Meyer-Kahlen, Nils; Kastemaa, Miranda; Schlecht, Sebastian J.; Lokki, Tapio

Measuring Motion-to-Sound Latency in Virtual Acoustic Rendering Systems

Published in:
AES: Journal of the Audio Engineering Society

DOI:
10.17743/jaes.2022.0089

Published: 03/06/2023

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Meyer-Kahlen, N., Kastemaa, M., Schlecht, S. J., & Lokki, T. (2023). Measuring Motion-to-Sound Latency in Virtual Acoustic Rendering Systems. *AES: Journal of the Audio Engineering Society*, *71*(6), 390-398. https://doi.org/10.17743/jaes.2022.0089

Measuring Motion-to-Sound Latency in Virtual Acoustic Rendering Systems

NILS MEYER-KAHLEN, MIRANDA KASTEMAA,
SEBASTIAN J. SCHLECHT TAPIO LOKKI

Abstract

Few studies that employ virtual acoustic rendering systems accurately specify motion-to-sound latency. To make such assessments more common, we present two methods for latency measurements using either impulsive or periodic movements. The methods only require hardware available in every acoustics lab: a small microphone and a loudspeaker. We provide open-source tools that implement analysis according to the methods. The methods are evaluated on a high-quality optical tracking system. In addition, three small trackers based on inertial measurement units were tested. The results show the reliability of the method for the optical system and the difficulties in defining the latency of IMU-based trackers.

1 INTRODUCTION

Virtual acoustic rendering requires real-time adaptation to the listener's movement as the relative timing, level, and direction of the direct sound, early reflections, and late reverberation depend on the listener's position [1]. A listener should be able to rotate their head with three Degrees-of-Freedom (3DoF) or even move in the three Cartesian coordinates of space (6DoF) while maintaining the impression that virtual sound sources rendered via headphones remain static at their desired position [2].

Spatial audio tools for implementing dynamic rendering have become more accessible in recent years. Many auralization VST plugins, such as [3, 4, 5] and [1],[2] can be hosted either in a digital audio workstation (DAW), or in real-time environments such as Pure Data or Cycling 74 MAX MSP. Audio renderers for virtual reality such as Steam Audio[3], Oculus Spatializer[4], and Dear VR[5] also include position- and direction-dependent acoustic rendering, some including reflections and occlusion. See [6] for a recent review of different plugins. Such tools have enormous advantages in terms of versatility of use, and the host software's arranging, routing, and automation capabilities facilitate the design of virtual acoustic experiences and perceptual experiments for acoustic research. See [7, 8, 9, 10, 11] for recent examples of various experiments performed at different labs, using different plugins.

Similarly, tracking technologies to realize dynamic binaural rendering have become more accessible and diverse in recent years. Motion tracking systems like those commercialized by OptiTrack [6] represent high-quality systems, but also smaller and less expensive head trackers based on inertial measurement units (IMUs) are widely in use, as for example the Supperware Head Tracker 1 [7] or the open source MrHeadTracker [12]. Furthermore, experiments are more and more often performed using the built-in tracking capabilities of head-mounted displays (HMDs), for example, in [13].

However, what has not become more accessible are tools to assess a vital performance metric of such systems: motion-to-sound latency. It, in turn, depends on the latency caused by the tracking and transmitting tracking data, the rendering latency caused by the rendering software, and the audio output latency of the digital-to-analog converter (DAC); see Fig. 1 for an overview. For instance, in perceptual experiments using dynamic rendering, the motion-to-sound latency needs to be sufficiently low not to impact the listener's experience. It is assumed that a motion-to-sound latency of less than 60 ms is imperceptible in most scenarios [14]. Methods for measuring latency are not widely available, even though studies that examine perceptual thresholds of latency exist [14, 15]. For this reason, some perceptual studies could only assess relative differences [16]. Further, it has not yet become a standard practice to determine latency with new listening experiments using dynamic rendering.

In this work, we present and evaluate two simple methods for measuring motion-to-sound latency, i.e., the time it takes for a movement of the listener to affect the rendered output. The first method involves tapping the tracked object impulsively. This method has been used in a recent study [17]. The second method uses the periodic movement of a pendulum.

[1] http://www.matthiaskronlachner.com/?p=2015
[2] https://plugins.iem.at/
[3] https://valvesoftware.github.io/steam-audio/
[4] https://developer.oculus.com/resources/audio-intro-spatialization/
[5] https://www.dear-reality.com/
[6] https://optitrack.com/
[7] https://supperware.co.uk/

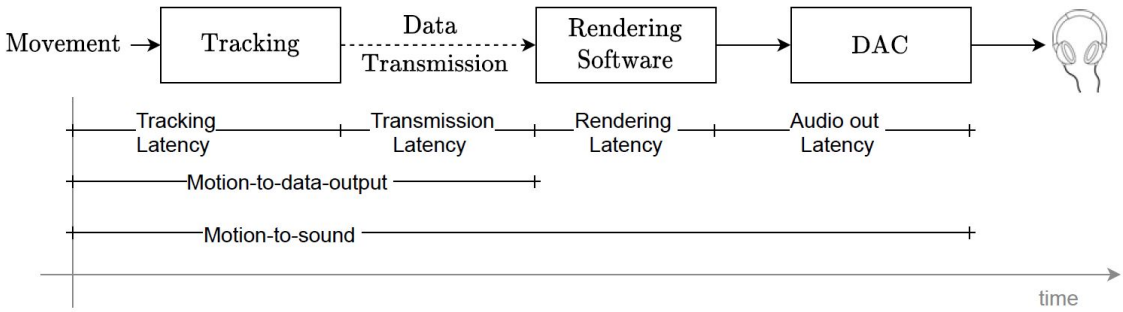


Figure 1: Factors that contribute to motion-to-sound latency. The motion-to-data output latency is measured using the plugin introduced in Sec. 2.2. Motion-to-sound measurements are described in Sec. 2.3 and 2.4.

