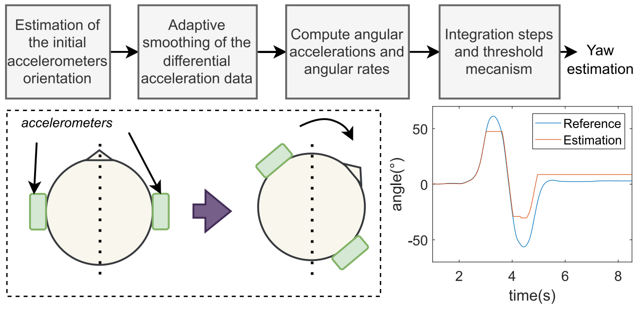
Human Head Yaw Estimation Based on Two 3-Axis Accelerometers

Vincent Grimaldi, Laurent S. R. Simon, Miquel Sans, Gilles Courtois, and Hervé Lissek

***Abstract*—The estimation of the orientation of an object, and a human head in particular, can be defined by the Euler angles: the yaw, pitch and roll. The robust and drift-free estimation of those angles is usually achieved with the data from several sensors such as accelerometers, gyroscopes and magnetometers, processed with sensor fusion algorithms. However, wearable devices such as hearing instruments are rarely equipped with all those sensors, and might usually only have a single accelerometer embedded per device. While it is possible to retrieve a correct estimation of the roll and pitch using only a single accelerometer, estimating the yaw is a more challenging task as accelerations around the gravity vector cannot be detected by an accelerometer. In the context of binaural communication devices and spatial hearing, the yaw of the head is a key information that helps achieving dynamic binaural synthesis. This work proposes an algorithm that aims at estimating the yaw of a human head by using only two 3-axis accelerometers that are placed at each side of the head. The algorithm is evaluated with acceleration measurements of human subjects achieving a realistic task. The results suggest that the strategy used is promising as a good performance can be achieved for a majority of the measures as long as an individualized set of parameters can be selected. The results also suggest that the performance of the current implementation is sensitive to sensor displacement after calibration or a wrong estimation during initialization.**


Estimation of the initial accelerometers orientation
Adaptive smoothing of the differential acceleration data
Compute angular accelerations and angular rates
Integration steps and threshold mecanism
Yaw estimation
accelerometers
Reference
Estimation
50
0
-50
angle(°)
2
4
6
8
time(s)

***Index Terms*—Head-tracking, accelerometers, yaw, head movements.**

I. Introduction

BILATERAL motion sensors worn on the head are typically found in earables and hearing instruments (HIs). Accelerometers have recently been integrated in those devices as they combine a limited size and a low power consumption [1]. Multiple functionalities are developed allowing detecting an abrupt fall of a user, tracking sudden peaks in the vertical dimension of an accelerometer signal, and confirming the fall by comparing the left and right acceleration patterns [2], [3]. More complex systems were designed to evaluate the type of motion of the user (immobile, walking, running). Those systems can be used to steer audio processing algorithms aiming at enhancing speech intelligibility and/or spatial awareness [4], [5]. Voss *et al.* [6] showed significant improvement in speech understanding, sound localization as well as preference in hearing-impaired listeners with such steering approaches. Being able to track the head motions of a user is of high interest since this would allow controlling the beam of a directional microphone system for example. In the context of spatial audio rendering with wearable devices, the knowledge of the orientation of the user's head in the azimuth plan enables achieving dynamic spatial rendering, i.e. the spatial filtering can be adapted to remain valid according to the listener's motions. Such processing can greatly improve the listener's spatial audio experience by increasing plausibility and reducing typical perception artefacts such as front-back confusions [7]. Previously developed binaural spatialization algorithms aim at improving the perception of auditory externalization and distance in normal hearing and hearing-impaired listeners [8]. In particular they are optimized for low computational cost, to be compatible with wearable devices [9]. The addition of head-tracking in this context is motivated by the potential increase in auditory externalization demonstrated in several studies [10], [11].

The orientation of an object, and a human head in particular, can be determined using Euler angles. Each angle, named the roll, pitch and yaw, corresponds to the rotation from an initial position, around the x-, y- and z-axis respectively. Euler angles are expressed as a sequence of those three rotations

Manuscript received 27 May 2022; accepted 14 June 2022. This work was co-financed by the Swiss Innovation Promotion Agency Innosuisse under Grant 41312.1 PFLS-LS. The associate editor coordinating the review of this article and approving it for publication was Prof. Chao Tan. *(Corresponding author: Vincent Grimaldi.)*

Vincent Grimaldi and Hervé Lissek are with the Signal Processing Laboratory 2, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: vincent.grimadli@epfl.ch).

Laurent S. R. Simon and Gilles Courtois are with Sonova AG, 8712 Stäfa, Switzerland.

Miquel Sans is with Sonova Communications AG, 3280 Murten, Switzerland.

Digital Object Identifier 10.1109/JSEN.2022.3185712

around the rigid body's local coordinate axes. One of the most common method for tracking each of these angles relies on inertial sensors, and is usually achieved using at least a combination of one or more three-axis accelerometers and one or more three-axis gyroscopes. The gyroscope measures the sensor's angular velocity, while the accelerometer measures the external specific force acting on the sensor, i.e. the earth's gravity and the sensor's acceleration. A magnetometer can also be used in combination, to help determine the orientation of the system using the earth's magnetic field. Devices using such sensors are usually referred to as inertial measurement units (IMUs). However, IMUs are typically prone to integration drift. Errors due to the measurement of acceleration and angular velocity can progressively be integrated and the accumulation leads to larger errors in the estimation of the velocity and the angle. Some devices combine an IMU and an on-board processing system to run sensor fusion algorithms and aim at reducing this type of drift [12], [13]. Such devices are usually referred to as Attitude and Heading Reference Systems (AHRSs). Drift from the gyroscopes integration can then be compensated by the reference vectors: gravity and the earth magnetic field. With all these devices, eventually in combination with machine learning algorithms, a reliable drift-free instantaneous estimation of the head orientation can be obtained [14], [15].

Nevertheless, not every device can be equipped with all the necessary sensors and few solutions have been proposed to track head movements relying solely on bilateral accelerometers so far. Some devices, such as HIs for which battery life and compactness are key factors, are only equipped with a single three-axis accelerometer. Such an accelerometer can be sufficient to determine the roll and pitch of the object to which it is attached, as detailed in [16], but does not allow estimating the yaw alone. An algorithm aiming at estimating the yaw rate of a vehicle using two 1-axis accelerometers installed on the center-line of the vehicle was presented in [17]. Nevertheless the proposed algorithm relies on a model and assumptions associated with the vehicle tracking use case, which are not applicable for tracking the motions of a human head.

In [18], an algorithm is proposed to determine the yaw and roll angular rotation rates of a spinning rigid object using only two linear three-axis accelerometers. The method was found to achieve satisfactory performance when the rotational motions were rather short (< 2 s) and fast (140 °/s to 1200 °/s). The proposed method is dependant on the sensors noise and relies on a fixed threshold to prevent drifting due to this noise. It is also specified that the angular accelerations need to be lower than the accelerometer sensing range in order to not saturate the sensor. For example, in [18], this corresponded to 1200 °/s. In the use case of head motions, typical movements rarely exceed peak values of 350 °/s in the case of healthy subjects [19].

This article describes a real-time method aiming at estimating the relative yaw position of a human head using only two 3-axis accelerometers such as the ones found on wearable devices. Such solution relying only on accelerometers may provide significant power consumption savings over solutions relying on the use of a gyroscope and additional sensors.

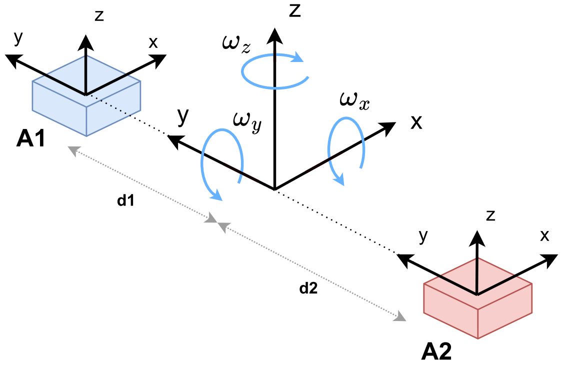


Fig. 1. Model of the accelerometers setup. The accelerometers A1 and A2 are spaced by a distance $D = d_1 + d_2$ and supposed to be aligned together with the center of the system on the y-axis.

II. Description of the Algorithm

This section describes a method algorithm for estimating the yaw position of a human head using two 3-axis accelerometers. The base concept of the proposed algorithm is directly inspired by the one described in [18], which aims at estimating the yaw velocity of a spinning object. The principle of this base algorithm is to combine the data from two accelerometers placed on the spinning object in order to estimate the angular rotation speeds. The present method proposes several additional steps and improvement to extend it to the more complex task of tracking the yaw of human head motions using the accelerometers embedded in ear-worn hearing devices.

