t,(1,23)} \\ K_{t,(2,1)} & K_{t,(2,2)} & K_{t,(2,3)} & K_{t,(2,4)} & \cdots & K_{t,(2,20)} & K_{t,(2,21)} & K_{t,(2,22)} & K_{t,(2,23)} \\ K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & K_{t,(1,1)} & \cdots & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(3,23)} & K_{t,(1,23)} \end{bmatrix} \right)^T \\
&= \begin{bmatrix} \Sigma_{xx,t} & \Sigma_{xy,t} & \Sigma_{x\theta,t} \\ \Sigma_{yx,t} & \Sigma_{yy,t} & \Sigma_{y\theta,t} \\ \Sigma_{\theta x,t} & \Sigma_{\theta y,t} & \Sigma_{\theta\theta t} \end{bmatrix}_{(3\times3)}
\end{aligned} \tag{3.32}
$$

Table 3.4 depicts the complete pseudo code of UKF correction step with the correspondence calculation pseudo code. The next sections discuss the implementation of UKF localization algorithm with known correspondence in our experiment.

```
Correction step $(\bar{\mu}_{t-1}, \bar{\Sigma}_{t-1}, z_t, m)$
-----------------------------------------------------------------
for the correspondence calculation
-----------------------------------------------------------------
for all actual sensor data $z_{t,k}^{usat} = \left(r_{t,k}^{usat}\right)_{(4\times1)}$
        for all USAT beacons $j$ in the map $'m'$ do
            $\bar{Z}_{t,j}^{usat} = (h(\bar{\chi}_t^{x}) + \bar{\chi}_t^{z_{usat}})_{(1\times57)}$
            $\hat{z}_{t,j}^{usat} = (\sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{t,i}^{usat})_{(1\times1)}$
            $S_{t,j}^{usat} = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{t,i}^{usat} - \hat{z}_t^{usat})(\bar{Z}_{t,i}^{usat} - \hat{z}_t^{usat})^T)_{(1\times1)}$
            $ML_j^k = (\det(2\pi S_{t,j}^{usat})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})^T \left[S_{t,j}^{usat}\right]^{-1} (z_{t,k}^{usat} - \hat{z}_{t,j}^{usat})\})_{4\times1}$
        end
    $c_t^k = \mathbf{argmax}_k \ (ML)$
end
-----------------------------------------------------------------
$\bar{Z}_t = (h(\bar{\chi}_t^{x}) + \bar{\chi}_t^{z})_{(23\times57)}$
$\hat{z}_t = (\sum_{i=0}^{2N} w_i^{(m)} \bar{Z}_{i,t})_{(23\times1)}$
$S_t = (\sum_{i=0}^{2N} w_i^{(c)} (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T)_{(23\times23)}$
$\Sigma_t^{x,z} = (\sum_{i=0}^{2N} w_i^{(c)} (\bar{\chi}_{i,t}^{x} - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T)_{(3\times23)}$
$K_t = \Sigma_t^{x,z} S_t$
$\mu_t = \bar{\mu}_t + K_t^j (z_t - \hat{z}_t)$
$\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$
$\mu_t = \bar{\mu}_t$
$\Sigma_t = \bar{\Sigma}_t$
return $\mu_t, \Sigma_t$
-----------------------------------------------------------------
```

Table 3.4. Pseudo code of UKF correction step

3.4 UKF Localization Algorithm using ultrasonic sensor with unknown correspondence

This section summarizes the UKF algorithm using only ultrasonic sensor measurement. As it is already describes in the previous section of this chapter we have used $N_{usat} = 4$ ultrasonic measurements in robot localization. This algorithm assumes all the ultrasonic beacons detection is contained in the observation z_t and we assume the data association to beacon is unknown. The correspondence problem is solved through maximum likelihood estimation (MLE). MLE first determines the most likely value of the correspondence vector c_t, and then take this value for granted.

In this algorithm the dimensionality L of the augmented state vector is given by the sum of robot state, control and $N_{usat} = 4$ ultrasonic measurements dimensions, which is $3+2+N_{usat}$. The augmented state estimate generates $2L+1=19$ sigma points each having components in state, control and ultrasonic measurement space.

Table 3.5 illustrates the complete pseudo code of UKF localization algorithm using ultrasonic measurement with unknown correspondence. Lines 3 and 4 generate the augmented mean and covariance matrices. Line 5 calculates the sigma points χ^a_{t-1}. Line 6 to 8 performs the prediction step. In line 6 which the location components of sigma point χ^x_{t-1} undergoes the motion model g using the control and additional control noise components χ^u_t. Line 7 and 8 computes the predicted mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$.

Line 11 computes the measurement sigma points $\bar{Z}^j_t$. In line 9 the variable z^k_t denotes the each k^{th} actual ultrasonic sensor actual range information r_t. Line 12 shows the calculated measurement $\hat{z}^j_t$ from each j^{th} beacon position in the map m. Line 13 gives measurement covariance S_t^j of each $\hat{z}_t^j$. Line 14 computes the maximum likelihood of the measurement $\hat{z}_t^j$ of any possible beacon the in the map. In line 16 c_t denotes the correspondence vector containing the computed correspondence values associated with each k^{th} actual ultrasonic sensor measurement.

Measurement model h in line 17 generates the measurement sigma points $\bar{Z}_t$ based on the correspondence c_t computed in line 16. Line 18 and 19 calculates the measurements $\hat{z}_t$ and measurement covariance S_t. Line 20 shows the cross covariance $\Sigma_t^{x,z}$ between robot predicted state and calculated measurement. Lines 21 to 25 shows the filter update step which computes Kalman gain K_t and updates the predicted mean μ_t and covariance Σ_t.

```
Algorithm: UKF Localization using Ultrasonic sensor $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$
$M_t = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{bmatrix}_{(2\times2)}$
$Q_t = \begin{bmatrix} \sigma_{r_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{r_2}^2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_{r_4} \end{bmatrix}_{(4\times4)}$
$\mu_{t-1}^a = \begin{bmatrix} \mu_{x,t-1} & \mu_{y,t-1} & \mu_{\theta,t-1} & \mu_{v,t-1} & \mu_{\omega,t-1} & \mu_{r_1,t-1} & \cdots & \mu_{r_4,t-1} \end{bmatrix}^T_{(9\times1)}$
    $\because \left( \mu_{v,t-1}, \mu_{\omega,t-1}, \mu_{r_1,t-1}, \ldots\ldots\ldots, \mu_{r_4,t-1} = 0 \right)$
$\Sigma_{t-1}^a = \begin{bmatrix} \Sigma_{x,t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{bmatrix}_{(9\times9)}$
$\chi_{t-1}^a = \left( \mu_{t-1}^a \quad \mu_{t-1}^a + \gamma\sqrt{\Sigma_{t-1}^a} \quad \mu_{t-1}^a - \gamma\sqrt{\Sigma_{t-1}^a} \right)_{(9\times19)}$
$\bar{\chi}_t^x = g\,(u_t + \chi_t^u, \chi_{t-1}^x)_{(3\times19)}$
$\bar{\mu}_t = (\sum_{i=0}^{2L} w_i^{(m)} + \bar{\chi}_{i,t}^x)_{(3\times1)}$
$\bar{\Sigma}_t = (\sum_{i=0}^{2L} w_i^{(c)} \left(\bar{\chi}_{i,t}^x - \bar{\mu}_t\right)\left(\bar{\chi}_{i,t}^x - \bar{\mu}_t\right))_{(3\times3)}$
for all actual sensor data $z_t^k = (r_t)_{(4\times1)}$
    for all USAT beacons $j$ in the map $'m'$ do
        $\bar{Z}_t^j = (h(\bar{\chi}_t^x) + \bar{\chi}_t^z)_{(1\times19)}$
        $\hat{z}_t^j = (\sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{t,i}^j)_{(1\times1)}$
        $S_t^j = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{Z}_{t,i}^j - \hat{z}_t^j)(\bar{Z}_{t,i}^j - \hat{z}_t^j)^T)_{(1\times1)}$
        $ML_j^k = \det(2\pi S_t^j)^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_t^k - \hat{z}_t^j)^T S_t^{j-1}(z_t^k - \hat{z}_t^j)\}_{(4\times1)}$
    end
    $c_t = \underset{k}{\mathbf{argmax}}\ (ML)$
end
$\bar{Z}_t = (h\,(\bar{\chi}_t^x) + \bar{\chi}_t^z)_{(4\times19)}$
$\hat{z}_t = (\sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{i,t})_{(4\times1)}$
$S_t = (\sum_{i=0}^{2L} w_i^{(m)} (\bar{Z}_{i,t} - \hat{z}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T)_{(4\times4)}$
$\Sigma_t^{x,z} = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{\chi}_{i,t}^x - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t)^T)_{(3\times4)}$
$K_t = (\Sigma_t^{x,z} S_t)_{(3\times4)}$
$\mu_t = (\bar{\mu}_t + K_t(z_t - \hat{z}_t))_{(3\times1)}$
$\Sigma_t = (\bar{\Sigma}_t - K_t S_t K_t^T)_{(3\times3)}$
$\mu_t = \bar{\mu}_t$
$\Sigma_t = \bar{\Sigma}_t$
return $\mu_t, \Sigma_t$
```

Table 3.5. Complete Pseudo code of UKF Localization with unknown correspondence using ultrasonic measurement

3.5 UKF Localization Algorithm using laser range finder

In this section we discuss the UKF localization algorithm using laser range finder measurements. In this algorithm all the correspondences related to laser range finder measurement, and the map information is known. We have used $N_{LRF} = 19$ laser beams to calculate the distance from the robot to the surroundings. The grid map information is used to model the geometry of robot's work space.

