International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Mobile Robot Localization in Indoor Environments Using Fuzzy Adaptive Unscented Kalman Filter and Random Tree Routing Algorithm with Fast Exploration

Mohamadrasol Hajiali[1] | Ramazan Havangi[2]
Faculty of Electrical Engineering and Computer, University of Birjand, Iran.[1,2]
Corresponding author's email: Havangi@Birjand.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 10-July-2024
Received in revised form: 25-November-2024
Accepted: 11-December-2024
Published online: 23-Sep-2025

Keywords:
Fuzzy,
Random tree routing,
Robot localization,
Unscented Kalman filter.

ABSTRACT

In the field of mobile robot navigation, challenges such as nonlinear conditions, uncertainties, and the advancement of methods have made accurate position estimation essential. This study evaluates the effectiveness of a fuzzy-based adaptive unscented Kalman filter (FAUKF) for improving the state estimation accuracy for mobile robot localization. The proposed approach leverages the FAUKF algorithm to address noise uncertainty effectively by adaptively adjusting the covariance of the measurement noise based on a defined adaptation law. The Mamdani Fuzzy Inference System (FIS) serves as an observer, enhancing the matching law and improving overall system performance. The findings of this study demonstrate that the FAUKF algorithm provides superior position estimation accuracy compared to conventional Unscented Kalman Filter (UKF) methods. Furthermore, the research introduces an innovative navigation framework for mobile robots by integrating the Random Tree Routing algorithm with Rapidly exploring Random Tree Star (RRT*) for optimal path planning in indoor environments. The RRT* integration aims to generate efficient and optimal paths while addressing safety considerations and environmental constraints. By combining the prediction and update phases of the Kalman filter, the proposed methodology effectively minimizes the propagation of uncertainty during the localization process, thereby enabling precise localization and robust path planning for designated targets. The simulation results confirmed the effectiveness of this method in maintaining constant uncertainty levels in localization over time. The proposed adaptive method enables efficient navigation in complex environments. Path planning is a critical element in robotics applications, and the RRT*-based approach presented herein offers a comprehensive solution for generating optimal and efficient paths. By providing an up-to-date perspective, this research contributes to the evolving landscape of mobile robot localization methods. The proposed method highlights the importance of utilizing adaptive algorithms and advanced path-planning techniques to enhance navigation capabilities in indoor environments.

NOMENCLATURE

v	Forward speed of the robot.	Q_k	Covariance matrix of procedure noise at moment k.
w	The robot's turning speed.	R_k	Covariance matrix of measurement noise at moment k.
θ	Orientation changes of the robot.	Z_k	Measurements of sigma points at time k.
L	The distance between the wheels.	$\overline{Z}_k$	Weighted average of forecast measurements at time k.
R	Radius of the robot wheel.	$P_{Z_kZ_k}$	Predicted measurement covariance at time k.
$\hat{X}_k$	Estimation of state at moment k.	$P_{X_kZ_k}$	Covariance between measurement and state at time k.
u_k	control input.	C_T	Theoretical covariance.
V_k	The measurement noise matrix at moment k.	C_A	True residual covariance.

I. Introduction

With technological improvements in the present era, the importance of mobile robots in various operational applications has increased. The discussion on mobile robot localization has undergone significant progress, but there are still challenges in developing efficient and reliable localization strategies for robots operating in indoor

environments. Existing methods are often associated with complex and dynamic environments, leading to inappropriate and suboptimal performances. The categorization of mobile robots is based on both operational specifications and the specific environmental conditions in which they function. In the current technology landscape, mobile robots play an important role in various operational scenarios. Mobile robots face complex localization challenges. Understanding and optimizing the localization patterns of these robots are essential for their effective navigation and performance. In [1], regarding the localization of mobile robots, localization problems were analyzed and showed that position tracking accounted for 55% of the challenges, followed by global localization with 26%, and the issue of retrieving a kidnapped robot with 19%. These are among the significant challenges in the field of mobile-robot localization. Accurate position tracking is the biggest challenge for autonomous robots owing to the complexity and variability of localization across different environments. In [2], mobile robots can be classified based on their movement systems, including fixed (e.g., robotic arms), ground-based, aerial, and underwater configurations. The choice of robot design is significantly influenced by the operational environment and application requirements. Mobile robots have important applications in industrial, medical, and therapeutic fields, social services, agricultural work, space research, and exploration. A moving robot requires accurate localization and high navigation accuracy to achieve control goals. Localization is a serious problem for mobile robots. For optimal navigation, the robot progresses through several key stages, including perception, localization, recognition, and motion control. During the perception phase, the robot analyzes sensor data to derive valuable insights. In the localization stage, the robot determines its present position within an operational setting by using external sensor data. Subsequently, in the recognition phase, the robot strategizes the actions necessary to achieve its objectives. The motion control step enables the robot to navigate its intended path by adjusting its motor function. Over the past decade, localization has emerged as a focal point in extensive research [3]. The odometry calculation algorithm is a common localization method used in the navigation processes of mobile robots [4-5]. Selecting a filtering method is crucial for acquiring precise assessments of the target states. Robot position estimation involves determining the current and future status of the robot by analyzing data from a moment ago and previous data [7]. In robot localization applications, states usually represent parameters such as position, velocity, and acceleration of the robot. Measurements or observations may include variables such as sensor beam angle, robot distance to obstacles, wheel odometer information, and other relevant parameters based on application-specific needs. UKF is a widely used filtering method for nonlinear systems characterized by higher-order nonlinear systems and complex models. Experimental evidence shows that the UKF provides a higher estimation accuracy than the EKF. In [30], a novel adhesion estimation method for rail systems is presented, combining an extended Kalman filter (EKF) and particle swarm optimization (PSO). This method addresses challenges in adhesion estimation and demonstrates that AI and machine learning techniques improve accuracy, potentially benefiting the rail transport industry.

In the UKF framework, it is assumed that the covariances of the process noise and measurement noise remain constant during the estimation process. In reality, the majority of nonlinear systems encounter uncertainties that are dynamic and unpredictable in their nature. To achieve a higher estimation accuracy with the UKF algorithm, adjusting or adapting the noise covariances during the estimation process is essential [8]. Various methods for matching noise covariances can be found in the literature. In [9], a Differential Drive Mobile Robot (DDMR) was proposed that uses fuzzy logic for obstacle avoidance. The robot processes the ultrasonic sensor inputs to control its motor speed, enabling effective navigation in a low-cost design. The primary trait of the FLC is its capability to regulate a nonlinear dynamic system by establishing a mapping between the input and output parameters, based on the designer or expert input, without relying on mathematical implementations [10]. Several studies have been conducted on the capabilities of the adaptive approach based on fuzzy innovation with the UKF for nonlinear systems. In [11], the use of an unscented Kalman filter for vehicle state estimation is investigated, along with an exploration of adaptive fading. In [12], a study is presented on Doppler-bearing passive target tracking (DBT) with nonlinear measurement equations, which pose notable challenges to tracking accuracy. This research focuses on improving the efficiency of sigma-point Kalman filtering by developing a transformed unscented simplex cubature Kalman filter designed to minimize dependency on prior statistical knowledge. The filter is initialized via a range-parameterized Unscented Kalman Filter, and its effectiveness is verified through Monte Carlo simulations, with tracking accuracy assessed against the Cramér-Rao lower bound (CRLB).

A comparative study of the Adaptive Fuzzy Extended Kalman Filter (AFEKF) and FAUKF for state estimation in unmanned aerial vehicles (UAVs), based on real flight data from a fixed-wing aircraft is presented in [13]. This analysis illustrates the superior performance of FAUKF in enhancing the estimation accuracy and reducing maneuvering errors in fixed-wing UAV operations, marking a notable improvement over the AFEKF method. Additionally, the combination of the adaptive fuzzy unscented Kalman filter with an adaptive fuzzy neural extension has shown promising potential for advancing the effectiveness of state estimation algorithms in

UAV applications, with particular relevance for motion planning algorithms.

Reference [14] highlighted the application of the RRT path-planning algorithm in managing robotic baggage trolleys within airport environments and demonstrated its efficacy in mitigating collision risks with both static and dynamic obstacles. A notable innovation of this approach lies in its deviation from traditional methods, which rely exclusively on rooted trees for state-space exploration. Instead, it employs a structure that traverses multiple subtrees, thereby offering greater adaptability. Each subtree independently evaluates its surroundings, while the main rooted tree consolidates this information to accelerate progress towards the target state. Comprehensive evaluations, including simulations and real-world scenarios, indicate that the RRT algorithm outperforms one-way and two-way risk-based algorithms in terms of scheduling efficiency and resilience.