A. Description of the Model

Two accelerometers A1 and A2 are mounted on the opposite sides of a head, at the position of the ears. In practice, in the work reported in this paper, they were embedded in hearing devices placed above the ears. It is assumed that each of those accelerometers is fixed on the head, and cannot move independently from each other nor from the head. Their coordinate system $(x, y, z, \omega_x, \omega_y, \omega_z)$ is defined as shown in Fig. 1. In this algorithm, both accelerometers are supposed to be aligned with the center of the head, on the y-axis of rotation.

B. Base Algorithm

For this part of the algorithm, both x-axes of the accelerometers are considered parallel to each other, and the same applies to both z-axes. This is not the case in practical cases where accelerometers are placed on a human head, but the issue is addressed in II-C.4.

1) Computation of the Rotational Speed Magnitude: The acceleration measured by each of the accelerometer A1 and A2 along the y-axis is the superposition of a rotation component (the radial acceleration) and a translation component:

$$a_{y1} = -a_{r_{y_1}} + a_{t_y} = -\Omega_r^2 d_1 + a_{t_y} \tag{1}$$

$$a_{y2} = a_{r_{y_2}} + a_{t_y} = \Omega_r^2 d_2 + a_{t_y} \tag{2}$$

where d_1 (respectively d_2) is the distance between the center of the head and the accelerometer A1 (respectively A2) and

Ω_r is the angular speed. By taking the difference between a_{y1} and a_{y2}, it is possible to eliminate the translation component, and deduce the angular speed:

$$\Omega_r = \sqrt(|\frac{a_{y1} - a_{y2}}{D}|) \tag{3}$$

2) Computation of the Angular Velocities: To compute the angular velocities ω_x and ω_z around the x- and z-axis, it is necessary to compute and integrate the angular accelerations α_x and α_z. The acceleration along the x-axis measured by each accelerometer A1 and A2 is the superposition of a rotation component (the tangential acceleration) and a translation component:

$$a_{x1} = -a_{r_{x_1}} + a_{t_x} = -\alpha_z d_1 + a_{t_x} \tag{4}$$

$$a_{x2} = a_{r_{x_2}} + a_{t_x} = \alpha_z d_2 + a_{t_x} \tag{5}$$

By taking the difference between a_{x2} and a_{x1}, it is possible to deduce the angular accelerations around the z-axis as:

$$\alpha_z = \frac{a_{x2} - a_{x1}}{D} \tag{6}$$

Similarly, the angular accelerations around the x-axis is:

$$\alpha_x = \frac{a_{z2} - a_{z1}}{D} \tag{7}$$

The integration of α_z and α_x enables determining ω_z and ω_x from an initial time instant t_{init} at which the speed is known, e.g. $\omega_x(t_{init}) = \omega_z(t_{init}) = 0$:

$$\omega_z = \int_{t_{init}}^{t} \alpha_z dt \tag{8}$$

$$\omega_x = \int_{t_{init}}^{t} \alpha_x dt \tag{9}$$

3) Computation of the Yaw Estimation: In the setup described in Fig. 1, the accelerometers A1 and A2 are located on the rotation y-axis. Consequently, an arbitrary angular motion around the y-axis has a contribution equal to zero in $a_{x1}, a_{x2}, a_{y1}, a_{y2}, a_{z1}$ and a_{z2}. Hence, the total angular rotation measurable by this setup is only affected by rotations around the x-axis and z-axis. So the total angular rotation rate ω_{total} is given by:

$$\omega_{total} = \sqrt{\omega_z^2 + \omega_x^2} \tag{10}$$

Using the total rotation rate magnitude calculated in Eq. (10) and the relative magnitude and direction of rotation obtained from Eq. (3), the yaw velocity $y\dot{a}w$ can be determined as:

$$y\dot{a}w = \Omega_r \frac{\omega_z}{\omega_{total}} \tag{11}$$

Then, it is possible to add another integration step to the algorithm described in [18] to derive the yaw:

$$yaw = \int_{t_{init}}^{t} \Omega_r \frac{\omega_z}{\omega_{total}} dt \tag{12}$$

where t_{init} is a time instant at which the orientation of the user's head relative to a fixed point in the environment is known, e.g. when the user is facing forward at 0° at the start.

However, the accelerometer signals contain a certain amount of noise. Hence the signals integrated in Eq. (12), i.e. ω_x, ω_z and Ω_r, are noisy. Even with efficient denoising, the residual sensor noise remains challenging for the estimation of the rotation speed magnitude Ω_r. Indeed, in Eq. (3) the absolute rectification of a signal corrupted by a zero-mean noise produces a biased noisy Ω_r. This leads to $\Omega_r \geq 0$ even in the absence of movement. Integrating a biased Ω_r would yield erroneous rotation estimations. To avoid integrating the noise bias when there is no movement, any value of Ω_r that is below a threshold Ω_T is set to 0:

$$\Omega_r = \begin{cases} \sqrt(|\frac{a_{y1} - a_{y2}}{D}|) & \text{if } \Omega_r \geq \Omega_T \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Ω_T is constant in the algorithm described in [18] while it is dynamically controlled in the algorithm proposed in this article, as described in section II-C.1.

C. Upgrades of the Algorithm for the Head Tracking Application

1) Dynamic Definition of the Threshold: Ω_r, as defined in Eq. (3) is always positive, even when the system or the head of the user does not move. The mean value of Ω_r is then defined by the mean and standard deviation of the noise of the sensors a_{y1} and a_{y2}. A non-null Ω_r implies a non-null rotation speed. In order to ensure that Ω_r is null when there is no movement, a threshold Ω_T is used to force any smaller value of Ω_r to be set to 0. i.e. when a steady-state is detected. This value should depend on $\overline{\Omega_r}$, the mean of Ω_r and its standard deviation $\sigma(\Omega_r)$. As the noise is likely to evolve between measures, and during the estimation, it is necessary to dynamically define Ω_T so that the value can be optimally adapted to every situation. The priority is to have a robust computation, as an underestimation of Ω_T would lead to integrate noise in the next step. It is also important to have a reactive computation of Ω_T that adapts to the evolution of Ω_r with a small delay, in order to avoid integrating noise for too many samples nor missing too much of the motion if Ω_T is temporarily overestimated. A satisfying method for defining and updating Ω_T was defined using the following computation. First let K_{Ω_T} be defined as:

$$\begin{aligned} K_{\Omega_T}(n, Nb, Nc) = {} & \overline{\Omega_r}(n - Nb : n - Nc) \\ & + \sigma(\Omega_r(n - Nb : n - Nc))\gamma \end{aligned} \tag{14}$$

where γ is a parameter of the algorithm which value can be set empirically and Nb is the fixed size of the sliding buffer used for this computation. Nc is used to select the first samples of the buffer and can be set in relation to Nb, e.g. $Nc = \frac{3}{4}Nb$. The following conditions are checked at every sample n:

$$\begin{cases} \sigma(\Omega_r(n - Nb : n)) < S_{\Omega_R} \\ \Omega_r(n) > K_{\Omega_T}(n, Nb, Nc) \\ clock \bmod(r_{\Omega_T}) = 0 \end{cases} \tag{15}$$

where S_{Ω_R} is the tolerance on the standard deviation that can be set empirically as an absolute value. Finally $clock$ is the running sample counter and r_{Ω_T} is the refresh rate in samples

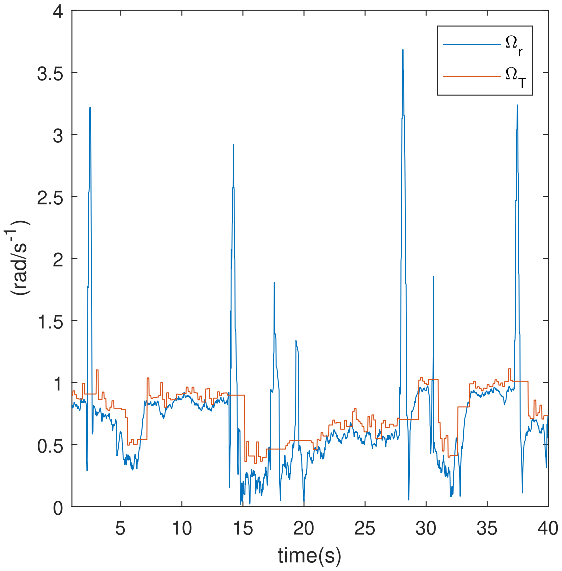


Fig. 2. Example of the computation of Ω_T in relation to the evolution of Ω_r. This computation is the result of Eq. (16) based on the conditions (15), during realistic measurements of head movements (see section. III-A). When the user is steady, Ω_T is set to remain just above Ω_r, and no computation is performed. As soon as a movement is performed (peak in Ω_r), Ω_T remains constant until the next steady state, and the values of Ω_r are used.

