International Journal of Robotics and Automation (IJRA)
Vol. 3, No. 3, September 2014, pp. 168~183
ISSN: 2089-4856

Filtering Method for Location Estimation of an Underwater Robot

Nak Yong Ko*[1], Tae Gyun Kim[2]
[1]Dept. Electronics Eng., Chosun University, Korea
[2]Dept. Control & Instrumentation Eng., Chosun University, Korea

Article Info

Article history:

Received Mar 12, 2014
Revised May 12, 2014
Accepted Jun 10, 2014

Keyword:

Collective measurement update
Extended Kalman filter
Filtering
Individual measurement update
Localization
Underwater robot

ABSTRACT

This paper describes an application of extended Kalman filter (EKF) for localization of an underwater robot. For the application, linearized model of robot motion and sensor measurement are derived. Like usual EKF, the method is recursion of two main steps: the time update (or prediction) and measurement update. The measurement update uses exteroceptive sensors such as four acoustic beacons and a pressure sensor. The four beacons provide four range data from these beacons to the robot and pressure sensor does the depth data of the robot. One of the major contributions of the paper is suggestion of two measurement update approaches. The first approach corrects the predicted states using the measurement data individually. The second one corrects the predicted state using the measurement data collectively. The simulation analysis shows that EKF outperforms least squares or odometry based dead-reckoning in the precision and robustness of the estimation. Also, EKF with collective measurement update brings out better accuracy than the EKF with individual measurement update.

Corresponding Author:

Nak Yong Ko
Dept. Electronics Eng., Chosun University, Korea
Email: nyko@chosun.ac.kr

1. INTRODUCTION

Knowing the location and orientation is vital for navigation of an underwater mobile robot [1-2]. Localization is also needed for map building, decision making, exploration, environment monitoring, and object manipulation in underwater environment [3-4].

There have been several technologies for underwater localization. Inertial navigation aided by GPS (Global Positioning System) was one of the practical methods. This method uses inertial navigation technology when the robot navigates underwater. On the surface, it uses GPS to fix the bias accumulated through the dead-reckoning. It uses IMU (Inertial Measurement Unit) and DVL (Doppler Velocity Log) for dead-reckoning, and corrects accumulated location error using GPS when the robot surfaces once in a while [5]. This method requires frequent surfacing only for localization which consumes time and energy. Also the DVL data is not available if the robot gets out of bottom tracking range when surfacing, thus lets the robot lose track of the location.

Another methods use distance and/or bearing of the robot from acoustic beacons. The acoustic beacon systems such as USBL (Ultra Short Base Line), SBL (Short Base Line), and LBL (Long Base Line) provide locations information through trilateration or triangulation along with least squares method. Unlike the dead-reckoning in inertial navigation, they don't accumulate error since they rely only on the information relative to beacons whose location is given in advance. However, they require expensive acoustic beacon systems and extensive calibration efforts. Besides, they are available when the robot is within some limited range from the beacons.

Other approaches appropriate for using both the dead-reckoning and ranges from beacons are suggested. These approaches are based on Bayes filtering method. They usually use particle filter [6-10] or Kalman filter [11] methodology. Generally, the method can fuse data from several exteroceptive sensors and internal motion information. Also, they can be used for SLAM (Simultaneous Localization and Mapping)[12-13]. They have been used widely for localization and SLAM of ground robot or indoors robot, and it was extended to underwater localization [14]. It is generally known that the particle filter produces more precise and robust estimation than the Kalman filter while it requires more extensive calculations. In cases where computation time is critical, Kalman filter approach is more feasible than particle filter [15].

This paper develops an EKF based method for localization of an unmanned underwater robot. Though the paper adopts EKF which is prevalent approach for estimation and has hundreds of variants [16], the paper has the following contributions. It derives formulations for application of the EKF approach for localization of an underwater robot and investigates the collective application and individual application of the measurement update. Applications of EKF for underwater localization are relatively few and derivation of Jacobian matrices for the implementation has not been clearly revealed yet. Also, there has not been clear distinction between the collective application and individual application of the measurement update.

This paper derives and applies Kalman filter algorithm for underwater localization in the section 2. The data of depth and ranges from beacons are fused together with the velocity or odometry information which is obtained internally from the robot motion. In the section 3, the proposed method is simulated and compared with least squares method and dead-reckoning. Section 4 concludes the paper.

2. LOCATION ESTIMATION BY EXTENDED KALMAN FILTER (EKF)

The proposed method follows conventional approach of Kalman filtering method consisting of two recursive steps: prediction of location using internal motion information and correction by measurement relative to external environment. Table 1 depicts pseudo code of the localization method. The procedure repeats at every time step using the estimation result from the previous time step. The procedure produces two estimations: the location $\boldsymbol{X}_t$ and covariance $\boldsymbol{\Sigma}_t$ of the estimated location uncertainty. Along with the location estimation $\boldsymbol{X}_{t-1}$ and covariance estimation Σ_{t-1} at time t-1, the information on robot motion $\boldsymbol{u}_t$ which is fed by internal sensors such as IMU or odometer sensors are used for prediction of the robot location $\bar{\boldsymbol{X}}_t$ and covariance $\bar{\boldsymbol{\Sigma}}_t$ at time t. This step is described on the line 1 of the Table 1. The predicted robot location $\bar{\boldsymbol{X}}_t$ and covariance $\bar{\boldsymbol{\Sigma}}_t$ is corrected at the line 2. The correction step uses measurement $\boldsymbol{z}_t$ related to the landmarks, the identification of the landmark $\boldsymbol{c}_t$, and the data on the landmark $\boldsymbol{E}_t$ given beforehand. The landmark data $\boldsymbol{E}_t$ specifically refers to the location of the landmarks. Detailed derivation of the two steps of prediction and correction will be described in the following sections.

Table 1. Procedure for EKF location estimation

Localization EKF$(\mathbf{X}_{t-1}, \Sigma_{t-1}, \boldsymbol{u}_t, \mathbf{z}_t, \mathbf{c}_t, \mathbf{E})$
1. $\bar{\mathbf{X}}_t, \bar{\Sigma}_t = \textit{Prediction step}(\mathbf{X}_{t-1}, \Sigma_{t-1}, \boldsymbol{u}_t)$
2. $\mathbf{X}_t, \Sigma_t = \textit{Correction step}(\bar{\mathbf{X}}_t, \bar{\Sigma}_t, \mathbf{z}_t, \mathbf{c}_t, \mathbf{E})$
3. *return* $\mathbf{X}_t, \Sigma_t$

The Figure 1 shows a simple example of the estimation result for robot location and covariance. The robot navigates through planar trajectory indicated by the bold line segments and four TOA (Time of arrival)'s are used. Bi (i = 1, 2, 3, 4) represents an acoustic beacon. Arcs indicate the range measurement data of the robot from the beacons. Estimated locations $\boldsymbol{X}_t$'s are marked together with ellipse around the location which indicates covariance $\boldsymbol{\Sigma}_t$ of the estimation error.