The dimensionality L of the augmented state vector is given by the sum of robot state, control and $N_{LRF} = 19$ laser range finder measurements dimensions, which is $3 + 2 + N_{usat}$. The augmented state estimate generates $2L + 1 = 49$ sigma points each having components in state, control and laser range finder measurement space. Table 3.6 depicts the complete pseudo code of this algorithm. Augmented mean and covariance matrices are generated in lines 3 and 4. Sigma points χ_{t-1}^{a} are calculated in line 5. The prediction step illustrated in lines 6 to 8. In line 6 the location components of sigma point χ_{t-1}^{x} undergoes the motion model using the control and additional control noise components χ_{t}^{u} . The line 7 and 8 compute the predicted mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$.

In Line 17 measurement model h generates the measurement sigma points $\bar{Z}_t$. Line 18 and 19 calculates the measurements $\hat{z}_t$ and measurement covariance S_t . Line 20 shows the cross covariance $\Sigma_t^{x,z}$ between robot predicted state and calculated measurement. Lines 21 to 25 shows the filter update step which computes Kalman gain K_t and updates the predicted mean μ_t and covariance Σ_t .

```
***Algorithm: UKF Localization using Laser Range Finder*** $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$
$M_t = \begin{bmatrix} \alpha_1 v_t^2 + \alpha_2 \omega_t^2 & 0 \\ 0 & \alpha_3 v_t^2 + \alpha_4 \omega_t^2 \end{bmatrix}_{(2\times2)}$
$Q_t = \begin{bmatrix} \sigma_{r_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{r_2}^2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_{r_{19}}^2 \end{bmatrix}_{(19\times19)}$
$\mu_{t-1}^a = \begin{bmatrix} \mu_{x,t-1} & \mu_{y,t-1} & \mu_{\theta,t-1} & (0 \quad 0) & (0 & \cdots & 0)_{1\times19} \end{bmatrix}^T_{(24\times1)}$
    $\because (\mu_{v,t-1}, \mu_{\omega,t-1}, \mu_{r_1,t-1}, \ldots\ldots\ldots, \mu_{r_{19},t-1} = 0\ )$
$\Sigma_{t-1}^a = \begin{bmatrix} \Sigma_{x,t-1} & 0 & 0 \\ 0 & M_t & 0 \\ 0 & 0 & Q_t \end{bmatrix}_{(24\times24)}$
$\chi_{t-1}^a = \left( \mu_{t-1}^a \quad \mu_{t-1}^a + \gamma\sqrt{\Sigma_{t-1}^a} \quad \mu_{t-1}^a - \gamma\sqrt{\Sigma_{t-1}^a} \right)_{(24\times49)}$
$\bar{\chi}_t^x = g(u_t + \chi_t^u, \chi_{t-1}^x)\ _{(3\times49)}$
$\bar{\mu}_t = (\sum_{i=0}^{2L} w_i^{(m)} + \bar{\chi}_{i,t}^x\ )_{(3\times1)}$
$\bar{\Sigma}_t = (\sum_{i=0}^{2L} w_i^{(c)} \left( \bar{\chi}_{i,t}^x - \bar{\mu}_t \right)\left( \bar{\chi}_{i,t}^x - \bar{\mu}_t \right))_{(3\times3)}$
$\bar{Z}_t = (h(\bar{\chi}_t^x) + \bar{\chi}_t^z)_{(19\times49)}$
$\hat{z}_t = (\sum_{i=0}^{2L} w_i^{(m)} \bar{Z}_{i,t}\ )_{(19\times1)}$
$S_t = (\sum_{i=0}^{2L} w_i^{(m)} (\bar{Z}_{i,t} - \hat{z}_t\ )(\bar{Z}_{i,t} - \hat{z}_t\ )^T\ )_{(19\times19)}$
$\Sigma_t^{x,z} = (\sum_{i=0}^{2L} w_i^{(c)} (\bar{\chi}_{i,t}^x - \bar{\mu}_t)(\bar{Z}_{i,t} - \hat{z}_t\ )^T\ )_{(3\times19)}$
$K_t = (\Sigma_t^{x,z} S_t)_{(3\times19)}$
$\mu_t = (\bar{\mu}_t + K_t(z_t - \hat{z}_t))_{(3\times1)}$
$\Sigma_t = (\bar{\Sigma}_t - K_t S_t K_t^{\ T}\ )_{(3\times3)}$
$\mu_t = \bar{\mu}_t$
$\Sigma_t = \bar{\Sigma}_t$
***return*** $\mu_t, \Sigma_t$
```

Table 3.6. Complete Pseudo code of UKF Localization using laser range finder measurement

4. Experiment and Analysis of Proposed Approach

This chapter presents the details of robot navigation experiment and performance evaluation of proposed approach in UKF localization algorithm. The navigation experiment is performed to collect the robot proprioceptive information and sensor measurement information. The experiment data set is utilized to implement UKF localization algorithm. Moreover, the performance of UKF localization algorithm is evaluated by considering the effect of control parameters values, and the comparatively analysis of UKF design parameter α values.

4.1 Mobile Robot Platform

4.1.1 Complete Robot Hardware

The complete robot hardware is divided into three main blocks: the control system, the robot, and the sensor system. Control system consists of a laptop computer that connected to robot via Ethernet. The mobile robot used for experiment is “MRP-NRLAB02” differential drive robot (shown in Figure 4.1) manufactured by REDONE Technologies. The sensor system consists of USAT A105 ultrasonic sensor system from Korea LPS Company, and LMS-511 laser range finder system of SICK. USAT A105 receiver sensor is connected to laptop computer via serial to USB connector. LMS-511 is connected to laptop via USB 2.0 to fast Ethernet adaptor. Figure 4.2 displays the connectivity diagram of the complete robot hardware for experiment.

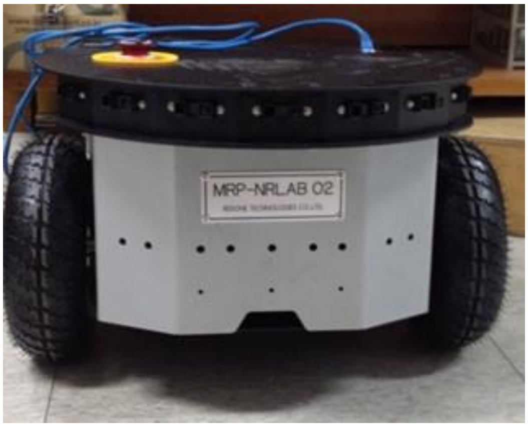


Figure 4.1. MRP-NRLAB02 Differential drive robot

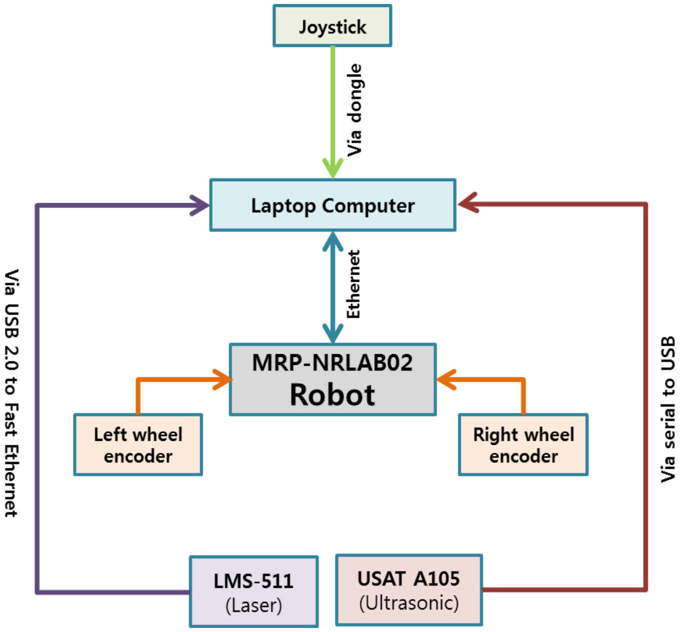


Figure 4.2. Connectivity diagram of MRP-NRLAB02 Hardware

The robot is controlled by a joystick. The robot's wheel encoder calculates the proprioceptive information consisting of translational velocity v and rotational velocity ω . The range information from ultrasonic beacons to robot receives from ultrasonic receiver. The distance from walls to robot is measured by laser range finder. Figure 4.3 shows the complete view of robot hardware used in experiment.

