

# Kalman Filtering for Precise Indoor Position and Orientation Estimation Using IMU and Acoustics on Riemannian Manifolds

Mohammed H. AlSharif, Mohanad Ahmed, Mohamed Siala, Tareq Y. Al-Naffouri

**Abstract**—Indoor tracking and pose estimation, i.e., determining the position and orientation of a moving target, are increasingly important due to their numerous applications. While Inertial Navigation Systems (INS) provide high update rates, their positioning errors can accumulate rapidly over time. To mitigate this, it is common to integrate INS with complementary systems to correct drift and improve accuracy. This paper presents a novel approach that combines INS with an acoustic Riemannian-based localization system to enhance indoor positioning and orientation tracking. The proposed method employs both the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) for fusing data from the two systems. The Riemannian-based localization system delivers high-accuracy estimates of the target’s position and orientation, which are then used to correct the INS data. A new projection algorithm is introduced to map the EKF or UKF output onto the Riemannian manifold, further improving estimation accuracy. Our results show that the proposed methods significantly outperform benchmark algorithms in both position and orientation estimation. The effectiveness of the proposed methods was evaluated through extensive numerical simulations and testing using our in-house experimental setup. These evaluations confirm the superior performance of our approach in practical scenarios.

**Index Terms**—Riemannian optimization, tracking, localization, Extended Kalman filter, Inertial measurement unit, acoustic positioning, Unscented Kalman filter.

## I. INTRODUCTION

Accurate position and orientation estimation is crucial for various applications such as inventory management [1], smart homes, smart manufacturing, and health monitoring [2], [3]. High accuracy is fundamental to achieving the desired performance in these applications. Advances in sensor technology and computational techniques have led to the increased use of inertial sensors for navigation [4]. Historically, inertial sensors were mechanically isolated from the target’s rotational motion, which ensured high accuracy [4]. Modern inertial systems have simplified the mechanical complexity by rigidly attaching sensors to the target. These attached “strapped down” sensors reduced the cost and size but increased the computational complexity. In this paper, we focus on a strapdown inertial

Mohammed H. AlSharif is with the research and development department, Aramco Research Center, Thuwal 23955, Saudi Arabia (e-mail: mohammed.alsharif@kaust.edu.sa).

Mohanad Ahmed, and Tareq Y. Al-Naffouri are with the Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia.

Mohamed Siala is with the Department of Applied Mathematics, Signals and Communication, Higher School of Communication of Tunis (Sup’Com), Ariana 2083, Tunisia.

Preprint. Under review.

navigation system (INS) to estimate position, orientation, and velocity.

The INS operates based on kinematics and classical Newtonian mechanics, utilizing inertial measurement units (IMUs) comprised of accelerometers (motion sensors) and gyroscopes (rotation sensors) to measure acceleration and angular velocity. While the INS can use these parameters to estimate position, orientation and velocity at a high rate without external references, e.g. beacons, positioning and orientation errors accumulate over time [2]. To mitigate these errors, navigation systems typically integrate additional positioning methods. In our previous work [5], we developed an acoustic-based location and orientation estimation system using Riemannian optimization, which performs well under line-of-sight (LOS) conditions but suffers in non-line-of-sight (NLOS) scenarios due to the nature of the acoustic waves. Moreover, the system in [5] has a lower update rate compared to INS.

To address these limitations, we propose a Kalman filtering approach to integrate estimates from Riemannian localization methods [5] with those from the INS. This integration aims to enhance accuracy of the estimated position and orientation in NLOS conditions and over extended periods.

The main contributions of this paper, detailed in Section V, are as follows:

- A novel orientation correction method that models the orientation matrix as a state vector alongside position and velocity, resulting in new Extended Kalman Filter (EKF) formulations.
- A new projection algorithm to improve the accuracy of both position and orientation estimates obtained using EKF and Unscented Kalman Filter (UKF).
- Development and implementation of a new experimental setup to validate the proposed algorithms.
- An experimental evaluation of the proposed algorithms, complemented by numerical simulations.

This paper includes several symbols, which are defined in the nomenclature table for quick reference.

## II. RELATED WORK

With the increasing demand for precise indoor location estimation, indoor localization has gained significant attention. Therefore, various indoor positioning and orientation estimation systems have been developed to meet the need for highly accurate navigation solutions. Technologies used for indoor navigation include acoustic waves [6], radio-frequency [7],

infrared radiation [8], and laser signals [9]. While Bluetooth and WiFi-based systems are cost-effective, they suffer from low accuracy and require pre-calibration. In contrast, infrared and laser-based systems offer high accuracy but come with high hardware costs [10]. Acoustic and ultra-wideband (UWB) systems provide high accuracy at a lower cost [10].

While the choice of technology provides the foundation for indoor localization systems, the effectiveness of these systems ultimately hinges on the techniques employed to leverage these technologies. Positioning techniques have evolved significantly over the past decades, with the most popular being received signal strength (RSS) [11], angle of arrival (AoA) [12], time of flight (ToF) [10], and time difference of arrival (TDoA) [13]. RSS-based navigation is simple and low-cost but requires pre-calibration and suffers from low accuracy [14]. AoA-based systems offer high accuracy at close range but degrade significantly with distance due to angular estimation errors [15]. In contrast, ToF and TDoA-based techniques provide higher accuracy, especially over longer distances [16]. These techniques are categorized as infrastructure-based, requiring base stations and line-of-sight (LOS) for accurate measurements, which limits their effectiveness in non-LOS scenarios [17].

To overcome the LOS requirement, infrastructure-free techniques based on inertial sensors have been developed [17]. INS uses IMUs containing accelerometers and gyroscopes to estimate position and orientation [4]. While numerical integration of acceleration and angular velocity can provide accurate estimates over short periods, accuracy degrades over time due to error accumulation [17]. Pedestrian dead reckoning (PDR) [18], which estimates position based on step counts and step length, is another technique but is limited to specific movement patterns, making it less suitable for robots [19].

Infrastructure-free systems like INS offer high accuracy in the absence of LOS but suffer from accuracy degradation over time [17]. Conversely, infrastructure-based systems maintain high accuracy over long periods but require LOS [17]. Consequently, integrating INS with infrastructure-based systems is common practice to achieve high accuracy over extended periods, even without LOS.

One notable indoor localization system is the Cricket system [20], which estimates both position and orientation using TDoA and AoA methods with five receivers. However, it requires LOS and does not leverage the fixed geometry of the receiver array, which could enhance accuracy. Fusion of INS with infrastructure-based systems can improve navigation accuracy. For instance, Fan *et al.* [21] developed an improved adaptive Kalman filter (IAKF) to fuse TDoA-based localization with INS, achieving accurate position estimates in 2D. Similarly, Zhong *et al.* [22] used EKF to integrate ToF-based positioning with INS for better accuracy. Li *et al.* [23] employed EKF to fuse UWB-based and PDR-based positioning. The algorithm developed in [23], focused on using PDR for 2D navigation, which would necessitate significant modifications to adapt the approach for 3D navigation. Another navigation system that combines acoustic-based positioning with PDR-based positioning using an EKF was proposed in [24]. A limitation of the system described in [24] is that it requires a

smartphone to be attached to the pedestrian's leg. To enhance the accuracy of PDR-UWB-based positioning, Lee *et al.* [19] proposed a deep learning approach aimed at automatically adapting the PDR parameters for each individual. However, the algorithm in [19] has a drawback in that it relies on the pedestrian walking in a straight line to accurately determine heading. Additionally, extending the algorithm in [19] from 2D to 3D navigation poses significant challenges. In [25], the authors fuse a UWB-based positioning system with INS using a fixed-lag extended finite impulse response smoother (FEFIRS) algorithm. The algorithm described in [25] has the main drawback of requiring a certain lag to improve positioning results. Furthermore, Feng *et al.* [2] proposed a navigation system that integrates INS with a UWB-based positioning system using both EKF and UKF alongside two novel motion models.

While the fusion algorithms discussed in [2], [19], [21]–[25] improve positioning accuracy, they do not necessarily enhance orientation accuracy, which can degrade performance over time. Orientation estimation typically relies on accelerometers, gyroscopes [26]–[30], magnetometers [31]–[34], or a combination of them [35]–[38]. Among these, magnetometer-based approaches are widely used [39], but their accuracy can be significantly impacted by magnetic disturbances in indoor environments [39]. On the other hand, accelerometer-based methods suffer from error accumulation, while gyroscope-based methods experience drift errors in addition to error accumulation [39]. Consequently, relying solely on these three sensors for orientation estimation can compromise both orientation and positioning accuracy. The performance analysis provided in [39] compares various sensor fusion techniques for orientation estimation using smartphone sensors, including the Linear Kalman Filter (LKF), EKF, UKF, Particle Filter (PF), and Complementary Filter (CF). Although the work in [39] addresses orientation fusion, it relies solely on inertial sensors, which leads to accuracy degradation over time.

While previous work improves positioning accuracy through various fusion algorithms, orientation accuracy often remains unaddressed. Our focus is on fusing acoustic-ToF-based positioning with INS to enhance both position and orientation estimation. We propose novel algorithms based on Riemannian optimization to correct both position and orientation estimates obtained from INS. The acoustic system employs an array of receivers arranged in the shape of an isosceles triangle. As demonstrated in our previous work [5], Riemannian optimization effectively addresses the localization problem with high accuracy. The proposed algorithm leverages the positions of the triangle vertices to refine both the position and orientation estimates provided by the INS via EKF-based and UKF-based fusion. The primary contribution of this paper is the development of a novel algorithm that integrates Riemannian optimization to enhance the correction of position and orientation estimates derived from the INS. Thanks to the innovative projection algorithm detailed in Section V, the proposed method improves the accuracy of position and orientation in typical indoor environments, as validated by numerical simulations and experiments.

### III. PROBLEM FORMULATION

Accurate estimation of the position and orientation of a moving target is pivotal in INS [4]. While inertial navigation based on IMUs has high accuracy over short periods, its accuracy severely degrades over time due to error accumulation from the integration operations used in INS [4]. Therefore, it is common practice to fuse measurements from different sensors to improve the tracking accuracy of INS. We propose using three acoustic receivers arranged at the vertices of an isosceles triangle to enhance the accuracy of the estimated orientation and position obtained using INS.

The proposed system consists of three acoustic receivers with unknown positions placed at the vertices of an isosceles triangle, four acoustic transmitters (beacons) with known positions, and an IMU placed at the centroid of the isosceles triangle. The three receivers, together with the IMU, form the mobile device (MD). While the proposed system is evaluated using an equilateral triangle, which is a special case of an isosceles triangle, the proposed algorithm can be applied to any isosceles triangle. Figure 1 shows an illustration of the mobile device in the body frame.

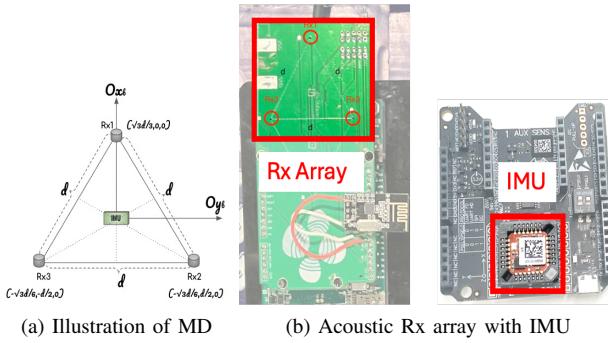


Fig. 1: Mobile device

As mentioned previously, while INS provides high rates, it suffers from accuracy degradation due to error accumulation. In contrast to INS, the position and orientation of the mobile device can be estimated with high accuracy using the three acoustic receivers without degradation over time. However, the acoustic positioning system requires an LOS between the transmitters and receivers to achieve high accuracy. Additionally, acoustic-based measurements are carried out at a slower rate compared to INS. Therefore, the proposed algorithm fuses the two measurements to achieve high-accuracy position and orientation estimation at a high rate, even in NLOS scenarios and over long periods.

The position and velocity of the centroid of the triangle in Figure 1, along with its orientation, are estimated using IMU measurements and classical INS equations [4]. These estimates are corrected using the positions of the vertices of the triangle, estimated using Riemannian localization [5]. The main contributions of this paper, presented in Section V, include the correction algorithm with a new derivation of the EKF equations and a novel projection algorithm on the Riemannian manifold that improves tracking accuracy.

### IV. POSITION AND ORIENTATION ESTIMATION

This section provides a brief summary of the inertial navigation [4] and Riemannian localization algorithms [5] used in this work. For more details on INS and Riemannian localization, we refer the reader to [4] and [5].

#### A. IMU Position and Orientation Estimation

Navigation systems should provide the position and orientation of an MD with respect to a fixed reference frame, known as the navigation frame. However, the IMU measures acceleration and angular velocity with respect to a moving frame, known as the body frame, which is attached to the MD. To explain the motion of the MD, we need to select a suitable coordinate system [2]. Let's denote the navigation (reference) frame and body frame axes as  $[O_{x_n}, O_{y_n}, O_{z_n}]$  and  $[O_{x_b}, O_{y_b}, O_{z_b}]$ , respectively. The centroid of the triangle, shown in Figure 1, represents the origin of the body frame. The axes of the body frame are aligned with the roll, pitch, and yaw axes of the MD. The origin of the navigation frame is fixed in the simulation and experimental setup. The axes of the navigation frame are aligned with the directions of north, east, and up. Each of the two frames form a right-hand system. Figure 2 shows the axes of the body and navigation frames.

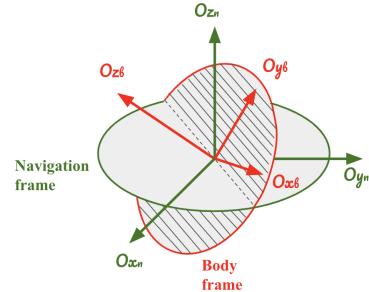


Fig. 2: The navigation and body frames axes

Any vector  $\mathbf{v}^b$  expressed in the body frame can be transformed to the navigation frame by pre-multiplying by the orientation matrix  $\mathbf{C}_b^n$  [4]:

$$\mathbf{v}^n = \mathbf{C}_b^n \mathbf{v}^b, \quad (1)$$

where  $\mathbf{C}_b^n$  is a  $3 \times 3$  matrix that defines the attitude of the body frame with respect to the navigation frame, and  $\mathbf{v}^n$  is the transformed vector expressed in the navigation frame. The attitude (orientation) matrix can be represented using direction cosines, Euler angles, or quaternions [4].

