

# Underwater Vehicle Localisation using Extended Kalman Filter

Miroslav Radojević

Supervisor: Yvan Petillot

Ocean Systems Lab  
Heriot-Watt University, Edinburgh

Escola Politècnica Superior  
Universitat de Girona

Centre Universitaire Condorcet  
Université de Bourgogne

A Thesis Submitted for the Degree of  
MSc Erasmus Mundus in Vision and Robotics (VIBOT)

· 2011 ·

## Abstract

In order to accomplish various missions, autonomous underwater vehicles need to be capable of estimating their position within the environment. This is a prerequisite of a successful mission since further tasks that need to be achieved strongly rely on navigation information as a source of valuable information. Much has been said about navigation underwater, yet the research has not reached the level of having as equally precise solution for navigation as the one available above the water surface.

This thesis is a study on application of an algorithm that would accomplish the localisation of the Ocean Systems Lab's Nessie underwater vehicle using measurements from a number of sensors mounted on it. Well known Extended Kalman Filter (EKF) algorithm approach was suggested as a solution for robot self-localisation. Chosen method uses measurements from the set of sensors installed on the vehicle. Standard Kalman Filtering routine involves two stages: prediction and correction. Prediction model is derived from the mathematical model of the vehicle dynamics. Based on the laws of kinematics, prediction model calculates the next vehicle dynamic state. Five degrees of freedom (d.o.f.) *constant-speed* model for motion of the rigid body was proposed for this purpose. Measurements from different sensors are fused together to make the observations. Observations are responsible for the correction stage. In other words, observations have been combined together with the process model in order to make a quality estimate of the current location of the robot within the environment.

Prediction model applies the laws of physics in order to make a suitable role model of robot movement. Inspiration for the solution comes from previous works on EKF-based localisation. Practical application of the EKF implies management of the observations in terms of time and sensor type. Additional practical issue that was addressed in the thesis is the choice of heading sensor and quality of obtained heading as an important ingredient of the navigation. Implementation of Unscented Kalman Filter (UKF) was investigated as potential improvement in working with nonlinearities. Finally, the absolute position observations tend to be quite noisy, nevertheless very important navigation measurements. EKF was demonstrated as a tool for sensor fusion and simultaneous filtering of the position measurements.

In addition, the work gives a summary of the underwater localisation methods and capabilities. Theoretical background on nonlinear filtering was investigated in order to justify the reasons for choosing EKF. Finally, experiments with recorded sensor data and some real missions have been carried out. Their results have been presented as a part of navigation performance test and analysis. The emphasis is on the engineering of the solution, improving the navigation and finding the way to overcome known deficiencies, mostly due to nonlinearities, volatile heading measurement and measurement imprecisions. Thesis is intended to report pros and cons of a practical piece of work, implemented in C++ within the Robot Operating System

---

(ROS) framework and Nessie software and hardware platform - where a scientific concept was adopted to solve the real task.

*If one does not know to which port one is sailing, no wind is favourable.*

Seneca, Roman philosopher, mid-1st century AD

# Contents

<b>Acknowledgments</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Why underwater? . . . . .	1
1.2 Localization - Where the Robot Is? . . . . .	1
1.3 Autonomous underwater vehicle (AUV) . . . . .	2
1.4 Contribution of the thesis . . . . .	2
1.5 Thesis outline . . . . .	3
<b>2 Problem Definition</b>	<b>4</b>
2.1 Robot localization . . . . .	4
<b>3 Navigation capabilities of AUVs</b>	<b>7</b>
3.1 State estimation . . . . .	7
3.2 Stochastic state estimators . . . . .	8
3.2.1 Linear stochastic state estimators . . . . .	8
3.2.2 Nonlinear stochastic state estimators . . . . .	8
3.3 Kalman Filter (KF) . . . . .	9
3.4 Deterministic state estimators . . . . .	12
3.5 Strategy . . . . .	12
3.6 Acoustic-based localization techniques . . . . .	13

3.7	Terrain-aided navigation . . . . .	14
3.8	SLAM . . . . .	15
3.9	Cooperation for navigation . . . . .	16
3.10	Active localisation . . . . .	16
3.11	Related work on AUV navigation . . . . .	16
3.11.1	Kalman filtering localisation . . . . .	17
3.11.2	Monte Carlo-based filtering methods . . . . .	25
3.11.3	GPS aided localisation . . . . .	27
3.11.4	Localisation using <i>a-priori</i> map . . . . .	27
3.11.5	Optical sensing . . . . .	28
3.11.6	Simultaneous location and mapping . . . . .	30
3.11.7	Cooperative navigation . . . . .	30
<b>4</b>	<b>Navigation sensors</b>	<b>31</b>
4.1	Inertial navigation system (INS) . . . . .	31
4.1.1	Gyroscope . . . . .	31
4.1.2	Doppler Velocity Log (DVL) . . . . .	32
4.1.3	Magnetic compass . . . . .	32
4.1.4	Depth sensor . . . . .	33
4.1.5	Global Positioning System (GPS) . . . . .	33
4.2	Acoustic positioning system . . . . .	33
<b>5</b>	<b>Methodology</b>	<b>35</b>
5.1	Process model . . . . .	36
5.2	Measurement model . . . . .	39
5.3	Correction . . . . .	43
5.4	Sensor fusion . . . . .	43
5.5	Implementation . . . . .	44

<b>6 Results</b>	<b>45</b>
6.1 Real navigation scenario . . . . .	45
<b>7 Conclusions</b>	<b>57</b>
7.1 Future work . . . . .	58
<b>Bibliography</b>	<b>63</b>

# List of Figures

- 1.1 Nessie AUV design. . . . . 2
- 3.1 Filtering process. . . . . 11
- 3.2 Different variants of LBL: A - transponder, B - transducer. . . . . 15
- 3.3 Nonlinear transformation of a Gaussian random variable and estimation of its statistical parameters - mean and variance, using linearisation approximation and unscented transform. . . . . 24
- 4.1 Sensor fusion diagram. . . . . 34
- 5.1 AUV state vector values and five degrees of freedom. . . . . 37
- 5.2 Two modes for combining together sensor measurements into observation. . . . . 42
- 5.3 Importance of angle wrapping. Angular values used as gain should be wrapped around zero. . . . . 44
- 6.1 AUV localisation using EKF with high confidence in yaw measurement,  $SD_{yaw} = 0.01rad \approx 0.6^\circ$ .  $SD_{yawRate} = 0.004 \frac{rad}{s}$ ,  $SD_{u/v} = 1 \frac{cm}{s}$ . . . . . 48
- 6.2 AUV localisation using EKF with low confidence in yaw measurement,  $SD_{yaw} = 0.2rad \approx 11.5^\circ$ .  $SD_{yawRate} = 0.004 \frac{rad}{s}$ ,  $SD_{u/v} = 1 \frac{cm}{s}$ . . . . . 48
- 6.3 Multipathing can cause outliers in LBL position measurement. Due to reflection, several distances are detected, some being false measurements. . . . . 51
- 6.4 Spiral trajectory and the trajectory estimation using EKF. . . . . 54

6.5	EKF localisation using only inertial measurements as observation. . . . .	54
6.6	EKF localisation aided with GPS position updates weighted by setting appropriate parameters. . . . .	55
6.7	Linear and angular velocities during square-shaped trajectory. . . . .	55



# List of Tables

3.1	Overview of the state-system models. . . . .	10
4.1	Navigation sensors characteristics. . . . .	34
6.1	EKF navigation parameters. . . . .	47

# Acknowledgments

I would like to thank European Commission for financing and VIBOT consortium for giving me the opportunity to participate in an Erasmus Mundus program. It's been a valuable experience to live and study all across the Europe past two years. I am grateful to my course colleagues for the memorable moments and help, university staff and administration and Ocean Systems Lab for being an inspiring environment. I express special gratitude to my thesis supervisor Yvan for all the advice and the time spent solving problems, and to Joel for his patience and knowledge. Finally, I am grateful my family who has been an honest support and a true friend all this time.

# Chapter 1

## Introduction

### 1.1 Why underwater?

Manhood has managed to conquer variety of environments. At some point, humans could walk on the moon, send expeditions to cold or remote areas in different corners of the planet. Seabed embraces the largest part of the Earth's surface. Deep ocean world is a harsh environment with many discoveries yet to be revealed. Robots have potential to help in achieving those discoveries.

### 1.2 Localization - Where the Robot Is?

One of the important features of the autonomous underwater vehicle is being capable of localising itself within an environment - to estimate its metric position and orientation in three dimensional space. Knowing where it actually is enables further tasks of an autonomous robot such as path tracking or various manipulation tasks. Therefore, it has the same role as parts of human brain devoted to the navigation.

The main topic of the thesis is localisation of an autonomous underwater vehicle. Localisation essentially deals with the problem of estimating the position and orientation of the vehicle with respect to the defined reference system. Simply, answering where the robot is. Basic instrument in accomplishing the localisation is the sensor set. Sensory system is used to supply the localisation algorithm with the feedback on vehicle movement, control or state of the surrounding environment and its relation with the vehicle. One of the difficulties when interpreting the sensory information is the very nature of the measurement - noisy and uncertain. Majority of the measurements obtained from sensors are based on evaluation of the distance in water environment. Many conditions can influence the performance of the localisation, including the



Figure 1.1: Nessie AUV design.

very beginning: whether we can make some estimate on the initial position or not [34].

### 1.3 Autonomous underwater vehicle (AUV)

Different sorts of underwater robots have been developed throughout recent decades of research. They all have various performance capabilities, costs, accuracy, power consumption characteristics. The algorithm presented in this thesis was applied on Ocean System Lab's (<http://osl.eps.hw.ac.uk/>) torpedo-shaped Nessie vehicle (figure 1.1) that belongs to special-task AUV category combining together FOG-based Inertial Navigation System, DVL and LBL sensing for aiding navigation (Chapter 4). The architecture of the vehicle and its system capabilities are flexible on number and types of sensors employed leaving space for additional equipment to assist in missions, or improve the current navigation with already existing devices, such as forward or downward looking cameras.

### 1.4 Contribution of the thesis

Thesis is reporting the application of Extended Kalman Filter for localisation of the above mentioned Nessie AUV in an unstructured environment. The concept of sensor fusion was explained. The main contribution is the implementation of an EKF estimator adopted to work on a real underwater vehicle with real-time signals received from sensors. Furthermore, design of the estimator requires establishing a prediction model. Five d.o.f. model of the vehicle dynamics was introduced to take the role of the prediction. Apart from the algorithm-level investigation, following work examines the problem from the perspective of engineering a successful AUV navigation in general. The issue of accurate heading and the outliers in absolute

position measurement was analysed. Unscented Kalman Filter was implemented as an attempt to improve the performance and compensate for the shortcomings of the EKF. Eventually, simulations and some real scenarios were carried out resulting in analysis of the performance, advantages and disadvantages of the method.

## 1.5 Thesis outline

The first chapter § 2 introduces the task of robot navigation. Chapter § 3 explores the possible methods to navigate the robot and gives an overview of the literature that covers AUV navigation. A digression on theory of nonlinear filtering and Kalman Filtering algorithms such as was EKF and UKF was included in the chapter. Since the investigated navigation task relies on sensory data, a short overview on vehicle sensors is given in Chapter § 4. Methodology describing the details of the implementation was shown in Chapter § 5. Finally, results obtained from the real missions were given in § 6, with concluding remarks placed in last section § 7.

## Chapter 2

# Problem Definition

Robots are designed to help humans by replacing them or helping them in accomplishing certain task. This observation stands for the underwater robots in their mission of exploring vast environment such as water. It is essential for successful application of an underwater vehicle that the vehicle can accurately estimate its own position in the environment. It autonomously makes decisions that are influenced by position information. Moreover, usability and quality of the data corresponds to the precision of the localisation.

### 2.1 Robot localization

With the discovery and the development of Global Positioning System (GPS), the issue of the localisation of all robots operating on land or in the air was solved with the considerably cheap and reliable system. However, designers of those machines that operate underwater remain being in quest for the localization system of similar performance that would work in vast and unique environment such as water. Due to absorption of radio frequencies in the salty water, GPS signal is not available in the sea depths, and radio-wave-based localization which serves as standard “above the surface” cannot be used. As a consequence, various methods are developed to establish the localization underwater using all potentially useful sensor information.

The improvements of inertial instruments’ performance enabled their usage in localization for calculating the odometry. Different types of sensors are typically used and the role of mathematical algorithm such as Kalman filter or its variations in this application would be to combine together sensor measurements from all the different sources and make an optimal estimate of the vehicle state: its position and orientation.

The aim of the project is to implement and evaluate navigation algorithm for an underwater vehicle. Navigation, as introduced in literature, implies two capabilities [16]:

- localization - accurate determination of the vehicle position and velocity with respect to a known reference point
- planning and the execution of the movements between locations

The work presented in the thesis will focus on the first capability. Moreover, the first capability is a necessary step in carrying out the second capability correctly. To accomplish the task, sensor information is integrated in calculations. The purpose is to calculate the vehicle position in every moment as accurately as possible. Mathematical tool that will integrate the measurements and establish the final estimate is well known Kalman filter [26] algorithm. More details about Kalman filter in Sections § refsec:kf and § refsec:ekf.

It is important to mention that localization underwater tends to be a different challenge compared with the localization on land or in the air. Namely, the usage of GPS signal (absolute position information) is limited since it is available only on water surface. Therefore, speed and heading information obtained from inertial navigation system (INS) sensors is used together with the mathematical model of the motion to calculate the position, orientation and velocity via dead reckoning. Such strategy eventually leads to progressive error since measurement errors are integrated each time. To overcome this, GPS information which gives absolute position is used to make corrections. It updates the position information whenever available - either directly, if the vehicle is on the surface or using long baseline acoustic positioning (LBL).

Another obstacle in managing localization is the water environment itself. Usage of sensors based on light transmission such as camera, is limited because environment is such that it deforms or decreases the signal. A useful tool to determine the distance or the speed in water is sound, a mechanical wave that does not severely depend on light conditions and moves faster in water.

**Issues:** Localisation algorithm starts with the initial estimate of the location and continues making estimation by processing available information obtained from measurements. How the information from several sensors can be combined together used to estimate vehicle position? Measurements are of various kinds, therefore, one of the problems to solve on the way to having a successful localisation is the integration of those different sensors together into one unique representation of the environment [31]. Such problem is recognized as *multi - sensor fusion*. It is certainly one of the crucial tasks to solve in navigation of an underwater vehicle. Localisation algorithm is intended to take the input from different sensors and calculate the best possible estimate of the vehicle position.

Having measurements is a necessity, but having noisy or even false measurements or vehicle states is a problem to deal with. Kalman filter is intended to treat the signal as being a random variable or a function of a random variable with Gaussian distribution. Standard deviation of random variable manifests its uncertainty which compensates for the role of the noisy state or

measurement.

Navigation of underwater robot is vital part of inspection and mapping tasks. Large submarines use precise measurements taken from high-priced sensory devices. Future seems to involve more and more usage of small, less expensive systems, that tend to fuse together information from variety of inexpensive, less accurate sensors in order to improve the quality of the navigation.



## Chapter 3

# Navigation capabilities of AUVs

This chapter gives an overview of the main navigation elements for an underwater vehicle. Methods and existing algorithms for underwater vehicle localisation have been summarized. Section § 3.11 gives an overview of the literature and related work reporting various AUV navigation methods.

Carrying out underwater vehicle localisation implies using concepts such as *vehicle state* within some *navigation strategy* framework. Those two concepts will serve as a starting point for reviewing different methods. It is possible to refer to the definition of the vehicle state and its features when categorizing navigation solutions. On the other hand, navigation solutions can be essentially based on different ideas (strategies). In addition, we could treat any kind of localisation as absolute or relative, depending on which reference system we use when obtaining measurements. Absolute localisation takes environment point as reference system while relative considers the vehicle itself to be the reference. Most of the techniques surveyed here deal with absolute localisation.

### 3.1 State estimation

Vehicle navigation state describes its position within the environment. The state is a vector that contains variables relevant for localizing the vehicle. State interpretation would further categorize navigation methods on those that treat state as stochastic: linear or nonlinear, or deterministic [29]. Thus, navigation state estimators can be based on stochastic state estimators or deterministic state observers [29].

## 3.2 Stochastic state estimators

The name of stochastic state estimation methods suggests that states are treated as ultimately having feature of randomness built-in. That means being or having a random variable, or grouping random values in certain manner. It does not seem to be a wrong conclusion after recognizing that state of a system, is seldom known precisely. It is the essential nature of the process or the instrument used for measuring or the estimation algorithm itself that it fails at submitting utterly accurate data all the time. We could say that many natural phenomena are random with certain distribution of the randomness. Random variable distributions are conveniently treated as Gaussians. Statistically speaking, estimation is a rule used to calculate an estimate of a variable of interest using the observed data. As it is the case with random variables, we can say that certain state has an expected value, and that such “randomness” can be expressed with the distribution formula, resulting in descriptor values such as mean and standard deviation that fully describe the distribution in particular case of Gaussian, for instance. This approach has been applied often in underwater navigation. Most notable stochastic state estimator is Kalman filter (Section § 3.3). Kalman filter is an unbiased, optimal estimator [20, 26]. KF works through iterations by employing the process model for making the state prediction and the observations for doing the state correction (Figure 3.1). Kalman filter and the Extended Kalman Filter (EKF) treat random variable as having a Gaussian distribution. Sections § 3.3 and § 3.11.1 provide more details on Kalman filtering. A number of works report on usage of different variations of Kalman filters for state estimation.

### 3.2.1 Linear stochastic state estimators

Localisation of a robot naturally requires sensor measurements. Methodologies that work with state estimation employ measurements in terms of adding them as supplementary information to the mathematical model of object movement. Linear kinematic models are not suitable for describing the dynamics of the vehicle, therefore most of the solutions implement nonlinear stochastic state estimation.

