HeadBower: Towards Expressive Violin Control for Musicians with Quadriplegia

Nicola Davanzo[1][0000−0003−3073−5326], Pier Giorgio Gariboldi[1][0009−0007−4595−6056], and Federico Avanzini[1][0000−0002−1257−5878]

Laboratory of Music Informatics (LIM), Department of Computer Science, University of Milan, Via Celoria 18, 20133 Milan, Italy
{nicola.davanzo,federico.avanzini}@unimi.it,
piergiorgio.gariboldi@studenti.unimi.it
http://www.lim.di.unimi.it

Abstract. Accessible Digital Musical Instruments (ADMIs) offer significant potential for musical expression to individuals with quadriplegia, yet achieving fine-grained, expressive real-time control remains a considerable challenge. This paper introduces HeadBower, an experimental ADMI designed to investigate expressive violin control for this user group. HeadBower translates head movements into nuanced violin bowing gestures—horizontal rotation velocity for intensity/sweep, directional changes for note events, and head roll for modulations like vibrato/pitch bend—while also exploring supplementary jaw and mouth sensor inputs for finer articulatory control over parameters such as bow pressure. For note selection, HeadBower implements gaze tracking with two distinct visual paradigms—including a novel circular dynamic layout—both aimed at fluid playability while mitigating the Midas Touch problem. To enhance performer connection, HeadBower experimentally integrates smartphone-delivered haptic feedback. This research details these mapping strategies and interface designs, highlighting their potential to provide multifaceted, expressive violin performance capabilities for individuals with severe motor impairments. All hardware and software components are released under open-source licenses to promote accessibility and further development.

Keywords: Accessible Digital Musical Instruments · Quadriplegia · Head Tracking · Gaze Tracking · Haptic Feedback · Violin · Expressive MIDI Mapping · Open-Source

1 Introduction

Musical engagement, especially instrument play, is a profoundly enriching activity conferring cognitive, emotional, and social benefits [5, 45], enhancing skills like verbal intelligence and executive function, and boosting well-being across age groups [24]. The Universal Declaration of Human Rights affirms the right to participate in cultural life and enjoy the arts [46]. However, for individuals with severe motor impairments such as quadriplegia, conventional music-making presents significant barriers [8]. Quadriplegia, or tetraplegia, involves paralysis of all four limbs and the torso, often resulting from cervical spinal cord injuries. This condition can stem from traumatic spinal cord injuries (SCI)—with the World Health Organization estimating approximately 15.4 million people living with SCI globally in 2021[1]—or neurological disorders like amyotrophic lateral sclerosis (ALS), multiple sclerosis (MS), and severe cerebral palsy (CP) [43]. The consequent loss of upper extremity movement profoundly restricts the motor control needed for most traditional instruments, which typically demand intricate hand/finger dexterity and often foot or breath coordination. This exclusion limits personal expression and access to music's developmental and therapeutic benefits [26]. To bridge this gap, *Accessible Digital Musical Instruments* (ADMIs) have become a vital research area [18]. ADMIs use digital technology for novel interaction modes, enabling individuals with diverse abilities to perform music [39]. For users with quadriplegia, ADMIs focus on residual motor capabilities from the neck upwards—head movements, facial gestures, eye movements, and breath control [8]—mapping these to musical parameters via specialized interfaces.

Developing ADMIs for this population is crucial for inclusion, artistic expression, therapeutic goals, and quality of life. Digital interfaces can be tailored to individual needs, offering previously unattainable pathways to musical engagement and education [11]. Yet, expressive music interaction solutions tailored specifically for the limitations of quadriplegia remain relatively scarce [18, 8], underscoring a need for innovation. Such instrument design must address ergonomic challenges, potential fatigue, and the necessity for intuitive, responsive interaction for nuanced musical expression.

While past ADMIs for users with quadriplegia have enabled real-time performance, achieving broad and deep expressive control remains a key challenge. Many existing instruments offer control mainly over fundamental aspects like note onset, pitch, and volume. Consequently, conveying finer musical nuances—detailed timbral variations, vibrato, glissando, and the rich spectrum of articulations often managed by numerous MIDI parameters (e.g., Continuous Controllers)—is a significant area for advancement.

