An Advanced Navigation System for Remotely Operated Vehicles (ROV)

Sérgio Lourenço
sergio.lourenco@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

March 30, 2017

Abstract

The underwater environment of Remotely Operated Vehicles (ROV) navigation creates a number of difficulties that do not arise in land based robotic navigation. The most proeminent of these is the inability of GPS signals to penetrate the water medium, making acoustic based sensors the fallback choice. This type of sensors constrain the ROV capabilities in a dramatic manner, due to the low speed of propagation of acoustic signals in the water. Dealing with this, and the common problems (sensor noise, outliers, model noise, ...) of robotic navigation while building a filter capable of estimating accurately the position of an ROV are the key challenges addressed in this work. An Extended Kalman Filter was used to solve the problem, with adaptations to deal with the specific details of the underwater scenario. A threshold was created based on the covariance of the estimated position to reject outliers. To deal with the different update rates of the sensors, the filter is capable of estimating in dead reackoning, or with only a partial set of the observations. To deal with noise, the EKFs natural estimation capabilities are used and tuned with the models noise parameters, in order to increase the importance of dead reckoning or sensor information.

Keywords: Navigation, DVL, USBL, Extended Kalman Filter, ROV

1. Introduction

Underwater robotic navigation is an increasingly important aspect in aquatic exploration. This document details an approach to solve a navigation problem for a ROV.

Considered a remotely operated vehicle in an underwater scenario. The operator pilots the vehicle from a support vessel, inserting commands that are sent to the vehicle nearly instantaneously through a long cable that connects the ROV to the support vessel. The operator has no visual knowledge of the vehicles location.

The ROV is equipped with several sensors, namely a sensor to measure the velocity, a sensor to measure the heading and turning rate, and a sensor to measure the X, Y, Z position of the ROV, relative to the support vehicle. All the measurements are sent to the support vessel to perform calculations through the connecting cable.

The problem to solve is to estimate the localization of the ROV relative to the support vessel given the sensor measurements. It is a variant (due to the underwater environment) of a classical localization problem. To solve that problem an Extended Kalman Filter was used.

The state representation chosen consists of the X, Y, Z positions, the heading and turning rate, and the water currents velocity. The model used in the Extended Kalman Filter was the Random Walk with Constant Turning Rate. This model describes a circular or linear movement of the ROV, with a constant turning rate or acceleration. For the Z plane, movement is linear. This depicts only emerges and dives. The velocity inputs do not enter in the internal state of the model.

To deal with outliers, the measurements are subjected to a threshold before being used in the filter. Delay of measurements is disregarded as non existant, and the assynchronous measurement problem has no specific adaptation. The update is done when there are measurements available and can be done partialy, i.e., only some values of the internal state are updated.

2. Navigational Algorithms

A common problem in engineering is to estimate the position of an object given some internal and external measurements of that object. For a static object this is the trilateration or triangulation problem. However, for a dynamic object, the knowledge of the dynamics of the object can help reduce the estimation error.

Consider an object with some approximately known non linear, time varying dynamics,

$$\frac{dx(t)}{dt} = f(x(t), u(t), w(t)). \tag{1}$$

Let $x(t) \in \mathbb{R}^n$ be the state vector of the object, $u(t) \in \mathbb{R}^m$ the input vector and $w(t)$ the vector that captures system modeling errors.

Assume there is some kind of sensors that gives some observations of the object in question. The observation vector $y(t) \in \mathbb{R}^r$ will be related to the current state of the system $x(t)$. Let $h(x(t), v(t))$ be the measurement function and $v(t)$ be the vector that represents measurement errors, that is,

$$y(t) = h(x(t), v(t)). \tag{2}$$

The most common problem in general filtering is to estimate the internal state $x(t)$ given an approximate knowledge of the system dynamics, $f(x(t), u(t), w(t))$, the control vector $u(t)$, an approximate measurement function $h(x(t), v(t))$ and some approximate sensor measurements $y(t)$. The system dynamics and measurement functions are considered constant for the entire problem while the control vector and sensor measurements change at each time instant.

It is important to note that in most situations the observations and control strategies are made at discrete time instants, such as when using a computer to monitor and control the system. The system can still be a continuous system and the measurements can be obtained by sampling the said system. Consider the system in equation (1). Consider two time instances t_2, t_1 and a sampling period $\Delta = t_2 - t_1$. Then $t_2 = (k+1)\Delta$ and $t_1 = k\Delta, k \in \mathbb{Z}$. Approximating a continuous derivative $\frac{dx(t)}{dt}$ by a discrete difference $\frac{x((k+1)\Delta) - x(k\Delta)}{\Delta}$, the system dynamics become

$$x((k+1)\Delta) - x(k\Delta) = \Delta f(x(k\Delta), u(k\Delta), w(k\Delta)). \tag{3}$$

Simirarly the discrete version of the observations is given by

$$y(k\Delta) = h(x(k\Delta), v(k\Delta)). \tag{4}$$

To simplify the notation consider the general filtering problem in a discrete scenario, where $f_k(x_k, u_k, w_k) = x(k\Delta) + \Delta f(x(k\Delta), u(k\Delta), w(k\Delta))$, that is,

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{5}$$

To solve the general filtering problem the optimal value of x_k must be found. Optimality is expressed as the estimate of x_k that minimizes the state estimate error in some respect. One manner of evaluating the estimate is by considering the Bayesian belief for that estimate. The Bayesian belief gives a measure of how probable it is that the system has a certain state, given the measurements and the control inputs at the previous time instants. This corresponds to the conditional probability density function (pdf)

$$Bel(\hat{x}_k) = P(\hat{x}_k \| y_1, ..., y_k, u_0, ..., u_{k-1}), \tag{6}$$

where $\hat{x}$ is the estimate of x.

A common algorithm to perform the filtering process is to divide each estimation, at each time instant, into two major steps: prediction and correction.

1. Given an initial estimate of x_0
2. Apply u_0 and the system evolves to the state x_1
3. Make a measurement to the state x_1, y_1
4. Estimate the state x_1 from equation (6), obtaining $\hat{x}_1$
5. Repeat process with $\hat{x}_1$

Normally, it is assumed that the system is a Markov process. In that case equation (6) can be reduced to

$$Bel(\hat{x}_k) = P(\hat{x}_k \| y_k, u_{k-1}, \hat{x}_{k-1}). \tag{7}$$

This estimate $\hat{x}_k$ obtained is dependent on the optimization criterion. Considering a general pdf, the estimate can be obtained with the center of probability mass (mean), the highest probable value (Maximum a Posterior (MAP) criteria) or the value of $\hat{x}_k$ such that half the probability weight lies to the left and half to the right of it. The most comonly used estimation criteria is the center of probability mass or minimum variance estimate (MVE) criteria, that is,

$$E[\|X - \hat{x}\|^2 | Y = y] \leq E[\|X - z\|^2 | Y = y], \forall z \text{ obtained from } Y. \tag{8}$$

Using MVE, $\hat{x}$ is uniquely defined as the conditional mean of X. given that $Y = y$, as shown in [1], ch. 2, theorem 3.1., that is,