Reference [15] introduced an algorithm designed to overcome the challenges of sampling efficiency and convergence rates encountered by RRT* variants in environments characterized by extended corridors. By integrating heuristic sampling into path expansion, the algorithm enables RRT* to explore the environment swiftly and iteratively optimize the sampling region to identify the most efficient path between the start and goal points. Comparative simulation results indicate that RRT* offers improved node utilization, faster convergence, and produces higher-quality paths than the traditional RRT algorithm within an equivalent number of iterations.

In summary, RRT* is a valuable algorithm for addressing path-planning challenges in complex environments. This study advances the field of autonomous navigation for mobile robots in indoor settings by introducing an innovative combination of a fuzzy-based adaptive unscented Kalman filter and random tree routing algorithm. The primary objective is to develop a robust and adaptive navigation system capable of efficiently operating in intricate and dynamic indoor spaces. This study aims to establish an optimal automatic localization strategy for mobile robots in indoor environments, demonstrating the effectiveness of the proposed approach in achieving precise navigation and localization. By integrating the adaptive unscented Kalman filter with fuzzy logic and random tree routing, this study provides valuable insights into mobile robot localization, offering practical implications for various indoor applications.

The structure of this paper is organized as follows: Section 2 provides a comprehensive overview of the robot localization problem, along with relevant equations. Section 3 delves into the localization analysis, detailing the application of the fuzzy adaptive unscented Kalman filter and RRT* path-planning algorithm, along with their associated equations. Finally, Section 4 presents the simulation results and a detailed evaluation.

II. Methodology

A. The problem of localization

The localization of mobile robots in indoor environments poses a complex challenge owing to the presence of obstacles, uncertainty of sensors, and need for real-time localization. Traditional localization strategies often have problems in dynamic environments and may not be able to realize optimal solutions in complex scenarios. A mobile robot tracks its position using an odometer while moving in a known environment. Nonetheless, owing to the inaccuracies in odometer readings, the robot's current position becomes uncertain during the localization process. Therefore, it is necessary to localize the robot based on the measurement noise. Using an adaptive approach ensures that the uncertainty in localization does not increase. External sensors such as lasers, vision, and lidar are employed to detect the operating environment and facilitate the localization of the robot. The information from the sensors can be combined with the odometer information of the robot's wheels to localize the robot accurately. Even with a global positioning sensor (GPS), the exact position of a robot cannot be measured directly [1]. Data from the robot's sensors can only be extracted to provide information about the best estimate of the robot's position. Beliefs related to robot configurations are usually represented as probability density functions (PDF) [1]. During the prediction update, the robot used external sensors to estimate its position. However, owing to the odometry error, the uncertainty of the robot configuration is greatly increased, and adaptive approaches need to be applied. The kinematics of a robotic system is intricately related to the configuration and characteristics of its wheels. Kinematics play an important role in determining the movement capabilities of robots. A common and basic configuration is the differential-drive system, which has two wheels of the same size.

The differential drive system manages both forward speed and steering angle. The kinematic model of the robot is shown in Fig.1, which allows precise calculation of the robot's position with a comprehensive understanding of its structure. Estimating the position of a mobile robot poses a challenge because sensor measurements are not directly available. Factors such as motion inaccuracy and wheel slip require an accurate robot model that limits uncertainties. The geometric characteristics of each wheel not only affect the motion of the robot but also impose limitations. For example, lateral slip in the wheels can lead to deflection and uncertainty in localization. By understanding and combining these kinematic principles, robotic systems can increase their localization accuracy and operational efficiency. Fig.2 shows the position of a robot according to the global coordinate

system. The vector represents the position of the robot at time step and is expressed as follows:

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (1)$$

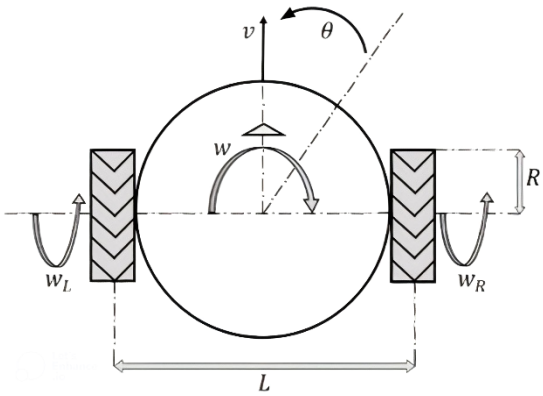


Fig.1. Kinematic model of moving robot.

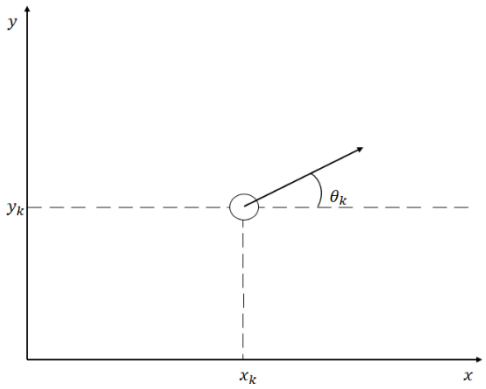


Fig.2. Position of the robot represented by coordinates (x_k, y_k, θ_k) in 2D cartesian space. The robot's orientation is represented by θ_k.

The position vector X_k consists of the x_k and y_k coordinates, which represent the robot's position in the plane, along with θ_k, the orientation of the robot, indicating its direction. Fig.1 shows the kinematic model of the moving robot. The movement of the robot is characterized by two key parameters: the forward speed v and the rotational speed w, which describe the robot's linear velocity and angular velocity, respectively. The wheel radius of the robot is represented by R, and the distance between the wheels is denoted by L. The parameter θ indicates the angular change in the robot's direction.

The robot workspace defines the range of possible positions that the robot can achieve. The wheel radius of the robot and the distance between the wheels are indicated by R and L, respectively, and θ represents the amount of angle changes. Referring to Fig.3, the distance traveled by the left wheel Δ_{s_l} and the right wheel Δ_{s_r} along with the total distance traveled by the robot which is indicated by Δs. The angle α represents the orientation of the rod connecting the wheels relative to the horizontal axis. The equations related to the calculation of Δ_{s_r} and Δ_{s_l} as well as Δs are as follows:

$$\Delta_{s_l} = R\alpha \quad (2)$$

$$\Delta_{s_r} = (R + L)\alpha \quad (3)$$

$$\Delta s = \frac{\Delta_{s_r} + \Delta_{s_l}}{2} \quad (4)$$

According to Fig.3, the distance traveled by the left wheel and the right wheel can be calculated. In the next step, the new angle of the robot is calculated as follows:

$$\Delta\theta = \frac{\Delta_{s_r} - \Delta_{s_l}}{L} \quad (5)$$

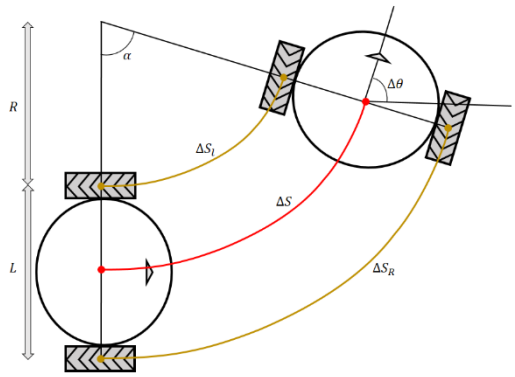


Fig.3. A simple rotation example that shows the basic parameters of the robot's kinematics.

Since the method of calculating the angle and distance traveled by the robot wheels is already available, it is possible to calculate the position of the robot. According to Fig.4, it is possible to obtain the relevant equations.

The current position of a moving robot can be estimated from its initial position information by calculating the coordinate changes over time $(\Delta x, \Delta y, \Delta\theta)$ and finally the odometry distance can be calculated as follows:

$$\Delta x = \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \quad (6)$$

$$\Delta y = \Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right) \quad (7)$$

The actual state is a function that depends on the previous state. In Equation (8), X_k represents the robot's current position, while X_{k-1} denotes its position at the preceding time step $k-1$. Based on (8), the robot's current position can be determined using its previous position. It is important to highlight that the system under consideration operates in a discrete framework.

$$X_k = X_{k-1} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = X_{k-1} + \begin{bmatrix} \Delta s \cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta s \sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \Delta\theta \end{bmatrix} \quad (8)$$

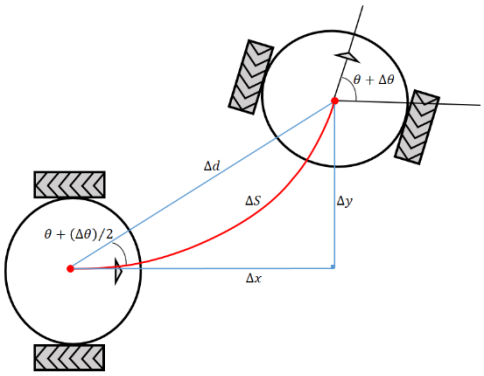


Fig.4. Kinematic parameters with which the distance traveled in the detour path is calculated.