### 3.2.2 Nonlinear stochastic state estimators

The necessity of linearising the plant and observation models to comply with linear Kalman Filter is the basis of EKF derivation. Various works report the usage of other interesting nonlinear estimators such as Unscented Kalman Filters (UKF). Methods that are based on random sampling (“Monte Carlo methods”), for instance Particle Filters (PF), are also used for localization underwater, which brings us back to the concept of stochastic value, but from the perspective of sampling those stochastic values. The work presented in the thesis focuses

on nonlinear stochastic state estimators such as EKF and UKF. **Nonlinearity** is a common phenomenon. Real world consists of various nonlinear systems. Practical situations often demand the usage of approximations that eventually lead to linearisation. The solution obtained this way is claimed to be sub-optimal - not perfectly tuned, but, indeed, useful. The problem is to consecutively make an estimation of the state of a dynamic system using a sequence of noisy measurements [35]. State-space approach turns out to be suitable choice when dealing with nonlinearity and estimating (filtering) values of the group of variables. Number of filters have been designed to deal with the phenomenon. Depending on the methodology, they could be roughly categorized as [35]:

- those that use analytic approximations (e.g. EKF)
- those that use numerical approximations
- those that use multiple models
- those that use sampling (e.g. UKF)

Since approximations eventually lead to linearising the system, a short overview of linear Kalman Filter (KF) is given in order to make an introduction on something that will be the basis for methods presented.

### 3.3 Kalman Filter (KF)

KF [26] is a well known mathematical tool that offers solution to *linear-quadratic problem* in form of an estimator [20,35]. Such linear estimator is optimal in terms of any quadratic function of the estimation error [20]. It is based on an iterative and recursive process. In addition, it is well suited framework for blending together different sensor measurements. Mathematically speaking - world consists of variety of systems that change their state ( $\mathbf{x}(k)$ ) in time. Guided by this foundation, science has established a concept of *linear dynamic system model* (Table 3.1) consisted of *process model* and *measurement model*. *Process model* (Table 3.1) is perturbed by Gaussian white noise ( $\mathbf{n}$ ). It emulates the behaviour of a phenomenon (change of the states) together with its hereditary randomness. *Measurement model* (Table 3.1) emulates observations of the system state. Observations are expressed as linear functions of state variables corrupted with Gaussian white measurement noise ( $\mathbf{m}$ ), similarly as the process model itself. System can receive control inputs ( $\mathbf{u}(k)$ ). Covariances of the process ( $\mathbf{n}$ ) and measurement noise ( $\mathbf{m}$ ), are  $\mathbf{Q}$  and  $\mathbf{R}$  respectfully. Covariances are important components of filtering algorithm 1. They can be interpreted as uncertainties in particular prediction or measurement.

Table 3.1: Overview of the state-system models.

System models overview	
$\mathbf{x}$ - system state vector, $\mathbf{u}$ - control input, $\mathbf{n}$ - process noise, $\mathbf{m}$ - measurement noise	
Linear System Model	Nonlinear System Model
process model:	
$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k) + \mathbf{n}(k-1)$	$\mathbf{x}(k) = f(\mathbf{x}(k-1), \mathbf{u}(k), \mathbf{n}(k-1))$
measurement model:	
$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{m}(k)$	$\mathbf{z}(k) = h(\mathbf{x}(k), \mathbf{m}(k))$
$\mathbf{A}$ matrix associates $(\mathbf{x}(k-1))$ and $(\mathbf{x}(k))$	$f()$ nonlinear process function
$\mathbf{B}$ matrix associates $(\mathbf{u}(k-1))$ and $(\mathbf{x}(k))$	$h()$ nonlinear measurement function
$\mathbf{H}$ matrix associates $(\mathbf{x}(k))$ and $(\mathbf{z}(k))$	

$$E\{\mathbf{n}(k)\} = E\{\mathbf{m}(k)\} = 0$$

$$E\{\mathbf{n}(k)\mathbf{n}(j)^T\} = \delta_{kj}\mathbf{Q}, E\{\mathbf{m}(k)\mathbf{m}(j)^T\} = \delta_{kj}\mathbf{R}$$

Discrete Kalman Filter is an optimal unbiased minimum mean squared error estimator. It is a calculation process that works recursively, passing iterations as shown in diagram 3.1. Kalman filter uses three basic steps: prediction, measurement and update. One iteration uses process equation, next one proceeds further using prediction result within the observation equation. Recursion continues each time referring to previous filter output.

Assumptions that KF uses:

- distribution of a random variable is assumed to be Gaussian, therefore mean and variance can fully describe it
- linear transform of a Gaussian distribution gives another Gaussian distribution

In spirit of that, noise vectors  $(\mathbf{n}, \mathbf{m})$  and thus linearly derived state and observation vectors  $(\mathbf{x}, \mathbf{z})$  are Gaussian. Another assumption is that noise vectors  $n, m$  have zero mean values (“white Gaussian”) and that their elements are not correlated, resulting in diagonal matrices  $\mathbf{Q}$  and  $\mathbf{R}$  (Table 3.1).

KF can be summarized with the set of formulas given in Algorithm 1. The aim of the equations is to recursively obtain the estimate of the state vector  $\hat{\mathbf{x}}$  and the uncertainty of such estimate. Uncertainty is described as state variance  $\mathbf{P}(k) = E\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\}$ . KF defines states as collection of elements with Gaussian distribution, thus mean value is used as an estimate and variance as a measure of how far the values are spread out around mean. Notation follows the one from the Table 3.1.  $\hat{\mathbf{x}}(k | l)$  is a state estimate at the time  $k$  using observations obtained until time moment  $l$ . It is a recursive estimator since

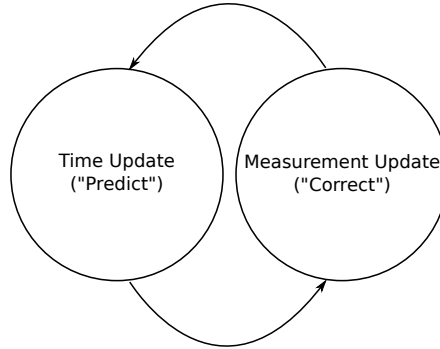


Figure 3.1: Filtering process.

every estimation relies on previous state estimation and current observation (measurement). Although history of previous observations is not directly used, it is still incorporated in previous state estimation. Previous state estimation is used in current state estimate due to recursive nature of the algorithm. Transformation uncertainty  $\mathbf{Q}$  is updated simultaneously with the state. Innovation  $\nu$  presents difference between the real observation and predicted observation. Both innovation and its covariance matrix  $\mathbf{S}$  are included in calculation of Kalman gain  $\mathbf{K}$ . Kalman gain does the final state estimate correction influenced by recent observation and makes the uncertainty optimal with respect to quadratic estimation error criteria. Although

---

**Algorithm 1** The Discrete Kalman Filter
 

---

**Require:**  $E\{\mathbf{x}(0)\} = \mathbf{x}(0) = \hat{\mathbf{x}}(0)$  {initialize state}

**Require:**  $\mathbf{P}(0) = \delta_{jk} \mathbf{P}_0$  {initialize covariance}

**loop**

$k \leftarrow k + 1$

$\hat{\mathbf{x}}(k | k - 1) = \mathbf{A}\hat{\mathbf{x}}(k - 1) + \mathbf{B}\mathbf{u}(k)$  {state prediction}

$\mathbf{P}(k | k - 1) = \mathbf{A}\mathbf{P}(k - 1)\mathbf{A}^T + \mathbf{Q}$  {state prediction uncertainty}

$\nu = \mathbf{z}(k) - \mathbf{H}\hat{\mathbf{x}}(k | k - 1)$  {innovation}

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

$\mathbf{K} = \mathbf{P}(k | k - 1)\mathbf{H}^T\mathbf{S}^{-1}$  {"Kalman gain"}

$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k | k - 1) + \mathbf{K}\nu$  {state correction}

$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}(k | k - 1)$  {state correction uncertainty}

**return**  $\hat{\mathbf{x}}(k), \mathbf{P}(k)$

**end loop**

---

essentially intended for dealing with linear system, KF formulas are be important starting point in understanding nonlinear filtering accomplished with Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF).

### 3.4 Deterministic state estimators

Deterministic state estimators refer to non-stochastic system (plant, process) and observation (measurement) model. Stochastic means that the input and output can manifest in some random behaviour. Concept of deterministic state does not imply any uncertainty. However the estimation, apart from being precise and accurate, can be characterized with its stability. Estimation is exact (no randomisation of the variables) and estimator should be asymptotically stable [28]. Stability is defined using different mathematical criteria [28]. In this case, navigation elements such as speed or position or even full-state vector are an output of some defined transfer function. In order to know the transfer function, an estimator of the nonlinear transfer function model is used. Such approach utilizes the exact knowledge of nonlinear dynamics of the vehicle. Essentially, estimating deterministic state implies passing the input data through a certain transfer function so that the outputs are localisation related values. Transfer function is defined using known formulas of the transfer function and parameters which are estimated by giving particular input and observing the corresponding output of the system in order to recognize its behaviour. Being a completely different concept, in terms of methodology they are not the focus of the thesis.

### 3.5 Strategy

Now that the state vector is revealed as a storage for describing the vehicle location, finding a way to filter that state vector - stochastic or deterministic, linear or nonlinear, can employ different approaches. At this level, we can talk about the strategy - general approach, an idea. The primary navigation system in most of the applications, including underwater navigation, is Inertial Navigation System (INS) (Section § 4.1). Since such system accumulates noisy data, it introduces the drift errors that need to be occasionally corrected inside the navigation algorithm. Various ways of correcting those errors were developed. Most widely known “correction tool” is the incorporation of an absolute position measurement. Numerous literature that considers integrating occasional GPS or LBL measurement within the stochastic state estimation algorithm is presented in Section 3.11. Oceanographic community typically uses three different strategies to handle the absolute positioning underwater [40]: (1) transponder networks on the seafloor (long baseline, LBL), (2) ship-AUV communication (short baseline), and (3) sensors mounted on the underwater vehicle that measure range and inertial motion. They can be combined together depending on the idea, purpose or conditions. Each of the strategies is different in terms of accuracy, costs, complexity [12] or types of sensors used. Section § 4 gives more insight into performance and categorization of each inertial sensor device used for underwater vehicle navigation. Naturally, most of the conventional methods rely on acoustic waves used

for measuring the distance. Nevertheless, visual information is used, especially in transparent or structured environment [5]. Navigation can be combined with simultaneous localization and mapping approach (SLAM, Section § 3.8), or can be terrain aided (Section § 3.7) [29]. Although not common for many applications, visual information recorded by the camera or a pair of cameras can be used to aid navigation (reduce the drift) [2, 13]. If the localization involves control of the vehicle movement coupled with localisation and environment information, then it is addressed as active localisation. On top of already mentioned methods, some novel strategies in which vehicles communicate among themselves, such as cooperation for navigation (Section § 3.9), are explored [2]. Following chapter gives an overview of the mentioned strategies.

### 3.6 Acoustic-based localization techniques

In the absence of possibility to transmit radio waves, acoustic communication emerges as solution for communication underwater. Considering that the electromagnetic waves are absorbed and the propagation of light is limited, positioning cannot rely on GPS signal, laser scanners, visually aided navigation, or radio communication. Therefore, state of the art in absolute positioning of the robot underwater implies triangulation using distances from navigation buoys positioned at the known locations (Figure 3.2(d)). Alternative solution is to surface back in order to update the position using GPS.

Underwater acoustic positioning system is the main tool used to track underwater vehicles. Reason for relying on acoustics is the nature of the water environment: resistant to radio waves, leaving out mechanical acoustic disturbance as the only mean of communication. Three classes of underwater acoustic positioning systems are used (Figure 3.2): Long Baseline (LBL), Ultra Short Baseline Systems (USBL), Short Baseline Systems (SBL) and GPS intelligent buoys (GIB).

LBL systems (figure 3.2(a)) use a network of two or more sea-floor mounted (anchored) baseline transponders to reference the navigation. Such system is considered to be accurate, generally with accuracy better than 1 meter - usually around few centimetres [22]. However, communication-wise, such system is convenient for small number of vehicles, since one vehicle can query the network each time [2]. Hence, having a large number of vehicles can cause delays in update. Elapsed time between moment of sending the query and receiving the response is used to estimate the time-of-flight ( $t_{flight}$ ) of the wave and eventually the distance ( $d$ ) between beacon and the vehicle, considering that the speed of the sound ( $c$ ) is known and the essential relation  $c = \frac{d}{t_{flight}}$  is used for calculation. By using methods such as triangulation, these distances can be used to compute the AUV's absolute position. LBL systems can be long-range or short-range. Long-range systems use 12 kHz frequencies for communication range of 10 km

of distance with the error varying from 1 up to 10 m [2, 41]. Short-range systems use 300 kHz frequencies and operate within the range of 100 m with sub-centimetre precision [2, 41].

SBL systems (figure 3.2(c)) do not require sea-floor mounted transponders. Instead SBL system uses a vessel equipped with high-frequency directional emitter in order to accurately determine the AUV position with respect to the vessel [30]. Disadvantage of such system is the need for providing a vessel and the distance limits since the range between the ship and the AUV has to be short. Moreover, SBL accuracy improves with transducer spacing (possibility of longer baseline). Similarly, the range measurements are used to triangulate the position. Transducer sends a signal, transponder located on the vehicle responds yielding distance information. AUV's location is determined with respect to transducers' location.

USBL systems (figure 3.2(b)) uses similar beacons as LBL system. Difference is that vehicle has transceiver with several receiving elements positioned close to each other on a known distance so that the reply from beacons is detected by all of them. It is possible to calculate the phase difference between received signals this way which is enough to determine the bearing to the beacon. If the distance information is combined together with bearing, then the absolute position of the vehicle can be estimated just by considering the response of only one beacon.

GIB systems (figure 3.2(d)) consist of floating buoys supplied by GPS signal carrying transducers. Vehicle has a transponder that replies to transducer query with acoustic signal, enabling buoys to register the time-of-flight. Such concept communicates opposite way from the one accomplished in standard LBL.

Common feature for all the mentioned systems is that the position is inferred from the acoustic feedback of transponders so that the vehicle is capable of locating itself with respect to transponders. Outliers and noise are

### 3.7 Terrain-aided navigation

Terrain-aided navigation can be used to determine the vehicle position using topographic, magnetic or gravitational data [29]. Terrain-aided navigation can rely on the map of the sea bottom, its elevation or some particular landmarks that are detected to fix the vehicle position. In such circumstances, it is possible to define the map first and try to navigate with respect to that map (map known a-priori) or manage the mapping and navigation simultaneously using sensor data to build the map from the scratch, step by step. Disadvantage of terrain based methods lies in fact that they depend on precision of the map of the floor and ability of the vehicle to sense the depth or image the sea floor. In most of navigation scenarios, a-priori maps are not available [29]. The essential sensor for terrain aided navigation is sonar that measures the



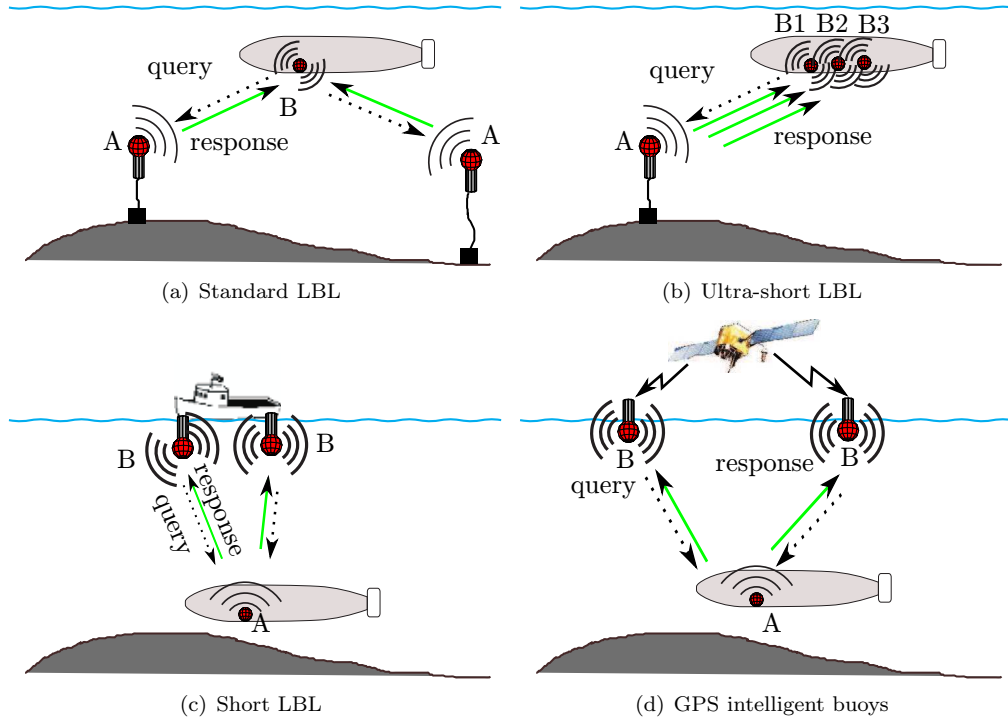


Figure 3.2: Different variants of LBL: A - transponder, B - transducer.

distance (“time-of-flight”). However it is possible to use the optical sensor devices, for instance cameras, and process the visual information. Range of optical sensors is much shorter, they usually require structured environment, fairly good visibility within the water and the optical information cannot spread as freely and as far as the acoustic information does. However, its nature of information is different, richer with different types of data, including the raw position data.

### 3.8 SLAM

SLAM concentrates on establishing an autonomous navigation algorithm that would be environment based. Therefore, it is reducing the need for additional infrastructure and using the spatial information from environment to bound the position error. Idea of the concept called SLAM is to localize the robot with respect to environment landmarks [37]. To accomplish that, two challenging tasks have to be solved: extracting the features and finding a way to measure correspondence between measurement and the feature. Literature offers some solutions to SLAM based approach. Brief overview of some of the methods is given in Section § 3.11.

### 3.9 Cooperation for navigation

Advancement in acoustic modems has made it possible to establish more reliable communication and develop communication technologies that engage data from more than one vehicle [2, 15]. Idea of involving number of vehicles in exploration is to consider sensor and state information sent from the other vehicles. Advantage of this approach is reduction in amount of sensor equipment, for instance LBL transponders or bathymetry sonars. However, more vehicles need to be deployed.

### 3.10 Active localisation

Active localisation integrates together control of the robot motion and localisation algorithm. Active refers to vehicle “activity”, mobility or movement since these strategies sense the environment and combine together localisation with control so that the quality of the localisation is improved. Idea is to guide the motion of the vehicle in order to make it more convenient for the localisation algorithm to work out the tracking [33].

### 3.11 Related work on AUV navigation

Following paragraphs summarize the documented ways to process the sensor information in order to be able to estimate the position within the environment.

In existing survey on underwater vehicle navigation, Kingsey [29] gives a summary presenting the methods used for navigation. As introduced in [29], current vehicle position is referred as navigation state - a vector whose elements express where the vehicle is and how it is oriented in space with respect to some reference. Localization simply means finding a way to estimate navigation state vector. Naturally, sensors are providing the data for the estimation.