Theorem 1. *Let X and Y be two jointly distributed random vectors, and let Y be measured as taking the value y. Let $\hat{x}$ be a minimum variance estimate of X, as defined in equation (8). Then $\hat{x}$ is also uniquely specified as the conditional mean of X given that $Y = y$, i.e.,*

$$\hat{x} = E[X|Y = y] = \int_{-\infty}^{+\infty} x p_{X|Y}(x|y) dx$$

3. Kalman Filter

The Kalman Filter arises as a particularization of the general filtering problem, considering gaussian noise and a linear system. Consider two jointly gaussian random vectors, X and Y. From [24], result 3.3.1, we have

$$E[X|Y] = E[X] + R_{XY}R_{YY}^{-1}[Y - E[Y]], \tag{9}$$

$$\begin{aligned} R_{XY} &= E[(X - E[X])(Y - E[Y])^T], \\ R_{YY} &= E[(Y - E[Y])(Y - E[Y])^T]. \end{aligned} \tag{10}$$

Note that this result, taking into account Theorem 1, shows that the MVE for the conditional mean-square error is a linear combination of the observations.

Assuming that the system is linear and time-invariant, affected by gaussian noise it can be represented by

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + G w_k, k \geq 0 \\ y_k &= C_k x_k + v_k \end{aligned} \tag{11}$$

$$E[w_k] = E[v_k] = 0, \tag{12}$$

where $x_k \in \mathbb{R}^n, u_k \in \mathbb{R}^m, w_k \in \mathbb{R}^n, v_k \in \mathbb{R}^r, y_k \in \mathbb{R}^r$ and w_k, v_k are sequences of white, zero mean, Gaussian noise with zero mean.

Let

$$E[\begin{pmatrix} w_k \\ v_k \end{pmatrix}(w_k^T v_k^T)] = \begin{pmatrix} Q_k & 0 \\ 0 & R_k \end{pmatrix} \tag{13}$$

be the joint covariance matrix for w_k and v_k.

Assume also that u_k is deterministic and the initialization is

$$\begin{aligned} E[x_0] &= \hat{x}_0 \\ E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] &= \varepsilon_0 \end{aligned} \tag{14}$$

For this particular specification of the general filtering problem a few remarks can be made:

- The conditional probability density functions $p(x_k|Y_k, U_k)$ are gaussian for any k,
- The $p(x_k|Y_k, U_k)$ are fully represented by their mean and covariance, $\mathcal{N}(\hat{x}(k|k), P(k|k))$,
- the mean, mode and median for this pdf coincide,
- the filter that propagates $p(x_k|Y_k, U_k)$ and estimates the state by optimizing a given criteria, i.e., the Kalman Filter, is the optimal filter.

The mean is given by

$$\begin{aligned} \hat{x}(k|k) &= E[x_k|Y_k, U_k] \\ P(k|k) &= E[(x_k - \hat{x}(k|k))(x_k - \hat{x}(k|k))^T|Y_k, U_k]. \end{aligned} \tag{15}$$

The covariance of a gaussian variable Z is $E[(Z - E[Z])(Z - E[Z])^T]$ and for this particular case can also be found in 15. Since the pdf is represented by its mean and covariance, the Kalman filter only propagates the first and second moments of the pdf, resulting in a computational lighter and faster solution than more complex filter, albeit more limited.

The common approach to use the Kalman filter is to do a recursive algorithm, computing each step k based on the observations at the step k, the state estimate from step $k-1$ and the system inputs at step $k-1$, with a Markov assumption for the system. This approach is normally devided into 2 steps:

1. Prediction Step: Computing $p(x_k|Y_{k-1}, U_{k-1})$. This corresponds to inputting some signal in the system and estimating the next state without taking any observations. Its also called Dead Reckoning.
2. Update Step: Computing $p(x_k|Y_k, U_{k-1})$. Having the dead reckoning estimate of the state, this step introduces the observation made after inputting the signal, correcting the dead reckoning estimate.

The derivation of the new filter can be found in [26], chapter 6.2. To summarize, the filter becomes

$$\begin{aligned} I_{k+1|k} &= Q_k^{-1} - Q_k^{-1}F_k(I_{k|k} + F_k^T Q_k^{-1} F_k)^{-1} F_k^T Q_k^{-1}, \\ I_{k+1|k+1} &= I_{k+1|k} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}, \\ K_{k+1} &= I_{k+1|k+1}^{-1} H_{k+1}^T R_{k+1}^{-1}, \\ \hat{x}_{k+1|k} &= F_k \hat{x}_{k|k} + G_k u_k, \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - H_{k+1}\hat{x}_{k+1|k}), \end{aligned} \tag{16}$$

with the system dynamics already described in equation (11). With the information filter, instead of computing the inverted matrix of an $m \times m$ matrix, it is necessary to compute the inverses of a few $n \times n$ matrix, where n is the length of the state vector. Even considering the matrices of the noise covariance constant, at least one $n \times n$ matrix must be inverted at each iteration. One criteria to choose between the standard filter and the information filter is comparing n and m. If $m \gg n$, then the information filter is preferable to the standard filter. Another particular of this alternative formulation is the initialization of the filter. Using P_k, we represent the uncertainty of the filter. If the initial state is fully known, it is equal to zero. However, if the state is completely unknown, it is not possible to numerically represent P_k as infinite. For the information matrix, the problem is reversed. Unknown

states can be represented as zero, but fully known states can not be numerically represented.

4. Extended Kalman Filter

The kalman filter, as was seen before, is the optimal filter for a linear problem. However linear problems are the scarcest of real world problems. In most cases they are simply approximations of a non linear problem. In a non linear scenario, the KF loses its' optimality and other solutions must be found. One solution is the Extended Kalman Filter, which, as implied, is an extension over the regular KF. Other examples include the Unscented Kalman Filter and the Particle Filter. Although varying in complexity, accuracy and computational effort, none of these solutions is an optimal filter, but in a strongly non linear scenario all of them are better than the kalman filter.

Consider the system dynamics

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, w_k), \\ y_k &= h_k(x_k, v_k), \end{aligned} \tag{17}$$

with nonlinear state and observations. Note that $x_k \in \mathbb{R}^n$ represents the state variable, $u_k \in \mathbb{R}^m$ the controller input, $y_k \in \mathbb{R}^r$ the output of the system and $v_k \in \mathbb{R}^n$ and $w_k \in \mathbb{R}^r$ are the noise present for state and observations, respectively. The assumption of white gaussian, independent random processes with zero mean and covariance remains for the noise variables, with $E[v_k v_k^T] = R_k$ and $E[w_k w_k^T] = Q_k$. The set of measurements until instant k is refered as Y_k.