Odometers serve as sensors designed to monitor wheel rotations over time. Typically, these devices acquire data from wheel encoders, recording the total revolutions of each wheel. Errors in odometer readings can be categorized into systematic (certain) and unsystematic (uncertain) types. Systematic errors arise from intrinsic flaws in the physical structure of the odometer system and can be mitigated through precise calibration. Conversely, unsystematic errors persist even after calibration. Significant discrepancies in calculating travel distance errors directly impact the angular orientation accuracy of the robot. Over time, the cumulative effects of directional and orientation deviations surpass navigation measurement errors, playing a more substantial role in localization inaccuracies due to their non-linear contributions.

For this purpose, measurement noise should be minimized. The basic equation for updating the odometric position for the mobile robot is according to (8). Assume that the position of the moving robot at time $k-1$ is as follows:

$$X_{k-1} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} \tag{9}$$

At moment k, the robot moves to the new position X_k. Using (10), the new position $\hat{X}_k$ at time k can be predicted from the previous position X_{k-1}:

$$\hat{X}_k = f(X_{k-1}, u_k) = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta_{s_r} + \Delta_{s_l}}{2} cos\left(\theta_{k-1} + \frac{\Delta_{s_r} - \Delta_{s_l}}{2L}\right) \\ \frac{\Delta_{s_r} + \Delta_{s_l}}{2} sin\left(\theta_{k-1} + \frac{\Delta_{s_r} - \Delta_{s_l}}{2L}\right) \\ \frac{\Delta_{s_r} - \Delta_{s_l}}{L} \end{bmatrix} \tag{10}$$

In equation (10), u_k is a variable representing a control input. Also, y_{k-1} and $x_{k-1}$are the position of the robot at time $k-1$ and θ_{k-1} is the direction of the robot at time $k-1$. The control input u_k is according to (11):

$$u_k = \begin{bmatrix} \Delta_{s_l} \\ \Delta_{s_r} \end{bmatrix} \tag{11}$$

B. Localization using fuzzy adaptive unscented Kalman filter and RRT algorithm*

In the standard implementation of the unscented Kalman filter (UKF), the noise covariance matrices are typically assumed to remain constant throughout the estimation process. However, achieving optimal performance of the UKF algorithm requires an accurate specification of the process and measurement noise covariance matrices. While existing research highlights the effectiveness of the standard UKF under Gaussian noise distributions and well-defined initial conditions, its performance significantly declines in the presence of non-Gaussian noise or uncertain conditions. In practical applications involving nonlinear dynamic systems, noise is often characterized by uncertainties and time-varying disturbances. Consequently, the real-time adjustment of noise covariances becomes crucial for enhancing UKF performance. An adaptive strategy that dynamically updates the noise covariances online offers a robust solution to these challenges, improving estimation accuracy and overall algorithm efficiency.

The techniques used for covariance adjustment can be classified into two main frameworks [16]. The first framework includes methods based on mathematical principles such as Bayesian inference, covariance matching, initial correlation techniques, and maximum likelihood estimation. On the other hand, the second framework includes techniques based on global optimization strategies. At present, AI-based approaches are increasingly popular in the optimization of these processes in various fields, and fuzzy logic stands out as an important component in the field of artificial intelligence. The dynamics of the robot and its measurement equation are presented according to Equations (12) and (13), respectively:

$$X_k = f(X_{k-1}, u_{k-1}) + W_k \tag{12}$$

$$Z_k = H(X_k) + V_k \tag{13}$$

where z_k is the measurement output vector, u_{k-1} is the input vector and X_k is the state vector. The index k represents the time step. The process noise is represented by W_k, characterized by its mean value and covariance matrix Q_k. The function H represents the nonlinear measurement model of the system. The measurement noise is denoted as V_k, which has a mean of zero and a covariance matrix R_k. For a lidar sensor, the measurement vector consists of two parameters: r, which is the distance from the robot to the lidar, and α, which is the angle between the robot and the lidar. The lidar sensor reading is accompanied by measurement noise v_k with covariance R_k. Sensor observations are denoted by Z:

$$Z_k = \begin{bmatrix} r_k \\ \alpha_k \end{bmatrix} \tag{14}$$

The nonlinear functions of the system and measurement are according to (15) and (16), respectively:

$$f(\hat{X}_{k-1}, u_{k-1}) = \begin{bmatrix} x_{k-1} + \Delta s\ cos(\Delta\theta) \\ y_{k-1} + \Delta s\ sin(\Delta\theta) \\ \theta_{k-1} + \Delta\theta \end{bmatrix} \tag{15}$$

$$H\left(\hat{X}_{k-1}, S_i\right) = \begin{bmatrix} \sqrt{\left(S_{i,x} - x\right)^2 + \left(S_{i,y} - y\right)^2} \\ atan2\left(S_{i,x} - x, S_{i,y} - y\right) - \theta \end{bmatrix} \quad (16)$$

In relation (16), S_i is a vector containing the coordinates of the turning points and their special identifiers, which are defined in advance:

$$S_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (17)$$

A key limitation of the odometry method is the accumulation of errors and the reliance on an oversimplified motion model. Odometry estimates the distance traveled by measuring wheel rotations; however, errors inevitably accumulate over time, leading to inaccuracies in the robot's position and orientation estimates. Furthermore, odometry assumes an idealized motion model in which the robot moves either in a straight line or rotates with perfect precision. In practice, nonlinear environmental factors and irregularities in motion introduce deviations from this idealized model, exacerbating errors in odometry-based calculations. For this reason, a sensor fusion approach should be used regarding this problem. The unscented Kalman filter or UKF is a derivative of the EKF. The UKF employs a deterministic sampling approach to represent the state distribution using a minimal set of carefully chosen sample points, known as sigma points. These sigma points are selected with high precision to capture the mean and covariance of the state distribution effectively. Similar to EKF, the UKF operates through two fundamental steps: model prediction and data integration [17]. Unlike the EKF, the UKF does not require the computation of the Jacobian matrix, making it particularly advantageous for systems with highly nonlinear dynamics [18]. The UKF is selecting a certain amount of points from previous milestones [19]. The robot motion mode model is presented according to the equations (12) and (13). The nonlinear functions of the system and measurement are according to (15) and (16), respectively. By receiving the error covariance matrix of the previous moment P_{k-1}, the state vector $\hat{X}_{k-1}$ and sigma points are formed as follows:

$$\begin{aligned} X_{i,k-1} &= \hat{X}_{k-1} && i = 0 \\ X_{i,k-1} &= \hat{X}_{k-1} + \left(a\sqrt{nP_{k-1}}\right)_i && i = 1, \dots, n \\ X_{i,k-1} &= \hat{X}_{k-1} - \left(a\sqrt{nP_{k-1}}\right)_i && i = L+1, \dots, 2n \end{aligned} \quad (18)$$

where a is a scalar and positive and determines the expansion of sigma points around $\hat{X}_{k-1}$. The i column of the square root of the matrix P is represented by $\left(\sqrt{P}\right)i$. The new sigma points work through the unscented conversion principle and the transfer function f of the previous sigma points:

$$X_{i,k} = f\left(X_{i,k-1}, u_{k-1}\right) \quad (19)$$

The predicted mean according is as follows:

$$\hat{X}_k = \sum_{i=0}^{2n} w_i , X_{i,k} \quad (20)$$

and the covariance of the estimation error according to (21):

$$P_k = \sum_{i=0}^{2n} w_i \left(X_{i,k} - \hat{X}_k\right)\left(X_{i,k} - \hat{X}_k\right)^T + Q_k \quad (21)$$

where in (21), $\hat{X}_k$ is the predicted value of a state parameter, P_k is the mean squared error of $\hat{X}_k$ and w_i is the weight of sigma points, also $X_{i,k}$ is the updated sampling point. The equations related to the calculation of weights are shown in (22):

$$\begin{aligned} w_i &= 1 - \frac{1}{a^2} && i = 0 \\ w_i &= \frac{1}{2na^2} && i = 1, \dots, 2n \end{aligned} \quad (22)$$

The formulation for sigma measurements is expressed in Equation (23):

$$Z_k = H\left(X_{i,k}, u_k\right) \quad (23)$$

The weighted average of the predicted measurements is as follows (24):

$$\overline{Z}_k = \sum_{i=0}^{2n} w_i Z_k \quad (24)$$

UKF gain is calculated according to the following relations:

$$P_{Z_k Z_k} = \sum_{i=0}^{2n} w_i \left(Z_k - \overline{Z}_k\right)\left(Z_k - \overline{Z}_k\right)^T + R_k \quad (25)$$

$$P_{X_k Z_k} = \sum_{i=0}^{2n} w_i \left(X_{i,k} - \hat{X}_k\right)\left(Z_k - \overline{Z}_k\right)^T \quad (26)$$

$$K_k = P_{X_k Z_k} {P_{Z_k Z_k}}^{-1} \quad (27)$$

UKF has updated the measurement as follows in the update phase:

$$\hat{X}_k = \hat{X}_k + K_k\left(Z_k - \overline{Z}_k\right) \quad (28)$$

$$P_k = P_k - K_k P_{Z_k Z_k} {K_k}^T \quad (29)$$

where $P_{Z_k Z_k}$ is the covariance parameter of the predicted measurement, $P_{X_k Z_k}$ is the covariance parameter between the measurement and the state, K_k is the Kalman gain, P_k is the covariance parameter and $\hat{X}_k$ is the state evaluation [20].