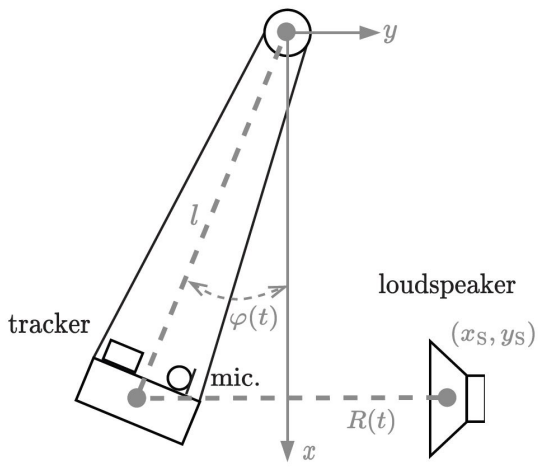


Figure 2: Schematic view of the pendulum setup. The coordinate system (x, y) is centered at the pendulum pivot at $(0, 0)$ for a common and simple representation of the movement in Sec. 3.1.

As opposed to other realizations of pendulum-based methods [18, 19], and those using optical switches and oscilloscopes [20], we only use acoustical components available in every acoustics lab, and we provide open-source software tools to analyze the measurements [8]. Additionally, we measure what we term the motion-to-data-output latency and introduce a new VST plugin for this purpose, which directly writes motion data to an audio file without audio processing and DAC [9].

The methods and the test plugin are explained in Sec. 2. Sec. 3 shows theoretical verifications and presents measurements on an OptiTrack-based rendering system and results for IMU trackers. The latter highlights the difficulty of even defining latency for devices that internally perform sensor fusion involving prediction. Sec. 4 concludes the report.

[8] https://github.com/ahihi/latency-analyzer
[9] https://github.com/ahihi/SPARTA/tree/natnet-integration

2 METHODS

The proposed methods employ a setup in which the tracked object is suspended in the air. Like this, it can easily be moved away from a resting position, which is important for the impulsive method presented. Once moved, the tracker oscillates back and forth, which is required for the second, periodic method.

2.1 Hardware Setup

A schematic view of the setup used for both methods is shown in Fig. 2. A weight is suspended from a horizontal bar at length l. A microphone is attached to the weight. Tracking markers or the positional tracker under test are also affixed to the weight. Furthermore, a loudspeaker is placed in front of the pendulum at position (x_S, y_S). The distance between the loudspeaker and the weight at time t is denoted by $R(t)$. The time-varying angular displacement from the resting position is $\varphi(t)$. The microphone and the loudspeaker are connected to an audio interface that is connected to the computer running the rendering software under test. Additionally, the output of the sound renderer is looped back to another input of the same audio interface.

For the impulsive method, the weight is tapped with another object to move away from its resting position. For the periodic measurement method, the loudspeaker emits a stationary signal such that the signal envelope recorded by the microphone follows the swinging movement of the pendulum. The signal amplitude is at its highest when the weight is closest to the loudspeaker and decreases monotonically as the weight moves toward the other side. Assuming that the loudspeaker is a point source, the recorded amplitude follows the $1/R$ law. If the loudspeaker is placed too far away, the change in amplitude may become too small to be reliably measured. At close distances, the level can become a more complicated function of the distance due to the microphone entering the near-field of the driver, where the sound-field is more complex, and interference effects can

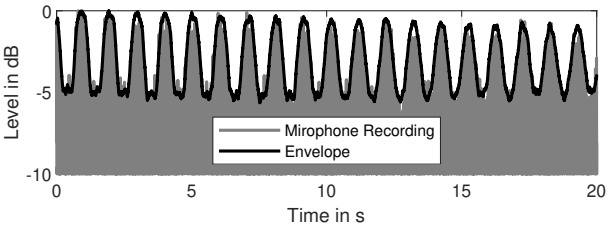


Figure 3: Example of a recorded microphone signal and the extracted envelope during movement of the pendulum. The modulation depth is about 5 dB. There is slight clipping and ripple, which is within 0.5 dB.

occur. Additionally, interactions between the box and the loudspeaker can lead to ripple in the envelope. In the measurements shown below, the loudspeaker was placed approximately $y_S = 60\,\text{cm}$ away from the resting position horizontally and $l - x_S = 30$ cm above it. The same values were used in the simulation shown in Fig. 6. In our measurements, this choice lead to a modulation depth of about 5 dB shortly after the pendulum was started. Visually examining the recorded data as in Fig. 3 is a good way of ensuring that the modulation depth is in the range of a few dB and that clipping or ripple effects are withing some 0.5 dB.