In this context, we present innovative experiments with HeadBower, an ADMI addressing these expressive limitations in violin performance for users with quadriplegia. HeadBower distinctively focuses on mapping head movements to violin bowing dynamics: horizontal rotation dictates bow sweep and intensity, directional changes trigger note articulations, and head roll controls modulations. Note selection employs gaze tracking with two alternative visual layouts designed to mitigate interaction challenges. The system also explores supplementary sensor inputs (jaw pressure, mouth opening) for finer articulatory control and integrates haptic feedback via smartphone vibration to enhance performer engagement. HeadBower relies on NITH framework (which is a general-purpose framework for developing hands-free interaction solutions, described in Sec. 3.2)

[1] From World Health Organization's Spinal Cord Injury factsheet (accessed on May 2025): https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury

for sensor integration and interaction logic. All HeadBower hardware, software, and enabling NITH framework components are open-source[2] (e.g., GNU GPLv3 for software, Creative Commons for hardware), promoting accessibility and development.

This paper details experiments with HeadBower, focusing on its novel mapping strategies and interface paradigms. The primary contributions are:

1. The proposal and implementation of a mapping strategy translating head movements into violin bowing dynamics: horizontal head rotation (yaw) controls bow sweep and its velocity dictates intensity, while directional changes trigger MIDI Note On/Off events. Head roll (or pitch) is mapped to expressive modulations like vibrato or MIDI pitch bend, offering a novel hands-free expressive control paradigm for users with quadriplegia.
2. The design and integration of a smartphone-delivered haptic feedback system using the NITHphoneWrapper application, providing tactile sensations correlated with bowing actions to enhance user engagement and control with a touch-less ADMI.

To accomplish this, minor enhancements to the NITH framework—designed for developing accessible hands-free interaction solutions, as described in Sec. 3.2—were implemented and are discussed in the text, and result as a further contribution to the topic.

The remainder of this paper is structured as follows: Sec. 2 provides a background on ADMIs, relevant interaction channels, and related work. Sec. 3 details the software and hardware components, including sensors, libraries and sound producing means. Sec. 4 describes HeadBower's implementation, including its visual layouts, interaction strategies, and haptic feedback system. Finally, Sec. 5 discusses the contributions, limitations of the current experiments, and outlines future work, particularly regarding formal user evaluation.

2 Background

The field of Accessible Digital Musical Instruments (ADMIs) has been systematically surveyed in several key works, providing a robust foundation for ongoing research. Frid's systematic review analyzes 83 ADMIs, identifying ten distinct control interface types, highlighting trends like the prevalence of tangible controllers and infrequent vibrotactile feedback, and underscoring adaptable design and user participation as crucial for successful ADMI development [18]. A conceptual framework for "Hands-free DMIs" (HeaDMIs) was presented by Davanzo and Avanzini, listing and analyzing physical interaction channels available from the neck upwards, surveying existing instruments for this demographic, and discussing open research challenges [8]. Vamvakousis's doctoral thesis contributes

[2] The HeadBower open-source project, including source code, further documentation, and demonstration videos, is available at: https://github.com/LIMUNIMI/HeadBower

significantly by detailing the design and implementation of DMIs tailored for various physical disabilities, including those with limited upper limb function, with a notable focus on gaze control, as exemplified by the development and evaluation of the EyeHarp [47]. Beyond this specialized focus, the broader field of interaction with digital musical instruments (DMIs) has been extensively documented. Foundational texts like "New Digital Musical Instruments: Control and Interaction Beyond the Keyboard" by Miranda and Wanderley establish key concepts for DMI paradigms, control mechanisms, and gesture-sound interplay [39]. McGlynn's work on Interaction Design for DMIs further explores crucial aspects like mapping strategies, performer-instrument relationships, and the design principles for effective musical interfaces [35].

2.1 Interaction Channels

The following paragraphs will explore various interaction channels available for ADMIs, with particular emphasis on those that are relevant for HeadBower.