B.1. RRT path planning algorithm*

The Rapidly-exploring Random Tree Star (RRT*) algorithm is a sampling-based path planning approach designed to identify efficient and feasible paths for robotic systems navigating complex environments [21]. RRT* incrementally constructs a tree structure, starting from an initial configuration and progressing towards a target configuration, effectively exploring the state space to determine optimal paths [22]. A notable feature of RRT* is its capability to dynamically refine the tree structure by reorganizing nodes, thereby enhancing path quality and ensuring convergence [23]. This adaptive framework makes RRT* particularly suitable for addressing challenges in high-dimensional spaces and systems with non-holonomic

constraints [24–25].

The configuration of the unknown region is denoted as X, where X is a set with members $X = \{x, y, \theta\}$. Here x and y represent the two-dimensional coordinates, while θ represents the direction. In addition, X_{obs} includes all regions with obstacles, while X_{free} refers to the parts of the environment that are free of obstacles. The equation (30) expresses the mathematical connection between these two sets:

$$X_{free} = \frac{X}{X_{obs}} \tag{30}$$

In the following, more important concepts are presented for consideration. These concepts are: x_{init} and x_{goal}, which mean the starting point and the goal point, respectively [25]. The solution of the motion planning problem is $\sigma: [0,1] \rightarrow X_{free}$ where $\sigma(0) = x_{init}$ and $\sigma(1) = X_{goal}$. In RRT*, a tree pattern $T = (V, E)$ is also constructed, which consists of the set of vertices $V \subset X$ and branches $E \subset V \times V$. The sample randomly samples a state $x_{rand} \in X_{free}$ from the unobstructed space. Distance with two states $x, x' \in X$, the function $dist(x, x')$ returns the cost of connecting path x and x'. The cost used in this article is in terms of Euclidean distance. $Nearest$ and $Near$ are defined by a set $V \subset X$ and state $x \in X$. These two procedures return the nearest node in the tree and all the nearest nodes that are centered inside a sphere of volume $\gamma\,((log\ n)/n)^{1/d}$. About x in terms of distance γ is a constant as described in [25], d is the dimension of X space and n is the number of vertices. Steer function $Steer(x, x')$ returns a path $\sigma: [0,1] \rightarrow X$ connecting x and x'. Checking for a collision with a path $\sigma: [0,1] \rightarrow X$, the function $InCollision(\sigma)$ checks whether the path is in X_{free}, and if not, returns the first failed state $xfail$. The cost function with a vertex x, let $C(x)$ be the cost of the entire unique path that starts from the root vertex x_{init}, goes along the nodes of the tree and reaches the state x, while $c(\sigma)$ is the cost function of the path is $\sigma: [0,1] \rightarrow X$. Sorting a list L, an ordered set of elements of the form (c_i, x_i, σ_i), the $L.sort()$ function sorts all elements of L based on c_i in ascending order. In configuration, RRT* builds the search tree based on nodes by randomly collecting samples incrementally, so they are constructed in a way that rapidly reduces the expected distance of randomly selected nodes in the tree. As the nodes grow, using x_{rand} (random nodes), it is generated to connect x_{near} node and x_{rand} node [25]. It is shown in Fig.5 that by repeating this process, the nodes form a tree together to reach the target state.

The pseudo code of the RRT* algorithm is in the form of Fig.6. The RRT* algorithm is sampling-based and decentralized, reducing the process waste of dealing with too

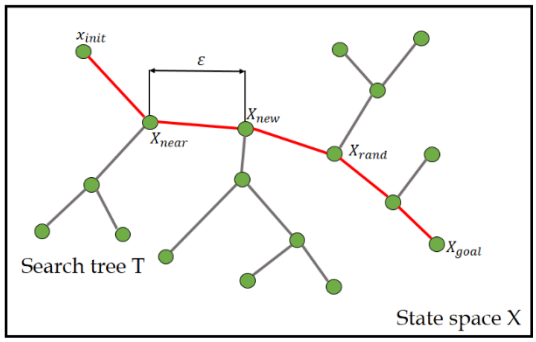


Fig.5. RRT* path planning development.

many data sets for the processor. Additionally, during exploration, RRT* finds an optimal path to avoid obstacles. This means that the robot can navigate in an unknown area using obstacle avoidance and routing by generating probabilistic models of the environment. RRT* is close to reality considering the robot model, without assuming that the robot has the ability to move in all directions. Based on kinematic models, it is known that any number of positions can be achieved by the robot after a certain period of time, and accordingly, the number of new positions is created based on the previous position.

Algorithm 1: The RRT* Algorithm

$V \leftarrow \{X_{init}\}; E \leftarrow \emptyset; T \leftarrow (V, E);$
for $i = 1 \rightarrow N$ **do**
$X_{rand} \leftarrow Sample(i);$
$X_{near} \leftarrow Near(V, X_{rand});$
if $X_{near} = \emptyset$ **then**
$X_{near} \leftarrow Nearst\ (V, X_{rand});$
end if
$X_{parent} \leftarrow Find\ Best\ Parents\ (X_{near}, X_{rand});$
if $X_{parent} \neq$ Null **then**
$V \leftarrow V \cup \{X_{rand}\}; E \leftarrow E \cup \{(X_{near}, X_{rand})\};$
$E \leftarrow Rewire(E, X_{near}, X_{rand});$
end if
end for
return T = (V, E);

Fig.6. RRT* Pseudo Code.

This creates a large number of possible values over time. Given any expected X_{rand} position, the nearest X_{near} position can always be found in a large number of values. Then, a set of corresponding trajectories is obtained by inverting the status points. The method used here is similar to the random sampling strategy, and then the best control input is selected from the sample results. It is also very simple to deal with obstacle related problems: when X_{rand} is hitting an obstacle, it is directly ignored, and if the path intersects the obstacle, another X_{near} is directly selected. This leads to a series of different paths to avoid collisions

with obstacles. Then, based on this, a route plan is generated and the results can be drawn and checked.

B.2. matching AUKF with fuzzy system

The proposed methodology involves combining a fuzzy-based adaptive unscented Kalman filter with a random tree routing algorithm to enhance the precision and efficiency of data processing within indoor environments. This integration leverages the complementary strengths of both techniques, effectively addressing the challenges associated with measurement noise and the uncertainties inherent in external sensing sensors. The core of the proposed approach is a fuzzy-based adaptive unscented Kalman filter, designed to estimate the position of a mobile robot in the presence of uncertainties. By incorporating fuzzy logic, the filter can effectively handle uncertainties that traditional Kalman filters may have difficulty dealing with. This approach enables real-time adjustment of the measurement noise covariance based on the evolving characteristics of the data, ensuring robust performance in dynamic scenarios. The moving robot uses RRT* routing algorithm. This algorithm uses an optimal structure for dynamic routing in the map and minimizes delay. Fig.7, which shows the different steps of the approach, provides an overview of the method.

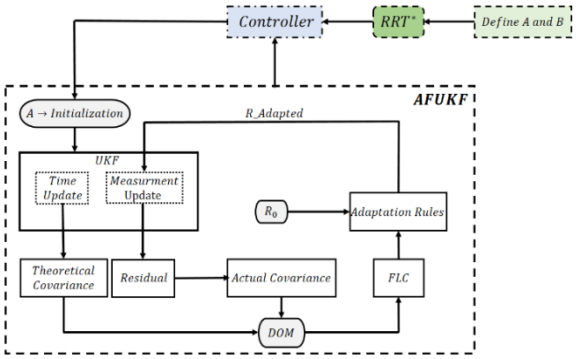


Fig.7. Schematic representation of the FAUKF approach.

In Fig.7, the starting and target points are defined as A and B, respectively. Then, the RRT* algorithm receives these points and designs an optimal path for robot operation according to the map environment. The designed path contains the coordinates transferred to the controller, which follow these points. The task of the UKF block is to estimate the position of the robot. The UKF receives point A as the initialization, and thus the estimation starts online. In the time-update phase, the covariance is computed theoretically to predict the state of the system and uncertainty. Subsequently, in the second phase, known as the Measurement Update, measurement data are incorporated into the filter, contributing to the calculation of the residual value. These values are transferred to the fuzzy block after mathematical operations as DOM, i.e. deviation angle. The fuzzy rules, when applied and considering the initial measurement covariance, facilitate the computation of adaptive measurement noise. This process provides an overview of the FAUKF system, with each component detailed in the subsequent sections. In contrast to this adaptive approach, the standard Unscented Kalman Filter typically assumes constant noise covariances across successive iterations.

