

Modeling the Kinematics of an Autonomous Underwater Vehicle for Range-Bearing Simultaneous Localization and Mapping

O. Matsebe¹ S. Holtzhausen¹ C.M. Kumile² N.S. Tlale¹

¹Department of Mechatronics and Micro Manufacturing, Centre for Scientific and Industrial Research, Pretoria, South Africa

(Tel: 012 841 3700, E-mail: OMatsebe@csir.co.za)

(Tel: 012 841 3090, E-mail: SHoltzhausen@csir.co.za)

(Tel: 012 841 3107, E-mail: Ntlale@csir.co.za)

²Department of Mechanical Engineering, Tshwane University of Technology, Pretoria, South Africa,

(E-mail:kumilecm@tut.ac.za)

Abstract-The “solution” of the Simultaneous Localisation and Mapping (SLAM) problem has been one of the notable successes of the robotics community. SLAM has been formulated and solved as a theoretical problem in a number of different forms. SLAM has also been implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems. At a theoretical and conceptual level, SLAM can now be considered a solved problem. However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm. This paper describes the Autonomous Underwater Vehicle (AUV) kinematic and sensor models, it overviews the basic theoretical solution to the Extended Kalman Filter (EKF) SLAM problem, it also describes the way-point guidance based on Line of Sight (LOS). In this paper, it has been shown through Matlab simulation that the vehicle is able to localize its position using features that it observes in the environment and at the same time map those features. The vehicle is expected to follow a pre-defined sinusoidal path.

Keywords: SLAM, EKF, AUV, Kinematics

I. INTRODUCTION

SLAM is the generalised navigation problem. It asks if it is possible for a robot, starting with no prior information, to move through its environment and build a consistent map of the entire environment. Additionally the vehicle must be able to use the map to navigate and hence plan and control its trajectory during the mapping process. The applications of a robot capable of navigating, with no prior map, are diverse indeed. Domains in which 'man in the loop' systems are impractical or difficult such as sub-sea surveys and disaster zones are obvious candidates. Beyond these, the sheer increase in autonomy that would result from reliable, robust navigation in large dynamic environments is simply enormous. Autonomous navigation has been an active area of research for many years [1].

In many applications the environment is unknown. A priori maps are usually costly to obtain, inaccurate, incomplete, out of date and they limit the robot to that particular environment [2].

The “solution” of the SLAM problem has been one of the notable successes of the robotics community. SLAM has been formulated and solved as a theoretical problem in a number of different forms. SLAM has also been implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems. At a theoretical and conceptual level, SLAM can now be considered a solved problem. However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm [3].

Many land-based robots use Global Positioning System (GPS) or maps of the environment to provide accurate position updates, a robot navigating underwater does not have access to this type of information. In typical underwater scientific missions, a-priori maps are seldom available.

This paper describes the AUV kinematic model, range-bearing sensor models and their uncertainties. It overviews the basic theoretical solution to the EKF-SLAM problem. It also describes the way-point guidance mechanism based on Line of Sight (LOS). Simulation results using Matlab are presented.

The paper is structured as follows: Section II describes the vehicle kinematic model and the sensor models. Section III details the theoretical aspects of EKF-SLAM solution. Section IV describes the way-point guidance mechanism based on Line of Sight (LOS). Section V presents a discussion of the simulation results and the conclusions are found in Section VI.

