A mobile Wi-Fi and IMU head-tracking system for auralization in XR

Yousef Masa'd
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
yousef.masad@student.kuleuven.be

Giulio Vitolo
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
giulio.vitolo@student.kuleuven.be

Mateo Sakr
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
mateo.sakr@student.kuleuven.be

Dunia Tornila Jichi
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
dunia.tornilajichi@student.kuleuven.be

Yusuf Hussein
Faculty of Engineering Technology
KU Leuven
Leuven, Belgium
yusuf.hussein@student.kuleuven.be

Valerio Lorenzoni
Department of Electrical Engineering (ESAT-STADIUS)
KU Leuven
Leuven, Belgium
valerio.lorenzoni@esat.kuleuven.be

Toon van Waterschoot
Department of Electrical Engineering (ESAT-STADIUS)
KU Leuven
Leuven, Belgium
toon.vanwaterschoot@esat.kuleuven.be

***Abstract*—Head-Tracking systems play a crucial role in various fields such as augmented reality, gaming, and immersive audio. It is therefore essential for these systems to be as accurate as possible, in order to provide immersive experiences for the users. Specifically in the context of spatial audio, knowledge of the positioning of the head allows the alignment of auditory experiences with visual stimuli. One example of that would be in VR environments, where sound sources must correspond to visual cues to create a realistic 3-D audio experience. The current problem with most of the state-of-the-art solutions, is that they are often complex, expensive, and computationally intensive. This ends up limiting their accessibility. This project aims to tackle these problems by combining several signal processing algorithms to enhance a cost-effective, head-tracking system. We explore multiple approaches to estimating position and orientation, and we present a collected and labelled dataset of 110 minutes of motion. Our method achieves average error of 2.2m and 39 degrees on our collected dataset.**

***Index Terms*—auralization, extended reality, head tracking**

I. Introduction

Extended Reality (XR), encompassing technologies for virtual (VR), augmented (AR), and mixed reality (MR), has found its way to numerous applications in business, education, industry, and culture [1]. While the development of early XR systems was largely focused on the rendering of visual and haptic cues, auralization, i.e., the rendering of auditory cues, has recently received much attention in the audio and XR research communities [2]. Most of the research efforts in this area have been devoted to auralization using hearables (i.e., headphones or headsets with on-board processing capabilities [3]), commonly referred to as binaural auralization or rendering, which is also the scope of this paper. Commercially available solutions to the binaural auralization problem are currently limited to three-degrees-of-freedom (3DoF) approaches. These approaches allow for rotational head movements of the listener around each of the 3-D Cartesian coordinate axes (pitch, yaw, and roll) but not for a translational movement of the listener. The step towards six-degrees-of-freedom (6DoF) approaches, in which also translational listener movements along each of the 3-D Cartesian coordinate axes (surge, strafe, and elevation) are allowed [4], has been the topic of numerous recent research efforts and is expected to create a considerable economic and socio-economic leverage effect in XR applications.

Early approaches to 6DoF auralization have focused on binaural auralization of virtual sources without attempting to render the appropriate room acoustics [5], [6]. A key element of such approaches is to accurately render the acoustic scattering of the impinging sound wave that occurs at the ear pinna, head, and torso of the listener, commonly represented by left- and right-ear pairs of head-related transfer functions (HRTFs). Auralization for dynamic observers then consists in filtering the anechoic virtual source signal with a pre-measured HRTF pair that most closely matches the instantaneous distance and direction of arrival (DoA) of the virtual source relative to the listener's ears. Commonly, HRTF measurements are available only for a sparse set of distance and DoA values, and HRTF

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven internal funds C3/23/056, FWO SBO Project S005525N, and FWO Research Project G0A0424N.

interpolation is required for dynamic auralization [7], [8].

More recent methods include auralization of room acoustics, and can roughly be classified into three categories. Model-based methods render virtual sources with virtual room acoustics, based on a room model that is inferred from a simulated rather than a real acoustic environment. These methods are relevant in virtual reality and computer games, but do not allow for auralization of virtual sources in real environments. It is common to simulate early room reflections differently from late reverberation, and to adapt only the early reflections model to accommodate for the dynamic listener position while keeping the late reverberation model static [9], [10]. Measurement-based methods exploit the linear time-invariant behavior of room acoustics by measuring source-to-observer responses and rendering these measured responses by convolution with linear filters. Binaural room impulse response (BRIR) measurements represent the complete response from the source to the listener's ears, but their excessive length and their distance and DoA dependency make them less suitable for dynamic listening scenarios. Modifications of the early part of the BRIR to accommodate for listener movements have been explored for simple room geometries [11], [12]. To circumvent the DoA dependency and allow for simple rendering despite head rotations of the listener, measurements using ambisonic microphones have become more widespread and interpolation strategies have been developed to cope with translational listener movements [13]–[18]. Hybrid methods rely on a parametric room acoustics model, the parameters of which are estimated from room response measurements and modified according to the listener's position and head orientation. The parametrization is typically more compact than the set of measured responses, which makes hybrid methods computationally more interesting than measurement-based methods. Various room parametrizations have been explored in literature, including the plane-wave and spherical-wave decomposition [19]–[21], as well as spatial room impulse response decompositions [22], [23].

In all these various 6DoF auralization methods, the dynamic listener position and head orientation needs to be estimated and tracked over time, either to select the most appropriate HRTF or BRIR from a pre-measured database, or to provide positional estimates to a sound field interpolation algorithm. Head tracking and positioning is usually achieved by equipping the hearables device with an inertial measurement unit (IMU). The task of IMU-based head tracking and positioning is a decades old task with significant literature exploring many different methods and solutions. A large portion of existing literature exploits knowledge of the system's behavioral model and multiple measurement sensors to implement Kalman filtering schemes [24], [25]. More recent methods also include step detection and counting methods in order to reduce drift issues in dead reckoning applications [26], [27]. Finally, with the modern advancements in artificial intelligence, machine learning methods have been implemented to create end-to-end solutions for position and orientation tracking using data-driven models and large labelled datasets. In addition to IMU data, many position tracking systems exploit the ubiquity of WLAN/Wi-Fi to incorporate signal strength information into their system for enhanced positioning accuracy [28], [29].