As noted before, the mean-square error estimator computes the conditional mean of x_k given the observations Y_k. For that computation the propagation of the entire conditional pdf is required, except in the special case of linear system dynamics, which results in the aforementioned Kalman Filter. With a linear system the Kalman Filter is the optimal filter, as seen before, but with non linear conditions that is not the case. The conditional pdfs $p(x_k|Y_k)$,$p(x_{k+1}|Y_k)$ and $p(x_{k+1}|Y_{k+1})$ are not gaussian in this case. As such, the optimal filter needs to propagate the entirety of these functions, resulting in a heavy computational burden. Neglecting optimality, the Extended Kalman Filter aproximates the optimal filter by linearizing the system dynamics about the most recent estimate. By linearizing the system then it is not necessary to propagate the entirety of the pdf, and the Kalman Filter can be applied to the linearized system, although the accuracy of this linearization can greatly affect the result of the filter.

To linearize the system dynamics, consider the Taylor series expansion: $f(x) = f(x_0) + \frac{\partial f(x)}{\partial x}(x - x_0) + \frac{1}{2} * \frac{\partial^2 f(x)}{\partial x^2}(x-x_0)^2 + ... + \frac{1}{n!}\frac{\partial^n f(x)}{\partial x^n}(x-x_0)^n + ...$ about the most recent estimate $\hat{x}_{k|k}$ and the noise $w_{k-1} = 0$. Together with equation (17) one arrives at

$$\begin{aligned} x_{k+1} &= f_k(\hat{x}_{k|k}, u_k, 0) + \frac{\partial f_k(\hat{x}_{k|k}, u_k, 0)}{\partial x}(x_k - \hat{x}_{k|k}) + \\ &\quad \frac{\partial f_k(\hat{x}_{k|k}, u_k, 0)}{\partial w} w_k \\ &= f_k(\hat{x}_{k|k}, u_k, 0) + F_k(x_k - \hat{x}_{k|k}) + L_k w_k \\ &= F_k x_k + [f_k(\hat{x}_{k|k}, u_k, 0) - F_k \hat{x}_{k|k}] + L_k w_k \\ &= F_k x_k + \bar{u}_k + \bar{w}_k, \end{aligned} \tag{18}$$

disregarding higher order members.

$\bar{u}_k$ and $\bar{w}_k$ are defined in the equation above, with $\bar{u}_k = f_k(\hat{x}_{k|k}, u_k, 0) - F_k \hat{x}_{k|k}$ and $\bar{w}_k$ $(0, L_k Q_k L_k^T)$. Repeating the derivation for the measurement equation around the estimate $\hat{x}_{k+1|k}$ and $v_k = 0$ results in

$$\begin{aligned} y_k &= h_{k+1}(\hat{x}_{k+1|k}, 0) + \frac{\partial h_{k+1}(\hat{x}_{k+1|k}, 0)}{\partial x}(x_{k+1} - \hat{x}_{k+1|k}) \\ &\quad + \frac{\partial h_{k+1}(\hat{x}_{k+1|k}, 0)}{\partial v} v_k \\ &= h_{k+1}(\hat{x}_{k+1|k}, 0) + H_{k+1}(x_{k+1} - \hat{x}_{k+1|k}) + M_k v_k \\ &= H_{k+1} x_{k+1} + [h_{k+1}(\hat{x}_{k+1|k}, 0) - H_{k+1}\hat{x}_{k+1|k}] + M_k v_k \\ &= H_{k+1} x_{k+1} + z_k + \bar{v}_k. \end{aligned} \tag{19}$$

As before, z_k and $\bar{v}_k$ are defined as $z_k = h_{k+1}(\hat{x}_{k+1|k}, 0) - H_{k+1}\hat{x}_{k+1|k}$ and $\bar{v}_k$ $(0, M_k R_k M_k^T)$. From this point we have a linear state space system and a linear measurement equation. As such, the Kalman Filter can be applied to estimate the state of the system. The Extended Kalman Filter algorithm can be stated as

$$\begin{aligned} \hat{x}_{k+1|k} &= f_k(\hat{x}_{k|k}), \\ F_k &= \frac{\partial f_k(\hat{x}_{k|k}, 0, 0)}{\partial x}, \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k, \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}[y_{k+1} - h_{k+1}(\hat{x}_{k+1|k})], \\ H_{k+1} &= \frac{\partial h_{k+1}(\hat{x}_{k+1|k}, 0)}{\partial x}, \\ K_{k+1} &= P_{k+1|k} H_{k+1}^T [H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1}]^{-1}, \\ P_{k+1|k+1} &= [I - K_{k+1} H_{k+1}] P_{k+1|k}, \end{aligned} \tag{20}$$

considering no external inputs.

4.1. Extensions to the EKF

The general Extended Kalman Filter is a powerful tool that is employed in many different situations. Naturally, to obtain better results to a specific problem, namely with a underwater ROV, it is important to specialize it to the situation. The main modifications made to the EKF in an underwater scenario are to deal with assynchronous sensor measurements, delay in measurements due to the water medium and to deal with outliers in the measurements.

4.1.1 Assynchronous Measurements

The EKF equations require the measurements to be available at each iteration. That, however, is not

always possible. This is due to the rate at which measurements are available to the ROV. Consider that there are m sensors available. Each sensor can make one measurement in t_m seconds. That means that it has an update rate of $\frac{1}{t_m}$. This update rate may be different for each sensor. In order to use the most measurements, and assuming no limitations in computation velocity, each discrete time instant must differ from the previous one by a time interval $t \leq min(t_1, ..., t_m)$. This means that there will be some time instants where some of the measurements in sensors 1 to m will not be available, due to still being in computation by the sensors, and are therefore not available to be used in the EKF equations. If these measurements are used in the update step, they can simply be assumed as non existant and removed from the equations for that iteration. If the measurements are used as inputs, as in the case of the velocity, this is not possible. The equations require this variable. To deal with this the filter simply uses the last known value for that input.

4.1.2 Delayed Measurements

The usage of acoustic transmission requires a much more significant travel time than an eletrocmagnetic transmission, as in GPS. To address this delay is necessary to take into account that when a sensor reading arrives at the computations' physical device, normally the support boat for the ROV, it may correspond to a previous time instant than the one being computed at the time. This topic is treated in many cases, namely in [23]. For the case related with this thesis the travel time was considered not significant due to a very slow moving ROV being used.

4.1.3 Outliers

The techniques described so far rely on an addictive, zero-mean, noise. In general, however, there is a significant amount of outliers in the sensor readings that can hurt the performance of these techniques. In order to further specialize the EKF to the underwater problem, an outlier rejection algorithm is necessary. One possible way to perform outlier rejection is to define a threshold ϵ that rejects observations x_r. In order to make that threshold dependent on that knowledge the approach used is to make it dependent on the covariance $P_{k+1|k}$. With this, the threshold will increase when there is less certainty on the estimation, and decrease when the certainty increases. To avoid that the threshold grew too high it was upper bounded. Simirarly, when we have a large certainty of the estimate, such as in the start position or by resurfacing the ROV and using GPS, the threshold would be so small that all noisy measurements would be disregarded as outliers. To avoid such a situation a lower bound was also imposed on the threshold calculation. The computation of the threshold used is in equation (21), with β being a static minimum for the threshold, θ and Θ the variables to adjust lower and upper bounds and σ a constant to adjust the mean of the diagonal entries of the covariance matrix.

$$\begin{aligned} \epsilon &= (\alpha + \theta) * \|\hat{x}_{k+1|k}\| + \beta, \\ \alpha &= min(\gamma, \Theta), \\ \gamma &= \sigma * \frac{diag(P_{k+1|k})}{n}. \end{aligned} \tag{21}$$

4.2. Model

All of the techniques described so far, namely the one used, rely on a model, albeit an approximate one, of the system dynamics. This model, in pratice, can be as detailed as the sensors available allow. Recall equation (1) on page 2, which represents the system dynamics. In the problem considered, the state observed y_k will always be the cartesian position of the ROV p_k. The function $f(.)$ must be defined. Although there are many diferent possible functions available, due to the sensors used, the model choosen was a model with constant turning rate in X, Y and linear velocity in Z.