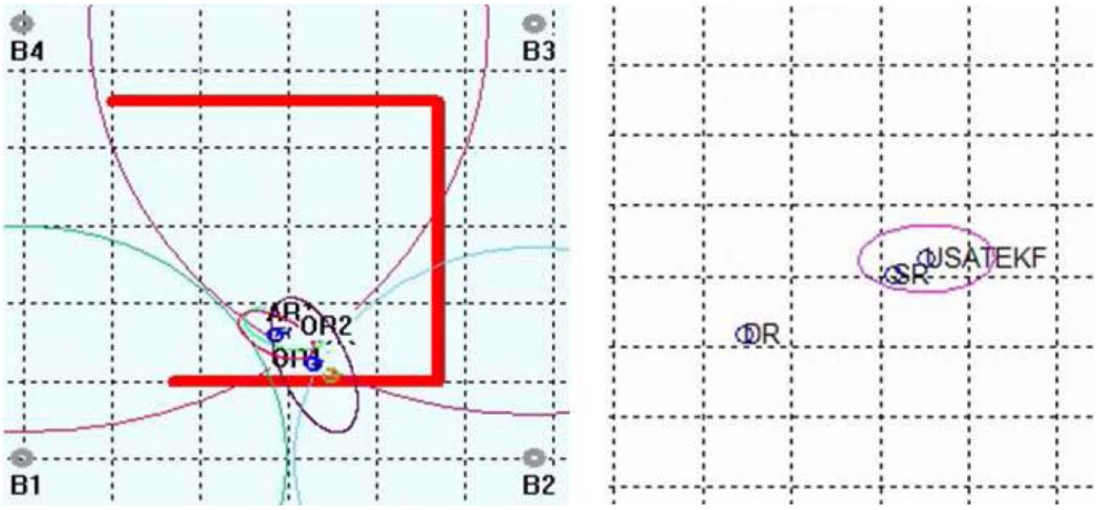


Figure 1. An example of localization using EKF

2.1. Prediction

The prediction step updates the location and covariance of the estimated locations using the velocity information of the robot. The velocity can be sensed using the accelerometer, gyroscope, and odometry sensors or be calculated from the motion command to the actuator. The prediction of the robot location is described as the state transition equation (1).

$$\mathbf{x}_t = g(\boldsymbol{u}_t, \mathbf{X}_{t-1}) = \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ \phi_{t-1} \\ \theta_{t-1} \\ \psi_{t-1} \end{pmatrix} + \begin{pmatrix} uc\theta c\psi\Delta t + vs\phi s\theta c\psi\Delta t - vc\phi s\psi\Delta t + wc\phi s\theta c\psi\Delta t + ws\phi s\psi\Delta t \\ uc\theta s\psi\Delta t + vs\phi s\theta s\psi\Delta t + vc\phi c\psi\Delta t + ws\phi s\theta s\psi\Delta t - ws\phi c\psi\Delta t \\ -us\theta\Delta t + vs\phi c\theta\Delta t + wc\phi c\theta\Delta t \\ p\Delta t + qs\phi t\theta\Delta t + rc\phi t\theta\Delta t \\ qc\phi\Delta t - rs\phi\Delta t \\ qs\phi\sec\theta\Delta t + rc\phi\sec\theta\Delta t \end{pmatrix} \tag{1}$$

In (1), $\boldsymbol{u}_t = (u, v, w, p, q, r)$ is the velocity of the robot in 3-dimensional underwater environment with respect to the body fixed frame. $\boldsymbol{X}_t = (x, y, z, \phi, \theta, \psi)$ is the position and orientation of the robot with respect to an Earth-fixed and inertial coordinate frame. $\boldsymbol{u}_t$ and $\boldsymbol{X}_t$ are represented according to the common notations from SNAME(Society of Naval Architects and Marine Engineers). Δt is the time difference between the two consecutive sampling time t-1to t. The prediction of the covariance is subject to the equation (2).

$$\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \tag{2}$$

In (2), $\boldsymbol{G}_t$ and $\boldsymbol{V}_t$ are the Jacobian of the $g(\boldsymbol{u}_t, \boldsymbol{X}_{t-1})$ with respect to the state $\boldsymbol{X}_{t-1}$ and $\boldsymbol{u}_t$ respectively. $\boldsymbol{M}_t$ is the error covariance of the velocity $\boldsymbol{u}_t$. The following equations show how the Jacobian $\boldsymbol{G}_t$ and $\boldsymbol{V}_t$ are derived. In the derivation, for notational simplicity, the subscripts t-1 representing the time index in ϕ_{t-1}, θ_{t-1}, and ψ_{t-1} are deleted. The $\boldsymbol{G}_t$ is derived as the following.

$$G_t = \frac{\partial g(\boldsymbol{u}_t, \mathbf{X}_{t-1})}{\partial \mathbf{x}_{t-1}} = \begin{pmatrix} \frac{\partial x'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial x'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial x'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial x'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial x'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial x'}{\partial \mathbf{X}_{t-1,\psi}} \\ \frac{\partial y'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial y'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial y'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial y'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial y'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial y'}{\partial \mathbf{X}_{t-1,\psi}} \\ \frac{\partial z'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial z'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial z'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial z'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial z'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial z'}{\partial \mathbf{X}_{t-1,\psi}} \\ \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial \phi'}{\partial \mathbf{X}_{t-1,\psi}} \\ \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial \theta'}{\partial \mathbf{X}_{t-1,\psi}} \\ \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,x}} & \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,y}} & \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,z}} & \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,\phi}} & \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,\theta}} & \frac{\partial \psi'}{\partial \mathbf{X}_{t-1,\psi}} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & G_{1,t}(1,1) & G_{1,t}(2,1) & G_{1,t}(3,1) \\ 0 & 1 & 0 & G_{2,t}(1,1) & G_{2,t}(2,1) & G_{2,t}(3,1) \\ 0 & 0 & 1 & vc\phi c\theta\Delta t - ws\phi c\theta\Delta t & -uc\theta\Delta t - vs\phi s\theta\Delta t - wc\phi s\theta\Delta t & 0 \\ 0 & 0 & 0 & 1 + qc\phi t\theta\Delta t - rs\phi t\theta\Delta t & qs\phi\sec^2\theta\Delta t + rc\phi\sec^2\theta\Delta t & 0 \\ 0 & 0 & 0 & -qs\phi\Delta t - rc\phi\Delta t & 1 & 0 \\ 0 & 0 & 0 & qc\phi\sec\theta\Delta t - rs\phi\sec\theta\Delta t & qs\phi\sec\theta t\theta\Delta t + rc\phi\sec\theta t\theta\Delta t & 1 \end{pmatrix} \quad (3)$$

The $\boldsymbol{G}_{1,\mathrm{t}}$ and $\boldsymbol{G}_{2,\mathrm{t}}$ are as the followings.

$$G_{1,\mathrm{t}} = \begin{pmatrix} vc\,\phi s\,\theta c\,\psi\Delta t + vs\,\phi s\,\psi\Delta t - ws\,\phi s\,\theta c\,\psi\Delta t + wcs\ \psi\Delta t \\ -us\,\theta c\,\psi\Delta t + vs\,\phi c\,\theta c\,\psi\Delta t + wc\,\phi c\,\theta c\,\psi\Delta t \\ -uc\,\theta s\,\psi\Delta t - vs\,\phi s\,\theta s\,\psi\Delta t - vc\,\phi c\,\psi\Delta t - wc\,\phi s\,\theta s\,\psi\Delta t + ws\,\phi c\,\psi\Delta t \end{pmatrix}$$

$$G_{2,\mathrm{t}} = \begin{pmatrix} vc\,\phi s\,\theta s\,\psi\Delta t - vs\,\phi c\,\psi\Delta t - ws\,\phi s\,\theta s\,\psi\Delta t - wc\,\phi c\,\psi\Delta t \\ -us\,\theta s\,\psi\Delta t + vs\,\phi c\,\theta s\,\psi\Delta t + wc\,\phi c\,\theta s\,\psi\Delta t \\ uc\,\theta c\,\psi\Delta t + vs\,\phi s\,\theta c\,\psi\Delta t - vc\,\phi s\,\psi\Delta t + wc\,\phi s\,\theta c\,\psi\Delta t + ws\,\phi s\,\psi\Delta t \end{pmatrix} \quad (4)$$

The Jacobian $\boldsymbol{V}_{\mathrm{t}}$ which associates the location at time t to the velocity $\boldsymbol{u}_{\mathrm{t}}$ is derived as the following.