Gaze Interaction. Gaze-based interaction is prominent in ADMIs, especially for users with extensive motor limitations [8]. Eye movement characteristics—rapid *saccades*, stable *fixations*, and *smooth pursuit* [29, 10]—present unique design challenges and opportunities. A primary challenge is the "Midas Touch" problem: unintended actions triggered by looking at interactive elements [29, 10]. This is critical in musical performance, demanding high precision and minimal latency; various musical mitigation strategies exist [10]. Common approaches [10] include *dwell time* (risking detrimental Delayed Auditory Feedback, DAF), *filtering* algorithms (potential latency issues), and *hybrid interaction*. Hybrid interaction decouples gaze selection from activation, using a secondary channel like breath [1, 13, 14], head movements [12], or blinks/winks [4]. Strategic *keys displacement* and layout design are also crucial. For instance, *Pie-shaped layouts* (e.g., EyeHarp [48]) reduce accidental activations via a central neutral area [10]; EyeHarp also offers expressive control via gaze position within selected slices [47]. Other methods include *spacing between keys* or careful gaze-sensitive area design (e.g., Netytar [13]). Visual cues are vital: *color* can aid note localization (peripheral vision), interfaces might *highlight selections* or use "discrete cursors" to avoid distraction, and "visual hooks" (e.g., dots in EyeHarp keys [48]) enhance fixation precision [10]. Moreover, *auto-scrolling* interfaces (e.g., Netytar [13], Netychords [12]), leveraging smooth pursuit, improve accuracy and enable theoretically infinite playing surfaces. Smooth pursuit also facilitates unique interaction paradigms, like in Kiroll [14].

Head Tracking. Head movement, common in ADMIs for users with quadriplegia and cervical mobility, offers multiple DoF for cursor control, direct parameter mapping (e.g., pitch in Hi Note [34], Magic Flute [41]), and chord strumming (Netychords [12]) [8]. HeadBower uses head rotation extensively: yaw for bow sweep/intensity, directional changes for notes, and roll/pitch for modulation. A previous ADMI named Resin employed a similar but simpler yaw-based bowing gesture [9]. Sensors include IMUs and camera-based systems. Design must consider potential fatigue and distinguishing intentional from natural movements [8].

Mouth Tracking. Mouth and lip movements offer valuable channels. Mouth aperture, for instance, can map to continuous musical controls (e.g., HeadBower's bow pressure; EyeConductor; Mouthesizer [27, 8]). Common sensors are camera-based systems with facial landmark tracking. Design must avoid vocalization interference and manage fatigue [8].

Teeth Pressure. Teeth pressure or jaw clenching, a less explored channel, offers discrete or continuous control. Typical sensors (FSRs, piezoelectric) face challenges like comfort, hygiene, calibration, and potentially limited expressive range [8]. In HeadBower, jaw pressure via NITHteethPressureSensor (Sec. 3.3) can modulate bow pressure.

2.2 ADMI Design Choices and Challenges

The subsequent paragraphs will examine past literature contributions in ADMIs design choices and challenges, again emphasizing aspects pertinent to HeadBower.

Haptic Feedback. Haptic feedback (tactile/kinaesthetic information) is crucial in traditional instruments for state/action information but often reduced/absent in DMIs due to decoupled control/sound mechanisms, causing a "poor feel". Its integration in DMI design is uncommon: a 2008 survey found $< 9\%$ of NIME DMIs had physical feedback, $< 6\%$ vibrotactile [33]. The ADMI situation is similar: Frid's review of 83 ADMIs showed only 15.6% had vibrotactile feedback [18]. This scarcity is acute for ADMIs for users with quadriplegia (often no direct physical contact). HeadBower addresses this with smartphone-delivered vibrations [8].