used to update $\mathbf{\Omega}_T(n)$, defined such as:

$$\begin{cases} \mathbf{\Omega}_T(n) = K_{\mathbf{\Omega}_T}(n, Nb, 0) & \text{if all conditions (15) true} \\ \mathbf{\Omega}_T(n) = \mathbf{\Omega}_T(n-1) & \text{otherwise} \end{cases} \tag{16}$$

An example of the results obtained for $\mathbf{\Omega}_T$ with this method is displayed in Fig. 2

2) Integration: In practice, the sensor data acquisition is discrete. The various stages of integration could therefore be processed with either a cumulative sum or with the trapezoidal rule. Both approaches have been compared and no significant difference were found. The simpler cumulative sum was therefore chosen for the integration, and Eqs. (8) and (9) therefore become:

$$\omega_z(n) = \frac{1}{f_s}\sum_{n_0}^{n} \alpha_z(k) + \omega_z(n_0 - 1) \tag{17}$$

$$\omega_x(n) = \frac{1}{f_s}\sum_{n_0}^{n} \alpha_x(k) + \omega_x(n_0 - 1) \tag{18}$$

where k varies between the first sample of the buffer n_0 and the last sample n, and f_s is the sampling frequency. Similarly, Eq. (12) in discrete time becomes:

$$yaw(n) = \frac{1}{f_s}\sum_{n_0}^{n} y\dot{a}w(k) + yaw(n_0 - 1) \tag{19}$$

These integration steps introduce some errors, due to the noise of the sensors and the discrete approximations. Consequently, the initial conditions of the two stages of integration should be reset whenever possible. For Eqs. (17) and (18), this corresponds to when the rotation speed of the system $\mathbf{\Omega}_r$ is null. For Eq. (19) this corresponds to when the rotation speed $\mathbf{\Omega}_r$ is null and when the system receives information about its orientation compared to a fixed point in the room.

In the current implementation of the system, the reset of the integrals in Eqs. (17) and (18) is performed with the following method:

$$\begin{cases} \omega_z(n_0 - 1) \text{ is set to } 0 & \text{if } \mathbf{\Omega}_r(n - N_{Init\Omega_T} + 1 : n) = 0 \\ \omega_z(n_0 - 1) & \text{unchanged otherwise} \end{cases} \tag{20}$$

$$\begin{cases} \omega_x(n_0 - 1) \text{ is set to } 0 & \text{if } \mathbf{\Omega}_r(n - N_{Init\Omega_T} + 1 : n) = 0 \\ \omega_x(n_0 - 1) & \text{unchanged otherwise} \end{cases} \tag{21}$$

This means that the integral is reset when no movement is detected during $N_{Init\Omega_T}/f_s$ seconds. The integral of Eq. (19) is only reset at the beginning of the yaw estimation. Otherwise, the last estimated orientation before detection of a steady-state is always kept in a buffer as initial condition of the next integration.

3) Sensor Denoising Using an Adaptive Smoother: Let A_Y, A_X and A_Z be defined as the difference of the accelerations measured by each sensor on the x-, y- and z-axis in Eq. (13), Eq. (6) and Eq. (7) respectively:

$$A_Y = a_{y1} - a_{y2} = a_{r_{y_1}}(D/d_1) \tag{22}$$

$$A_X = a_{x1} - a_{x2} = -a_{r_{x_1}}(D/d_1) \tag{23}$$

$$A_Z = a_{z1} - a_{z2} = -a_{r_{z_1}}(D/d_1) \tag{24}$$

The axis label will be discarded in the following as the concepts developed in this section are independent of the axis. Because the sensors are noisy, only the noisy counterparts $\tilde{a}_1$ and $\tilde{a}_2$ of the true accelerations a_1 and a_2 are available. The noise model can reasonably be assumed as additive:

$$\tilde{A} = \tilde{a}_1 - \tilde{a}_2 = (a_1 + \upsilon_1) - (a_2 + \upsilon_2) = A + \upsilon \tag{25}$$

with the noise components υ_1 and υ_2 orthogonal to each other and to A (i.e. zero mean and uncorrelated ($\mathbb{E}[\upsilon_1\upsilon_2] = \mathbb{E}[\upsilon_1]\,\mathbb{E}[\upsilon_2] = 0$, $\mathbb{E}[A\upsilon_1] = \mathbb{E}[A]\,\mathbb{E}[\upsilon_1] = 0$ and $\mathbb{E}[A\upsilon_2] = \mathbb{E}[A]\,\mathbb{E}[\upsilon_2] = 0$). υ can be interpreted as an average noise signal orthogonal to A (i.e. $\mathbb{E}[A\upsilon] = \mathbb{E}[A(\upsilon_1 + \upsilon_2)] = 0$). The variance of $\tilde{A}$ is:

$$\begin{aligned} \sigma_{\tilde{A}}^2 &= \mathbb{E}[(\tilde{A} - \mathbb{E}[\tilde{A}])^2] = \mathbb{E}[(A - \mathbb{E}[A] + \upsilon)^2] \\ &= \mathbb{E}[(A - \mathbb{E}[A])^2] + \mathbb{E}[\upsilon^2] + 2\,\mathbb{E}[A\upsilon] - 2\,\mathbb{E}[A]\,\mathbb{E}[\upsilon] \\ &= \sigma_A^2 + \sigma_\upsilon^2 = \sigma_A^2 + \sigma_{\upsilon_1}^2 + \sigma_{\upsilon_2}^2 \\ &= \sigma_{a_1}^2 (D/d_1)^2 + 2\sigma_{\upsilon_1}^2 \end{aligned} \tag{26}$$

Compared to a method based on a single signal $\tilde{a}_1$, using $\tilde{A}$ leads to up to 3 dB signal-to-noise ratio improvement for $d_1 = d_2$:

$$SNR_{\tilde{A}} = \frac{\sigma_A^2}{\sigma_\upsilon^2} = \frac{D^2\sigma_{a_1}^2}{2d_1^2\sigma_{\upsilon_1}^2} = \frac{1}{2}\left(\frac{d_1 + d_2}{d_1}\right)^2 SNR_{a_1} \tag{27}$$

The signal $\tilde{A}$ should be further denoised before being processed in the integration stages and prior to compute $\mathbf{\Omega}_T$.

A second reasonable assumption about A and υ can be made: υ is stationary and its power spectral density is relatively uniformly distributed over frequencies. A is a non-stationary

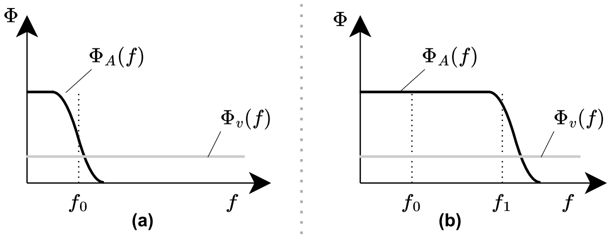


Fig. 3. Power spectral density Φ_A as a function of frequency in slow mode (a), i.e. when the user is nearly static and fast mode (b), i.e. when the user is moving.

band-limited signal with a short-term power spectral density $\Phi_A(f, n)$ upper bounded by a maximal frequency $f_B(n)$ that can vary over time, depending on the head motion. The motion can be qualitatively characterized by two different motion modes: “slow” and “fast”. In the slowest mode, the head is either moving and/or accelerating very slowly or not moving at all. $\Phi_A(f, n)$ is band-limited at $f_B(n) = f_0$, which means that v is the dominant signal above f_0 as qualitatively pictured in Fig. 3(a). In the fastest mode, the head is moving quickly and/or accelerating strongly. $\Phi_A(f, n)$ is smeared over a broader frequency range, delimited by a frequency $f_B(n) = f_1$ with $f_1 \gg f_0$ as qualitatively depicted in Fig. 3(b). A is the dominant signal up to f_1. In particular, compared to slow mode, $\Phi_v(f)$ is negligible compared to $\Phi_A(f, n)$ between f_0 and f_1.

Based on these assumptions, one can expect a good estimation $\hat{A}(n)$ of $A(n)$ using a low complexity smoothing of $\tilde{A}(n)$ made of a first order recursive filter with a 3 dB cut-off frequency $f_c(n)$ than can be adapted at each sample n to track $f_B(n)$, with $f_B(n)$ in the range bounded by f_0 and f_1 for the slowest and fastest modes respectively. This is equivalent to the application of an input power adaptive threshold mechanism between f_0 and f_1: the input power in the spectral band below f_c is passed through, while the input power beyond f_c is stopped with f_c being equal to the frequency f_B at which the input power distribution ends. The heuristic behind this approach is the following: if the detected power between f_0 and f_B cannot be produced by the frequency components of v alone, some participation of A is necessary. Otherwise, the frequency components of v are sufficient to explain the amount of power detected beyond f_B. So, it makes sense to set $f_c(n) = f_B(n)$ as adaptive cut-off frequency. In practice, it is $f_c = \hat{f}_B(n)$ with $\hat{f}_B(n)$ an estimate of $f_B(n)$.