$$V_t = \frac{\partial g(\boldsymbol{u}_t, \mathbf{X}_{t-1})}{\partial \boldsymbol{u}_t} = \begin{pmatrix} \frac{\partial x'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial x'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial x'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial x'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial x'}{\boldsymbol{u}_{t,q}} & \frac{\partial x'}{\partial \boldsymbol{u}_{t,r}} \\ \frac{\partial y'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial y'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial y'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial y'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial y'}{\boldsymbol{u}_{t,q}} & \frac{\partial y'}{\partial \boldsymbol{u}_{t,r}} \\ \frac{\partial z'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial z'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial z'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial z'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial z'}{\boldsymbol{u}_{t,q}} & \frac{\partial z'}{\partial \boldsymbol{u}_{t,r}} \\ \frac{\partial \phi'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial \phi'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial \phi'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial \phi'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial \phi'}{\boldsymbol{u}_{t,q}} & \frac{\partial \phi'}{\partial \boldsymbol{u}_{t,r}} \\ \frac{\partial \theta'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial \theta'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial \theta'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial \theta'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial \theta'}{\boldsymbol{u}_{t,q}} & \frac{\partial \theta'}{\partial \boldsymbol{u}_{t,r}} \\ \frac{\partial \psi'}{\partial \boldsymbol{u}_{t,u}} & \frac{\partial \psi'}{\partial \boldsymbol{u}_{t,v}} & \frac{\partial \psi'}{\partial \boldsymbol{u}_{t,w}} & \frac{\partial \psi'}{\partial \boldsymbol{u}_{t,p}} & \frac{\partial \psi'}{\boldsymbol{u}_{t,q}} & \frac{\partial \psi'}{\partial \boldsymbol{u}_{t,r}} \end{pmatrix}$$

$$= \begin{pmatrix} c\theta c\psi\Delta t & s\phi s\theta c\psi\Delta t - c\phi s\psi\Delta t & c\phi s\theta c\psi\Delta t + s\phi s\psi\Delta t & 0 & 0 & 0 \\ c\theta s\psi\Delta t & s\phi s\theta s\psi\Delta t + c\phi c\psi\Delta t & c\phi s\theta s\psi\Delta t - s\phi c\psi\Delta t & 0 & 0 & 0 \\ -s\theta\Delta t & s\phi c\theta\Delta t & c\phi c\theta\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t & s\phi t\theta\Delta t & c\phi t\theta\Delta t \\ 0 & 0 & 0 & 0 & c\phi\Delta t & s\phi\Delta t \\ 0 & 0 & 0 & 0 & s\phi\sec\theta\Delta t & c\phi\sec\theta\Delta t \end{pmatrix} \quad (5)$$

The error covariance $\boldsymbol{M}_t$ of the velocity $\boldsymbol{u}_t$ is assumed to be diagonal for the computational convenience. It implies that the linear and angular velocity in each direction has no correlation with the other components of the velocity.

$$M_t = \begin{pmatrix} P_t(1,1) & 0 & 0 & 0 & 0 & 0 \\ 0 & P_t(2,1) & 0 & 0 & 0 & 0 \\ 0 & 0 & P_t(3,1) & 0 & 0 & 0 \\ 0 & 0 & 0 & P_t(4,1) & 0 & 0 \\ 0 & 0 & 0 & 0 & P_t(5,1) & 0 \\ 0 & 0 & 0 & 0 & 0 & P_t(6,1) \end{pmatrix} \tag{6}$$

where,

$$P_t = \begin{pmatrix} (\alpha_{uu}|u| + \alpha_{uv}|v| + \alpha_{uw}|w| + \alpha_{up}|p| + \alpha_{uq}|q| + \alpha_{ur}|r| + \alpha_{us})^2 \\ (\alpha_{vu}|u| + \alpha_{vv}|v| + \alpha_{vw}|w| + \alpha_{vp}|p| + \alpha_{vq}|q| + \alpha_{vr}|r| + \alpha_{vs})^2 \\ (\alpha_{wu}|u| + \alpha_{wv}|v| + \alpha_{ww}|w| + \alpha_{wp}|p| + \alpha_{vq}|q| + \alpha_{wr}|r| + \alpha_{ws})^2 \\ (\alpha_{pu}|u| + \alpha_{pv}|v| + \alpha_{pw}|w| + \alpha_{pp}|p| + \alpha_{pq}|q| + \alpha_{pr}|r| + \alpha_{ps})^2 \\ (\alpha_{qu}|u| + \alpha_{qv}|v| + \alpha_{qw}|w| + \alpha_{qp}|p| + \alpha_{qq}|q| + \alpha_{qr}|r| + \alpha_{qs})^2 \\ (\alpha_{ru}|u| + \alpha_{rv}|v| + \alpha_{rw}|w| + \alpha_{rp}|p| + \alpha_{rq}|q| + \alpha_{rr}|r| + \alpha_{rs})^2 \end{pmatrix}$$

In the equation (6), the parameter α_{v1v2} relates the velocity v_2 to the uncertainty of the velocity v_1. The parameter α_{v1s} addresses the uncertainty of velocity v_1 when the robot stays still.
Table 2 shows the algorithm for prediction of the robot location and error covariance. It corresponds to the line 1 of the Table 1. Lines 3 to 5 calculate the Jacobian $\boldsymbol{G}_t$ which projects the estimated robot location at t-1 to the a priori location at time t. Line 6 calculates the Jacobian $\boldsymbol{V}_t$ which maps the velocity $\boldsymbol{u}_t$ to the a priori location at time t. Lines 7 and 8 provides the error covariance $\boldsymbol{M}_t$ of the velocity $\boldsymbol{u}_t$. Line 9 transforms the linear velocity and angular velocity represented with respect to the body fixed frame to those represented with respect to the Earth-fixed and inertial coordinate frame. $\boldsymbol{T}_{E1}$ and $\boldsymbol{T}_{E2}$ are the Euler transformation matrices relating the body fixed frame velocity to Earth-fixed frame velocity. $\boldsymbol{T}_{E1}$ is for transformation of linear velocity and $\boldsymbol{T}_{E2}$ is for angular velocity.

$$\boldsymbol{T}_{E1} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \tag{7}$$

$$T_{E2} \underline{\underline{\Delta}} = \begin{pmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{pmatrix} \tag{8}$$

Finally, lines 10 and 11 yield a priori estimation of robot location and error covariance at time t.

Table 2. Prediction of a priori robot location and error covariance from previous estimates at time t-1.

===

Prediction step$(\mathbf{X}_{t-1}, \Sigma_{t-1}, \boldsymbol{u}_t)$

===

1: $\phi = \mathbf{X}_{t-1,\phi},\ \ \theta = \mathbf{X}_{t-1,\theta},\ \ \psi = \mathbf{X}_{t-1,\psi}$

2: $u = \boldsymbol{u}_{t,u},\ v = \boldsymbol{u}_{t,v},\ w = \boldsymbol{u}_{t,w},\ \ p = \boldsymbol{u}_{t,p},\ q = \boldsymbol{u}_{t,q},\ r = \boldsymbol{u}_{t,r}$

3: $$G1_t = \begin{pmatrix} vc\phi s\theta c\psi\Delta t + vs\phi s\psi\Delta t - ws\phi s\theta c\psi\Delta t + wcs\psi\Delta t \\ -us\theta c\psi\Delta t + vs\phi c\theta c\psi\Delta t + wc\phi c\theta c\psi\Delta t \\ -uc\theta s\psi\Delta t - vs\phi s\theta s\psi\Delta t - vc\phi c\psi\Delta t - wc\phi s\theta s\psi\Delta t + ws\phi c\psi\Delta t \end{pmatrix}$$

4: $$G2_t = \begin{pmatrix} vc\phi s\theta s\psi\Delta t - vs\phi c\psi\Delta t - ws\phi s\theta s\psi\Delta t - wc\phi c\psi\Delta t \\ -us\theta s\psi\Delta t + vs\phi c\theta s\psi\Delta t + wc\phi c\theta s\psi\Delta t \\ uc\theta c\psi\Delta t + vs\phi s\theta c\psi\Delta t - vc\phi s\psi\Delta t + wc\phi s\theta c\psi\Delta t + ws\phi s\psi\Delta t \end{pmatrix}$$