II. VEHICLE AND SENSOR MODELS

A. Kinematic Model of the vehicle

Modeling of the kinematic states involves the study of the geometrical aspects of motion. The motion of the vehicle through the environment can be modeled through the following discrete time non-linear kinematic model:

$$X_v(k) = f(X_v(k-1), u(k), k). \quad (1)$$

Thus $X_v(k)$ is given by:

$$x_v(k) = x_v(k-1) + V \nabla T \cos(\psi_d(k-1)). \quad (2)$$

$$y_v(k) = y_v(k-1) + V \nabla T \sin(\psi_d(k-1)). \quad (3)$$

$$\psi_v(k) = \psi_v(k-1) + \nabla T \psi_s(k-1). \quad (4)$$

(x_v, y_v) is the position of the vehicle and $\psi_v(k)$ is its orientation in the global frame of reference at time step k . ∇T is the sample time. ψ_s is the change in angle required to reach next pose. The vehicle is controlled through demanded constant forward velocity V and desired heading, $\psi_d(k)$. These are used with the kinematic model to predict the position of the vehicle. The control input vector is given by:

$$u(k) = \begin{bmatrix} V \\ \psi_d(k) \end{bmatrix}. \quad (5)$$

Clearly the above kinematic model is a little unrealistic, we need to model uncertainty. The complete non-linear model can now be expressed in general form as:

$$X_v(k) = f(X_v(k-1), u_n(k) + \gamma_u(k)) + \gamma_f(k). \quad (6)$$

One popular way to model uncertainty is to insert noise terms into the control signal $u(k)$ such that:

$$u(k) = u_n(k) + \gamma_u(k). \quad (7)$$

$u_n(k)$ is a nominal control signal and $\gamma_u(k)$ is a zero mean Gaussian distributed noise vector such that:

$$\gamma_u(k) = \begin{bmatrix} \sigma_v \\ \sigma_{\psi_d} \end{bmatrix} \quad (8)$$

σ_v and σ_{ψ_d} are noises in the velocity V and desired heading $\psi_d(k)$ respectively. $\gamma_f(k)$ is the process noise.

$$\gamma_f(k) = \begin{bmatrix} \sigma_{x_v} \\ \sigma_{y_v} \\ \sigma_{\psi_v} \end{bmatrix} \quad (9)$$

The strength of control noise (covariance) is thus given by:

$$Q_u = \text{diag}[\sigma_v^2 \sigma_{\psi_d}^2]. \quad (10)$$

The strength of process noise (covariance) is thus given by:

$$Q_f = \text{diag}[\sigma_x^2 \sigma_y^2 \sigma_{\psi}^2]. \quad (11)$$

The noises are assumed to be uncorrelated, zero mean and white.

B. Non-Linear Observation model

Assume that the vehicle is equipped with external sensors capable of measuring the range and bearing to features in the environment. The measurement model is thus given by:

$$z(k) = h(X_v, x_i, y_i) = \begin{bmatrix} r_i \\ \theta_i \end{bmatrix} + \gamma_h(k). \quad (12)$$

$$r_i = \sqrt{(x_i - x_v(k))^2 + (y_i - y_v(k))^2}. \quad (13)$$

$$\theta_i = \tan^{-1} \left[\frac{y_i - y_v(k)}{x_i - x_v(k)} \right] - \psi_v. \quad (14)$$

r_i and θ_i are range and bearing to the observed feature respectively. The observation noise vector is given as:

$$\gamma_h(k) = \begin{bmatrix} \zeta_{r_i} \\ \zeta_{\theta_i} \end{bmatrix}. \quad (15)$$

The strength of the observation noise (covariance) is thus given by:

$$R = \text{diag}[\sigma_r^2 \sigma_{\theta}^2]. \quad (16)$$

The noises are assumed to be uncorrelated, zero mean and white.

III. SLAM

Most of the actions performed by the vehicle rely on its ability to estimate pose. To avoid obstacles and follow waypoints the vehicle need to have reliable estimate of its position and orientation [4]. This section presents the feature based EKF-

SLAM technique used for generating vehicle pose estimates and positions of features in the vehicle's operating environment.