The simplest approach would be dead reckoning - to take the raw sensor measurements and use them directly or within a simple mathematical model that describes the vehicle dynamics. Instead, many techniques presented in literature utilize sensor data as supplementary information together with the information from the kinematic model.

Underwater navigation is using several instrumentation methods to carry out the robot localization in the sea [40]. These include transponder networks placed on the bottom of the sea, tracking systems between the ship and the underwater vehicle, and sensory devices that measure range and dynamics mounted on the vehicle itself. Each of the methods has its advantages and disadvantages. Transponder network gives accurate position information, however using it requires installing and calibrating additional equipment [12].

### 3.11.1 Kalman filtering localisation

Some past works have already dealt with the issue of managing robot localization using Kalman filtering. Master thesis of Negneborn [31] gives a useful overview of the theoretical knowledge in field of probability and estimation. Thesis also surveys the utilization of Kalman filtering for localizing vehicles in general. Emphasis is on application in robotics. Several experiments have been reported in the thesis with a detailed discussion. This work is a good starting point in learning and understanding the problem of localization on physical robots in general and, in particular, usage of Kalman filtering for that purpose.

Blain et al. [3] study the application of Kalman filter in navigation of an underwater vehicle used for water dam inspection focusing on merging position orientation and velocity information. This algorithm uses acoustic positioning sensor together with the integration the DVL sensor (Chapter § 4) measured velocities [3] to estimate the position. Apart from reporting performance of *sensor-fusion* in real application, several issues have been pointed out and dealt with in their work, such as managing asynchronous information that arrives from sensors. It is reported that Kalman filter output could be corrupted in situations when observations (sensor measurements) are subject to interruption or periodic stopping. Due to the fact that not all the sensors can be always available, it is usually necessary to be capable of adding or removing sensor observations from the system without changing the navigation algorithm.

Asynchronous data delivery in this particular case means that the DVL sensor (Chapter § 4) provides data with higher rate [3]. Such obstacle was solved by switching to estimation procedure so that it is suitable for that particular sensor measurement scenario. This simply means that if the acoustic sensor and DVL sensor asynchronously provide new measurements, Kalman filtering is used to carry out the fusion by doing filter switching process. Each sensor has a filter process attached to it and the sensor itself defines which filter becomes active for the actual measurements. Otherwise, pure DVL velocity measurement is just integrated to update the position, as dynamic model would suggest. Blain also analyses delays in absolute position measurement and validity of absolute position data if they are delayed. This effect evident in case of acoustic waves, where the real time of the measurement is current measurement time minus the time it took for the acoustic signal to arrive. It is especially visible in cases when acoustic signal moves across a longer distance. This causes a delay in data arrival. To overcome this, position estimate between two acoustic measurements is memorized. Position is meanwhile normally updated by integrating the DVL data that happen more frequently. To compensate for the time it took for the acoustic signal to arrive, timestamp of the moment when the data was produced was estimated and memorized together with the current position at the timestamp. The procedure consists of two stages: at first, a new position estimate is made using the received acoustic position data to update the position at the timestamp. Finally,

the correction is applied on the updated position by integrating velocities that happened in meanwhile - from the timestamp till the actual time. This serves as the correction of the position estimate error influenced by significant time of flight of the signal.

Drolet et al. [8] introduce a flexible localization strategy based on sensor fusion and usage of several Kalman filters arranged together in a bank. Each filter is reduced to express a simple kinetic equation. In addition, each filter processes one state - works in one “dimension”. Idea is to integrate together sensor measurements that arrive at different time moments from different sensors. Method takes asynchronous information from sensors, manages a filter switching process so that the most recent data is used to update those filters that can be updated with such measurement [8]. Such sensor fusion strategy is adaptable in terms of number of sensors so that the best is taken out from the available input data, more robust to data loss. Moreover, asynchronous inputs are allowed.

Di Massa et al. report usage of Kalman filter framework for slightly different concept of navigation that takes surrounding terrain as reference for estimating the position of the sonar (“terrain-relative navigation”), [6]. In their work, sonar image is matched to the map using mean absolute difference (MAD) as the matching criterion. Matching map location is considered as the measurement of the vehicle position [6]. Matching process is not entirely transparent - there are always several candidates eligible to be candidate for best matching. Thus several matchings are selected and weighted depending on how much they relate to the terrain images. Weights correspond to uncertainties in estimation theory. Quality of similarity is used to weight each measurement. Solution consists of having resulting best estimate of location [6]. Information from selected matches is combined to make the best estimate. The role of Kalman filter framework is to carry out the estimation. Each of the chosen matches is considered as one measurement together with its weight as uncertainty. The filtered state is the position of the image within the map.

Gade and Jalving introduce post processing aided navigation system deployed on a commercial underwater vehicle [17]. Idea is that the underwater vehicle records sensor data while accomplishing mission under the sea surface (capturing images of the seabed). At the same time, a vessel is positioned on the surface receiving information of its position through the reliable Differential Global Positioning System (DGPS). After the mission is over, data are combined together with position data that was simultaneously recorded on the survey vessel located on the surface. Kalman filtering is used when merging the data. *Error-state* Kalman filter [17] is used to combine sensor measurements and their error models. Observations in case of such filter consist of the difference between measured and computed values. Instead of working directly with states, presented algorithm filters the errors, so that the ultimate position and heading estimate can be derived by subtracting the estimated errors from, as authors suggest, corresponding calculated state elements. This way, final aim of obtaining more accurate vehicle

position and heading together with the tunable accuracy of such estimate [17] contributes in improving the accuracy of obtained seabed maps.

Dissertation [36] suggests how to improve the vehicle position estimation when reconstructing maps of the sea floor. Visual information of the terrain is used as the feedback that makes terrain mapping data and the vehicle navigation data more consistent. Inspiration for investigating lies in fact that map-making depends on localization quality. Navigation errors are potentially large scale particularly seriously affecting the results when mapping is vehicle-based [36]. Existing local navigation is used together with terrain-relative measurements. Namely, terrain sub-maps are created over short periods while the vehicle works out the inaccurate localization using dead reckoning. Sub-maps are registered resulting in position measurements between two vehicle states, placing an additional constraint on the vehicle position estimates. Delayed EKF is used to merge together the measurements (“sub-map” registrations and previously reached vehicle locations) into the navigation framework. Delayed state version of the recursive EKF enables retaining knowledge of prior platform positions.

Yun et al. introduce simulation and present testing results of the navigation system that combines the usage of Inertial Measurement Unit (IMU) together with GPS fixes that occur less frequent and asynchronously [45]. Asynchronous Kalman filter with six states for orientation and eight states for position estimation is implemented [45]. Process model takes the velocities and GPS bias, models them as white noises passed through the first order systems with the time constant. Measurement consists of synchronous (periodical) velocity measurements and asynchronous DGPS information. The design of the filter for the position estimation algorithm conforms to the standard routine, with the difference that the measurement vector has different length depending on the number of available valid sensor inputs, hence it has a flexible size, but each observation updates the state vector of the fixed size [45]. The idea of the asynchronism enables that DGPS signals are used, if available and as soon as they are available, together with the speed measurements. This way, the localization algorithm uses the most of the data that are currently available.

The usage of the stochastic estimators implies using a known model that describes system state transition from one moment to another (*plant model*) and model that describes transition from state to the measurement (*observation model*). Such model does not have to be the same each time. Jakuba and Yoerger [21] study the way to optimize navigation by estimating the vehicle model parameters, for instance various dynamics or buoyancy coefficients that normally influence the model, but are treated as constant. Their study involves post-processing of the navigation data and heuristic estimate of these coefficients’ optimal value. Real missions that applied the technique resulted in reduced noise in localization data, therefore giving clearer tracking.

Julier and Uhlmann introduce the method that carries out nonlinear filtering [24]. It is

an alternative generalization of the KF that changes the approach of representing mean and variance of the random variable after it passes through an nonlinear transform. Their research is a quite useful and comprehensive theoretical overview of filtering in general and the role of Extended Kalman Filter (EKF) in switching to nonlinearity world. Introduced filter, later known as Unscented Kalman Filter (UKF) is regarded as more precise alternative to EKF that is, in addition, fairly easy for implementation. Julier and Uhlmann in their work point out the shortcomings of the EKF. In search of general method that would overcome the problem, instead of using proposed equations for projecting mean and covariance, a discrete set of points is chosen and projected using a chosen non-linear transformation (Section § 3.11.1). Idea is to use the parameters to approximate the Gaussian distribution instead of approximating the nonlinear transformation. This way, propagation of the information is accomplished directly, and the aim is to find a way to parametrise the information about the mean and covariance of the distribution. Advantages of the filtering algorithm are in terms of precision and simplicity (no need for Jacobian derivation) with empirical results for highly nonlinear problems including vehicle control indicating as good as or better performance than EKF and higher robustness.

Wan and van der Merwe [39] go further in exploring the concept of UKF introduced by [24]. Their research briefly reminds of disadvantages manifested in EKF and improvements gained with the usage of UKF. UKF theoretical backgrounds, the idea itself and meaning of used variables were explained in comprehensive manner in one of the document sections. Usage of UKF was reported together with the results in different estimation problems such as nonlinear system identification, state estimation, parameter estimation and dual estimation problems [39]. UKF according to the authors achieved higher accuracy compared with EKF in all the domains that were examined.

At this point, it is useful to make a short digression in order to give more details on EKF and UKF.

### Extended Kalman Filter

System can be described with set of states that evolve in time according to mathematical functions that are nonlinear in many applications. Table 3.1 gives an overview, categorizing systems as linear or nonlinear. Nonlinear state prediction would use previous state estimate and mean value of the process noise:

$$\hat{\mathbf{x}}(k | k - 1) = f(\hat{\mathbf{x}}(k - 1), \mathbf{u}(k), 0) \quad (3.1)$$

EKF is intended for solving sub-optimal state estimation of a nonlinear system. The main characteristic of EKF is that it analytically approximates - linearises - the process and measurement functions ( $f()$  and  $h()$ , Table 3.1). Linearisation implies approximating these functions with

their first derivative around current prediction, similarly as the ordinary functions are approximated with Taylor polynomials of first degree. In this case, derivation is slightly more complex since model functions  $f()$  and  $h()$  take several input vectors and output the resulting vector. Hence, the derivation will consist of partial derivation of process per state input vector (Equation 3.2) and per noise input vector (Equation 3.3). And partial derivation of measurement function per state (Equation 3.4) and measurement noise (Equation 3.5). Partial derivatives themselves will be Jacobian matrices considering that vector is derived per vector.

$$\mathbf{F}(k) = \frac{\partial f}{\partial x}(\hat{\mathbf{x}}(k | k - 1), \mathbf{u}(k), 0) \quad (3.2)$$

$$\mathbf{W}(k) = \frac{\partial f}{\partial n}(\hat{\mathbf{x}}(k | k - 1), \mathbf{u}(k), 0) \quad (3.3)$$

$$\mathbf{H}(k) = \frac{\partial h}{\partial x}(\hat{\mathbf{x}}(k | k - 1), 0) \quad (3.4)$$

$$\mathbf{V}(k) = \frac{\partial h}{\partial m}(\hat{\mathbf{x}}(k | k - 1), 0) \quad (3.5)$$

Subsequently, filtering process can be treated similarly as classic, discrete linear Kalman filter introduced in previous section. Process model mathematically describes how the state changes

---

**Algorithm 2** The Discrete Extended Kalman Filter

---

**Require:**  $E\{\mathbf{x}(0)\} = \mathbf{x}(0) = \hat{\mathbf{x}}(0)$  {initialize state}

**Require:**  $\mathbf{P}(0) = \delta_{jk} \mathbf{P}_0$  {initialize covariance}

**loop**

$k \leftarrow k + 1$

$\hat{\mathbf{x}}(k | k - 1) = f(\hat{\mathbf{x}}(k - 1), \mathbf{u}(k), 0)$  {state prediction}

$\mathbf{P}(k | k - 1) = \mathbf{F}(k)\mathbf{P}(k - 1)\mathbf{F}^T(k) + \mathbf{W}(k)\mathbf{Q}\mathbf{W}^T(k)$  {state prediction uncertainty}

$\nu = \mathbf{z}(k) - h(\hat{\mathbf{x}}(k | k - 1), 0)$  {innovation}

$\mathbf{S} = \mathbf{H}(k)\mathbf{P}(k | k - 1)\mathbf{H}^T(k) + \mathbf{V}(k)\mathbf{R}\mathbf{V}^T(k)$  {innovation uncertainty}

$\mathbf{K} = \mathbf{P}(k | k - 1)\mathbf{H}^T(k)\mathbf{S}^{-1}$  {"Kalman gain"}

$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k | k - 1) + \mathbf{K}\nu$  {state correction}

$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}\mathbf{H}(k))\mathbf{P}(k | k - 1)$  {state correction uncertainty}

**return**  $\hat{\mathbf{x}}(k), \mathbf{P}(k)$

**end loop**

---

for the given input (Equation 3.1). Essential invention in EKF algorithm is the linearisation of the given function around current state mean and variance which further results in estimation process similar to the one described for linear Kalman filter.

Kalman gain determines the importance of each new observation  $\mathbf{z}(k)$ . Essentially - it expresses how much we allow the measurement to influence the change of the predicted state at the correction stage. According to the state correction formula (Algorithm 2), it is determined

by the values of matrices  $Q$  and  $R$  which represent process noise covariance and the measurement noise covariance (uncertainty), respectively. Values contained in  $\mathbf{K}$  takes the range between 0 and 1 with zero (0) meaning that we use only prediction as the estimation and give no importance to the measurement. One (1) on the other hand, means that we give all the importance to the measurement.

Real world models are rarely linear. The motive for developing the Extended Kalman Filter is based on the adaptation of the linear Kalman Filter for dealing with nonlinear problems. However, it is possible that EKF significantly declines the performance quality [24]. When making a prediction of state, observation or uncertainty of any of those for nonlinear systems, linearisation can introduce errors. The inaccuracies of the EKF estimates are caused by truncating errors in Taylor series when making an approximation. One example of such case is given in [24] where the simulated object trajectory follows circular path. The reason for failing in prediction stage in such scenario is in linear approximations that EKF, naturally, uses when predicting the next state and its characteristics.

### Unscented Kalman Filter (UKF)

UKF refers to usage of the unscented transform in EKF framework [23,35,39]. Instead of propagating random Gaussian variable through nonlinear transform functions  $f()$  and  $g()$ , UKF deterministically chooses set of sample points, large enough to capture the distribution of the variable. Idea is that the probability distribution is approximated using weighting of the propagated samples rather than the approximation of the nonlinear function. Distribution is modelled as Gaussian, thus represented with mean and variance, as it is general principle with Kalman filtering. The role of Unscented Transform (UT) is in providing the technique for estimating statistical parameters of the Gaussian random variable (GRV) that undergoes nonlinear transform (Figure 3.3). UT provides the pattern for sample selection and scheme for estimation of statistical parameters (such as mean and variance for GRV) after nonlinear transformation of selected samples.

**Unscented transformation (UT).** Provides a method for calculating statistics of a random variable that undergoes nonlinear transform. Gaussian random variable  $\mathbf{a}$  of vector length  $L$ , mean value  $\bar{\mathbf{a}}$  and variance  $\mathbf{P}_a$  is passed on to a nonlinear function  $g()$  that eventually creates another random variable,  $\mathbf{b}$  of the same length.

$$\mathbf{b} = g(\mathbf{a})$$

The aim is to estimate mean and variance of the variable after the transformation. Such solution will be useful since the nonlinearities in process model can be significant. First order Taylor approximation was used within EKF for such task.



UT defines the set of  $2L + 1$  samples ( $A_i$ ) which are capturing the distribution (Figure 3.3(c)). Each sample has a weight coefficient ( $W_i$ ) attached to it. Weights are normalized so that they all sum up to one. Weights will be used in calculation of the mean and variance of obtained variable  $\mathbf{b}$ :  $\bar{\mathbf{b}}$  and  $\mathbf{P}_b$ . Samples and their respective weights are formed using equations:

$$\begin{aligned} \mathbf{A}_i &= \bar{\mathbf{a}}, & \mathbf{W}_i &= \frac{k}{L+k}, & i &= 0 \\ \mathbf{A}_i &= \bar{\mathbf{a}} + (\sqrt{(L+k)\mathbf{P}_a})_i, & \mathbf{W}_i &= \frac{1}{2(L+k)}, & i &= 1, \dots, L \\ \mathbf{A}_i &= \bar{\mathbf{a}} - (\sqrt{(L+k)\mathbf{P}_a})_{i-L}, & \mathbf{W}_i &= \frac{1}{2(L+k)}, & i &= L+1, \dots, 2L \end{aligned} \quad (3.6)$$

with parameter  $k$  setting the distance of sample points from  $\mathbf{a}$  (Figure 3.3(c)).  $(\sqrt{(L+k)\mathbf{P}_a})_i$  presents  $i$ -th row of matrix square root of  $(\sqrt{(L+k)\mathbf{P}_a})_i$ . Eventually, each selected sample is substituted in nonlinear function  $g()$ , resulting in samples  $\mathbf{B}_i = g(\mathbf{A}_i)$ ,  $i = 0, \dots, 2L$ . Mean value (first moment) and covariance (second moment) of the obtained random variable are:

$$\bar{\mathbf{b}} = \sum_{i=0}^{2L} \mathbf{W}_i \mathbf{B}_i, \quad \mathbf{P}_b = \sum_{i=0}^{2L} \mathbf{W}_i (\mathbf{B}_i - \bar{\mathbf{b}})(\mathbf{B}_i - \bar{\mathbf{b}})^T \quad (3.7)$$

$$\mathbf{b} = g(\mathbf{a}) = \begin{bmatrix} \mathbf{a}(1) \cos \mathbf{a}(2) \\ \mathbf{a}(1) \sin \mathbf{a}(2) \end{bmatrix} \quad (3.8)$$

The concept was explained through an example [35]. Two-dimensional GRV was used to express the position in polar coordinates. Such position information contains number of samples of range (distance) and bearing (angle). An example of nonlinear transformation where polar coordinates are transferred into Cartesian coordinates using function  $g()$  (Equation 3.8) was analysed. Original set of samples was shown in Figure 3.3(a). Figure 3.3(b) shows estimation of the mean and covariance using first order linearisation (Jacobian) of  $g()$  - scheme that is included in standard EKF. It is visible that both mean and covariance estimates have errors. Although dissonant, EKF still performs well in practise implying the nonlinearities are moderate [35]. In search of more accurate estimate, or at least less prone to nonlinearity, UT was investigated. Figures 3.3(c) and 3.3(d) demonstrate the usage of UT. At first, samples have been computed (Equation 3.6), transferred into Cartesian using function  $g()$  and their statistic moments revealed using weighted average functions (Equation 3.7). Compared with linear approximation, UT proves to be more accurate. It emulates nonlinearities of the prediction model better.