Such a smoothing has to be transparent while the user is moving (fast mode), in order to accurately track the variations during those segments (f_c close to f_1). When the user is nearly static (slow mode), it is desirable to have a more radical smoothing (f_c close to f_0), in particular because one of the goals is to reduce the standard deviation of $\boldsymbol{\Omega}_r$ as much as possible so that the computation of $\boldsymbol{\Omega}_T$ can be optimized to the lowest possible value. Overestimating $\boldsymbol{\Omega}_T$ means that slow motions will not be detected, whereas underestimating $\boldsymbol{\Omega}_T$ means that some noise will be integrated, and thus the final estimation of the yaw will drift.

The estimate $\hat{A}(n)$ of A is obtained via recursive averaging:

$$\hat{A}(n) = \lambda(n)\tilde{A}(n) + (1 - \lambda)\hat{A}(n - 1) \tag{28}$$

where $\lambda(n)$ is the time variable smoothing coefficient that must be adaptive controlled to guarantee a good approximation of $A(n)$ independently of the context (slow or fast mode). $\lambda(n)$ can be derived from $f_c(n)$ [21]:

$$\lambda(n) = \lambda(f_c(n)) = c(n) + \sqrt{c^2(n) - 2\,c(n))} \tag{29}$$

$$c(n) = \cos(2\pi f_c(n)/f_s) - 1 \tag{30}$$

with f_s the sampling frequency. $\lambda(n)$ is bounded by $\lambda_0 = \lambda(f_0)$ and $\lambda_1 = \lambda(f_1)$.

To use the above equation, the adaptive control of $\lambda(n)$ requires the signal $\tilde{A}(n)$ to be transformed into the frequency domain to compute an estimate $\hat{f}_B(n)$ from the evaluation of $\Phi_{\tilde{A}}(f, n)$. To keep the complexity low, it is desirable to realize the adaption control in the time domain. $\lambda(n)$ must be adapted as a function of the time constant $\tau_c(n)$ instead of the cut-off frequency $f_c(n)$:

$$\lambda(n) = 1 - e^{\frac{-1}{\tau_c(n) f_s}} \tag{31}$$

Eq. (31) can be obtained as follows: In the $\mathcal{Z}$ domain, Eq. (28) produces the following transfer function [21]:

$$\mathcal{H}(z) = \frac{\hat{A}(z)}{\tilde{A}(z)} = \frac{\lambda}{1 - (1 - \lambda)z^{-1}} = \lambda \sum_{m=0}^{\infty} (1 - \lambda)^m z^{-m} \tag{32}$$

the last equality being valid because $|(1 - \lambda)z^{-1}| < 1$. The above transfer function can be $\mathcal{Z}$ inverse transformed [21] into the impulse response $h[k]$:

$$h[k] = \lambda \sum_{m=0}^{\infty} (1 - \lambda)^m \delta(k - m) \tag{33}$$

Using either the Impulse Invariance [22]–[24] or the matched $\mathcal{Z}$-Transform [23], [24] methods, one can derive Eq. (31) from Eq. (33).

Similarly to $f_c(n)$, $\tau_c(n)$ has to be adapted to suit the motion at sample n. Instead of using $\tau_c(n) = \hat{\tau}_B(n)$ with $\hat{\tau}_B(n)$ an estimate for $\tau_B(n)$, the time constant $\lambda_c(n)$ that corresponds to $f_B(n)$ is directly obtained using the heuristic procedure described below.

In the slowest mode, there is only little power above f_0. Thus, the variation between the current and the previous observed samples defined as:

$$\Delta\tilde{A}(n) := \tilde{A}(n) - \tilde{A}(n - 1) \tag{34}$$

cannot be large. Indeed, in the slowest mode, we can expect the acceleration to be approximately constant, i.e.:

$$\Delta A(n) := A(n) - A(n - 1) \approx 0 \tag{35}$$

This means that the observed change is approximately the noise variation between two successive samples:

$$\Delta\tilde{A}(n) \approx \Delta v(n) := v(n) - v(n - 1). \tag{36}$$

with $|\Delta v(n)| \sim \sigma_v$. By replacing $A(n - 1)$ by its estimate $\hat{A}(n - 1)$ in Eq. (35), a distance function $d(n)$ can be defined as:

$$d(n) := |\tilde{A}(n) - \hat{A}(n - 1)| \tag{37}$$

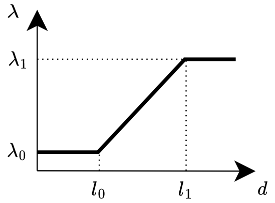


Fig. 4. Mapping function used to adapt the smoothing factor $\lambda(n)$ in relation to the distance $d(n)$.

such that in slow mode, the magnitude of $\Delta v(n)$ can be approximated using this distance $d(n)$:

$$|\Delta v(n)| \approx |d(n)| \tag{38}$$

In fast mode, the acceleration signal contains power in the high frequencies, allowing $A(n)$ to be very different from $A(n-1)$. Hence, the approximation of Eq. (38) is not valid anymore. Slow and fast modes can thus be distinguished from each other by comparing the distance function $d(n)$ to a threshold $l_0 \sim \sigma_v$. Practically, $\lambda(n)$ is mapped from the distance $d(n)$ using a piece-wise linear curve as pictured in Fig. 4 and implemented using the following equation:

$$\lambda(n) = \min(\lambda_1, \max(\lambda_0, (d(n) - l_0)\eta + \lambda_0)) \tag{39}$$

or

$$\lambda(n) = \min(\lambda_1, \max(\lambda_0, (d(n))\eta + \lambda_h)) \tag{40}$$

with

$$\eta = \frac{\lambda_1 - \lambda_0}{l_1 - l_0} \tag{41}$$

and

$$\lambda_h = \frac{\lambda_0 l_1 - \lambda_1 l_0}{l_1 - l_0} \tag{42}$$

where λ_0 and λ_1 are obtained from τ_0 and τ_1, respectively, using Eq. (31). Note that the time constants τ_0 and τ_1 have to be set empirically, trading good denoising performance in slow mode for low-latency tracking ability in fast mode. They are however expected to correspond approximately to their frequency domain counterpart f_0 and f_1, respectively.

Other mapping functions (e.g. arctangent, hyperbolic tangent, sigmoid, etc), could also be used to provide a smoother control of $\lambda(n)$, potentially with larger computational costs.

Finally, Ω_r, α_z and α_x are obtained with the following equations:

$$\Omega_r = \begin{cases} \sqrt(|\frac{\hat{A}_Y}{D}|) & \text{if } \Omega_r \geq \Omega_T \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

$$\alpha_z = -\frac{\hat{A}_X}{D} \tag{44}$$

$$\alpha_x = -\frac{\hat{A}_Z}{D} \tag{45}$$

that replace Eq. (13), Eq. (6) and Eq. (7), respectively.

4) Initialization of the Head-Tracking Algorithm: Because of the human head morphology, when using two sensors placed on the ears of a human head, the axes of the accelerometers may not be aligned with the axis as described in the algorithm. The y axis should be aligned with the interaural axis, the z-axis should be vertical, and the x-axis should be in the forward direction. In order to correct for this, a multi-stage rotation initialization has been implemented. A 12-stages calibration of multiple sensors was proposed in [20]. The method proposed here requires only two stages of calibration.

First, the user is asked to remain upright, looking in front of them, and steady. This is the first steady-state, and helps doing an initial alignment of the z-axis by applying a roll and a pitch correction. The users then need to lower their head, i.e. doing a pitch movement without any roll nor yaw movement, and keep their head down steadily for several seconds. This step allows to estimate the initial yaw of each accelerometer.

5) Steady States Detection for the Calibration: An automatic detection of the calibration states has been developed. It detects the time interval when the head of the user is steady, in the first few seconds of the tracking algorithm. The goal is to trigger the correct computations for each of the two steady states. For this purpose, a sliding buffer of $N_{SteadySTD}$ samples ($N_{SteadySTD}/f_s$ seconds) for each acceleration data is used. The acceleration element exhibiting the highest standard deviation is selected during the first $N_{SteadySTD}$ samples to define a threshold S_{cal} based on this standard deviation and a factor δ determined empirically.

$$S_{cal} = max(\sigma(a_{x1}), \sigma(a_{x2}), \sigma(a_{y1}), \sigma(a_{y2}), \sigma(a_{z1}), \sigma(a_{z2}))\delta \tag{46}$$

After S_{cal} is determined, if the standard deviation of the last $N_{SteadySTD}$ samples remains below this threshold for every acceleration, the head is considered in steady state. A counter is incremented for each new steady state detected, enabling triggering the appropriate computation corresponding to each of the two first steady states.