The precise definition of measurement and process noise covariance matrices is critical for the optimal functioning of the UKF algorithm. However, the algorithm's performance may degrade when faced with non-Gaussian noise distributions and uncertainties. In practical applications, particularly those involving nonlinear dynamical systems, time-varying uncertainties and disturbances are common. Consequently, online adaptation of the measurement noise covariance is essential to enhance the UKF's performance. Employing an adaptive strategy that dynamically adjusts the noise covariance parameters can effectively address these challenges, thereby improving estimation accuracy in complex and uncertain environments. Strategies for adapting noise covariance can be categorized into two primary frameworks. The first framework involves mathematical approaches, including Bayesian techniques, covariance matching, principal correlation analysis, and maximum likelihood estimation. The second framework relies on optimization methodologies. Recently, artificial intelligence (AI)-based techniques, particularly those utilizing fuzzy logic, have gained prominence in process optimization across various domains. Fuzzy logic, a key component of AI, has garnered increasing attention in the scientific community due to its unique ability to handle the uncertainties and ambiguities inherent in complex systems. Its strength lies in providing an approximate representation of such systems, making it a powerful tool for managing dynamic complexities. Notably, the integration of fuzzy logic with Kalman filter-based estimation methods has become a significant area of research, offering promising solutions in diverse application fields.

The standard unscented Kalman filter algorithm faces challenges when faced with uncertain or unknown noise distributions, leading to ambiguity and inaccuracy that can compromise estimation accuracy. To address this issue, researchers have turned to fuzzy set theory as a promising solution. Fuzzy set theory, which is famous for its efficiency in managing uncertainties, serves as a powerful mathematical tool to reduce uncertainty in system modeling and estimation processes. It is worth noting that the structure of fuzzy logic reflects the complex functioning of the human brain, which operates through linguistic variables to guide complex decision-making scenarios with elegance and adaptability. The main elements of a fuzzy logic controller (FLC) include a structured sequence of processes aimed at effectively managing complex system dynamics. The initial phase, known as fuzzification, involves converting input and

output data into precise linguistic terms, laying the groundwork for subsequent operations. After fuzzification, two critical components—the database and rule base—come into play, whose membership functions are carefully determined by domain experts or operators. This step is seamlessly integrated with the inference engine and collectively forms the FIS. Fuzzy inference, akin to approximate reasoning, facilitates the evaluation of linguistic descriptions and finally produces results as processed parameter values. Subsequently, in the defuzzification phase, the outputs of the comprehensive system are transformed into measurable numerical values through the hub method, ensuring a seamless transition from linguistic representations to practical, quantitative data for informed decision-making processes.Regarding advanced estimation techniques, the integration of fuzzy logic principles into the FAUKF framework provides a robust approach to enhancing tracking performance. In this method, a Mamdani fuzzy inference system dynamically adjusts the measurement noise covariance, as presented in Equation (34). At the core of this process is the degree of match (DOM) parameter, defined as the ratio between the actual residual covariance C_A and the theoretical covariance C_T. This index serves as a key indicator of deviation or relative differences between covariances and guides the adjustment mechanism with accuracy and efficiency.

The FIS acts as a complex decision engine and generates a scaling parameter that fine-tunes the noise covariance in the measurement space and optimizes estimation accuracy.

The adaptation law governing the iterative adjustment of R_T, described in Equation (35), is carefully established through a methodical trial-and-error approach and employs fuzzy logic principles to refine estimation tracking performance. This innovative integration of fuzzy logic and adaptive strategies represents a paradigm shift in estimation methodology, demonstrating the transformative potential of fuzzy-based approaches to optimize noise covariance control mechanisms and enhance system performance. The difference between predicted and actual measurements is determined using Equation (31).

$$e_k = Z_k - \overline{Z}_k \tag{31}$$

$$C_T = P_{Z_k Z_k} = \sum_{i=0}^{2n} w_i \left(Z_k - \overline{Z}_k\right)\left(Z_k - \overline{Z}_k\right)^T + R_k \tag{32}$$

$$C_A = \frac{1}{W} \sum_{k-W+1}^{k} e_k e_k^T \tag{33}$$

$$DOM = \frac{tr(C_A)}{tr(C_T)} \tag{34}$$

$$\Delta R_k = \alpha^3 \times R_{k-1} \tag{35}$$

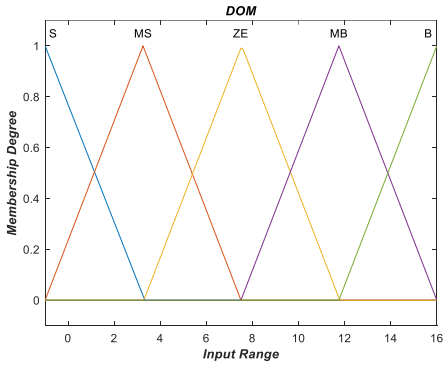
$$\Delta R_{k+1} = R_k + \Delta R_k \tag{36}$$

Here, only R_k is adapted for two reasons: first, it is assumed that the noise associated with the measurement information significantly affects the filter in any form. The second reason is to reduce the computational load of the filter. The outline of the proposed algorithm is shown in Figure 8. This algorithm is executed iteratively until the end of the simulation time. The proposed FAUKF pseudocode, illustrating the different steps for better understanding, is also presented in Fig. 8. Here, kkk represents the number of Monte Carlo runs, and TTT denotes the total simulation time.

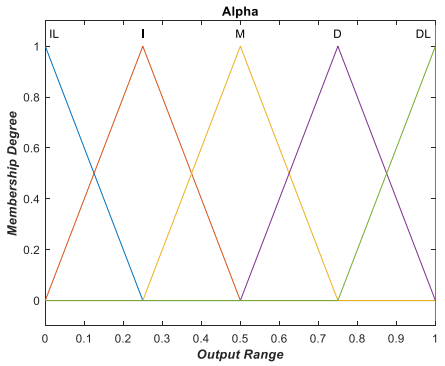
Algorithm 2 : Pseudede of the Proposed FAUKF

Require: Initialization of x_0, P_0, Q, R, UKF and system parameters.
Ensure: Demention of f_k, h_k, Z_k and others.
for i = 1 : Time Vector **do**
for k = 1 : T **do**
Find residual by using(31).
Caculate C_T and C_A from (32) and (33).
Find DOM to FIS from (34).
Caculate tuning parameter (α) from FIS.
Update (ΔR_k) as given in (35) ($\Delta R_k \rightarrow \Delta R_{k+1}$).
Stored states and other required parameters.
end for
Stored parameters for each Monte Carlo run.
end for

Fig.8. Algorithm.2. FAUKF a Pseudo Code.



(a)



(b)

Fig.9. Membership functions: (a) input membership function (b) output membership function.

Five sets of triangular membership functions are considered for input (DOM) and output (α) parameters.

Membership functions for input and output parameters are shown in Fig.9 MS (small medium), ZE (zero), MB (large medium), B (large) and IL (large increase), I (increase), M (maintain), R (decrease), RL (large decrease) respectively . To avoid the computational complexity of the problem, a fuzzification strategy based on the conventional method is used. Fuzzy rules (IF-THEN) for input and output parameters are as follows:

1) If DOM was in part S, then α is equal to RL.
2) If DOM was in the MS part, then α is equal to R.
3) If DOM was in part ZE, then α is equal to M.
4) If DOM was in the MB part, then α is equal to I.
5) If DOM was in part B, then α is equal to IL.

A fuzzification method is needed to convert the fuzzy output into a crisp value. Here the center of area or center of gravity method is implemented for this purpose. It is very common and widely used. Mathematically, it is expressed as follows:

$$\alpha = \frac{\sum_{j=1}^{m} \alpha_j \mu_{a,j}}{\sum_{j=1}^{m} \mu_{a,j}} \quad (37)$$

where α_jth clear value, m is the number of rules and $\mu_{a,j}$. j represents the membership function of j according to the defined fuzzy rule. Furthermore, for a more detailed theoretical background and mathematical formulation of fuzzy logic, the interested reader can refer to the following references: [26-29].