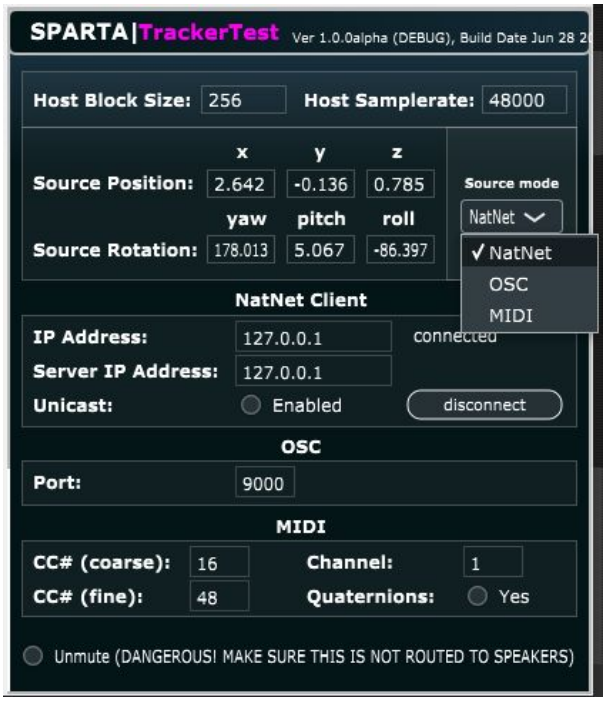


Figure 4: Screenshot of the newly developed TrackerTest plugin, which writes motion data received via OSC, MIDI, or NatNet directly to the audio output. TrackerTest is used to measure the motion-to-data-output latency.

2.2 Motion-to-data-output latency with TrackerTest

Before we discuss the two different methods used for measuring motion-to-sound and motion-to-data-output latency in more detail, a tool required for the latter is introduced. It is a dedicated VST plugin called TrackerTest; see Fig. 4. TrackerTest receives positional data as Cartesian coordinates and orientation data as either Euler angles or quaternions. The received data is directly written as audio samples to the six output channels (X, Y, Z, yaw, pitch, roll). The tracked data at time t, is denoted as a vector $\boldsymbol{d}(t) = \begin{bmatrix} d_x(t) & d_y(t) & d_z(t) & d_\varphi(t) & d_\theta(t) & d_\psi(t) \end{bmatrix}^\top$. If the tracker only provides orientation data, the position values are set to 0. When hosted in a DAW, this six-channel signal can be recorded directly on one of the audio tracks. The samples may have amplitudes exceeding 1 and must therefore be saved in a floating-point format without limiting to ensure the full dynamic range of the data is preserved. For the same reason, care must be taken to ensure the output is not sent to playback hardware.

With motion-to-data-output latency, we refer to the latency seen in the output of the TrackerTest plugin, i.e., the latency that would be measured without processing or audio digital-to-analog conversion. Since the data is buffered in the host software, the delay of one processing block size is always included in the measured motion-to-data-output latency. TrackerTest can receive positional data sent over NatNet (OptiTrack), MIDI (MrHeadTracker or Supperware), and Open Sound Control (OSC).

2.3 Measurement using Impulsive Movement

The first proposed measurement method employs an impulsive movement to determine the latency. The tracked object, e.g., the pendulum weight, is placed into its resting position. Then the weight is tapped with another solid object, such as a small hammer, while recording the impact on the microphone signal $s(t)$. Simultaneously, the TrackerTest output $\boldsymbol{d}(t)$ and the renderer output $r(t)$ are recorded for motion-to-data-output latency and motion-to-sound measurements, respectively. The loudspeaker see in Fig. 2 is not used for this method.

Motion-to-sound latency can be measured by rendering a stationary signal. At the resting position, the spatial rendering plugin applies a fixed spatial transfer function such that the rendering output is stationary. Once the spatial rendering plugin receives the positional change, the transfer function changes, so the output changes. In many cases, the changes in the spatial transfer functions are subtle when making small changes to position or orientation.

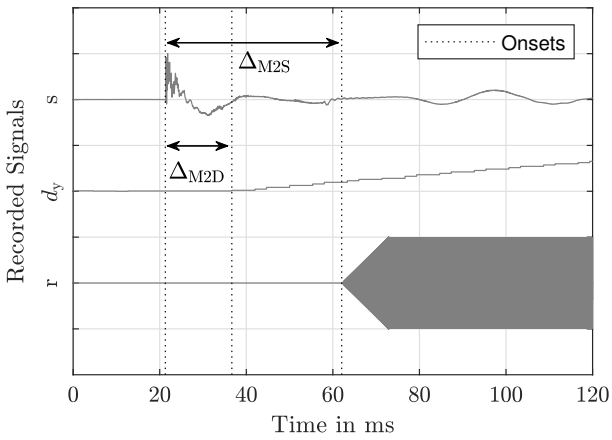


Figure 5: Signals recorded with the impulsive measurement method. The difference between the onsets corresponds to the motion-to-data-output and the motion-to-sound latency.

Therefore, we propose a unique dataset of transfer functions designed to magnify the change from the resting position. The transfer functions in close vicinity of the equilibrium point of the pendulum are set to zeros. All other positions have unit transfer functions, i.e., the renderer output is equal to its input. The 6DoFconv plugin dataset has zero transfer functions when all six coordinates are close to the resting position. A similar dataset can be manufactured for the 3DoF binaural renderer.

At the resting position, the spatial renderer outputs silence. When the pendulum weight is tapped, it moves away from the equilibrium and enters the zone that contains the unit transfer functions. As a consequence, the renderer produces output. The motion-to-sound latency is then measured through the time difference between the impact measured by the microphone and the time at which the renderer $r(t)$ became non-zero, see Fig. 5. We recommend using a signal alternating between ones and zeros as input to the renderer, i.e., a sinusoid at the Nyquist frequency, as it will make onset detection most straightforward.