6) Compensation of the Initial Orientation of Each Sensor: The first step consists in computing the initial pitch and roll angle for each accelerometer individually. This can be done at the start, as soon as the first steady state is detected. For each sensor ($j = 1$ or $j = 2$ for the accelerometer A1 or A2, respectively), the following equations, for which the demonstration can be found in [16], are used for the pitch and the roll, respectively:

$$\theta_j = \frac{180}{\pi} \arctan\left(\frac{a_{xj}}{\sqrt{a_{y,j}^2 + a_{zj}^2}}\right) \tag{47}$$

$$\phi_j = \frac{180}{\pi} \arctan\left(\frac{a_{yj}}{\sqrt{a_{x,j}^2 + a_{zj}^2}}\right) \tag{48}$$

For each sensor, the following rotation matrices are then used to compensate for the initial pitch and roll respectively:

$$R_{y,j}(\theta_j) = \begin{pmatrix} \cos(\theta_j) & 0 & \sin(\theta_j) \\ 0 & 1 & 0 \\ -\sin(\theta_j) & 0 & \cos(\theta_j) \end{pmatrix} \tag{49}$$

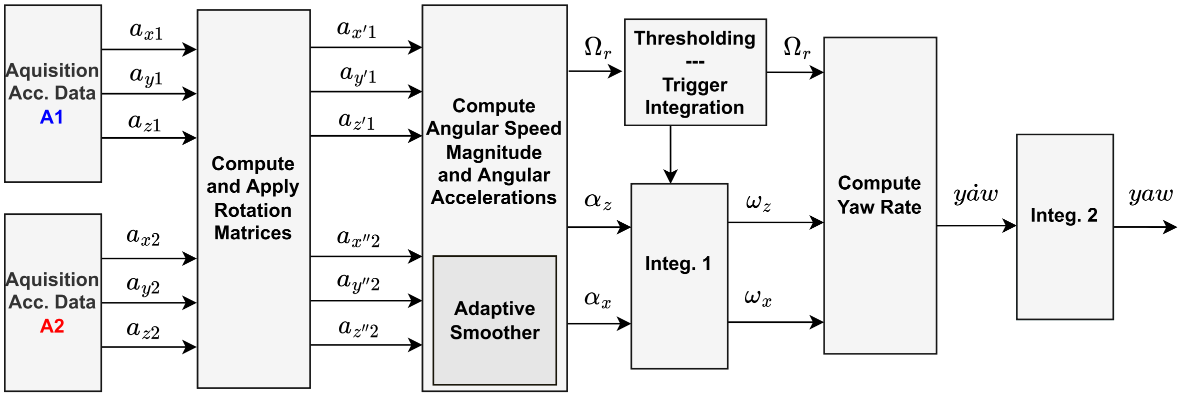


Fig. 5. Simplified block diagram of the real-time implementation, showing the main steps and data. ($a_{x'1}, a_{y'1}, a_{z'1}$) and ($a_{x''2}, a_{y''2}, a_{z''2}$) denote the accelerations of the sensors A1 and A2 after applying their respective rotation matrices.

$$R_{x,j}(\phi_j) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_j) & -\sin(\phi_j) \\ 0 & \sin(\phi_j) & \cos(\phi_j) \end{pmatrix} \tag{50}$$

Once the first steady-state is detected, the means $\overline{a}_{xj,steady1}$, $\overline{a}_{yj,steady1}$, $\overline{a}_{zj,steady1}$ for each sensor j are calculated. The means are computed over $N_{SteadySTD}$ samples of smoothed sensor data. For this specific part of the algorithm, the sensor data is smoothed using a basic exponential smoothing, independent from the adaptive smoother as described in II-C.3. A fixed smoothing factor is used, with a value $\lambda_R = 0.05$ ($\tau = 0.39\ s$ with $f_s = 50Hz$), and the smoothed accelerations for each axis i ($i = x, y$ or z) and each sensor j are computed as:

$$a_{ij,lp}(n) = \lambda_R a_i(n) + (1 - \lambda_R) a_{ij,lp}(n-1) \tag{51}$$

The roll and pitch value of both sensors are then computed using the means for each sensor. In order to correct the yaw ψ_j of each sensor, a second steady-state is required. When users rotate their head so as to have a non-null new pitch, but no new roll nor yaw, if the yaw of one of the sensor is non-null, the y acceleration of this sensor will also be non-null. Therefore when a second steady-state is detected, the algorithm assumes the user moved only in pitch (around the y-axis). It computes the means $\overline{a}_{xj,steady2}$, $\overline{a}_{yj,steady2}$, $\overline{a}_{zj,steady2}$ of each sensor j on the last $N_{SteadySTD}$ samples, and tests all possible yaw values for each sensor between $-30°$ and $+30°$ by steps of $0.1°$. Potential adjusted values of $\overline{a}_{yj,steady2}$ are tested for each of the possible yaw values, such as:

$$\psi_j = \text{argmin}_{\psi} \overline{a}_{xj,steady2} \sin(\psi) + \overline{a}_{yj,steady2} \cos(\psi) \tag{52}$$

The values of ψ_j are then used in the rotation matrix associated with the initial yaw of each sensor:

$$R_{z,j}(\psi_j) = \begin{pmatrix} \cos(\psi_j) & -\sin(\psi_j) & 0 \\ \sin(\psi_j) & \cos(\psi_j) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{53}$$

In order to increase the precision of the estimations, it is possible to run this estimation in a loop. After the rotation matrix $R_{z,j}(\psi_j) R_{y,j}(\theta_j) R_{y,j}(\phi_j)$ is applied to the $\overline{a}_{steady1}$ and $\overline{a}_{steady2}$ values, the roll, pitch, and yaw correction can be re-estimated and added to the initial estimations. This can be done in a loop until the values of $\overline{a}_{steady1}$ and $\overline{a}_{steady2}$ are below a predefined threshold.

D. Real-Time Implementation

The real-time version of the algorithm was implemented on Simulink (The Mathworks, Portola Valley, CA, USA). A simplified descriptive block diagram is depicted in Fig. 5. The sampling frequency was fixed to 50 Hz. This value was determined mainly by the hardware implementation of the prototype, and is sufficient to cover the needed range of accelerations. The algorithm uses sliding block processing. Most of the processing is performed sample by sample, inside each block. The size of the block was therefore reduced to 1 sample, keeping the latency low. No overlapping blocks are necessary in the implemented algorithm.

III. Evaluation of the Algorithm

In order to evaluate the reliability of the algorithm, a database of head movements was first recorded, on which the algorithm was then tested. Next, the evaluation was performed by comparing the obtained estimation against the performance of a complete state-of-the art Euler angles tracking algorithm.

A. Database

The aim of this database is to display a large and realistic range of head movements that can occur in conditions where the algorithm could be used. It was therefore chosen to measure data from subjects, doing a task where they would not limit their head movements to the horizontal plane.

1) Setup and Task: The setup consisted of 5 loudspeakers located at −90°, −45°, 0°, 45° and 90° around the participant, at a distance of 2 m, as shown in Fig. 6.

For the data acquisition, HIs with embedded accelerometers were placed over the pinna of the participants. A headband was used to carry the small circuit boards used to process the data of the sensors. The cable linking the HIs to the circuit board and the circuit board to the computer were chosen to be thin and flexible to avoid unwanted tension forces to be applied

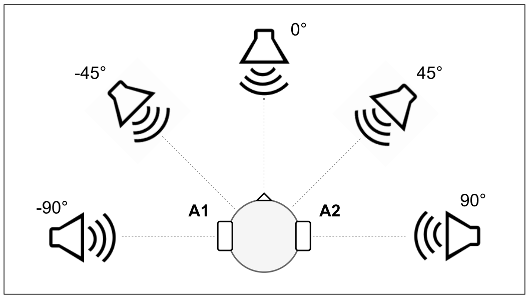


Fig. 6. Schematic representation of the setup used for recording the database. The user is equipped with the HIs with embedded accelerometers and the reference IMU/AHRS device. 5 loudspeakers are located around the user.

on the HIs. The device fitting is shown with a close-up shot in Fig. 7. The subject was also equipped with a NGIMU[1] IMU/AHRS device.

The acceleration data from both sensors was retrieved via USB. Euler angles measured with the IMU/AHRS device were recorded via USB as well, and served as a reference measurement. All the data was acquired and recorded on a dedicated Simulink model.

The listeners were asked to point their head toward the loudspeaker which emitted a sound (a speech sample pronouncing the number designating the loudspeaker). Then they had to wait in this position until a word is pronounced through the same loudspeaker. Subsequently they had to write down the word on a piece of paper on the table. Then they could move their head back to the last loudspeaker emitting the sound, and wait for the next number and word. The total duration of the measure was about 120 s and included 12 target positions.

For this database, 11 participants were fitted with the prototype and the reference IMU/AHRS sensor, and followed the protocol mentioned above. The instruction was given to the participants to turn their head as naturally as possible, as they would be attending a meeting with several speakers around them.