5: $$G_t = \begin{pmatrix} 1 & 0 & 0 & G1_t(1,1) & G1_t(2,1) & G1_t(3,1) \\ 0 & 1 & 0 & G2_t(1,1) & G2_t(2,1) & G2_t(3,1) \\ 0 & 0 & 1 & vc\phi c\theta\Delta t - ws\phi c\theta\Delta t & -uc\theta\Delta t - vs\phi s\theta\Delta t - wc\phi s\theta\Delta t & 0 \\ 0 & 0 & 0 & 1 + qc\phi t\theta\Delta t - rs\phi t\theta\Delta t & qs\phi\sec^2\theta\Delta t + rc\phi\sec^2\theta\Delta t & 0 \\ 0 & 0 & 0 & -qs\phi\Delta t - rc\phi\Delta t & 1 & 0 \\ 0 & 0 & 0 & qc\phi\sec\theta\Delta t - rs\phi\sec\theta\Delta t & qs\phi\sec\theta t\theta\Delta t + rc\phi\sec\theta t\theta\Delta t & 1 \end{pmatrix}$$

6: $$V_t = \begin{pmatrix} c\theta c\psi\Delta t & s\phi s\theta c\psi\Delta t - c\phi s\psi\Delta t & c\phi s\theta c\psi\Delta t + s\phi s\psi\Delta t & 0 & 0 & 0 \\ c\theta s\psi\Delta t & s\phi s\theta s\psi\Delta t + c\phi c\psi\Delta t & c\phi s\theta s\psi\Delta t - s\phi c\psi\Delta t & 0 & 0 & 0 \\ -s\theta\Delta t & \sin\phi\cos\theta\Delta t & c\phi c\theta\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta t & s\phi t\theta\Delta t & c\phi t\theta\Delta t \\ 0 & 0 & 0 & 0 & c\phi\Delta t & s\phi\Delta t \\ 0 & 0 & 0 & 0 & s\phi\sec\theta\Delta t & c\phi\sec\theta\Delta t \end{pmatrix}$$

7. $$P_t = \begin{pmatrix} (\alpha_{uu}|u| + \alpha_{uv}|v| + \alpha_{uw}|w| + \alpha_{up}|p| + \alpha_{uq}|q| + \alpha_{ur}|r| + \alpha_{us})^2 \\ (\alpha_{vu}|u| + \alpha_{vv}|v| + \alpha_{vw}|w| + \alpha_{vp}|p| + \alpha_{vq}|q| + \alpha_{vr}|r| + \alpha_{vs})^2 \\ (\alpha_{wu}|u| + \alpha_{wv}|v| + \alpha_{ww}|w| + \alpha_{wp}|p| + \alpha_{vq}|q| + \alpha_{wr}|r| + \alpha_{ws})^2 \\ (\alpha_{pu}|u| + \alpha_{pv}|v| + \alpha_{pw}|w| + \alpha_{pp}|p| + \alpha_{pq}|q| + \alpha_{pr}|r| + \alpha_{ps})^2 \\ (\alpha_{qu}|u| + \alpha_{qv}|v| + \alpha_{qw}|w| + \alpha_{qp}|p| + \alpha_{qq}|q| + \alpha_{qr}|r| + \alpha_{qs})^2 \\ (\alpha_{ru}|u| + \alpha_{rv}|v| + \alpha_{rw}|w| + \alpha_{rp}|p| + \alpha_{rq}|q| + \alpha_{rr}|r| + \alpha_{rs})^2 \end{pmatrix}$$

8: $$M_t = \begin{pmatrix} P_t(1,1) & 0 & 0 & 0 & 0 & 0 \\ 0 & P_t(2,1) & 0 & 0 & 0 & 0 \\ 0 & 0 & P_t(3,1) & 0 & 0 & 0 \\ 0 & 0 & 0 & P_t(4,1) & 0 & 0 \\ 0 & 0 & 0 & 0 & P_t(5,1) & 0 \\ 0 & 0 & 0 & 0 & 0 & P_t(6,1) \end{pmatrix}$$

9: $(\dot{x}, \dot{y}, \dot{z})^T = T_{E1}(u,v,w)^T,\ \ (\dot{\phi}, \dot{\theta}, \dot{\psi})^T = T_{E2}(p,q,r)^T$

10: $\overline{\mathbf{X}}_t = \mathbf{X}_{t-1} + (\dot{x}\ \ \dot{y}\ \ \dot{z}\ \ \dot{\phi}\ \ \dot{\theta}\ \ \dot{\psi})^T \Delta t$

11: $\overline{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T$

12: *return* $\overline{\mathbf{X}}_t, \overline{\Sigma}_t$

===

2.2. Correction of the a Priori Estimates

The correction stage which is also called the measurement update corrects the a priori estimates of the robot location and error covariance. While the prediction stage uses only the internal information of robot velocity and previous estimates, the correction stage uses the measurement information relative to external environment to adjust the a priori estimates. In our application the external environment refers to the acoustic beacons. After the beacons emit acoustic signal, the hydrophone[17] at the robot receives the acoustic signals and calculates the distance between the hydrophones and the robot using the TOA(time of arrival) of the acoustic signals. Also, the method uses depth of the robot from the surface which is detected by a pressure sensor. The following equations are used for correction stage.

$$\begin{aligned}
S_t &= H_t \overline{\Sigma}_t [H_t]^T + Q_t \\
K_t &= \overline{\Sigma}_t [H_t]^T [S_t]^{-1} \\
\mathbf{X}_t &= \overline{\mathbf{X}}_t + K_t(z_t - \hat{z}_t) \\
\Sigma_t &= (I - K_t H_t)\overline{\Sigma}_t
\end{aligned} \tag{9}$$

The matrix $\boldsymbol{H}_t$ is the Jacobian which relates the robot location to the measurement. $\boldsymbol{Q}_t$ is the error covariance of the measurement process. The process calculates the Kalman gain $\boldsymbol{K}_t$ and uses it for the correction of the a priori estimate $\overline{\boldsymbol{X}}_t$ to $\boldsymbol{X}_t$, and $\overline{\boldsymbol{\Sigma}}_t$ to $\boldsymbol{\Sigma}_t$. We apply the correction step in two ways: applying the procedure for each measurement individually in sequence and applying it for all the measurements at once collectively. The two application approaches are explained in the following section and they are tested in the simulations.

2.2.1. Dealing with Range Data Individually: Correcting the Prediction using Only One Data at a Time

The predicted location can be corrected every time a measurement data is available. A data of range from a beacon or the depth data by the pressure sensor can be used for correction. Measurement model $h_{TOA}(\cdot)$ for case of range from a beacon and the model $h_{Depth}(\cdot)$ for the case of depth are described by the following formulas.

$$\begin{aligned}
z^i_{t,TOA} &= h(\overline{\boldsymbol{X}}_t, j, \boldsymbol{E}) = \begin{pmatrix} r^i_{t,TOA} \\ s^i_{t,TOA} \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{(\boldsymbol{E}_{j,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{j,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{j,z} - \overline{\boldsymbol{X}}_{t,z})^2} \\ \boldsymbol{E}_{j,s} \end{pmatrix}
\end{aligned} \tag{10}$$

$$z_{t,Depth} = h\left(\overline{\boldsymbol{X}}_t\right) = \overline{\boldsymbol{X}}_{t,z} \tag{11}$$

$z^i_{t,TOA}$ is the data related to the i-th beacon where $r^i_{t,TOA}$ is the distance from the i-th beacon to the robot and $s^i_{t,TOA}$ is the signature for the measurement. ($E_{i,x}$, $E_{i,y}$, $E_{i,z}$) is the coordinate of the i-th beacon and ($\overline{\boldsymbol{X}}_{t,x}$, $\overline{\boldsymbol{X}}_{t,y}$, $\overline{\boldsymbol{X}}_{t,z}$) is the location of the robot at time t. $Z_{t,Depth}$ is the depth data. The Jacobians for the measurement model needed for application of EKF are derived from the equations (10) and (11). $H^i_{t,TOA}$ and $H_{t,Depth}$ in the equations (12) and (13) correspond to the linearization of $h_{TOA}(\cdot)$ and $h_{Depth}(\cdot)$, respectively.