The impulsive method has several disadvantages. First, the apparatus must be placed precisely at the resting position, which is a cumbersome process. Also, determining the tracker latency variance requires performing the measurement many times in a row. Furthermore, only spatial renderers that operate on discrete sets of transfer functions, like the 6DoFconv plugin or the SPARTA Binauraliser, can be used. In contrast, a spatial renderer that employs binaural Ambisonics decoding could not be measured, as it inherently interpolates, making it impossible to create a zone where the output is zero. But even with a suitable test data set, onset detection of the microphone peak and the plugin output is error-prone and can be inaccurate.

Note that the tracked object does not necessarily need to be suspended in the air for this method. One could also place it on a surface and tap it. However, we have found that putting it at the correct starting point for repeated measurements is difficult, and friction will impede the movement and influence the measurement.

2.4 Measurement using Periodic Movement

The second proposed method uses periodic movements, which simplifies the measurement procedure. Again, the microphone signal $s(t)$ is recorded together with the output of the TrackerTest plugin $\boldsymbol{d}(t)$ and the output of a spatial renderer $r(t)$. Additionally, the loudspeaker plays back noise during the measurement.

For the signal played back by the loudspeaker, it is not advisable to use a sinusoid since reflections from the room and measurement equipment can lead to undesired modulation. Instead, we use a broadband signal. The envelope of the signal picked up by the microphone $e_{\mathrm{s}}(t)$ is extracted as the root mean square (RMS) with a sliding window of size M, i.e.,

$$e_{\mathrm{s}}(t) = \sqrt{\sum_{\tau=-M/2}^{M/2} s(t+\tau)^2}. \tag{1}$$

To obtain a smooth envelope $e_{\mathrm{s}}(t)$ of the microphone signal, it is beneficial if the loudspeaker signal already has a smooth envelope. We suggest using a binary noise sequence, with pulses of unit magnitude and random signs at each sample.

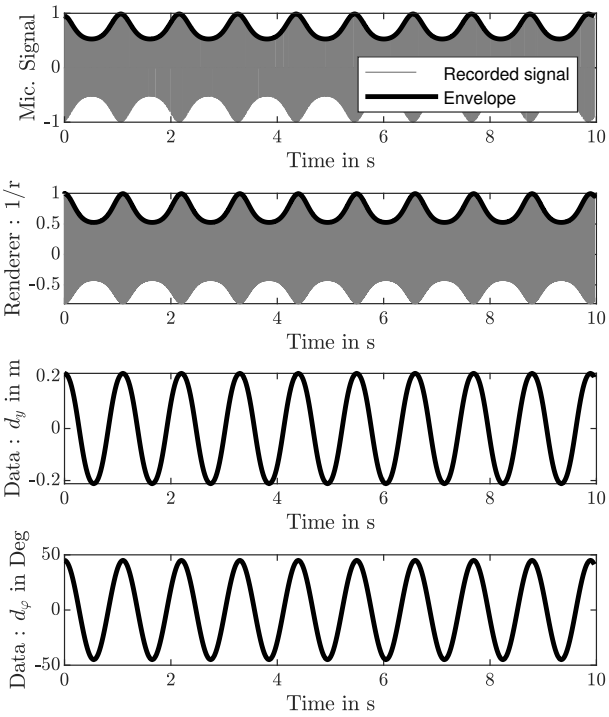
The motion-to-data-output latency is the position of the maximum of the cross-correlation between the microphone envelope $e_{\mathrm{s}}(t)$ and an appropriate positional stream $\boldsymbol{d}(t)$; see Fig. 6a for an example measurement. Fig. 6 shows two examples for positional data streams; the position coordinates like $d_y(t)$ or the rotation in the $x-y$ plane, $d_\varphi(t)$.

For signals with many periods of oscillation, cross-correlation analysis is unproblematic. If, however, the pendulum frequency is as low as 1 Hz and the measurement duration is 10 s (see Fig. 6a), the peak of the cross-correlation function can be broad, see Fig. 6b. Therefore, it is important to perform the cross-correlation carefully, with adequate normalization. To avoid edge effects, no zero padding is employed.

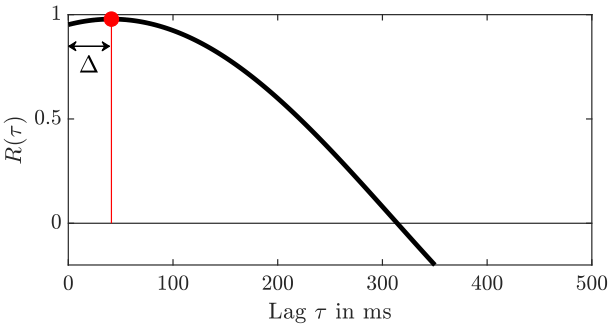
We compute the cross-correlation only for positive lags smaller than half of the pendulum period, i.e.,

$$R(\tau) = \frac{\sum_{t=1}^{N-\frac{T}{2}} d(t+\tau) e_{\mathrm{s}}(t)}{\sigma(\tau)}, \qquad 0 \le \tau < T/2 \tag{2}$$

where N is the total length of the signal, τ is the lag, T is the pendulum period, d is the selected positional



(a) From top to bottom: (1) Recorded microphone signal, (2) recorded output of a rendering plugin with a virtual source at the same position as the loudspeaker (3) y-coordinate $d_y(t)$ (4) rotation angle $d_\varphi(t)$ recorded with TrackerTest.