In this paper, an IMU and Wi-Fi based system for dynamic listener positioning and head tracking is implemented and evaluated in a 6DoF auralization XR use case. Moreover, we propose the use of an additional atmospheric pressure sensor to estimate and track the listener's altitude, and we show how the information acquired by this sensor along with prior knowledge of the geometry of the listening environment can be fused into the head tracking and positioning algorithm to further increase its positioning accuracy.

The paper is organised as follows. In Section II we provide details on the XR system and use case considered in this paper, and we elaborate on the collection of the data used throughout the paper. In Section III we introduce various methods for listener positioning, based on IMU, Wi-Fi and barometer data, as well as novel Kalman filtering schemes in which the sensor data are fused with prior information about the geometry of the environment. Section IV contains implementation details for positioning as well as head tracking. In Section V we present experimental results, which are discussed in Section VI. Finally, Section VII concludes the paper.

II. System overview

A. XR use case environment

The XR use case considered throughout this paper consists of a sound art installation specifically designed for the KU Leuven Group T Campus. This installation has been conceptualized in [30] and provides an augmented auditory experience for listeners walking along the campus. The listener, wearing a pair of wireless headphones, is presented with various virtual sound sources along their walk, which are auralized using the acoustics of the real environment. To this end, multichannel room impulse response (RIR) measurements have been performed along the walking path. During auralization, the RIR corresponding to the measurement position closest to the actual listener position is convolved with a pair of generic HRTFs selected based on the head orientation to yield a set of BRIRs, which are then convolved with the virtual sound source signal.

A peculiar feature of this environment is that the envisaged walking path has the shape of a 3-D helix, as shown in Fig. 1, for which a closed-form equation exists. This prior knowledge on the geometry of the XR use case environment will be included in the proposed positioning algorithm.

B. Sensors and processor module

A self-designed, battery-driven module consisting of a mobile processor and multiple sensors is attached on top of the headphones worn by the listener. The processor is an Arduino MKR 1010 including a Wi-Fi module, whereas the sensors consist of an IMU and a barometer. The IMU used in our module is the Bosch BNO055, a 9-axis intelligent absolute orientation sensor, that integrates a gyroscope, accelerometer, and magnetometer with an onboard microcontroller that fuses

Fig. 1. The XR use case environment considered in this paper, featuring a helix-shaped walkway at KU Leuven Campus Group T, Belgium.

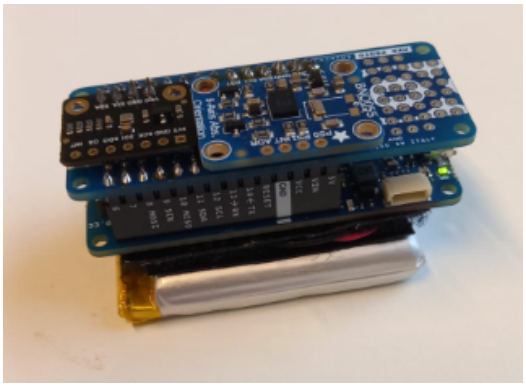

Fig. 2. The Arduino module and sensors used.

sensor data, offering orientation accuracy of ±1° in stable conditions [31]. Its self-calibrating features reduce the complexity of manual calibration, though environmental interference occasionally introduces noise. The barometer used in our module is the BMP390, a high-precision barometric pressure sensor for motion and altitude tracking. The BMP390 ensures altitude measurements in the precision of ±0.03 hPa, thereby allowing for height estimations within a range of ±0.25 meters [32]. The proposed module is cost-effective and made with readily available components. The module is shown in Fig. 2.

C. Data acquisition

In order to evaluate and test the various methods discussed in this paper, we conducted controlled experiments to capture and label head position and orientation data along the helix pathway. Only for this data acquisition, the module discussed above was extended with two additional devices: a second Arduino microcontroller to collect Wi-Fi data, and an iPhone to collect ground-truth head position and orientation data. The extended module and its mounting on a pair of headphones is shown in Fig. 2.

The data acquisition was done on the helix pathway discussed above. For each data recording session, the person wearing the measurement device must start and end the session at the same point, more precisely at the lower end of the

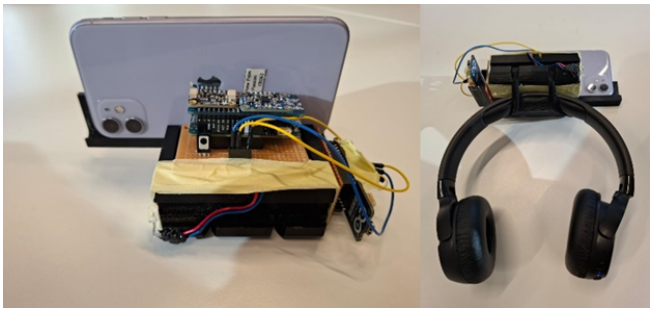

Fig. 3. The module extended with a second Arduino and an iPhone, mounted on a pair of headphones as used for data acquisition.

pathway. The person walks up the entire pathway and then walks back down to the starting point. Every measurement session lasts approximately 10 minutes and includes 2 full walks along the helical pathway starting and ending at the same point, but lowering the walking speed and hence extending the session duration can further improve the accuracy of the measurements. The data was collected at a sampling frequency of approximately 50Hz.

Two types of measurements were taken. The first one represents an ideal walking sequence where the subject walks along the middle of the helix in a straight line, with no interference, steady pace, and limited head rotation. In the second type of measurement, the subject is allowed to make more natural and random movements while walking, such as moving up and down sections of the pathway multiple times, moving the head in random orientations, moving away from the center of the pathway, etc. In total, 1 hour and 50 minutes of data was collected. The data was collected over 11 distinct instances taken at different times and dates, to account for slight pressure differences and weather conditions.