III. Test Results

A. Evaluation of Performance

The robot moves within the map environment, recording measurement data from its internal wheel movement sensors and collecting measurements related to environmental signs and obstacles using its external sensors. The corresponding simulation environment is implemented in MATLAB. To address the navigation accuracy problem, the FAUKF-based localization method and RRT*-based route planning method are introduced. The robot simulation environment is designed to resemble a hypothetical closed environment. Fig.10 shows the actual path of the robot and the map of the environment, with the robot's path marked in red. The robot's path is designed using the random tree algorithm with fast exploration. Localization is performed by identifying landmarks from the measured data and matching them with the landmarks on the map. The difference between the positions of the estimated and measured landmarks is used to calculate the robot's position and state. The robot is equipped with a lidar sensor, which is installed at the front and measures the distance and angle to the observed target. The speed of the robot is 1 m/s and the maximum steering angle is 1.59 degrees. The distance between the wheels is 0.05 and the maximum range of the laser range finder is 0.5 meters and its field of view is 90 degrees. The control frequency is 50 Hz and the standard deviation of the noise of direct speed and angular speed are as follows σ_v and σ_γ respectively:

$$\sigma_v = 0.01m/s \ \ , \sigma_\gamma = 0.017^\circ \quad (38)$$

$$\gamma = [0.1\ 0.001\ 0.001\ 0.001\ ] \quad (39)$$

The sampling frequency of the sensors is 5Hz and the standard deviation of the measurement noise in distance and direction is as (39):

$$\sigma_r = 0.5m/s \ \ , \sigma_\theta = 0.05^\circ \quad (40)$$

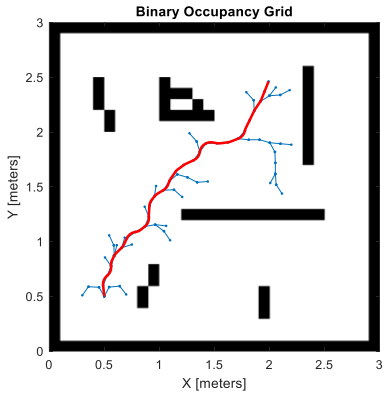


Fig.10. path designed by RRT* algorithm and environment map.

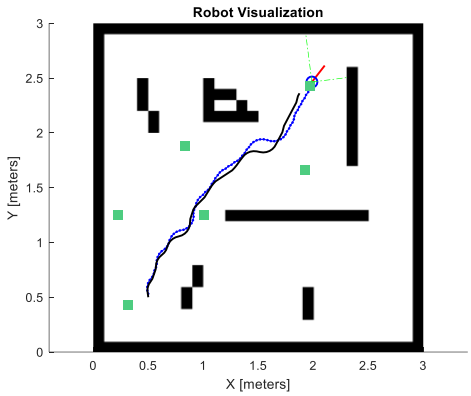


Fig.11. Real path and robot position estimation by odometry.

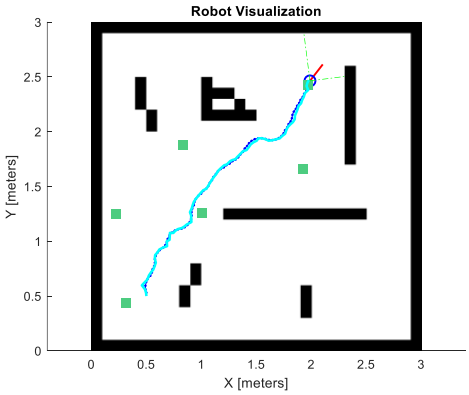


Fig.12. Real path and robot position estimation by FAUKF.

Fig.11 and Fig.12 show the results of positioning based on odometry and FAUKF. The signs on the map are marked with the symbol ■. In Fig.11, the black line corresponds to the localization of the robot by odometry. In Fig.12, the

turquoise color corresponds to FAUKF estimation. It is clear that the performance results of FAUKF method are better than odometry. In other words, in positioning with the proposed algorithm, the estimated path of the robot is as close as possible to the real path. The robot has successfully localized from the starting point to the target point and according to the route planning designed by the RRT* algorithm, it has reached the target point. In the combination of FAUKF and RRT* algorithm, it is clear that the path generated by the simulation of the proposed algorithm is optimal. Fig.12 shows the localization accuracy of FAUKF.

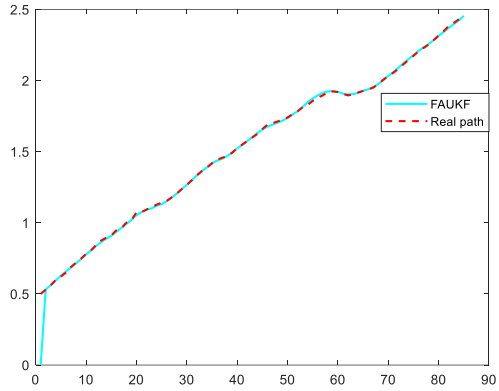


Fig.13. The accuracy of FAUKF related to the estimation of the real position of the moving robot.

In Fig.13, the localization error and the generated angle error are very small and the error is not additive. As a result, the proposed approach has increased the localization accuracy significantly compared to odometry. Fig.14 to Fig.16 show the localization error in the X and Y axes, as well as the orientation of the front angle of the robot relative to the path angle.

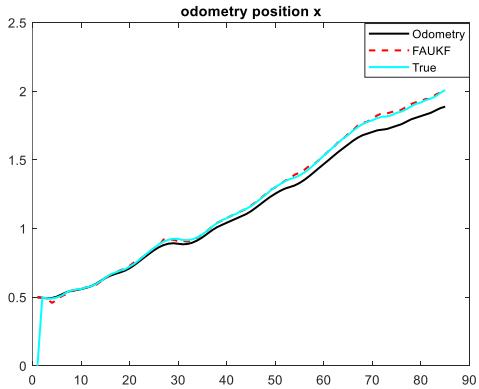


Fig.14. Accuracy of FAUKF compared to odomety in X axis.

In Fig.14, the localization error of FAUKF is compared with odometry in the X axis. The odometry only worked according to the real position of the robot in the initial moments, and the accumulation of noises caused the unsuccessful performance of the odometry. This is because the noise of the sensors and related noises are not considered. It is angular velocity and forward velocity. And as for localization with FAUKF, because the characteristics of noises are taken into account and the information of internal and external sensors are combined, this has made the robot no longer face the problem of accumulating errors. In Fig.15, FAUKF localization error and odometry are compared in Y axis. Localization errors, such as system error and sensor error, which are created using the odometry algorithm, are continuously accumulating. Therefore, the accuracy of odometry localization results is poor.

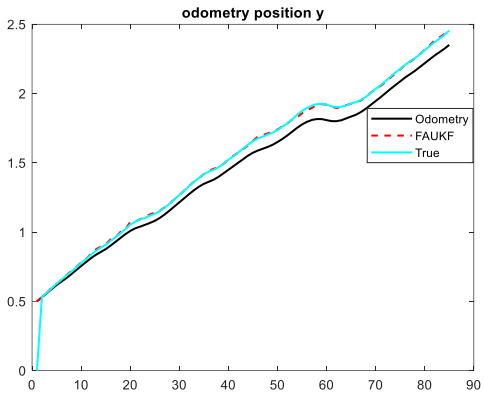


Fig.15. Accuracy of FAUKF compared to Y-axis odometry.

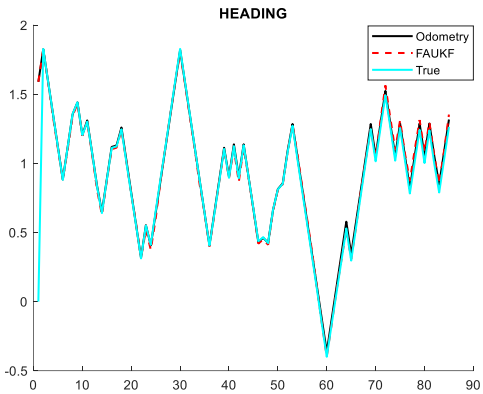


Fig.16. Accuracy of FAUKF compared to odometry related to the differential angle of the robot in orientation.

In Fig.16, the orientation of the front angle of the robot has followed the path angle almost correctly. The root mean square error (RMSE) is a measure commonly used to measure the difference between predicted values and actual values in regression analysis. RMSE shows the average deviation or error between the predicted and actual values of a model's predictions. By taking the root mean square of the errors between the predicted values and the actual values, it is calculated. A lower RMSE value indicates a better fit of the model. According to the explanation, the simulation results are given in Table (1).

A disadvantage of the proposed method is its longer processing time due to its complexity, whereas the other methods demonstrate shorter processing times because of

their simplicity. However, this trade-off is outweighed by the superior performance and accuracy of the proposed method,

TABLE 1:PERFORMANCE COMPARISON OF LOCALIZATION TECHNIQUES.