(b) Energy normalized cross-correlation function $R(\tau)$ for positive lags τ. Δ denotes the measured latency.

Figure 6: (a) Simulated signals with extracted envelopes. (b) Cross-correlation function of the envelopes.

data stream, and the normalization factor is

$$\sigma(\tau) = \left[\left(\sum_{t=1}^{N-\frac{T}{2}} d^2(t+\tau) \right) \left(\sum_{t=1}^{N-\frac{T}{2}} e_{\mathrm{s}}^2(t) \right) \right]^{\frac{1}{2}}. \quad (3)$$

The location of the maximum of the cross-correlation function

$$\Delta_{\mathrm{M2D}} = \arg\max_{\tau} R(\tau), \quad (4)$$

corresponds to the motion-to-data-output latency in samples, or $\Delta = \Delta_{\mathrm{M2D}}/f_{\mathrm{s}}$ in seconds, where f_{s} is the sampling frequency.

Fixing the window size in Eqs. (2) and (3) avoids the need for zero padding and possible bias due to edge effects. In addition, the normalization is important to avoid bias that could occur as the number of periods used for the analysis is typically low (e.g., 10 in the examples shown below), so that the peak of the cross-correlation function is not very prominent, see Fig. 6b.

For motion-to-sound latency, the position data is replaced with the envelope of the renderer's output. The input to the renderer, is selected to be a high-frequency sinusoid. The advantage of using a sinusoidal signal here is that envelope extraction can be performed very precisely by taking the absolute of the analytical signal

$$e_{\mathrm{r}}(t) = \left| r + \imath \left[\frac{1}{t\pi} * r(t) \right] \right|, \quad (5)$$

where $\imath$ is the imaginary unit, $|.|$ denotes the absolute value of a complex number and $*$ denotes convolution. $\frac{1}{t\pi} * r(t)$ is called the Hilbert transform of $r(t)$. The motion-to-sound latency is determined similarly to motion-to-data-output using Eqs. (2)-(4) with $e_{\mathrm{r}}(t)$ instead of one of the datastreams in $\boldsymbol{d}(t)$.

Compared to the impulsive method, the requirements on the dataset loaded by the spatial renderer are less strict. The cross-correlation method only requires that the level change with the pendulum movement is monotonic in at least one dimension. Thus, spatial renderers that do not allow the loading of a specific dataset or only produce interpolated output can be measured. The only challenge is to choose tracker placement and positional mappings that lead to sufficient amplitude modulation of the signal output. For example, when assessing the output of a 3DoF binaural Ambisonics decoding plugin, one may place a sound source on the side of the virtual listener and map the orientation of the tracker in the z-axis (rotation in the x-y plane, see Fig. 2) to the azimuth of the virtual listener. For a 6DoF plugin, a sound source can be placed at the position of the actual loudspeaker. The specific choice of analyzed position parameter and how the renderer's output is modulated may influence the result, which is described in the next section.

3 RESULTS

This section evaluates the proposed methods using simulation and actual measurement data.

3.1 Simulations with Theoretical Envelopes

Before using actual data, we simulate the setup and examine the resulting delay estimates. The advantage of such a simulation is that the ground truth de-

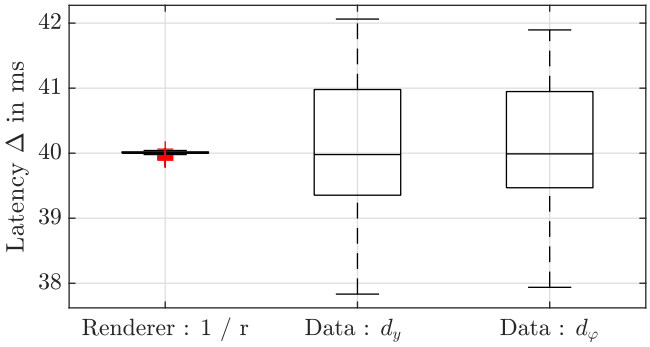


Figure 7: Distribution of the estimated latency over 100 runs of a simulated pendulum oscillating for 10 s using a spatial renderer output or tracker data output.

lay data is readily available. For these simulations, we assume a pendulum of length $l = 0.3$ m, which is mounted in the coordinate origin, and a loudspeaker at position $(x_\mathrm{S}, y_\mathrm{S}) = (0, 0.6)$ m (see Fig. 2 for coordinate definitions), and a motion-to-data-output latency of 40 ms.

Once started, an ideal, friction-less pendulum moves periodically, such that the angle φ is

$$\varphi(t) = \hat{\varphi}\cos(2\pi f t + \varphi_0), \tag{6}$$

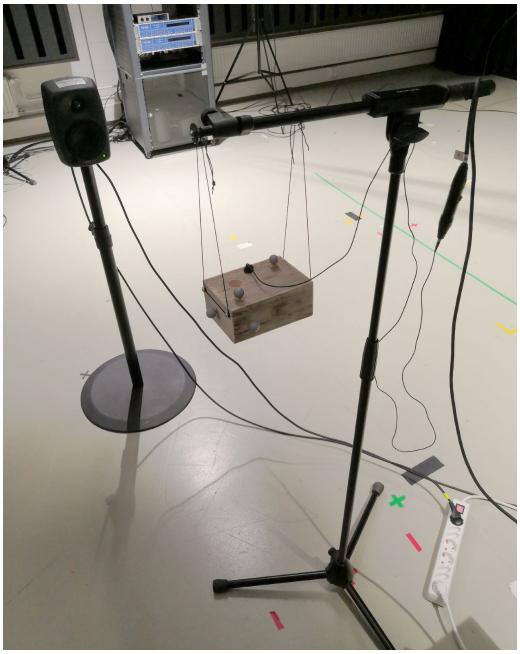
where f denotes the pendulum frequency, which is related to the pendulum length as $f = 1/T \approx \sqrt{g/l}$, where g is the local acceleration of gravity. In this arrangement, the distance between the pendulum and the loudspeaker is given by

$$R(t) = \sqrt{(l\cos\phi(t) - x_\mathrm{S})^2 + (l\sin\phi(t) - y_\mathrm{S})^2}. \tag{7}$$