The localisation and map building process consists of a recursive, three-stage procedure comprising prediction, observation and update steps using an EKF. The EKF estimates the two dimensional pose of the vehicle made up of the position (x_v, y_v) and orientation ψ_v , together with the estimates of the positions of the N features $x_{f,i}$ where $i = 1 \dots N$ using observations from the sensors on board the vehicle [4]. Here we will constrain ourselves to using the simplest feature possible – a point feature such that the coordinates of the i^{th} feature in the global reference frame are given by:

$$x_{f,i} = \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \quad (17)$$

SLAM considers that all landmarks are stationary. The state transition model for the i^{th} landmark is given by:

$$x_{f,i}(k) = x_{f,i}(k-1) = x_{f,i}. \quad (18)$$

It can be seen that the model for the evolution of the landmarks does not have any uncertainty.

A. Non-Linear Prediction

The prediction stage is achieved by passing the last estimate through the non-linear model of the motion of the vehicle to compute the vehicle position at instant k based on a control signal $u(k)$ and using the information up to instant $k-1$ [4].

The predicted vehicle state X_v is thus given by:

$$X_v(k \setminus k-1) = f(X_v(k-1 \setminus k-1), u(k)). \quad (19)$$

Under SLAM the system detects new features at the beginning of the mission and when exploring new areas. Once these features become reliable and stable they are incorporated into the map becoming part of the state vector. The positions of features are predicted as follows:

$$\begin{bmatrix} x_{f,1}(k \setminus k-1) \\ \cdot \\ \cdot \\ x_{f,N}(k \setminus k-1) \end{bmatrix} = \begin{bmatrix} x_{f,1}(k) \\ \cdot \\ \cdot \\ x_{f,N}(k) \end{bmatrix} \quad (20)$$

The augmented state vector containing both the state of the vehicle and the state of all feature locations is denoted:

$$x(k \setminus k-1) = [X_v^T(k) \dots x_{f,1}^T \dots x_{f,N}^T]^T. \quad (21)$$

Now we need to propagate the covariance. The covariance of the vehicle and feature states, $P(k \setminus k-1)$ is computed using the gradient of the state propagation equation, $F(k)$, linearized about the current best estimate, the process noise covariance, Q_f , and the control noise covariance, Q_u .

$$P(k \setminus k-1) = F(k)P(k-1 \setminus k-1)F^T(k) + J_u(k)Q_u(k)J_u^T(k) + Q_f(k) \quad (22)$$

J_u is the gradient of the state propagation equation with respect to the control input $u(k)$.

If an observation of a new feature is made then the error covariance is augmented with the new feature covariance.

B. Observation prediction

The measurement that we would expect (predicted observation) if $z(k)$ corresponded to the i^{th} feature and the prediction $x(k \setminus k-1)$ was correct is given as:

$$z(k \setminus k-1) = h(x(k \setminus k-1)). \quad (23)$$

C. Prepare for Update

The solution presented in the next section glosses over a very important aspect of SLAM: it assumes that each measurement is automatically associated with the correct landmark. In practice, landmarks have similar properties which make them good features but often make them difficult to distinguish one from the other. When this happens we must address the problem of data association, which is the question of which landmark corresponds to a particular measurement [6].

For each landmark observed we compute the innovation $v(k)$, which is defined to be the difference between the actual measurement $z(k)$ and the measurement that we would expect if $z(k)$ corresponded to the i^{th} feature and the prediction $x(k \setminus k-1)$ was correct. This means that:

$$v(k) = z(k) - z(k \setminus k-1). \quad (24)$$

The smaller the innovation $v(k)$, the more likely that the measurement corresponds to the i^{th} feature. The innovation is assumed to be white and uncorrelated with covariance S . The uncertainties in the predictions and observations are encoded in the innovation covariance matrix S which is computed using the current state covariance estimate $P(k \setminus k-1)$, the gradient of

the observation model, $H(k/k-1)$ and the covariance of the observation model $R(k/k-1)$ [6].

$$S(k/k-1) = H(k/k-1)P(k/k-1)H^T(k/k-1) + R(k/k-1) \quad (25)$$

$$H(k/k-1) = [H_v \dots 0 \dots H_f \dots 0] \quad (26)$$

H_v is the gradient of the observation model with respect to the vehicle states. H_f is the gradient of the observation model with respect to the observed feature