To measure accurate ground truth data, the RTAB-Map application (Real-Time Appearance-Based Mapping) [33] was used. RTAB-Map is a Simultaneous Localization and Mapping (SLAM) framework that uses RGB-D, stereo, and LiDAR technologies, as well as graph-based incremental appearance-based loop closure detection. In general, SLAM involves emitting laser pulses, analyzing their return times, and generating 3-D point clouds that represent the environment. The data can then be used to determine a subject's position and orientation. RTAB-Map boasts high accuracy and robust localization, with an average percentage translational error of 1.26% and rotational error of 0.0026 deg/m on the KITTI dataset. For more details, one can refer to [33] and the RTAB-Map repository [34].

The process of collecting ground truth data involved initializing RTAB-Map on an iPhone and running the application during the measurement sessions described above. The phone's camera was kept unobstructed throughout the recording to maximize the accuracy of the recorded data. RTAB-MAP was used as the ground truth data for benchmarking our methods, as it is an efficient and sufficiently accurate method, achieving near 1% average translational error as mentioned previously.

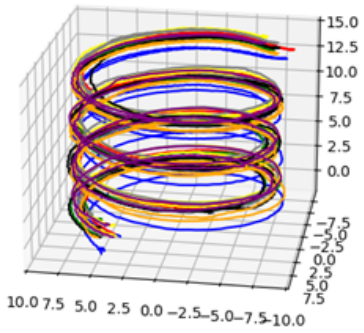

Fig. 4. The helix-shaped ground-truth position measurements after centering and alignment.

Due to the computational complexity, on a mobile iPhone device the sampling rate was limited to 10Hz.

D. Data preprocessing

After collecting our dataset, it had to be preprocessed in multiple ways before it was usable for our system design and evaluation.

First of all, since the sensor data and the ground truth data are collected on two separate independently clocked systems, a synchronization step is required. To compensate for different clock frequencies, a computer vision-based synchronization mechanism was implemented. As previously mentioned, both the raw data collection module and the iPhone were mounted on a single measurement apparatus. An LED, connected to the Arduino, was positioned in the iPhone camera's field of view. Such LED lights up when the sensors connected to the Arduino are collecting data, serving as a synchronization marker. Using the OpenCV library, the irrelevant SLAM data that was recorded when the LED was off was filtered out. In addition, the sampling frequency of the raw data was much higher than the sampling frequency of the ground truth data, meaning that at the end of each recording session there were less ground truth samples than raw ones. Because of this, linear interpolation was used when necessary to match the oversampled raw data to the closest ground truth data points. Furthermore, to compensate for the overall clock offset between the two devices, a cross-correlation of the sensor and ground truth data was computed and a time offset was applied to the sensor data accordingly.

Due to the inconsistent starting orientations and initial conditions for each measurement session, the ground truth data is not always aligned to the same 3-D spatial coordinate vector. A two-step alignment procedure was implemented, in which first all the helix-shaped position measurements are centered about the same axis, and then the measurements are rotated around this axis such that their starting points align. The centered and aligned helix-shaped position measurements are shown in Fig. 4.

Furthermore, the sensor and ground truth data are normalized such that the initial orientation values at the start of a measurement are equal to zero. To this end, the IMU and ground-truth orientation measurements all undergo a subtraction of the initial measurement value.

To ensure data accuracy, the barometer readings are preprocessed by eliminating erroneous values that are well outside the typical range expected at the measured location and altitude, often a result of sensor startup issues.

III. Dynamic Listener Positioning Methods

A. IMU-based positioning

The simplest and most well-known approach to positioning using IMU data is naive double integration dead reckoning (NDI) [35], where the acceleration components are numerically integrated into velocities, which are then in turn integrated into positions. The issue with this approach is that sensor offsets and noise can easily snowball due to the double integration procedure and as such the predictions drift very quickly [35]. In NDI experiments we ran, the algorithm's predictions went out of control very quickly. Consequently, NDI is not a solution for the indoor positioning problem.

B. Wi-Fi-based positioning

Given the ubiquitous nature of Wi-Fi networks, using Wi-Fi data for positioning is a very attractive approach. Additionally, the target environment, being a university campus, was filled with stationary Wi-Fi access points, which could be used as anchors for a fingerprinting system. To this end, we implemented a random forest regressor (RFR) model and trained it over our dataset in order to predict 3-D positions using the detected Basic Service Set Identifier (BSSID) to Received Signal Strength Indicator (RSSI) data measured from the Arduino's Wi-Fi module. The random forest architecture provides many benefits and is a strong candidate for WiFi fingerprinting based localization, being able to handle high dimensional noisy data, and being robust to outliers [36].

The RFR model input is a vector of RSSI values, with each position being associated with a unique Wi-Fi BSSID found in the training set. During inference, any BSSIDs not found in the dataset are disregarded, and any BSSIDs found have their associated RSSI values placed at the corresponding position in the input vector. Any undetected BSSIDs in the input vector are set to a value of -100 dB. A grid search was conducted over the number of estimators and the maximum depth. The results of the grid search are shown along with the model's hyperparameters in Table I, using the hyperparameter definitions from the Scikit-learn RFR implementation [37].