By considering the speed and turning rate the Random Walk with Constant Turning Rate model can learn the curvature of the trajectory, allowing a better estimation if observations are missing. The state will be more complex than a model considering linear motion, with the aditional variables ψ_k that represents the angle and r_k representing the turning rate. In most cases the movement in the Z axis, when it corresponds to altitude or depth, is linear. As such the added complexity of learning the curvature in the Z axis is unnecessary.

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} p_{k+1} \\ v_{w_{k+1}} \\ \psi_{k+1} \\ r_{k+1} \end{bmatrix} = \\ &\begin{bmatrix} p_k + v_{w_k} + S(\psi_k)v_k + \xi_{p_k} \\ v_{w_k} + \xi_{v_{w_k}} \\ \psi_k + r_k + \xi_{\psi_k} \\ r_k + \xi_{r_k} \end{bmatrix} = \\ &\begin{bmatrix} I_n & I_n & 0 & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & I_{n-1} & I_{n-1} \\ 0 & 0 & 0 & I_{n-1} \end{bmatrix} \begin{bmatrix} p_k \\ v_{w_k} \\ \psi_k \\ r_k \end{bmatrix} + \\ &\begin{bmatrix} S(\psi_k)v_k \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \xi_{p_k} \\ \xi_{v_{w_k}} \\ \xi_{\psi_k} \\ \xi_{r_k} \end{bmatrix} = \\ &Ax_k + u_k + \xi \end{aligned} \tag{22}$$

The velocities used in equation (22), v_{w_k}, v_k, correspond to the velocity vectors' magnitude, with the

angles being represented by ψ_k. As such, $S(\psi_k)v_k$ projects the velocity in each of the n directions of p_k. For the 3D case ($n = 3$), $S(\psi_k)$ is defined by equation (23).

$$S(\psi_k) = \begin{bmatrix} cos(\psi_k) & 0 & 0 \\ 0 & sin(\psi_k) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (23)$$

The noise variable $\xi \sim (0, Q_k)$ allows to change the velocity of adptation of the filter response. For small covariance the model will be of a vehicle with slow dynamics and changes in trajectory or velocity will take a long time to be learned. It is more robust to noisy measurements and outliers as an upside. By having larger covariances, the model will be of a vehicle with fast dynamics, that expects rapid changes in trajectory and behaviour. Noisy measurements will have a greater impact on the state, but it will learn a new trajectory faster.

5. Results

5.1. EMEPC

EMEPC is an acronym for Continental Platform Expansion Mission Estructure (Estrutura da Missão de Espansão da Plataforma Continental). It is a portuguese project with the purpose of, as the name implies, expanding the continental plataform through the use of technological innovation and research. From its website: "The EMEPC's mission is to prepare and monitor the process of extending the continental shelf of Portugal. Given the nature of the work required to achieve this mission, the EMEPC contributes to the enhancement of the technical and scientific skills required for deep sea operations, projecting Portugal as a credible, internationally recognised partner." [8].

The EMEPCs project more relevant to this work in the investment in the ROV Luso, for up do 6000 meters deep sea exploration. During 2015, the EMEPC partnered with IST/ISR/LARSyS and CEiiA [8] to perform exploration missions. The experimental data derived from those mission is used in this work. These dives occured between the 25^{th} of may and 3^{rd} of june of 2015. More information about EMEPC and its missions can be found in [8].

5.2. ROV Luso

The ROV Luso is a scientific Remotely Operated Vehicle that was adapted for collecting geological and biological samples from the sea. The detailed technical specifications for the ROV Luso can be found in [25].

The most relevant sensors for this work present in the ROV Luso are:

- the Doppler Velocity Log (DVL), model WorkHorseNavigator 1200
- the Acoustic Positioning System (Ultra Short BaseLine), model TrackLink10000HA
- the Gyroscope, model KVHDSP3000
- the Compass, model KVHC100
- the Altimeter, model Kongsberg1007
- and the Depth Sensor, model SAIVDepthMeter

The remaining sensors can be found in [25].

5.3. Experimental Results

In this sub section the results from a experimental run will be presented. Unlike the simulation, there is no true data to rely on to determine the performance of the algorithm, so an analysis of the innovation and the threshold will be used. In this case, the ROV does some exploration on a depth of around 300 meters and then emerges. The model and sensor noise parameters used are described in detail in table 1 and in table 2. The approach used was to consider the both the model and observations at a similar relevance, giving no preferance to either. The water current values are low, to try to mimic a small variance in them during the dive. This stems from the idea that the water currents will not change significantly during a short period of time, and that there aren't many variations in the Z axis. Many and rapid variations in the Z axis could mean that the velocity measurements are taken relatively to the water and the ground simultaneously during the experiment. It is expected that this does not happen most of the time.

Table 1: Observational Noise Parameters

Axis	Noise Value
X	0.1
Y	0.1
Z	0.1
Heading	0.5

Table 2: Model Noise Parameters

Axis	Noise Value
X	0.1
Y	0.1
Z	0.1
Heading	0.5
Angular Rate	0.05
Water Current X	0.001
Water Current Y	0.001
Water Current Z	0.001

In figure 1 on the following page the resulting XY Plane sensor measurements and estimated position

can be seen. Most of the iterations there is a similar result from the sensor measurements and the estimated position. To better understand the results, in a global view, refer to table 3 on page 12. In the table, one can see that the mean innovation is very small, meaning that each sensor measurements adds little information to the state. There are, however, large maximum jumps in the state, that can be a indicator of the filter having a bad estimation in some iterations. It is important to note that, during the iterations, only 8 measurements were rejected and not used to improve the state estimation.

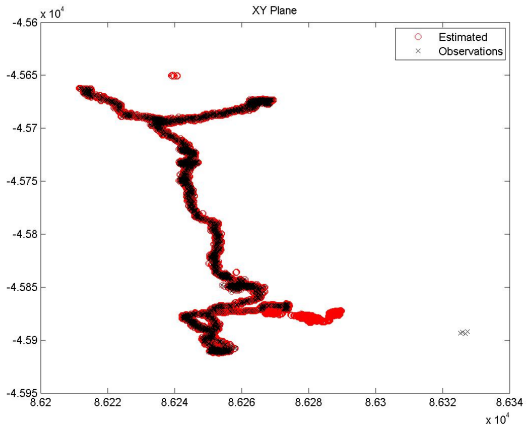


Figure 1: XY Plane

To better understand the quality of estimation in each of the position and heading parameters, the detailed innovation and threshold evolutions are shown below.