$$\begin{aligned}
H^i_t &= \frac{\partial h(\overline{\boldsymbol{X}}_t, j, \boldsymbol{E})}{\partial \boldsymbol{x}_t} \\
&= \begin{pmatrix}
\frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial r^i_t}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \\
\frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial s^i_t}{\partial \overline{\boldsymbol{X}}_{t,\psi}}
\end{pmatrix} \\
&= \begin{pmatrix}
-\left(\frac{\boldsymbol{E}_{j,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q}}\right) & -\left(\frac{\boldsymbol{E}_{j,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q}}\right) & -\left(\frac{\boldsymbol{E}_{j,z} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q}}\right) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{aligned} \tag{12}$$

$$\sqrt{q} = \sqrt{(\boldsymbol{E}_{j,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{j,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{j,z} - \overline{\boldsymbol{X}}_{t,z})^2}$$

$$\begin{aligned} H_{t,Depth} &= \frac{\partial h(\overline{\boldsymbol{X}}_t)}{\partial \boldsymbol{x}_t} \\ &= \begin{pmatrix} \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{13}$$

Table 3 and 4 show the correction procedure for the measurement of distance from a beacon and for the measurement of depth respectively. Line 6 of the Table 3 uses the equation (12), and line 3 of the Table 4 does the equation (13). They follow the usual EKF procedure described in the equation (9).

In the Table 3, $z^i_t = (r^i_t, s^i_t)$ refers to the distance r^i_t from the i-th beacon and the signature s^i_t of the measurement. In the Table 4, z_t represents depth data. It is notable that in case of correction by range data, the location $\boldsymbol{E}_i$ of the i-th beacon is needed as well as the range data z^i_t from the beacon $\boldsymbol{E}_i$. These Tables return the final estimation of the robot location and the error covariance $\boldsymbol{\Sigma}_t$ of the location estimation.

Table 3. Procedure for the correction using the range data z^i_t from a beacon $\boldsymbol{E}_i$

Correction step on TOA($\overline{\boldsymbol{X}}_t, \overline{\Sigma}_t, z_t, c_t, \boldsymbol{E}$)

1: $Q_t = \begin{pmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_s^2 \end{pmatrix}$

2: *for all observed features of TOA* $z^i_{t,TOA} = \begin{pmatrix} r^i_t & s^i_t \end{pmatrix}^T$ *do*

3: $j = c^i_t$

4: $q = (\boldsymbol{E}_{j,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{j,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{j,z} - \overline{\boldsymbol{X}}_{t,z})^2$

5: $\hat{z}^i_{t,TOA} = \begin{pmatrix} \sqrt{q} \\ \boldsymbol{E}_{j,s} \end{pmatrix}$

6: $H^i_t = \begin{pmatrix} -\left(\frac{\boldsymbol{E}_{j,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q}}\right) & -\left(\frac{\boldsymbol{E}_{j,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q}}\right) & -\left(\frac{\boldsymbol{E}_{j,z} - \overline{\boldsymbol{X}}_{t,z}}{\sqrt{q}}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

7: $S^i_t = H^i_t \ \overline{\Sigma}_t \ \left[H^i_t\right]^T + Q_t$

8: $K^i_t = \overline{\Sigma}_t \ \left[H^i_t\right]^T \ \left[S^i_t\right]^{-1}$

9: $\overline{\boldsymbol{X}}_t = \overline{\boldsymbol{X}}_t + K^i_t(z^i_{t,TOA} - \hat{z}^i_{t,TOA})$

10: $\overline{\Sigma}_t = (I - K^i_t H^i_t)\overline{\Sigma}_t$

11: *endfor*

12: $\overline{\boldsymbol{X}}_{t,TOA} = \overline{\boldsymbol{X}}_t, \ \overline{\Sigma}_{t,TOA} = \overline{\Sigma}_t$

13: *return* $\overline{\boldsymbol{X}}_{t,TOA}, \overline{\Sigma}_{t,TOA}$

Table 4. Procedure for the correction using the depth measurement.

=====================================

Correction step on depth$(\overline{\boldsymbol{X}}_t, \overline{\Sigma}_t, z_t)$

=====================================

1: $Q_t = \sigma_d^2$

2: $\hat{z}_t = \overline{\boldsymbol{X}}_{t,z}$

3: $H_t = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$

4: $S_t = H_t \overline{\Sigma}_t [H_t]^T + Q_t$

5: $K_t = \overline{\Sigma}_t [H_t]^T [S_t]^{-1}$

6: $\boldsymbol{X}_t = \overline{\boldsymbol{X}}_t + K_t(z_t - \hat{z}_t)$

7: $\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t$

8: *return* $\boldsymbol{X}_t, \Sigma_t$

=====================================

2.2.2. Dealing with Range Data Collectively: Correcting the Prediction using all the Range Data from Every Beacons and Depth Data Collectively

All the measurement data can be used collectively for the correction of the predicted estimation of the location and error covariance at a time. It is assumed that there are n range data $r^i_{t,TOA}$ (i=1,…,n) from n beacons and one data of depth $d_{t,Depth}$. Each range data $r_{t,TOA}$ comes together with one more data of signature $s^i_{t,TOA}$. So the observed measurement data is $\mathbf{z}_t=(r^1_{t,TOA}, s^1_{t,TOA}, \ldots, r^n_{t,TOA}, s^n_{t,TOA}, d_{t,Depth})$. The measurement model is described as the equation (14).

$$z_t = h(\overline{\boldsymbol{X}}_t, \boldsymbol{E}) = \begin{pmatrix} r^1_{t,TOA} & s^1_{t,TOA} & \cdots & r^i_{t,TOA} & s^i_{t,TOA} & d_{t,Depth} \end{pmatrix}^T$$
$$= \begin{pmatrix} \sqrt{(\boldsymbol{E}_{1,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{1,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{1,z} - \overline{\boldsymbol{X}}_{t,z})^2} \\ \boldsymbol{E}_{1,s} \\ \vdots \\ \sqrt{(\boldsymbol{E}_{i,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{i,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{i,z} - \overline{\boldsymbol{X}}_{t,z})^2} \\ \boldsymbol{E}_{i,s} \\ \overline{\boldsymbol{X}}_{t,z} \end{pmatrix} \tag{14}$$

From the measurement equation (14), the Jacobian matrix $\boldsymbol{H}_t$ is derived as the following.

$$
\begin{aligned}
H_t &= \frac{\partial h\left(\overline{\boldsymbol{X}}_t, \boldsymbol{E}\right)}{\partial \boldsymbol{x}_t} \\
&= \begin{pmatrix}
\frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial r_t^1}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \\
\frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial s_t^1}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial r_t^i}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \\
\frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\partial s_t^i}{\partial \overline{\boldsymbol{X}}_{t,\psi}} \\
\frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,x}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,y}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,z}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\phi}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\theta}} & \frac{\overline{\boldsymbol{X}}_{t,z}}{\partial \overline{\boldsymbol{X}}_{t,\psi}}
\end{pmatrix} \\
&= \begin{pmatrix}
-\left(\frac{\boldsymbol{E}_{1,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q_1}}\right) & -\left(\frac{\boldsymbol{E}_{1,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_1}}\right) & -\left(\frac{\boldsymbol{E}_{1,z} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_1}}\right) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\left(\frac{\boldsymbol{E}_{n,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q_i}}\right) & -\left(\frac{\boldsymbol{E}_{n,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_i}}\right) & -\left(\frac{\boldsymbol{E}_{n,z} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_i}}\right) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
\end{aligned}
\tag{15}
$$

$$\sqrt{q_i} = \sqrt{(\boldsymbol{E}_{i,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{i,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{i,z} - \overline{\boldsymbol{X}}_{t,z})^2}$$

Table 5 describes the correction step of EKF which uses all the measurements collectively. It requires all the measurement data $\mathbf{z}_t = (r^1_{t,TOA}, s^1_{t,TOA}, \ldots, r^n_{t,TOA}, s^n_{t,TOA}, d_{t,Depth})$ and all the beacon locations $\boldsymbol{E} = (E_{1,x}, E_{1,y}, E_{1,z}, \ldots, E_{n,x}, E_{n,y}, E_{n,z})$ corresponding to the ranges $r^1_{t,TOA}$, $r^n_{t,TOA}$. Line 5 uses the linearization derived at the equation (15).