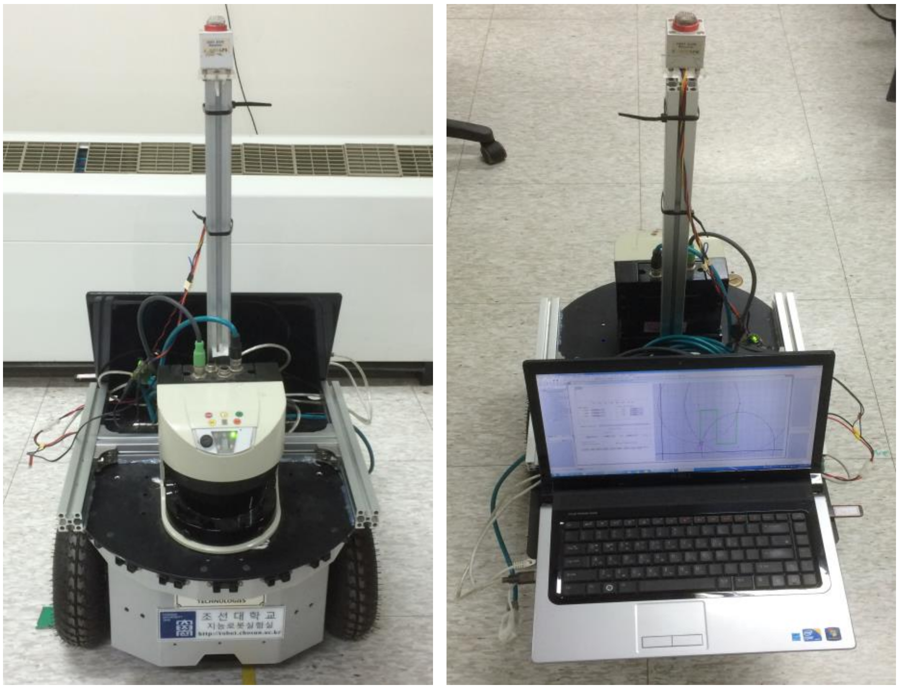


Figure 4.3. Front and rear view of complete robot hardware

4.1.2 Experiment Workspace

The experiment is conducted in a Capstone design laboratory, IT building 6th floor, Chosun University. The total work area for navigation is 15m×8m. The experiment work space consists of chairs, tables on which desktop computers are located, and ultrasonic beacons are attached on the ceiling of the room. The robot navigates through the defined way points on the floor. The initial position and the goal position of the mobile robot are set as (5.3m, 1.21m, π rad) and (10.26m, 6.16m, π rad). Figure 4.4 shows the actual environment of the workspace. The green line shows the nominal path of real robot.

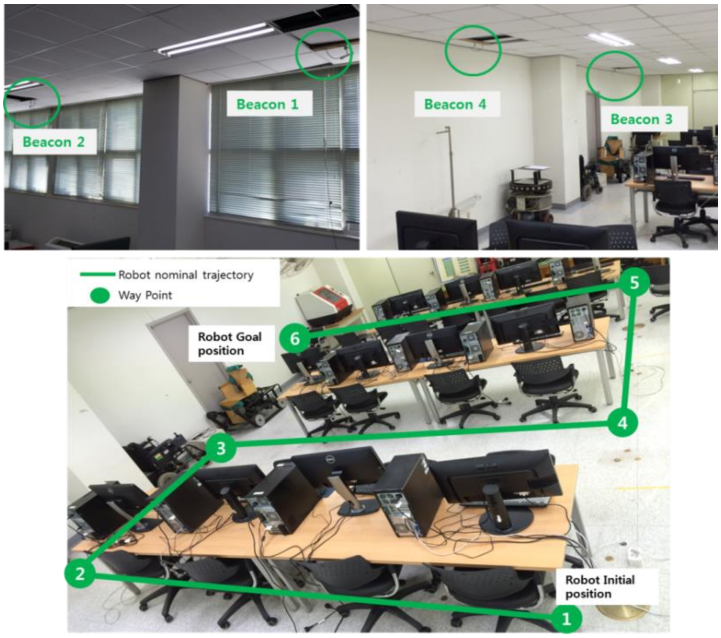


Figure 4.4. Robot Navigation work space (Capstone Design Laboratory)

Beacon	*x* position(m)	*y* position (m)
1	5.06	0.6
2	10.6	0.6
3	10.6	6.94
4	5.06	6.94

Table 4.1. Position of beacons

Due to the cluttered structure of work space environment some artificial arrangements are made in room. Because of the laser range finder which sometimes fail to detect the narrow legs of chair and tables. Therefore we have made some changes in the work space environment by creating the boundaries with the wooden blocks under each table. Figure 4.6 shows the artificial arrangement of workspace with nominal robot path and guided way points. Table 4.2 shows the position of way points specified on the robot’s nominal path.

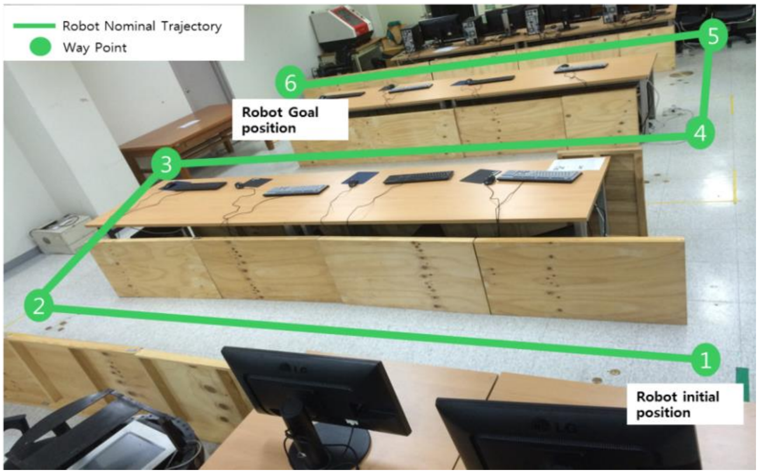


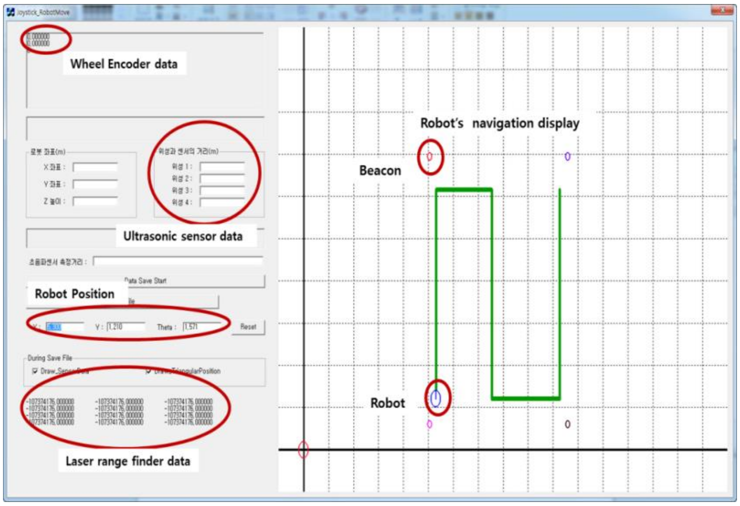
Figure 4.5. Workspace environment with artificial arrangement

Way Points	*x* position(m)	*y* position(m)
1	5.3	1.21
2	5.3	6.16
3	7.55	6.16
4	7.55	1.21
5	10.26	1.21
6	10.26	6.16

Table 4.2. Location of way points

4.1.3 Navigation Experiment

The robot navigation experiment is performed to collect wheel encoder data, range data from ultrasonic sensor system and laser range finder. We develop GUI which allows us to visualize the sensor data and wheel encoder data simultaneously during robot navigation. Figure 4.5 shows the snapshot of GUI display.



```
0.000000
0.000000
```

Y : 1.210   Theta : 1.571

```
-107374176.000000   -107374176.000000   -107374176.000000
-107374176.000000   -107374176.000000   -107374176.000000
-107374176.000000   -107374176.000000   -107374176.000000
-107374176.000000   -107374176.000000   -107374176.000000
```

Figure 4.6. Robot navigation display GUI

During the navigation experiment there is a time difference observed on receiving ultrasonic sensor, and laser range finder data, because of their different time of flights (TOFs). The USAT A105 ultrasonic sensor system we have used in our experiment has TOF values of 0.5 secs to propagate ultrasonic signal from beacon to receiver. On contrary LMS-511 laser range finder has TOF of 20 milli secs to measure the distance from wall to robot. The complete data set received from the navigation experiment consisting of wheel encoder data, ultrasonic sensor data, and laser range data is then saved into a text file. The text file is then use to implement the UKF localization algorithm in a simulated environment using MATLAB.

4.2. Implementation of UKF Localization Algorithm

The proposed approach used in UKF localization algorithm is to estimates the position of the robot in case of partially unknown environment. The experiment work space is modeled by grid map while the ultrasonic beacons are considered as landmarks, and the correspondence of range data to ultrasonic beacons is unknown while the robot location is estimated.