UKF, similarly as other Kalman filtering algorithms consists of two independent stages of prediction and update. Both stages utilize UT and its features. GRV, similar with the one from the example is used to store together state, process noise and measurement noise variables - making altogether an *augmented state* vector 3.

Augmented state provides GRVs that will be sampled, weighted and propagated through

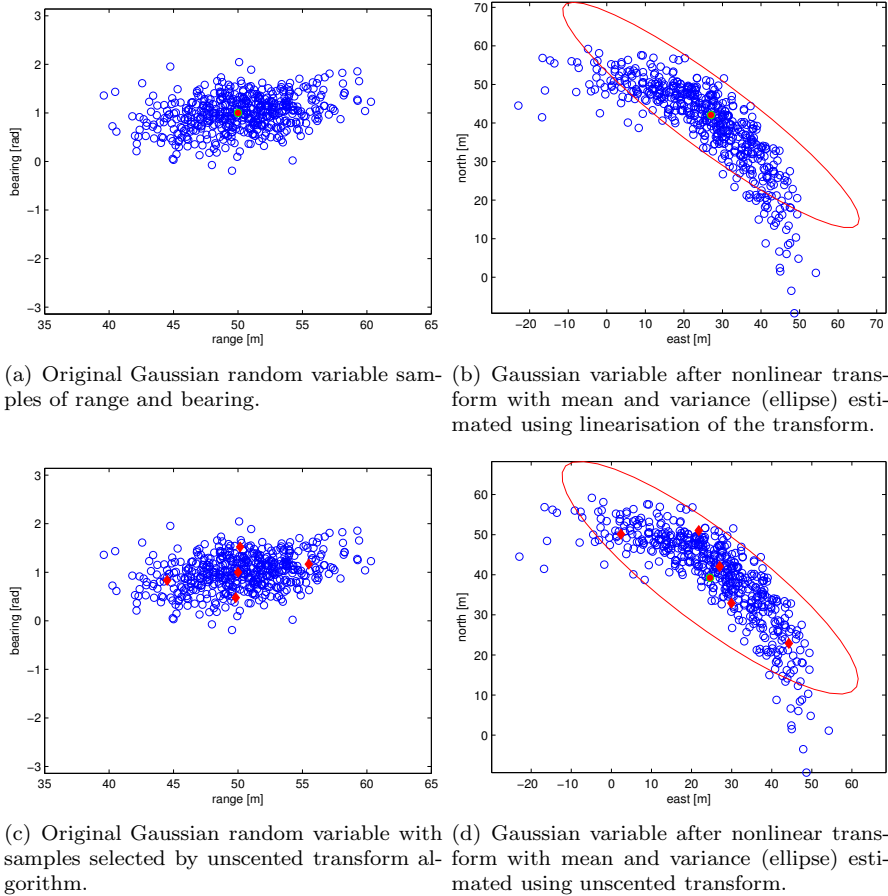


Figure 3.3: Nonlinear transformation of a Gaussian random variable and estimation of its statistical parameters - mean and variance, using linearisation approximation and unscented transform.

nonlinear process ( $f()$ ) and measurement ( $h()$ ) functions as a part of prediction and correction stage. After each correction, UKF will hold an estimate of current mean and variance, ready to continue with recursion (Algorithm 3).

Calculation-wise, the whole procedure is fairly easier compared with linearisation algorithm since there is no need for calculating the Jacobian or Hessian, total number of computations stays the same and it is easier to improvise with the algorithm with constraints or parameters which define the way samples are selected. In addition, UT shown in example (Figure 3.3) handles the whole distribution and its transformation by tracking only five samples. Higher dimensionality of GRV would result in more samples taken. However that number is still

reasonably small compared with Monte Carlo methods. The only difficulty in implementation could be the non-trivial solution of the square root of the matrix as the necessary stage in unscented transform computation (Equation 3.6).

---

**Algorithm 3** The Discrete Unscented Kalman Filter
 

---

**Require:**  $\hat{\mathbf{x}}(0) = E\{\mathbf{x}(0)\}$  {initialize state}

**Require:**  $\mathbf{P}(0) = \delta_{jk} \mathbf{P}_0$  {initialize covariance}

**Require:**  $\hat{\mathbf{x}}^{aug}(0) = \begin{bmatrix} \hat{\mathbf{x}}^x(0) \\ \mathbf{x}^n(0) \\ \mathbf{x}^m(0) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}(0) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$  {init. augmented state: state + process noise + meas. noise}

$L = \text{length}(\hat{\mathbf{x}}^{aug})$   $k = \text{const}$  {augmented state length and  $k$  parameter set}

**Require:**  $\mathbf{P}^{aug}(0) = \begin{bmatrix} \mathbf{P}_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_m \end{bmatrix}$  {initialize augmented state covariance}

**loop**

$k \leftarrow k + 1$

$\mathbf{A}^{aug}(k-1) = \left[ \hat{\mathbf{x}}^{aug}(k-1) \hat{\mathbf{x}}^{aug}(k-1) \pm \sqrt{(L) \mathbf{P}^{aug}(k-1)} \right]$  {compute samples - unscented transform}

$\mathbf{W}_i, i = 0, \dots, 2L$  {compute weights}

$\mathbf{A}^x(k|k-1) = f(\text{vect} \mathbf{A}^x(k-1), \text{vect} \mathbf{A}^n(k-1))$  {nonlinear process model}

$\hat{\mathbf{x}}(k|k-1) = \sum_{i=0}^{2L} \mathbf{W}_i \mathbf{A}_i^x(k|k-1)$  {state prediction}

$\mathbf{P}(k|k-1) = \sum_{i=0}^{2L} \mathbf{W}_i (\mathbf{A}_i^x(k|k-1) - \hat{\mathbf{x}}(k|k-1)) (\mathbf{A}_i^x(k|k-1) - \hat{\mathbf{x}}(k|k-1))^T$  {state prediction uncertainty}

$\mathbf{Z}(k|k-1) = h(\text{vect} \mathbf{A}^x(k-1), \text{vect} \mathbf{A}^m(k-1))$  {nonlinear measurement model}

$\hat{\mathbf{Z}}(k) = \sum_{i=0}^{2L} \mathbf{W}_i \mathbf{Z}_i(k|k-1)$  {measurement prediction - unscented transform}

$\mathbf{P}_{zz} = \sum_{i=0}^{2L} \mathbf{W}_i (\mathbf{Z}_i(k|k-1) - \hat{\mathbf{Z}}(k)) (\mathbf{Z}_i(k|k-1) - \hat{\mathbf{Z}}(k))^T$

$\mathbf{P}_{xz} = \sum_{i=0}^{2L} \mathbf{W}_i (\mathbf{A}_i^x(k|k-1) - \hat{\mathbf{x}}(k|k-1)) (\mathbf{Z}_i(k|k-1) - \hat{\mathbf{Z}}(k))^T$

$\mathbf{K} = \mathbf{P}_{xz} \mathbf{P}_{zz}^{-1}$

$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K} (\mathbf{z}(k) - \hat{\mathbf{Z}}(k))$  {state correction}

$\mathbf{P}(k) = \mathbf{P}(k|k-1) - \mathbf{K} \mathbf{P}_{zz} \mathbf{K}^T$  {state correction uncertainty}

**return**  $\hat{\mathbf{x}}(k), \mathbf{P}(k)$

**end loop**

---

### 3.11.2 Monte Carlo-based filtering methods

Monte Carlo methods, based on repeated random sampling as a part of the results computation are covered in several works, mostly dealing with Particle Filters (PF). Gordon et al. enclose *bootstrap filter* [19], also known as Particle Filter (PF) - a recursive algorithm based on representing state vector as set of random samples which are updated and propagated. Update stage of such algorithm uses Bayes rule, however the sampling strategy implies that the state

space grid is not necessary the samples are localized in regions of high probability density [19]. Arulampalam et al. review Bayesian algorithms for nonlinear or non Gaussian problems. The emphasis of the review is on Particle Filters (PF), their features, variants and, finally, inevitable comparison with the standard EKF [1]. Doucet in his book on Monte Carlo methods [7] focuses on creating an broad summary of theory and various applications of bootstrap filters, optimal Monte Carlo filters and Particle Filters. Common feature of both “Unscented” and “Monte Carlo Method” estimation techniques is the sampling phenomenon. By using sampling, linearisation of plant and observation models is avoided, hence the cause of approximation error that existed in EKF-based methods is cancelled this way. PF can handle non Gaussian and nonlinear processes, particularly exhibited in AUV models. Moreover, PF does not need to have the initial information about the state. Sampling techniques, particularly PF, have been recently and increasingly applied as tool for navigation of an underwater vehicle.

Karlsson et al. study a sea navigation method that relies on the underwater maps (depth map) and sonar measurements that support the navigation system [27]. Particle Filter is used for state estimation. Since the problem of underwater navigation using depth map is nonlinear, sequential Monte Carlo methods are used, therefore state probability density is approximated with set of particles where each particle has a location and weight assigned to it. Both values reflect the value of the density of the region in the state space [27]. Hence, instead of updating mean and covariance of the state, particle location and the weight of each particle are updated with each observation using sampling importance resampling (SIR) algorithm [27], [19]. Prior to navigation, terrain map (reference) was created using sonar depth measurements together with the Differential GPS measurements and the obtained grid was used for navigation. Moreover, the usage of Cramér-Rao bound was investigated in tasks such as INS system design, sensor performance or even the amount of control of that is needed for the navigation. This work presents a successful application of particle filtering for underwater navigation.

Above-mentioned work of Di Massa [6] presents navigation guided by the depth measured with bathymetric sonar. Map-matching with digital bathymetric map stored on-board has been accomplished using the Probabilistic Data Association Filter (PDAF) - a recursive algorithm similar to Kalman filter, designed for one target of interest and several measurements of the target state available each time step [6]. Position within the bathymetric map was stored as the state vector that was filtered. Results proved that such navigation is possible and that having a more diverse sea floor leads to more accurate navigation.

Maurelli et al. [30] propose a particle filter based underwater vehicle localization method for both structured and unstructured environment. Mechanically scanned profiling sonar is used as sensory device which provides information on distances from surrounding objects in environment. There is no information about initial position and orientation of the vehicle prior to localization. The work explores the possibility of dealing with dynamical situations when

carrying out the localization results in movements of the vehicle that contribute in improving the particle filter algorithm output - active localisation. Improvements of the PF algorithm concern computational efficiency and effective way of treating state space in order to recover from wrong convergence when using PFs. Simulation and real experimental results are available.

### 3.11.3 GPS aided localisation

Caiti explored localization technique that uses floating acoustic buoys provided with GPS connection [4]. Idea is that buoys supply the vehicle with their GPS location by emitting the information at regular time intervals. This way, vehicle can calculate the time of flight of the acoustic signal and locate itself with respect to the buoys. In such constellation, additional equipment has to be installed and maintained. Furthermore, acoustic signals are not reliable, their range is limited and signals not always available.

Erol [9] proposes a method for localization of the network of underwater sensors using single AUV as aiding device. This is just one of the examples of utilization of knowledge on AUV location. The aim is to use it to maintain localization of group of other objects in the water such as acoustic sensors. It is a system where AUV initially and occasionally receives GPS signals while being on the surface. Once the GPS location is received, vehicle dives to a certain depth and follows the defined path in between the sensor network. Set of freely deployed acoustic sensors is receiving messages containing coordinates from the vehicle, since the vehicle maintains updating its position using dead reckoning combined with occasional GPS correction. Emphasis is on algorithms for distance estimation so that proper values for sensor coordinates can be passed on to sensor network localization algorithm.

### 3.11.4 Localisation using *a-priori* map

Eustice experiments with re-navigation for AUVs [10]. The aim is to use a-priori given, ship derived bathymetric maps to reduce dead reckoning drift by comparing ship-derived depth map with the depth map created by vehicle. The difference between them is used as correction, a tool for removing long-term drift.

Williams is presenting a new method that uses terrain features for aiding the tracking of underwater vehicles in unstructured environments [43]. Benefit of such research lies in creating a vehicle capable of adopting to terrain changes therefore capable of being reliably deployed for longer time deep underwater, on a real task, with real environment. This work is revolutionary in solving the position update for a vehicle. Most common solution is the usage of acoustic transponders and triangulation algorithm as already exposed in Chapter § 3. Williams uses a priori elevation maps of the sea floor, recorded by ships. Depth information obtained from such mapping is assisting the localization process. Localization uses Monte Carlo methods,

particularly PF, to manage map-based localization with position and velocity kept within state vector. Non-Gaussian estimates obtained using particles are bounded using depth and altitude observations by using range information to rule out less probable particles. Update is accomplished by resampling the particle distribution with respect to likelihood that the observation detected is received, given the sample of the state space [43]. Apart from dealing with non-Gaussian estimates, advantages of such method include ability to track more than one possible target placement, which proves to be useful feature when handling map-based information. Newman and Durrant-Whyte propose a navigation filter which uses inertial measurement unit (IMU) and sonar to carry out the terrain-aided navigation [32]. IMU is used to measure the kinematic parameters such as accelerations and rotations with respect to body frame, while sonar measures absolute distances and orientations with respect to the world frame. In such configuration, sonar system does the correction of the noise-corrupted, drift-prone position indicated by IMU. Standard Kalman filter is used to integrate different sensor measurements with a-priori dynamics model of the vehicle. Sonar observations are processed with target extraction algorithm isolating terrain beacons. Map is simultaneously created during the vehicle mission. It map serves as the representation of the world and contains distinctive features from environment. Map is intended to be a sparse world representation, containing least possible but still enough information (features) to make a reference to and it is derived from estimation of the floor gradient using sonar measurements. Pose observation is defined relative to features from the map that correspond to features from the real world.

Similarly, Williams et al. present the results of operational algorithm that processes sonar scans in order to extract robust features of the environment further used to build an environment map consisting of those features [44]. The vehicle location is estimated within the built environment map. The work is an example of practical application of SLAM (Section § 3.8).

Another contribution of Eustice [14], based on vehicle localization with simultaneous terrain-features map creation (SLAM), reports usage of delayed state model. Robot movement is observed with respect to its previous position. Benefit of the delayed state framework lies in having precisely sparse information matrix. Main idea of the work is that, unlike previous algorithms, sparseness of the SLAM information matrix used in localization is guaranteed, not approximated.

### 3.11.5 Optical sensing

Tena Ruiz investigates terrain aided localization of an AUV from the perspective of Simultaneous Localization and Mapping (SLAM) [37]. Sonar device is used to sense the environment and its readings are used to find targets located not far from the vehicle. SLAM algorithm uses detected targets together with a vehicle model to simultaneously construct the environment

map and localize the vehicle using filtering. Multiple Hypothesis Tracking Filter is adjusted to the SLAM framework. Necessary step of matching the sonar images with environment targets was accomplished by extracting the features and associating them with the sonar recordings by calculating a score that expresses the probability of a certain target causing a certain sonar output.

Williams presents the results of sonar usage together with the vision based system in SLAM algorithm for terrain-aided navigation [42]. Seabed covered with visually distinctive textured reefs was used in estimating vehicle motion and creating a map of the reef structure.

Eustice proposes several works on vision based localisation for AUVs in an unstructured undersea environment. Framework presented in [11] blends together sensor data and camera measurements in order to determine the relative pose. The final aim is to concurrently determine the vehicle position and the past trajectory [11]. Augmented state Kalman filter is used to filter out the pose of the vehicle. History of poses defines vehicle trajectory. Registered images obtained with camera provide spatial constraints for the position hence providing the necessary feedback for pose estimation. Moreover, the registration of the image pairs is more robust compared with the situation when only camera is used for sensing. The other vision based SLAM approach [12] addresses the problem of localization within large areas. Precise localization is a prerequisite for high resolution underwater imaging of large objects placed on the sea-bed [12]. Precise navigation would enable decent coverage of the spacious site of interest which is mission task. Proposed solution uses a vision-based SLAM approach together with vehicle's inertial sensors' measurements. The originality of the paper is that it suggests a genuine solution for evaluating covariance bounds within the EKF used for filtering SLAM information.

Carreras [5] experiments with the usage of visual information for underwater robot localisation within the pool with clear visibility. This method is not intended for ocean environment, however, it is useful from the computer-vision side and potentially a tool to aid video mosaicking algorithms. Camera is mounted on the vehicle and used to simultaneously record the bottom of the tank covered with the coded pattern. This way, structured environment is used for, map-based 3D localisation of the vehicle. Number of computer-vision related matters are examined, such as: necessary camera calibration, filtering methods, landmark detection. Localisation performance is reported to be high, with high computation rate and, importantly, prone to drift. Hence, reliable position information is good enough to derive the velocity estimates from it. In addition, work contains the analysis of the source of localisation errors and real time application examples.

### 3.11.6 Simultaneous location and mapping

In order to deal with SLAM issues, several methodologies are covered in literature. View-based approach generally relies on registering different types of imagery: optical imagery or depth imagery in order to establish spatial constraints that are vision-based, therefore, not prone to drift error ([10, 18, 36]). This approach does not include precise representation of features. SLAM frameworks that, however, use features (feature-based) instead of view are reported in ([37, 42]).

### 3.11.7 Cooperative navigation

The dissertation of Bahr ([2]) focuses on cooperative navigation, and furthermore, general background on localisation is surveyed. Bahr proposes a probabilistic-based algorithm, particularly tailored to the underwater environment. Cooperation in navigation is already available in air or the surface of the Earth. The work focuses on cooperative localization where different vehicles, arranged in group, communicate between each other. To accomplish cooperative localization, algorithm estimates inter vehicle range, resulting in position and the uncertainty of the position, then exchanges that information with other vehicles. Main idea is to exchange localisation-related information. The whole process consists of data acquisition and data processing stage. Forward propagation of position information causes more than one solution available. Further data processing gives an estimate of the location by processing all the possibilities. Stated advantages of such approach are that, apart from having more than one vehicle, no additional infrastructure is necessary [2]. Everything comes down to the usual sensor and communication package already available on vehicles [2].