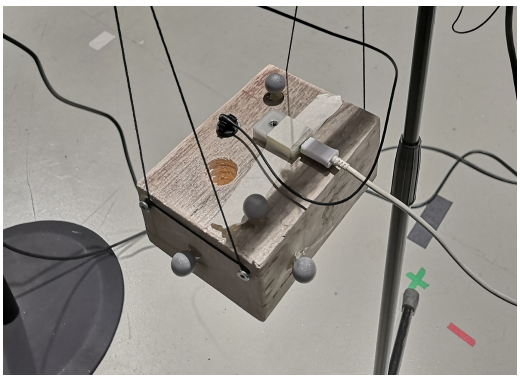
With these basic equations, we can simulate most of the data that may be used for the statistical analysis. If it can be assumed that the microphone is in the far field of the loudspeaker, the envelope of the recorded microphone signal is proportional to $1/R$. Hence, the microphone signal envelope $e_\mathrm{s}(t)$ is not sinusoidal, as shown in Fig. 6a.

Similarly, for motion-to-data-output latency measurement, the data streams are periodic, but not necessarily sinusoidal either. Fig. 6a shows two more examples of a position tracker using position d_y, and the rotation angle d_φ in the $x - y$ plane. If motion-to-sound latency is measured using a 6DoF plugin, a sound source can be placed at the position of the actual loudspeaker. Then, the envelope of the rendered output $e_\mathrm{r}(t)$ is ideally the same as the microphone signal envelope $e_\mathrm{s}(t)$, only shifted by the latency.

We show that correlating the microphone envelope $e_\mathrm{s}(t)$ with the other signals produces unbiased latency estimates. Therefore, we performed a simulation experiment, running 100 simulated pendulums for 10 s and correlating the obtained microphone envelope with the same envelope obtained from a hypothetical 6DoF renderer, as well as for d_y and d_φ, as they would be obtained from the tracker. The starting phase φ_0 was randomized. The result in Fig. 7 shows that estimation is exact when using the same $1/R$ relationship as for the loudspeaker. For d_y and d_φ, there is more variation (the standard deviation is about 1 ms in both cases). The important result is that the estimate is unbiased, i.e., if one performed several measurements or extracted different segments of one measurement, it would converge to the actual latency.



(a) The complete setup with Genelec loudspeaker.



(b) Close-up of the swinging weight with the markers used for OptiTrack and an IMU tracker attached

Figure 8: Photographs of the setup.

3.2 Measurement Setup

The specific setup for the measurements presented here was realized as follows. A wooden block was used as a weight, with a DPA IMK 4060 microphone mounted on it. The weight is suspended with strings from the boom of a microphone stand. The microphone is held in place by a magnetic clip attached

to a screw. Also, tracking markers or the tracker under test are affixed to the box (see Fig. 8b for both). To constrain the movement of the weight such that it mostly swings in a single plane, we use two strings on opposite sides of the weight, each attached to two screws. We used a Genelec 8331AP loudspeaker and an RME Fireface UCX audio interface. Measurements were performed in the variable acoustics room "Arni" at the Aalto Acoustics Laboratory, which was set to a dry setting, with mid-frequency reverberation times RT < 0.5 s.

3.3 6DoF Tracker

First, we used the proposed methods to test a current measurement-based 6DoF virtual acoustic rendering system using the 6DoFconv plugin [13, 17] hosted in Cockos Reaper. That system uses an OptiTrack tracking system with six Prime 13 W cameras (240 Hz update rate). The 6DoFconv plugin receives OptiTrack data directly over the NatNet protocol.

Five measurements were conducted with the impulsive method. For these measurements, a dataset was loaded, which contained a zero transfer function at $y = 0$, and the first unit transfer function at $y = \pm 5$ mm. For the periodic method, five measurements were conducted as well, where the pendulum was left swinging for one minute each. Then, each recording was split into segments of 10 seconds, and cross-correlation based latency analysis was applied to each segment separately.

Figure 9 shows the measured motion-to-sound latency in dark gray and the motion-to-data-output latency measured with the TrackerTest Plugin in a lighter shade. The median movement-to-data-output latency of this system is 12.41 ms, and the motion-to-sound latency is 36.82 ms. Thus, audio rendering and output comprise about 2/3 of the total motion-to-sound latency.

Between different time segments within one measurement, the results vary within a range of 1.83 ms in the best case (measurement 5) and 3.13 ms in the worst case (measurement 4).

The median values over the segments (white dots in Fig. 9) vary by approximately 1 ms between the measurements. Therefore, conducting one measurement and taking the median value between the results of different segments of that measurement can be assumed to provide a robust estimate. The median is preferred over the mean, as it generally is a more robust estimator of central tendency when the sample size is small.

