SICE Annual Conference 2007
Sept. 17-20, 2007, Kagawa University, Japan

A Solution to the SLAM Problem Based on Fuzzy Kalman Filter using Pseudolinear Measurement Model

Chandima Dedduwa Pathiranage ,Keigo Watanabe and Kiyotaka Izumi

Department of Advanced Systems Control Engineering, Saga University, Saga, Japan
(Tel : +81-952-28-8602; E-mail: dpchandima@yahoo.com, {watanabe, izumi}@me.saga-u.ac.jp)

Abstract: This paper proposes a fuzzy logic based solution to the SLAM problem. Less error prone vehicle process model is proposed to improve the accuracy and the faster convergence of state estimation. Evolution of vehicle motion is modeled using dead-reckoned odometry measurements as control inputs. Nonlinear process model and observation model are formulated as pseudolinear models and approximated by local linear models according to the T-S fuzzy model. Linear Kalman filter equations are then used to estimate the state of the approximated local linear models. Combination of these local state estimates results in global state estimate. The above system is implemented and simulated with Matlab to claim that the proposed method yet finds a better solution to the SLAM problem. The proposed method shows a way to use nonlinear systems in Kalman filter estimator without using Jacobian matrices. It is found that a fuzzy logic based approach with the pseudolinear models provides a demanding solution to state estimation.

Keywords: T-S fuzzy model, Kalman filter, Pseudolinear model, State estimation, Stability

1. INTRODUCTION

The simultaneous localization and mapping (SLAM) [1][2] problem, also known as concurrent mapping and localization (CML) problem, is often recognized in the robotics literature as one of the key challenges in building autonomous capabilities for mobile vehicles. The goal of an autonomous vehicle performing SLAM is to start from and unknown location in an unknown environment and build a map (consisting of environmental features) of its environment incrementally by using the uncertain information extracted from it sensors, whilst simultaneously using that map to localize itself with respect to a reference coordinate frame and navigate in real time.

A vehicle capable of preforming SLAM using naturally occurring environmental features and capable of running for hours or possibly day in completely unknown and unstructured environments will indeed be invaluable in several key areas of robotics. These include autonomous vehicle operation in unstructured terrain, driver-assistance systems, mining, surveying, cargo handling, autonomous underwater explorations, aviation applications, autonomous planetary exploration, and military applications. The first solution to the SLAM problem was proposed by Smith *et al.* [3]. The emphasized the importance of map and vehicle correlations in SLAM and introduced the extended Kalman filter (EKF)-based stochastic mapping framework, which estimated the vehicle pose and the map feature(landmark) positions in an augmented state vector using second order statistics. Although EKF-based SLAM within the stochastic mapping framework gained wide popularity among the SLAM research community, over time, it was shown to have several shortcomings [4][5]. Notable shortcomings are its susceptibility to data-association errors and inconsistent treatment of nonlinearities.

Data association, registration, or the correspondence problem in one of the extremely difficult problems encounted in SLAM even in static environments and much more challenging in dynamic environments consisting of objects moving at varying velocities. Almost every state estimation algorithm has to deal with the correspondence problem in the form of maximum-likelihood assignment or correlation search in establishing the correspondence between the elements of observations and the available features. Uncertainties in vehicle pose, variable feature densities, dynamic objects in the environment, and spurious measurements complicate data association in the SLAM problem in many respects. An efficient data-association scheme must aid feature or track initialization, maintenance, termination and map management.

Here we propose some remedies to overcome the shortcomings of EKF algorithm. To preserve the nonlinearity in the system, motion and observation models are represented by the pseudolinear models [6][7][8]. This avoids the direct linearization of the system. Discrete time motion model is derived from the dead-reckoned measurements of the vehicle pose as to reduce the error associated with the control inputs. This assures the less error prone motion model producing faster convergence. We propose a fuzzy Kalman filter based state estimation algorithm to the SLAM problem. Fuzzy logic has been a promising control tool for the nonlinear systems. Fuzzy state estimation is a topic that has received very little attention. Fuzzy Kalman filtering [9] is a recently proposed method to extend Kalman filter to the case where the linear system parameters are fuzzy variables within intervals. We draw the superiority of fuzzy Kalman filtering for the state estimation through the SLAM algorithm developed with T-S fuzzy model in this paper. The proposed T-S fuzzy model [10] based algorithm to the SLAM problem has proven that a demanding (not conventional) solution to the SLAM problem exists and it overcomes limitations of the EKF based SLAM hinting a new path explored is much suitable in finding an advanced solution to localization and mapping problems.

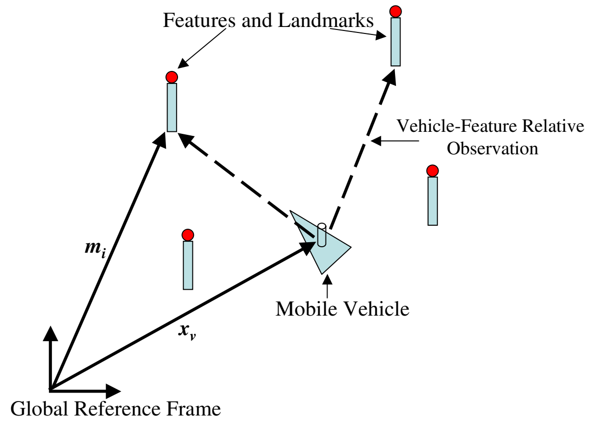


Fig. 1 A vehicle taking relative measurements to environmental landmarks

1.1 Overview of Feature-Based SLAM

The SLAM algorithm addresses the problem of a vehicle with a known kinematic model, starting at an unknown position, moving through an unknown environment populated with artificial or natural features. The objective of SLAM is to localize the vehicle and at the same time build an incremental navigation map with observed features. The vehicle is equipped with a sensor capable of taking measurement of the relative location of the feature and the vehicle itself. This scenario is shown in Fig. 1.