To reiterate, the X axis follows a turning rate model, which straight and circular, with constant turning rate, movements. In figure 2a on page 14 the results of the estimations are shown. The black x marks show all of the observations, including the rejected ones. The red line shows the filter estimation for the X axis. Due to the order of the value of X, 10000, the scale can be deceiving when trying to understand the sensor accuracy and the filters perfomance. Considering the number of measurements rejected was 8, the filter closely follows the sensor observations.

In figure 3a on page 15, the innovation due to sensor information can be seen. There are spikes which show times where the filter started to deviate from the sensors significantly. This behaviour is expected however, as there are many iterations without measurements. As one could deduce from the previous image, the innovation is mostly below 5 meters. The numerical results are detailed in table 4 on page 12, globally, and in table 5 on page 12 in histogram form.

The maximum and minimum value of innovation are related with the peaks seen in figure 3a on page 15. In the figure one can already have and idea of how often do these peaks occur. Altough the mean may be affected by them, the histogram is a better alternative to understand the distribution of innovation values. Looking at table 5 on page 12, the values of innovation higher than 10 or lower than -10 are 6, which correspond to the 6 peaks in the figure.

The mean value of innovation is very small, which colaborates figure 2a on page 14, which shows that the filter is always close to the sensor observations. The median value is not particulary relevant, and only shows that the filter equally over and underestimates the X axis.

The histogram table 5 on page 12 was generated without the zero values, considering only when there is innovation. Looking closely at the histogram, one can see that around 91% of the innovations are between -1 and 1 meters. 58% are between -0.1 and 0.1 meters. This suggest a good filter estimation most of the time, considering that the scale of the X axis values are on the order of 10000.

As was the case with the X axis, the Y axis also has a turning rate model, which straight and circular, with constant turning rate, movements. In figure 2b on page 14 we can see, similarly as the X axis, that the filter is closely following the sensor measurements. Apart from a few spaced cases, there does not seem to be a big innovation.

When looking at the innovation in figure 3b on page 15, most cases seem to fall below 2.5 meters. The peaks are still present, as before, due to times where the filter was more significantly different from the sensor observations.

Comparing the mean value for the innovation in the Y axis, it is around 50% higher than with the X axis. This indicates that the Y axis has a poorer estimation, maybe due to a more complex and rapid motion than the X axis, or to a more erratic sensor output. The maximum and minimum values are lower, although don't appear to be significantly so.

Considering the histogram for the Y axis, one can see that 89% of innovations fall between -1 to 1 and that 57% are between -0.1 and 0.1. As with the X axis, this histogram is done ignoring the 0 values for innovation. These small values for most of the innovations indicate a stable and consistently approximate estimation.

Looking now again at the XY Plane, in figure 1, one can say that the filter is successfully estimating the position. Each sensor adds, on average, very little information, on the order of milimeters. There are however, times when the filter starts diverging from the sensors, but never so much as to required a significant number of iterations to return to a cor-

rect estimation.

For the Z axis, the motion is simpler. Assuming only dives and emerges, the Z axis motion predicted by the model is linear. As seen in figure 2c on page 14 this is only approximately true, as the Z axis value is not constant during the deep sea exploration. Even though that is the case, the figure seems to indicate that the filter is still able to estimate correctly the value for the Z axis.

When looking at the innovation, in figure 3c on page 15, the innovations appear to be smaller than with the X and Y axis. Apart from the peaks, it is much smaller than 2.5, the value that the other 2 axis seem be limited by.

The mean value of the innovations, seen in table 8 on page 12, don't seem to corroborate that though. It sits between the Y axis and the X axis, although the minimum and maximum values are significantly lower.

Finally, looking at the histogram, the total number of non zero innovations were 5624. This falls far below the 15079 of the previous two axis. 97% of the bins are situated below 1 and above -1, while only 15% are between -0.1 and 0.1. Comparing this with the previous axis, it shows that although the peaks of innovation are smaller and there are less innovations, the innovations that exist are larger. The most probable cause is the model predicting a linear behaviour on the Z axis, with is not the case.

The position estimate, made up by the three axis X,Y and Z, has a small innovation on average, on the order of milimeters. The sensors are mostly not adding a lot of information. There is however one sensor with a contribution not shown in the innovation. The DVL enters the filter in the predict stage, having no internal state. It has naturally a big weight on the accuracy of the estimation, being one of the sensors with the higher update rate. A small increase in the errors of the DVL could make a significant difference on the results of the estimation.

The heading estimation, due to its nature, is prone to higher innovations around its limit value. The filter does not take this into account, treating it like any other measurement. As such, the jumps in the sensor from 360 degrees to 0 degrees will normally cause the filter to be erroneous during a number of iterations. In figure 2d on page 14 we can see that this happens a few times during the run. The innovation, in figure 3d on page 15 has significant peaks because of this. Apart from those cases, the filter seem to follow closely the sensor estimations.

In table 10 on page 12, we can see that in average the innovation is still small, on the order of a millionth of a degree. The maximum and minimum innovation is much higher than any other axis, due to these edge cases.

For the histogram, a signficant portion of the innovations are still between the -1 and 1 values, specifically 78%. It is smaller than the other axis, altough most of the innovations are below -0.1 and 0.1 (54%). These edge cases are few and far between, but if they were more proeminent the filter estimation of the heading could suffer greatly.

Another factor to consider when analysing the performance of the filter is the threshold. The threshold is dependent on the covariance of the estimation, growing in extended periods of dead reckoning. Since the threshold is used to determine if a sensor measurement is an outlier, and reject it, its peaks also indicate where there may have been measurements being rejected. The threshold used was divided into a threshold for the position and a threshold for the heading.

In figure 4a on page 15 the threshold evolution for the position, axis X, Y and Z, is represented. The more problematic areas seem to be in the first part of the iterations, where there is an higher covariance of the estimations. Due to the sheer number of interations, it is hard to understand completely the details of the evolution of the threshold, and more general observations must be made. In table 12 on page 13 the mean, median and limit values are shown. The threshold is capped between 5 and 55, but since the covariance is never 0, as the estimation is never certain, the minimum value is a little higher than that. In the figure we can see that the maximum value is only attained in the last iterations. Recalling figure 1 on the previous page, and each individual axis figure shown before, these last iterations correspond to an area where there are no sensor measurements. This means that the covariance simply increases with each iteration, never diminishing.

From figure 4a on page 15, the threshold values appear to be mainly a few specific values distributed between the minimum and around 20. This seems to be due to the update rate of the sensors. Each iteration the threshold increases to the next level and around 20 the next measurement of the sensor is used, reverting the threshold back to a minimum value, by lowering the covariance. The peaks correspond then to times where that measurement was discarded due to the innovation being larger than the threshold.