The IMU used in this work contains a triaxial accelerometer and a triaxial gyroscope which measure the 3D acceleration and angular velocity, represented in the body frame of the MD, respectively. However, the final estimated position and velocity of the mobile device should be expressed in the navigation frame. Therefore, it is required to estimate the orientation matrix  $\mathbf{C}_b^n$ . Let the angular velocity measured by the IMU be  $\omega_{nb}^b = [\omega_x, \omega_y, \omega_z]^T$ , which represents the turn rate of the MD with respect to the navigation frame as measured by the gyroscope. Under the assumption of known initial conditions,

the angular velocity can be integrated over time to obtain the orientation of the MD using the following equation [4]:

$$\mathbf{C}_b^n(k+1) = \mathbf{C}_b^n(k) + \Delta t \dot{\mathbf{C}}_b^n(k), \quad (2)$$

where  $\Delta t$  is the IMU sampling period and  $\dot{\mathbf{C}}_b^n$  is given by

$$\dot{\mathbf{C}}_b^n(k) = \mathbf{C}_b^n(k) \boldsymbol{\Omega}_{nb}^b(k), \quad (3)$$

where  $\boldsymbol{\Omega}_{nb}^b$  is the skew-symmetric matrix given by

$$\boldsymbol{\Omega}_{nb}^b(k) = \begin{bmatrix} 0 & -\omega_z(k) & \omega_y(k) \\ \omega_z(k) & 0 & -\omega_x(k) \\ -\omega_y(k) & \omega_x(k) & 0 \end{bmatrix}. \quad (4)$$

The orientation matrix is an orthogonal matrix, which means  $\mathbf{C}_n^b(k) = (\mathbf{C}_b^n(k))^{-1} = (\mathbf{C}_b^n(k))^T$ . The orthogonality of the orientation matrix is approximately achieved, assuming that the first-order approximation given in Equation 2 is accurate under a high sampling rate. While the matrix  $\mathbf{C}_b^n(k)$  transforms a vector from the body frame to the navigation frame,  $\mathbf{C}_n^b(k)$  transforms a vector from the navigation frame to the body frame. The operation  $[.]^T$  is the transpose operation. Consequently, the acceleration  $\mathbf{a}^b = [a_x^b, a_y^b, a_z^b]^T$  measured by the IMU can be expressed in the navigation frame using the following equation:

$$\mathbf{a}^{n'}(k) = [a_x^{n'}(k), a_y^{n'}(k), a_z^{n'}(k)]^T = \mathbf{C}_b^n(k) \mathbf{a}^b(k). \quad (5)$$

The acceleration vector  $\mathbf{a}^{n'}$  contains a gravitational acceleration component  $g$  which could be removed to obtain the acceleration at the center of the mobile device in the navigation frame as follows [2]:

$$\mathbf{a}^n(k) = \begin{bmatrix} a_x^{n'}(k) \\ a_y^{n'}(k) \\ a_z^{n'}(k) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \quad (6)$$

Under a high IMU update rate, the acceleration during the sampling period can be assumed to be constant. Consequently, the velocity at the center of the MD  $\mathbf{v}_c^n$  at time  $k+1$  can be written as:

$$\mathbf{v}_c^n(k+1) = \mathbf{v}_c^n(k) + \Delta t \mathbf{a}^n(k). \quad (7)$$

Similarly, the position of the MD  $\mathbf{p}_c^n$  can be written as:

$$\mathbf{p}_c^n(k+1) = \mathbf{p}_c^n(k) + \Delta t \mathbf{v}_c^n(k) + \frac{\Delta t^2}{2} \mathbf{a}^n(k). \quad (8)$$

Therefore, assuming the initial conditions  $\mathbf{p}_c^n(0)$ ,  $\mathbf{v}_c^n(0)$ , and  $\mathbf{C}_b^n(0)$  are known, the angular velocity  $\boldsymbol{\omega}_{nb}^b$  and the acceleration  $\mathbf{a}^b$ , measured using the IMU, can be used to estimate the position, velocity, and orientation of the MD at each time step.

### B. Position and Orientation Estimation Using Riemannian Optimization

This section summarizes our previous work on Riemannian optimization to estimate the centroid position and orientation of the MD [5]. The Riemannian optimization algorithm described in [5] is used to determine the positions of three receivers arranged as vertices of an isosceles triangle. In this paper, we specifically consider the case where the triangle is

equilateral, meaning all sides, including the base, are of equal length, denoted as  $d$ . The localization system includes three acoustic receivers arranged in an equilateral triangle rigidly attached to the MD, and four acoustic transmitters, or beacons, with known positions. Figure 3 illustrates this localization setup.

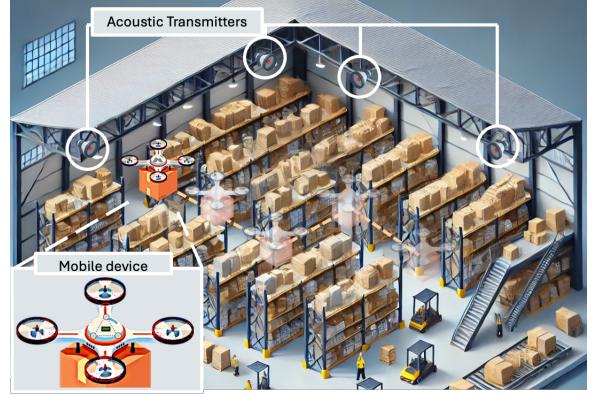


Fig. 3: Application for indoor localization

The MD's position is defined as the centroid of the equilateral triangle formed by the receivers. Thus, by estimating the 3D positions of the three vertices of this triangle, we can determine the 3D position and orientation of the MD. Let  $\mathbf{p}_i \in \mathbb{R}^3$  denote the 3D location of the  $i$ -th receiver, and let  $\mathbf{b}_j \in \mathbb{R}^3$  represent the position of the  $j$ -th beacon. These positions are organized in a matrix  $\mathbf{B} \in \mathbb{R}^{4 \times 3}$ , where the  $j$ -th row of the matrix corresponds to the location of the  $j$ -th beacon, i.e.,  $\mathbf{b}_j^T$ . The time of flight (TOF),  $\tau_{ij}$ , between the  $j$ -th transmitter and the  $i$ -th receiver can be estimated using correlation-based methods [40] [16]. The distances,  $r_{ij}$  between the transmitters and receivers are then calculated by multiplying the TOFs,  $\tau_{ij}$ , by the speed of sound. Assuming that all the distances  $r_{ij}$ , are subject to normally distributed noise and accounting for the fixed geometry of the receiver array, the position estimation problem can be formulated as the following constrained optimization problem [5]:

$$\min_{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^3} \sum_{i=1}^3 \left\| \mathbf{B} \mathbf{p}_i - \frac{1}{2} \|\mathbf{p}_i\|_2^2 \mathbf{1}_4 - \mathbf{y}_i \right\|_2^2 \quad (9a)$$

$$\text{s.t. } (\mathbf{p}_1 - \mathbf{p}_2)^T (\mathbf{p}_2 - \mathbf{p}_3) = -d^2 \cos\left(\frac{\pi}{3}\right) \quad (9b)$$

$$(\mathbf{p}_1 - \mathbf{p}_3)^T (\mathbf{p}_2 - \mathbf{p}_3) = d^2 \cos\left(\frac{\pi}{3}\right), \quad (9c)$$

where  $\mathbf{1}_4$  is the all-ones vector of dimension 4, and  $\mathbf{y}_i = \frac{1}{2}(\mathbf{b}^2 - \mathbf{r}_i^2)$  with  $\mathbf{b}^2$  and  $\mathbf{r}_i^2$  given by

$$\mathbf{b}^2 = \begin{pmatrix} \|\mathbf{b}_1\|_2^2 \\ \|\mathbf{b}_2\|_2^2 \\ \|\mathbf{b}_3\|_2^2 \\ \|\mathbf{b}_4\|_2^2 \end{pmatrix}, \mathbf{r}_i^2 = \begin{pmatrix} r_{i1}^2 \\ r_{i2}^2 \\ r_{i3}^2 \\ r_{i4}^2 \end{pmatrix}.$$

As demonstrated in our previous work [5], the constraints (9b) and (9c) fix the length of the base  $\|\mathbf{p}_2 - \mathbf{p}_3\|_2 = d$  and ensure that the lengths of the two side legs of the triangle are equal, i.e.,  $\|\mathbf{p}_1 - \mathbf{p}_2\|_2 = \|\mathbf{p}_1 - \mathbf{p}_3\|_2$ . We showed that

the set of points satisfying constraints (9b) and (9c) forms a Riemannian manifold [5]. Therefore, each feasible solution  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^3$  to (9) belongs to a set, named the *isosceles triangle manifold*, which is defined as follows [5]:

$$\mathcal{M} = \left\{ \{\mathbf{p}_i\}_{i=1}^3 \in \mathbb{R}^3 \mid (\mathbf{p}_1 - \mathbf{p}_2)^T(\mathbf{p}_2 - \mathbf{p}_3) = -d^2 \cos\left(\frac{\pi}{3}\right) \right. \\ \left. (\mathbf{p}_1 - \mathbf{p}_3)^T(\mathbf{p}_2 - \mathbf{p}_3) = d^2 \cos\left(\frac{\pi}{3}\right) \right\}.$$

Our algorithm in [5] efficiently addresses the constrained optimization problem using Riemannian optimization techniques. This section provides a summary of the first- and second-order Riemannian positioning algorithms, along with a straightforward method for determining the orientation of the MD based on the estimated positions. For further details on the algorithm, please refer to our previous work [5].

At each point  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] \in \mathcal{M}$ , the manifold can be locally approximated by a linear space called the *tangent space*, denoted by  $T_{\mathbf{P}}\mathcal{M}$ . To perform optimization on a manifold  $\mathcal{M}$ , it is necessary to define the notion of length that applies to tangent vectors by equipping each tangent space with an inner product known as the Riemannian metric. In our Riemannian positioning algorithm [5], we use the canonical inner product for matrix space,  $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Tr}(\mathbf{X}^T \mathbf{Y})$ .

Riemannian optimization begins with a feasible point  $\mathbf{P} \in \mathcal{M}$ . At each iteration, it determines a descent direction  $\xi_{\mathbf{P}}$  in the tangent space  $T_{\mathbf{P}}\mathcal{M}$  and a step size  $\eta$  to update the point  $\mathbf{P}$ . The tangent space  $T_{\mathbf{P}}\mathcal{M}$  provides a local linear approximation of the manifold's curvature around  $\mathbf{P}$ . Descent directions in this tangent space can be found using methods similar to those in unconstrained optimization due to its linear nature. However, since the Euclidean gradient  $\nabla_{\mathbf{P}} f$  and Hessian  $\nabla_{\mathbf{P}}^2 f$  are defined on the original high-dimensional space  $\mathcal{E}$  and not necessarily on  $T_{\mathbf{P}}\mathcal{M}$ , we need to define the Riemannian gradient  $\bar{\nabla}_{\mathbf{P}} f$  and Hessian  $\bar{\nabla}_{\mathbf{P}}^2 f$  [5]. The Riemannian gradient  $\bar{\nabla}_{\mathbf{P}} f$  is the projection of the Euclidean gradient  $\nabla_{\mathbf{P}} f$  onto the tangent space  $T_{\mathbf{P}}\mathcal{M}$ . Similarly, the Riemannian Hessian is the orthogonal projection of the directional derivative of the Riemannian gradient onto the tangent space. After determining a descent direction  $\xi_{\mathbf{P}}$ , which is a tangent vector in the tangent space  $T_{\mathbf{P}}\mathcal{M}$ , and a step size  $\eta$ , the current point is updated using  $\mathbf{Z} = \mathbf{P} + \eta \xi_{\mathbf{P}}$ . This update may yield a point outside the manifold, i.e.,  $\mathbf{Z} \in \mathbb{R}^{3 \times 3}$ . Therefore, it is necessary to "project" this update back onto the manifold while preserving its descent property, a process achieved using a *retraction* [41].

First-order Riemannian optimization algorithms require only gradient information to determine the descent direction, while second-order algorithms also require the Riemannian Hessian. For instance, in the steepest descent method on Riemannian manifolds, the descent direction is given by  $\xi_{\mathbf{P}} = -\bar{\nabla}_{\mathbf{P}} f / \|\bar{\nabla}_{\mathbf{P}} f\|_{\mathbf{P}}$  [41]. In contrast, Newton's method on Riemannian manifolds, a second-order algorithm, determines the descent direction  $\xi_{\mathbf{P}} \in T_{\mathbf{P}}\mathcal{M}$  by solving the Newton's equation  $\bar{\nabla}_{\mathbf{P}}^2 f[\xi_{\mathbf{P}}] = -\bar{\nabla}_{\mathbf{P}} f$ .

*1) Riemannian-Steepest-Descent-based Positioning Algorithm:* The algorithm begins with an initial point  $\mathbf{P} = \mathbf{P}_0 \in \mathcal{M}$ . It then iterates between determining a search direction and updating the current position. For two real numbers  $\alpha$  and  $\beta$ ,

we define a matrix  $\mathbf{U}_{\alpha}^{\beta}$  and a matrix  $\mathbf{S}$  as follows:

$$\mathbf{U}_{\alpha}^{\beta} = \begin{pmatrix} 0 & \alpha + \beta & -\alpha - \beta \\ \alpha + \beta & -2\alpha & \alpha - \beta \\ -\alpha - \beta & \alpha - \beta & 2\beta \end{pmatrix},$$

$$\mathbf{S} = \begin{pmatrix} \|\mathbf{P}\mathbf{U}_1^0\|_{\mathbf{P}}^2 & \langle \mathbf{P}\mathbf{U}_0^1, \mathbf{P}\mathbf{U}_1^0 \rangle_{\mathbf{P}} \\ \langle \mathbf{P}\mathbf{U}_1^0, \mathbf{P}\mathbf{U}_0^1 \rangle_{\mathbf{P}} & \|\mathbf{P}\mathbf{U}_0^1\|_{\mathbf{P}}^2 \end{pmatrix}, \quad (10)$$

where  $\langle \cdot, \cdot \rangle_{\mathbf{P}}$  denotes the inner product for matrices, and  $\|\mathbf{P}\|_{\mathbf{P}}^2 = \langle \mathbf{P}, \mathbf{P} \rangle_{\mathbf{P}}$ . The search direction is given by  $\xi_{\mathbf{P}} = -\bar{\nabla}_{\mathbf{P}} f / \|\bar{\nabla}_{\mathbf{P}} f\|_{\mathbf{P}}$ , where the Riemannian gradient  $\bar{\nabla}_{\mathbf{P}} f$  is computed using the following equation:

$$\bar{\nabla}_{\mathbf{P}} f = \nabla_{\mathbf{P}} f - \mathbf{P}\mathbf{U}_{\alpha}^{\beta} \quad (11)$$

with the reals  $\alpha$  and  $\beta$  are determined as the solution to the linear system:

$$\mathbf{S} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \langle \nabla_{\mathbf{P}} f, \mathbf{P}\mathbf{U}_1^0 \rangle_{\mathbf{P}} \\ \langle \nabla_{\mathbf{P}} f, \mathbf{P}\mathbf{U}_0^1 \rangle_{\mathbf{P}} \end{pmatrix}. \quad (12)$$

The step size  $\eta$  is then chosen via backtracking to satisfy the Wolfe conditions [42]. Then, the point is updated using the step size and descent direction  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3] = \mathbf{P} + \eta \xi_{\mathbf{P}}$ . The tangent vector  $\xi_{\mathbf{P}}$ , scaled by the step size  $\eta$ , is retracted onto the manifold using the following retraction formula:

$$\mathbf{R}_{\mathbf{P}}(\xi_{\mathbf{P}}) = \lambda \begin{pmatrix} \gamma \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{pmatrix}, \quad (13)$$

where  $\lambda$  and  $\gamma$  are defined as:

$$\lambda = \sqrt{\frac{d^2 \cos\left(\frac{\pi}{3}\right)}{(\gamma \mathbf{z}_1 - \mathbf{z}_3)^T (\mathbf{z}_2 - \mathbf{z}_3)}} \quad (14)$$

$$\gamma = \frac{(\mathbf{z}_2 + \mathbf{z}_3)^T (\mathbf{z}_2 - \mathbf{z}_3)}{2 \mathbf{z}_1^T (\mathbf{z}_2 - \mathbf{z}_3)}.$$

Define the isosceles triangle  $\mathbf{U}$  as  $[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = [\gamma \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3]$ . To obtain the point on the manifold, scale the sides of  $\mathbf{U}$  using:

$$\mathbf{P} = \sqrt{\frac{d^2 \cos\left(\frac{\pi}{3}\right)}{(\mathbf{u}_1 - \mathbf{u}_3)^T (\mathbf{u}_2 - \mathbf{u}_3)}} \mathbf{U}.$$

The issues with poles in (14) are addressed using a simple algorithm detailed in [5].