For simplicity, an EKF-based stochastic mapping approach is employed in this formulation. The methodology is still considered to be the primary framework of the most feature-based stochastic SLAM algorithms [3][5]. The major highlight of the formulation is its consistent probabilistic representation of the pose of the vehicle (or robot), the positions of features, their uncertainties, and their interrelationships. In the EKF-SLAM formulation, the pose of the vehicle and the positions of features are concatenated to form an augmented or composite state vector $\boldsymbol{x}(k)$. The pose of vehicle (robot) $\boldsymbol{x}_v(k)$ and feature locations $\boldsymbol{m}(k)$ (known as the map) at a time instant k are represented by absolute coordinates with reference to a global coordinate frame. In the following, the EKF-based SLAM solution is stated for a vehicle traversing a 2-D terrain with point features or landmarks in the environments. The vehicle-map composite or SLAM state vector $\boldsymbol{x}(k)$ at time instant k is thus

$$\boldsymbol{x}(k) = \left[\boldsymbol{x}_v^T(k) \quad \boldsymbol{m}^T(k)\right]^T \tag{1}$$

where $\boldsymbol{x}_v(k) = [x(k)\ y(k)\ \phi(k)]^T$ is the vehicle pose with $x(k), y(k)$, and $\phi(k)$ denoting position coordinates and heading of the vehicle and $\boldsymbol{m}(k) = [x_1(k)\ y_1(k)$ .. .... $x_n(k)\ y_n(k)]^T$ is the vector of feature positions with $[x_i(k)\ \ y_i(k)]^T,\ i = 1, 2, ..., n$ denoting the feature coordinates with respect to the global coordinate frame. In general, motion model of the vehicle is nonlinear and can be represented in closed form as

$$\boldsymbol{x}_v(k+1) = \boldsymbol{f}\left(\boldsymbol{x}_v(k),\ \boldsymbol{u}(k)\right) + \boldsymbol{w}(k) \tag{2}$$

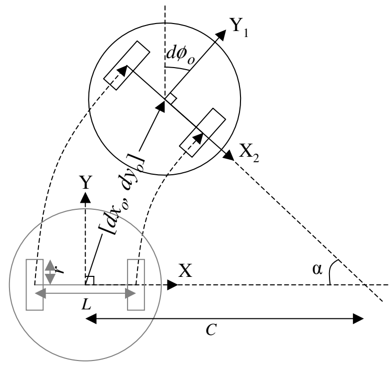


Fig. 2 Geometric construction of rear wheel movements

where $\boldsymbol{u}(k)$ is the control input at time k and $\boldsymbol{w}(k)$ is a zero-mean temporally uncorrelated noise sequence with covariance matrix $\boldsymbol{Q}_v(k)$. Assuming static features (landmarks), the process model of the feature map is

$$\boldsymbol{m}(k+1) = \boldsymbol{m}(k) \tag{3}$$

The observation model is represented by

$$\boldsymbol{z}(k) = \boldsymbol{h}\left(\boldsymbol{x}_v(k),\ \boldsymbol{m}(k)\right) + \boldsymbol{v}(k) \tag{4}$$

where $\boldsymbol{v}(k)$ is a zero-mean temporally uncorrelated noise sequence with covariance matrix $\boldsymbol{R}(k)$. When the covariance matrix of the composite state vector $\boldsymbol{x}(k)$, which is known as the covariance matrix, is denoted by $\boldsymbol{P}(k)$, and the observation prediction is specified by $\hat{\boldsymbol{z}}(k)$, then the EKF predictor equations are as follows:

$$\hat{\boldsymbol{x}}(k+1|k) = \left[\left(\boldsymbol{f}\left(\hat{\boldsymbol{x}}_v(k),\ \boldsymbol{u}(k)\right)\right)^T \quad \hat{\boldsymbol{m}}^T(k)\right]^T \tag{5}$$

$$\boldsymbol{P}(k+1|k) = \boldsymbol{F}(k)\boldsymbol{P}(k|k)\boldsymbol{F}^T(k) + \boldsymbol{Q}(k) \tag{6}$$

$$\hat{\boldsymbol{z}}(k+1|k) = \boldsymbol{h}\left(\hat{\boldsymbol{x}}_v(k+1|k),\ \hat{\boldsymbol{m}}(k+1|k)\right) \tag{7}$$

where $\boldsymbol{F}$ is the Jacobian $(\partial\boldsymbol{x}(k+1|k)/\partial\boldsymbol{x})$ of the composite process model evaluated at time k and $\boldsymbol{Q}(k)$ is the composite process noise covariance matrix. When observations $\boldsymbol{z}(k)$ are made at time k, and after correct observation to feature associations are resolved using an appropriate date association algorithm, then the EKF update equations are applied as follows:

$$\boldsymbol{\nu}(k+1) = \boldsymbol{z}(k+1) - \hat{\boldsymbol{z}}(k+1) \tag{8}$$

$$\hat{\boldsymbol{x}}(k+1|k+1) = \hat{\boldsymbol{x}}(k+1|k) + \boldsymbol{K}(k+1)\boldsymbol{\nu}(k+1) \tag{9}$$

$$\boldsymbol{P}(k+1|k+1) = \boldsymbol{P}(k+1|k) - \boldsymbol{K}(k+1)\boldsymbol{S}(k+1) \times\boldsymbol{K}(k+1)^T \tag{10}$$

where $\boldsymbol{\nu}(k+1)$ is the observation innovation, $\boldsymbol{S}(k+1)$ is its covariance matrix, and $\boldsymbol{K}(k+1) = \boldsymbol{P}(k+1|k)\boldsymbol{H}^T(k+1)\boldsymbol{S}^{-1}(k+1)$ is the Kalman gain, where $\boldsymbol{H}(k+1)$ is the Jacobian $(\partial\boldsymbol{h}(k+1)/\partial\boldsymbol{x})$.

2. VEHICLE MODEL AND ODOMETRY

The way of formulating the motion model for a mobile robot vehicle has a significant influence in state estimation in SLAM problem. Direct state estimation is not possible through the pure motion model of vehicle because it is obvious that control inputs to vehicle are distorted by noise. Modeling the motion model of a mobile robot vehicle with less error prone control inputs has invaluable advantages in state estimation. Lower errors in control inputs will always help state estimator to find desired state estimates as faster as possible. Thus it is of paramount importance that we find low noisy control inputs in modeling the motion model of the robot vehicle.

In the history of SLAM problem, it has been the common practice of generating the motion model with forward velocity and steering angle as control inputs. In this representation, measurement errors in control inputs propagate into the next stage with the same noise strength. But the model that we propose has less error prone control inputs as control inputs to the motion model are derived from the successive dead-reckoned poses where current dead-reckoned pose subtracts the immediate previous dead-reckoned pose to produce the control input and it is hopeful that this subtracts the common dead-reckoned error giving a control input with low noise level.