Taking a closer look at the histogram of the threshold, we can confirm what is seen in figure 4a on page 15. From the 89291 values, 95% of the threshold limits sit between the minimum value and 20. The remaining 5% are then the areas where the filter was more divergente from the sensor measurements.

This behaviour of the evolution of the threshold is, as was stated before, due to the update-rate na-

ture of the sensor measurements. It closely follows the covariance of the estimation, imposing a maximum and minimum value.

The heading threshold is severely affected by the edge cases seen before. From figure 4b on page 15 one can see that it has very high peaks. In table 14 on page 13, the mean is very high. While the minimum value is small, it seems that the threshold is systematically high. This means that, although the innovations of each sensor were on average small, refering back to table 10 on page 12, the covariance of the heading estimation is very high. It seems to be a weak point of the filter.

Looking at the histogram, in table 15 on page 13, we can see that the values are much higher than the previous measurements.

This indicates that although the innovations are small, the filter has little confidence on its estimation of the heading value.

Refering back to table 3 on page 12, the filter has small innovation in each variable of the state. It closely follows the sensors and, apart from special cases on the heading estimation, does not deviate significantly during a significant number of iterations from the trajectory. This analysis is itself an indicator only, as the actual values of each axis are not known, and the errors introduced by the sensors are difficult to understand.

6. Conclusions

The localization and navigation problem in underwater environments is still an open problem, with some classical approaches, such as the one presented in this work, and new algorithms inspired by the successes of land robots. The restrictions of the underwater environment make this problem more complex, due to the number of sensors that can be used and their cost. That being said, the underwater navigation problem can be separated in three scenarios: surface, bottom and middle sea. The surface scenario is the easiest of the three, having access to all technologies, namely GPS, for a precise localization of the Remotely Operated Vehicle (ROV). The bottom sea cames second in difficulty, as the DVL can very accurately measure the velocity relatively to a fixed point (bottom of the sea). Although the position is less accurate, having lost access to a GPS, the USBL can provide an estimation. The middle depth scenario is the most problematic of the three. The precision of the USBL does not increase signficantly, depending on the depth, and the DVL now measures the velocity relative to a moving body, the water fluid. This increases the amount of variables to determine, making the estimations more difficult to obtain. The costs associated with the underwater problem are also higher than the ones associated with land robots, either indoor or outdoor. Apart from the costs of the robot itself, there is always the necessity of a supporting vessel, where the remainder of the sensors lie and normally where the computations are made. Some sensor arrangements, such as a Long BaseLine, may require installations on the bottom of the sea. These are of course expensive operations that reduce the number of configurations one can use and try to develop an effective algorithm.

Apart from the limitation in terms of the number of sensors available and the costs of the sensors or of their installation, the GPS signals poor penetration capacity in the aquatic medium also cause other obstacles. The sensors used, namely the USBL, rely on acoustic waves. These travel at a signficant lower velocity than electromagnetic waves and that time may not be negligible. This means that at a certain iteration the measurements available may correspond to previous iterations only. For the specific problem addresses this is not problematic, as the nature of the ROV implies that it moves slowly. This means that one can safely discard the travel time of the acoustic waves.

The sensors used are also not equivalent in terms of update rate. This means that the ROV has some measurements available at more iterations than others, and there are several iterations where no measurement is available. This implies a significant portion of the estimation to be done in dead reckoning. As with the travel time, the slow velocity of the specific ROV used mitigates this particular problem.

This problem can be formulated as a localization problem. On a gaussian scenario, this problem would be classically solved with a Kalman Filter, which gives the optimal solution for said problem. Since the gaussian condition rarely holds in real world scenarios, the solution implemented was an Extended Kalman Filter. To deal with the particulars described above, the EKF was adapted to perform dead reckoning when no measurements were available, and to use measurements independently of one another. To deal with outliers a threshold based on the covariance of the estimation was implemented, rejecting severly deviated observations.

The model used to describe the trajectory of the ROV was a model that considered a trajectory with constant turning rate in the XY plane, linear motion in the Z plane and in the heading axis. The noise parameters of the model allowed for deviations of this model, and served as a factor between the importance of the dead reckoning model and the observations in the filter estimation.

In the experimental run, the filter could mostly follow the sensor measurements, with small average values of innovation for each component. The water current was approximately zero, with the DVL giving most measurements of velocity considering the

bottom of the sea. There are a few peaks where the filter was deviating significantly from the sensor values, but the increasing threshold due to the large covariance allowed the filter to regain its course. The most problematic estimation was the heading, which had large innovation values near the limits of the circle. If this where to be a more consistent case, the filter should be adapted to deal with this situation, maybe through a transformation of the sensor values to a more controlled degree interval.

The internal filter state not considering the DVL sensor values was not particulary damaging due to the low noise of the sensor and its high update rate. When DVL measurements were not available, the filter used the last available measurement, disregarding other sensor information. If the DVL had a lower update rate, or if the sensor had higher noise, this could severely damage the estimation. Again, this was not the case as the DVL is one of the most accurate sensors of the ROV and, being situated on the ROV itself, has a high update rate of around 1 measurement per second.

References

[1] Anderson, Brian D.O., Moore, John B., in *Optimal Filtering*, 1^{st} ed., New York, Doover Pub., 2005.

[2] Barkby, S., Williams, S., Pizarro, O., Jakuba, M., "Bathymetric particle filter SLAM using trajectory maps.", J. Robot Res., vol 31, no 12, pp. 1409-1430, 2012.

[3] Bellingham, J., Vaganay, J., Leonard, J., "Outlier rejection for autonomous acoustic navigation", Woods Hole Oceanographic Institution, Cambridge, MA, May, 1996.

[4] Brown, R., "Integrated Navigation systems and Kalman fitlering: A perspective.", Journal of the Institute of Navigation, vol. 19, 4, pp. 355-362, Winter 1972-1973.

[5] C100 Compass. Specifications at: http://www.emepc.pt/images/pdf/KVHC100.pdf

[6] Donovan, G., "Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter", IEEE J. OCEAN, vol. 37, no. 3, pp. 431-445 , Jul. 2012.

[7] DSP 3000 Gyroscope. Specifications available at: http://www.emepc.pt/images/pdf /KVHDSP3000.pdf

[8] Estrutura da Missão de Expansão da Plataforma Continental (2015). Available at: http://www.emepc.pt/

[9] Eustice, R., Whitcomb, L., Singh, H., & Grund, M., "Recent advances in synchronous-clock one-way-travel-time acoustic navigation", IEEE OCEANS, pp. 1-6, Sep. 2006.

[10] Eustice, R., Singh, H., Leonard, J., & Walter, M., "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters", J. Robot Res, vol 25, no 12, pp. 1223-1242, Sep. 2006.

[11] Eustice, R., "Large-area visually augmented navigation for autonomous underwater vehicles", M.S. thesis, Dept. Mec. Eng., Michigan State Univ., East Lansing, MI, 2005

[12] Henriksen, R., "The truncated second-order nonlinear filter revisited", IEEE Transactions on Automatic Control, vol.27, pp. 247-251, 1982.

[13] Howell, B., Wood, S., "Passive sonar recognition and analysis using hybrid neural networks", Prov OCEANS Conf., vol 4, pp. 1917-1924, 2003.