It is noted from the navigation experiment data set that ultrasonic sensor and laser range finder data does not receive at the same time. In this case we use separate UKF algorithm for each sensor measurement. The UKF algorithm using ultrasonic sensor measurements also computes the correspondence which associates the range data to a beacon. On the other hand it can happen very rarely that either of the sensor data can be received at the same time, or we can regard them roughly at the same time. This situation depends upon the algorithm computation time or it may happen due to the negligible TOF difference of each sensor’s data availability. Therefore, in this situation both of the sensor data are fused and we use UKF algorithm using the fusion of both the measurements. Table 4.3 illustrates the 3 separate instances to access any of these situations using UKF algorithms (discussed in Chapter 3).

```
Algorithm: Accessing Sensor data in UKF (Ultrasonic sensor data, Laser range finder data)
--------------------------------------------------------------------------------
if  Ultrasonic sensor data available
    execute Algorithm:UKF Localization using Ultrasonic sensor $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$
elseif  Laser range finder data available
    execute Algorithm : UKF Localization using Laser Range Finder $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$
elseif  Both Ultrasonic sensor and Laser range finder data available
    execute Algorithm:UKF Localization using Ultrasonic and Laser Range Finder $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t, m)$
endif
```

Table 4.3. Pseudo code for accessing UKF algorithms

4.2.1 UKF Localization Result

In this thesis we used the first two instances of Table 4.3. The data sets that we collected from the navigation experiment have ultrasonic sensor and laser range finder data which are not received at the same time. Altogether we have 23 measurement values and among them 4 are ultrasonic sensor data and 19 are laser range finder data. For the implementation of UKF localization algorithm we have setup the simulation on MATLAB. Figure 4.6 shows the Simulation setup for UKF localization.

In Figure 4.7 the light blue line shows the robot’s nominal path. The four small colored (red, green, blue, and pink) circles denoted by labels B1, B2, B3, and B4 are ultrasonic beacons. The numbers shown on the nominal path indicate the way points. The arcs centered at the beacons indicate the range from the corresponding beacons. The laser range finder data represented by the blue lines emerging from estimated robot shown by small red circle. The black ellipse around the estimated robot position indicates the robot pose error covariance or robot pose uncertainty. The black connected dots display the map information of the experiment work space. The variable ‘i’ shows the number of iterations. The correct correspondence and the calculated correspondence of ultrasonic beacons denoted as “ct” and “calc c”. The ‘tv’ and ‘rv’ variable displays the robot translational and rotational velocity calculated from its wheel encoders.

To check the estimation of robot location given by the UKF localization, the estimated position of the robot followed by the robot’s nominal trajectory. Because in real situation we don’t know the actual location of robot, in this case we compare the estimated robot trajectory with robot’s nominal trajectory.

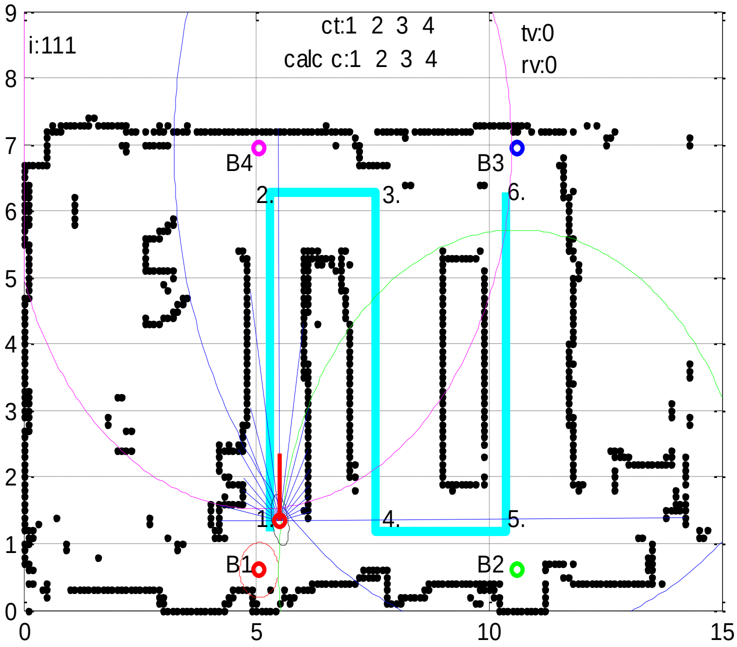


Figure 4.7. Simulation setup for UKF Localization

Table 4.4 the values motion noise and measurement noise used for simulation. Figure 4.5 and shows the simulated results of UKF estimation for robot localization.

Motion Noise				Measurement Noise	
α_1	α_2	α_3	α_4	σ_{usat}	σ_{LRF}
0.0012	0.050001	0.0011	0.13	0.5	0.1

Table 4.4. Uncertainty parameter values used for simulation

For displaying robot pose uncertainty in simulation, the uncertainty ellipse is magnified by the factor of 10 to make it easier to examine. Figure 4.8(a) shows the estimated robot at its initial position, it can explicitly observed through the pose uncertainty ellipse that robot has high error covariance also the sigma points quiet far spread around the estimated robot mean position. However it can be seen that estimated robot allocate correct correspondence value to beacons. In

Figure 4.8(b) the robot progresses towards way point 2. The pose uncertainty with respect to robot x pose coordinate is considerably small than y pose coordinate, also the sigma points gets condensed around the estimated robot mean position. Moreover, the robot correctly specify the correspondence value to beacons.

Figure 4.8(c) displays the estimated robot at way point 2. At this position it can be clearly through uncertianty ellipse that robot pose uncertainty is large. Due to large pose uncertainty, estimated robot allocates incorrect hypothese for the correspondence to beacon 1. Another reason of this situation is the range data received from beacon 1 and 3, which is same, in this case robot selects the measurement likelihood for beacon 1 and 3 is same. Figure 4.8(d) depicts the estimation with high ambiguity. Although the robot pose uncertainty seems to be very small still the hypothesis for the correspondence to beacons asserts false data association. Through intuitive obervation of this situation the distance from beacons 1 and 2 to robot is same and the distance from beacon 3 and 4 is same, However, the range data from each beacon which is represented by their respective arcs are measured to be same. Therefore this situation enable the robot to determines the same correspondence values to each beacon.

Figure 4.8(e) indicate the robot postion at way point 4. This figure depicts the two situations, first the robot has high pose uncertainty. Secondly hypothesis for correspondence to beaons is incorrect. It can be seen that robot is diagonally equidistant from beacons 3 and 4 therefore, robot deterimine the most likely correspondence value is 3. On the other hand the location of the robot not equally distant to beacons 1 and 2, but allocate incorrect correspondence value to them. This situation can be seen in two ways first at way point 4 robot has high pose uncertainty and secondly the range measured from beacons 1 and 2 has high measurement noise.