2.1 Dead-Reckoned Odometry Measurements

The low level controller on the vehicle reads the encoders on the vehicle's wheels and outputs an estimate of its location. Assume left and right wheels of radius r mounted on either side of the rear axel in one time interval turn an amount $\partial\theta_l, \partial\theta_r$ as shown in Fig. 2. We want to express the change of position of the center of rear axel of the vehicle (dx_o, dy_o) and the change of orientation $(d\theta_o)$ as a function of $\partial\theta_l, \partial\theta_r$:

$$r\partial\theta_r = (c - L/2)\alpha \tag{11}$$

$$r\partial\theta_l = (c + L/2)\alpha \tag{12}$$

$$c = \frac{L}{2}\frac{\partial\theta_l + \partial\theta_r}{\partial\theta_l - \partial\theta_r} \tag{13}$$

$$\alpha = \frac{r}{L}(\partial\theta_l - \partial\theta_r) \tag{14}$$

immediately then we have

$$\begin{bmatrix} dx_o \\ dy_o \\ d\theta_o \end{bmatrix} = \begin{bmatrix} (1 - \cos\alpha)c \\ c\sin\alpha \\ -\alpha \end{bmatrix} \tag{15}$$

The dead-reckoning system in the vehicle simply compounds these small changes in position and orientation to obtain a global position estimate. Starting from an initial nominal frame at each iteration of its sensing loop it deduces a small change in position and orientation, and then "adds" this to its last dead-reckoned position. Of course the "addition" is slightly more complex than simple adding. What actually happens is that the vehicle composes successive coordinate transformation. This is an important concept and will be discussed in the next section.

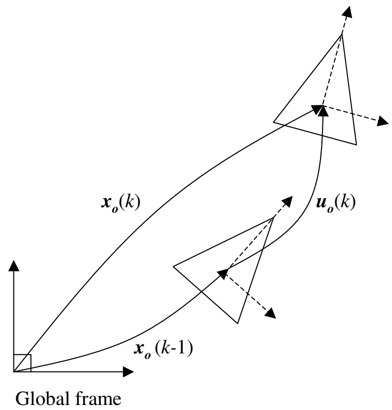


Fig. 3 Deducing a new dead-reckoned state from a prior dead-reckoned state with a local odometry measurement

2.2 Compounding of Odometry Measurements

We define two operators $\oplus$ (compound) and $\ominus$ (inverse) to allow us to compose multiple transformations. They allow us to express something (perhaps a point or vehicle) described in one frame, in another alternative frame. We can use this notation to explain the compounding of odometry measurements. Figure 3 shows a vehicle with a prior pose $\boldsymbol{x}_o(k-1)$. The processing of wheel rotations between successive readings (via Eq. 15) has indicated a vehicle-relative transformation (i.e. in the frame of the vehicle) $\boldsymbol{u}_o$. The task of combining this new motion $\boldsymbol{u}_o(k)$ with the old dead-reckoned estimate $\boldsymbol{x}_o(k-1)$ to arrive at a new dead-reckoned posed $\boldsymbol{x}_o(k)$ is trivial. It is simple:

$$\boldsymbol{x}_o(k) = \boldsymbol{x}_o(k-1) \oplus \boldsymbol{u}_o(k) \tag{16}$$

We have now explained a way in which measurements of wheel rotations can be used to estimate dead-reckoned pose. However we set out to figure out a way in which a dead-reckoned pose could be used to form a control input or measurement into a navigation system. In other words we are given the low-level vehicle software to generate a sequence $\boldsymbol{x}_o(1)$, $\boldsymbol{x}_o(2)$, $\cdots$, $\boldsymbol{x}_o(k)$, etc and we want to figure out $\boldsymbol{u}_o(k)$. This is now simple. We can invert Eq. (16) to get

$$\boldsymbol{u}_o(k) = \ominus\boldsymbol{x}_o(k-1) \oplus \boldsymbol{x}_o(k) \tag{17}$$

Looking at the Fig. 3 we can see that the transformation $\boldsymbol{u}_o(k)$ is equivalent to going back along $\boldsymbol{x}_o(k-1)$ and forward along $\boldsymbol{x}_o(k)$. This gives us a small control vector $\boldsymbol{u}_o(k)$ derived from two successive dead-reckoned poses that is suitable for use in another hopefully less error prone navigation algorithm. Effectively Eq. (17) subtracts out the common dead-reckoned gross error.

We are now in a position to write down a plant model for a vehicle using a dead-reckoned position as a control input:

$$\begin{aligned} \boldsymbol{x}_v(k+1) &= \boldsymbol{f}\left(\boldsymbol{x}_v(k), \boldsymbol{u}_o(k)\right) \\ &= \boldsymbol{x}_v(k) \oplus (\ominus\boldsymbol{x}_o(k-1) \oplus \boldsymbol{x}_o(k)) \\ &= \boldsymbol{x}_v(k) \oplus \boldsymbol{u}_o(k) \end{aligned} \tag{18}$$

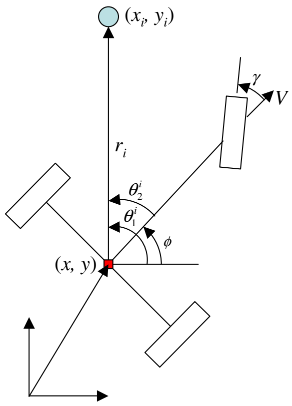


Fig. 4 Vehicle-landmark model

3. PSEUDOLINEAR SYSTEM MODELING

In the following, the vehicle state is defined by $\boldsymbol{x}_v = [x, y, \phi]^{\mathrm{T}}$ where x and y are the coordinates of the center of the rear axel of the vehicle with respect to some global coordinate frame and ϕ is the orientation of the vehicle axis. The landmarks are modeled as point landmarks and represented by a Cartesian pair such that $\boldsymbol{m}_i = [x_i, y_i]^{\mathrm{T}}, i = 1, ..., N$. Both vehicle and landmark states are registered in the same frame of reference.