For comparison, five measurements with the impulsive method were made, shown on the right in Fig. 9. The variation observed in the movement-to-data-output measurement is comparable to that seen in the periodic method. However, the variation of the measured movement-to-sound latency is higher (the difference between the largest and the smallest value is 11.2 ms). This may be due to the fact that detecting the onset of the rendered signal is more prone to errors than detecting the onset in the data stream. Also, the median result of the impulsive measurement is about 2 ms higher (39.00 ms vs. 36.82 ms for motion-to-sound latency). Since the presented method should be theoretically unbiased, it is likely that this deviation occurs because in the impulsive method, the swing needs a certain time to leave the silent zone.

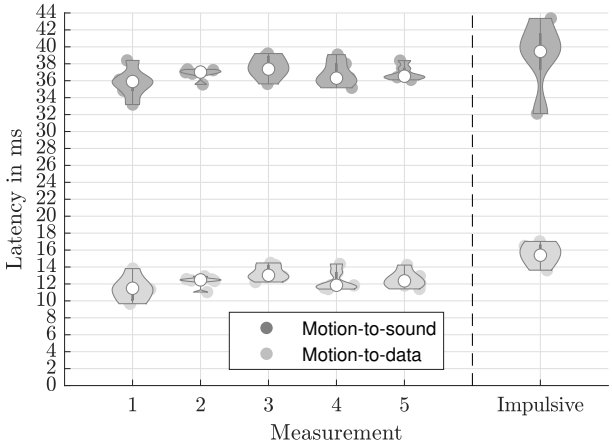


Figure 9: Measurement using the proposed periodic method on the left side, where five different trials are shown. The points show variation across 10 s windows within these. The rightmost points show the results of five trials using the impulsive method. Median values are shown as white dots.

Considering the result itself, it appears that the rendering chain employing OptiTrack and the 6Dofconv plugin hosted in Reaper can produce renderings with a motion-to-sound latency that is likely to be undetectable in most situations, assuming a limit of about 60 ms from [14].

3.4 IMU Trackers

We also show an example of motion-to-data-output measurements using the TrackerTest plugin for an IMU-based tracker, the MrHeadTracker (MrHT) [12]. Again, five different measurements were made with each device and each method. Fig. 10 shows the results. Whereas the latency measured with the periodic method is even lower for MrHT (10.80 ms) than for the OptiTrack (12.42 ms), the result obtained with the impulsive method is much higher for MrHT (29.98 ms) than for the OptiTrack (15.39 ms). A possible explanation for this discrepancy is that the sensor fusion algorithm implemented in the IMU tracker is capable of predicting the periodic movement and thereby reducing latency. In contrast, impulsive movement can not be predicted.

In fact, measurements were performed with two more trackers, the Supperware HeadTracker 1, and a new prototype based on the MrHT. The Head

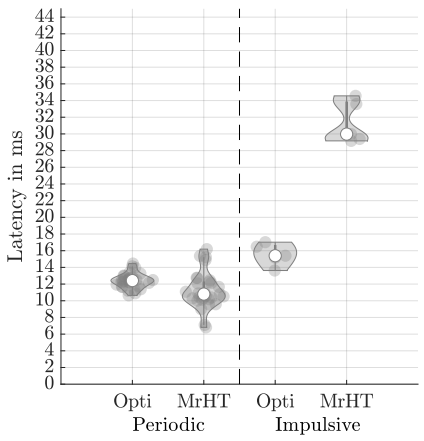


Figure 10: Measurement results obtained for Optitrack (Opti) and MrHeadTracker (MrHT) using the periodic and the impulsive method. For the periodic method, the MrHT latency is slightly lower, yet for the impulsive measurement, it is much higher.

Tracker 1 had an impulsive latency of 37.38 ms and the prototype had 36.00 ms. However, both these trackers showed negative latencies in the periodic measurements in some segments. A discussion of negative latencies and their possible origins in the specific sensor fusion algorithms that the IMU trackers employ is beyond the scope of this study.

It should be noted that neither the impulsive nor the periodic methods employ realistic head movement trajectories. In the future, such movements should be used to assess the latency of IMU trackers. They could be performed by a robot or by a human volunteer. Now that the tools are available, which allow for obtaining reference results for the OptiTrack system, these movements could be tracked with it and the trajectories could be correlated with IMU tracking results. Such experiments will be the subject of future studies. For now, the impulsive method provides a benchmark for these trackers.

4 CONCLUSION

This paper proposes two different latency measurement methods and open-source tools that implement them. For the periodic method, we have shown that, energy-normalized cross-correlation leads to unbiased estimates. Evaluation of the periodic methods has shown variations of the result of approximately 1 ms on a high-quality OptiTrack 6DoF tracking system when using the median between 10 s long measurement segments. Theoretical analysis has shown that the method can be used on various types of tracking data.

Also, we have shown measurements on an IMU head tracker. It showed a higher latency than the OptiTrack system when measured using the impulsive method, and a lower latency when using the periodic method. Other IMUs even produced negative latencies for the periodic method. Therefore, the impulsive measurement method, although more cumbersome and with more variability between measurements, is currently the more suitable approach for benchmarking IMU-based trackers. In the future, assessment using a set of typical head movements executed by a human or robot could be an alternative.