Table 5. Correction using all the available measurement data collectively

$$\text{Correction step on TOA and Depth}(\overline{\boldsymbol{X}}_t, \overline{\Sigma}_t, \boldsymbol{z}_t, \boldsymbol{E})$$

1: $$\boldsymbol{Q}_t = \begin{pmatrix} \sigma_{r^1}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{s^1}^2 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \sigma_{r^i}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{s^i}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_d^2 \end{pmatrix}$$

2: $$\boldsymbol{z}_t = \begin{pmatrix} r_t^1 & s_t^1 & \cdots & r_t^i & s_t^i & d_t \end{pmatrix}^T$$

3: $$q = \begin{pmatrix} q_1 \\ \vdots \\ q_i \end{pmatrix} = \begin{pmatrix} (\boldsymbol{E}_{1,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{1,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{1,z} - \overline{\boldsymbol{X}}_{t,z})^2 \\ \vdots \\ (\boldsymbol{E}_{i,x} - \overline{\boldsymbol{X}}_{t,x})^2 + (\boldsymbol{E}_{i,y} - \overline{\boldsymbol{X}}_{t,y})^2 + (\boldsymbol{E}_{i,z} - \overline{\boldsymbol{X}}_{t,z})^2 \end{pmatrix}$$

4: $$\hat{\boldsymbol{z}}_t = \begin{pmatrix} \sqrt{q_1} & \boldsymbol{E}_{1,s} & \cdots & \sqrt{q_i} & \boldsymbol{E}_{i,s} & \overline{\boldsymbol{X}}_{t,z} \end{pmatrix}^T$$

5: $$\boldsymbol{H}_t = \begin{pmatrix} -\left(\frac{\boldsymbol{E}_{1,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q_1}}\right) & -\left(\frac{\boldsymbol{E}_{1,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_1}}\right) & -\left(\frac{\boldsymbol{E}_{1,z} - \overline{\boldsymbol{X}}_{t,z}}{\sqrt{q_1}}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\left(\frac{\boldsymbol{E}_{i,x} - \overline{\boldsymbol{X}}_{t,x}}{\sqrt{q_i}}\right) & -\left(\frac{\boldsymbol{E}_{i,y} - \overline{\boldsymbol{X}}_{t,y}}{\sqrt{q_i}}\right) & -\left(\frac{\boldsymbol{E}_{i,z} - \overline{\boldsymbol{X}}_{t,z}}{\sqrt{q_i}}\right) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

6: $$\boldsymbol{S}_t = H_t \ \overline{\Sigma}_t \ \left[H_t\right]^T + Q_t$$

7: $$\boldsymbol{K}_t = \overline{\Sigma}_t \ \left[H_t\right]^T \ \left[S_t\right]^{-1}$$

8: $$\overline{\boldsymbol{X}}_t = \overline{\boldsymbol{X}}_t + K_t(z_t - \hat{z}_t)$$

9: $$\overline{\Sigma}_t = (I - K_t H_t)\overline{\Sigma}_t$$

10: $$\boldsymbol{X}_t = \overline{\boldsymbol{X}}_t, \ \ \Sigma_t = \overline{\Sigma}_t$$

11: return $\boldsymbol{X}_t, \Sigma_t$

3. SIMULATION RESULTS

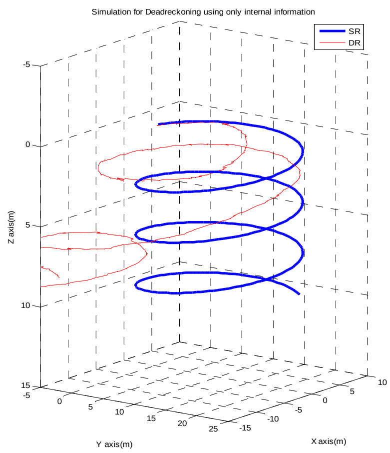
The simulation compares the localization performance for the four methods of dead reckoning (DR), least squares (LS), the EKF method applied individually, and the EKF method applied collectively. They are tested under the same conditions. There are four acoustic beacons located at $E_1(-10,0,0)m$, $E_2(10,0,0)m$, $E_3(10,10,0)m$, and $E_4(-10,0,0.001)m$. The robot is equipped with depth sensor which measures the distance of the robot from the surface. The simulated robot motion and range measurement inevitably include uncertainty, which are described in the Table 6. The uncertainty parameters α_{uu}, α_{ww}, and α_{rr} address the uncertainty of proprioceptive sensors which are used for dead reckoning as the equation (6) describes. The σ_r^i represents standard deviation of the range measurement from the i-th beacon. Likewise, σ_d indicates standard deviation of the depth measurement. The case A has lower uncertainty in both the proprioceptive and exteroceptive sensing than the case B.

Table 6. Uncertainty parameters used for simulations.

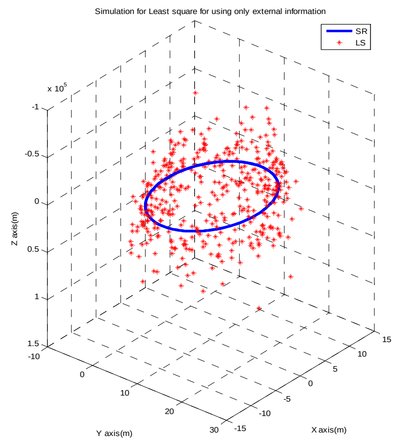
sensors	proprioceptive sensor		exteroceptive sensors		
uncertainty parameters	α_{uu}	α_{ww}	α_{rr}	σ^i_r for range from a beacon	σ_d for depth
case A	1	1	1	1m	1m
case B	2	2	2	2m	2m

3.1. Result for the Case A

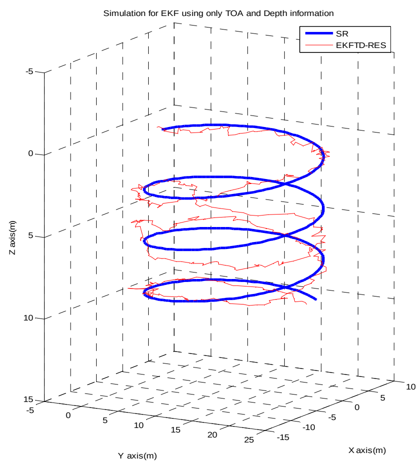
Figure 2 shows the simulation for the case A. In the Figure, (a), (b), (c), and (d) represent the location estimation result of dead reckoning, least squares, individual EKF, and collective EKF, respectively. As well known, the dead-reckoned result deteriorates with time because the error accumulates with time. The LS method which uses only the exteroceptive measurement shows better performance than the dead reckoning and worse performance than the EKF methods. The individual EKF and collective EKF show the best results. To compare these results, deviation distance of the estimated location from the nominal trajectory is calculated and displayed in histogram in the Figure 3, and the distribution of the deviation is calculated and shown in the Table 7. The distribution is described by mean, standard deviation, and maximum deviation. Table 7 reveals that EKF applied collectively is slightly better than the EKF applied individually.