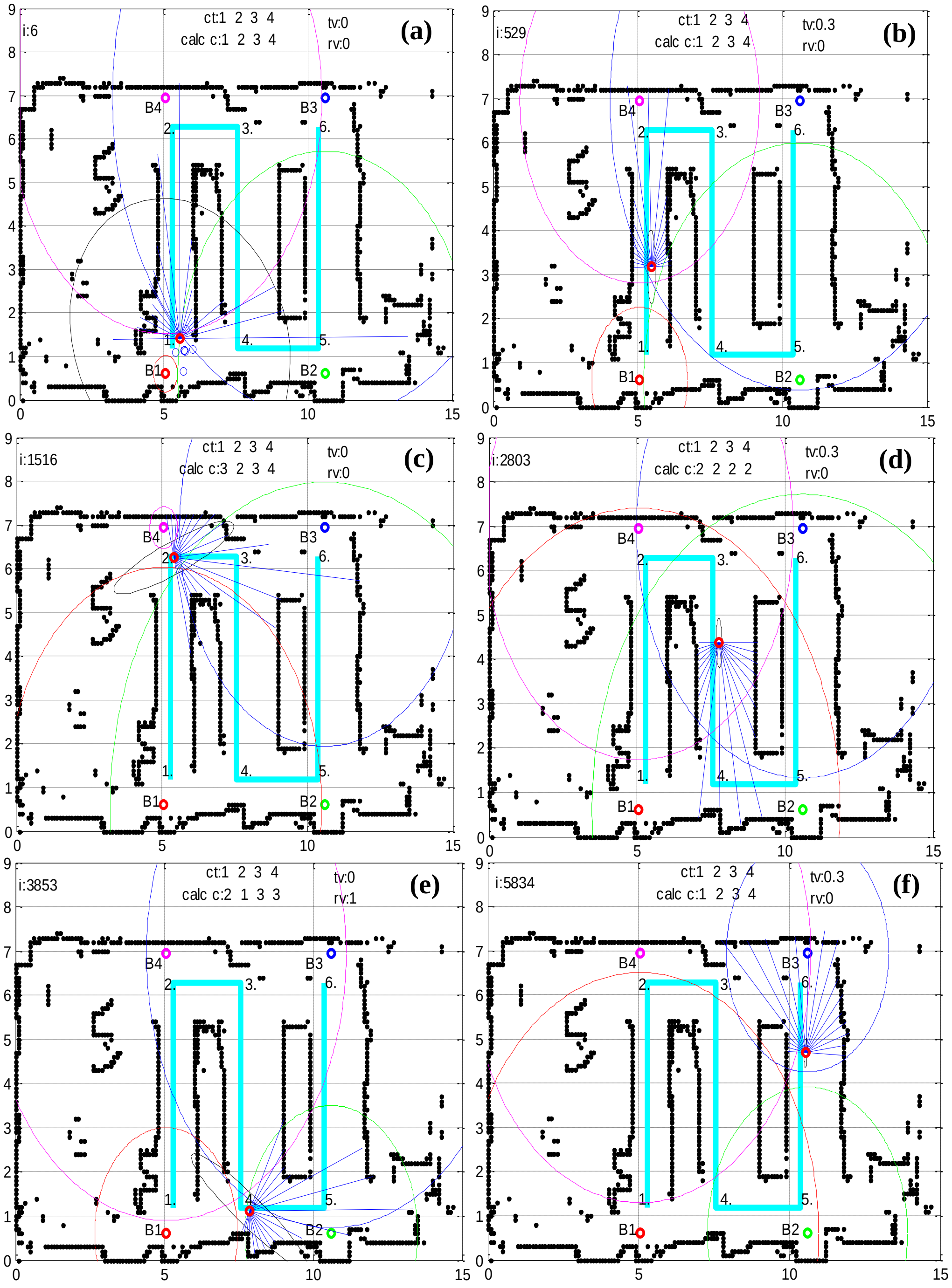


Figure 4.8 UKF estimation results for robot localization

4.2.1.1 Discussion

The implications made for deriving the simulation results to process the proposed approach in UKF localization are: (a) The robot has partial information about its work space i.e robot does not know the actual correspondence of data association for beacons. However correspondence of associating range data to beacon is known actually. (b) The robot actual pose is unknown and the estimation of robot postion is according to the nominal path of real robot. The simulation results regards the two aspects of estimation. First the robot pose uncertainty during estimation and second the hypothesis of the correspondence to beacons.

As on the basis of first implication the robot has partial information about its work space so it can be explicitly observed in the Figure 4.8(a) the robot pose uncertainty ellipse is large and also the sigma point located far apart from the mean position of the robot. However, it is observed from the other figures of Figure 4.8 the sigma points postion get condensed at the estimated robot mean position throughout the simulation. The large robot pose uncertainty is also observed in Figure 4.8 (c) and (e) suggest that the motion and measurement errors are relatively high. However, it is also observed that pose uncertainty decreases when the estimated robot moves straight along the nominal path. The cause of the incorrect data association to beacons has two ramifications: It is due to the high robot pose error covariance. Another ambiguity occurs if there are many likely hypothesis for the correspondence (i.e the similar range data measured from the beacons). To reduce the danger of asserting the false data association, there exist essentially two techniques: First, select the beacons that are sufficiently apart from each other so it doest not that confuse them with each other while they have same range information. Second, make sure that the robot pose uncertainty remains small. Unfortunately, these two strategies are somewhat counter to each other, and finding the right granularity of landmarks in the environment can be regarded somewhat as an art [21].

4.2.1.2 Performance Evaluation

To evaluate the performance of proposed UKF localization method we have performed the analysis according to two aspects. First by considering the effect of motion and measurement noise on localization and correspondence calculation. Secondly, the comparative analysis of UKF “α” parameter for localization and correspondence calculation.

4.2.1.2.1 Effect of Control Parameters

The control parameters considered as robot motion noise and sensor measurement noise in mobile robot localization. In actual case the real control parameter values are unknown to us. Therefore, in this section the simulated values of motion and measurement noise are used to evaluate the perfromance of proposed UKF localization method. We have performed analysis using 4 different sets of motion noise values and 2 sets of measurement noise values to investigate the behaviour of proposed method. We have assigned one set of measurement noise values for two of four sets of motion noise values. Table 4.5 shows six different set for the control parameter values used in simulation.

Motion Noise					Measurement Noise	
Case	α_1	α_2	α_3	α_4	σ_{usat}	σ_{LRF}
1	0.001001	0.033001	0.001001	0.10001	0.5	0.1
2	0.001003	0.051001	0.001201	0.15001		
3	0.001002	0.050001	0.001100	0.13001	0.25	0.09
4	0.001003	0.053001	0.001101	0.12001		

Table 4.5. Simulated control parameters values used for estimation (for each case the UKF parameter α set to 0.7.

The analysis result is evaluated by plotting the estimated trajectory, distance error between nominal robot postion, and estimated robot position for each case and also computing the rate of correct correspondence caluculation for each case. Figure 4.9 and 4.10 shows the estimated trajectories and distance error. Table 4.6 and 4.7 listed mean, standard deviation and root mean square error (RMSE) for estimation, and the rate of correct correspondence for beacons.

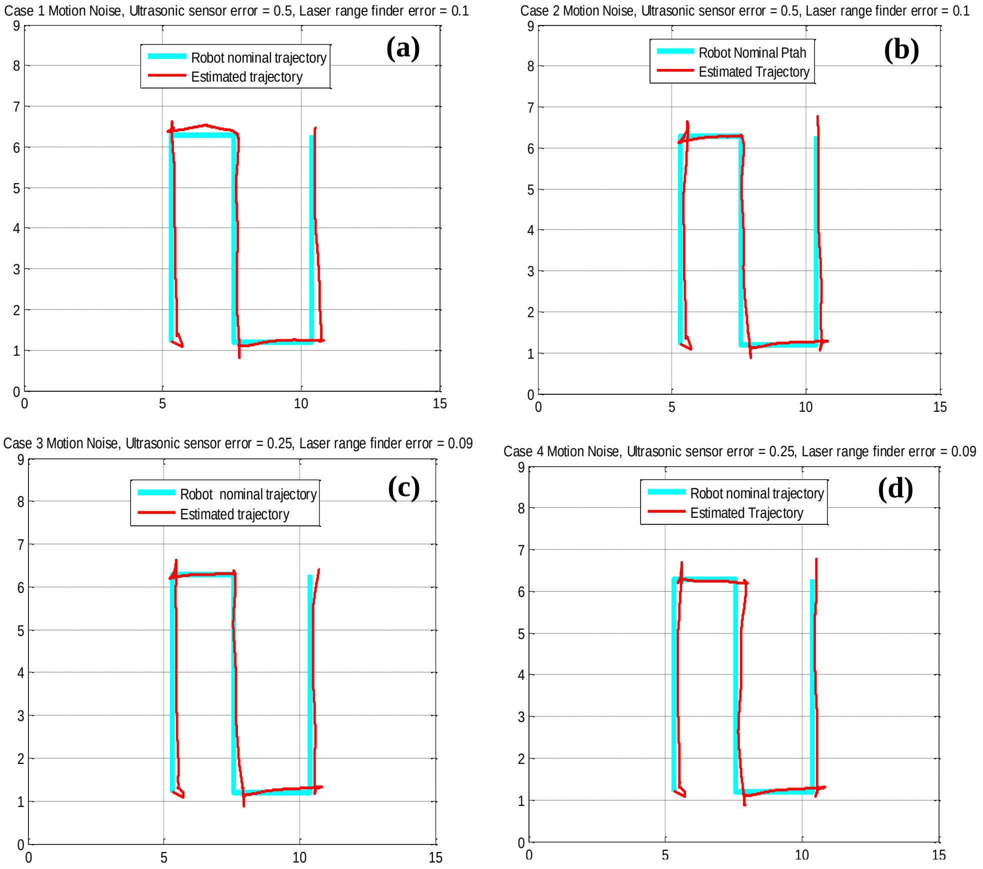


Figure 4.9. Estimated trajectories according to control parameter values of Table 4.5

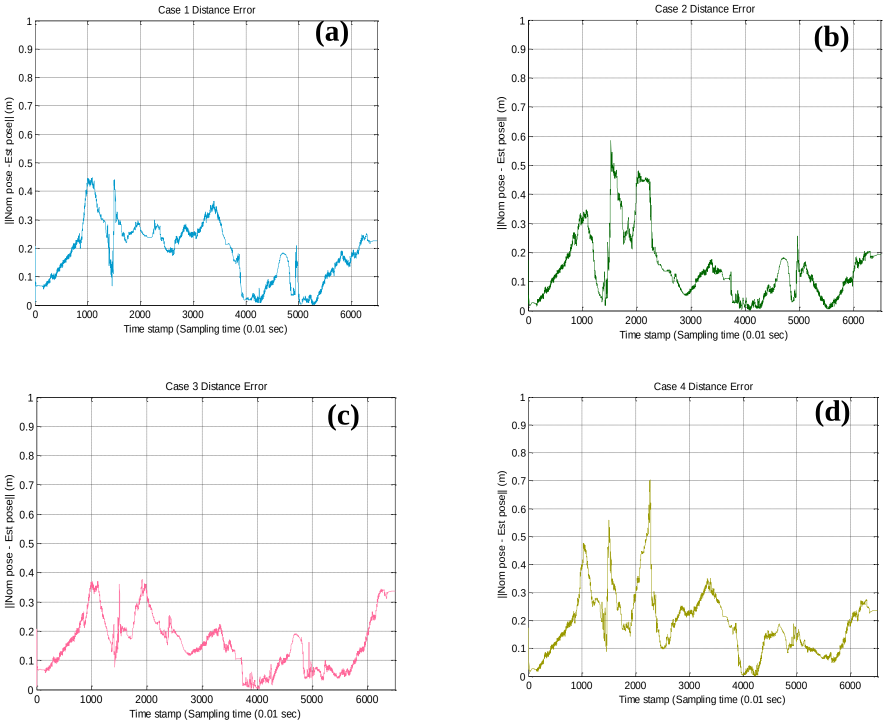


Figure 4.10. Distance error according to control parameter values of Table 4.5

Case	Mean	Standard deviation	RMSE
1	0.1991	0.1047	0.2249
2	0.1547	0.0927	0.2034
3	0.1458	0.1150	0.1857
4	0.2104	0.1546	0.2611