*2) Riemannian Trust Region Localization Algorithm:* The Riemannian trust region method on the manifold  $\mathcal{M}$  for a cost function  $f$  involves adding to the current iterate  $\mathbf{P} \in \mathcal{M}$  the update vector  $\xi_{\mathbf{P}} \in T_{\mathbf{P}}\mathcal{M}$ , obtained by solving the following trust-region subproblem on  $T_{\mathbf{P}_k}\mathcal{M}$  [43] :

$$\min_{\xi \in T_{\mathbf{P}_k}\mathcal{M}} m_{\mathbf{P}_k}(\xi) = f(\mathbf{P}_k) + \langle \bar{\nabla}_{\mathbf{P}_k} f(\mathbf{P}_k), \xi \rangle + \frac{1}{2} \langle \bar{\nabla}_{\mathbf{P}_k}^2 f(\xi), \xi \rangle \quad (15)$$

$$\text{s.t. } \|\xi\|_{\mathbf{P}_k} \leq \Delta_k^2,$$

where  $\Delta_k$  is the trust region radius, and  $\bar{\nabla}_{\mathbf{P}_k}^2 f$  is the Riemannian hessian which can be obtained using the following

equations:

$$\bar{\nabla}_{\mathbf{P}_k}^2 f[\xi_{\mathbf{P}_k}] = \Pi_{\mathbf{P}_k} (\nabla_{\mathbf{P}_k}^2 f[\xi_{\mathbf{P}_k}] - \xi_{\mathbf{P}_k} \mathbf{U}_\alpha^\beta - \mathbf{P}_k \mathbf{U}_\alpha^\dot{\beta}), \quad (16)$$

where the orthogonal projection  $\Pi_{\mathbf{P}_k}$  is given by:

$$\Pi_{\mathbf{P}}(\mathbf{Z}) = \mathbf{Z} - \mathbf{P}\mathbf{U}_\alpha^\beta.$$

The parameters  $\alpha$  and  $\beta$  are the solution to 12, where  $\mathbf{S}$  is defined in 10. Their directional derivatives  $\dot{\alpha}$  and  $\dot{\beta}$  are obtained as the solution to the system:

$$\begin{aligned} \mathbf{S} \begin{pmatrix} \dot{\alpha} \\ \dot{\beta} \end{pmatrix} &= \left( \begin{array}{l} \langle \nabla_{\mathbf{P}_k}^2 f[\xi_{\mathbf{P}_k}], \mathbf{P}_k \mathbf{U}_1^0 \rangle_{\mathbf{P}_k} + \langle \nabla_{\mathbf{P}_k} f, \xi_{\mathbf{P}_k} \mathbf{U}_1^0 \rangle_{\mathbf{P}_k} \\ \langle \nabla_{\mathbf{P}_k}^2 f[\xi_{\mathbf{P}_k}], \mathbf{P}_k \mathbf{U}_0^1 \rangle_{\mathbf{P}_k} + \langle \nabla_{\mathbf{P}_k} f, \xi_{\mathbf{P}_k} \mathbf{U}_0^1 \rangle_{\mathbf{P}_k} \end{array} \right) \\ &\quad - \dot{\mathbf{S}} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \end{aligned}$$

The directional derivative of  $\mathbf{S}$  in the direction  $\xi_{\mathbf{P}_k}$  is given by:

$$\dot{\mathbf{S}} = \begin{pmatrix} 2\langle \xi_{\mathbf{P}_k} \mathbf{U}_1^0, \mathbf{P}_k \mathbf{U}_1^0 \rangle_{\mathbf{P}_k} & \begin{array}{l} \langle \xi_{\mathbf{P}_k} \mathbf{U}_0^1, \mathbf{P}_k \mathbf{U}_1^0 \rangle_{\mathbf{P}_k} \\ + \langle \mathbf{P}_k \mathbf{U}_0^1, \xi_{\mathbf{P}_k} \mathbf{U}_1^0 \rangle_{\mathbf{P}_k} \end{array} \\ \begin{array}{l} \langle \xi_{\mathbf{P}_k} \mathbf{U}_1^0, \mathbf{P}_k \mathbf{U}_0^1 \rangle_{\mathbf{P}_k} \\ + \langle \mathbf{P}_k \mathbf{U}_1^0, \xi_{\mathbf{P}_k} \mathbf{U}_0^1 \rangle_{\mathbf{P}_k} \end{array} & 2\langle \xi_{\mathbf{P}_k} \mathbf{U}_0^1, \mathbf{P}_k \mathbf{U}_0^1 \rangle_{\mathbf{P}_k} \end{pmatrix}.$$

The trust region subproblem is solved using any of the available methods to obtain  $\xi_k$ . The new iterate is then computed by retracting  $\xi_k$  onto the manifold, i.e.,  $\mathbf{P}_+ = \mathbf{R}_{\mathbf{P}_k}(\xi_k)$ . To decide whether to accept or reject the new iterate and to update the trust region radius, the following quotient is used:

$$\rho_k = \frac{f(\mathbf{P}_k) - f(\mathbf{R}_{\mathbf{P}_k}(\xi_k))}{m_{\mathbf{P}_k}(\mathbf{0}_{\mathbf{P}_k}) - m_{\mathbf{P}_k}(\xi_k)}. \quad (17)$$

Here,  $\rho_k$  is compared against predefined thresholds to determine the next steps. If  $\rho_k$  is very small, then the new iterate should be rejected, and the trust region radius must be reduced. If  $\rho_k$  is small but not too small, the new iterate should be accepted and the trust radius should be kept the same. If  $\rho_k$  is close to 1, then the new iterate should be accepted and the radius can be increased.

3) *Centroid Position and Orientation of the MD*: The centroid position and orientation of the MD can be determined using the estimated positions of the vertices of the equilateral triangle. The centroid position is calculated by averaging the positions of the vertices:

$$\mathbf{p}_c^R = \sum_{i=1}^3 \frac{\mathbf{p}_i}{3}. \quad (18)$$

Here, the superscript  $R$  in  $\mathbf{p}_c^R$  distinguishes the centroid position obtained using the Riemannian algorithm from the position obtained using an INS. The orientation matrix  $\mathbf{C}_b^n$  is determined by computing the body frame axes  $\mathbf{C}_b^n = [O_{x_b}^n, O_{y_b}^n, O_{z_b}^n]$ , expressed in the navigation frame as follows:

$$O_{x_b}^n = \frac{\mathbf{p}_1 - \mathbf{p}_c^R}{\|\mathbf{p}_1 - \mathbf{p}_c^R\|} \quad (19)$$

$$O_{y_b}^{n'} = \mathbf{p}_2 - \mathbf{p}_c^R - <\mathbf{p}_2 - \mathbf{p}_c^R, O_{x_b}^n> O_{x_b}^n \quad (20)$$

$$O_{y_b}^n = \frac{O_{y_b}^{n'}}{\|O_{y_b}^{n'}\|} \quad (21)$$

$$O_z^n = O_{x_b}^n \times O_{y_b}^n. \quad (22)$$

Finally, as shown in Figure 1, the centroid position can be transformed into the vertices' positions using the orientation matrix and the following equations:

$$\mathbf{P} = [\mathbf{p}_c^R, \mathbf{p}_c^R, \mathbf{p}_c^R] + \mathbf{C}_b^n \begin{bmatrix} \frac{\sqrt{3}d}{3} & -\frac{\sqrt{3}d}{6} & -\frac{\sqrt{3}d}{6} \\ 0 & \frac{d}{2} & -\frac{d}{2} \\ 0 & 0 & 0 \end{bmatrix}. \quad (23)$$

## V. FUSION OF INS AND ACOUSTIC POSITION AND ORIENTATION USING KALMAN FILTER

This section introduces a new formulation of the state equations that incorporates the orientation matrix as part of the state vector, leading to new EKF equations. Additionally, a novel retraction operator is presented, which projects the output of the EKF and UKF onto the isosceles triangle manifold, thereby enhancing the accuracy of tracking the target's position and orientation.

Let  $\mathbf{x} = [\mathbf{p}_c^T, \mathbf{v}_c^T, \mathbf{c}^T]^T \in \mathbb{R}^{15}$  denote the state vector to be estimated. Here,  $\mathbf{v}_c$  represents the velocity of the center of the MD, and  $\mathbf{c}$  is the vectorized orientation matrix  $\mathbf{C}_b^n$ . It is established that the linear Kalman filter is optimal for state estimation when the IMU and acoustic sensor noises are independent and normally distributed with zero mean [44]. However, since the state equations are nonlinear in this case, we compare the performance of the EKF and UKF for estimating the MD's position and orientation. The state equations are given by:

$$\mathbf{x}(k+1) = \underbrace{\begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t \mathbf{I}_{3 \times 3} & \frac{\Delta t^2}{2} \mathbf{a}_b^\times \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \Delta t \mathbf{a}_b^\times + \frac{\Delta t^2}{2} (\mathbf{a}_b^\times)(\Omega_{nb}^b)^\times \\ \mathbf{0}_{9 \times 3} & \mathbf{0}_{9 \times 3} & \mathbf{I}_{9 \times 9} + \Delta t (\Omega_{nb}^b)^\times + \frac{\Delta t^2}{2} (\Omega_{nb}^b)^\times \end{bmatrix}}_{\mathbf{F}(\mathbf{a}_b, \Omega_{nb}^b)} \mathbf{x}(k), \quad (24)$$

where  $\mathbf{a}_b^\times = \mathbf{a}_b^T \otimes \mathbf{I}_{3 \times 3}$  is the Kronecker product of the acceleration vector  $\mathbf{a}_b^T$  and the identity matrix  $\mathbf{I}_{3 \times 3}$ ,  $\Omega_{nb}^b = \Omega_{nb}^{b \ T} \otimes \mathbf{I}_{3 \times 3}$  is the Kronecker product of  $\Omega_{nb}^{b \ T}$  and the identity matrix, and  $(\Omega_{nb}^b)^\times = (\Omega_{nb}^b \otimes \mathbf{I}_{3 \times 3})(\Omega_{nb}^b \otimes \mathbf{I}_{3 \times 3})$ . Note that in the derivations throughout this section, the acceleration  $\mathbf{a}_b$  is assumed to be free acceleration (gravity component subtracted).

The nonlinearity in the state-transition matrix  $\mathbf{F}(\mathbf{a}_b, \Omega_{nb}^b)$  arises from the multiplication of the inputs, acceleration  $\mathbf{a}_b$  and angular velocity  $\omega_b$ , by the state vector. The input noise can be modeled as additive Gaussian noise:

$$\mathbf{u}(k) = \begin{bmatrix} \mathbf{a}_b(k) \\ \omega_b(k) \end{bmatrix} = \tilde{\mathbf{u}}(k) + \mathbf{n}(k) \quad (25)$$

where  $\tilde{\mathbf{u}}(k)$  is the noiseless input, i.e. the actual acceleration and angular velocity of the MD, and  $\mathbf{n}(k)$  is the input noise, produced by the IMU's electronic components, normally distributed with zero mean and covariance matrix  $\mathbf{Q} = \text{diag}(\sigma_{\mathbf{a}_x}^2, \sigma_{\mathbf{a}_y}^2, \sigma_{\mathbf{a}_z}^2, \sigma_{\omega_x}^2, \sigma_{\omega_y}^2, \sigma_{\omega_z}^2)$ .

Let the measurement vector  $\mathbf{q} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T]^T \in \mathbb{R}^9$  represent the positions of the vertices, estimated using the Riemannian localization method. The measurement equation is:

$$\mathbf{q}(k+1) = \mathbf{H}\mathbf{x}(k+1) + \nu(k+1), \quad (26)$$

where the measurement (observation) matrix  $\mathbf{H}$  is:

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{d}_1^T \otimes \mathbf{I}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{d}_2^T \otimes \mathbf{I}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{d}_3^T \otimes \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad (27)$$

where  $\mathbf{d}_i$  represents the vector from the centroid of the triangle to the  $i^{\text{th}}$  vertex, and  $\nu(k)$  is the measurement noise, normally distributed with zero mean and covariance matrix  $\mathbf{R}_k = \text{diag}(\sigma_{\mathbf{p}_{1x}}^2, \sigma_{\mathbf{p}_{1y}}^2, \dots, \sigma_{\mathbf{p}_{3z}}^2)$ . While the measurement equation is linear, the process equation is nonlinear. Therefore, the linear Kalman filter is not optimal. This paper evaluates the performance of the EKF and UKF in estimating the position and orientation of the MD.