3.1 The Pseudolinear Process Model

Figure 4 shows a schematic diagram of the vehicle in the process of observing a landmark. The dead-reckoned measurements obtained from successive vehicle frames can be used to predict the vehicle state from the previous state. The discrete-time vehicle process model can be obtained according to the Eq. (18) and expressed in the following form for the vehicle pose:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} \cos(\phi(k)) & -\sin(\phi(k)) \\ \sin(\phi(k)) & \cos(\phi(k)) \end{bmatrix} \begin{bmatrix} dx_o(k \\ dy_o(k) \end{bmatrix} \quad (19)$$

and for the orientation:

$$\phi(k+1) = \phi(k) + d\phi_o(k) \quad (20)$$

Combination of Eqs. (19) and (20) produces the vehicle motion model:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} + \begin{bmatrix} \cos(\phi(k)) & -\sin(\phi(k)) & 0 \\ \sin(\phi(k)) & \cos(\phi(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx_o(k) \\ dy_o(k) \\ d\phi_o(k) \end{bmatrix} \quad (21)$$

which can be represented by the discrete-time pseudolinear vehicle motion model expressed:

$$\boldsymbol{x}_v(k+1) = \boldsymbol{x}_v(k) + \boldsymbol{B}_v(k)\boldsymbol{u}_o(k) \quad (22)$$

for use in the prediction stage of the vehicle state estimator.

The landmarks in the environment are assumed to be stationary point targets. The landmark process model is thus

$$\begin{bmatrix} x_i(k+1) \\ y_i(k+1) \end{bmatrix} = \begin{bmatrix} x_i(k) \\ y_i(k) \end{bmatrix} \quad (23)$$

for all landmarks $i = 1, ..., N$. Equation (21) together with Eq. (23) defines the vehicle-landmarks process model.

3.2 The Observation Model

Range $r_i(k)$ and two bearing measurements $\theta_1^i(k)$ and $\theta_2^i(k)$ to landmark i are recorded by the range and bearing sensors. The vertical axes of two sensors are assumed to be aligned. The range measurements and bearing measurements are taken from the center of rear vehicle axel where the vehicle position (x, y) is taken. One sensor starts reading measurements from the x axis and the other from the center axis of the vehicle. Referring to Fig. 4, the observation model for i^{th}landmark $\boldsymbol{z}_i(k) = [r_i(k), \theta_i(k), \beta_i(k)]^T$ can be written in direct form as

$$r_1^i(k) = r_2^i(k)$$

$$= r_i(k) = \sqrt{(x_i - x(k))^2 + (y_i - y(k))^2} + v_r(k) \quad (24)$$

$$\theta_1^i(k) = \theta_i(k) = \arctan\left(\frac{y_i - y(k)}{x_i - x(k)}\right) + v_{\theta_1}(k) \quad (25)$$

$$\theta_2^i(k) = \arctan\left(\frac{y_i - y(k)}{x_i - x(k)}\right) - \phi(k) + v_{\theta_2}(k) \quad (26)$$

$$\theta_2^i(k) - \theta_1^i(k) = \beta_i(k) = -\phi(k) + v_{\theta_2}(k) - v_{\theta_1}(k) \quad (27)$$

where v_r and v_θ are the white noise sequences associated with the range and bearing measurements with zero means and standard deviations σ_r and σ_θ respectively. The covariance matrix $\boldsymbol{R_z}$ for the observation model given by Eqs. (24), (25) and (27) is then in the form:

$$\boldsymbol{R_z} = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (28)$$

Equations (24) – (27) define the observation model for the i^{th}landmark.

3.2.1 Pseudolinear Observation Model

In this section, we present the pseudolinear measurement model [8] that is employed in the feature based estimation for localization and map building. The pseudomeasurement method relies on representing the nonlinear measurement model (Eqs. (24) – (27)) in the following pseudolinear form:

$$\boldsymbol{y}(\boldsymbol{z}) = \boldsymbol{H}(\boldsymbol{z})\boldsymbol{x} + \boldsymbol{v_y}(\boldsymbol{x}, \boldsymbol{v}) \quad (29)$$

where the pseudomeasurement vector $\boldsymbol{y}(\boldsymbol{z})$ and matrix $\boldsymbol{H}(\boldsymbol{z})$ are *known* functions of the actual measurement $\boldsymbol{z}$ and $\boldsymbol{v_y}(\boldsymbol{x}, \boldsymbol{v})$ is the corresponding pseudomeasurement error, now *state dependent*. The underlying idea of the approach is clear. Once a pseudolinear model (29) is available, a linear Kalman filter can be readily used with

$\boldsymbol{y}(\boldsymbol{z}), \boldsymbol{H}(\boldsymbol{z})$, and $\boldsymbol{R}_{\boldsymbol{y}}(\boldsymbol{x}^*) = \mathrm{cov}[\boldsymbol{v}_{\boldsymbol{y}}(\boldsymbol{x}^*, \boldsymbol{v})]$, where a common choice of $\boldsymbol{x}^*$ is the predicted state estimate $\hat{\boldsymbol{x}}$. Equations (24) – (27) can be rearranged by algebraic and trigonometric manipulations to obtain the following model expressed by

$$r_i(k) = (x_i - x(k))\cos(\theta_i(k)) + (y_i - y(k))\sin(\theta_i(k)) + v_r(k) \tag{30}$$

$$0 = (x_i - x(k))\sin(\theta_i(k)) - (y_i - y(k))\cos(\theta_i(k)) + v_{\theta p}(k) \tag{31}$$

$$\beta_i(k) = -\phi(k) + v_{\theta_2}(k) - v_{\theta_1}(k) \tag{32}$$

The model given by Eqs. (30) – (32) can be expressed in the following pseudolinear form for the i^{th} landmark:

$$\boldsymbol{y}(\boldsymbol{z}_i) = \begin{bmatrix} r_i(k) \\ 0 \\ \beta_i(k) \end{bmatrix} = \boldsymbol{H}(\boldsymbol{z}_i)\boldsymbol{x} + \boldsymbol{v}_{\boldsymbol{y}i}(\boldsymbol{x}, \boldsymbol{v}) \tag{33}$$

where

$$\boldsymbol{H}(\boldsymbol{z}_i) = \left[\begin{matrix} -\cos(\theta_i(k)) & -\sin(\theta_i(k)) & 0 & 0 & \cdots \\ -\sin(\theta_i(k)) & \cos(\theta_i(k)) & 0 & 0 & \cdots \\ 0 & 0 & -1 & 0 & \cdots \end{matrix}\right.$$
$$\left.\begin{matrix} 0 & \cos(\theta_i(k)) & \sin(\theta_i(k)) & 0 & \cdots & 0 \\ 0 & \sin(\theta_i(k)) & -\cos(\theta_i(k)) & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{matrix}\right] \tag{34}$$

where state vector $\boldsymbol{x} = [\boldsymbol{x}_v^T \ \ \boldsymbol{m}^T]^T$ and $\boldsymbol{v}_{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{v})$ is considered to be white and its covariance is expressed in the form:

$$\boldsymbol{R}_{\boldsymbol{y}}(\hat{\boldsymbol{x}}) = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \hat{r}_i^2\sigma_\theta^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{35}$$