C. Barometer-based positioning with prior knowledge

The atmospheric pressure P is directly related to the altitude z via the barometric equation,

$$P = P_0 \, e^{-\frac{gM}{RT}(z - z_0)} \tag{1}$$

where P_0 represents the reference pressure at the reference altitude z_0 (in this case the starting point of the helix pathway), $g = 9.81$ m/s^2 is the gravitational acceleration constant, $M = 29.0$ kg/kmol is the mean molecular weight of air at sea level,

Parameter	Value	Description
bootstrap	True	Whether bootstrap samples are used when building trees.
ccp_alpha	0.0	Complexity parameter used for Minimal Cost-Complexity Pruning.
criterion	'squared_error'	The function to measure the quality of a split.
max_depth	28	The maximum depth of the tree.
max_features	1.0	The number of features to consider when looking for the best split.
max_leaf_nodes	None	The maximum number of leaf nodes.
max_samples	None	The number of samples to draw from X to train each base estimator.
min_impurity_decrease	0.0	A node will be split if this split decreases the impurity by this amount.
min_samples_leaf	1	The minimum number of samples required to be at a leaf node.
min_samples_split	2	The minimum number of samples required to split an internal node.
min_weight_fraction_leaf	0.0	The minimum weighted fraction of the sum total of weights required to be a leaf node.
monotonic_cst	None	A monotonic constraint on the features.
n_estimators	525	The number of trees in the forest.
n_jobs	None	The number of jobs to run in parallel.
oob_score	False	Whether to use out-of-bag samples to estimate the generalization score.
random_state	None	Controls both the randomness of the bootstrapping and the sampling of the features.
verbose	0	Controls the verbosity when fitting and predicting.
warm_start	False	When true, reuse the solution of the previous call to fit as initialization.

TABLE I
MODEL HYPERPARAMETERS OF THE RFR MODEL USING THE SCIKIT-LEARN IMPLEMENTATION [37], AND THEIR OPTIMAL VALUES FOR OUR TASK.

$R = 8.31$ N·m/mol·K is the universal gas constant, and $T = 293$ K is the room temperature.

Barometer-based measurements of atmospheric pressure can henceforth be used to estimate altitude in indoor positioning [38]. Moreover, as the 3-D environment considered in this XR use case approximately admits a closed-form relation between the x, y, and z coordinates, the complete 3-D position coordinate vector can be estimated from barometer data. The parametric equation for a perfect helix having a central axis parallel to the z-axis, is given by the system of equations

$$\begin{cases} x = x_0 + r\cos(\theta + \varphi) \\ y = y_0 + r\sin(\theta + \varphi) \\ z = \frac{p}{2\pi}\theta \end{cases} \tag{2}$$

where $x, y, z \in \mathbb{R}$ represent the 3-D Cartesian coordinates of a point on the helix and the parameter $\theta \in [0, 2\pi)$ represents the azimuthal angle. The equation also depends on a number of constant parameters related to the shape of the helix: x_0, y_0 represent the x and y coordinates of the center point of the circle obtained by projecting the helix onto the x-y plane, φ is a fixed phase shift that follows from the alignment procedure explained in Section II-D, and p and r represent the radius and pitch of the helix. The helix considered in the XR use case environment introduced in Section II-A is characterized by the parameter values $p = 4.2$ m and $r = 8.0$ m, whereas x_0, y_0 follow from the choice of the coordinate system and φ follows from the alignment procedure. Note that the helix-shaped pathway in our use case environment has a width of 2.4 m, but this is not taken into account in its geometric model (2). Fig. 5 shows the resulting helix curve, overlaid with the curves corresponding to the ground truth position measurements. This model was also quantitatively compared to the dataset, with average deviation from the measured positions of 5.35m and root mean squared deviation of 6.30m. Due to the width of the spiral of 2.4m in practice and the various differences in height and movement during the dataset, these deviations are acceptable and show the accuracy of the derived spiral model.

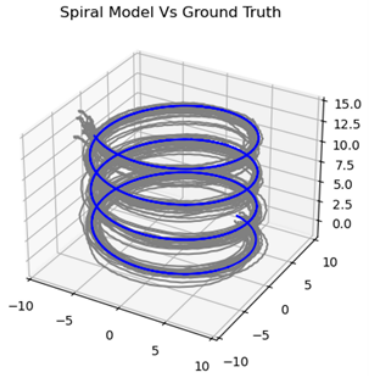

Fig. 5. Curve of the helix model (blue) overlaid with curves resulting from the ground truth position measurements (gray).

The barometer-based positioning method first estimates the (time-varying) z coordinate from the (time-varying) atmospheric pressure measurement $P_b(t)$, and then uses the estimate $\hat{z}_b(t)$ together with the prior knowledge on the geometry of the environment to also estimate the (time-varying) x and y coordinates:

$$\hat{z}_b(t) = z_0 - \frac{RT}{gM}\ln\frac{P_b(t)}{P_0} \tag{3}$$

$$\hat{x}_b(t) = x_0 + r\cos\left(\frac{2\pi}{p}\hat{z}_b(t) + \varphi\right) \tag{4}$$

$$\hat{y}_b(t) = y_0 + r\sin\left(\frac{2\pi}{p}\hat{z}_b(t) + \varphi\right) \tag{5}$$

D. Kalman-filter-based sensor fusion

The main conjecture of this paper is that sensor fusion may lead to more accurate positioning estimates compared

to using only one sensor modality. We propose a sensor fusion approach based on Kalman filtering. A first Kalman filtering scheme fuses the IMU's accelerometer data with the BMP390's atmospheric pressure data. To this end we propose the following model defined below. The state vector $\mathbf{x}(t)$ is defined as a 6×1 vector containing the listener position coordinates and velocity components along the x, y, and z axes,

$$\mathbf{x}(t) = \begin{bmatrix} x(t) & y(t) & z(t) & \dot{x}(t) & \dot{y}(t) & \dot{z}(t) \end{bmatrix}^T. \quad (6)$$