Table 4.6. Mean, standard deviation and root mean sqaure values of distance error according Table 4.5

Case	Rate of calc. correspondence of each Beacon				Over all correct correspondence rate
	B1	**B2**	**B3**	**B4**	
1	70.36%	74.36%	73.25%	76.62%	68.78%
2	74.06%	76.50%	80.43%	79.01%	73.18%
3	72.87%	77.10%	75.21%	78.30%	70.41%
4	66.03%	73.28%	67.72%	74.09%	65.40%

Table 4.7. Rate of correct correspondence caluation for each beacon according to control parameter values of Table 4.6

4.2.1.2.2 Comparative analysis of UKF “α” parameter values

The UKF “α” parameter is considered as a design parameter which determine how far the sigma point spread around the mean $\bar{\mu}_t$. Its values can vary between 10-4 and 1. In this section the comparative analysis of “α” is performed to evaluate the estimation performance of proposed appraoch to process UKF localization. We have performed series of comparative analysis using different alpha parameter values. In this thesis we have include the result of 4 suitable alpha values among them. Because the estimation results from these alpha values found to be more feasible as compared to other values of alpha. The comparative analysis of alpha parameter is performed with fixed values of control parameter values. Table 4.8 shows the alpha parameter values used for the estimation UKF localization. The analysis result is evaluated by plotting the estimated trajectory, distance error between nominal robot, and estimated robot trajectory for each case. Also computing the rate of correct correspondence caluculation for each case. Figure 4.11 and 4.12 shows the estimated trajectories and distance error. Table 4.9 and 4.10 list mean, standard deviation and root mean square error (RMSE) for estimation, and the rate of correct correspondence for beacons.

Case	1	2	3	4
α	0.2	0.3	0.45	0.55

Table 4.8. UKF **α** parameter values used for the estimation (for each case control parameter values are set to α_1=0.001002, α_2=0.05001, α_3=0.001100, α_4=0.13001, and σ_{usat}=0.5, σ_{LRF}=0.1)

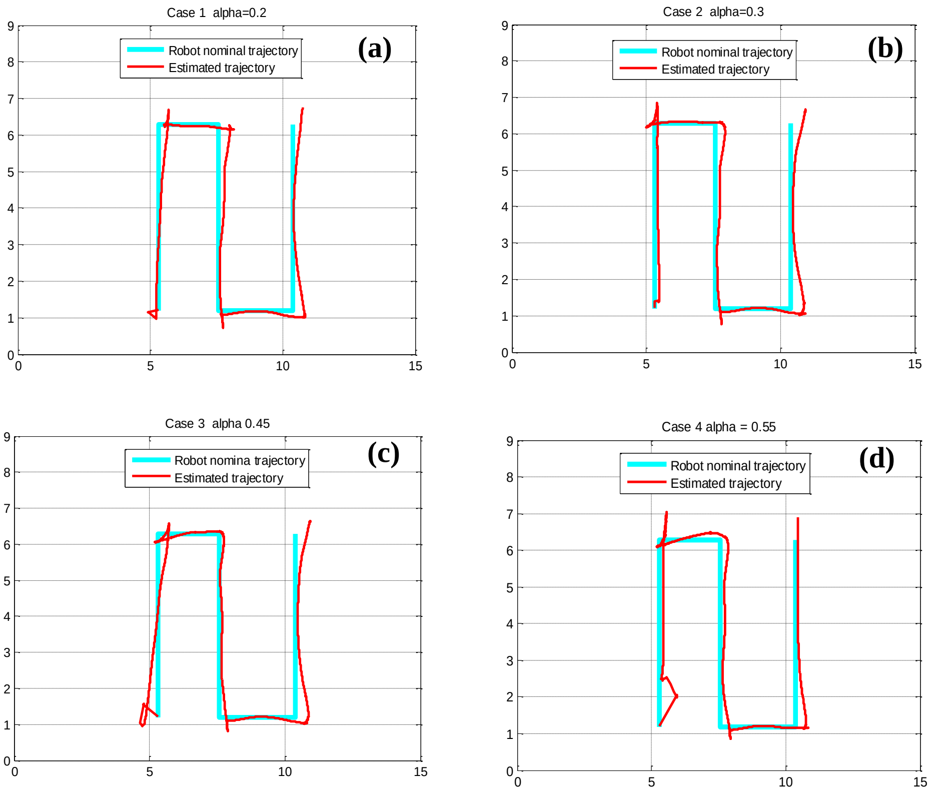


Figure 4.11. Estimated trajectories according UKF **α** parameter values

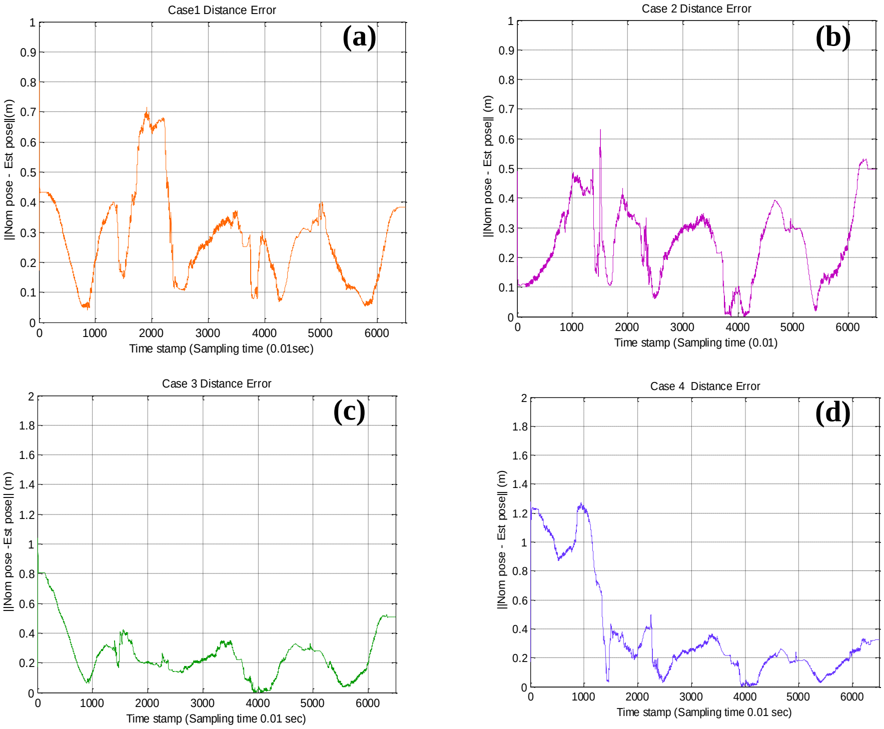


Figure 4.12. Distance error according to UKF **α** parameter values of Table 4.8

Cas	Mean	Standard	RMSE
1	0.2776	0.1532	0.3171
2	0.2492	0.1300	0.2810
3	0.2581	0.1718	0.3100
4	0.3767	0.3567	0.5188

Table 4.9. Mean, standard deviation and root mean sqaure values of distance error according UKF α of Table 4.8

Case	Rate of calc. correspondence of each Beacon				Over all correct correspondence rate
	B1	**B2**	**B3**	**B4**	
1	64.37%	67.23%	68.12%	70.03%	63.03%
2	66.85%	67.18%	70.43%	71.65%	65.10%
3	56.21%	66.56%	67.45%	71.56%	55.46%
4	59.23%	59.48%	70.52%	68.09%	57.40%

Table 4.10. Rate of correct correspondence caluation for each beacon accrding to UKF α parameter values of Table 4.8

4.2.1.2.3 Discussion

The pupose of performance evaluation is to analyse the estimation behavior of proposed approach for robot localization. The results which considers effect of control parameter values infer from the estimated trajectories, statisitical analysis and the correct correspondence rate. The value of UKF α parameter kept as 0.7 throughout while considering the effect of control parameters. Estimated trajectoy result from (b) and (c) of Figure 4.9 represents significantly better estimation results and correct correspondence rate as compared to (a) and (d) of Figure 4.9. It is evident form the results that propoer selection of control parameter values is curcial for the use of propsed approach.