4. TAKAGI-SUGENO (T-S) FUZZY MODEL

The fuzzy model proposed by Takagi and Sugeno is described by fuzzy IF-THEN rules, which represent local linear input-output relations of a nonlinear system. The i^{th} rule of the T-S fuzzy model is of the following form:

Rule i :
IF $z_1(k)$ is M_{i1} and $\cdots$ and $z_p(k)$ is M_{ip}

$$\text{THEN} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{C}_i\boldsymbol{x}(k) \quad i = 1, 2, \cdots, r. \end{cases} \tag{36}$$

M_{ij} is the fuzzy set and r is the number of IF-THEN rules. $\boldsymbol{x}(k) \in \boldsymbol{R}^n$ is the state vector, $\boldsymbol{u}(k) \in \boldsymbol{R}^m$ is the input vector, $\boldsymbol{y}(k) \in \boldsymbol{R}^q$ is the measurement vector, $\boldsymbol{A}_i \in \boldsymbol{R}^{n\times n}$, $\boldsymbol{B}_i \in \boldsymbol{R}^{n\times m}$, and $\boldsymbol{C}_i \in \boldsymbol{R}^{q\times n}$. $z_1(k), ..., z_p(k)$ are the premise variables. Each linear consequent equation represented by $\boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k)$ is called "subsystem."

Given a pair of $(\boldsymbol{x}(k), \boldsymbol{u}(k))$, the final outputs of the fuzzy systems are inferred as follows:

$$\boldsymbol{x}(k+1) = \frac{\sum_{i=1}^r w_i(\boldsymbol{z}(k))\{\boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k)\}}{\sum_{i=1}^r w_i(\boldsymbol{z}(k))}$$

$$= \sum_{i=1}^r h_i(\boldsymbol{z}(k))\{\boldsymbol{A}_i\boldsymbol{x}(k) + \boldsymbol{B}_i\boldsymbol{u}(k)\} \tag{37}$$

where

$$\boldsymbol{z}(k) = [z_1(k) \ \ z_2(k) \ \ \cdots \ \ z_p(k)]$$

$$w_i(\boldsymbol{z}(k)) = \prod_{j=1}^p M_{ij}(z_j(k))$$

$$\begin{cases} \sum_{i=1}^r w_i(\boldsymbol{z}(k)) > 0 \\ w_i(\boldsymbol{z}(k)) \geq 0 \end{cases} \quad i = 1, 2, \cdots, r \tag{38}$$

$$h_i(\boldsymbol{z}(k)) = \frac{w_i(\boldsymbol{z}(k))}{\sum_{i=1}^r w_i(\boldsymbol{z}(k))}$$

for all k. $M_{ij}(z_j(k))$ is the grade of membership of $z_j(k)$ in M_{ij}. From Eqs. (37) – (38) we have

$$\begin{cases} \sum_{i=1}^r h_i(\boldsymbol{z}(k)) = 1 \\ h_i(\boldsymbol{z}(k)) \geq 0 \end{cases} \quad i = 1, 2, \cdots, r \tag{39}$$

for all k.

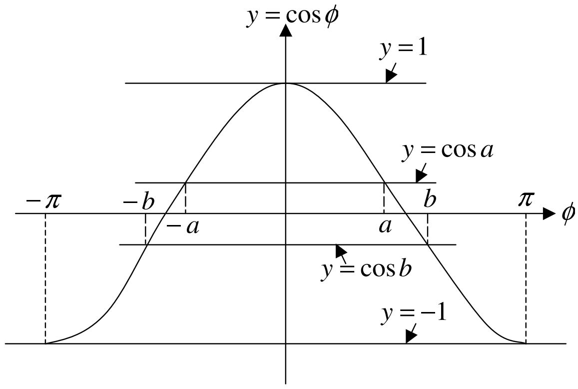


Fig. 5 Approximation of nonlinear term $\cos\phi$

4.1 Fuzzy Modeling of Nonlinear Terms

Fuzzy description of nonlinear term $\cos\phi$ can be expressed as follows. It is assumed that ϕ varies in between $-\pi$ and π. $\cos\phi$ can be approximated for two cases by using two linear models for each case. This is illustrated in Fig. 5. They can be represented as follows

$$\cos\phi = F_1^1(\phi)\cdot 1 + F_1^2(\phi)\cdot \cos a \quad \text{for } |\phi| \leq \pi/2 \tag{40}$$

$$\cos\phi = F_2^1(\phi)\cdot(-1) + F_2^2(\phi)\cdot\cos b \quad \text{for } \pi/2 < |\phi| < \pi \tag{41}$$

where

$$F_1^1(\phi), F_1^2(\phi), F_2^1(\phi), F_2^2(\phi) \in [0, 1]$$
$$F_1^1(\phi) + F_1^2(\phi) = 1, \quad F_2^1(\phi) + F_2^2(\phi) = 1 \tag{42}$$

Solving the above equation gives

$$F_1^1(\phi) = \frac{\cos\phi - \cos a}{1 - \cos a}$$

$$F_1^2(\phi) = 1 - F_1^1(\phi) = \frac{1 - \cos\phi}{1 - \cos a}$$

$$F_2^1(\phi) = \frac{\cos b - \cos\phi}{1 + \cos b}$$

$$F_2^2(\phi) = 1 - F_2^1(\phi) = \frac{1+\cos\phi}{1+\cos b} \quad (43)$$

In the same way, $\sin\phi$ can also be approximated by linear models and can be deduced from approximated $\cos\phi$ by the following formula:

$$\sin\phi = \mathrm{sgn}(\phi)\sqrt{1-\cos^2\phi} \quad (44)$$

where

$$\mathrm{sgn}(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ -1 & \text{if } \phi < 0 \end{cases}$$

5. FORMATION OF FUZZY ALGORITHM IN SLAM PROBLEM

To reduce the computational cost in using the T-S fuzzy model in SLAM problem, fuzzification of the process model and the pseudolinear measurement model is split into two cases based on the value of the angle of the vehicle. A set of fuzzy rules is formed for each case and is executed based on the initial separation of vehicle angle.