The state transition matrix $\mathbf{A}$ includes a finite-difference approximation of the first derivative, relating the velocity to the position, as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \Delta t\, \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix} \quad (7)$$

where $\mathbf{I}_{n\times n}$ and $\mathbf{0}_{n\times n}$ represent $n \times n$ identity and all-zero matrices, and Δt is the time step between two successive measurements. The IMU data are fused into the model by means of the input vector $\mathbf{u}(t)$, which is defined as a 3×1 vector containing the measured acceleration components along the x, y, and z axes,

$$\mathbf{u}(t) = \begin{bmatrix} \ddot{x}_{\text{IMU}}(t) & \ddot{y}_{\text{IMU}}(t) & \ddot{z}_{\text{IMU}}(t) \end{bmatrix}^T. \quad (8)$$

The 6×3 input matrix $\mathbf{B}$ is then based on a finite-difference approximation of the first and second derivative, relating the acceleration to the position and the velocity, as follows

$$\mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2\, \mathbf{I}_{3\times3} \\ \Delta t\, \mathbf{I}_{3\times3} \end{bmatrix} \quad (9)$$

Note that in this way, the NDI method is implicitly included in our state-space model. The measurement vector $\mathbf{y}(t)$ is made up of the position estimates obtained from the barometer-based method with prior knowledge,

$$\mathbf{y}(t) = \begin{bmatrix} \hat{x}_b(t) & \hat{y}_b(t) & \hat{z}_b(t) \end{bmatrix}^T. \quad (10)$$

As no velocity measurements are made, the 3×6 measurement matrix $\mathbf{H}$ is defined as follows,

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}. \quad (11)$$

The process noise $\mathbf{w}(t)$ is assumed to be zero-mean Gaussian white noise with covariance matrix $\mathbf{R}_w = \sigma_w^2 \mathbf{I}_{6\times6}$. The measurement noise $\mathbf{n}(t)$ on the x, y, z position measurements is assumed to be uncorrelated with zero mean and a Gaussian distribution, but the noise variance on the x, y, z components can be different, which provides more flexibility to model the effect that one coordinate is estimated more accurately than others. In particular, only the z coordinate is estimated based on the atmospheric pressure measurement, while the x and y coordinates are derived from prior knowledge. We hence assume a measurement noise covariance matrix

$$\mathbf{R}_n = \begin{bmatrix} \sigma_{n,x}^2 & 0 & 0 \\ 0 & \sigma_{n,y}^2 & 0 \\ 0 & 0 & \sigma_{n,z}^2 \end{bmatrix} \quad (12)$$

where, in the case the barometer-based method with prior knowledge is used for the measurement, we also assume that $\sigma_{n,z}^2 > \sigma_{n,x}^2, \sigma_{n,y}^2$.

Using the defined terms and parameters we end up with the following state-space model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \quad (13)$$

$$\mathbf{y}(t) = \mathbf{H}\mathbf{x}(t) + \mathbf{n}(t) \quad (14)$$

The noise variances $\sigma_w^2, \sigma_{n,x}^2, \sigma_{n,y}^2, \sigma_{n,z}^2$ are hyperparameters of the resulting Kalman filter were tuned manually in order to choose the best performing values. The tuning was done by iteratively varying the diagonal elements of the Q and R matrices, and applying a grid search over a log scale from 10^{-2} to 10^5. This approach allowed for systemic exploration of the parameter space, allowing us to identify the optimal balance between trust in the dynamic model versus the measurements. Lower weights on the IMU data and higher weight on the barometer data results in smoother and more stable estimates, which are however less responsive to rapid high-frequency changes. The selected values minimized the residual error across the test dataset while maintaining a stable and responsive state estimate.

A second Kalman filtering scheme also incorporates the Wi-Fi positioning measurements, in addition to the data used in the first scheme. As the Wi-Fi and barometer sensors are not synchronously sampled, their measurements come in at different time instants. At time instants where barometer data are received, the measurement vector in (10) and the noise covariance matrix in (12) are used to update the Kalman filter. At time instants where Wi-Fi data are received, the measurement vector contains the Wi-Fi positioning estimates, and the measurement noise variances $\sigma_{n,x}^2, \sigma_{n,y}^2, \sigma_{n,z}^2$ are modified to represent the different error-proneness of the Wi-Fi positioning approach.

IV. Implementation aspects

A. Orientation estimation

The second step of the head-tracking system was implementing an attitude estimation system. Our implemented attitude system uses the BNO055 IMU's orientation estimation system, with some further preprocessing. The data is retrieved as roll, pitch, yaw Euler angles in degrees, and are baselined on an initial assumption of the user starting approximately at the orientation origin [31].

B. Hardware limitations

The Arduino MKR 1010 is not designed for computationally intensive tasks like matrix multiplication. Small matrices (e.g., 3×3 or 4×4) can be handled reasonably, but as matrix sizes grow, performance drops significantly due to limited processing power and memory. NumPy on the other hand uses highly optimized libraries like BLAS and LAPACK. These libraries leverage multi-threading, SIMD instructions, and efficient memory access patterns to maximize speed. The performance scales well with larger matrices [25]. To that end,

Type	Name	Size (bytes)	Notes
unsigned long	microsT	4	4 bytes padding
double	linaccelx, linaccely, linaccelz	$8 \times 3 = 24$	Linear acceleration
double	gyrox, gyroy, gyroz	$8 \times 3 = 24$	Gyroscope values
double	magnx, magny, magnz	$8 \times 3 = 24$	Magnetometer values
double	roll, pitch, yaw	$8 \times 3 = 24$	Orientation values
int8_t	tempnbo	1	7 bytes padding
double	tempbmp	8	Temperature
double	pressure	8	Pressure

TABLE II
STRUCTURE OF DATAENTRY (SIZE: 128 BYTES).

Type	Name	Size (bytes)	Notes
unsigned long	microsT	4	-
int8_t	rssiCnt	1	1 byte padding
byte	BSSIDs[25][6]	$25 \times 6 = 150$	1 byte padding
int32_t	RSSIs[25]	$25 \times 4 = 100$	-

TABLE III
STRUCTURE OF RSSIDATAENTRY (SIZE: 256 BYTES).

we decided to implement the Kalman filtering scheme in a Python desktop environment instead of on the Arduino.