For the calibration, visual marks were placed in front of the subject and on the table for the first and second steady-states respectively. To ensure a precise movement from the subject, a visual feedback of the roll angle, as measured by the IMU/AHRS device, was provided to them on a screen. When the subject performed the second steady state, with a targeted pure pitch motion, they were instructed to keep this roll value between $-2°$ and $+2°$, which was empirically found to be sufficient precision. The participants managed to perform the calibration with those constraints after a maximum of 3 trials.

B. Evaluation Method

A batch evaluation was conducted using the measures and various settings of the algorithms. The goal was first to assess the effect of various parameters used in the algorithm and look at the best performance achievable for each measurement.

The evaluation was achieved by comparing the estimation obtained with the proposed method to the results obtained with a state-of-the-art fusion algorithm for tracking Euler angles. The latter reference was taken from the Euler angles estimation performed by the NGIMU device, which is based on the implementation described in [25]. More precisely, a score was computed using the root-mean-square error (RMSE) between the reference yaw obtained with the NGIMU device and the estimation obtained from the proposed algorithm using two hearing devices with embeded accelerometers. This corresponds to:

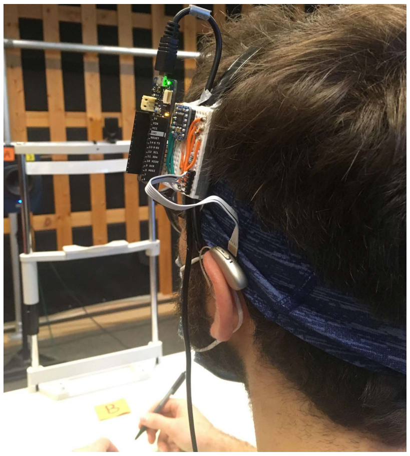


Fig. 7. Fitting of the HIs with embedded accelerometers.

TABLE I
LIST OF THE TESTED PARAMETERS AND THEIR VALUES

Parameter	Tested values
D	0.15 and 0.17 (m)
δ	2 and 4
γ	4, 5.5, and 7
Nb	16, 24, and 32 (samples)
Nc	3/4Nb and 1/2Nb (samples)
S_{Ω_R}	0.07, 0.08, 0.09, and 0.10
r_{Ω_T}	1/4Nb and 1/2Nb (samples)

$$RMSE = \sqrt{\frac{1}{N_m}\sum_{k=1}^{N_m}\left(yaw_{est}(k) - yaw_{ref}(k)\right)^2} \qquad (54)$$

where N_m is the length of the measurement in samples, yaw_{est} is the estimated yaw, and yaw_{ref} is the reference. The choice of the tested parameters and their range of interest were set empirically and are shown in Tab. I. The factorial design with those factors and levels leads to total of n = 576 combinations.

C. Results

During the measure, the USB cable was attached to the headband and could potentially apply some tension on the sensors if the participant involuntarily pulled the cables. Participants were wearing face masks, which made it complicated to have the HI fixed behind the ear firmly enough. Consequently, the HIs could sometimes move, in which case the calibration would not be adequate anymore, leading to drift in the yaw estimation. Hence the occurrences of displacement had to be monitored both with feedback from the participants and by scanning the acceleration data and detect unexpected behaviors. For that reason, in order to maximize the chances

[1] https://x-io.co.uk/ngimu/

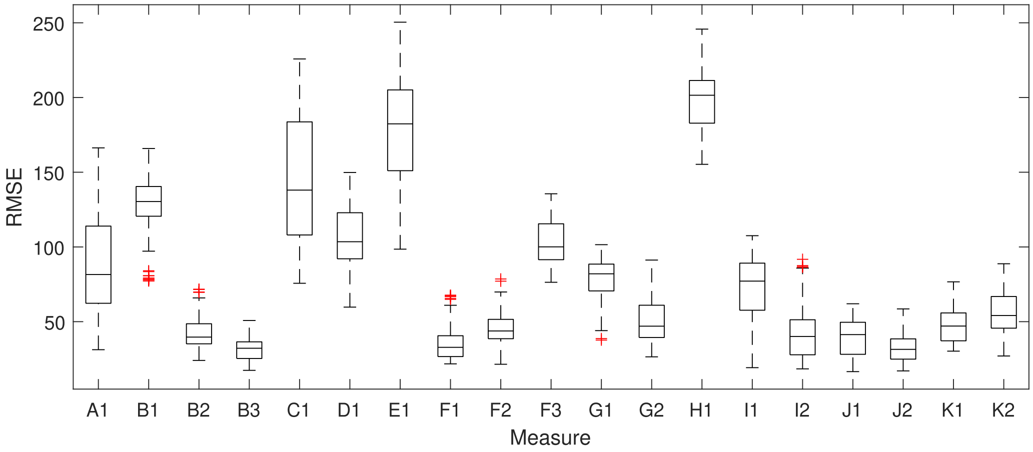


Fig. 8. Boxplots of the RMSE as a function of the measurement. The RMSE is computed from the estimation in comparison to the state-of-the-art fusion algorithm used in the reference IMU/AHRS device. The letter corresponds to the anonymized participant, and the number is the repetition in case of several usable measure. n = 576 combinations of the parameters were performed for each measure.

of obtaining at least one measure per participant with no hardware issues, two to three measures were recorded per participant. Out of the total of 30 measures, 19 were kept, for which no obvious hardware issue causing the sensors to move was detected.

For those 19 measurements and for all tested combinations of parameters, the results of the RMSE of the estimation compared to the state-of-the-art algorithm are displayed in Fig. 8. Examples of various yaw estimations on different measures are depicted in the following figures. First two measures, associated with lower RMSE values, are shown in Fig. 9(a) and 9(b). In Fig. 9(a), it can be seen that the main movements are detected in the correct direction and are well tracked while the subject is moving in azimuth. While the subject performs the writing task, the azimuth might not always be tracked. The same observations can be made in Fig. 9(b), while some local errors decrease the performance.

Fig. 9(c) depicts a case for which most of the movements are underestimated, and a part is not detected at all. This could be explained by head movements which are too slow. In this case, the acceleration values are too small and thus, a part or the totality of the movement is not properly detected and set to 0 (see Eq. (43)) as the magnitude of the acceleration is too close to the magnitude of noise. Finally an example of a low performance of the algorithm is depicted in Fig. 9(d). In this case a drift can be observed resulting in a large shift at the end of the measure. Some sign inversions sometimes occur at the end of the movement, which yields large errors. An hypothesis can be made in this case that the initial yaw of each sensor was poorly estimated during the calibration, which results in an overestimation for a part of the movements in one direction, and underestimation for the other direction, resulting in a drift over time.

IV. Limitations of the Algorithm

A. Measure Dependent Performance

In a view to assessing the possibility to find an optimal setting which might yield a performance close to the state-of-the-art method on most measurements, a factorial analysis was conducted, as described in [26]. For seven of the measures (A1, B1, C1, D1, E1, F3, H1), it is considered that the errors are too large to be included in the data used to find a common set of parameters. It is hypothesized that sensors displacement might have occurred during these measurements, and that those were not visible when checking the acceleration data. This could happen if a displacement occurred during a movement. In this case, the displacement-induced shift in acceleration would not be visible while the acceleration is changing because of the head movement. Another hypothesis is that the calibration phase was not achieved precisely enough, notably to compute the initial yaw of the HI behind the ear. Hence, it was decided to investigate the possibility of an optimal parametrization with the twelve other measures.

The factorial analysis method described in [26] assumes that the dependent variable Y of the experiment is determined by the experimental conditions, and can be approximated by a polynomial function with second order interaction terms. An example is given below for three hypothetical parameters x_1, x_2 and x_3:

$$\begin{aligned}Y &= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_{12}x_1x_2 \\ &\quad + b_{13}x_1x_3 + b_{23}x_2x_3 + residual \qquad (55)\end{aligned}$$

b_0 corresponds to the mean value of all possible RMSE over all parameter combinations for a single measurement. The parameters and their two levels can be summarized in a factorial table, where each line is a unique combination of all parameters. Then a sign table is built from this table, for each column (associated with one parameter), a "−" is associated with the minimum value and a "+" is associated with the maximum value. The coefficients b_1, b_2...are evaluated by either adding or subtracting the value of the response (in this case the RMSE value) for each line based on the signs in their corresponding columns from the factorial sign table. The result is divided by the total number of observations. Once the coefficients b_0, b_1, b_2...are evaluated, the function describes qualitatively how the experimental variables and their interactions influence the response. This analysis was conducted for the twelve selected measurements and the seven