### A. Extended Kalman Filter

The EKF operates in two main steps; prediction and correction. Given an initial estimate of the state  $\mathbf{x}(0)$ , the state equations are used to predict the state and its covariance matrix as follows:

$$\bar{\mathbf{x}}(k+1) = \mathbf{F}(\mathbf{a}_b, \omega_b)\hat{\mathbf{x}}(k) \quad (28)$$

$$\bar{\mathbf{P}}(k+1) = \mathbf{F}(\mathbf{a}_b, \omega_b)\hat{\mathbf{P}}(k)\mathbf{F}^T(\mathbf{a}_b, \omega_b) + \mathbf{F}_u \mathbf{Q}_k \mathbf{F}_u^T, \quad (29)$$

where  $\bar{\mathbf{x}}(k+1)$  and  $\bar{\mathbf{P}}(k+1)$  represent the predicted state and its covariance matrix at time step  $k+1$ , while  $\hat{\mathbf{x}}(k)$  and  $\hat{\mathbf{P}}(k)$  denote the corrected state and covariance matrix at time step  $k$ .  $\mathbf{F}_u$  is the Jacobian matrix of  $\mathbf{F}(\mathbf{a}_b, \omega_b)$  with respect to  $\mathbf{p}_c$ ,  $\mathbf{v}_c$ , and  $\mathbf{c}$ . The Jacobian matrix  $\mathbf{F}_u$  is given by:

$$\mathbf{F}_u = \begin{bmatrix} \nabla_{\mathbf{a}_b} \mathbf{p}_c^T & \nabla_{\omega_b} \mathbf{p}_c^T \\ \nabla_{\mathbf{a}_b} \mathbf{v}_c^T & \nabla_{\omega_b} \mathbf{v}_c^T \\ \nabla_{\mathbf{a}_b} \mathbf{c}^T & \nabla_{\omega_b} \mathbf{c}^T \end{bmatrix}, \quad (30)$$

where the elements of  $\mathbf{F}_u$  are calculated as follows:

$$\nabla_{\mathbf{a}_b} \mathbf{p}_c^T = \frac{\Delta t^2}{2} \mathbf{C}_b^n, \quad \nabla_{\omega_b} \mathbf{p}_c^T = \mathbf{0}_{3 \times 3}, \quad \nabla_{\mathbf{a}_b} \mathbf{c}^T = \mathbf{0}_{9 \times 3}, \quad (31)$$

$$\nabla_{\omega_b} \mathbf{c}^T = \Delta t [\mathbf{E}_1^{\times T} \mathbf{c}, \mathbf{E}_2^{\times T} \mathbf{c}, \mathbf{E}_3^{\times T} \mathbf{c}] + \quad (32)$$

$$\frac{\Delta t^2}{2} [(\mathbf{x} \Omega_{nb}^2)^{\times} \mathbf{c}, (\mathbf{y} \Omega_{nb}^2)^{\times} \mathbf{c}, (\mathbf{z} \Omega_{nb}^2)^{\times} \mathbf{c}], \quad (33)$$

$$\nabla_{\mathbf{a}_b} \mathbf{v}_c^T = \Delta t \mathbf{C}_b^n + \frac{\Delta t^2}{2} \mathbf{C}_b^n \Omega_{nb}^b, \quad (34)$$

$$\nabla_{\omega_b} \mathbf{v}_c^T = \frac{\Delta t^2}{2} \mathbf{C}_b^n \Xi_b^T, \quad (35)$$

where the  $\mathbf{A}^{\times}$  denotes the Kronecker product of  $\mathbf{A}$  with the identity matrix  $\mathbf{I}_{3 \times 3}$ ,  $\mathbf{e}_1 = [1, 0, 0]$ ,  $\mathbf{e}_2 = [0, 1, 0]$ ,  $\mathbf{e}_3 = [0, 0, 1]$ , and  $\mathbf{E}_i$  is the skew-symmetric matrix of  $\mathbf{e}_i$  where  $i = 1, 2, 3$ . Moreover, the matrices  $x\Omega_{nb}^2$ ,  $y\Omega_{nb}^2$ ,  $z\Omega_{nb}^2$ ,

and  $\Xi_b^T$  are given by:

$$x\Omega_{nb}^2 = \frac{\partial}{\partial \omega_x} (\Omega_{nb}^2) = \begin{bmatrix} 0 & \omega_y & \omega_z \\ \omega_y & -2\omega_x & 0 \\ \omega_z & 0 & -2\omega_x \end{bmatrix}, \quad (36)$$

$$y\Omega_{nb}^2 = \frac{\partial}{\partial \omega_y} (\Omega_{nb}^2) = \begin{bmatrix} -2\omega_y & \omega_x & 0 \\ \omega_x & 0 & \omega_z \\ 0 & \omega_z & -2\omega_y \end{bmatrix}, \quad (37)$$

$$z\Omega_{nb}^2 = \frac{\partial}{\partial \omega_z} (\Omega_{nb}^2) = \begin{bmatrix} -2\omega_z & 0 & \omega_x \\ 0 & -2\omega_z & \omega_y \\ \omega_x & \omega_y & 0 \end{bmatrix}, \quad (38)$$

$$\Xi_b^T = \begin{bmatrix} 0 & \mathbf{a}_{bz} & -\mathbf{a}_{by} \\ -\mathbf{a}_{bz} & 0 & \mathbf{a}_{bx} \\ \mathbf{a}_{by} & -\mathbf{a}_{bx} & 0 \end{bmatrix}. \quad (39)$$

To correct the predicted state and covariance matrix at time step  $k+1$ , compute the Kalman gain  $\mathbf{K}_{k+1}$  as follows:

$$\mathbf{K}_{k+1} = \bar{\mathbf{P}}(k+1) \mathbf{H}^T (\mathbf{H} \bar{\mathbf{P}}(k+1) \mathbf{H}^T + \mathbf{R}_{k+1})^{-1} \quad (40)$$

Finally, update the corrected state and covariance matrix using:

$$\hat{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k+1) + \mathbf{K}_{k+1} (\mathbf{q}(k+1) - \mathbf{H} \bar{\mathbf{x}}(k+1)) \quad (41)$$

$$\hat{\mathbf{P}}(k+1) = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}) \bar{\mathbf{P}}(k+1). \quad (42)$$

### B. Unscented Kalman Filter

EKF uses a first-order approximation of the nonlinear state equations, which introduces linearization errors. In contrast, UKF avoids linearization. Instead, the UKF estimates the mean and covariance of the state variable through a process known as the unscented transform. The UKF employs a deterministic sampling method to represent the state vector's distribution using a set of carefully chosen samples called sigma points. These sigma points capture the mean and covariance of the state vector, and after being propagated through the nonlinear transformation, they provide estimates of *a posteriori* mean and covariance [45].

The unscented transform (UT) is an algorithm used to compute the mean and covariance of a random variable undergoing a nonlinear transformation [45]. Given a state vector  $\mathbf{x}$  of dimension 15, 31 sigma points  $\chi_i$  are computed [45]. Let  $\mu_{\mathbf{x}}$  be the mean of the state vector and  $\mathbf{P}_{\mathbf{x}}$  its covariance matrix. The sigma points are calculated as follows:

$$\chi^{(0)} = \mu_{\mathbf{x}}, \quad (43)$$

$$\chi^{(i)} = \mu_{\mathbf{x}} + \sqrt{15 + \kappa} \text{Col}_i \mathbf{L}, \quad i = 1, 2, \dots, 15, \quad (44)$$

$$\chi^{(i+15)} = \mu_{\mathbf{x}} - \sqrt{15 + \kappa} \text{Col}_i \mathbf{L}, \quad i = 1, 2, \dots, 15, \quad (45)$$

where  $\mathbf{L}$  is a lower triangular matrix obtained from the Cholesky decomposition of  $\mathbf{P}_{\mathbf{x}}$ , i.e.,  $\mathbf{LL}^T = \mathbf{P}_{\mathbf{x}}$ . The parameter  $\kappa = (\alpha^2 - 1) \times 15$  is a scaling factor where  $\alpha$  determines the spread of the sigma points around the mean, and  $\text{Col}_i \mathbf{L}$  represents the  $i^{\text{th}}$  column of the matrix  $\mathbf{L}$ . After calculating the sigma points, they are propagated through the state equation:

$$\bar{\chi}^{(i)} = \mathbf{F}(\mathbf{a}_b, \omega_b) \chi^{(i)}, \quad i = 0, 1, \dots, 30. \quad (46)$$

The predicted mean and covariance of the state vector are then computed using the propagated sigma points:

$$W_m^{(0)} = \frac{\kappa}{\kappa + 15}, \quad W_c^{(0)} = \frac{\kappa}{\kappa + 15} + 3 - \alpha^2, \quad (47)$$

$$W_m^{(i)} = W_c^{(i)} = \frac{1}{2(\kappa + 15)}, \quad (48)$$

$$\bar{\mathbf{x}}(k+1) = \sum_{i=0}^{30} W_m^{(i)} \bar{\chi}^{(i)}, \quad (49)$$

$$\bar{\mathbf{P}}_{\mathbf{x}}(k+1) = \sum_{i=0}^{30} W_c^{(i)} \left( \bar{\chi}^{(i)} - \bar{\mathbf{x}}(k+1) \right) \left( \bar{\chi}^{(i)} - \bar{\mathbf{x}}(k+1) \right)^T. \quad (50)$$

Next, the measurements are predicted from the propagated sigma points:

$$\bar{\mathcal{Y}}^{(i)} = \mathbf{H} \bar{\chi}^{(i)}, \quad i = 0, 1, \dots, 30. \quad (51)$$

The mean and covariance of the measurements are:

$$\bar{\mathbf{q}}(k+1) = \sum_{i=0}^{30} W_m^{(i)} \bar{\mathcal{Y}}^{(i)}, \quad (52)$$

$$\bar{\mathbf{P}}_{\mathbf{q}}(k+1) = \sum_{i=0}^{30} W_c^{(i)} \left( \bar{\mathcal{Y}}^{(i)} - \bar{\mathbf{q}}(k+1) \right) \left( \bar{\mathcal{Y}}^{(i)} - \bar{\mathbf{q}}(k+1) \right)^T. \quad (53)$$

Finally, the cross-covariance and Kalman gain are calculated as follows:

$$\bar{\mathbf{P}}_{\mathbf{xq}}(k+1) = \sum_{i=0}^{30} W_c^{(i)} \left( \bar{\chi}^{(i)} - \bar{\mathbf{x}}(k+1) \right) \left( \bar{\mathcal{Y}}^{(i)} - \bar{\mathbf{q}}(k+1) \right)^T. \quad (54)$$

$$\mathbf{K}(k+1) = \bar{\mathbf{P}}_{\mathbf{xq}}(k+1) \bar{\mathbf{P}}_{\mathbf{q}}^{-1}(k+1). \quad (55)$$

The corrected mean and covariance are then computed as:

$$\hat{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k+1) + \mathbf{K}(k+1)(\mathbf{q}(k+1) - \bar{\mathbf{q}}(k+1)), \quad (56)$$

$$\hat{\mathbf{P}}_{\mathbf{x}}(k+1) = \bar{\mathbf{P}}_{\mathbf{x}}(k+1) - \mathbf{K}(k+1) \hat{\mathbf{P}}_{\mathbf{q}}(k+1) \mathbf{K}^T(k+1). \quad (57)$$

### C. Projection of the Estimated State to the Manifold

The estimated position and orientation matrix are transformed into the positions of three vertices of a triangle. However, whether using the EKF or the UKF, these vertices positions are not guaranteed to lie on the isosceles triangle manifold. Therefore, the final step in the proposed algorithm is to "project" the estimated vertices positions onto the isosceles triangle manifold. This section introduces a novel and efficient retraction algorithm that projects the vertices obtained from the EKF or UKF onto the isosceles triangle manifold.

The retraction operator described in [5] achieves high localization accuracy, even when the initial estimate is far from the optimal solution. However, it assigns more weight to one of the vertices when the term  $\mathbf{z}_1^T(\mathbf{z}_2 - \mathbf{z}_3)$  in (14) approaches zero, which may not be efficient near the true solution. This section presents a more efficient retraction method for points close to the optimal solution. Given high sampling rates, the output

of the EKF or UKF, while not necessarily on the manifold, is assumed to be close to the optimal solution. Hence, an efficient retraction is one that adjusts the EKF or UKF output to the manifold with minimal perturbation.

Let  $\mathbf{Z}^k = [\mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{z}_3^k]$  denote the positions of the vertices obtained using EKF or UKF. The new retraction algorithm is implemented in four steps. First, the centroid of  $\mathbf{Z}^k$  is translated to the origin. This is achieved using the translation:

$$\begin{pmatrix} \mathbf{z}_1' \\ \mathbf{z}_2' \\ \mathbf{z}_3' \end{pmatrix} = \begin{pmatrix} \mathbf{z}_1^k - \mathbf{z}_c \\ \mathbf{z}_2^k - \mathbf{z}_c \\ \mathbf{z}_3^k - \mathbf{z}_c \end{pmatrix}, \quad (58)$$

where  $\mathbf{z}_c = (\mathbf{z}_1^k + \mathbf{z}_2^k + \mathbf{z}_3^k)/3$ . The second step involves scaling the triangle such that the length of the base side is  $d$ . This scaling is performed as follows:

$$\begin{pmatrix} \mathbf{z}_1'' \\ \mathbf{z}_2'' \\ \mathbf{z}_3'' \end{pmatrix} = \lambda^k \begin{pmatrix} \mathbf{z}_1' \\ \mathbf{z}_2' \\ \mathbf{z}_3' \end{pmatrix}, \quad (59)$$

where  $\lambda^k = d/\|\mathbf{z}_2' - \mathbf{z}_3'\|$ . In the third step,  $\mathbf{z}_1''$  is translated in the direction of the unit vector  $\mathbf{u}_z = (\mathbf{z}_3'' - \mathbf{z}_2'')/d$ . The magnitude of this translation vector  $\gamma^k$  is chosen such that the translated point  $\mathbf{z}_1'''$  falls onto the bisector of the triangle's base. The result of this operation is an isosceles triangle centered around the origin. The median  $[\mathbf{z}_1'' \mathbf{z}_{2,3}'']$ , from the vertex  $\mathbf{z}_1''$  that bisects the base is  $\mathbf{z}_1'' - (\mathbf{z}_3'' + \mathbf{z}_2'')/2$ . The magnitude of the translation  $\gamma^k$  is simply found as the projection of the median  $[\mathbf{z}_1'' \mathbf{z}_{2,3}'']$  onto the unit vector  $\mathbf{u}_z$ :

$$\gamma^k = \left\langle \mathbf{z}_1'' - \frac{\mathbf{z}_3'' + \mathbf{z}_2''}{2}, \frac{\mathbf{z}_3'' - \mathbf{z}_2''}{d} \right\rangle, \quad (60)$$

where  $\langle \cdot, \cdot \rangle$  is the inner product of vectors. The updated positions of the vertices after this step are:

$$\begin{pmatrix} \mathbf{z}_1''' \\ \mathbf{z}_2''' \\ \mathbf{z}_3''' \end{pmatrix} = \begin{pmatrix} \mathbf{z}_1'' + \gamma^k \frac{\mathbf{z}_3'' - \mathbf{z}_2''}{d} \\ \mathbf{z}_2'' \\ \mathbf{z}_3'' \end{pmatrix}, \quad (61)$$

An illustration of the third step is shown in Figure 4.

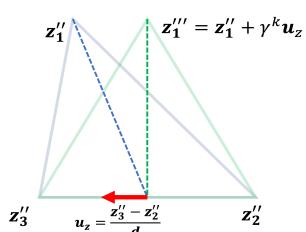


Fig. 4: Illustration of step 3 in the new retraction algorithm

The fourth and final step of the new retraction algorithm involves translating all the points back by the centroid  $\mathbf{z}_c$ . This ensures that the final positions of the vertices are properly adjusted relative to their original placement. An illustration of all four steps of the new retraction process is provided in Figure 5. After completing these steps, the output of the retraction algorithm consists of the three vertices of an isosceles triangle, with the base length  $d$ .

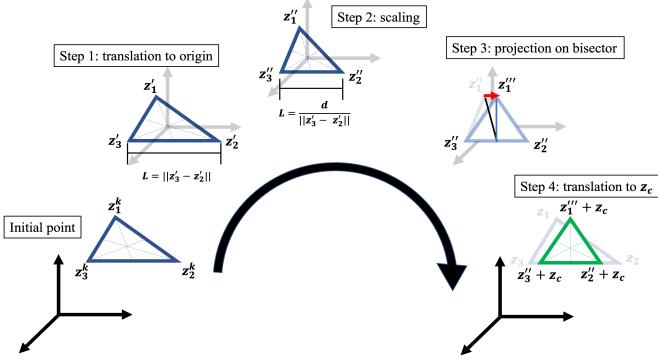


Fig. 5: Illustration of the new retraction algorithm

With straightforward mathematical manipulations, the new retraction process can be simplified as detailed in Algorithm 1.