Average error in different directions and RMSE					
Technique	X	Y	z	RMSE	**Processing Time(sec)**
FAUKF+RRT*	0.001	0.01	0.012	0.046	**6.5 sec**
UKF + RRT*	0.002	0.019	0.020	0.047	**1.49 sec**
EKF + RRT*	0.027	0.025	0.095	0.055	**1.23 sec**
Odometry	0.037	0.06	0.032	0.077	**1.36 sec**

which is essential for achieving reliable results in our application. Table 2 shows the advantages and disadvantages of the proposed method compared to other methods. Fig.17 to Fig.20 show the navigation simulation results. In order to make the simulation results more objective, the navigation simulation of the mobile robot is carried out in four different groups of selected paths. The mobile robot moves from the starting point to the target point. The red paths show the results of the RRT* algorithm. The black lines are results based on odometric calculations. The cyan paths are the position estimation by FAUKF algorithm and the green dots are the map markers.

TABLE 2: COMPARISON OF THE PROPOSED METHOD COMPARE WITH OTHERS METHOD

Method	Advantage	Disadvantage
Proposed Method	High accuracy	Longer processing time
	Robust performance	
Other Method	Faster processing time	Low accuracy
	Simplicity	Limited robustness

The FAUKF approach shows the lowest value of RMSE (root mean square error) compared to other methods. UKF also performs well, with a root mean square error of 0.047, which is slightly higher than the FAUKF approach. Also, EKF has a root mean square error of 0.055, which is higher than FAUKF and UKF approaches. And finally, the odometric approach that relies solely on the robot's internal sensors has the highest error squared, i.e. 0.077, which indicates a greater deviation from the actual path compared to other methods. Table 1 shows that the FAUKF approach, when integrated with the RRT* route planning method, provides the most accurate route estimation among the techniques evaluated. The low RMSE values for the FAUKF and UKF approaches indicate that these methods can better handle the time-varying noise characteristics of the system compared to the oedometric and EKF methods. The RMSE difference between the FAUKF and UKF approaches, although small (0.001), shows that the FAUKF approach has been able to perform better. Therefore, according to these results, it can be said that the FAUKF approach has performed promisingly and has made a significant improvement over the conventional UKF. This shows that combining fuzzy logic with UKF has been able to increase the accuracy of the system. The Table 1 shows that the FAUKF approach has the longest run time of 6.5 seconds. This delay in the FAUKF approach's run time could be a significant weakness, as it may not be suitable for time-sensitive robotic localization and navigation tasks.

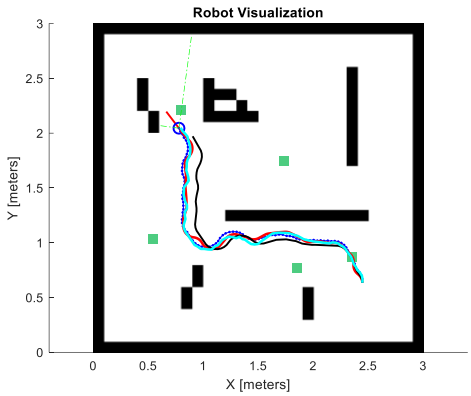


Fig.17. The first path estimated by the FAUKF.

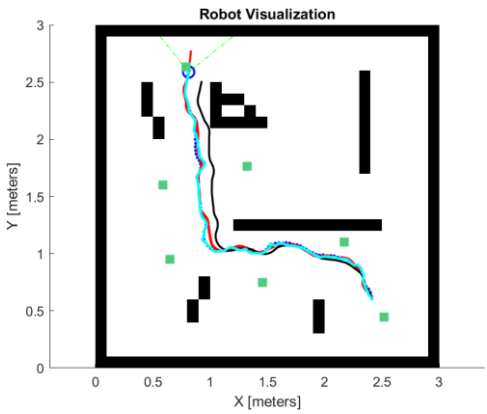


Fig.18. The second path estimated by the FAUKF.

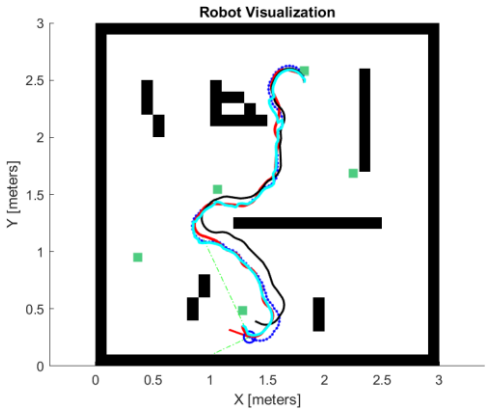


Fig.19. The third path estimated by the FAUKF.

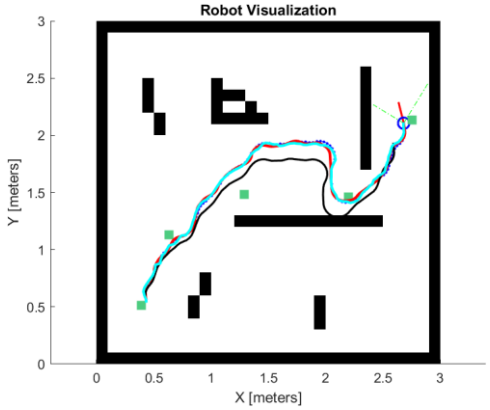


Fig.20. The Fourth path estimated by FAUKF.

The findings clearly demonstrate the effectiveness of the RRT* routing algorithm in constructing optimal paths. The results indicate that the algorithm not only ensures route optimization but also prioritizes safety requirements and adheres to the constraints defined by the operational environment. Simulations presented in the conclusion section further affirm that the RRT* algorithm successfully designs safe and efficient paths, transmitting the path information as control inputs to the robot. Additionally, it is evident that the FAUKF provides accurate estimations of the robot's actual position across various trajectories, even when odometry data deviates significantly from the true position. This highlights the high precision and adaptability of the mobile robot, showcasing the method's overall efficiency and stability. These results suggest the potential for broader application of this approach in dynamic and uncertain environments.

B. Evaluation of Stability

The stability of FAUKF is achieved through the adaptive tuning of the measurement noise covariance, which is controlled by a fuzzy logic system. This tuning mechanism helps the filter remain stable by responding dynamically to variations in measurement noise, particularly in nonlinear or fluctuating environments. Additionally, the Unscented Kalman Filter framework itself contributes to stability by accurately propagating sigma points through nonlinear functions without requiring linearization, which reduces approximation errors and enhances convergence stability.

In terms of robustness, the FAUKF demonstrates resilience against modeling inaccuracies and unexpected disturbances. The adaptive fuzzy component plays a critical role in adjusting the filter to real-time conditions by refining noise covariance values, which helps maintain accurate state estimation even when system dynamics deviate from expected models. This approach ensures that FAUKF can handle diverse operational conditions and provides consistent performance across a variety of nonlinear and unpredictable scenarios.

Checking the stability of a filter through consistency analysis is crucial for evaluating its performance, particularly in terms of reliable and accurate state estimation. To assess the filter's consistency, the estimated values are compared with the probability density function (PDF) of an ideal filter. Consistency is verified by the normalized estimation error squared (NEES):

$$NEES = (x - \hat{x})^T P^{-1} (x - \hat{x})$$

where x is the ground truth, $\hat{x}$ is the estimated mean, and P is the estimated covariance. Consistency is further evaluated by performing multiple Monte Carlo runs and calculating the average NEES (ANEES). Given N runs, the ANEES is computed as:

$$ANEES = \frac{1}{N} \sum_{i=1}^{N} NEES$$

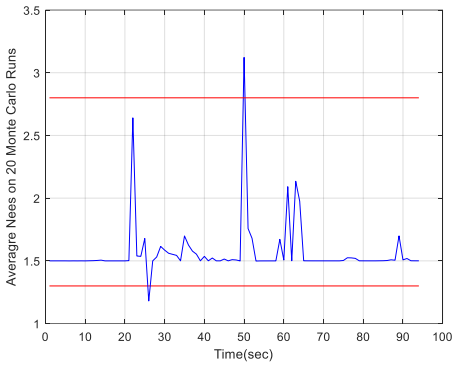


Fig.21. Consistency of Proposed Method.

The NEES value indicates whether the estimated covariance P accurately reflects the error in state estimates. For a stable and consistent filter, NEES values should be statistically bounded within certain confidence limits based on the degrees of freedom in the state vector. If NEES consistently falls outside these bounds, it may indicate that the filter's assumptions or adjustments are inadequate or not well-tuned. For the 2-dimensional vehicle position with 20 Monte Carlo simulations the two-sided 95% probability concentration region for $ANEES$ is bounded by interval [1.3, 2.79] and algorithm is consistent if $ANEES$ with probability 95% belong to [1.3, 2.79]. Fig.21 shows that the proposed method is consistent.