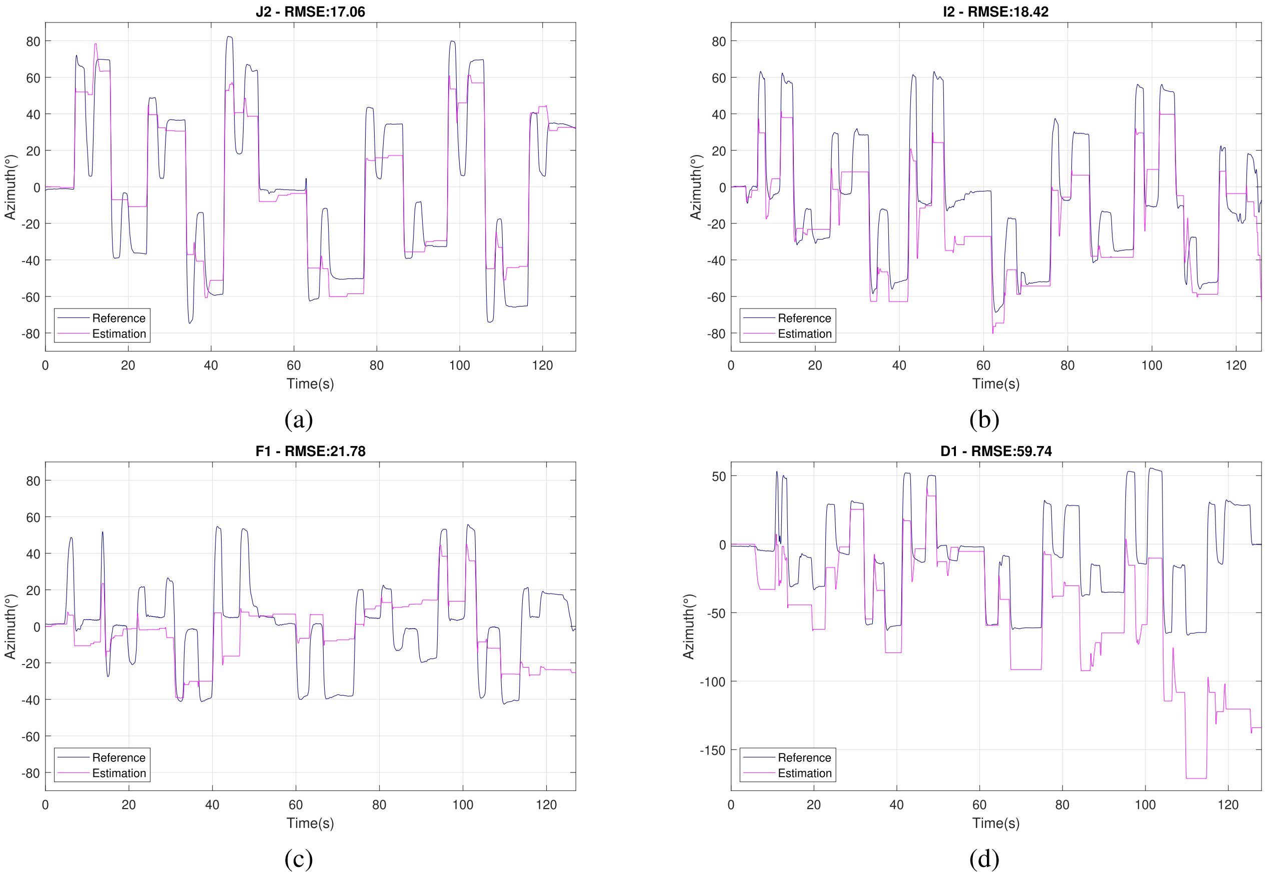


Fig. 9. Examples of various degrees of performance of the algorithm during a realistic task. The plots show the head yaw estimations and the related references obtained with the state-of-the-art algorithm of the IMU/AHRS device for several measures with different participants and their best setting. (a) Illustrates an example of a performance of the algorithm that is close to that of the state-of-the-art method. (b) Shows an example of an overall acceptable estimation with small errors yielding local mismatches with the state-of-the-art method. (c) Illustrates a scenario where most of the movements are partially or not detected. (d) Shows the worst type of performance, where large errors yield a general drift of the estimation toward one direction.

TABLE II
COEFFICIENTS WEIGHTS OF THE SEVEN PARAMETERS USED IN THE FACTORIAL ANALYSIS

Parameter	Mean coefficient	STD coefficient
D	-0.084	3.85
δ	0.53	1.47
γ	1.96	5.16
Nb	-0.17	6.85
Nc	-0.032	0.96
S_{Ω_R}	-0.12	2.07
r_{Ω_T}	-0.51	1.87

parameters of Tab. I were used for the analysis. The minimum and maximum levels of the table are used for the two levels in the analysis. These parameters were combined in a full factorial design, leading to a total of $2^7 = 128$ combinations. A batch evaluation was run for all the measurements with all the 128 parameters combinations. The mean b_0 and the coefficients for the main effects of the seven parameters for each of the twelve selected measures are displayed in Fig. 10. The means and standard deviations across measures for the main effect of the seven parameters are reported in Tab. II. The interactions are not displayed for conciseness.

Those results indicate that the largest influence in the range of tested values is due to γ, Nb and D. It also shows that the interaction coefficient between γ and Nb is important as well ($mean = 0.52, STD = 4.21$). However, no clear trend can be observed about which direction those parameters tend to affect the RMSE value, as the STD is large and the mean mostly centered around 0.

The results show that with this algorithm, a performance close to the state-of-the-art method can be achieved for certain of the measurements. The attempts to find a satisfying tuning that could work on every measure revealed that it is difficult at this point to find a unique set of parameters which might improve the performance of all measures. In particular, Nb, γ and D have the largest influence in the range of the tested values, but a change in certain direction in those values might improve the performance for some measurements and decrease it for other measurements. This indicates that the algorithms could benefit from a dynamic tuning of those three parameters depending on some characteristics of the accelerations retrieved from the sensors.

B. Other Limitations and Related Perspectives for Improvement

The acquisition of the data revealed that the algorithm is sensitive to displacements of the sensors. Indeed, if one of the sensors is displaced after calibration, the initial orientation

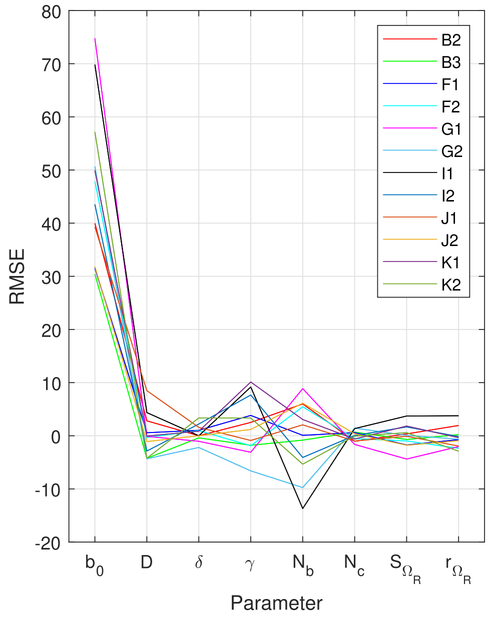


Fig. 10. Mean b_0 and main effects for the seven parameters (D, δ, γ, N_b, N_c, S_{Ω_R} and r_{Ω_T}) obtained from the factorial analysis for the twelve selected measurements.

of the sensors as computed during initialization becomes erroneous. This means that a differential is generated between the two sensors, and this offset might be interpreted as a constant speed movement. The dynamic definition of $\boldsymbol{\Omega}_T$ as described in Eq. (16) should prevent to drift during static parts. Nevertheless, the differential created by a displacement of the sensor might result in overestimation for certain movements and underestimation of other movements which will create an accumulation of error with time. The recording of the data with the prototype included several hardware constraints (cables in particular) which might have made the measures particularly sensitive to this kind of issue, compared to HIs that would be placed behind the ears without being attached to other devices, and might be less prone to displacement.

The integration errors that might occur during the estimations are likely to accumulate over time, and generate a shift of the overall estimation. The current implementation would benefit from an absolute reference in the environment where it is used, that could punctually help re-defining the initial condition of the second stage of integration. HIs being equipped with two microphones per device, it is e.g. imaginable to use direction-of-arrival (DOA) algorithms with sound emitted from a known position in the room.

In the theory behind the equations of this algorithm, the sensors are supposed to be approximately aligned with the center of rotation of the head. In reality, this is dependent on the morphology of the head of the user. Further investigations and developments should study the influence of this parameter in the results.

V. Conclusion

In this article, a method was proposed to estimate the azimuth orientation of a human head in the horizontal plane, using two 3-axis accelerometers only. The method was evaluated with measures of realistic movements made by human subjects, and acquired with the accelerometers of HIs.

The results are promising and a performance close to a state-of-the-art algorithm can be achieved for a part of the measures as long as certain parameters are set individually, as revealed by a factorial analysis and a batch analysis investigating the effects of several parameters. The algorithm is sensitive to sensor displacement and wrong estimation of the initial orientation of the sensors, which both yield poor results. With the current implementation, it is not possible to find a unique set of the fixed parameters that would yield good performance for every measurements. This encourages to investigate the possibility to tune dynamically some of them, based on the acceleration data.