#### Algorithm 1 Retraction onto isosceles triangle manifold

**Require:** Length  $d > 0$ , initialization  $\mathbf{z}_1^k, \mathbf{z}_2^k, \mathbf{z}_3^k$

- 1: Compute the centroid  $\mathbf{z}_c = \frac{\mathbf{z}_1^k + \mathbf{z}_2^k + \mathbf{z}_3^k}{3}$  and scaling factor  $\lambda^k = \frac{d}{\|\mathbf{z}_3^k - \mathbf{z}_2^k\|}$
  - 2: Compute the parameter  $\gamma^k = (\lambda^k)^2 \left\langle \mathbf{z}_1 - \frac{\mathbf{z}_3 + \mathbf{z}_2}{2}, \frac{\mathbf{z}_3 - \mathbf{z}_2}{d} \right\rangle$
  - 3: Compute the projected positions  $[\mathbf{p}_1^R, \mathbf{p}_2^R, \mathbf{p}_3^R]$
- $$\begin{pmatrix} \mathbf{p}_1^R \\ \mathbf{p}_2^R \\ \mathbf{p}_3^R \end{pmatrix} = \begin{pmatrix} \lambda^k(\mathbf{z}_1 - \mathbf{z}_c) + \frac{\gamma^k}{d} \lambda^k(\mathbf{z}_3 - \mathbf{z}_2) + \mathbf{z}_c \\ \lambda^k(\mathbf{z}_2 - \mathbf{z}_c) + \mathbf{z}_c \\ \lambda^k(\mathbf{z}_3 - \mathbf{z}_c) + \mathbf{z}_c \end{pmatrix}$$

## VI. EXPERIMENTAL SETUP

The acoustic localization system, depicted schematically in Figure 6, comprises three main components: the Master Station, Acoustic Receiver, and Acoustic Transmitters. Each component is equipped with an NRF24L01 wireless module. The Master Station's module operates in transmit-only mode, while those on the Acoustic Transmitters and Receiver operate in receive-only mode.

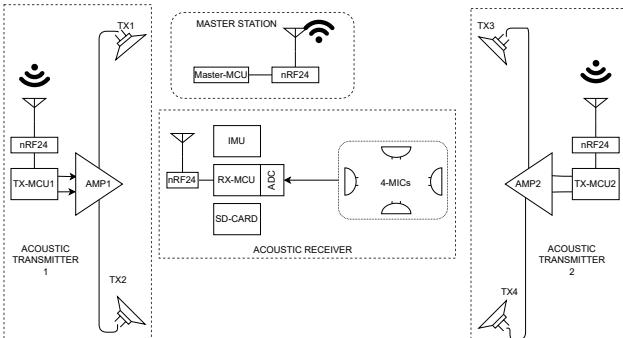


Fig. 6: The system is composed of three conceptual components: (1) Master Station (2) Acoustic Receiver (3) Acoustic Transmitters

The master station, depicted in Figure 7, utilizes an STM32F469 Discovery kit [46] with an NRF24L01 module [47]. Each of the two acoustic transmitter stations also employs STM32F469 Discovery kits with NRF24L01 modules, XM-N1004 Sony stereo amplifiers [48], and two Pioneer TS-T110 tweeters mounted on fixtures with retro-reflective spheres on tripods. The Pioneer TS-T110 tweeter has a bandwidth (BW) of 7 kHz and a central frequency of 20 kHz. Figure 8 illustrates the TS-T110 tweeter setup with its fixture.

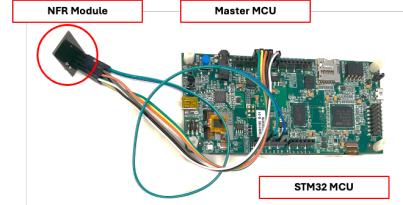


Fig. 7: Master station

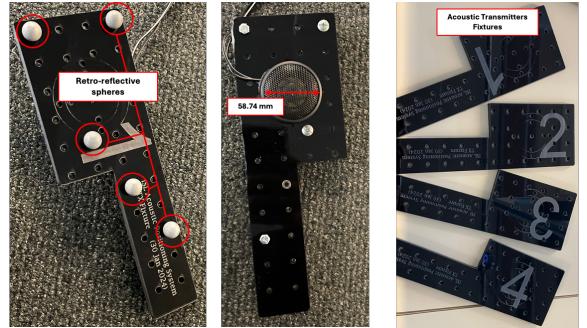


Fig. 8: Acoustic transmitter

The acoustic receiver station, forming the MD, utilizes an STM32F469 Discovery kit with an NRF24L01 module, an MTI-1 IMU kit [49], and a custom-made PCB board with four microphones arranged at the vertices of two side-to-side equilateral triangles, as shown in Figure 9. The side length of this equilateral triangle is 36.80 mm. For this study, three out of the four microphones are utilized. The experimental setup includes the master station, two acoustic transmitter stations, the MD (acoustic receiver station), and an OptiTrack motion capture system [50] providing ground truth for position and orientation, as depicted in Figure 10. The master station emits a periodic signal via a wireless link, establishing a common time reference upon reception by the acoustic devices. Acoustic transmitters emit signals detected by MEMS microphones, processed to determine time-of-flight. The acoustic receiver includes an IMU and an SD card for recording IMU data and acoustic signals at regular intervals.

During lab experiments, the master station connects to a PC. A user initiates experiments of specified duration via commands relayed through the master's RF link to prepare the acoustic devices. The master initiates acoustic transmission with a predetermined RF command, triggering the receiver to record signals from all four microphones, though only three are used in this work.

Ground truth positions of the acoustic transmitters and receiver are captured using an OptiTrack motion capture sys-

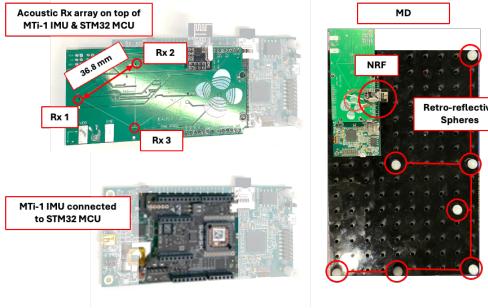


Fig. 9: Acoustic receiver

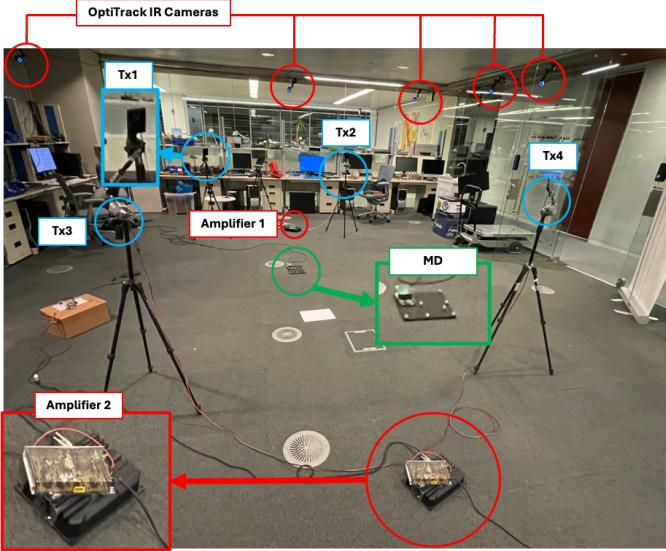


Fig. 10: Experimental setup

tem [50]. retro-reflective markers on each acoustic transmitter and the receiver ensure accurate tracking. Post-experiment, OptiTrack records are synchronized with acoustic receiver data to facilitate error calculation and further analysis.

## VII. RESULTS

### A. IMU and Acoustic Emulation Setup

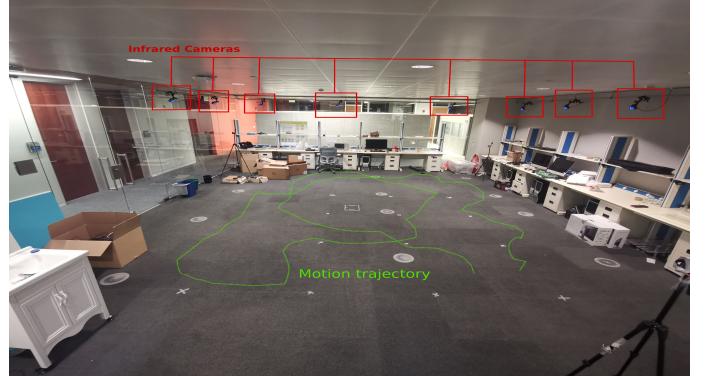
The evaluation setup consists of a  $10 \text{ m} \times 5 \text{ m} \times 3 \text{ m}$  room equipped with an OptiTrack infrared tracking system. This system offers ground truth with  $0.1 \text{ mm}$  3D position accuracy, a maximum update rate of  $250 \text{ Hz}$  [50], and a rotational error of less than  $0.05$  degrees. OptiTrack, utilizing up to sixteen infrared cameras, tracks small spheres coated with retro-reflective film to provide accurate 3D location and orientation information. Figure 11 illustrates the target tracked by the IR cameras and the motion trajectory in the experimental setup. In the first part of the results section, the system measures the 3D position and orientation of the moving target to derive acceleration and angular velocity in the body frame, effectively creating synthetic IMU measurements.

To simulate IMU data, we add additive Gaussian noise with zero mean and variances  $\sigma_a^2$  and  $\sigma_\omega^2$  to the acceleration and angular velocity, respectively. Additionally, the vertices of

an equilateral triangle are computed from the measured centroid position using Equation 23. In this mixed experimental-simulation setup, no actual IMU or acoustic beacons are used. Instead, we numerically simulate what would have been IMU or acoustic range measurements. Distances from each of the three vertices to each beacon are calculated using Euclidean distance, with additive Gaussian noise of zero mean and variance  $\sigma_d^2$  added to simulate an acoustic ranging system. In the numerical simulation setup, the side length of the equilateral triangle is set to  $20 \text{ cm}$ .



(a) Localization's Target



(b) Room equipped with high-precision infrared tracking system

Fig. 11: Mixed Experimental-Simulation setup

### B. Numerical Simulations

We evaluate the performance of the proposed algorithm by computing the Root Mean Square Error (RMSE) of the three vertices' positions. The algorithms assessed include the EKF and UKF combined with Riemannian Steepest Descent (RSD) and Riemannian Trust Region (RTR) methods. The setup employs an acoustic ranging system to determine the positions of a triangle's vertices, which then provides the centroid position and orientation of the MD. The effectiveness of fusing the estimated position and orientation is assessed as follows:

- Centroid Position Correction: We compare two systems: (1) A single-receiver acoustic positioning system, labeled “1RX” in the RMSE plots, and (2) A three-receiver acoustic positioning system, labeled “ $P_c$ -GN” in the RMSE plots, which corrects the INS using linear Kalman filter. The localization method used in these two systems is based on Gauss-Newton (GN) method.
- Centroid Position Correction: We also compare two three-receiver acoustic systems: (1) those employing Riemannian-based methods, labeled “ $P_c$ -RSD” and “ $P_c$ -RTR”, and (2) those using GN-based methods, labeled “ $P_c$ -GN”. These systems use EKF and UKF.

- Centroid Position and Orientation Correction: We compare: (1) three-receiver acoustic systems using Riemannian-based methods, labeled “EKF-RSD”, “EKF-RTR”, “UKF-RSD”, and “UKF-RTR”, and (2) GN-based methods, labeled “EKF-GN” and “UKF-GN”.

The proposed and benchmark algorithms utilize Kalman Filters (KF), EKF, and UKF to fuse position and orientation estimates from the INS with those obtained via acoustic-based positioning. Note that the input data for INS and acoustic positioning algorithms, such as acceleration, angular velocity, and acoustic ranges, are numerically generated. The acoustic-based positioning uses either; (1) Gauss-Newton (GN) method as an unconstrained-localization solver or (2) RSD and RTR as a constrained-localization solvers.

To generate test data, a person holding the target shown in Figure 11 walks around the test area at a normal walking speed (1 m/s to 1.8 m/s), with the motion trajectory illustrated in Figure 12. The “emulated” IMU update rate is fixed at 100 estimates per second for all simulations. True and estimated

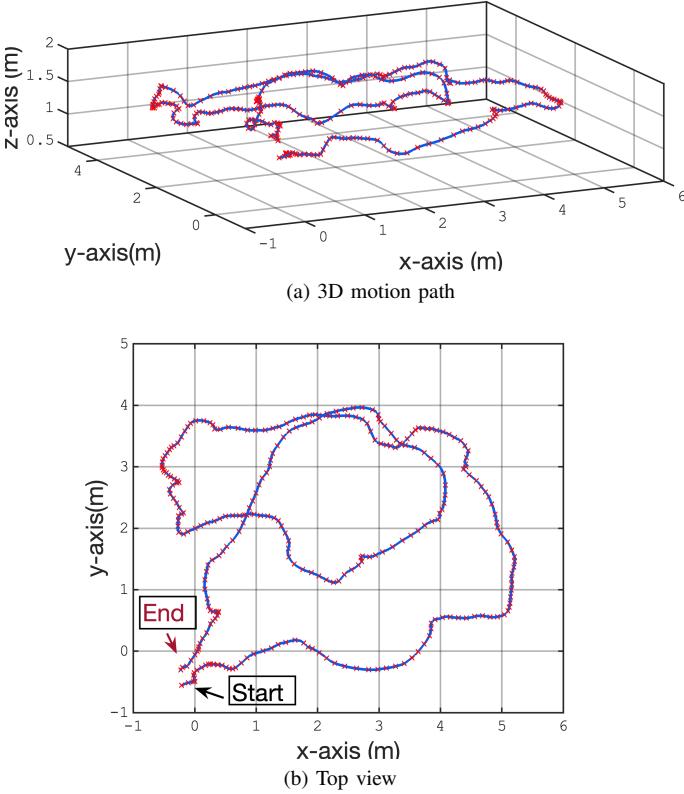


Fig. 12: Motion trajectory in emulation

Euler angles of the moving target are shown in Figure 13.

First, we evaluate the impact of varying Gaussian noise added to the measured acceleration. In this simulation, Gaussian noise added to measured angular velocity and distances have fixed standard deviations of  $\sigma_\omega = 0.5$  rad/s and  $\sigma_d = 2.5$  cm, respectively, while the standard deviation of the Gaussian noise added to measured acceleration  $\sigma_a$  ranges from 50 to 450 cm/s<sup>2</sup>. The average RMSE of the three vertices’ positions for all algorithms increases with higher  $\sigma_a$ , as shown in Figure 14. The accuracy of the estimated position and

orientation improves with three receivers. Because of the proposed orientation-correction method using the three receivers, the average RMSEs of the three vertices get reduced. The Riemannian-based Kalman filter algorithms (RSD-based and RTR-based) outperform benchmark algorithms in vertex position accuracy. The average cumulative distribution function (CDF) of the RMSEs for the three vertices is plotted in Figure 15. Algorithms that do not correct orientation using INS, i.e., those without acoustic-based orientation correction, achieve lower accuracy due to having the acceleration measured in the body frame. RSD and RTR methods perform better than GN-based methods. EKF and UKF show similar performance because of the slow movement relative to the high update rate, making the first-order approximation of EKF accurate.