Fallon et al. [15] describe the implementation of the cooperative localisation algorithm that works within group of AUV vehicles, aided with the position information from the autonomous surface vehicle to bound the estimation errors. Instead of the usage of expensive navigation sensors for communicating under the water surface, an approach where cooperation with one surface vehicle was suggested. Surface vehicle is intended to contribute with accurate position information. One of the contributions of the work is a trial of EKF localisation run on AUV and supported by the surface vehicle moving along with the AUV. Work investigates the usage of different estimators - Particle Filters, non-linear Least-Squares Optimization (NLS) and EKF in cooperation localisation within a mission consisting of two AUVs and one surface vehicle. It ends up comparing performance of each algorithm with the final outcome giving more credits to PF and NLS.



## Chapter 4

# Navigation sensors

This chapter gives an overview of the sensors used for localisation of an underwater vehicle and their characteristics. Underwater positioning can be based on usage of different types of sensors combined together in one system. The role of these sensors is to measure absolute position, velocities and heading/orientation. Localization is influenced with the development and performance of sensor devices. Sensors can be regarded as the means for managing the localization. Faster they are, more accurate they are, localization has more chances to perform better. Each sensor has its own reference system in which it operates. It is important to say that sensors output measurements with reference either in body frame (Figure 5.1(b)) - the one fixed to the object or in global frame (Figure 5.1(a)). Basic navigation sensor set for a high-end AUV usually consists of depth sensor, magnetic compass, GPS device, LBL acoustic device, Doppler Velocity Log (DVL) and gyroscope, particularly fibre-optic gyroscope (FOG).

### 4.1 Inertial navigation system (INS)

Inertial navigation combines measurements of accelerometers and gyroscopes to track down the position and orientation of an object. Motion and rotation information obtained this way are processed in order to provide an estimate of objects location with respect to initial reference. Recent INS configurations tend to consist of compact, more accurate and higher performance sensor devices.

#### 4.1.1 Gyroscope

Gyroscope is an INS sensor device that essentially measures orientation of the device. Although classic gyroscope measures orientation, modern gyroscopes are capable of measuring the angu-

lar rate. Gyroscopes can be mechanical, optical or MEMS gyroscopes. Underwater navigation uses fibre-optic gyroscope (FOG). FOG is based on measuring the interference of two light beams that pass through a coiled optical fibre in both directions. FOG provides quite precise information on rotation and its usage is intended for applications that demand higher performance. FOG can provide the angular information: rate of change of heading (yaw rate) and the yaw/heading itself. Both measurements can be used, depending on configuration. Since the sensor actually measures yaw rate (absolute measurement), yaw can be derived by integrating yaw rate in time. Disadvantage of such principally precise measurement, is that it is relative to previous yaw value and if that value turned out to be imprecise, drift can increase or contain a permanent bias. The KVH DSP-3000 device provides the Nessie vehicle with accurate angular rates.

#### 4.1.2 Doppler Velocity Log (DVL)

DVL is intended to measure velocities. Transceiver components mounted on the device, pointing downwards (towards the bottom) emit acoustic impulses which are expected to be reflected if the DVL is close enough to the bottom. In case of existing reflectance it is called having “bottom-lock”. DVL usually has four transceivers. Each of them makes an angle with respect to the sea floor. If “bottom-locked”, those four sensors undergo Doppler shift effect. Besides Doppler speeds, DVL measures roll, pitch and heading angles and fuses them together when computing surge, sway and heave velocity within the 3D speed vector in world referenced frame. The Teledyne Explorer PA DVL is used on Nessie AUV. This unit supplies the vehicle with altitude, surge, sway and heave velocities. Operation altitude ranges from 0.5 m to 80 m. Accuracy of the measured speeds is 0.7 cm/s when moving at the speed of 1 m/s.

#### 4.1.3 Magnetic compass

Magnetic compass provides 3D vector of local magnetic field. Magnetic compass points at magnetic north. Direction is determined so that it aligns with Earth’s magnetic field. Important procedure before starting the compass usage is calibration. North direction as it appears on maps points to the geographic north (“true north”). That is the direction towards the rotation axis of the Earth. The direction of the *magnetic north* does not overlap with the direction of the geographic north. Magnetic declination is an angle between magnetic north (measured by compass) direction and the true north direction (the one that maps refer to). Depending on location where the compass is used, magnetic declination can vary and the variation is different on different spots on the Earth’s surface. Hence the calibration is necessary before usage. Magnetic compass gives measurements with slow variation in space. In addition, different magnetic effects caused by electric currents can affect the measurement, thus it is robust absolute mea-

surement of heading, nonetheless, prone to noise. Nessie vehicle uses TCM 2.6 compass that precisely measures heading, pitch, and roll. Accuracy of the heading measurement is  $0.8^\circ$ .

#### 4.1.4 Depth sensor

Bathymetry system accomplishes depth measurement. It is possible to use the acoustic system for this purpose, however, bathymeter using pressure information tends to be more precise and trustable. Pressure sensor is standard piece of the equipment for an AUV. By measuring the pressure, it becomes possible to correlate the value of pressure with the value of depth. Device can frequently ascertain the absolute depth with good precision. A Keller Series 33X depth sensor measures the distance to surface having the depth range of 0 - 90 m and the accuracy of 0.05 m.

#### 4.1.5 Global Positioning System (GPS)

This well known satellite-based navigation system provides position information anywhere on the Earth surface or in the air, reasonably close to the surface. Due to absorption of electromagnetic waves in the water GPS signal is not available underwater. Despite the fact that GPS is not available, vehicles are equipped with GPS receiver intended to be used for initial position information before submerging or for occasional position updates if the vehicle temporarily goes back to the surface. Precision of the GPS position information can vary. Namely, range of the standard deviation errors can have the order of 25.27 m [16]. Such huge deviation can cause significant inaccuracies in navigation. An example of the influence of GPS imprecision was shown in Figure 6.6. Differential GPS (DGPS) [16] can be used in situations when better precision is needed.

## 4.2 Acoustic positioning system

Provides the absolute position, a ground-based reference. Principal way of exchanging the information through the environment is sound wave. Long baseline (LBL) is used for measuring position with respect to several tethered beacons with known position, placed in water (Section § 3.6). It can be understood as the extension of the GPS information below the water surface. Such system employs acoustic signals to measure the distances. Vehicle uses the acoustic transponder to send the acoustic wave (“pinging”). The wave reaches beacon and reflects back to the vehicle. LBL system consists of transceiver and array-arranged collection of beacons. LBL transceiver pings each of the beacons and detects the signal travel time in order to calculate the distance, knowing the speed of sound in water. Distances from the beacons are combined

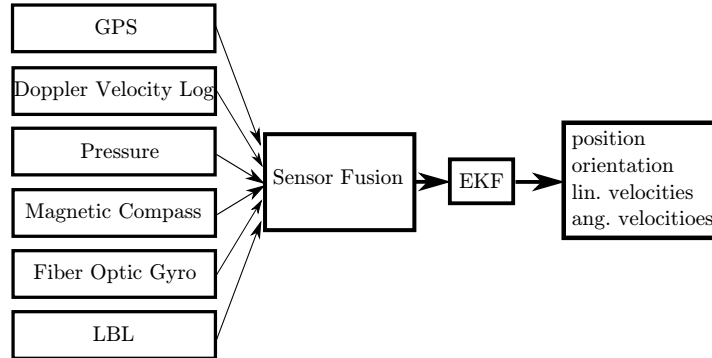


Figure 4.1: Sensor fusion diagram.

Table 4.1: Navigation sensors characteristics.

Sensor	Measures	Update rate	Precision	Accuracy	Range
Pressure	depth	10 Hz	0.0002 <i>bar</i> (0.002 <i>m</i> )	0.005 <i>bar</i> (0.05 <i>m</i> )	0 – 10 <i>bar</i> (0-90 <i>m</i> in water)
	✗ heave velocity is calculated by deriving depth in time - higher possibility of error ✓ distance from surface available no matter of distance from the seabed				
Compass	yaw(heading)		0.1°	0.8°	
	pitch	10 Hz			tilt ±50°
	roll				
✓ absolute measure of heading - no drift ✗ needs magnetic north correction (due to magnetic declination) ✗ prone to magnetic disturbance					
FOG	yaw rate	5Hz	< 1°/hr	±20°/hr	±375°/s
✓ high accuracy in heading measurement, compared to compass ✗ drifts over time, needs correction for the Earth rotation					
DVL	surge velocity			±0.7 $\frac{cm}{s}$ at 1 $\frac{m}{s^4}$	
	sway velocity	10 Hz	0.1 $\frac{cm}{s}$	±1.9 $\frac{cm}{s}$ at 3 $\frac{m}{s^4}$	±9.5 $\frac{m}{s}$
	heave velocity			±3.0 $\frac{cm}{s}$ at 5 $\frac{m}{s^4}$	
✗ relative measurement of velocity ✗ requires depth - loses lock at 0.6 <i>m</i> altitude resulting in no output					

together in triangulation method. Triangulation makes it possible to determine the position of the robot in the network of fixed beacons. LBL position update can introduce the outliers that need to be rejected or filtered.

## Chapter 5

# Methodology

Position, orientation and velocities of a vehicle underwater are stored within the state vector. Proposed solution for localization uses state-space approach and Extended Kalman Filter to estimate the value of the state vector using data from odometry sensors and acoustic positioning system (LBL), if available. Data are merged together using the EKF. Reasons for choosing this method are influenced by the application itself. Localisation is intended to work in unstructured environments, with no clear visibility, relying on kinetic and absolute position measurements. Therefore, approaches that use vision, or capture the terrain structure in order to aid the navigation were not an adequate solution. Relying on the remaining sensor configuration allows the usage of Kalman filter to iteratively and recursively estimate the state vector by combining together different types of dynamics-related data. State-space approach is convenient for manipulation with multivariate data and nonlinear or non Gaussian processes [35].

Mathematical model of the system is the integral part of the Kalman filter. It is used to define the state transition law by applying well known kinematic equations. Kinematic equations describe the object motion. *Constant velocity* kinematic model is used as system model to predict the movements of the submerged body. States are predicted at each time-step using the model and previous state (equation 5.1). It is important to note that localization reduces down to solving non-linear filtering problem. System model equations are nonlinear function of the state elements. Sensor measurements are used when available and incorporated into the observation (measurement) in order to update and correct the predicted filter state. Range of available sensors give information on vehicle's current location state. Overview of the sensors and the values that they measure is given in Chapter § 4.

## 5.1 Process model

Five d.o.f. process model is used to describe the state transition in time. In proposed discrete-time stochastic model, five degrees of freedom include position values and two angle states: yaw and pitch - making altogether five possible values to change in modelling vehicle position (figure 5.1(a)). Since the application uses state-space approach, focus will be on defining a state vector that would incorporate all the relevant values for the dynamic system that does the localisation - kinematic and position variables. In spirit of that, system state vector combines together metric and angular values. At discrete time moment  $k$ , it values:

$$\mathbf{X}(k) = \left[ x \quad y \quad z \quad a \quad u \quad v \quad w \quad \psi \quad \varphi \quad \dot{\psi} \quad \dot{\varphi} \right]^T$$

where  $x$  takes the value of *north* (expressed in meters),  $y$  is *east* and  $z$  is *depth*. Inspiration for the nomenclature was taken from [34].  $a$  marks the *altitude* with  $u$ ,  $v$  and  $w$  standing for linear velocities: *surge velocity*, *sway velocity* and *heave velocity*, respectfully. The rest of the state vector covers angular values (expressed in radians or degrees).  $\psi$  and  $\varphi$  are used as yaw and pitch, hence describing the vehicle orientation.  $\dot{\psi}$  and  $\dot{\varphi}$  are angular velocities: yaw rate and pitch rate, respectfully. The state vector incorporates all the relevant information necessary to describe the system under investigation. Angle and velocity for pitch degree of freedom is included in 5 d.o.f. system model since it can make a difference in estimating vehicle location in certain movement scenarios (figure 5.1(c)). System model is describing how the state  $\mathbf{X}$  evolves in time. It is defined as a constant speed model that uses previous state and noise to make a prediction on the next state vector value  $\mathbf{X}(k)$  using non-linear function  $f()$  and process noise vector  $\mathbf{n}$  (equation 5.1) where  $\mathbf{N} = \left[ \dot{u} \quad \dot{v} \quad \dot{w} \quad \ddot{\psi} \quad \ddot{\varphi} \right]^T$ . Process noise models inaccuracies or unpredictable disturbances in motion model [35].

$$\mathbf{X}(k) = f(\mathbf{X}(k-1), \mathbf{N}(k-1)) \quad (5.1)$$

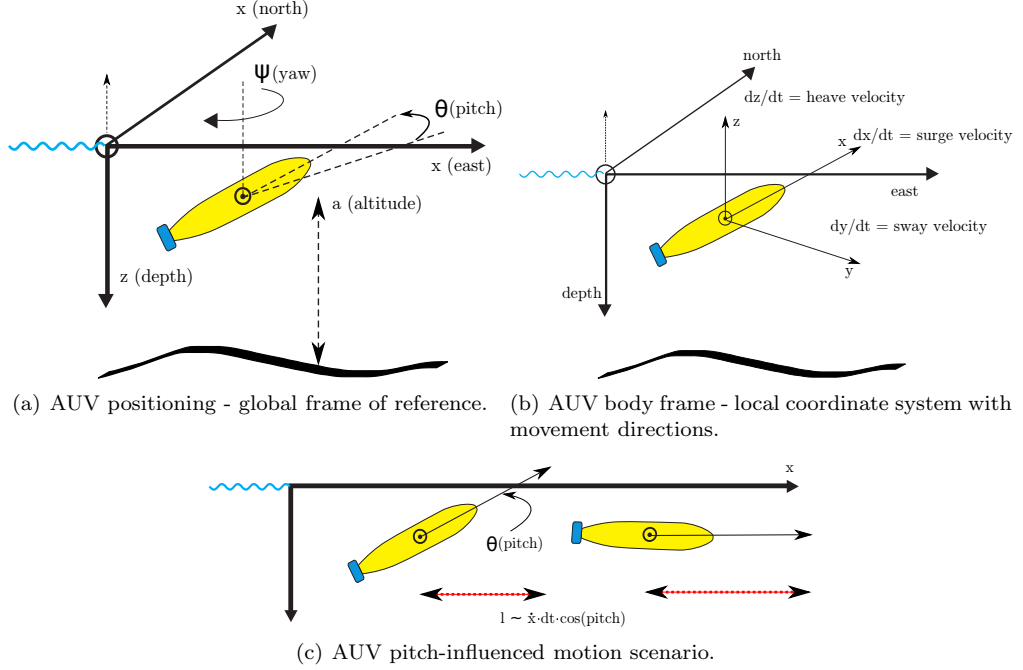


Figure 5.1: AUV state vector values and five degrees of freedom.

$$\begin{bmatrix} x \\ y \\ z \\ a \\ u \\ v \\ w \\ \psi \\ \varphi \\ \dot{\psi} \\ \dot{\varphi} \end{bmatrix}_{(k)} = \begin{bmatrix} x + (uT + \dot{u}\frac{T^2}{2}) \cos(\psi) \cos(\varphi) - (vT + \dot{v}\frac{T^2}{2}) \sin(\psi) \cos(\varphi) \\ y + (uT + \dot{u}\frac{T^2}{2}) \sin(\psi) \cos(\varphi) + (vT + \dot{v}\frac{T^2}{2}) \cos(\psi) \cos(\varphi) \\ z + (wT + \dot{w}\frac{T^2}{2}) \cos(\varphi) \\ a - (wT + \dot{w}\frac{T^2}{2}) \sin(\varphi) \\ u + \dot{u}T \\ v + \dot{v}T \\ w + \dot{w}T \\ \psi + \dot{\psi}T + \ddot{\psi}\frac{T^2}{2} \\ \varphi + \dot{\varphi}T + \ddot{\varphi}\frac{T^2}{2} \\ \dot{\psi} + \ddot{\psi}T \\ \dot{\varphi} + \ddot{\varphi}T \end{bmatrix}_{(k-1)} \quad (5.2)$$

To summarize, implementing vehicle localization using EKF demands establishing two models: first one describing the state evolution (system model) and the second model that associates noisy measurement with the state (measurement model).

EKF (§ 3.11.1) was chosen for the state estimation as a logic choice being an algorithm that integrates together different sensor measurements, makes a sub-optimal, recursive state

estimation and above all, is derived for nonlinear systems. As the theory of nonlinear filtering suggests, state-space approach is used for modelling discrete-time dynamic system in this application. The main feature of EKF is that it linearises the system model and measurement model nonlinear functions. System model is further developed according to formulas 3.2, 3.3, 3.4 and 3.5 from Chapter on Nonlinear filtering § 5, resulting in matrices:

$$\mathbf{F}(k) = \frac{\partial f}{\partial X}(\hat{\mathbf{X}}(k | k-1), 0) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & Tc(\psi)c(\varphi) & -Ts(\psi)c(\varphi) & 0 & -Tc(\varphi)(us(\psi) + vc(\psi)) & -Ts(\varphi)(uc(\psi) - vs(\psi)) & 0 & 0 \\ 0 & 1 & 0 & 0 & Ts(\psi)c(\varphi) & Tc(\psi)c(\varphi) & 0 & Tc(\varphi)(uc(\psi) - vs(\psi)) & -Ts(\varphi)(us(\psi) - vc(\psi)) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & Tc(\varphi) & 0 & -vTs(\varphi) & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -Tc(\varphi) & 0 & vTs(\varphi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where  $c(\psi)$  and  $c(\varphi)$  stand for  $\cos(\psi)$  and  $\cos(\varphi)$  respectfully. Similarly,  $s(\psi)$  and  $s(\varphi)$  mark the angle sinuses.

$$\mathbf{W}(k) = \frac{\partial f}{\partial N}(\hat{\mathbf{X}}(k | k-1), 0) =$$

$$\begin{bmatrix} \frac{T^2}{2} \cos(\psi) \cos(\varphi) & -\frac{T^2}{2} \sin(\psi) \cos(\varphi) & 0 & 0 & 0 \\ \frac{T^2}{2} \sin(\psi) \cos(\varphi) & \frac{T^2}{2} \cos(\psi) \cos(\varphi) & 0 & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} \cos(\varphi) & 0 & 0 \\ 0 & 0 & -\frac{T^2}{2} \cos(\varphi) & 0 & 0 \\ T & 0 & 0 & 0 & 0 \\ 0 & T & 0 & 0 & 0 \\ 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & T \end{bmatrix}$$

Knowing plant model and deriving  $\mathbf{F}(k)$ ,  $\mathbf{W}(k)$  enables EKF algorithm to complete the prediction stage using known formulas. Next step is the correction of the prediction using data obtained from measurement.