[14] Jalving, B., Gade, K., Hagen, O. K., & Vestgard, K., "A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system", IEEE MTS OCEANS, vol. 2, pp. 1146-1153, Sep. 2003.

[15] Kinsey, James C., Eustice, Ryan M., Whitcomb, Louis L., "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges", 2006.

[16] Kinsey, J. C., "Advances in Precision Navigation of Oceanographic Submersibles", Ph.D. dissertation, Dept. Mec. Eng., Johns Hopkins Univ., Baltimore, MD, 2006.

[17] Larsen, M., "Syntethic long baseline navigation of underwater vehicles", IEEE MTS OCEANS, vol. 3, pp. 2043-2050, Sep., 2000.

[18] Newman, P., Leonard, J., "Pure range-only sub-sea SLAM", Proc. IEEE Int. Conf. Robot Autom., vol 2, pp.1921-1926, Sep. 2003.

[19] Pascoal, A., Kaminer, L., Oliveira, P., "Navigation System Design Using Time-Varying Complementary Filters", IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, vol. 36, 4, pp. 1099-1114, Oct., 2000.

[20] Paull, L., Sacedi, S., Seto, M., & Li, H., "AUV Navigation and Localization: A Review", IEEE Journal of Oceanic Engineering, vol. 39, pp. 131-149, Dec. 2013.

[21] Peñas, A., "Positioning and Navigation Systems for Robotic Underwater Vehicles", Ph.D. dissertation, DEEC, Instituto Superior Técnico, Lisboa, Portugal, 2009.

[22] Ribas, D., Ridao, P., Tardos, J. D., Neira, J., "Underwater SLAM in man-made structured environments", J. Field Robot, vol. 25, no 11-12, pp 898-921, 2008.

[23] Ribas, D., Ridao, P., Mallios, A., Palomeras, N., "Delayed State Information Filter for USBL-Aided AUV Navigation", IEEE International Conference on Robotics and Automation, Minnesota, USA, May, 2012.

[24] Ribeiro, Maria Isabel, "Kalman and Extended Kalman Filters: Concept, Derivation and Properties", IST, Lisboa, 2004.

[25] ROV Luso technical specifications.

Available at: http://www.emepc.pt/images/pdf/O_ROV_LUSO_Descricao.pdf

[26] Simon, Dan., in *Optimal State Estimation*, 1st ed., Hoboken, New Jersey, USA/Canada, Wiley & Sons, Inc., 2006.

[27] Space and Naval Warfare Command (2015).

Available at: http://www.public.navy.mil/spawar/Pacific/Robotics/Pages/CURV.aspx

[28] TrackLink 10000 USBL Sensor.

Specifications available at: http://www.emepc.pt/images/pdf/TrackLink10000.pdf

[29] Walter, M., Eustice, R., Leonad, J., "Exactly sparse extended information filters for feature-based SLAM", Int. J. Robot. Res., vol 26, no 4, pp.335-359, 2007.

[30] Workhorse DVL. Specifications available at: http://www.emepc.pt/images/pdf/workhorseNav.pdf

7. Annex A - Experimental Data

Table 3: Innovation

	Mean	Median	Max	Min
X	0.0008245337	0	28.4922978425	-28.9512354263
Y	-0.0026811340	0	23.8132059543	-22.6653105810
Z	0.0011582780	0	9.1854007155	-10.8842648895
Heading	0.0088567783	0	418.4216264718	-390.2151369971
Turning Rate	0.0000070375	0	87.9691395753	-104.9881069366
Water Current (X)	-0.0000030799	0	0.1948231886	-0.1653758673
Water Current (Y)	-0.0000036470	0	0.2763812590	-0.2317421161
Water Current (Z)	0.0000631134	0	0.6602620267	-0.8747413794

Table 4: X Axis Innovation

Mean	Median	Max	Min
0.0008245337	0	28.4922978425	-28.9512354263

Table 5: X Axis Innovation Histogram

Bins	-10	-2	-1	-0.50	-0.10	-0.05	0	0.05	0.10	0.50	1	2	10
Value	3	71	599	1268	1130	697	7216	804	1303	1282	613	90	3

Table 6: Y Axis Innovation

Mean	Median	Max	Min
-0.0026811340	0	23.8132059543	-22.6653105810

Table 7: Y Axis Innovation Histogram

Bins	-10	-2	-1	-0.50	-0.10	-0.05	0	0.05	0.10	0.50	1	2	10
Value	4	157	747	1229	1254	732	7057	800	1166	1164	655	110	4

Table 8: Z Axis Innovation

Mean	Median	Max	Min
0.0011582780	0	9.1854007155	-10.8842648895

Table 9: Z Axis Innovation Histogram

Bins	-10	-2	-1	-0.50	-0.10	-0.05	0	0.05	0.10	0.50	1	2	10
Value	1	4	123	1052	1283	236	333	252	1252	1069	119	15	8

Table 10: Heading Innovation

Mean	Median	Max	Min
0.0088567783	0	418.4216264718	-390.2151369971

Table 11: Heading Innovation Histogram

Bins	-10	-2	-1	-0.50	-0.10	-0.05	0	0.05	0.10	0.50	1	2	10
Value	67	709	876	627	1044	933	6351	898	1134	720	916	751	53

Table 12: Position Threshold

Mean	Median	Max	Min
14.6682801442	13.1646615721	55.0000000000	5.9373129082

Table 13: Position Threshold Histogram

Bins	6	7	9	13	16	20	25	30	50	100
Value	5747	10505	16652	17429	20563	13896	1225	288	2986	0

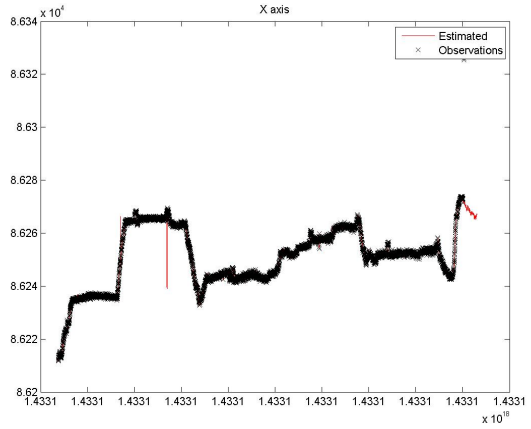
Table 14: Heading Threshold

Mean	Median	Max	Min
381.4702864541	350.0751030035	1471.2055903613	5.0397567123

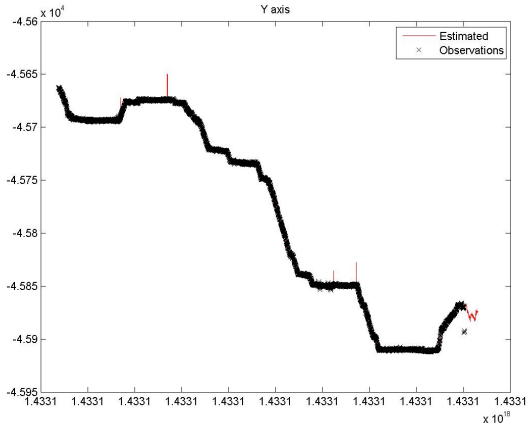
Table 15: Heading Threshold Histogram

Bins	Value
0	3553
45	7367
90	5380
135	7081
180	6728
225	3855
270	6156
315	3776
360	2347
360	2757
405	3573
450	4962
495	4082
540	3765
585	4168
630	3067
675	3509
720	1394
720	1025
765	3183
810	1845
855	2280
900	912
945	726
990	848
1035	240
1080	712