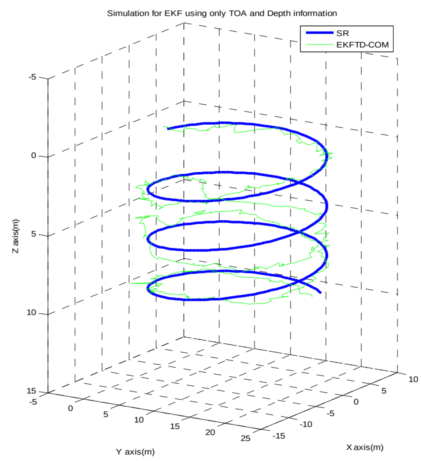
(a) Result of dead reckoning



(b) Result of least squares

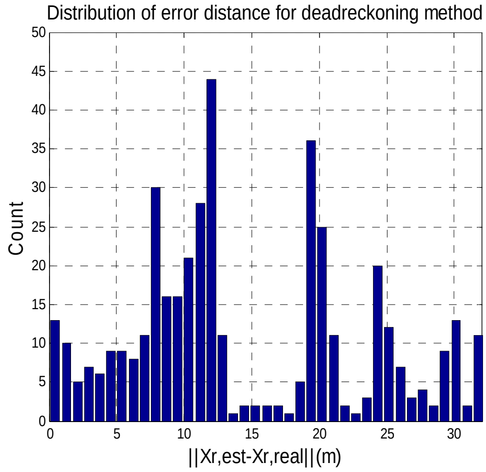


(c) Result of individual EKF

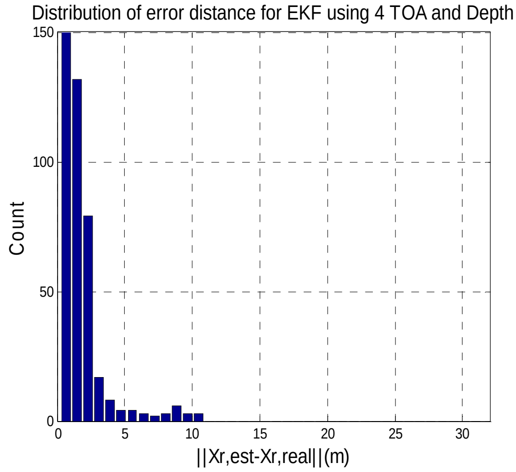


(d) Result of collective EKF

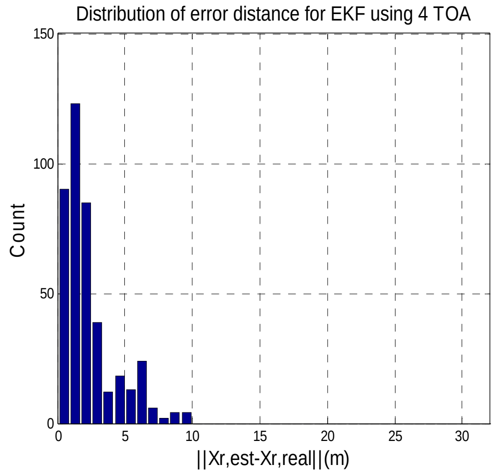
Figure 2. Simulations results for the case A



(a) Distribution for the dead reckoning



(b) Distribution for the individual EKF



(c) Distribution for the collective EKF

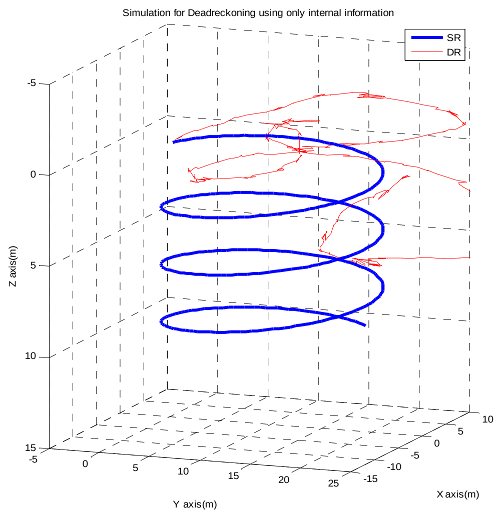
Figure 3. Error distribution of the location estimates

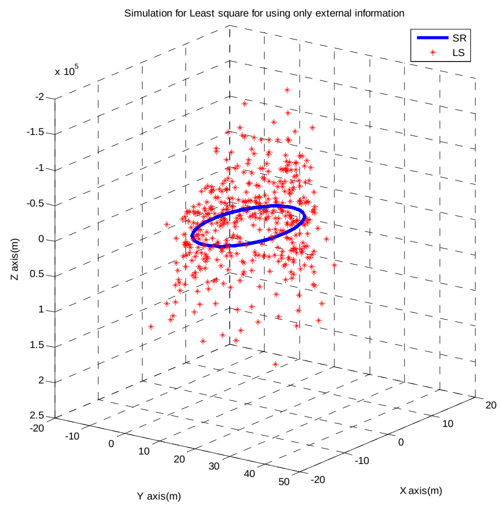
Table 7. distribution of estimation error(unit: *m*)

	mean	standard deviation	maximum error
DR	14.575	8.476	32.137
LS	25954.41	20768.84	120632.8
individual EKF	1.837	1.776	10.139
collective EKF	1.805	1.626	9.828

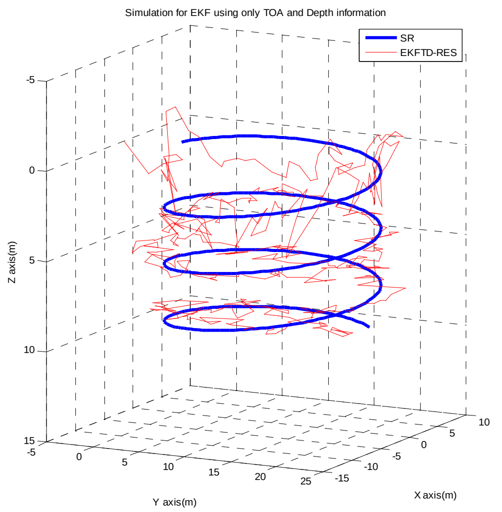
3.2. Result for the Case B

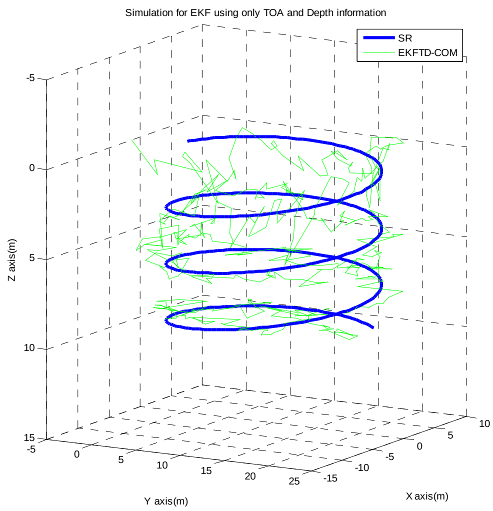
Case B tests the performance of the proposed method where the measurement data has more error than the case A. The uncertainty parameters for the case B are twice of those for the case A as shown in the Table 6. Figure 4 and 5 show the results for the simulation. Comparing the Figure 4 and 5 with the Figures 2 and 3 shows the location error increases with increased uncertainty in both the proprioceptive and exteroceptive sensors. Table 8 lists the statistical analysis of the localization error. Comparison of the error for the individual EKF and collective EKF shows that as the uncertainty increases, the collective EKF shows more noticeable improvement over the individual EKF. In case where uncertainty is smaller, the advantage of the collective EKF was negligible as shown in the last two lines of the Table 7.