D. Update

The update step need not happen at every iteration of the filter. If at a given time step no observations are available then the best estimate at time k is simply the prediction $x(k/k-1)$. If an observation is made of an existing feature in the map, then the state estimate can now be updated using the optimal gain matrix $W(k)$. This gain matrix provides a weighted sum of the prediction and observation. It is computed using the innovation covariance $S(k/k-1)$, the predicted state covariance $P(k/k-1)$ and the gradient of the observation model, $H(k/k-1)$.

$$W(k) = P(k/k-1)H(k/k-1)S^{-1}(k/k-1) \quad (27)$$

This is then used to compute the state update $x(k/k)$ as well as the updated state covariance $P(k/k)$.

$$x(k/k) = x(k/k-1) + W(k/k-1)v(k/k-1) \quad (29)$$

$$P(k/k) = P(k/k-1) - W(k/k-1)S(k/k-1)W(k/k-1)^T \quad (30)$$

IV. WAY-POINT GUIDANCE

The method used here to guide the vehicle through waypoints is Line of Sight (LOS). LOS is defined in terms of a desired heading angle. Let the vehicle mission be given by a set of waypoints $[x_d(k), y_d(k)]$ for $k = 1 \dots N$.

$$\psi_d(t) = \tan^{-1} \left(\frac{y_d(k) - y(t)}{x_d(k) - x(t)} \right) \quad (31)$$

Care must be taken to select the proper quadrant for the desired heading, ψ_d . After the quadrant check is performed, the next way point is selected based on whether the vehicle lies within a circle of acceptance with radius ρ_0 around a way-point

$(x_d(k), y_d(k))$. Moreover if the vehicle location $(x_v(t), y_v(t))$ at the time t satisfies:

$$[x_d(k) - x(t)]^2 + [y_d(k) - y(t)]^2 \leq \rho_0^2. \quad (32)$$

then the next way point $[x_d(k+1), y_d(k+1)]$ is selected [8].

V. RESULTS ANALYSIS

A. Filter Parameters

The filter parameters used to obtain the simulation results are shown in Table 1 below:

Table 1
FILTER PARAMETERS

Sampling Period	∇T	1 sec
Vehicle x_v process noise std dev	σ_{x_v}	0.075m
Vehicle y_v process noise std dev	σ_{y_v}	0.075m
Vehicle heading process noise std dev	σ_{ψ_v}	0.5 ⁰
Vehicle velocity V std dev	σ_V	0.075m/s
Vehicle desired heading std dev	σ_{ψ_d}	0.5 ⁰
Range measurement std dev	σ_r	0.5m
Bearing measurement std dev	σ_θ	2.0 ⁰

B. Filter Initialization

The recursive formulation of the EKF algorithm means that we must provide some reasonable guess for the initial conditions of the vehicle state prediction, $X_v(k/k-1)$ and state error covariance, $P(k/k-1)$. The choice of a reasonably good initial estimate improves convergence and is essential in the convergence of the EKF. The location of the vehicle is initialised

as $X_v(0/0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. The error covariance is initialised as:

$$P(0/0) = \text{sqr}t(\text{diag}[\sigma_{x_v} \sigma_{y_v} \sigma_{\psi_v}]). \quad (32)$$

C. Analysis of filter performance

The vehicle is required to follow a pre-defined sinusoidal path defined by a set of way-points. We assume that the vehicle is able to avoid obstacles. We also assume that we know which measurement comes from which landmark and that the vehicle maintains a constant depth. The simulation is such that as the vehicle moves through the environment it randomly observes a

feature checks if that feature has been observed, if the feature is new it is added into the map, if the feature has been observed the state vector is updated.