C. Real-time communication

Given that our signal processing schemes were implemented to run in a Python desktop environment, a real-time network communication has been set up between the Arduino module and the Python environment such as to allow for real-time operation of the head-tracking system. We implemented two interchangeable communication layers; one built on TCP, and one built on UDP. The UDP system is preferred and was used for our data collection and real-time testing, to minimize the latency overhead by eliminating TCP acknowledgement and retry packets. The system relies on two internal data structures used to represent the IMU and barometer sensor data and the Wi-Fi data respectively. The proposed data structures are detailed in Tables II and III.

It should be noted that the sampling rate of the ground truth data is lower than that of the measurement device. The sampling frequency of RTAB-MAP results in a maximum latency of 100ms, excluding processing and data transmission delays, which is near the average of 107.6ms detectibility threshold for temporal delay between movement and audio rendering, thus it would detract from the auralization experience. The proposed system samples at a 5x higher rate however, which would only account for a latency of 20ms, well below the average threshold and significantly below the minimum threshold measured in [39] of 53ms. This latency is then combined with the processing and network latency, which were measured to be below 20ms. This contextualizes the benefits of our proposed system over using a phone based RTAB-MAP SLAM system directly.

V. RESULTS AND EVALUATION

In order to assess the performance of the various components of the proposed system, we ran a number of experiments, the results of which are presented below.

TABLE IV
ERROR METRICS FOR WI-FI-BASED POSITIONING

Metric	Validation Set	Unseen Set
Average Error (m)	6.47	6.93
Maximum Error (m)	9.27	11.33
Mean Squared Error (m^2)	43.9	49.7
R^2 Score	0.536	-

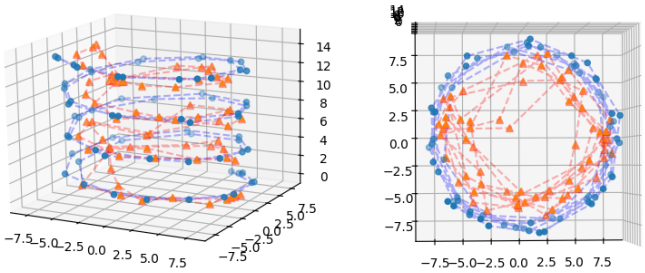

Fig. 6. An example of the Wi-Fi-based RFR model predictions (orange with triangle marker) and the ground truth path (blue with circle marker).

A. Positioning performance

1) Wi-Fi-based positioning: The Wi-Fi-based RFR model was evaluated in terms of the average error, maximum error, mean squared error, and R^2 score [37]. These error metrics were computed on a validation set (used to tune the model hyperparameters) and on the unseen test set. The results are shown in Table IV.

A qualitative sample of the RFR model running on a single data sequence is seen in Fig. 6, showing the Wi-Fi model's predictions in orange, and the ground truth positions in blue.

2) Kalman-filter-based sensor fusion: The Kalman filtering schemes were evaluated using three error metrics: the absolute trajectory error (ATE), the relative trajectory error (RTE), and the average positioning error (APE) [34], [35]. In order to be able to evaluate the metrics over two signals with different sampling period, the evaluation was done after linear interpolation. The results are shown in Table V. It should be noted that tuning the Kalman filter affects the error measurements, with the best found values producing errors similar to the barometer only case.

TABLE V
TRAJECTORY AND POSITIONING ERRORS USING KALMAN-FILTER-BASED SENSOR FUSION

Error metric	Barometer data only	Kalman filter fusing IMU and barometer data	Kalman filter fusing IMU, barometer and Wi-Fi data
ATE (m)	3.28	≥ 3.28	5.54
RTE (m)	3.01	≥ 3.01	3.07
APE (m)	2.25	≥ 2.25	4.07

TABLE VI
ERROR METRICS FOR OVERALL ORIENTATION ESTIMATION

Metric	Value (°)
Median Error	35
Average Error	39
Maximum Error	162

TABLE VII
ERROR METRICS FOR ORIENTATION ESTIMATION IN ROLL, PITCH, AND YAW COMPONENTS

Metric	Roll (°)	Pitch (°)	Yaw (°)
Median Error	34.39	3.75	2.40
Average Error	38.61	4.09	2.84
Maximum Error	162.46	21.50	18.21

B. Orientation estimation performance

To evaluate the orientation system, we used the average, median, and maximum error as evaluation metrics. The overall error metrics are given in Table VI, while the error metrics per component are shown in Table VII.

Furthermore, a qualitative example of a single sequence is shown in figs. 7 to 9. In this sequence, the measured values along all three axes (heading, pitch, yaw) are compared to the ground truth, represented in dashed gray and solid red respectively.

VI. Discussion

A. Positioning

In our use case, the pressure-based system heavily outperforms every other proposed system, barring the Kalman filtering schemes tuned to prioritize pressure data. This is due to the high accuracy and sampling rate of the pressure data in our system, which when combined with the well-tuned geometric model of the environment, allows us to calculate accurate positions at a very high sampling rate. Consequently, the noisy IMU and Wi-Fi data do not bring much benefit. It is expected, however, that in environments of which the geometry can be less precisely modeled, the IMU and Wi-Fi data may be more useful. The achieved accuracy allows for robust audio rendering, with an angular direction offset of approximately 12 and 24 degrees for objects at a distance of 10m and 5m respectively.

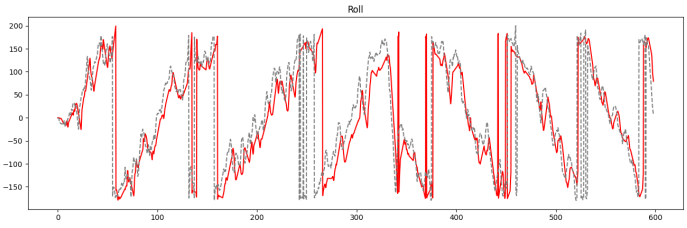

Fig. 7. The measured (gray - dashed) and ground truth (red - solid) rotation values for the heading axis.