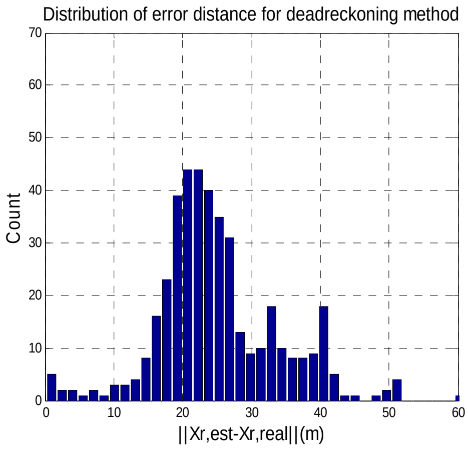
(a) Result of dead reckoning (b) Result of least squares



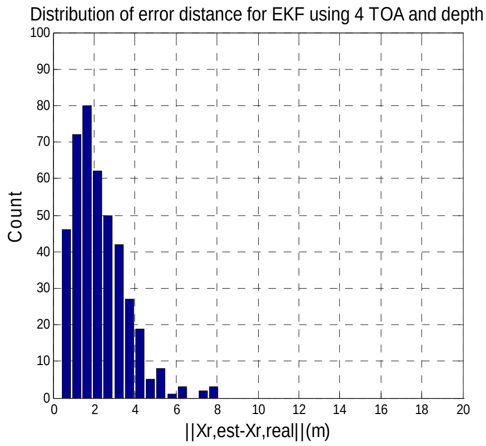


(c) Result of individual EKF (d) Result of collective EKF

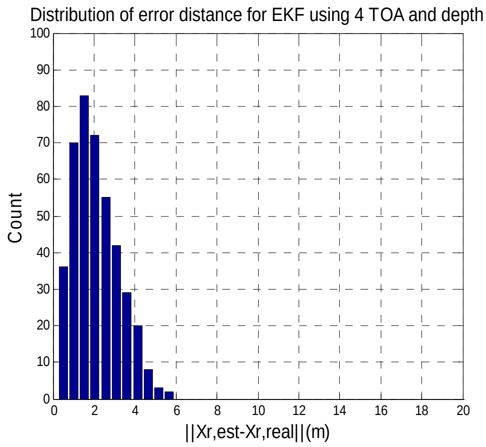
Figure 4. Comparison of location estimation for the case of higher measurement uncertainty



(a) Distribution for the dead reckoning



(b) Distribution for the individual EKF



(c) Distribution for the collective EKF

Figure 5. Error distribution for the case of higher measurement uncertainty

Table 8. Error distribution for the case of higher measurement uncertainty(unit: *m*)

	mean	standard deviation	maximum error
DR	24.922	8.605	51.856
LS	50379.42	40204.03	212778.5
individual EKF	2.282459	1.313	7.813
collective EKF	2.095	1.094	5.546

4. CONCLUSION

This paper describes the application of EKF to localization of underwater robot in 3 dimensional underwater environment. Linearized motion model of the underwater robot is derived together with the linearized measurement model. It uses the exteroceptive measurement data of range from beacons to the robot and the depth of the robot. The EKF implementation is derived in two ways: One is the application of the EKF whenever individual measurements are available, and the other is application of the EKF when a set of data is collected.

The performance is compared through simulation. It is shown that both of the EKF methods work better than the dead-reckoning and least squares method. In case of small uncertainty in the robot motion and measurement, collective implementation of EKF yields a little better performance than the individual implementation of EKF. The performance improves noticeably by the collective EKF collectively when the uncertainty of motion and measurement becomes higher.

The proposed method is useful when neither the precise inertial sensors nor the precise range data are available due to cost, dimension of the robot, or limitation on the underwater environment. The poor quality of proprioceptive measurement alone produces poor dead reckoning. Also, the coarse exteroceptive range data alone produces deficient performance of least squares. Nevertheless, the EKF fuses these data to get useful location data.

ACKNOWLEDGEMENTS

This work was supported by NRF-Grant funded by the Korean Government (NRF-2013-R1A1A4A01 -012469).

REFERENCES

[1] K.S. Song & P.C. Chu, Conceptual Design of Future Undersea Unmanned Vehicle (UUV) System for Mine Disposal, *IEEE Systems Journal*, 6(4), 2012, 1-9.

[2] Y.S. Moon, N.Y. Ko, J. Sur, & Y.O. Lee, *Development of Torpedo Type Mine Disposal Robot RUCUR MK-1*, Proc. 9th International Symposium on Technology and the Mine Problem, Monterey, CA: Naval Postgraduate School, 2010, 17-21.

[3] I. Tusseyeva, S.G. Kim, & Y.G. Kim, 3D Global Dynamic Window Approach for navigation of Autonomous Underwater Vehicles, *International Journal of Fuzzy Logic and Intelligent Systems*, 13(2), 2013, 91-99.

[4] K.Y. Kwon & J. Joh, Autonomous Navigation of an Underwater Robot in the Presence of Multiple Moving Obstacles, *International Journal of Fuzzy Logic and Intelligent Systems*, 5(2), 2005, 124-130.

[5] W.J. Alameda, SEADeViL *A Totally Integrated Inertial Navigation System (INS) Solution*, Proc. on Underwater Intervention Symposium, New Orleans, LA, 2002.

[6] N.Y. Ko, T.G. Kim, & S.W. Noh, *Monte Carlo Localization of Underwater Robot Using Internal and External Information*, Proc. 2011 Asia Pacific IEEE Conf. on Services Computing Conference, Jeju Island, 2011, 410-415.

[7] N.Y. Ko, T.G. Kim, & Y.S. Moon, *Particle Filter Approach for Localization of an Underwater Robot Using Time Difference of Arrival*, Proc. IEEE/MTS Conf. on OCEANS, Yeosu, 2012, 1-7.

[8] G.T. Donovan, Position Error Correction for an Autonomous Underwater Vehicle Inertial Navigation System (INS) Using a Particle Filter, *IEEE Journal of Oceanic Engineering*, 37(3), 2012, 431-445.

[9] N.Y. Ko & T.G. Kim, *Comparison of Kalman Filter and Particle Filter Used for Localization of an Underwater Vehicle*, Proc. 2012 9th International Conf. on Ubiquitous Robots and Ambient Intelligence, Daejeon, 2012, 350-352.

[10] T. Maki, T. Matsuda, T. Sakamaki, T. Ura, & J. Kojima, Navigation Method for Underwater Vehicles Based on Mutual Acoustical Positioning With a Single Seafloor Station, *IEEE Journal of Oceanic Engineering*, 38(1), 2013, 167-177.

[11] M.F. Fallon, G. Papadopoulos, J.J. Leonard, & N.M. Patrikalakis, Cooperative AUV Navigation using a Single Maneuvering Surface Craft, *International Journal of Robotics Research*, 29(12), 2010, 1461-1474.

[12] P. Newman & J. Leonard, *Pure Range-Only Sub-Sea SLAM*, Proc. 2003 IEEE International Conf. on Robotics and Automation, 2003, 1921–1926.

[13] R. Eustice, H. Singh, J. Leonard, M. Walter, & et. al, *Visually navigating the RMS Titanic with SLAM information filters*, Proc. of Robotics: Science and Systems, Cambridge MA, 2005.

[14] T.G. Kim & N.Y. Ko, Localization of an Underwater Robot Using Acoustic Signal, *Journal of Korea Robotics Society*, 7(4), 2012, 1461-1474.

[15] T.G. Kim & N.Y. Ko, Comparison and Analysis of Methods for Localization of a Mobile Robot, *Journal of Korean Institute of Information Technology*, 11(1), 2013, 79-89.

[16] F. Daum, Nonlinear Filters: Beyond the Kalman Filter, *Magazine of IEEE Aerospace and Electronic Systems*, 20(8), 2005, 57-69.

[17] Y.S. Moon, C.H. Yoon, Y.P. Lee, N Y. Ko, & et. al, *A hard-ware Design of Pinger/Hydrophone Module for Underwater aNNs and Sea Trial Experiment*, Proc. on Korea Unmanned Underwater Vehicle, 2012, 16-19.