Next, we assess the effect of varying the standard deviation of Gaussian noise added to the measured angular velocity  $\sigma_\omega$  from 0.4 rad/s to 0.8 rad/s while keeping the standard deviations of Gaussian noise added to measured acceleration and distances fixed at  $\sigma_a = 50$  cm/s<sup>2</sup> and  $\sigma_d = 2.5$  cm. As the angular velocity noise increases, the accuracy of vertices positions degrades for algorithms that do not correct INS-based orientation (“1RX”, “P<sub>c</sub>-GN”, “P<sub>c</sub>-RSD”, “P<sub>c</sub>-RTR”). Conversely, algorithms that use acoustic-based orientation correction maintain high accuracy despite increased angular velocity noise, as shown in Figure 16.

Figure 17 illustrates the average CDF of RMSEs at  $\sigma_a = 50$  cm/s<sup>2</sup>,  $\sigma_\omega = 0.8$  rad/s, and  $\sigma_d = 2.5$  cm, showing that orientation-correcting algorithms maintain high accuracy.

In a subsequent simulation, we evaluate performance with varying standard deviation  $\sigma_d$  of Gaussian noise added to measured distances from 3 cm to 7 cm while fixing the standard deviations of noise in measured acceleration and angular velocity at  $\sigma_a = 250$  cm/s<sup>2</sup> and  $\sigma_\omega = 1$  rad/s. At very low noise levels, the trilateration-based method performs similarly to Riemannian-optimization-based methods. However, as noise increases, benchmark methods degrade significantly compared to the proposed methods, as shown in Figure 18. Moreover, we notice that the degradation in the accuracy of the algorithms that correct the INS-based orientation is higher than the degradation in the algorithms that do not correct the orientation. This is expected since the accuracy of the estimated orientation using the acoustic-based method degrades with increasing the noise added to the measured distances.

Figure 19 shows the average CDF of RMSE for the three vertices at  $\sigma_a = 250$  cm/s<sup>2</sup>,  $\sigma_\omega = 1$  rad/s, and  $\sigma_d = 7$  cm. RTR- and RSD-based methods with orientation correction outperform all benchmark methods, with over 85% of position estimates having less than 10 cm RMSE, compared to less than 60% for GN-based methods with orientation correction and less than 35% without orientation correction. Finally, we assess the impact of varying the acoustic system’s update rate from 5 to 50 estimates per second, with fixed Gaussian noise standard deviations ( $\sigma_a = 50$  cm/s<sup>2</sup>,  $\sigma_\omega = 0.5$  rad/s and  $\sigma_d = 2.5$  cm.). All methods show improved positioning accuracy with higher update rates, though at different rates, as shown in Figure 20.

Orientation-correction methods consistently outperform oth-

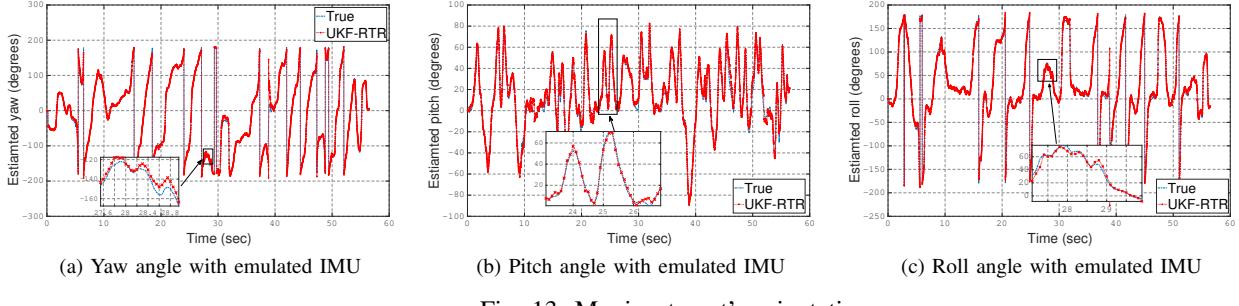
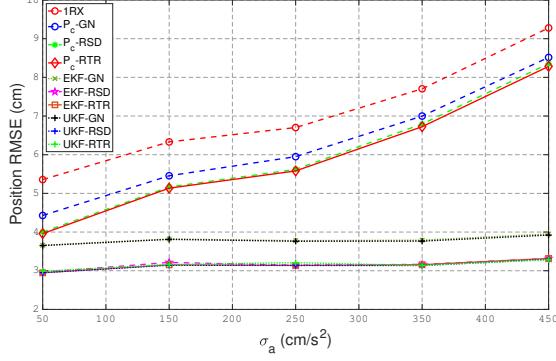
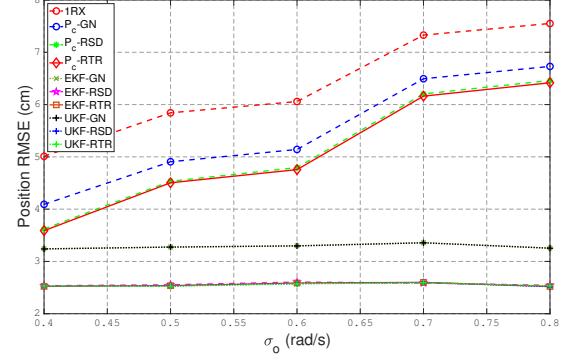
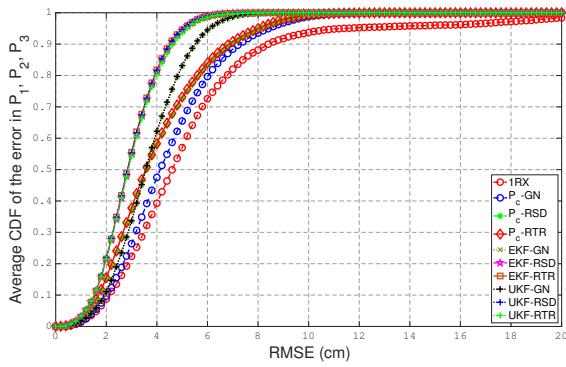
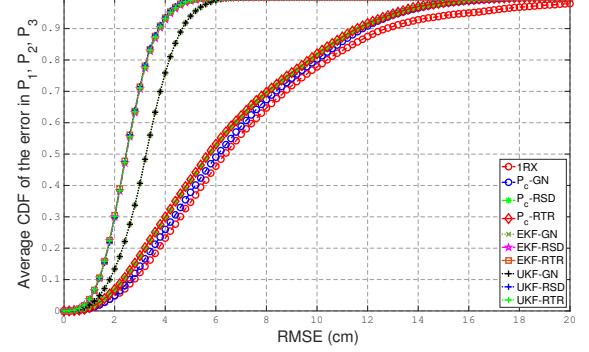


Fig. 13: Moving target's orientation

Fig. 14: RMSE vs  $\sigma_a$ Fig. 16: RMSE vs  $\sigma_\omega$ Fig. 15: Average CDF at  $\sigma_a = 50 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.5 \text{ rad/s}$ ,  $\sigma_d = 2.5 \text{ cm}$ Fig. 17: Average CDF at  $\sigma_a = 50 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.8 \text{ rad/s}$ ,  $\sigma_d = 2.5 \text{ cm}$ 

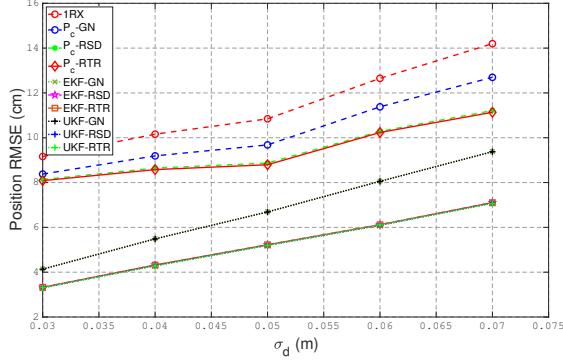
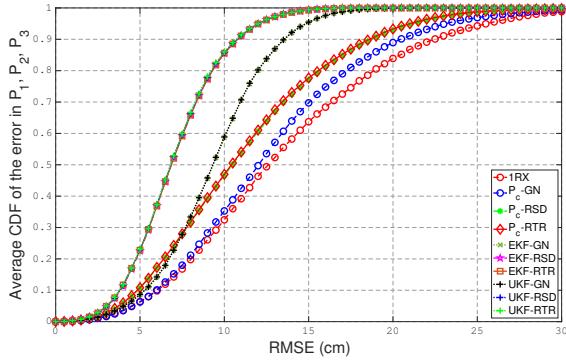
ers under all scenarios, as depicted in Figure 21. Figure 22 shows the error of estimated Euler angles for GN-based and RTR-based algorithms at  $\sigma_a = 250 \text{ cm/s}^2$ ,  $\sigma_\omega = 1 \text{ rad/s}$ , and  $\sigma_d = 3 \text{ cm}$ . The RMSE of estimated Euler angles is detailed in Table I. During the target's movement, rotation occurs around all three axes (yaw, pitch, and roll), as seen in Figure 13. Algorithms without orientation correction suffer from error accumulation, thus their RMSE in Euler angle estimation is not presented. RTR-based and RSD-based methods show similar performance. Consequently, only RTR-based orientation estimation results are presented. The RTR-based method provides lower RMSE for all three angles, consistent with the RMSE results for vertices positions.

### C. NLOS Scenarios

To simulate NLOS scenarios, distances between the three receivers and the first beacon are generated based on a uniform distribution over a 3-second period, simulating a beacon

Algorithm	Yaw RMSE	Pitch RMSE	Roll RMSE
EKF-GN	10.85	4.48	10.66
UKF-GN	10.87	4.48	10.68
EKF-RTR	7.80	4.26	7.60
UKF-RTR	7.79	4.14	7.63

TABLE I: RMSE in degrees of the estimated Euler angles

Fig. 18: RMSE vs  $\sigma_d$ Fig. 19: Average CDF at  $\sigma_a = 250 \text{ cm/s}^2$ ,  $\sigma_\omega = 1 \text{ rad/s}$ ,  $\sigma_d = 7 \text{ cm}$ 

blockage during this time. We will compare two situations. In the first situation, we input the distances between the three receivers and four beacons, along with synthetic IMU measurements, into the EKF and UKF algorithms and assess the accuracy of the estimated positions. In the second situation, we first determine whether the estimated distance is from an LOS or NLOS condition, using various NLOS detection algorithms [51], [52]. Subsequently, we apply the EKF and UKF algorithms only to the distances identified as LOS measurements. In the period of NLOS distances, the positioning rely completely on the synthetic IMU data. Figure 23 illustrates

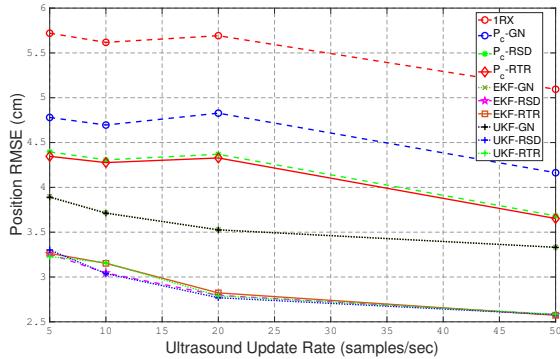
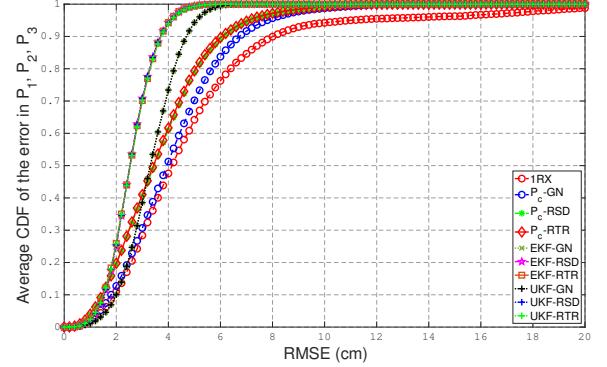
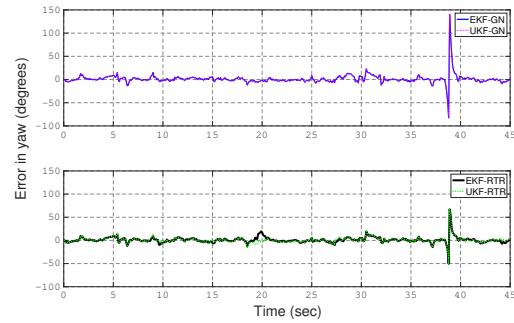
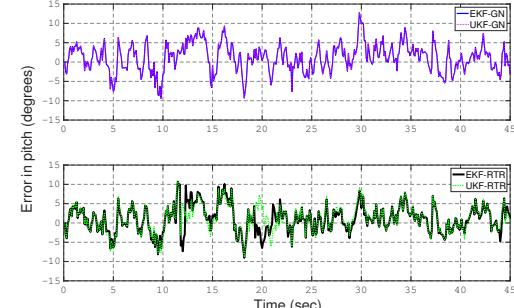


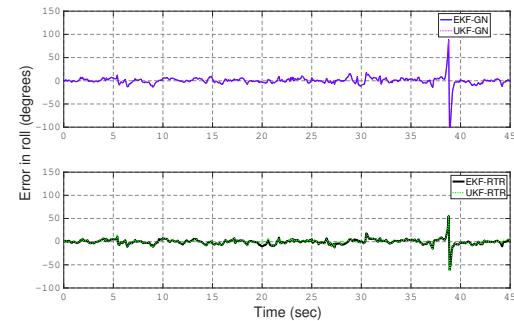
Fig. 20: RMSE vs acoustic system update rate

Fig. 21: Average CDF at  $\sigma_a = 50 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.5 \text{ rad/s}$ ,  $\sigma_d = 2.5 \text{ cm}$  and acoustic update rate of 50 Hz

(a) Error in yaw angle in the simulation



(b) Error in pitch angle in the simulation



(c) Error in roll angle in the simulation

Fig. 22: Error in the estimated Euler angles

the estimated position and motion trajectory, while Figure 24 presents the average CDF of RMSE for the three vertices, with parameters set at  $\sigma_a = 5 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.25 \text{ rad/s}$ , and

$\sigma_d = 2.5$  cm. The proposed algorithms maintain their accuracy in NLOS scenarios, provided that NLOS detection is effective. Conversely, performance degrades in algorithms that do not incorporate NLOS detection.