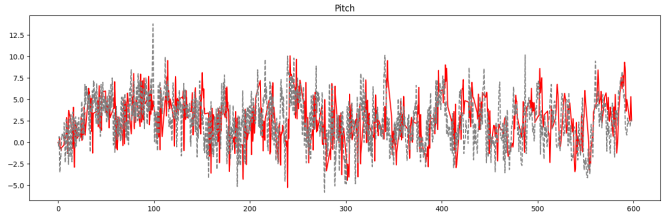

Fig. 8. The measured (gray - dashed) and ground truth (red - solid) rotation values for the pitch axis.

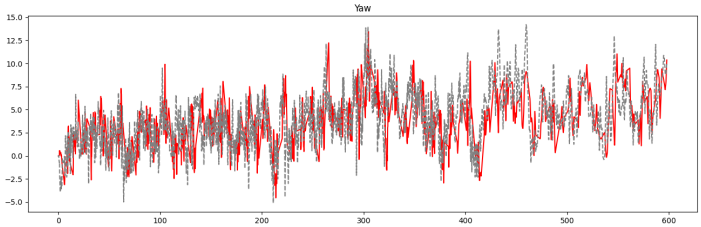

Fig. 9. The measured (gray - dashed) and ground truth (red - solid) rotation values for the yaw axis.

B. Orientation estimation

The head orientation estimation performs relatively well, with generally accurate orientation values along entire sequences. However, the system is prone to occasional drift issues, specifically on the heading (roll) axis. This is due to two main reasons. Firstly, the heading axis lacks an accurate baseline in the form of the gravity vector, which is only available on the other two axes. This issue can be mitigated by integrating a more accurate sensor, using a physical anchor for an effect similar to the gravity vector, or implementing a Kalman filtering or machine learning scheme to further improve the predicted angles. Secondly, the heading axis exhibits significantly larger variance than the other two axes of rotation, simply due to human anatomy and range of motion. The performance along the heading axis is too low for accurate audio rendering, however along the other two axes the system can produce an immersive auralization experience.

VII. Conclusion

In this paper, we presented a robust real-time head tracking system tailored for an auralization XR use case in a geometrically well-defined environment. Several positioning and head orientation estimation methods were implemented, based on atmospheric pressure models, Kalman filtering schemes, and machine learning approaches like Random Forest Regression, by fusing sensors with the Bosch BNO055 IMU, BMP390 pressure sensor, and Wi-Fi-based positioning. While the barometer-based system was the most accurate for vertical positioning, the Wi-Fi integration and Kalman filters have the potential to improve the accuracy in conditions where prior knowledge on the geometry of the environment is limited. The attitude estimation, while reliable, had issues such as drift in the heading (roll) axis because of the inherent limitations in

the IMU-based measurements. In addition, we released a well-documented and extensive dataset for head tracking, enabling future research and providing a benchmark for quantitative comparisons.

Future work is directed at both hardware and algorithm improvements. For example, moving to more powerful microcontrollers, such as the ESP32, would support on-device processing and decrease latency and system complexity. Alternatively, the existing system can be expanded to support multiple trackers by leveraging a distributed computing architecture. Other improvements involve enhancing synchronization during preprocessing, and using higher-order positioning techniques, such as step counting and machine-learning-driven alignment to the environment geometry. These improvements are aimed at solving current limitations, enhancing generalizability, and ensuring the robustness of the system in various real-world applications.

REFERENCES

[1] T. Jung, M. C. tom Dieck, and S. M. C. Loureiro, Eds., *Extended Reality and Metaverse: Immersive Technology in Times of Crisis*. Springer, 2023.

[2] R. Gupta, J. He, R. Ranjan, W.-S. Gan, F. Klein, C. Schneiderwind, A. Neidhardt, K. Brandenburg, and V. Välimäki, "Augmented/mixed reality audio for hearables: Sensing, control, and rendering," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 63–89, 2022.

[3] P. Crum, "Hearables: Here come the: Technology tucked inside your ears will augment your daily life," *IEEE Spectrum*, vol. 56, no. 5, pp. 38–43, 2019.

[4] K. Müller and F. Zotter, "Auralization based on multi-perspective ambisonic room impulse responses," *Acta Acustica*, vol. 4, no. 6, p. 25, 2020.

[5] A. Härmä, J. Jakka, M. Tikander, M. Karjalainen, T. Lokki, J. Hiipakka, and G. Lorho, "Augmented reality audio for mobile and wearable appliances," *J. Audio Eng. Soc.*, vol. 52, no. 6, p. 618–639, 2004.

[6] N. Mariette and B. Katz, "Sounddelta – large scale multi-user audio augmented reality," in *Proc. EAA Symp. Auralization*, Espoo, Finland, 2009, p. 15–17.

[7] K. Hartung, J. Braasch, and S. Sterbing, "Comparison of different methods for the interpolation of head-related transfer functions," in *Proc. AES 16th Int. Conf. Spatial Sound Reproduction*, Rovaniemi, Finland, 1999.

[8] H. Gamper, "Head-related transfer function interpolation in azimuth, elevation, and distance," *J. Acoust. Soc. Amer*, vol. 134, no. 6, p. 547–553, 2013.

[9] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating interactive virtual acoustic environments," *J. Audio Eng. Soc.*, vol. 47, no. 9, p. 675–705, 1999.

[10] H. Hacıhabiboğlu, E. Sena, Z. Cvetković, J. Johnston, and J. Smith, III, "Perceptual spatial audio recording, simulation, and rendering," *IEEE Signal Process. Mag.*, vol. 34, no. 3, p. 36–54, 2017.