Fig 1 below shows a pre-defined path (Desired Path) which the vehicle is required to track, the path where the vehicle actually went (True Path), the path estimated by SLAM (Estimated Path). It also shows the features that are in the simulated environment (True Map). It can be noticed that the vehicle is able to follow the desired path closely during the run. The observed features and estimated feature locations are not shown in the figure.

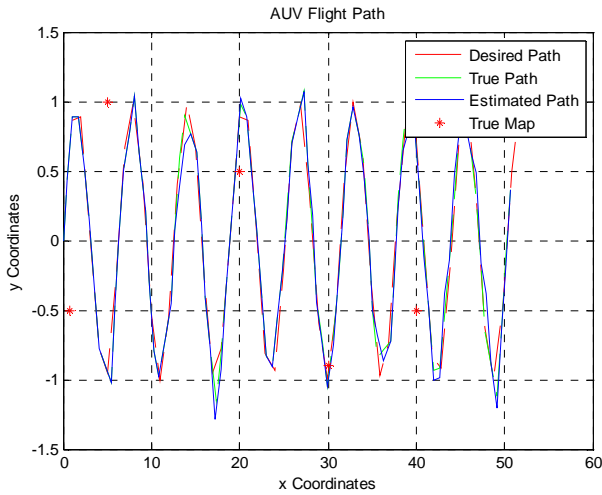


FIGURE 1. Showing the Desired Path (red), True Path (green), Estimated Path (blue) and the True map (red stars).

Fig 2 below shows the Estimated and True x positions of the vehicle. It can be noticed that the graphs are actually close to each other with small errors.

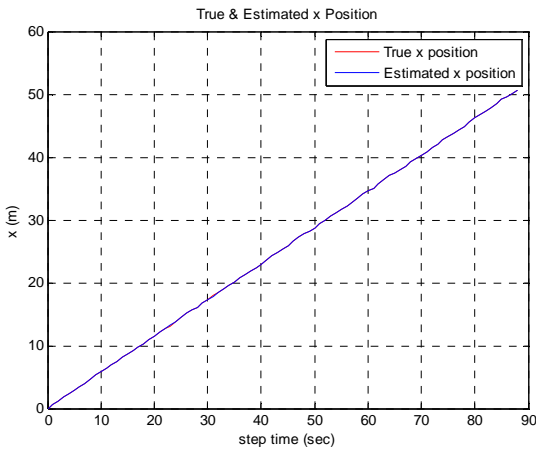


FIGURE 2. Showing Estimated and True X position

Fig 3 below shows the Estimated and True y positions of the vehicle. It can be noticed that the graphs are actually close to each other with small errors.

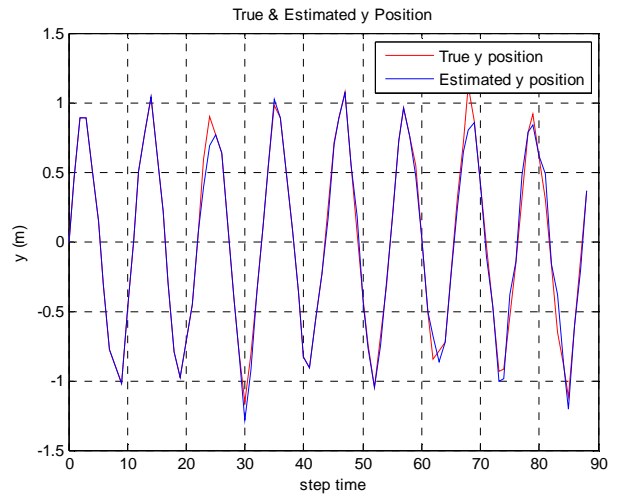


FIGURE 3. Showing Estimated and True y position

Fig 4 below shows the Estimated and True ψ_v heading of the vehicle. From the figure it is clear that the graphs are close to each other with small errors.