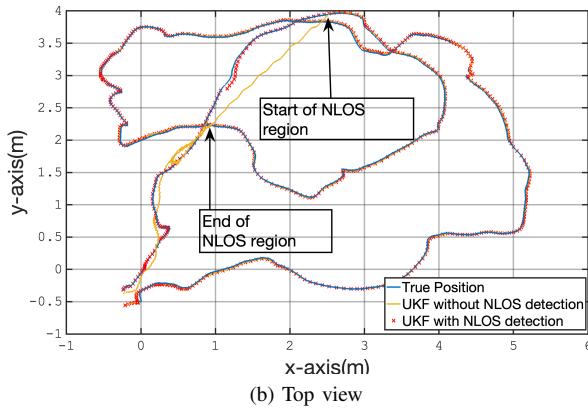
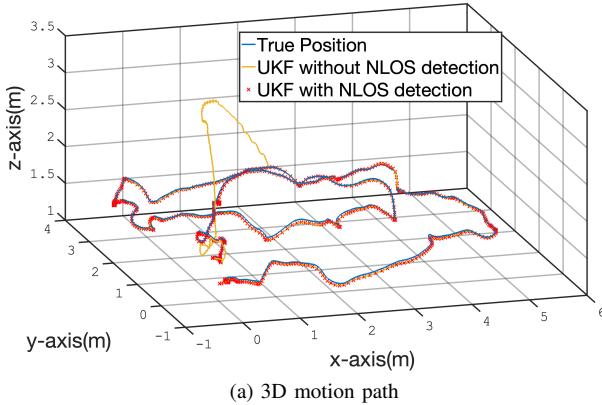


Fig. 23: Motion trajectory in NLOS scenario

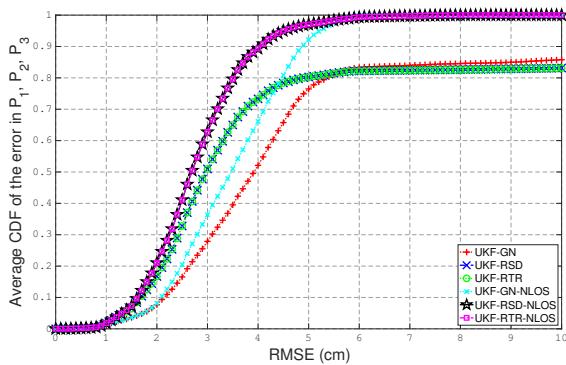


Fig. 24: Average CDF at  $\sigma_a = 5 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.25 \text{ rad/s}$ , and  $\sigma_d = 2.5 \text{ cm}$

#### D. Semi-Experimental Results

In this semi-experimental setup, we utilized real IMU measurements from an iPhone combined with synthetic distance measurements obtained using the OptiTrack system. Markers, as shown in Figure 11(a), were attached to the iPhone to provide ground-truth position and orientation data through

the OptiTrack system. We then applied both proposed and benchmark methods to estimate the position and orientation of the iPhone. The RMSE of the three vertices of an equilateral triangle was calculated. The motion trajectory for this experiment is depicted in Figure 25, and the true and estimated Euler angles of the phone are shown in Figure 26.

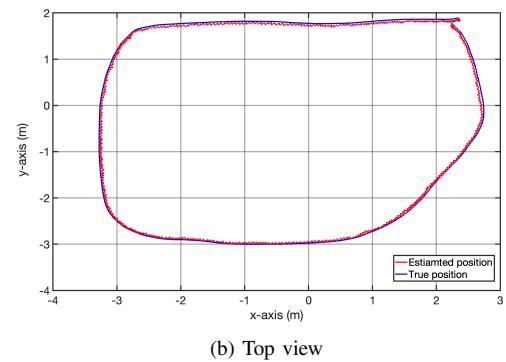
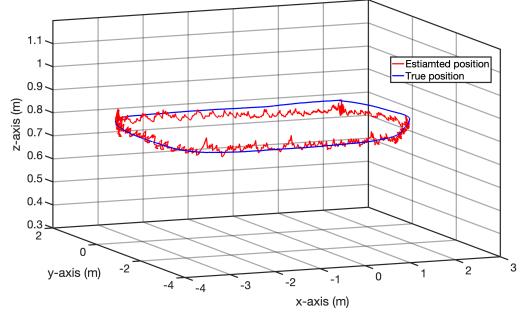
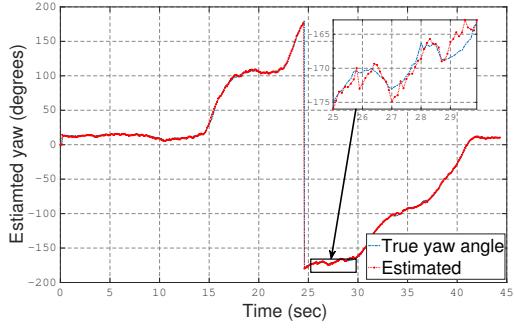


Fig. 25: iPhone's Experiment: Motion trajectory

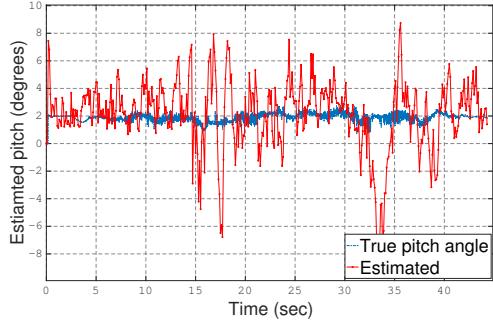
The noise characteristics of the iPhone's gyroscope and accelerometer were as follows: the gyroscope noise had standard deviations of 0.11 rad/s (x-axis), 0.05 rad/s (y-axis), and 0.07 rad/s (z-axis). The accelerometer noise had standard deviations of  $53 \text{ cm/s}^2$  (x-axis),  $62 \text{ cm/s}^2$  (y-axis), and  $49 \text{ cm/s}^2$  (z-axis). These parameters were estimated from the iPhone's IMU measurements and the synthetic IMU data provided by the OptiTrack system.

The experimental results are consistent with the previous numerical findings. The average RMSE of the three vertices' positions for all algorithms increases with the standard deviation of the distance measurements  $\sigma_d$ , as shown in Figure 27. Additionally, using three receivers rather than a single receiver improves the accuracy of the estimated position and orientation. The benefit of the proposed orientation-correction method using three receivers is evident, with Riemannian-based Kalman filtering methods outperforming benchmark algorithms in terms of vertices position accuracy.

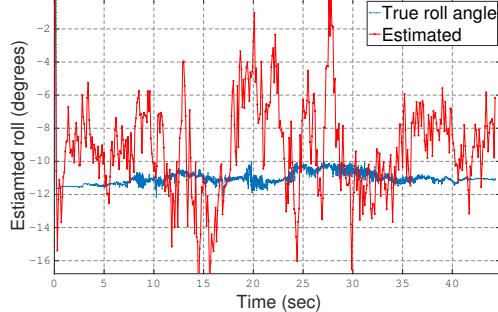
The average cumulative distribution function (CDF) of the RMSEs for the three vertices is presented in Figure 28. Algorithms that do not correct the orientation obtained from INS exhibit lower accuracy, as expected. Furthermore, RTR and RSD methods outperform GN-based methods. The performance of EKF and UKF is similar due to the slow



(a) Yaw angle during iPhone experiment

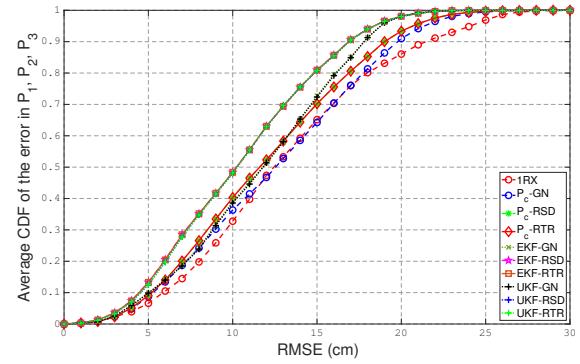


(b) Pitch angle during iPhone experiment



(c) Roll angle during iPhone experiment

Fig. 26: iPhone's Experiment: Orientation of the phone

Fig. 28: iPhone's Experiment: Average CDF at  $\sigma_a = 55 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.08 \text{ rad/s}$ ,  $\sigma_d = 12 \text{ cm}$ 

Algorithm	Yaw RMSE	Pitch RMSE	Roll RMSE
EKF-GN	3.49	2.07	9.70
UKF-GN	3.47	2.03	9.34
EKF-RTR	1.35	2.52	3.34
UKF-RTR	1.35	2.52	3.34

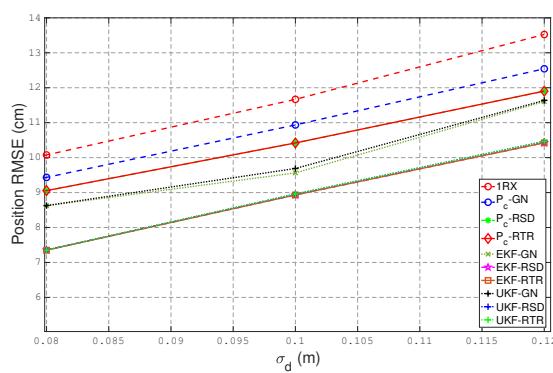
TABLE II: RMSE in degrees of the estimated Euler angles

provides the RMSE of the estimated Euler angles. During the experiment, the rotation primarily occurred around the yaw axis, as illustrated in Figure 26. For this reason, we present only the results for GN-based and RTR-based orientation estimation. The RTR-based method demonstrates lower RMSE in estimating yaw and roll angles and comparable RMSE for the pitch angle. This outcome is consistent with the evaluation of the RMSE of the three vertices, reflecting the accuracy of orientation estimation.

#### E. Experimental Results: MTi-1 with Acoustic Transmitters/ Receivers

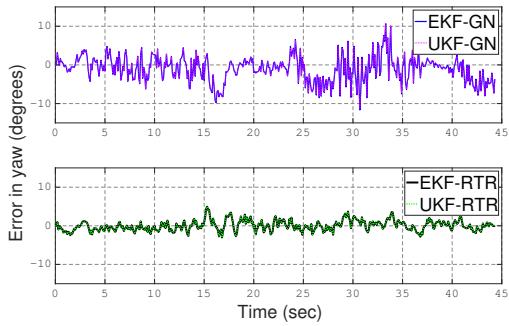
In this experiment, four acoustic transmitters (beacons) were positioned at known locations, while three acoustic receivers, arranged in an equilateral triangle on a printed circuit board, were fixed to an MD as shown in Figure 10. The MD was equipped with an MTi-1 IMU, which includes a triaxial accelerometer and a triaxial gyroscope [49]. The device was moved around the room, as depicted in Figure 30. The IMU sampled at 100 Hz, and the acoustic system measured distances between the transmitters and receivers at 5 Hz. Synchronization between the acoustic transmitters and receivers was achieved using an RF signal. The experiment lasted 30 seconds, and the recorded data were processed offline.

Three variations of Kalman filters were evaluated as follows. A linear Kalman filter (LKF) where the state vector includes the centroid position and velocity of the MD. The observed variable consists of the positions of the three vertices of the equilateral triangle, which are estimated using the GN and RTR methods. The centroid position is obtained by averaging the three vertex positions and then used in the LKF. The

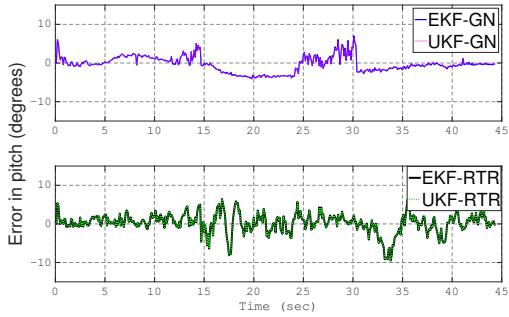
Fig. 27: iPhone's Experiment: RMSE vs  $\sigma_d$ 

movement relative to the high update rate, which makes the first-order approximation of EKF accurate.

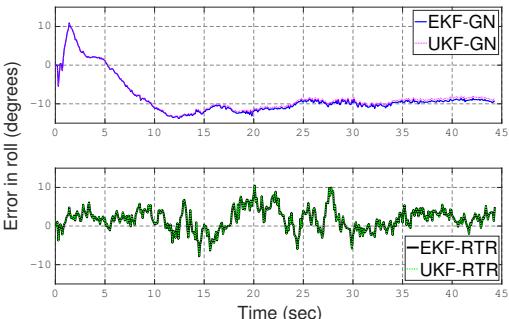
Figure 29 shows the error in the estimated Euler angles for both GN-based and RTR-based algorithms, and Table II



(a) Yaw angle during iPhone experiment



(b) Pitch angle during iPhone experiment

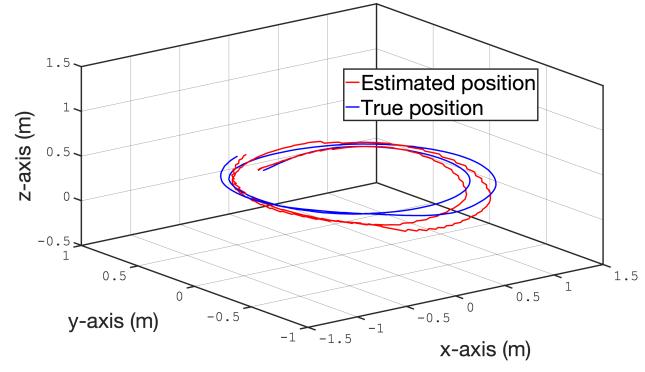


(c) Roll angle during iPhone experiment

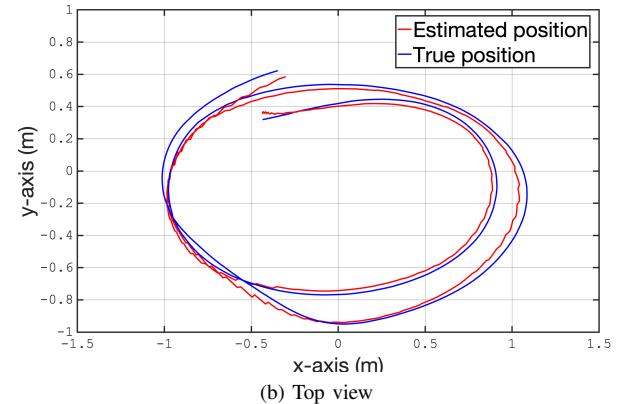
Fig. 29: iPhone's experiment: Error in the estimated Euler angles

other two variations of the Kalman filter were implemented using EKF and UKF where the state variable includes the centroid position, velocity, and the vector-form of the orientation matrix. In this case, the observed variable is the vertices' positions, which are utilized to estimate both the orientation matrix and the centroid position.

The RMSE of the estimated positions, averaged over the three vertices and across time, is presented in Table III. The Riemannian-based UKF and EKF demonstrate improved accuracy compared to GN-based methods. This improvement is also reflected in the CDF plot shown in Figure 31, where more than 80% of positions estimated using Riemannian-based UKF and EKF have an error of less than 9 cm, whereas less than 80% of positions estimated using GN-based UKF and EKF have an error of less than 13 cm.