Case 1:
If the angle of vehicle ($\phi(t)$) lies between $-\pi/2$ and $\pi/2$, the j^{th} rule for this case will be of the form:

Local linear system rule j:
IF $\phi(t)$ is F_ϕ^j and $\theta_i(t)$ is F_θ^j THEN

$$\boldsymbol{x}_j(k+1) = \boldsymbol{x}(k) + \boldsymbol{B}_j(k)\boldsymbol{u}(k) \quad \text{for } j = 1, 2, \cdots, 8.$$

$$\boldsymbol{y}_{ij}(k+1) = \boldsymbol{H}_{ij}(k+1)\boldsymbol{x}_j(k+1) + \boldsymbol{v}^i_{\boldsymbol{y}_j}(k+1) \quad (45)$$

F_ϕ^j and F_θ^j are the fuzzy sets of vehicle angel and measurement angle for the j^{th} rule respectively. These membership functions are defined based on the sectored region of the cosine angle of the premise variables. Calculation of grade of membership function of the premise variables is discussed and illustrated in the Section 4.1. Possible combinations of two membership functions will provide the conditions to linearize the nonlinear system and obtain the local linear systems accordingly.
where

$$F_\phi^j, F_\theta^j \in \{F_1^1, F_1^2, F_2^1, F_2^2\}, \quad \boldsymbol{u} = [\boldsymbol{u}_o \;\; \boldsymbol{0}_1]^T$$

$$\boldsymbol{B}_j = \begin{bmatrix} \boldsymbol{B}_j^v & \boldsymbol{0}_2 \\ \boldsymbol{0}_2^T & \boldsymbol{0}_3 \end{bmatrix}, \quad \boldsymbol{H}_{ij} = [\boldsymbol{H}_{ij}^v \;\; \boldsymbol{0}_4 \;\; \boldsymbol{H}_{ij}^f \;\; \boldsymbol{0}_5]$$

$\boldsymbol{u} \in \boldsymbol{R}^{(3+2N)\times 1}$, $\boldsymbol{u}_o \in \boldsymbol{R}^{3\times 1}$, $\boldsymbol{0}_1 \in \boldsymbol{R}^{2N\times 1}$, $\boldsymbol{B}_j \in \boldsymbol{R}^{(3+2N)\times(3+2N)}$, $\boldsymbol{B}_j^v \in \boldsymbol{R}^{3\times 3}$, $\boldsymbol{0}_2 \in \boldsymbol{R}^{3\times 2N}$, $\boldsymbol{0}_3 \in \boldsymbol{R}^{2N\times 2N}$, $\boldsymbol{H}_{ij} \in \boldsymbol{R}^{3\times(3+2N)}$, $\boldsymbol{H}_{ij}^v \in \boldsymbol{R}^{3\times 3}$, $\boldsymbol{H}_{ij}^f \in \boldsymbol{R}^{3\times 2}$, $\boldsymbol{0}_4 \in \boldsymbol{R}^{3\times 2m}$, $\boldsymbol{0}_5 \in \boldsymbol{R}^{3\times 2n}$
where $m+n+1 = N$ is the number of features added to the state vector. 'i' denotes the i^{th} feature being observed and note that

$$\boldsymbol{B}_j^v = \begin{bmatrix} \cos(\phi_j(k)) & -\sin(\phi_j(k)) & 0 \\ \sin(\phi_j(k)) & \cos(\phi_j(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (46)$$

$$\boldsymbol{H}_{ij}^v = \begin{bmatrix} -\cos(\theta_{ij}(k)) & -\sin(\theta_{ij}(k)) & 0 \\ -\sin(\theta_{ij}(k)) & \cos(\theta_{ij}(k)) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (47)$$

$$\boldsymbol{H}_{ij}^f = \begin{bmatrix} \cos(\theta_{ij}(k)) & \sin(\theta_{ij}(k)) \\ \sin(\theta_{ij}(k)) & -\cos(\theta_{ij}(k)) \\ 0 & 0 \end{bmatrix} \quad (48)$$

where $\boldsymbol{B}_j^v$ is the matrix with its nonlinear elements linearized as discussed in fuzzy description of nonlinear terms in the Section 4.1 and then it becomes a linearized matrix for fuzzy sets of vehicle angle (ϕ) for each rule in T-S fuzzy model of SLAM problem. In the similar way, the nonlinear elements of the matrices $\boldsymbol{H}_{ij}^v$ (observation matrix w.r.t the vehicle pose) and $\boldsymbol{H}_{ij}^f$ (observation matrix w.r.t the feature pose) are linearized for fuzzy sets of measurement angle (θ).

Case 2 is defined for $\pi/2 < |\phi(t)| < \pi$ and will be composite of eight similar local linear models based on the vehicle angle in the range of $\phi(t)$ defined above.