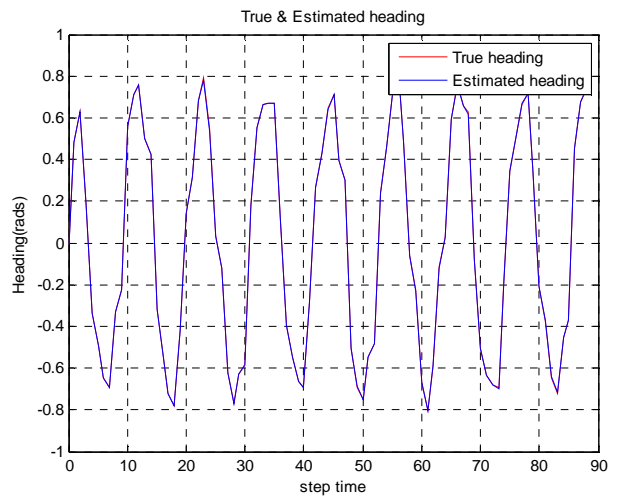


FIGURE 4. Showing Estimated and True ψ_v heading

Fig 5 below shows the actual errors (errors between true vehicle states and the estimated states). The left hand side graph shows the errors associated with x, y positions and the right hand side graph shows the errors in vehicle heading. These errors are relatively small as expected.

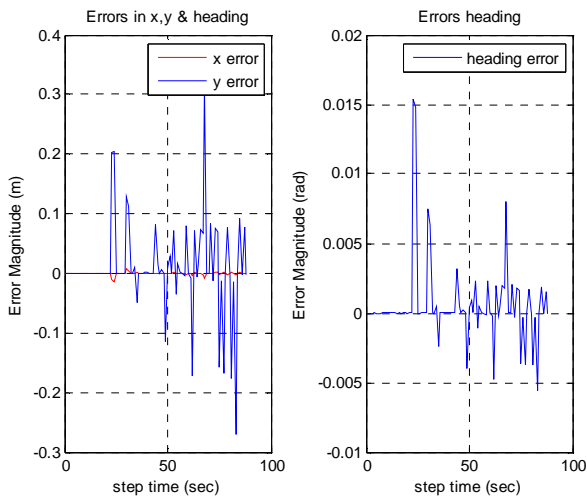


FIGURE 5. Showing estimated and true ψ_v heading

The most important method of analyzing the filter performance is using the innovation. The innovation is tested against the hypothesis that it is white and uncorrelated. Testing the innovation for these properties tells us a great deal about the performance of the filter and can be directly used to tune the filter performance. This will be analyzed in future.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, it has been shown that the vehicle is able to localize its position using the features that it observes in the environment and at the same time map those features.

This is performed while the vehicle follows a pre-defined sinusoidal path. It has also been shown that the actual errors are relatively small as expected although the filter needs some further tuning.

The focus of future work will be on analysis of the innovation sequence, steady state performance, estimated error and error conditions so as to tune the filter further. An Obstacle avoidance mechanism will be developed and the data association problem will also be addressed. The algorithm will also be expanded to handle multiple observations.

REFERENCES

- [1] P.M. Newman, "EKF based navigation and SLAM", 2006, unpublished.
- [2] J. Neira, "The state of the art in the EKF solution to SLAM", 2008, unpublished.
- [3] H.D. Whyte, Simultaneous Localisation and Mapping: Part 1, IEEE Robotics and Automation Magazine, pp. 100-108, 2006.
- [4] S.B. Williams, P. Newman, J. Rosenblatt, G. Dissanayaki and H.D. Whyte, "Autonomous underwater navigation and control". Robotica, vol.19, No.5, pp.481-496, 2001.
- [5] H.D. Whyte, "Introduction to estimation and the kalman filter", 2001, unpublished.
- [6] H. Choset et al. *Principles of robot motion, theory, algorithms, and implementation*, The MIT Press, 2005.
- [7] R. Siegwart and I.R. Nourbakhsh, *Introduction to autonomous mobile robots*, The MIT Press, 2004.
- [8] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, 1994.