References

[1] E. Branda and T. Wurzbacher, “Motion sensors in automatic steering of hearing aids,” *Seminars Hearing*, vol. 42, no. 3, pp. 237–247, 2021.

[2] J. R. Burwinkel, B. Xu, and J. Crukley, “Preliminary examination of the accuracy of a fall detection device embedded into hearing instruments,” *J. Amer. Acad. Audiol.*, vol. 31, no. 6, pp. 393–403, Jun. 2020.

[3] M. Rahme, P. Folkeard, and S. Scollie, “Evaluating the accuracy of step tracking and fall detection in the starkey livio artificial intelligence hearing aids: A pilot study,” *Amer. J. Audiol.*, vol. 30, no. 1, pp. 182–189, Mar. 2021.

[4] M. Froehlich, E. Branda, and K. Freels, “New dimensions in automatic steering for hearing aids: Clinical and real-world findings,” *Hearing Rev.*, vol. 26, no. 11, pp. 32–36, 2019.

[5] S. C. Voss, K. Pichora-Fuller, A. Pereira, J. Seiter, N. Guindi, and J. Qian, “Evaluating the accuracy of motion detection using a behind-the-ear sensor,” in *Proc. Annu. Sci. Technol. Conf. Amer. Auditory Soc.*, Scottsdale, AR, USA, 2020, p. 1.

[6] S. C. Voss *et al.*, “Evaluating the benefit of hearing aids with motion-based beamformer adaptation in a real-world setup,” *Int. J. Audiol.*, pp. 1–13, Aug. 2021, doi: 10.1080/14992027.2021.1948120.

[7] D. R. Begault, E. M. Wenzel, and M. R. Anderson, “Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source,” *J. Audio Eng. Soc.*, vol. 49, no. 10, pp. 904–916, 2001.

[8] G. Courtois, V. Grimaldi, H. P. Lissek Estoppey, and E. Georganti, “Perception of auditory distance in normal-hearing and moderate-to-profound hearing-impaired listeners,” *Trends Hearing*, vol. 23, Nov. 2019, Art. no. 2331216519887615.

[9] V. Grimaldi, G. Courtois, L. Simon, and H. Lissek, “Externalization of virtual sounds using low computational cost spatialization algorithms for hearables,” Forum Acusticum, Lyon, France, Tech. Rep., 2020, pp. 917–921, doi: 10.48465/fa.2020.0461.

[10] E. Hendrickx, P. Stitt, J.-C. Messonnier, J.-M. Lyzwa, B. F. Katz, and C. de Boishéraud, “Influence of head tracking on the externalization of speech stimuli for non-individualized binaural synthesis,” *J. Acoust. Soc. Amer.*, vol. 141, no. 3, pp. 2011–2023, Mar. 2017.

[11] S. Li, E. Jiaxiang, R. Schlieper, and J. Peissig, “The impact of trajectories of head and source movements on perceived externalization of a frontal sound source,” *AES Convention*, vol. 144, p. 9988, May 2018.

[12] J. Zhao, “A review of wearable IMU (inertial-measurement-unit)-based pose estimation and drift reduction technologies,” *J. Phys., Conf.*, vol. 1087, Sep. 2018, Art. no. 042003.

[13] M. Kok, J. D. Hol, and T. B. Schon, “Using inertial sensors for position and orientation estimation,” *Found. Trends Signal Process.*, vol. 11, nos. 1–2, p. 153, 2018.

[14] A. Ferlini, A. Montanari, C. Mascolo, and R. Harle, “Head motion tracking through in-ear wearables,” in *Proc. 1st Int. Workshop Earable Comput.*, Sep. 2019, pp. 8–13.

[15] I. Adnanul, “Detecting head movement using gyroscope data collected via in-ear wearables,” M.S. thesis, Faculty Inf. Technol. Elect. Eng., Univ. Oulu, Oulu, Finland, 2021.

[16] M. Pedley, *Tilt Sensing Using Linear Accelerometers*, document AN3461, Freescale Semiconductor, 2013.

[17] W. Chee, “Yaw rate estimation using two 1-axis accelerometers,” in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2005, pp. 423–428.

[18] *Using Two Tri-Axis Accelerometers for Rotational Measurements*, document AN 019, KIONIX, 2015.

[19] U. Röijezon *et al.*, "Kinematics of fast cervical rotations in persons with chronic neck pain: A cross-sectional and reliability study," *BMC Musculoskeletal Disorders*, vol. 11, pp. 1–10, Dec. 2010.

[20] K. Papafotis and P. P. Sotiriadis, "Multiple accelerometers and magnetometers joint calibration and alignment," *IEEE Sensors Lett.*, vol. 4, no. 3, pp. 1–4, Feb. 2020.

[21] R. G. Lyons, *Understanding Digital Signal Processing, 3/E*. London, U.K.: Pearson, 2012.

[22] A. V. Oppenheim and R. W. Schafer, *Discrete Time Signal Processing*. London, U.K.: Pearson, 2010.

[23] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*. New York, NY, USA: MPC, 1992.

[24] J. O. Smith, *Physical Audio Signal Processing: For Virtual Musical Instruments and Audio Effects*. Stanford, CA, USA: Stanford Univ., 2010.

[25] S. Madgwick, "AHRS algorithms and calibration solutions to facilitate new applications using low-cost MEMS," Ph.D. thesis, Univ. Bristol, Bristol, U.K., 2014.

[26] T. Lundstedt *et al.*, "Experimental design and optimization," *Chemometrics Intell. Lab. Syst.*," vol. 42, nos. 1–2, pp. 3–40, 1998.

Vincent Grimaldi received the M.Sc.Eng. degree in bioengineering from the ENSAM ParisTech, Paris, France, the M.Sc. degree in sciences and technologies "Parcours ATIAM–acoustics, signal processing, and computer science applied to music" from Sorbonne University, IRCAM, Telecom ParisTech, Paris, and the Ph.D. degree from the LTS2 Signal Processing Laboratory, Acoustic Group, EPFL, Lausanne, Switzerland, in 2022, working mainly on the enhancement of auditory spatial perception in the context of hearing aids. His main research interests include spatial hearing, psychoacoustics in general, as well as audio signal processing.

Laurent S. R. Simon received the Ph.D. degree in perceptual cues-based sound reproduction on the horizontal plane from the Institute of Sound Recording, University of Surrey, in 2011. His subsequent research work has been focusing on spatial auditory perception and processing, ranging from audio source separation to perceptual attributes for the evaluation of head-related transfer functions, applied either to consumer audio or to hearing research. Since 2021, he has been employed at Sonova AG, where he is a part of the Research and Development Team.

Miquel Sans received the M.Sc. degree in electrical engineering from the ETH, Zürich, Switzerland, in 2002, with a specialization in signal processing and information theory. In 2002, he joined the Electronic and Signal Processing Laboratory, EPFL, Neuchâtel, Switzerland, where he developed solutions for embedded video compression. From 2006 to 2016, he worked as a DPS Engineer with the William Demant Holding Group (Oticon, Bernafon, Sonic), Bern, Switzerland, where he designed denoising and hearing loss compensation algorithms. Since 2017, he has been working with Sonova Communications, Murten, Switzerland, in the fields of speech enhancement and spatial audio processing.

Gilles Courtois received the M.Sc./M.Eng. degree from the National Institute of Applied Science (INSA) Lyon, France, in 2012, and the Ph.D. degree in hearing technologies from the EPFL in 2016. During his academic path, he worked on the topic of spatial hearing preservation/recreation for hearing aid applications. Since 2018, he has been with Sonova AG and has been working on multiple hearing aid and consumer audio features, including frequency compression, audio stream spatialization, and audio processing with variable latency. In 2022, he was given an Hearing Performance Expert title, holding the topics of minimal-to-mild hearing losses and spatial hearing integrity.

Hervé Lissek received the M.Sc. degree in fundamental physics from the Université Parix XI, Orsay, France, in 1998, and the Ph.D. degree in physics, with a specialization in acoustics from the Université du Maine, Le Mans, France, in 2002. He was a Postdoctoral Researcher at the EPFL from 2003 to 2005. Since 2006, he has been leading the Acoustic Group, EPFL (affiliated since 2015 to the Signal Processing Laboratory LTS2). His current research interests include electroacoustic absorbers, acoustic metamaterials, signal processing for microphone arrays, and binaural hearing aids. He has also been elected as the Vice President of the Swiss Acoustical Society and the French Acoustical Society, and also the Swiss Representative of the European Acoustics Association, the International Institute of Noise Control Engineering, the International Commission for Acoustics, and the Aeroacoustics Specialists Committee of the Council of European Aerospace Societies.