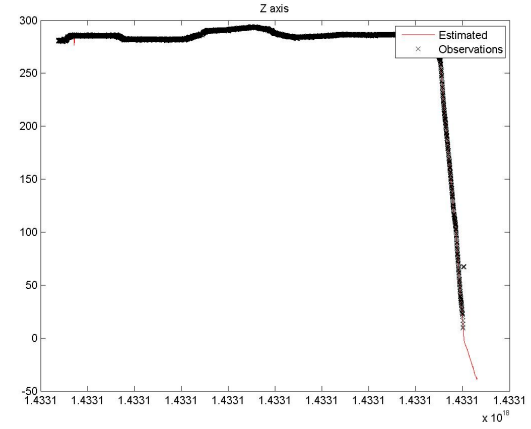
8. Annex B - Figures



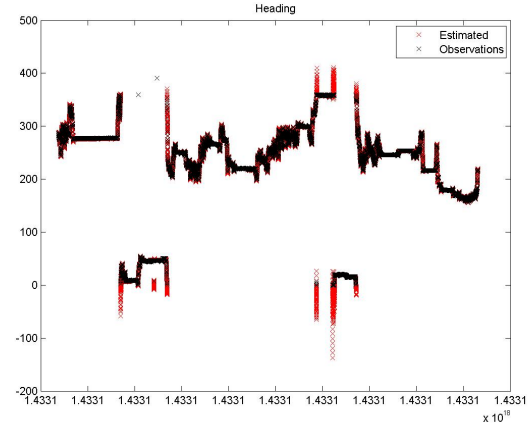
(a) X Axis



(b) Y Axis

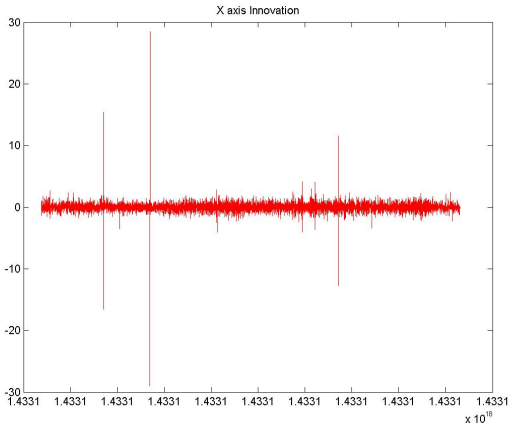


(c) Z Axis

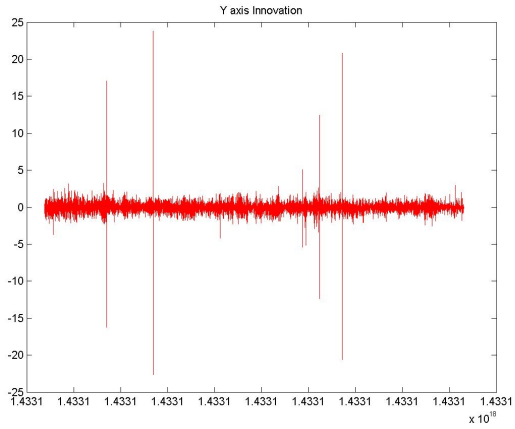


(d) Heading

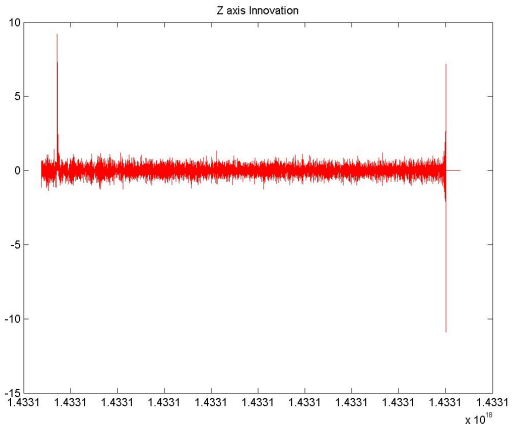
Figure 2: Experimental Results per Axis



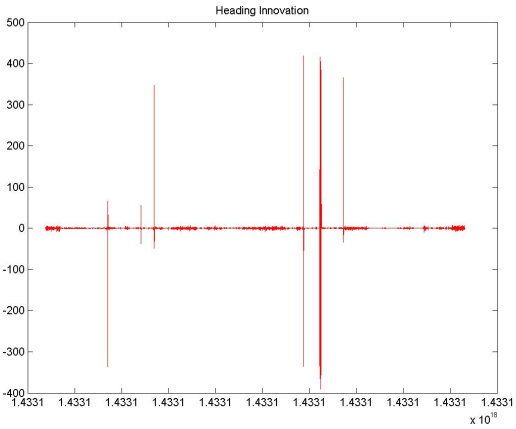
(a) X Axis Innovation



(b) Y Axis Innovation

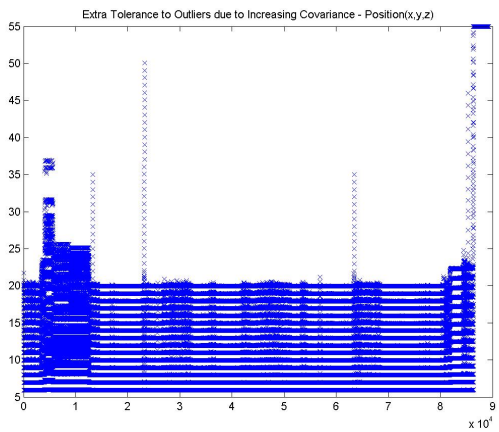


(c) Z Axis Innovation

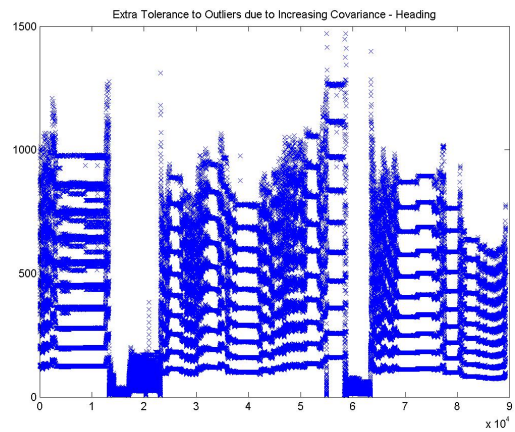


(d) Heading Innovation

Figure 3: Innovation



(a) Position Threshold



(b) Heading Threshold

Figure 4: Threshold