## 5.2 Measurement model

Measurement model introduces measurement equation which establishes the connection between the measurements and the target state (equation 5.3) where  $Z(k)$  represents the measurement at time  $k$ ,  $X(k)$  represents state vector and  $M(k)$  represents noise. Purpose of the measurement is to be able to update, correct the state  $X(k)$  using measurements  $Z(k)$ .  $h()$  is generally a non-linear function. EKF idea is to linearise the measurement model.

$$\begin{aligned}\mathbf{H}(k) &= \frac{\partial h}{\partial \mathbf{X}}(\hat{\mathbf{X}}(k | k-1), 0) \\ \mathbf{R}(k) &= \frac{\partial h}{\partial \mathbf{M}}(\hat{\mathbf{X}}(k | k-1), 0)\end{aligned}$$

However, for this particular application and available sensor configuration, state vector elements are measured directly, hence  $h()$  can be expressed with matrix containing ‘‘ones’’ at particular positions since the measurement relation becomes equality. There is no need for partial derivation. Measurement noise is submitted in form of an additive Gaussian zero-mean noise assigned to each measured value. Measurement (observation) noise is characterised with zero mean ( $E\{\mathbf{M}(k)\} = 0$ ) and standard deviation ( $E\{\mathbf{M}(k)\mathbf{M}^T(k)\}$ ) given as filter parameter for each of the measured values. It expresses how uncertain or varying measurement of each of the state values is. The measurement in vehicle configuration used for the thesis enables direct measurement of each of the states and each measurement of one of the state elements is described with its uncertainty - variance of the Gaussian noise. Variances are given as filter parameters expressing how much we trust in the measurement.

$$\mathbf{Z}(k) = h(\mathbf{X}(k), \mathbf{M}(k)) = \mathbf{H}\mathbf{X}(k | k-1) + \mathbf{M}(k) \quad (5.3)$$

One of the features of the process is that measurements are not available all the time. The reason is the nature of the process of estimating the location itself. Simply - messages from sensors arrive at different moments and it happens that some of the sensors could not be available due to different causes. The idea is to take all the available information at the moment of filtering and integrate it together in measurement model, as a filter observation. Alternatively, each message can be filtered upon its arrival. As a result, filter would keep carrying out the prediction and correction blending together various measurements in a estimate of higher quality, with the ability to compensate the missing sensor measurements.

A short overview of the sensor measurements and the pattern of formation of the measurement model is presented in following section. Idea for the solution has been introduced in [34]. Some other implementations of EKF for underwater navigation have reported the usage of similar strategy for merging the measurements together [3, 8]. Sensors have been introduced with

more details in chapter 4. Linear observation model varies depending on the measured values and sensors used for the measurement.

- **Pressure sensor** periodically “feeds” the filter with depth and heave velocity information. Heave velocity information is derived from depth measurement by derivation of its value per elapsed time. Measurement model is linear, described with the transformation matrix  $\mathbf{H}_{depth}$ . Uncertainty of the measurement is expressed with measurement noise covariance matrix  $\mathbf{R}$  (§ 3.3) - covariance of the measurement vector.

$$\mathbf{H}_{depth}(k) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{R}_{depth}(k) = \begin{bmatrix} \sigma_z^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix} \quad (5.4)$$

- **Compass** supplies angular measurements such as pitch, yaw, pitch rate (figure 5.1(a)). Similarly as with pressure sensor measurement model, rate of change of pitch is obtained with derivation of pitch measurement. Measurement model is linear, described with the transformation matrix  $\mathbf{H}_{compass}$ .  $\mathbf{R}_{compass}$  is measurement noise matrix (§ 3.3) containing variances depicting Gaussian noise of particular measurements.

$$\mathbf{H}_{compass}(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{R}_{compass}(k) = \begin{bmatrix} \sigma_\varphi^2 & 0 & 0 \\ 0 & \sigma_{\dot{\varphi}}^2 & 0 \\ 0 & 0 & \sigma_\psi^2 \end{bmatrix} \quad (5.5)$$

- **Fiber-optic-gyroscope sensor (FOG)** measures yaw rate ( $\dot{\psi}$ ). FOG, gives an absolute measurement of the rate. In case of compass usage, yaw rate is determined by doing derivation of the measured heading in time. FOG turns out to be quite precise and fast sensor. Yaw rate can be used to determine the yaw (heading) itself by integrating the yaw rates over time. That would imply having relative measurement of yaw - result would append to the previous value. Initial absolute measurement is obtained from compass. In case compass made an error with the initial value, relative FOG measurements cannot correct it. In case it existed - it propagates. Disadvantage of the usage of FOG for yaw measurement is that the compass-obtained initial heading value is not always accurate. This can result in constant bias of the measured heading in case the initial heading was measured with an error. Error would propagate through relative measurements in form of a bias. Given that compass is prone to noise and often needs re-calibration, this scenario is possible. Measurement model matrices are:

$$\mathbf{H}_{fog}(k) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{R}_{fog}(k) = \begin{bmatrix} \sigma_{\dot{\psi}}^2 \end{bmatrix} \quad (5.6)$$

- **Doppler Velocity Log (DVL)** measures linear speeds - surge and sway velocities with respect to the vehicle axes (Figure 5.1(b)) and the altitude.  $\mathbf{H}_{dvl}$  and  $\mathbf{R}_{dvl}$  are measurement matrix and measurement noise covariance matrix.  $\mathbf{R}_{dvl}$  contains variances depicting Gaussian noises of particular measurements. If DVL gives no valid velocity output due to insufficient signal (“DVL lock” lost), EKF is using the available data available from other sensors to make an update or it just maintains the prediction until the fresh observation arrives. Uncertainty increases in case there is no observation.

$$\mathbf{H}_{dvl}(k) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{R}_{dvl}(k) = \begin{bmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_u^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{bmatrix} \quad (5.7)$$

- **Long-baseline (LBL)** gives an absolute position update based on GPS signal from surface as reference and triangulation obtained from transponders with known global position place around the vehicle (figure 3.2(a)). This serves as a crucial anti-drift tool because it is not relative to previous measurements. It results in a fresh update of global position in form of latitude and longitude that can be mapped into north and east of the absolute position in meters. Algorithmically, this can be regarded as an update of north and east causing matrices of measurement update to become:

$$\mathbf{H}_{lbl}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{R}_{lbl}(k) = \begin{bmatrix} \sigma_n^2 & 0 \\ 0 & \sigma_e^2 \end{bmatrix} \quad (5.8)$$

- **GPS** similarly as LBL, directly outputs a fairly accurate position measurement. GPS signal is not available underwater, hence, it is used at the beginning of the mission, when the vehicle is still on the surface, to establish the initial estimate. Algorithm keeps the track of all the GPS messages received, and in case the vehicle reaches the surface after deployment being able to receive GPS signal - that information is used to update the absolute position and correct the dead reckoning drift that happened meanwhile. Kalman filter matrices keep the same form as with the LBL, uncertainties can be set to different values.

Once having the measurement, it is important to know its strategy of integration. Kalman filtering is intended to work in two modes:

- **asynchronous** filtering messages from each sensor individually upon retrieval. Observation uses only measurements from each sensor individually. Every time sensor device produces a new output, all the localization variables are updated.

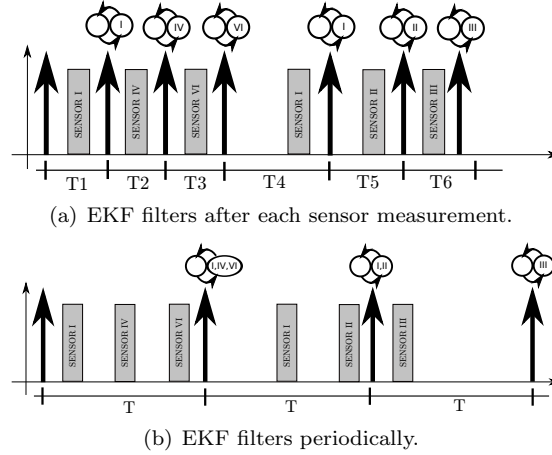


Figure 5.2: Two modes for combining together sensor measurements into observation.

- **synchronous** periodically updating with the most recent observation that combines together measurements from different sensors that occurred since the last update. User defined rate will determine the interval of the update.

Integrating together several different measurements into one observation would imply concatenating together several measurement matrices ( $\mathbf{Z}, \mathbf{H}$ ) and measurement noise matrices ( $\mathbf{R}$ ), as shown in 5.9 for the sample case of three sensor measurements included in observation. Kalman filter maintains its own timer. Observation is reset after each timer set period.

$$\mathbf{Z}(k) = \begin{bmatrix} \mathbf{Z}_{sensor I} \\ \mathbf{Z}_{sensor II} \\ \mathbf{Z}_{sensor III} \end{bmatrix}, \mathbf{H}(k) = \begin{bmatrix} \mathbf{H}_{sensor I} \\ \mathbf{H}_{sensor II} \\ \mathbf{H}_{sensor III} \end{bmatrix}, \mathbf{R}(k) = \begin{bmatrix} \mathbf{R}_{sensor I} & 0 & 0 \\ 0 & \mathbf{R}_{sensor II} & 0 \\ 0 & 0 & \mathbf{R}_{sensor III} \end{bmatrix} \quad (5.9)$$

Literature offers different approaches in managing Kalman filter update and managing observations. Matrix concatenation is used to gather together all the measurements into one observation. In case certain sensor measurement repeats during the update interval, the most recent measurement takes place of its preceding one within the periodic observation. This substitution is fairly realistic to happen, knowing that the user chooses filter update interval and that sensor devices have a range of frequencies, possibly more frequent that filter itself (table 4.1). Simply put, observation is a buffer that stores the most recent collection of different measurements that occurred in meanwhile. Depending on the mode of operation (asynchronous or synchronous) the observation can contain the measurement from an individual sensor or those sensors that were active during the predefined period  $T$  of update (Figure 5.2(b)). Choosing the newest

measurement is advantage from the point of having the most recent value as measurement, especially if the Kalman update interval is longer. However, it can also be a disadvantage in situations when certain sensor measurements happen at the beginning of the observation interval. For those “early” measured values, prediction stage will use less accurate time-stamp in EKF and the measurement itself would suit reality less, since it’s usage is delayed. This effect is visible if update periods of EKF are long, which is not so common case.

Having more than one measurement involved in estimation of the global state is a good characteristic. The estimate which uses more diverse data gives better estimate since it is possible to combine together more that one sort of observation. Another advantage follows the fact that the whole set of state variables is updated each time, resulting in more correlation between variables. Hence those that are missing for some reason can be compensated this way. Results of simulations using authentic data and the real missions are given in Chapter § 6.

### 5.3 Correction

Odometry integrates velocity and acceleration data collected from devices such as gyroscope or accelerometers. Integration of noisy data over time or usage of “relative measurements” (those calculated from absolute measurements) results in drift or bias of the final estimate. In order to recover from that, algorithms perform the correction. Correction takes an absolute measurement which should be less precise, possibly noisy, but not prone to drifting. An example which illustrates the phenomenon could be a man that walks with the eyes closed trying to keep the track of his position by measuring the steps and predicting where he could possibly be judging on number of steps and their size. Steps have the role of “relative measurement” - one quantity is used to estimate the other one that’s correlated with it. Naturally, a man will keep making errors in his position estimate. Moreover, these errors will accumulate over time producing drift. Correction would require the man to open his eyes and observe the current position (absolute measurement), compare it with the position predicted by counting steps and neutralise the drift before carrying on with the eyes closed.

### 5.4 Sensor fusion

Localisation algorithm collects the incoming sensor information and computes the pose of the vehicle by processing the whole data cluster obtained from sensor devices. Such procedure is regarded as *sensor fusion*. Basic sort of sensor fusion implementation is incorporated in navigation algorithm by combining different quantities into a jointly updated state vector with position, orientation and velocities (Figure 4.1).

## 5.5 Implementation

Localisation algorithm was planned to be part of navigation module of Ocean System Lab's Nessie vehicle. Vehicle's real-time operating system is Linux Ubuntu. Designing navigation requires programming the module using C++ programming language, conforming with available libraries. The aim is to produce an applicable module that fits in the existing software system and accomplishes the task theoretically explained in this chapter. In addition, Nessie is supported with Robot Operating System (ROS, <http://www.ros.org/wiki/>) - meta operating system that manages various processes maintained on vehicle and the real-time exchange of data between different modules. Hence, EKF navigation is implemented as a ROS package tailored to Nessie's messaging and processing scheme. One of the difficulties that needs to be taken into account when working with angles is angle wrapping - an inevitable issue in all the implementations that involve angular values. Fact is that filtering does calculations with the angular values. Moreover, angular functions are  $2\pi$  periodical. Hence, it is sufficient to represent the values in a limited interval, e.g. from  $(0, 2\pi]$ , or  $(-\pi, +\pi]$ . Angle values need to be scaled back to the interval after multiplication or subtraction because the value obtained without wrapping can be misleading, especially if it expresses discrepancy or if is used for the correction. Simply, raw subtraction does not always correctly express the relation between two angles in terms of their difference and can cause unwanted behaviour. An example is shown in Figure 5.3. Two similar angular signals (real and estimated heading value:  $ang_1$  and  $ang_2$ , wrapped in different intervals) have been subtracted (Figure 5.3, up). If the output of subtraction is used for control or gain, system would sense false jumps. Different intervals of angle wrapping (Figure 5.3, down) can make difference, too. Therefore angle wrapping to  $0, 2\pi$  and  $-\pi, +\pi$  does not give the same result.

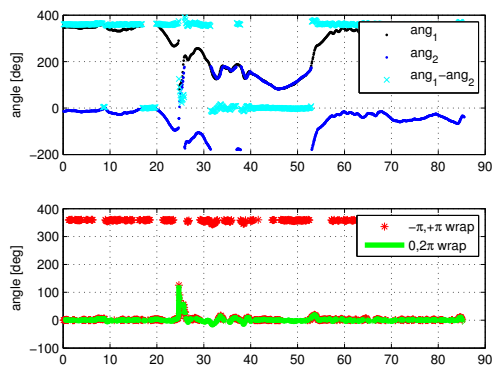


Figure 5.3: Importance of angle wrapping. Angular values used as gain should be wrapped around zero.

## Chapter 6

# Results

Main purpose of the thesis is the implementation of a navigation system that uses measurements from different sensors, fuses the sensory data together in order to make presumably better quality estimate of the position and the orientation of the underwater robot. Existing data from the real missions were used to carry out the initial trials. It is useful to mention that there is no exact ground truth for underwater robot localization available. GPS signal, if available, could serve as an absolute position reference: either directly or in form of LBL. Experimental results have been obtained for different missions. Good news, however, is that the absolute depth measurement is quite accurate and frequent, making AUV localisation a 2D task.

### 6.1 Real navigation scenario

Authentic data taken from previously recorded Nessie mission were used to simulate the algorithm “offline” as the part of the stage intended for testing and correcting. Besides, being able to repeat the same measurement scenario enables more insight in filtering process and benefits of fusing together the sensor data. *.bag* files (<http://www.ros.org/wiki/>) containing recorded real-time messages with sensor measurements, were used as source. Furthermore, it allows designing the code in its original C++ form that will require little modification once deployed on the vehicle in form of ROS package since *.bag* files emulate authentic messages and timestamps. One of the deficiencies of the evaluation of localisation results is the fact that there is no exact ground truth to compare the result with. Dead reckoning localisation substituted with occasional LBL position updates was compared with the localisation obtained after filtering (Figures 6.1, 6.2) for the recorded straight line trajectory mission.

Selecting a heading measurement with good performance For a high-end underwater vehicle such as Nessie, supplied with FOG-based INS, DVL and LBL, main source of navigation

error is influenced by transformation of vehicle-referenced velocities to world-referenced velocities [2], particularly due to yaw (heading) measurement errors. Yaw can be measured using several devices, each having different accuracy and performance. Simulation with data from previous missions was carried out to see which device gives the best performance for a given underwater vehicle.

Heading calculated by integrating FOG's yaw rate - tends to be accurate and fast, less prone to noise. Nevertheless, it is calculated each time by appending yaw rate value integrated in time on the previous yaw value (relative measurement). Therefore, it is sensible on initial absolute heading measurement. In case initial yaw is imprecise, a constant bias exists in yaw measurement (Figure 6.2). Constant bias causes sudden steps in position estimate obtained using EKF. This is expected scenario since the sensor responsible for measuring initial heading is magnetic compass, device sensitive to disturbances coming from environment (Figure 6.3(b)). In real experiments biggest obstacle was proper calibration of magnetic compass. In practice, tests have showed many failures in compass heading measurement, possibly due to calibration and magnetic declination. There is yet space to do more testing with better compass tuning. To overcome the problem of accurate yaw measurement, EKF used in experiments will either ignore the yaw measurement (it is possible since sensor fusion successfully compensates missing heading information with yaw rate obtained from FOG) or use it with high variance assigned to yaw measurement.

**Loch Earn dataset - straight line movement:** Example of basic EKF localisation using inertial measurements aided with LBL acoustic positioning system was tested on straight line movement of approximately 80 m length, recorded at the lake Loch Earn. For simulation purposes, sensor measurements are stored in a *.bag* file, that can be replayed, producing real-time messages of sensor measurements as they originally occurred. At this point, it is important to revise which sensors were used, their main features and, finally, filtering parameters.

Standard sensor configuration comprising of pressure sensor, magnetic compass, FOG and DVL is used in the mission. Absolute position correction was carried out using LBL system. Important fact is that the heading was measured with magnetic compass only at the beginning. Later on, it kept being calculated by integrating yaw rate obtained from FOG. Alternative solution for the heading measurement would be the usage of compass for direct acquiring of yaw, but such option exposed calibration difficulties. Result of EKF localisation algorithm was shown in north-east map (Figure 6.1(a), 6.2(a)). Different parameter values for EKF were tested. Table 6.1 revises all filter parameters used for filter tuning, together with their role. Essentially, setting high standard deviation for a Gaussian of a certain parameter can be interpreted as having more uncertainty in value that it represents - whether it is a measurement uncertainty or uncertainty of the predicted value (model uncertainty). Therefore, we can choose to be confident in certain sensor measurement and/or certain model prediction, and observe the



Table 6.1: EKF navigation parameters.