The comparative analysis of UKF α parameter values, deduce the results of esimtated trajectories, statistical analysis and the correct correspondence rate. It is evident from the results that proper selection of the values of UKF α parameter has primary importance for the implementation of proposed approach. After considering the results of performace evaluation it is concluded that UKF parameter α =0.7, and the set of control paramter values of case 2 of Table 4.6 are found to be optimal values for our simulation results.

However, the simulated control parameter values, and the UKF α parameter values are not fixed to implement the UKF localization algorithm used in proposed appraoch. Because we use these values in simulation. These values may vary depending upon the enviromnet of work space, robot’s propioceptive information, and the type of sensors we used. The values also depends upon the startegies adopt by the user to test the algorithm in simulation.

4.3 Comparitive analysis of UKF Localization

In this section we have performed the comparitive analysis for UKF localization without sensor fusion and with sensor fusion (i.e. the proposed approach). In case of without sensor fusion we implement UKF localization algorithm using only ultrasonic sensor measurement, and UKF localization algorithm using only laser range finder individually. The implementation of comparative analysis is done through using the data set collected from the navigation experiement. Figure 4.13 depicts the comparison result of UKF loclaization for each measurement case.

4.3.1 Statistical analysis and correspondence calculation rate

The statistical analysis is performed to evaluate the perfromance of UKF localization result without sensor fusion and with sensor fuison. The analysis distance error between nominal robot, and estimated robot trajectory for each case. Also computing the rate of correct correspondence caluculation for each case. Figure 4.13 and 4.14 shows the estimated trajectories and distance error. Table 4.11 and 4.12 listed mean, standard deviation and root mean square error (RMSE) for estimation, and the rate of correct correspondence for beacons.

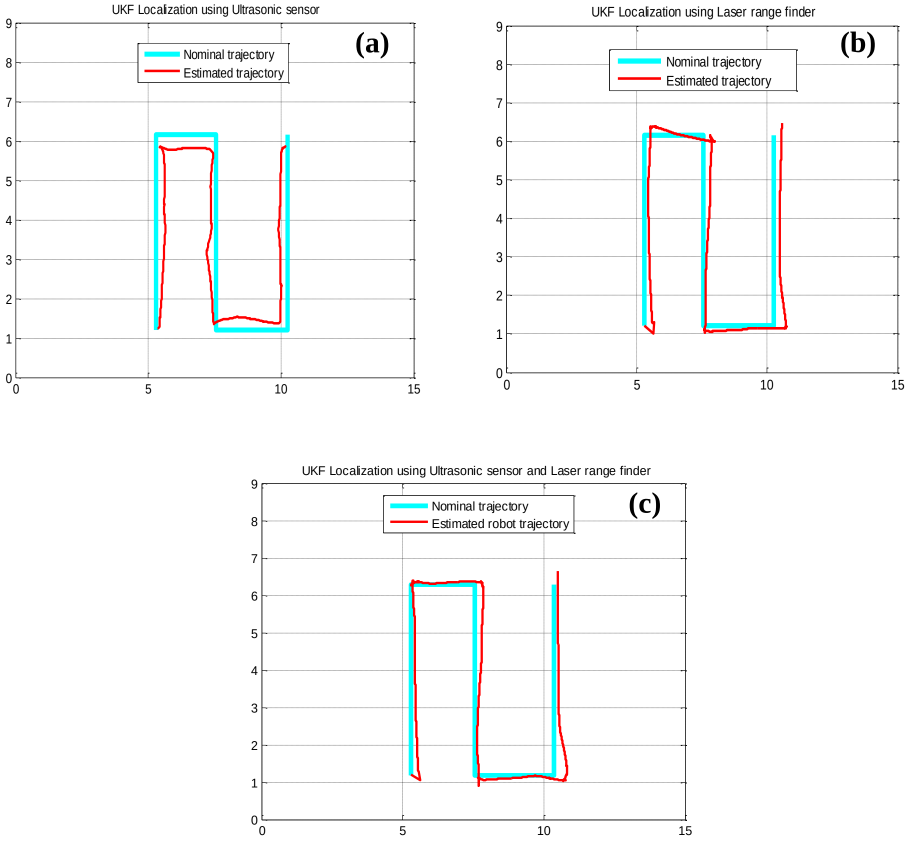


Figure 4.13. Estimated trajectories (a) UKF localization using ultrasonic sensor measurement (b) UKF localization using laser range finder measurement (c) UKF localization using fusion of ultrasonic sensor and laser range finder

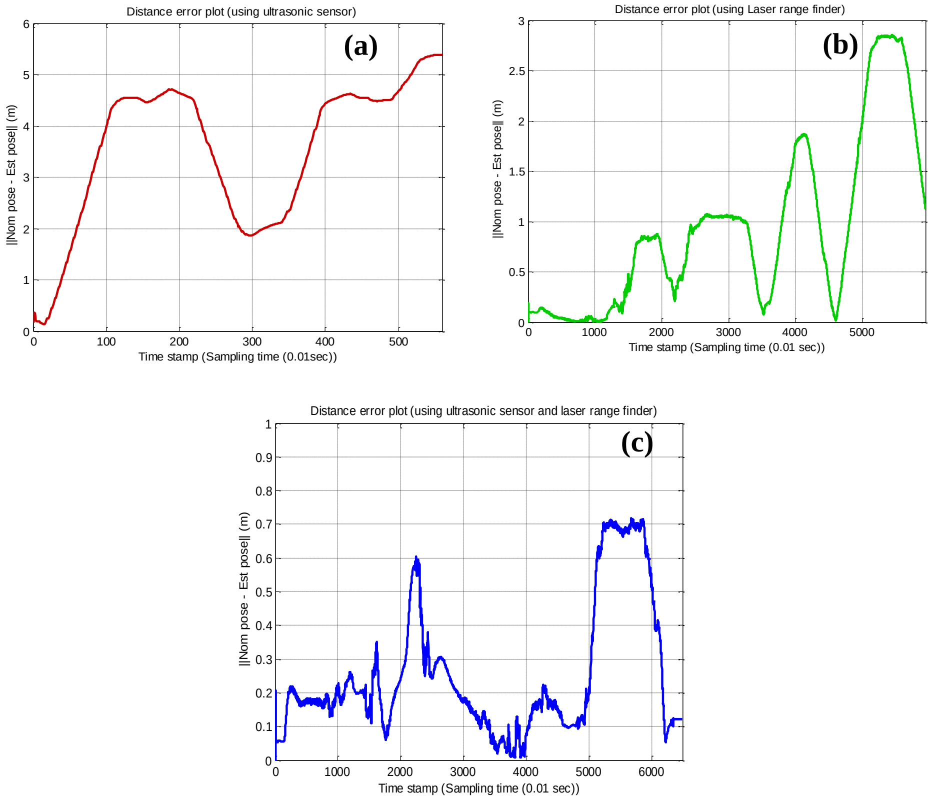


Figure 4.14. Distance error according according to comparative analysis

UKF Localization	Mean	Std dev.	RMSE
using ultrasonic sensor	3.5476	1.4028	3.8144
using laser range finder	0.9905	0.8254	1.2853
using fusion of ultrasonic sensor and laser range finder	0.278	0.2384	0.3630

Table 4.11. Mean, standard deviation and root mean sqaure values of distance error according to comparative analysis

UKF Localization	using ultrasonic sensor	using fusion of ultrasonic sensor and laser range finder
B1	66.85%	77.06%
B2	67.18%	78.50%
B3	70.43%	80.43%
B4	71.65%	79.01%
Over all correct correspondence rate	65.01%	76.18%

Table 4.12. Rate of correct correspondence caluation for each beacon according comparative analysis

4.3.2 Discussion

The comparative analysis is performed to investigate the estimation performance of UKF localization between without sensor fusion case and with sensof fusion (i.e. the proposed approach). Estimated trajectory of Figure 4.13(c) represents the improved estimation result from the proposed approach as compared to Figures 4.5(b) and Figure 4.5(c) which shows UKF localization using ultrasonic sensor measurement only, and UKF localization using laser range finder only. It is evident from the distance error plots, statistical evaluation, and correspondence rate calculation the propsed approach shows improved estimation results as comapred to UKF loclaization results using ultrasonic sensor measurement and UKF localization results using laser range finder measurement.

5. Conclusion

The research done for this thesis describes the method for mobile robot localization in partially unknown environment. We proposed the appraoch using UKF localization algorithm with fusion of ultrasonic sensor and laser range finder measurements. The proposed approach also deals with the data asocation problem in a partially unknown environement.

To implement the proposed approach we first perform the robot naivgation experiement in an indoor environment. The purpose for the navigation experiement is to collect the robot proprioceptive information, range data received from beacon to robot by ultrasonic receiver and the distance information from walls to robot by laser range finder. Form the navigation data set, it is noted that at every time stamp the robot receive measurement information from either of the sensor. It is also impractical to receive both of the sensor measurment simultanously due to their time of flight (TOFs) differences.

According to the experiment data set organization we use separate UKF loclaization algorithms for each sensor measurements. The UKF localization algorithm which uses ultrasonic sensor measurements also calculates the correspondence which associates range data to a beacon. We also considered the situation that if both of the sensor measurements received at the same time, then we use UKF algorithm for fusing both of the sensor measurements. All of these UKF algorithms are discussed in chapter 3. Table 4.3 shows how the UKF localization alogoritthms are accessed on the availabity of respective sensor data.