[11] J. Arend, S. Garí, C. Schissler, F. Klein, and P. Robinson, "Six-degrees-of-freedom parametric spatial audio based on one monaural room impulse response," *J. Audio Eng. Soc.*, vol. 69, no. 7/8, p. 557–575, Jul./Aug. 2021.

[12] S. Werner, F. Klein, A. Neidhardt, U. Sloma, C. Schneiderwind, and K. Brandenburg, "Creation of auditory augmented reality using a position-dynamic binaural synthesis system — technical components, psychoacoustic needs, and perceptual evaluation," *Appl. Sci.*, vol. 11, no. 3, p. 1150–, 2021.

[13] A. Plinge, S. Schlecht, O. Thiergart, T. Robotham, O. Rummukainen, and E. Habets, "Six-degrees-of-freedom binaural audio reproduction of first-order ambisonics with distance information," in *Proc. 2018 AES Int. Conf. Audio Virtual Augmented Reality*, 2018.

[14] J. Tylka and E. Choueiri, "Fundamentals of a parametric method for virtual navigation within an array of ambisonics microphones," *J. Audio Eng. Soc.*, vol. 68, no. 3, p. 120–137, 2020.

[15] F. Zotter, M. Frank, C. Schörkhuber, and R. Höldrich, "Signal-independent approach to variable-perspective (6DoF) audio rendering from simultaneous surround recordings taken at multiple perspectives," in *Fortschritte der Akustik (DAGA)*, 2020.

[16] M. Zaunschirm, M. Frank, and F. Zotter, "Binaural rendering with measured room responses: First-order ambisonic microphone vs. dummy head," *Appl. Sci.*, vol. 10, no. 5, p. 1631–, 2020.

[17] M. Blochberger and F. Zotter, "Particle-filter tracking of sounds for frequency-independent 3D audio rendering from distributed B-format recordings," *Acta Acustica*, vol. 5, no. 20, 2021.

[18] L. McCormack, A. Politis, T. McKenzie, C. Hold, and V. Pulkki, "Object-based six-degrees-of-freedom rendering of sound scenes captured with multiple ambisonic receivers," *J. Audio Eng. Soc.*, vol. 70, no. 5, p. 355–372, 2022.

[19] F. Schultz and S. Spors, "Data-based binaural synthesis including rotational and translatory head-movements," in *Proc. AES 52nd Int. Conf. Sound Field Control*, 2013.

[20] N. Iijima, S. Koyama, and H. Saruwatari, "Binaural rendering from microphone array signals of arbitrary geometry," *J. Acoust. Soc. Amer.*, vol. 150, no. 4, p. 2479–2491, 2021.

[21] E. Fernandez-Grande, D. Caviedes-Nozal, M. Hahmann, X. Karakonstantis, and S. Verburg, "Reconstruction of room impulse responses over extended domains for navigable sound field reproduction," in *Proc. Int. Conf. Immersive & 3D Audio (I3DA '21)*, 2021.

[22] O. Puomio, T. Pihlajakuja, and T. Lokki, "Sound rendering with early reflections extracted from a measured spatial room impulse response," in *Proc. Int. Conf. Immersive & 3D Audio (I3DA '21)*, 2021.

[23] T. Deppisch, S. Garí, P. Calamia, and J. Ahrens, "Perceptual evaluation of spatial room impulse response extrapolation by direct and residual subspace decomposition," in *Proc. 2022 AES Int. Conf. Audio Virtual Augmented Reality*, 2022.

[24] A. M. Sabatini, "Kalman-filter-based orientation determination using inertial/magnetic sensors: Observability analysis and performance evaluation," *Sensors*, vol. 11, pp. 9182–9206, 2011.

[25] H. Hellmers, A. Norrdine, J. Blankenbach, and A. Eichhorn, "An IMU/magnetometer-based indoor positioning system using Kalman filtering," *Proc. Int. Conf. Indoor Positioning and Indoor Navigation*, 2013.

[26] S. Tiwari and V. K. Jain, "A novel step detection technique for pedestrian dead reckoning based navigation," *ICT Express*, 2022.

[27] L. Huang, H. Li, W. Li, W. Wu, and X. Kang, "Improvement of pedestrian dead reckoning algorithm for indoor positioning by using step length estimation," *Int. Archives Photogrammetry, Remote Sensing, Spatial Inf. Sci.*, vol. XLVIII-3/W1-2022, pp. 19–24, 2022.

[28] I. Stanculeanu and T. Borangiu, "Enhanced RSSI localization system for asset tracking services using non expensive IMU," *IFAC Proc. Volumes*, vol. 45, pp. 1838–1843, 2012.

[29] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A location estimation algorithm based on RSSI vector similarity degree," *Int. J. Distributed Sensor Networks*, vol. 10, p. 371350, 2014.

[30] S. Devleminck, B. Debackere, and T. van Waterschoot, "Multi-viewpoint strategies: Ambisonic auralization and localization through walking and listening as places of negotiation in conditions of hybridity and change," in *Proc. Int. Symp. Electronic Art (ISEA '19)*, 2019, p. 99–105.

[31] "BNO055 datasheet." [Online]. Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf

[32] "BMP390 datasheet." [Online]. Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp390-ds002.pdf

[33] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[34] "RTAB-Map." [Online]. Available: https://introlab.github.io/rtabmap/

[35] H. Yan, S. Herath, and Y. Furukawa, "RoNIN: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods," arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1905.12853

[36] R. Gomes, M. Ahsan, and A. Denton, "Random forest classifier in SDN framework for user-based indoor localization," *Electro/Information Technology*, 2018.

[37] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Machine Learning Res.*, vol. 12, pp. 2825–2830, 2011.

[38] B. Li, B. Harvey, and T. Gallagher, "Using barometers to determine the height for indoor positioning," in *Proc. Int. Conf. Indoor Positioning and Indoor Navigation*, 2013, pp. 1–7.
[39] A. Lindau, "The perception of system latency in dynamic binaural synthesis," 01 2009.