Parameter	Signature <sup>1</sup>	Units	Description
			standard deviation of the ...
SDNorth	$\sigma_n$	$m$	north observation
SDEast	$\sigma_e$	$m$	east observation
SDDepth	$\sigma_d$	$m$	depth observation
SDAltitude	$\sigma_a$	$m$	altitude observation
SDu	$\sigma_u$	$\frac{m}{s}$	surge velocity observation
SDv	$\sigma_v$	$\frac{m}{s}$	sway velocity observation
SDw	$\sigma_w$	$\frac{m}{s}$	heave velocity observation
SDyaw	$\sigma_\psi$	$rad$	heading observation
SDpitch	$\sigma_\varphi$	$rad$	pitch observation
SDyawRate	$\sigma_{\dot{\psi}}$	$\frac{rad}{s}$	heading rate observation
SDpitchRate	$\sigma_{\dot{\varphi}}$	$\frac{rad}{s}$	pitch rate observation
			standard deviation of the ... process noise
SDuModel	$\sigma_{\dot{u}}$	$\frac{m}{s^2}$	surge acceleration
SDvModel	$\sigma_{\dot{v}}$	$\frac{m}{s^2}$	sway acceleration
SDwModel	$\sigma_{\dot{w}}$	$\frac{m}{s^2}$	heave acceleration
SDyawRateModel	$\sigma_{\dot{\psi}}$	$\frac{rad}{s^2}$	yaw acceleration
SDpitchRateModel	$\sigma_{\dot{\varphi}}$	$\frac{rad}{s^2}$	pitch acceleration

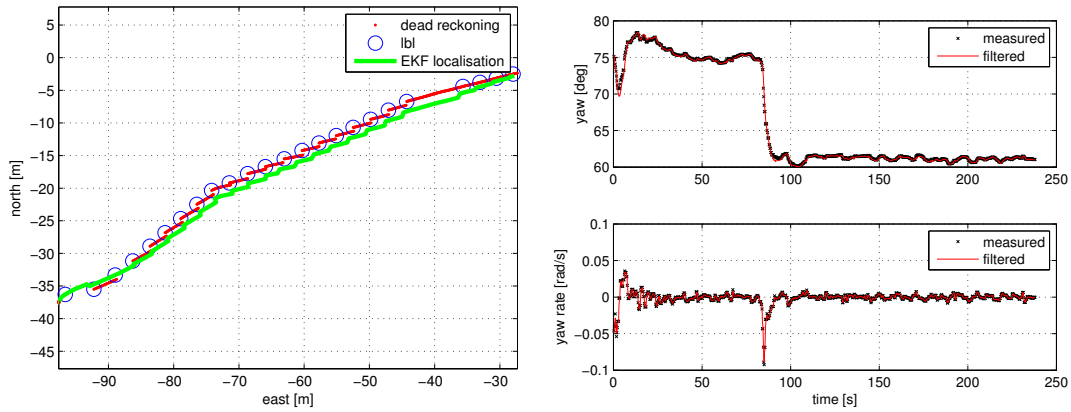
simulation outcome of such setting. Setting the parameters properly improves the performance of the filter. Straight line movement with authentic sensor measurements recorded in Loch Earn was a basis for initial tests of the EKF localisation algorithm. Red line shows the dead reckoning navigation, which is directly updated with absolute position update (LBL). Dead reckoning uses values periodically ( $\approx 5Hz$ ) obtained from DVL and FOG (linear velocities:  $u$  and  $v$  and heading  $\psi$ , respectfully), and substitutes them into equations similar to ones used for north and east prediction within prediction model:

$$north = north + (uT + \dot{u}\frac{T^2}{2}) \cos(\psi) - (vT + \dot{v}\frac{T^2}{2}) \sin(\psi)$$

$$east = east + (uT + \dot{u}\frac{T^2}{2}) \sin(\psi) + (vT + \dot{v}\frac{T^2}{2}) \cos(\psi)$$

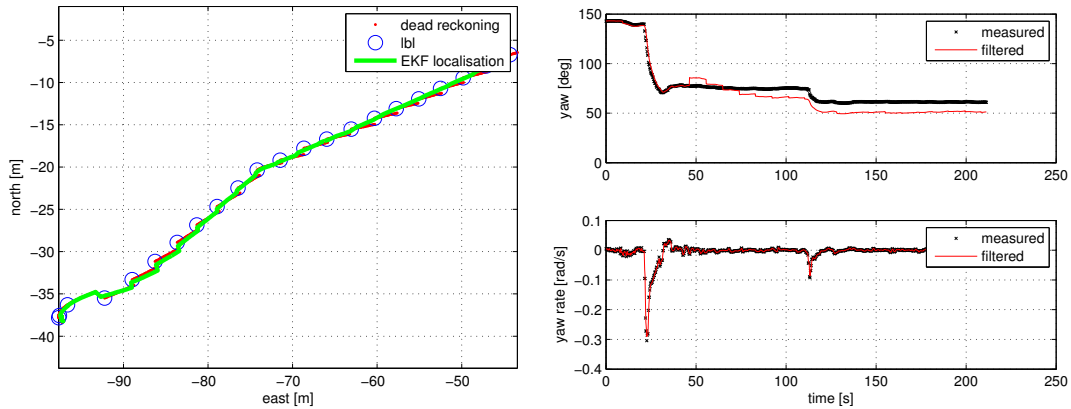
EKF updates periodically (synchronous mode), with period set to 230 ms. At first, simulation parameters  $SDnorth$ ,  $SDEast$ ,  $SDyaw$  and  $SDyawRate$  were set to low values - suggesting high trust in measurements. Resulting trajectory (Figure 6.1(a)) shows that the heading measurement has a constant bias, caused by the error in initial heading measurement obtained by compass. Thus, yaw measurement, calculated relative to previous value each time, propagates

<sup>1</sup>as it appears in algorithm equations



(a) N/E localisation. Yaw was calculated by integrating yaw rate periodically measured using FOG. (b) Heading estimation. Biased yaw measurement not being corrected due to high confidence in yaw measurement.

Figure 6.1: AUV localisation using EKF with high confidence in yaw measurement,  $SD_{yaw} = 0.01rad \approx 0.6^\circ$ .  $SD_{yawRate} = 0.004 \frac{rad}{s}$ ,  $SD_{u/v} = 1 \frac{cm}{s}$ .



(a) N/E localisation. Yaw was calculated by integrating yaw rate periodically measured using FOG. (b) Heading estimation. Biased yaw measurement is being corrected.

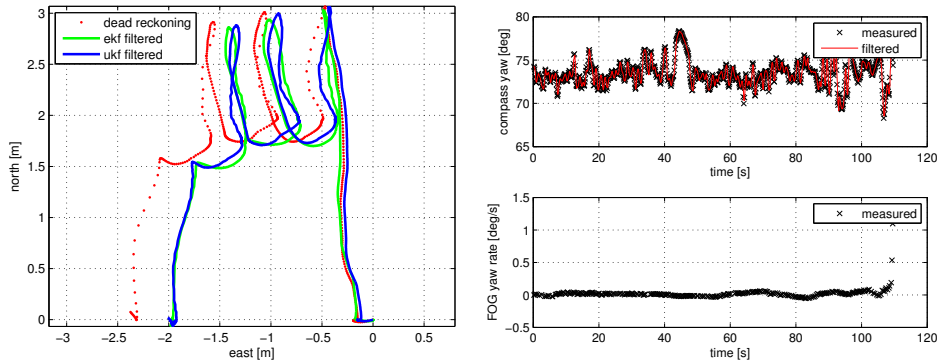
Figure 6.2: AUV localisation using EKF with low confidence in yaw measurement,  $SD_{yaw} = 0.2rad \approx 11.5^\circ$ .  $SD_{yawRate} = 0.004 \frac{rad}{s}$ ,  $SD_{u/v} = 1 \frac{cm}{s}$ .

the error (bias). Biased yaw observation further on causes EKF localisation to experience sharp jumps. To overcome this using EKF framework, less confidence was assigned to the yaw measurement ( $SD_{yaw}$ ) value. Eventually, bias becomes visible if measured and filtered heading are compared (Figure 6.2(b)). As for the rest of the heading information, rate of yaw measurement will be incorporated with a lot of confidence ( $SD_{yawRate}$  parameter having range of degrees) since it is a reliable device and it does not depend on the initial estimate. Good feature of sensor fusion is that lack of one measurement or its low performance can be compensated with some other measurement considering that they are combined together in mathematical model in the right manner. In case of yaw and yaw rate - the derivation in time is a relation that connects them together.

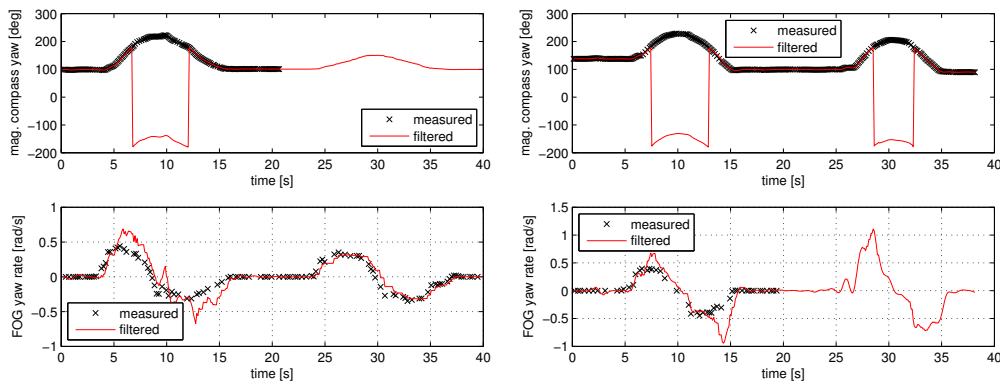
Simulation shows that localisation performance can be tailored by setting the confidence in prediction model or measurement values. Confidence is materialized as standard deviation (variance) of the random variable: the lower it is, more certain the value of the random variable is hence more confident in value of that variable we tend to be. Kalman filter tries to optimise the result within the defined boundaries of uncertainty.

Unscented Kalman filter was mentioned in Section § 3.11.1 as a good alternative in handling nonlinearities. UKF was implemented in MATLAB for the simulation purposes and its result compared with the EKF localisation, under same parameter settings and using the real data obtained from Nessie sensors. Test mission consisted of pipe tracking where the vehicle was guided along the underwater pipe three times and each time returned back to the initial position (Figure 6.3(a)). 2D north-east maps were compared, together with dead reckoning, same as one used for previous simulations. General characteristics of UKF are visible from the obtained shape of the UKF path (Figure 6.3(a)). Eventually, both filtered paths end up in approximately same position, having less drift than the dead reckoning. EKF does first order approximations, therefore, its path is slightly distorted compared with the UKF one, which was obtained with the same amount of calculation, and the inherited approximation of at least second order [24]. Simultaneously with filtering, UKF preserves the nonlinearity formula of prediction model better - its curves have shape closer to equation-based dead reckoning curves. Still, a question that is yet opened is whether we need to improve the approximation of the prediction model. Answering this question is a hard task without knowing the movement of the object and how much it actually matches the state prediction model. A difficulty with UKF implementation is that it involves calculation of covariance matrix square root which is a slightly more complex problem, solved with numerical methods. Nevertheless, it is possible that covariance matrix becomes singular which also depends on parameter  $k$  used for scaling (Section § 3.11.1).  $k$  was set to -0.5 for simulations shown.

**Sensor fusion for heading:** being in search for heading measurement less prone to initial error, and guided by simulation results, real scenarios were accomplished using magnetic



(a) N/E localisation. Comparison of the EKF and UKF. (b) Magnetic disturbances affecting the heading measurement by compass. Vehicle is not moving, FOG is not oscillating at the same time.



(c) Compass disabled.

(d) FOG disabled.

compass for heading measurement. Compass is more sensitive to magnetic disturbances (Figure 6.3(b)), slower than FOG but, importantly, gives an absolute measure. Therefore, it does not rely on previous measurements. An experiment was made by just manually rotating the robot horizontally while keeping the same position - changing its heading. Figures 6.3(d) and 6.3(c) show the performance of yaw filtering using EKF and the example of sensor fusion of compass and FOG. Namely, compass (Figure 6.3(c)) or FOG (Figure 6.3(d)) were disabled at one point during the experiment. When one of them stops working, the other one tries to compensate the failure.

**Trajectory filtering:** Spiral trajectory and surfacing action was taken with Nessie starting from the depth of around 12 m. EKF estimation results are shown in Figure 6.4 together with LBL position updates and dead reckoning starting from each position. Similarly as with previous plots, dead reckoning was shown together with LBL position updates. Filtered

trajectory does not experience severe jumps, and the curve seems to be smoother and less prone to drifting. Standard deviation of north and east measurement parameter (Table 6.1) was tested with different values, causing more or less confidence in LBL measurement hence shaping the filtered localisation curve.

LBL measurements exhibit quite diverse range of values. Causes of position correction errors are numerous: from “multipathing” outliers (Figure 6.3) till the imprecision inferred from the nature of volatile acoustic and GPS information. “Multipathing” causes outliers in position information as a result of false reflections for instance. Acoustic and GPS imprecision can be treated as Gaussian random variable.

It is likely that some of the LBL position updates deviate from the trajectory. Hence, a mechanism for rejecting the outliers was investigated. EKF was tested on raw LBL position updates. Intention is to manage the filtration of the “outliers” by using properly tuned EKF. Motivation to explore such possibility comes from two scenarios encountered in earlier missions. In such missions position was dead reckoned and LBL was used to assign each time a new value of north and east coordinate. LBL outliers were ruled out using a median filter applied on the last eleven position coordinates once the latest LBL exceeded the set threshold in position change. The missions showcased situations when:

- LBL rejection is carried out despite being a “false alarm” - Figure 6.4(a),
- rejection of the LBL is the right choice - Figure 6.4(b)

It is important to say that LBL position filtering was implemented in form of median filter. EKF was updated with raw LBL data instead of median filter. Rejecting an LBL measurement can turn out to be right (Figure 6.4(b)) as well as a wrong decision (Figure 6.4(a)). That is why EKF was suggested as an alternative. Examples of EKF’s performance are shown in both Figures 6.4(a) and 6.4(b). Solution is not as categoric as median filter. Moreover, it is more robust. By giving certain trust in LBL observation it always takes it into account. Median filter, on the other hand, can be too selective in being right or wrong. If it turns out that LBL positions do follow each other, EKF continues slowly following that direction. If the outliers are isolated, EKF successfully rules them out (Figure 6.4(b)).

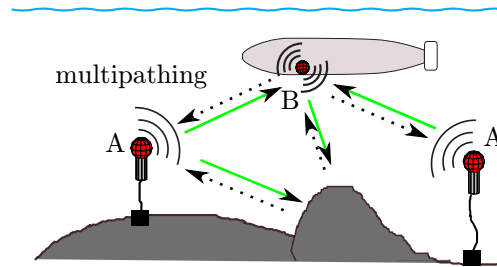
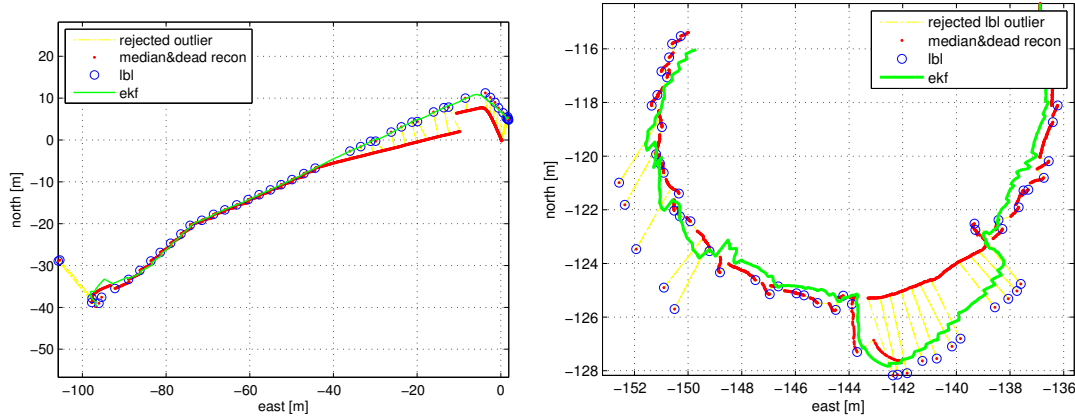


Figure 6.3: Multipathing can cause outliers in LBL position measurement. Due to reflection, several distances are detected, some being false measurements.

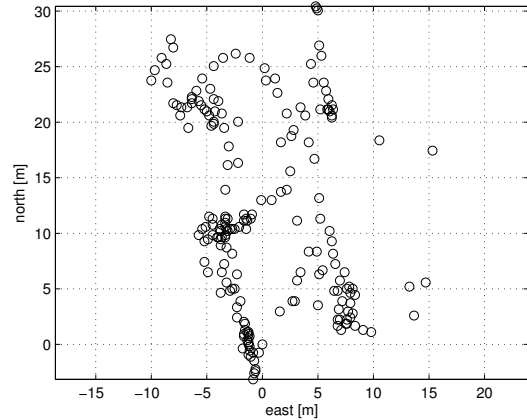
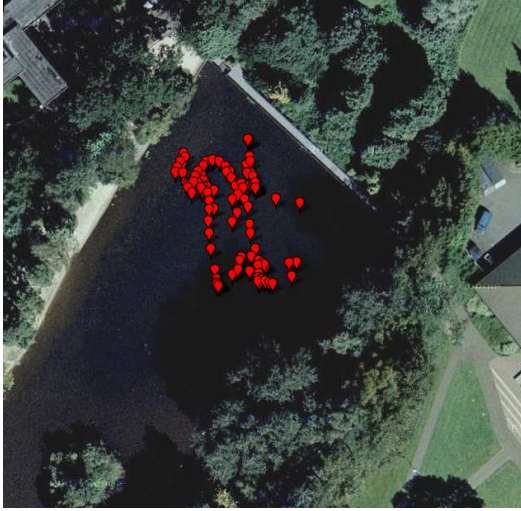


(a) Straight trajectory: LBL outliers erroneously rejected (red). EKF tends to recover the navigation (green). (b) Spiral trajectory: LBL outliers are rejected using median. EKF filtering introduces the position disturbance which recovers soon after.

**Square trajectories:** square trajectories were tested in low depths of a lake, with the GPS signal available to be used as a position reference and ground truth indication (Figures 6.4(c), 6.5 and 6.6). Dead reckoning navigation was used as a reference when controlling the vehicle movement during the experiment. This fact can cause slight confusion in analysis of the trajectory graphs since all the dynamics and forces were applied with respect to the dead reckoning navigation which is an estimated value, not the real existing one. It is a slightly inverse logic of testing, nevertheless further tests are yet to be accomplished. Emphasis of this experiment was to show that EKF can work successfully and analyse the main characteristics of the navigation design. It is likely that the GPS emulated square-shaped trajectories float as the elapsed path becomes longer. GPS signal available from the antenna located on the water surface is serving as a measure of absolute position within the lake - giving an idea about the actual vehicle position while it tries to moves within the boundaries of estimated dead reckoning position.