5.1 Estimation Process

In the formulation of T-S fuzzy model based SLAM algorithm, the linear discrete Kalman filter is used to provide local estimates of vehicle and landmark locations for each local linear model defined in T-S fuzzy model. The Kalman filter algorithm proceeds recursively in the three stages:

- Prediction:

The algorithm first generates a prediction for the state estimate, the observation (relative to the i^{th} landmark) and the state estimate covariance at the time $k+1$ for the j^{th} rule according to

$$\hat{\boldsymbol{x}}_j(k+1|k) = \hat{\boldsymbol{x}}(k|k) + \boldsymbol{B}_j(k)\boldsymbol{u}(k) \quad (49)$$
$$\hat{\boldsymbol{y}}_{ij}(k+1|k) = \boldsymbol{H}_{ij}(k+1)\hat{\boldsymbol{x}}_j(k+1|k) \quad (50)$$
$$\boldsymbol{P}_j(k+1|k) = \boldsymbol{P}(k|k) + \boldsymbol{B}_j(k)\boldsymbol{Q}(k)\boldsymbol{B}_j^T(k) \quad (51)$$

- Observation:

Following the prediction, the observation $\boldsymbol{y}_i(k+1)$ of the i^{th} landmark of the true state $\boldsymbol{x}(k+1)$ is made according to Eq. (33). Assuming correct landmark association, an innovation is calculated for the j^{th} rule as follows:

$$\boldsymbol{\nu}_{ij}(k+1) = \boldsymbol{y}_{ij}(k+1) - \hat{\boldsymbol{y}}_{ij}(k+1|k) \quad (52)$$

together with an associated innovation covariance matrix for the j^{th} rule given by

$$\boldsymbol{S}_{ij}(k+1) = \boldsymbol{H}_{ij}(k+1)\boldsymbol{P}_j(k+1|k)\boldsymbol{H}_{ij}^T(k+1) + \boldsymbol{R}_{ij}(k+1) \quad (53)$$

- Update:

The state update and corresponding state estimate covariance are then updated for the j^{th}rule according to

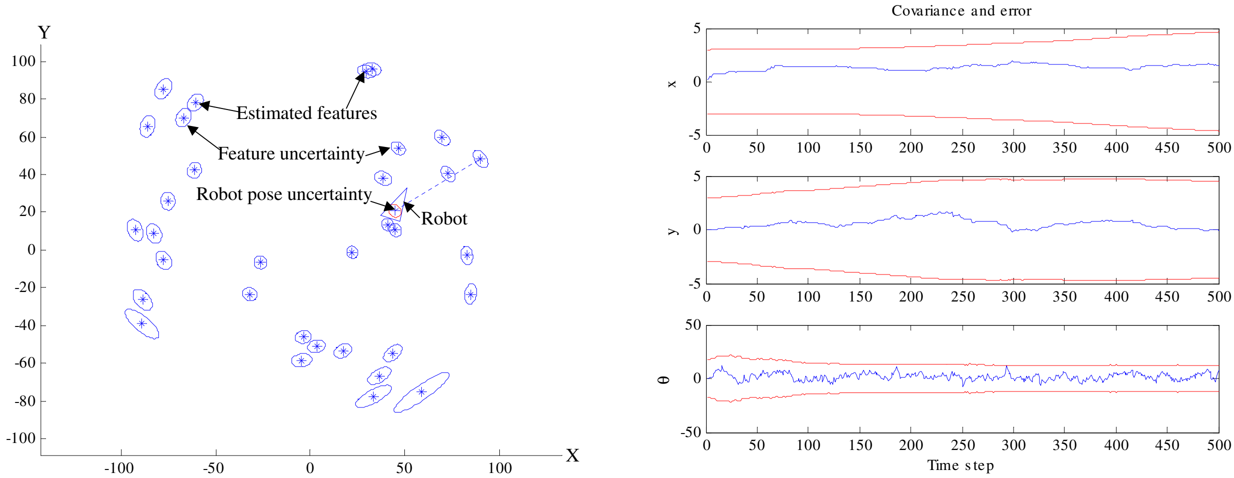
$$\hat{\boldsymbol{x}}_j(k+1|k+1) = \hat{\boldsymbol{x}}_j(k+1|k) + \boldsymbol{K}_j(k+1)\boldsymbol{\nu}_{ij}(k+1) \quad (54)$$
$$\boldsymbol{P}_j(k+1|k+1) = \boldsymbol{P}_j(k+1|k) - \boldsymbol{K}_j(k+1) \times \boldsymbol{S}_{ij}(k+1)\boldsymbol{K}_j^T(k+1) \quad (55)$$

Here the gain matrix $\boldsymbol{K}_j(k+1)$ is given by

$$\boldsymbol{K}_j(k+1) = \boldsymbol{P}_j(k+1|k)\boldsymbol{H}_{ij}^T(k+1)\boldsymbol{S}_{ij}^{-1}(k+1) \quad (56)$$

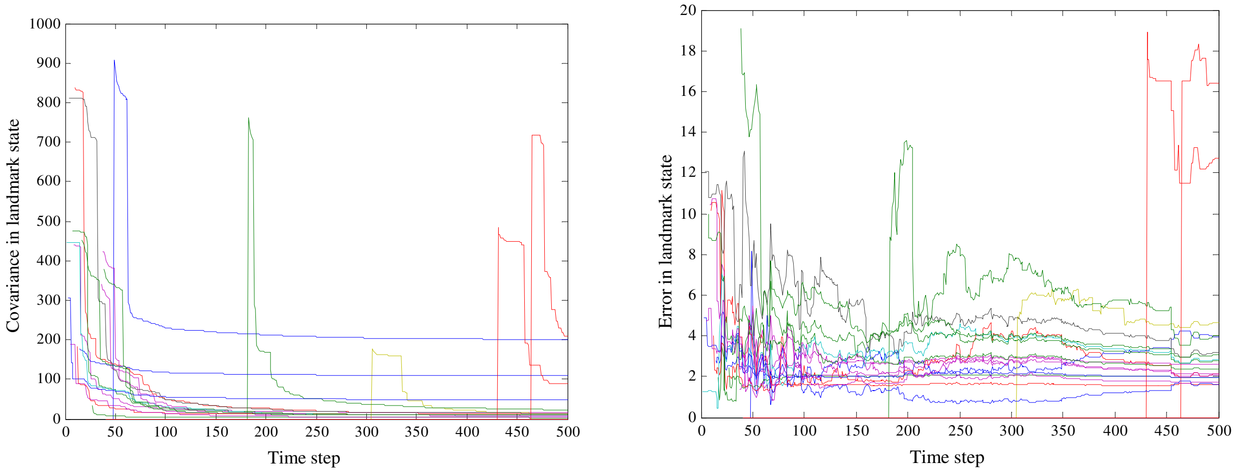
Local state estimates are then combined according to the Eq. (37) to obtain the global state estimate for the T-S fuzzy model given by Eq. (45). The global estimate is then obtained by the following equation.

$$\hat{\boldsymbol{x}}(k+1|k+1) = \sum_{j=1}^{8} h_j(\boldsymbol{z}_i(k))\hat{\boldsymbol{x}}_j(k+1|k+1) \quad (57)$$



(a) (b)

Fig. 6 Simultaneous localization and map building



(a) (b)

Fig. 7 Evolution of landmark state covariance and error

where $\boldsymbol{z}_i(k) = [z_{i1}(k) \;\; z_{i2}(k)] = [\phi(k) \;\; \theta_i(k)]$.