IV. Conclusions

In this study, the performance of an adaptive unscented Kalman filter has been investigated. This method involves matching the covariance of the measurement noise using an adaptation rule to handle uncertainty in the noise, and utilizing the output of the Mamdani fuzzy inference system as the adjustment factor for the adaptation rules, with triangular membership functions. The comparative analysis of this approach showed that the FAUKF method outperforms odometry, EKF, and UKF methods. Notably, the proposed FAUKF method demonstrated a square root

error of 0.046, which is lower than the values of 0.077 for odometry, 0.055 for EKF, and 0.047 for UKF. This indicates a significant improvement in accuracy, highlighting the superiority of the FAUKF method in state estimation accuracy for robot localization tasks. To further validate the effectiveness of the proposed method, experiments using the Random Tree Routing Algorithm with Rapid Exploration (RRT)* were conducted across four different paths.

The FAUKF technique exhibited the longest runtime of 6.5 seconds, substantially exceeding the execution times of the other evaluated methods. To mitigate this weakness, the researchers should consider optimizing the FAUKF algorithm or investigating alternative approaches that can achieve comparable accuracy without the considerable runtime overhead. The results emphasize the robustness and versatility of the FAUKF method, confirming its potential to revolutionize the accuracy of state estimation in indoor environments. The integration of the FAUKF method with the RRT* algorithm presents a promising approach to enhance state estimation performance and optimize robot localization tasks in dynamic indoor settings. Enhancing the computational efficiency of the FAUKF technique would be a crucial step to improve its practicality and applicability for real-time robotic tasks in dynamic environments.

REFERENCES

[1] Panigrahi, P.K. and S.K. Bisoy, Localization strategies for autonomous mobile robots: A review. Journal of King Saud University-Computer and Information Sciences, 2022. 34(8): p. 6019-6039.

[2] Y. Liu, S. Wang, Y. Xie, T. Xiong, and M. Wu, "A Review of Sensing Technologies for Indoor Autonomous Mobile Robots," Sensors, vol. 24, no. 4, p. 1222, Feb. 2024.

[3] Siegwart, R., I.R. Nourbakhsh, and D. Scaramuzza, Introduction to autonomous mobile robots. 2011: MIT press.

[4] Zheng, L., et al., Heading estimation for multimode pedestrian dead reckoning. IEEE Sensors Journal, 2020. 20(15): p. 8731-8739.

[5] Madray, I., et al., Relative angle correction for distance estimation using K-nearest neighbors. IEEE Sensors Journal, 2020. 20(14): p. 8155-8163.

[6] Guo, S., et al., An improved PDR/UWB integrated system for indoor navigation applications. IEEE Sensors Journal, 2020. 20(14): p. 8046-8061.

[7] Wang, X.-l., L.-q. Li, and W.-x. Xie, A novel TS fuzzy particle filtering algorithm based on fuzzy C-regression clustering. International Journal of Approximate Reasoning, 2020. 117: p. 81-95.

[8] A. H. Hassaballa, A. M. Kamel, I. Arafa, and Y. Z. Elhalwagy, "Adaptive Precise Attitude Estimation Using Unscented Kalman Filter in High Dynamics Environments," Unmanned Systems, vol. 12, no. 04, pp. 653–665, 2024.

[9] R. D. Puriyanto and A. K. Mustofa, "Design and Implementation of Fuzzy Logic for Obstacle Avoidance in Differential Drive Mobile Robot," Journal of Robotics and Control (JRC), vol. 5, no. 1, 2024.

[10] Shaheen, O., et al., Stable adaptive probabilistic Takagi–Sugeno–Kang fuzzy controller for dynamic systems with uncertainties. ISA transactions, 2020. 98: p. 271-283.

[11] Liu, Y. and D. Cui, Vehicle state estimation based on adaptive fading unscented kalman filter. Mathematical Problems in Engineering, 2022. 2022(1): p. 7355110.

[12] S. H. Hashemi and N. Pariza, "Adaptive Transformed Unscented Simplex Cubature Kalman Filter for Target Tracking," Automatika, vol. 62, no. 3, pp. 3279–3287, 2021.

[13] Al-sudany, H.N. and B. Lantos, Comparison of Adaptive Fuzzy EKF and Adaptive Fuzzy UKF for State Estimation of UAVs Using Sensor Fusion. Periodica Polytechnica Electrical Engineering and Computer Science, 2022. 66(3): p. 215-266.

[14] Sun, Z., et al., Multi-Risk-RRT: An Efficient Motion Planning Algorithm for Robotic Autonomous Luggage Trolley Collection at Airports. IEEE Transactions on Intelligent Vehicles, 2024.

[15] Ding, J., et al., An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. Journal of Computational Science, 2023. 67: p. 101937.

[16] Szabat, K., et al., A fuzzy unscented Kalman filter in the adaptive control system of a drive system with a flexible joint. Energies, 2020. 13(8): p. 2056.

[17] Fang, Y., et al., Adaptive Unscented Kalman Filter for Robot Navigation Problem (Adaptive Unscented Kalman Filter Using Incorporating Intuitionistic Fuzzy Logic for Concurrent Localization and Mapping). IEEE Access, 2022. 10: p. 101869-101879.

[18] V. E. Papageorgiou and G. Tsaklidis, "An improved epidemiological-unscented Kalman filter (hybrid SEIHCRDV-UKF) model for the prediction of COVID-19: Application on real-time data," *Chaos*, vol. 2022, Article 112914, 2022.

[19] Kumar, M. and S. Mondal, A Fuzzy-based Adaptive Unscented Kalman Filter for State Estimation of Three-dimensional Target Tracking. International Journal of Control, Automation and Systems, 2023. 21(11): p. 3804-3812.

[20] K. Feng, J. Wang, X. Wang, G. Wang, Q. Wang, and J. Han, "Adaptive state estimation and filtering for dynamic positioning ships under time-varying environmental disturbances," *Ocean Engineering*, vol. 2024, Article 117798, 2024.

[21] X. Cui, C. Wang, Y. Xiong, L. Mei, and S. Wu, "More Quickly-RRT*: Improved Quick Rapidly-exploring Random Tree Star algorithm based on optimized sampling point with better initial solution and convergence rate," Engineering Applications of Artificial Intelligence, vol. 108, p. 108246, 2024.

[22] Orthey, A., C. Chamzas, and L.E. Kavraki, Sampling-based motion planning: A comparative review. Annual Review of Control, Robotics, and Autonomous Systems, 2023. 7.

[23] Y. Lee, "Trajectory Manifold Optimization for Fast and Adaptive Kinodynamic Motion Planning," arXiv preprint arXiv:2410.12193, 2024.

[24] D. Debnath, F. Vanegas, J. Sandino, A. F. Hawary, and F. Gonzalez, “A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications,” *Remote Sens.*, vol. 16, no. 21, p. 4019, 2024.

[25] Szabat, K., et al., A fuzzy unscented Kalman filter in the adaptive control system of a drive system with a flexible joint. Energies, 2020. 13(8): p. 2056.

[26] Woo, R., E.-J. Yang, and D.-W. Seo, A fuzzy-innovation-based adaptive Kalman filter for enhanced vehicle positioning in dense urban environments. Sensors, 2019. 19(5): p. 1142.
[27] Asl, R.M., et al., Fuzzy-based parameter optimization of adaptive unscented Kalman filter: Methodology and experimental validation. IEEE Access, 2020. 8: p. 54887-54904.
[28] J. C. R. Alcantud, A. Z. Khameneh, G. Santos-García, et al., "A systematic literature review of soft set theory," *Neural Comput. Appl.*, vol. 36, pp. 8951–8975, Jun. 2024.
[29] Matía, F., et al., The fuzzy Kalman filter: Improving its implementation by reformulating uncertainty representation. Fuzzy Sets and Systems, 2021. 402: p. 78-104.
[30] R. Havangi and M. Moradi, "PSO Based EKF Wheel-rail Adhesion Estimation," *Research Articles*, University of Birjand, doi: 10.22111/ieco.2023.43360.1446, in press.

MohammadRasool Hajiali received the B.S. degree in Electronic Engineering from Imam Ali Kordkuy University, Kordkuy, Iran, in 2020. He then completed the M.S. degree in Control Engineering from the University of Birjand, Birjand, Iran, in 2024. His research interests include control engineering, parameter estimation, soft computing, industrial automation, robotics, robot navigation, and robot localization. He is currently working with the academic community to advance knowledge and innovation in these areas through his research and academic work. Additionally, he is exploring new developments and applications within his fields of interest with the goal of making meaningful contributions to the industry and research landscape.

Ramazan Havangi received his M.S. and Ph.D. degrees from the K.N. Toosi University of Technology, Tehran, Iran, in 2003 and 2012, respectively. He is currently an Associate Professor of control systems with the Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His main research interests are inertial navigation, integrated navigation, estimation and filtering, evolutionary filtering, simultaneous localization and mapping, fuzzy, neural network, and soft computing.Email: Havangi@Birjand.ac.ir