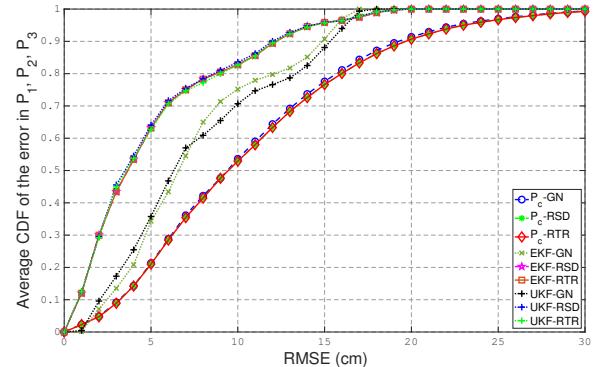


(a) 3D motion path



(b) Top view

Fig. 30: Experimental evaluation: Motion trajectory

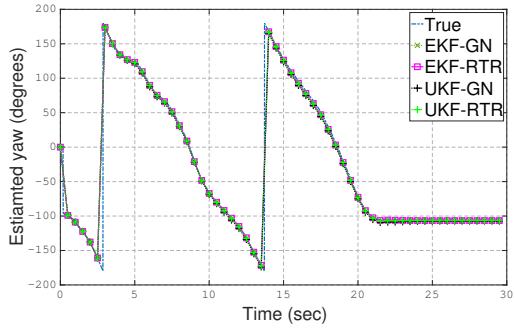
Fig. 31: Experimental evaluation: Average CDF at  $\sigma_a = 55 \text{ cm/s}^2$ ,  $\sigma_\omega = 0.08 \text{ rad/s}$ ,  $\sigma_d = 12 \text{ cm}$ 

Algorithm	RMSE (cm)
LKF-GN	10.44
LKF-RTR	10.60
EKF-GN	7.52
UKF-GN	7.67
EKF-RTR	5.09
UKF-RTR	5.00

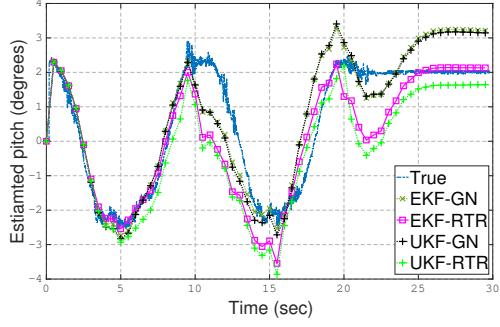
TABLE III: RMSE in cm averaged over the three receivers

The accuracy of the estimated Euler angles for the UKF and EKF methods was also evaluated. The estimated angles,

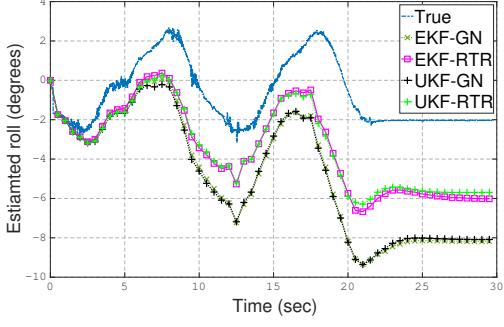
along with the ground truth, are shown in Figure 32.



(a) Yaw angle during experiment



(b) Pitch angle during experiment



(c) Roll angle during experiment

Fig. 32: Experimental evaluation: orientation of the MD

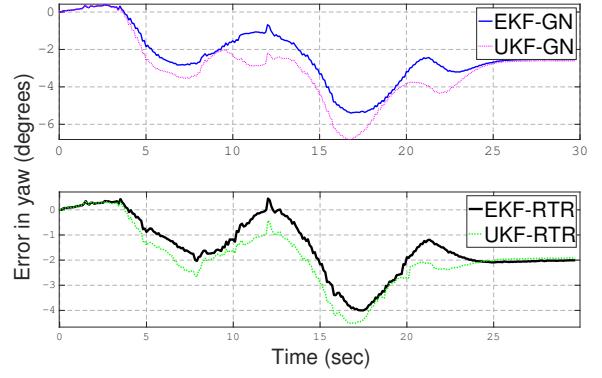
The RMSE of the estimated angles is detailed in Table IV. Riemannian-based UKF and EKF show slight improvements over GN-based methods, as illustrated in the error plots in Figure 33.

Algorithm	Yaw RMSE	Pitch RMSE	Roll RMSE
EKF-GN	4.07	0.67	2.40
UKF-GN	4.10	0.69	3.00
EKF-RTR	2.68	0.65	1.55
UKF-RTR	2.64	0.86	1.91

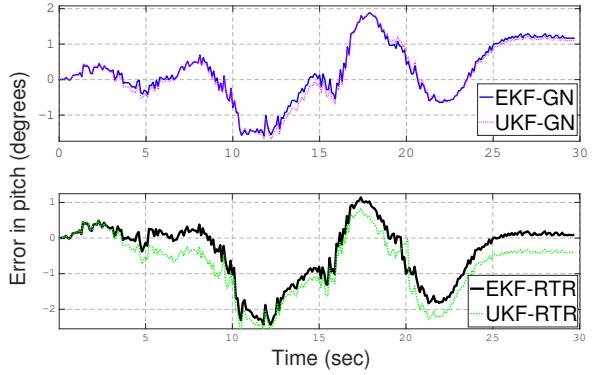
TABLE IV: RMSE in degrees of the estimated Euler angles

### VIII. CONCLUSION

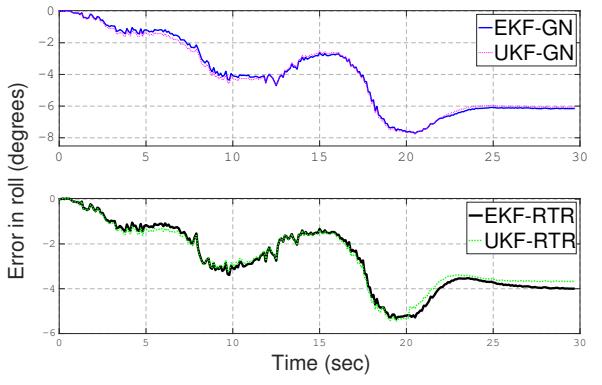
This paper introduced an enhanced implementation of the Extended Kalman Filter (EKF) and Unscented Kalman Filter



(a) Error in yaw angle



(b) Error in pitch angle



(c) Error in roll angle

Fig. 33: Experimental evaluation: error in the estimated Euler angles

(UKF) for high-precision estimation of the 3D position and orientation of a moving target. The proposed filters integrate measurements from two navigation systems: an Inertial Navigation System (INS) based on IMU data and an acoustic navigation system, leveraging our previous work on Riemannian optimization [5]. The output of these filters is projected onto an isosceles triangle manifold using a novel retraction algorithm.

The results demonstrate that using an array of acoustic receivers significantly improves accuracy compared to a single receiver. Furthermore, correcting the INS orientation with the receiver array results in substantial accuracy gains over benchmark methods. The Riemannian-based methods, which utilize the geometry of the receiver array as constraints, outperform

the Gauss-Newton-based solver in localization performance.

In summary, the proposed filter exhibits superior performance in all signal-to-noise ratio (SNR) scenarios, achieving better positioning and orientation accuracy than the benchmark methods.

Future research will focus on implementing the UKF on the isosceles triangle manifold. This involves deriving the necessary tools to apply the unscented transform within the context of Riemannian manifolds [53].

## REFERENCES

- [1] Y. Sun, W. Wang, L. Mottola, J. Zhang, R. Wang, and Y. He, "Indoor drone localization and tracking based on acoustic inertial measurement," *IEEE Transactions on Mobile Computing*, 2023.
- [2] D. Feng, C. Wang, C. He, Y. Zhuang, and X.-G. Xia, "Kalman-filter-based integration of imu and uwb for high-accuracy indoor positioning and navigation," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3133–3146, 2020.
- [3] F. Khelifi, A. Bradai, A. Benslimane, P. Rawat, and M. Atri, "A survey of localization systems in internet of things," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 761–785, 2019.
- [4] D. Titterton, J. L. Weston, and J. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.
- [5] M. H. AlSharif, A. Douik, M. Ahmed, T. Y. Al-Naffouri, and B. Hassibi, "Manifold optimization for high-accuracy spatial location estimation using ultrasound waves," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5078–5093, 2021.
- [6] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [7] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.
- [8] Ç. Yüzbasioglu and B. Barshan, "Improved range estimation using simple infrared sensors without prior knowledge of surface characteristics," *Measurement Science and Technology*, vol. 16, no. 7, p. 1395, 2005.
- [9] M.-C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux, "Laser ranging: a critical review of usual techniques for distance measurement," *Optical engineering*, vol. 40, no. 1, pp. 10–19, 2001.
- [10] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [11] I. Cappelli, F. Carli, A. Fort, M. Intravaia, F. Micheletti, G. Peruzzi, and V. Vignoli, "Enhanced visible light localization based on machine learning and optimized fingerprinting in wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.
- [12] A. Gabbielli, J. Bordoy, W. Xiong, G. K. Fischer, T. Schaechtle, J. Wendeberg, F. Höflinger, C. Schindelhauer, and S. J. Rupitsch, "Rails: 3-d real-time angle of arrival ultrasonic indoor localization system," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–15, 2022.
- [13] L. Santoro, M. Nardello, D. Brunelli, and D. Fontanelli, "Uwb-based indoor positioning system with infinite scalability," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.
- [14] M. Cypriani, F. Lassabe, P. Canalda, and F. Spies, "Open wireless positioning system: A wi-fi-based indoor positioning system," in *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*. IEEE, 2009, pp. 1–5.
- [15] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 483–494.
- [16] M. H. AlSharif, M. Saad, M. Siala, M. Ahmed, and T. Y. Al-Naffouri, "Range estimation of a moving target using ultrasound differential Zadoff-Chu codes," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [17] H. Wang, C. Xue, Z. Wang, L. Zhang, X. Luo, and X. Wang, "Smartphone-based pedestrian nlos positioning based on acoustics and imu parameter estimation," *IEEE Sensors Journal*, 2022.
- [18] L.-F. Shi, B.-L. Feng, Y.-F. Dai, G.-X. Liu, and Y. Shi, "Pedestrian indoor localization method based on integrated particle filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–10, 2023.
- [19] G. T. Lee, S. B. Seo, and W. S. Jeon, "Indoor localization by kalman filter based combining of uwb-positioning and pdr," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–6.
- [20] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.
- [21] Q. Fan, B. Sun, Y. Sun, Y. Wu, and X. Zhuang, "Data fusion for indoor mobile robot positioning based on tightly coupled ins/uwb," *The Journal of Navigation*, vol. 70, no. 5, pp. 1079–1097, 2017.
- [22] Y. Zhong, T. Liu, B. Li, L. Yang, and L. Lou, "Integration of uwb and imu for precise and continuous indoor positioning," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. IEEE, 2018, pp. 1–5.
- [23] B. Li, Z. Hao, and X. Dang, "An indoor location algorithm based on kalman filter fusion of ultra-wide band and inertial measurement unit," *AIP Advances*, vol. 9, no. 8, p. 085210, 2019.
- [24] D. Gualda, M. C. Pérez-Rubio, J. Ureña, S. Pérez-Bachiller, J. M. Villadangos, Á. Hernández, J. J. García, and A. Jiménez, "Locate-us: Indoor positioning for mobile devices using encoded ultrasonic signals, inertial sensors and graph-matching," *Sensors*, vol. 21, no. 6, p. 1950, 2021.
- [25] Y. Xu, Y. S. Shmaliy, C. K. Ahn, T. Shen, and Y. Zhuang, "Tightly coupled integration of ins and uwb using fixed-lag extended ukf smoothing for quadrotor localization," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1716–1727, 2020.
- [26] L. Xie, J. Tian, G. Ding, and Q. Zhao, "Holding-manner-free heading change estimation for smartphone-based indoor positioning," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–5.
- [27] P. Nguyen, T. Akiyama, H. Ohashi, G. Nakahara, K. Yamasaki, and S. Hikaru, "User-friendly heading estimation for arbitrary smartphone orientations," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–7.
- [28] J. Bravo, E. P. Herrera, and D. A. Sierra, "Comparison of step length and heading estimation methods for indoor environments," in *2017 IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. IEEE, 2017, pp. 1–4.
- [29] A. Manos, I. Klein, and T. Hazan, "Gravity direction estimation and heading determination for pedestrian navigation," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2018, pp. 206–212.
- [30] S. K. Hong and Y.-s. Ryuh, "Heading measurements for indoor mobile robots with minimized drift using a mems gyroscopes," in *Robot localization and map Building*. IntechOpen, 2010.
- [31] R. Zhou, "Pedestrian dead reckoning on smartphones with varying walking speed," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.
- [32] M. H. Afzal, V. Renaudin, and G. Lachapelle, "Magnetic field based heading estimation for pedestrian navigation environments," in *2011 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2011, pp. 1–10.
- [33] M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2012, pp. 1–9.
- [34] S. Y. Cho and C. G. Park, "Mems based pedestrian navigation system," *The Journal of Navigation*, vol. 59, no. 1, pp. 135–153, 2006.
- [35] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Journal of Global Positioning Systems*, vol. 3, no. 1–2, pp. 273–279, 2004.
- [36] B. Shin, J. H. Lee, H. Lee, E. Kim, J. Kim, S. Lee, Y.-s. Cho, S. Park, and T. Lee, "Indoor 3d pedestrian tracking algorithm based on pdr using smartphone," in *2012 12th International Conference on Control, Automation and Systems*. IEEE, 2012, pp. 1442–1445.
- [37] W. Ma, J. Wu, C. Long, and Y. Zhu, "Hiheading: smartphone-based indoor map construction system with high accuracy heading inference," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, 2015, pp. 172–177.
- [38] X. Yang, B. Huang, and Q. Miao, "A step-wise algorithm for heading estimation via a smartphone," in *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 2016, pp. 4598–4602.

- [39] A. Poulose, B. Senouci, and D. S. Han, "Performance analysis of sensor fusion techniques for heading estimation using smartphone sensors," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 369–12 380, 2019.
- [40] M. H. AlSharif, M. Saad, M. Siala, T. Ballal, H. Boujemaa, and T. Y. Al-Naffouri, "Zadoff-Chu coded ultrasonic signal for accurate range estimation," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 1250–1254.
- [41] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [42] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.
- [43] P.-A. Absil, C. G. Baker, and K. A. Gallivan, "Trust-region methods on riemannian manifolds," *Foundations of Computational Mathematics*, vol. 7, no. 3, pp. 303–330, 2007.
- [44] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [45] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.
- [46] STMicroelectronics. (2024) Stm32f469 discovery kit. [Online]. Available: <https://www.st.com/en/evaluation-tools/32f469idiscovery.html>
- [47] N. Semiconductor. (2024) nrf24l. [Online]. Available: <https://www.nordicsemi.com/Products/nRF24-series>
- [48] Sony. (2024) Xm-n1004 stereo amplifier. [Online]. Available: <https://www.sony.com/en-sa/electronics/car-amplifiers/xm-n1004>
- [49] Movella. (2024) Mt-1 datasheet. [Online]. Available: [https://www.xsens.com/hubfs/Downloads/Leaflets/MTI-1.pdf?\\_\\_hstc=157421285.3f2f86467b8e14374d4860c1552d523b.1715494986238.1719859798440.1721973471183.3&\\_\\_hssc=157421285.2.1721973471183&\\_\\_hsfp=3563223768](https://www.xsens.com/hubfs/Downloads/Leaflets/MTI-1.pdf?__hstc=157421285.3f2f86467b8e14374d4860c1552d523b.1715494986238.1719859798440.1721973471183.3&__hssc=157421285.2.1721973471183&__hsfp=3563223768)
- [50] <https://optitrack.com/cameras/primex-41/>. (2021) Optitrack system overview. [Online]. Available: <https://optitrack.com/cameras/primex-41/>
- [51] M. H. AlSharif, M. Ahmed, A. Felemban, A. Zayat, A. Muqabel, M. Masood, and T. Y. Al-Naffouri, "Robust 2d indoor positioning algorithm in the presence of non-line-of-sight signals," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1802–1806.
- [52] F. Zhu, K. Yu, Y. Lin, C. Wang, J. Wang, and M. Chao, "Robust los/nlos identification for uwb signals using improved fuzzy decision tree under volatile indoor conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2023.
- [53] S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented Kalman filtering on riemannian manifolds," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 1, pp. 103–120, 2013.