Propagation of uncertainty for the augmented state error of the T-S fuzzy model is realized by a common covariance which is chosen to be the local covariance which has the minimum *trace*. This assures the stability of the T-S fuzzy model based SLAM algorithm as it is of paramount importance in state estimation using fuzzy algorithm. The common covariance can be formulated as follows:

$$\boldsymbol{P}(k+1|k+1) = \min(\text{trace } (\boldsymbol{P}_j(k+1|k+1))) \; \forall j \tag{58}$$

The resulting global state estimate and common covariance are then proceeded to the next stage of prediction. Each rule in the T-S fuzzy model takes the global state estimate and the common covariance to generate the next stage prediction. This process is repeated until the required criteria for the state estimation is met.

6. SIMULATION RESULTS

In this section, we show the simulation results for the feature (landmarks) based SLAM for the system composite of Eqs. (21), (23) and (33) while assuming the initial estimate and covariance to start estimation process.

6.1 Simultaneous Localization and Map Building

The newly described method is applied to the feature based Simultaneous Localization and Map Building. Here we are given a stream of observations of measurements between the vehicle and these features. We assume the vehicle we are navigating is equipped with range-bearing sensors which return the range and bearing to point like objects (the features). Here, the proposed method was simulated using Matlab for the system presented in this paper. An environment populated with point landmarks was simulated with the fuzzy Kalman filter algorithm discussed above to generate the state estimates and state errors. The proposed SLAM algorithm was recursively processed over 500 time steps in the simulation. Simulation results are depicted in Fig. 6 and Fig. 7.

Figure 6 shows localization of the vehicle and map building simultaneously over time. Figure 6(a) shows covariance and error associated with the vehicle pose. It can be seen that the vehicle localization is perfectly performed by the newly presented method as vehicle pose error is decreasing to a minimum bound gradually. Figure 6(b) shows the estimated map built over 500 time steps. It can be seen that error ellipses of the features are getting converged as the map of the landmark locations is being built when the vehicle navigates through the environment. It is clear that the newly proposed algorithm can well map the environment while letting the vehicle localize itself in the environment accurately. This feature can be observed in Fig. 7. Figure 7(a) shows the evolution of the landmark uncertainty and it can be observed that the landmark uncertainty is gradually decreasing over time. Figure 7(b) shows the evolution of landmark state error and it is once proved that the proposed method works well in SLAM problem. It is observed that the landmark state error obtained from the pseudolinear model approach reaches to a minimum bound within a shorter time steps compared to that obtained from the EKF algorithm.

7. CONCLUSION

A fuzzy logic and pseudolinear model based solution to the SLAM problem was first proposed in this paper and validity of the method was proved with simulation results. The need for direct linearization of nonlinear systems for state estimation is diminished as the newly proposed method performed well and provided a better solution to the SLAM problem. Results obtained from the newly introduced method were compared with the results obtained from widely used EKF algorithm to highlight the merit of the pseudolinear model based system with fuzzy logic. It was proved that the pseudolinear model based fuzzy Kalman filter algorithm provided more satisfactory results over the EKF because the pseudolinear models did not lose its nonlinearity when employed in the Kalman filter equations. It was found that a fuzzy logic based approach with the pseudolinear models provided a remarkable solution to state estimation process because fuzzy logic has been always standing for a better solution.

Here we want to draw the attention on some other filters being used for the state estimation of nonlinear systems. There are some state estimators which fulfill the requirements of the problem settings and perform well in filtering out the states of nonlinear systems. The unscented Kalman filter (UKF) and the particle filter (PF) have been known to perform well. Certain points can be placed here to distinguish the newly proposed method for state estimation based on fuzzy Kalman filtering with pseudolinear system modeling from the UKF and the PF. It was experienced and proved that the extended Kalman filter gives particularly poor performance on highly nonlinear functions. The unscented Kalman filter (UKF) uses a deterministic sampling technique to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the nonlinear functions and the covariance of the estimate is then recovered. The result is a filter which more accurately captures the true mean and covariance. The basic idea behind the PF is that it maintains lots of different versions of the state vector. When a measurement comes in, we score how well each version explains the data. We do not have to make the assumption of Gaussian noise or perform a linearlization. The big problem with this elegant approach is the number of sample-states. We need to try out to increase geometrically with the number of states being estimated. For example, to estimate a 1D vector we may keep 100 samples. For a 2D vector we would require something like 100^2 and so on. Though the UKF and the PF perform well, computational complexity involved in the estimation process is enormous and it can not be considered to be simple and a handy tool for each and every case in which these techniques are applied. In addition, our proposed technique removes the requirement to analytically calculate Jacobians. We can conclude here that the newly proposed method overcomes the complexities in estimation and performs well producing much anticipated results.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automaton Magazine*, vol. 13, no. 2, pp. 99–108, June 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automaton Magazine*, vol. 13, no. 3, pp. 108–117, Sept. 2006.

[3] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. 4th Int. Symp. Robot. Res.*, pp. 467–474, 1987.

[4] J. J. Leonard and H. F. Durrant-White, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[5] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, June 2001.

[6] X. R. Li and V. P. Jikov, "A survey of maneuvering target tracking–part III: measurement models," in *Proc. SPIE Conference on Signal and Date Processing of Small Targets*, San Diego, CA, USA, July-August 2001.

[7] D. W. Whitcombe, "Pseudo state measurements applied to recursive nonlinear filtering," in *Proc. 3rd Symp. Nonlinear Estimation Theory and Its Application*, San Diego, CA, pp. 278-281, Sept. 1972.

[8] K. Watanabe, *Adaptive Estimation and Control: Partitioning Approach*, London, UK: Prentice-Hall, 1991.

[9] G. Chen, Q. Xie and L. S. Shieh,"Fuzzy Kalman filtering," *Journal of Information Sciences*, no. 109, pp. 197–209, 1998.

[10] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.,* vol. 15, pp. 116–132, 1985.