The motion-to-sound latency measured for the examplary 6DoF setup of about 37 ms is likely to be sufficiently low for most applications. Interestingly, it is in a similar range as the best-case end-to-end latencies reported for another rendering system 20 years ago [20]. Yet the modern plugin-based system is more flexible and its latency could most likely still be optimized. We suggest performing latency measurements when conducting listening tests using dynamic rendering, to assure that latency does not influence the outcome in undesired ways. Lastly, it should be noted that latency is not the only important parameter for assessing tracking. For example, drift can be a significant issue, too, especially with IMU-based sensors.

ACKNOWLEDGEMENTS

This research was supported by European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 812719.

References

[1] M. Vorländer, *Auralization: Fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality*, 1st ed., RWTHedition (Springer, Berlin, 2008), https://doi.org/10.1007/978-3-540-48830-9.

[2] B. Xie, *Head-related transfer function and virtual auditory display*, second edition ed. (J. Ross Publishing, Plantation, FL, 2013).

[3] L. McCormack and A. Politis, "SPARTA & COMPASS: Real-time implementations of linear and parametric spatial audio reproduction and processing methods," presented at the *AES International Conference on Immersive and Interactive Audio* (2019 Mar.).

[4] D. Poirier-Quinot and B. F. G. Katz, "The Anaglyph binaural audio engine," presented at the *144th AES Convention* (2018 May).

[5] M. Cuevas-Rodríguez, L. Picinali, D. González-Toledo, C. Garre, E. de la Rubia-Cuestas, L. Molina-Tanco, *et al.*, "3D Tune-In Toolkit:

An open-source library for real-time binaural spatialisation," *PLoS ONE*, vol. 14, no. 3, p. e0211899 (2019 Mar.), https://doi.org/10.1371/journal.pone.0211899.

[6] K. G. Rosero, D. Abreu, F. L. Grijalva, and B. S. Masiero, "How do spatial audio plug-ins work and what functionalities do they offer: a comparative perspective," presented at the *XII Congresso Iberoamericano de Acústica* (2022 Aug.).

[7] H. Lee, M. Frank, and F. Zotter, "Spatial and Timbral Fidelities of Binaural Ambisonics Decoders for Main Microphone Array Recordings," presented at the *AES International Conference on Immersive and Interactive Audio* (2019 Mar.).

[8] S. Wirler, N. Meyer-Kahlen, and S. J. Schlecht, "Towards transfer-plausibility for evaluating mixed reality audio in complex scenes," presented at the *AES International Conference on Audio for Virtual and Augmented Reality* (2020 Aug.).

[9] D. Poirier-Quinot and B. Katz, "Assessing the Impact of Head-Related Transfer Function Individualization on Task Performance: Case of a Virtual Reality Shooter Game," *J. Audio Eng. Soc.*, vol. 68, no. 4, pp. 248–260 (2020 May), https://doi.org/10.17743/jaes.2020.0004.

[10] K. Groß-Vogt, M. Weger, M. Frank, and R. Höldrich, "Peripheral Sonification by Means of Virtual Room Acoustics," *Computer Music Journal*, vol. 44, no. 1, pp. 71–88 (2020 Mar.), https://doi.org/10.1162/comj_a_00553.

[11] M. Geronazzo, J. Y. Tissieres, and S. Serafin, "A Minimal Personalization of Dynamic Binaural Synthesis with Mixed Structural Modeling and Scattering Delay Networks," presented at the *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 411–415 (2020 May), https://doi.org/10.1109/ICASSP40776.2020.9053873.

[12] M. Romanov, P. Berghold, D. Rudrich, M. Zaunschirm, and M. Frank, "Implementation and Evaluation of a Low-cost Head-tracker for Binaural Synthesis," presented at the *142th AES Convention* (2017 May).

[13] T. McKenzie, N. Meyer-Kahlen, R. Daugintis, L. Mccormack, S. J. Schlecht, and V. Pulkki, "Perceptually informed interpolation and rendering of spatial room impulse responses for room transitions," presented at the *24th International Congress on Acoustics* (2022 Oct.).

[14] D. S. Brungart, A. J. Kordik, and B. D. Simpson, "Effects of Headtracker Latency in Virtual Audio Displays," *J. Audio Eng. Soc.*, vol. 54, no. 1 (2006).

[15] A. Lindau, "The Perception of System Latency in Dynamic Binaural Synthesis," presented at the *DAGA - Jahrestagung für Akustik*, pp. 1063–1067 (2009 Jan.).

[16] P. Stitt, E. Hendrickx, J.-C. Messonnier, and B. F. Katz, "The Influence of Head Tracking Latency on Binaural Rendering in Simple and Complex Sound Scenes," presented at the *140th AES Convention* (2016).

[17] N. Meyer-Kahlen, S. V. Amengual Garí, T. McKenzie, S. J. Schlecht, and T. Lokki, "Transfer-plausibility of binaural rendering with different real-world references," presented at the *DAGA - Jahrestagung für Akustik* (2022 Mar.).

[18] J. Estrella and J. Plogsties, "Motion-to-Sound Latency Measurement Procedure for VR Sound Reproduction," presented at the *142nd Audio Engineering Society Convention* (2017 May).

[19] P. Fran, "Simple dynamic measurement system for testing IMU sensor precision in spatial audio," presented at the *Baltic Nordic-Acoustic Meeting*, pp. 297–305 (2022 May).

[20] J. D. Miller, M. R. Anderson, E. M. Wenzel, and B. U. McClain, "Latency measurement of a real time virtual acoustic rendering system," presented at the *International Conference on Auditory Display*, pp. 111–114 (2003 Jul.).