We implement our proposed approach on simulation using the data set collected from the navigation experiement. The simulation results show some brittle behavior of estimation perfomance. At some instant during simulation the robot pose has high position error covaraince which computes incorrect correspondence of associating data to ultrasonic beacons. At some instants of simulation the robot receive same range information form all the beacons which also results incorrect data association results. Although robot location is not equidistant from all the beacons.

The discrepancies in the results can be improved by some practical considerations i.e. the robot first selects those landmarks which are sufficiently apart from each other, this strategy reduce the confusion of likely correspondence hyothesis for them. Secondly the robot pose uncertainty should remain small. Unfortunately both of these considerations somewhat counter each other while the robot position is estimated. On the other hand in context with simulation results we faced some work space contraints. More generally speaking the positions of ultrasonic beacons are fixed in the workspace and the sequence of range data generated from the ultrasonic sensor sytem is also fixed.

Moreover the performance of propsed approach is analysed by considering the effect of control parameters, and by comparative analysis of UKF design parameter α values. From the performance evaluation it is explicitly shown that the appropriate selection of control parameter values, and UKF α parameter values are crucial for implementing the proposed approach.

We also perfromed the comparative analysis between the UKF localization algorithm using ultrasonic sensor measurement only, UKF localization algorithm using laser range finder only, and the propsed apporach which contributes the fusion of ultrasonic sensor and laser range finder measurments. The statistical evaluation, and the correspondence calculation rate shows that the proposed approach shows improved estimation as compared to UKF localization estimation without fusion cases.

The maximum likelihood estimation approach we used to cope with unknown correspondence issue for the ultrasonic beacons. However we have found that it may not be work efficiently as it could be. From the simulation results it is showed that MLE was able to correctly assocaite the arange data to beacons in all but at few instants. This may due to the experiment poor work space modeling which causes incorrect cooorespondence results. It can be improved by throught more thoughtful selection of landmarks in line 1 of algorithm in Table 3.3. Nevertheless the MLE teachnique is of great practical importance. The hypothesis framework use in MLE provides the perfect platform for applying more various and advanced techniques such as multi-hypothesis tracking (MHT) [36].

Throughout the progress of this research interesting ideas for future work came up. In this reaserch we discuss the implementation and performance of proposed approach for robot loclaization by considering fixed ultrasonic beacon as static landmark assuming the correspondence is unknown to robot. One of these is to look for ways to make this proposed approach applicable in dynamic environments, when the number of landmarks are not fixed and static.

Bibliography

[1] http://forums.trossenrobotics.com/tutorial/introduction-129/an -intorduction-to-mapping-and localization-3274

[2] S. Naveed, N. Y. Ko, "Analysis of indoor mobile robot localization using ultrasonic sensors," International Journal of Fuzzy and Intelligent Systems, Vol. 14, No. 1, pp. 41-48, 2014.

[3] F. Duchon, M. Dekan, L. Jurisica, and A. Vitko, “Some application of laser range finders in mobile robotics,” Journal of Control Engineering and Applied Informatics, Vol. 14, No. 2, pp. 50-57, 2012.

[4] D. Brscic, and H. Hashimoto, “Model based robot localization using on board and distributed laser range finders,” In Proceedings of the IEEE International Conference on Intelligent and Robot and Systems, pp. 1154-1159, 2008.

[5] B. Gary, and G. Welch. "An introduction to the Kalman filter." Proceeding of SIGGRAPH, Course 8 (2001): 27599-3175.

[6] O. E. R. Ponce, “Autonomous robots map-based EKF localization,” Universitat de Girona, 2010.

[7] D. Fred, "Nonlinear filters: beyond the Kalman filter," Aerospace and Electronic Systems Magazine, IEEE , Vol. 20, No. 8, pp. 57-69, 2005.

[8] E. A. Wan, and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation." Adaptive Systems for Signal Processing, Communications, and Control Symposium, pp. 153-158, 2000.

[9] C. Suliman, and F. Moldoveanu "Unscented Kalman filter position estimation for an autonomous mobile robot," Bulletin of Transilvania University of Brasov, Vol. 3, No. 52, 2010.

[10] S. Thrun, "Particle filters in robotics." Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, pp. 511-518 2002.

[11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots. Artificial intelligence," Vol. 128, No. 1, pp. 99-141, 2006.

[12] G. Cotugno, L. D'Alfonso, W. Lucia, P. Muraca, and P. Pugliese, "Extended and Unscented Kalman Filters for mobile robot localization and environment reconstruction," In IEEE 21st Mediterranean Conference on Control & Automation, pp. 19-26, 2013.

[13] L. Romero, E. Morales, and E. Sucar, "Building maps using indoor mobile robot with ultrasonic and laser range sensor," Computaciaon Sistemas, 2002.

[14] B. S. Choi, and J. J. Lee, " Mobile robot localization in indoor environment using RFID and sonar fusion system," in Intelligent Robots and Systems, IEEE/RSJ International Conference on pp. 2039-2044, 2009.

[15] B. S. Choi, J. W. Lee, and, J. J. Lee, " A hierarchal algorithm for indoor mobile robot localization using RFID sensor fusion ," In transactions on Industrial Electronics, Vol. 58, No. 6, pp. 2226-2235, 2011.

[16] A. Stroupe, M. Martin and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in Proceedings, of the IEEE Conference on Robotics and Automation, pp. 1092-1098, 2001.

[17] S. Thrun, D. Fox, W. Burgard, and F. Dellart, “Robust Monte Carlo localization for mobile robots,” Artificial intelligence, Vol. 128, No. (1-2), pp. 99-141, 2001.

[18] W. Haijun, C. Yimin, "Sensor data fusion using rough set for mobile robots system," in Proceedings of the IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, pp. 1-5, 2006.

[19] D’ A. Luigi, A. Grano, P. Muraca, and P. Pugliese, "Sensor fusion and surrounding environment mapping for a robot using a mixed extended Kalman filter, in 10th IEEE International Conference on Control and Automation, pp. 1520-1525, 2013.

[20] G. G. Rigatos, "Extended Kalman and particle filtering for sensor fusion in motion control of mobile robots," Mathematics and Computers in Simulation, Vol. 81, No. 3, pp. 590-607, 2010.

[21] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, MIT Press, 2005

[22] P. Jensfelt, “Approaches to mobile robot localization in indoor environments,” Doctor Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2001.

[23] V. Varveropoulus, “Robot localization and map construction using sonar data,” The Rossum Project, available at http:// roussum.sourceforge.net/accessed on April 2014.

[24] S. Feng, C. Wu, Y. Zhang, and Z. Jia, “Grid based improved maximum likelihood estimation for dynamic localization of mobile robots,” Open access article available at http://dx.doi.org/10.1155/2014/271547.

[25] R. Negenborn, “Robot localization and Kalman filters,” M.S. thesis, Institute of Information and Computing Sciences, Copenhagen, Denmark, 2003.

[26] S. Se, D. L. J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” International Journal Robot Res., Vol. 21, No. 8, pp. 735-758, 2002.

[27] P. Corke, “Robotic Vision and Control, Fundamental algorithms in MATLAB,” Springer Tracts in Advanced Robotics, Vol. 73, 2011.

[28] S. Kucuk, and Z. Bingul, “Robot Kinematics: Forward and Inverse Kinematics,” in Industrial Robotics: Theory Modeling and Control, ch. 4 pp. 117-148, 2006.

[29] S. Jeong, "Development of function Library for mobile robot localization," M.S thesis, Chosun University, Gwangju, South Korea, 2013.

[30] S. Naveed, N. Y. Ko, “Analysis of MCL for indoor mobile robot localization using ultrasonic sensors,” in Proceeding of International Conference on Control, Automation, Robotics, and Vision Engineering, pp. 72-76, 2014

[31] S. Suyoung, J. Choi, B. Kim, and M. Park, "Improved ultrasonic beacon system for indoor localization," in Proceedings of the 12th International Conference on Computer Applications in Ship Building (ICCAS 2005), 2005.

[32] A. Siddiqui, T. Hellstrom, “Laser-based localization of vehicles and robots in natural and unstructured environments,” Research Project, Umea University, Umea Sweden, 2005.

[33] C. Sauze, and M. Neal, “A raycast approach to collision avoidance in sailing robots,” in Proceedings of the 3rd International Robotic Sailing Conference, pp. 26-33, 2010.

[34] J. Manyika and H. Durrant-Whyte, Data Fusion and Sensor Management, a decentralized information-theoretic approach, Ellis Horwood Limited, Chichester, West Sussex 1995.

[35] I. J. Myung, “Tutorial on maximum likelihood estimation,” Journal of Mathematical Pshycology, Vol . 47, pp. 90-100, 2003.

[36] A. Amditis, G. Thomaidis, P. Maroudis, P. Lytrivis and G. Karaseitanidis “Multiple hypothesis tracking implementation,” in Laser Scanner Technology, pp. 199-220, 2012.