Main issue when performing the square trajectory tests was significant imprecision of GPS signal. Many reasons can possibly influence the imprecision: from the weather conditions till surrounding objects. Basically anything that can affect the satellite visibility and the quality of the signal. Drifting can reach up to several meters which is unacceptable considering the trajectory length. Finally, the trajectory of the experiment itself is quite short ( $\approx 10m$ ) to be seriously and accurately covered with precise GPS position update. Figures 6.4(c) and 6.4(d) show the tested trajectory and depict the encountered amount of GPS imprecision.

**Without GPS:** EKF localisation was tested in given conditions. Initially, only motion (inertial) sensors were used within the observations. That implies all the available linear and



(c) Coordinates of the tested square trajectory pasted on the lake map. (d) GPS signal as it appears originally when following the square trajectory.

angular velocity sensors. Absolute position (raw GPS signal in this constellation) was not included in observations. EKF periodically updates (synchronous mode, § 5), with the rate of 10 Hz. The aim was to measure the performance and the amount of drifting that occurs since the robot is intended to repeatedly attain square-like paths and return to the starting position in ideal scenario. It is useful to mention that proper parameter tuning can significantly change filtering performance. By giving more or less trust in particular measurement, or particular model behaviour, the role of certain parameters of dynamics (velocities, angular velocities) can be emphasized if necessary. Figure 6.5 shows the performance of EKF without GPS position correction after one trajectory cycle, and the original path that was followed, recorded using GPS. Initial position was taken from the first GPS measurement and the first measurement can indeed be away from the real position due to GPS imprecision. Note that the control of the vehicle trajectory refers to dead-reckoning calculated north-east values, not some physical beacons with known position. At the time of reporting the experiments, testing were not fully completed with all the planned scenario variations. Localisation expectedly tends to perform with a significant drift without the absolute position update (Figure 6.5(a)). After the second square-shaped cycle (Figure 6.5(b)), EKF shows that it roughly tracks the shape of the trajectory, smooths it by filtering out the measurement outliers. Vehicle has returned approximately to the same position after each cycle. Drift gained when following one of the sides of the rectangular path was compensated with the same amount of drift but of the opposite sign that was active when taking the return direction. Enormous amount of drift is present since the information on absolute position is not considered. The first available GPS coordinate

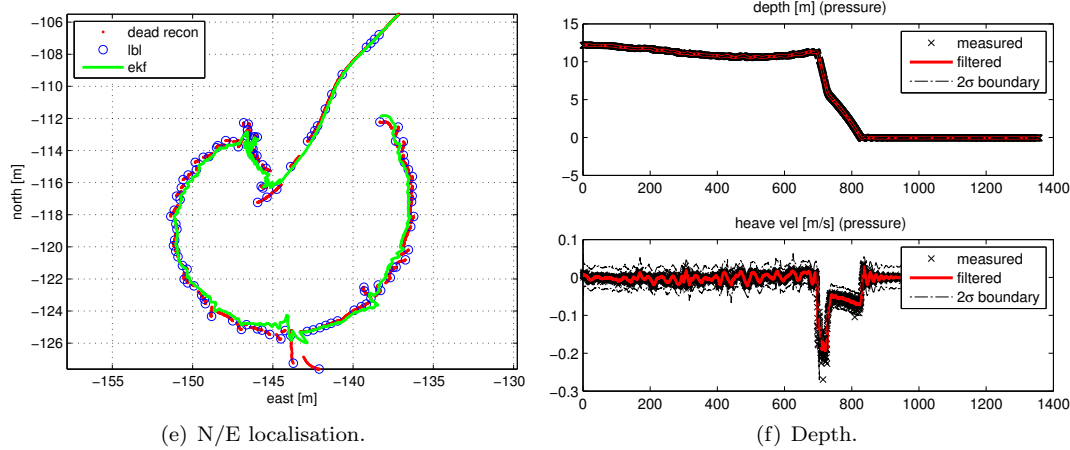


Figure 6.4: Spiral trajectory and the trajectory estimation using EKF.

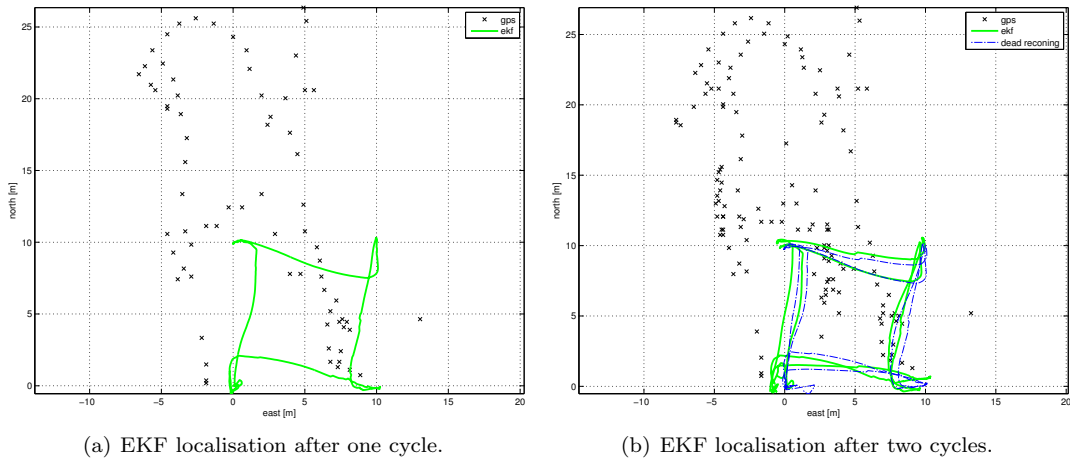
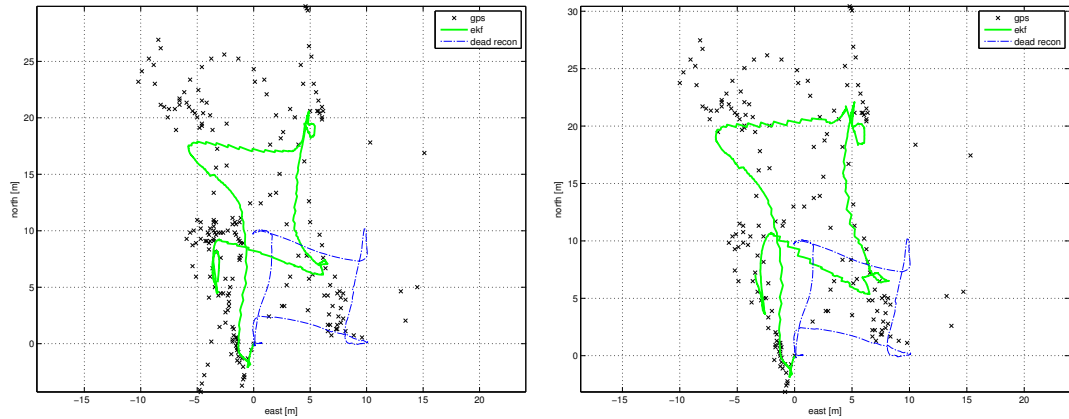


Figure 6.5: EKF localisation using only inertial measurements as observation.

fixes the starting position and part of the initial position error is caused by being incapable of setting the initial position accurately.

**With GPS:** Finally, GPS measurements were appended to the EKF observations. Localisation results were shown in Figure 6.6 for two different level of confidence (variances) in position measurement. Naturally, giving extremely high confidence does not seem to be the best choice, however, some empirically deduced values in range of decimetres significantly correct the dead reckoning drift. Furthermore, EKF tends to filter the GPS measured north east coordinates, hence partially corrects the GPS imprecisions stated at the beginning. At this point it is evident





(a) Setting standard deviation of 1 m in uncertainty position observation ( $SD_{north} = SD_{east} = 1.0$  m). (b) EKF localisation after tuning the position uncertainty ( $SD_{north} = SD_{east} = 0.5$  m).

Figure 6.6: EKF localisation aided with GPS position updates weighted by setting appropriate parameters.

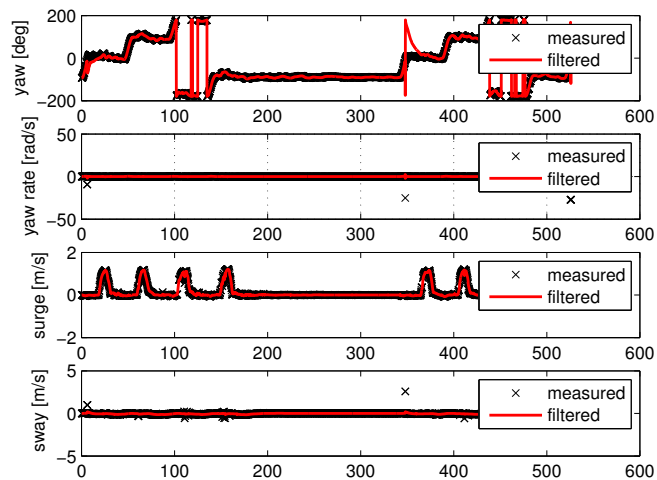


Figure 6.7: Linear and angular velocities during square-shaped trajectory.

---

why EKF is a great tool. Filter tries to satisfy the set uncertainty boundaries and fuse all the available information trying to make the most out of it combined together in one mathematical system. Moreover, fusing such imprecise and sketchy position data from GPS, still improves the localisation. Obtained trajectory tends to go towards what can be treated as expected path. From something that looked like a noisy collection of position observations at the beginning (Figure 6.4(d)), application of EKF together with sensor fusion enabled having generally better performance in navigation.

## Chapter 7

# Conclusions

The main focus of the work presented in the thesis is practical application of Extended Kalman Filter (EKF) for Ocean System Lab's Nessie AUV navigation module. EKF was designed to estimate the location of an underwater robot by processing real-time inertial and position information obtained from sensors. Furthermore, EKF algorithm was utilized as a framework for accomplishing sensor fusion - blending together measurements from different sensors as a part of the estimation process. Some principles of implementation for EKF-based localisation that were already available in literature were adopted and modified into a five d.o.f. vehicle model.

One of the issues that were addressed in the thesis was the suitable management of measurement tasks among mounted sensor devices and the role of EKF in correcting deficiencies. Specific case of heading measurement was tested, since this type of angular information is particularly important for the navigation. Usage of magnetic compass for obtaining heading is possible, however needs serious and precise calibration of the compass on the location itself. Advantage would be the possibility to make corrections based on absolute but quite noisy and unstable heading information that can be filtered using EKF tuned to compensate for the magnetic compass inaccuracies. Due to calibration difficulties, most of the experiments use the magnetic compass for initial heading, and EKF successfully appends the accurate heading rate information obtained from gyroscope to compensate for the lack of absolute heading.

In conclusion, EKF proves to be useful navigation tool with satisfactory navigation performance and several convenient features. Flexible with the number of sensors involved, capable of successfully combining together different sensory information into a location estimate that tends to be optimal with respect to set expectations, or recovering from the missing measurements, corrupted position information, outliers, or signal noise. The issues with nonlinearity have been addressed and the usage of Unscented Kalman Filter (UKF) tested on real data. Implemen-

tation of UKF for localisation would improve the accuracy of approximating nonlinearities in EKF at the same computational cost. However, this would have impact on navigation quality in cases when process model is fairly similar to the real robot movement. Simply said - UKF will tune up the emulation of the mathematical equations (prediction). If the prediction mode has an important role in estimation, UKF will contribute. A possibility to test adopted UKF that randomly takes samples is an idea to cope with filtering noisy position information such as LBL's "mutipathing" or imprecisions.

## 7.1 Future work

Future work on improving localisation performance involves more trials, particularly ones where the true trajectory has been fixed to known landmarks, so that the results of localisation could be thoroughly evaluated with trustful ground truth. Experiments that involve tilted vehicle movements could make an evaluation of the influence of the 5th degree of freedom on the quality of localisation. EKF could be improved so that it works with control inputs - which could contribute in robustness of the localisation.

Finally, the problem of correcting the absolute position with LBL information gives space for improvement since the measured position tends to be quite uncertain and prone to different sorts of noise. Solution for rejecting outliers could rely on some version of back-filtering - filtering based on history of received observations.

# Bibliography

- [1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.
- [2] A. Bahr and J. Leonard. Cooperative localization for autonomous underwater vehicles. In *Experimental Robotics*, pages 387–395. Springer, 2008.
- [3] M. Blain, S. Lemieux, and R. Houde. Implementation of a ROV navigation system using acoustic/Doppler sensors and Kalman filtering. In *OCEANS 2003. Proceedings*, volume 3, pages 1255–1260. IEEE, 2003.
- [4] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo. Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach. *Oceanic Engineering, IEEE Journal of*, 30(1):140–152, 2005.
- [5] M. Carreras, P. Ridao, R. Garcia, and T. Nicosevici. Vision-based localization of an underwater robot in a structured environment. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 971–976. IEEE, 2003.
- [6] D.E. Di Massa and WK Stewart Jr. Terrain-relative navigation for autonomous underwater vehicles. In *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 1, pages 541–546. IEEE, 1997.
- [7] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.
- [8] L. Drolet, F. Michaud, and J. Côté. Adaptable sensor fusion using multiple Kalman filters. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 2, pages 1434–1439. IEEE, 2000.

- 
- [9] M. Erol, L.F.M. Vieira, and M. Gerla. AUV-aided localization for underwater sensor networks. In *Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on*, pages 44–54. IEEE, 2007.
- [10] R. Eustice, R. Camilli, and H. Singh. Towards bathymetry-optimized Doppler re-navigation for AUVs. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1430–1436. IEEE, 2005.
- [11] R. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation in an unstructured environment using a delayed state history. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 25–32. IEEE, 2004.
- [12] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics: Science and Systems*, volume 2. Citeseer, 2005.
- [13] R.M. Eustice. Large-area visually augmented navigation for autonomous underwater vehicles. 2005.
- [14] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2417–2424. IEEE, 2005.
- [15] M.F. Fallon, G. Papadopoulos, J.J. Leonard, and N.M. Patrikalakis. Cooperative auv navigation using a single maneuvering surface craft. *The International Journal of Robotics Research*, 29(12):1461, 2010.
- [16] Jay Farrell and Jay A. Farrell. *The Global Positioning System & Inertial Navigation*. McGraw-Hill Professional, 1 edition, December 1998.
- [17] K. Gade and B. Jalving. An aided navigation post processing filter for detailed seabed mapping UUVs. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The 1998 Workshop on*, pages 19–25. IEEE, 1999.
- [18] R. Garcia, J. Batlle, X. Cufi, and J. Amat. Positioning an underwater vehicle through image mosaicking. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2779–2784. IEEE, 2001.
- [19] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, volume 140, pages 107–113. IET, 1993.
- [20] M.S. Grewal, A.P. Andrews, and Ebooks Corporation. *Kalman filtering: theory and practice using MATLAB*. Wiley Online Library, 2001.

- 
- [21] M. Jakuba and D. Yoerger. High-resolution multibeam sonar mapping with the autonomous benthic explorer (ABE). In *Proc. 13th Unmanned Untethered Submersible Technol. Conf*, page 2003.
- [22] J.T. Joiner. *NOAA diving manual: diving for science and technology*. 2001.
- [23] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *Automatic Control, IEEE Transactions on*, 45(3):477–482, 2000.
- [24] S. Julier and J.K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. *Robotics Research Group, Department of Engineering Science, University of Oxford, Oxford, OC1 3PJ United Kingdom, Tech. Rep*, 1996.
- [25] S.J. Julier. The scaled unscented transformation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4555–4559. IEEE, 2002.
- [26] R.E. Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
- [27] R. Karlsson, F. Gusafsson, and T. Karlsson. Particle filtering and Cramer-Rao lower bound for underwater navigation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 6, pages VI–65. IEEE, 2002.
- [28] J.C. Kinsey. *Advances in precision navigation of oceanographic submersibles*, volume 67. 2007.
- [29] J.C. Kinsey, R.M. Eustice, and L.L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *Proceedings of the 7th Conference on Maneuvering and Control of Marine Craft (MCMC2006). IFAC, Lisbon*. Citeseer, 2006.
- [30] F. Maurelli, S. Krupinski, Y. Petillot, and J. Salvi. A particle filter approach for AUV localization. In *OCEANS 2008*, pages 1–7. IEEE.
- [31] R. Negenborn. *Robot localization and kalman filters*. PhD thesis, Citeseer, 2003.
- [32] P. Newman and H. Durrant-Whyte. Using sonar in terrain-aided underwater navigation. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 440–445. IEEE, 1998.

- [33] Y. Petillot and F. Maurelli. A tree-based planner for active localisation: applications to autonomous underwater vehicles. In *ELMAR, 2010 PROCEEDINGS*, pages 479–483. IEEE.
- [34] D. Ribas, P. Ridao, and J. Neira. *Underwater slam for structured environments using an imaging sonar*. Springer Verlag, 2010.
- [35] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.
- [36] C.N. Roman. Self consistent bathymetric mapping from robotic vehicles in the deep ocean. 2005.
- [37] T. Ruiz, Y. Petillot, D.M. Lane, and C. Salson. Feature extraction and data association for AUV concurrent mapping and localisation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2785–2790. IEEE, 2001.
- [38] J.R. Van Zandt. A more robust unscented transform. In *Proc. 2001 SPIE Conf. on Signal and Data Processing of Small Targets*, volume 4473.
- [39] E.A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.
- [40] L. Whitcomb, D. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control, and survey operations. In *Navigation, Control and Survey Operations, in The Ninth International Symposium on Robotics Research*. Citeseer, 1999.
- [41] L.L. Whitcomb, D.R. Yoerger, H. Singh, and J. Howland. Combined Doppler/LBL based navigation of underwater vehicles. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*. Citeseer, 1999.
- [42] S. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, pages 1771–1776. IEEE, 2004.
- [43] S. Williams and I. Mahon. A terrain-aided tracking algorithm for marine systems. In *Field and Service Robotics*, pages 93–102. Springer, 2006.
- [44] S.B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Robotics and Automation, 2000*.



- Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1793–1798. IEEE, 2000.
- [45] X. Yun, E.R. Bachmann, and S. Arslan. An inertial navigation system for small autonomous underwater vehicles. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1781–1786. IEEE, 2000.