Expressive Control. Achieving nuanced expressive control in ADMIs remains challenging. Many, especially for users with severe motor impairments, offer limited expressive parameters (typically note onset, pitch, intensity), with few DoF and mainly one-to-one mappings, restricting the expressive palette [8]. Frid noted that past ADMIs often briefly discuss/omit sound synthesis/control, suggesting gaps in detailed timbral control [18, 19]. Lack of detailed, individualized assessment of disabled musicians' capabilities can also limit nuanced expression [36]. The NIME community has long recognized rich "expressive control" needs more than simple mappings, requiring sophisticated input-sound relationships [15]. The landscape evolves, distinguishing ADMIs for therapy/initial access from "performance-focused instruments" for virtuosity [6]. Several projects show commitment to richer expression: I-Ork uses one-to-many mappings and sophisticated sound environments [32]; WithFeelVR maps 3D texture interactions to granular synthesis for nuanced timbre/pitch [44]; MAMI Tech Toolkit includes VSTs and varied mapping [51]. "Smarter" machine learning instruments also show promise [25]. Though limited expressive mapping is still a challenge, the field seeks solutions via advanced sensors, sophisticated mappings, flexible synthesis engines, and user-centered design [8, 15].

Importance of Customization. Deep customization and personalization are paramount in ADMI design due to heterogeneous disabilities (e.g., varying quadriplegia impairments or CP functional abilities [3, 30]) and unique user-technology interactions. Severe motor impairments show wide-ranging, potentially time-varying residual abilities (e.g., progressive ALS [40]). Thus, one-size-fits-all approaches are often inadequate, risking technology abandonment [23, 17]. Individualization is crucial for user comfort, system performance, and UX in assistive technologies [31, 38]. Developing truly accessible applications requires designing for users' remaining motor abilities and carefully adjusting parameters for effective physical channels [8]. A Netychords case study for a child with CP vividly illustrates this adaptive design's importance, stressing the "necessity of a highly adaptive and customizable approach in designing ADMIs [11]".

Drawing from this landscape, as will be detailed in Sec. 4, HeadBower directly addresses several identified gaps, notably in expressive control and haptic feedback, while emphasizing customization to meet diverse user needs. Its head-to-bow mapping offers a more complex expressive model than the simple parameter controls often seen. The dual-layout gaze system is a direct response to the Midas Touch problem, and the integration of smartphone haptics tackles the noted scarcity of vibrotactile feedback in ADMIs

3 Software and Hardware components

In this section, we provide an overview of the software and hardware components that enable HeadBower's functionality.

3.1 SWAM solo instruments

HeadBower is a mute MIDI interface that requires a MIDI synthesizer to send messages for sound production. While many VST plugins can potentially work with standard MIDI messages from HeadBower, it is specifically designed for the SWAM solo string instruments developed by AudioModeling[3]. These Virtual Studio Technology (VST) plugins utilize physical modeling technology instead of pre-recorded samples. SWAM (Synchronous Wave Acoustic Modeling) technology allows for real-time shaping of various articulations and phrasing, aiming for playability and realism comparable to acoustic instruments. The SWAM Solo Strings collection includes the Violin, Viola, Cello, and Double Bass, which are meant to be played in real-time using expressive MIDI controllers.

A crucial aspect of SWAM instruments is their extensive MIDI controllability, especially via MIDI Control Change messages (CC).

[3] AudioModeling SWAM Solo Strings on AudioModeling website: https://audiomodeling.com/strings/solo-strings/

Beyond basic note control, several MIDI-controlled parameters are key for HeadBower's expressive mapping experiments[4]:

- **Expression (Dynamics)**: Typically CC11 or CC2, this primary control is linked to bow speed and overall dynamics for bowed strings.
- **Bow Pressure**: Simulates the "weight" of the bow on the string, essential for expressivity.
- **Vibrato Depth** and **Vibrato Rate (Modulation)**: Facilitate precise control over the amount and speed of vibrato.
- **Pitch Bend Up/Down**: Allows continuous pitch changes in semitones.

Other noteworthy SWAM parameters enhancing realism and flexibility include **Bow Lift** (bow-on-string status during release), **Bow/Pizz Position** (timbre changes, e.g., 'sul ponticello' to 'sul tasto'), **Legato/Portamento** transitions, **Play Modes** (e.g., Bow, Pizzicato, Col Legno via Key Switches), **Bow Start** direction, natural **Harmonics**, **Bow Noise**, and **String Resonance**.

In the realm of Digital Musical Instruments (DMIs), a mapping strategy describes the specific method by which parameters from a performer's physical interactions are linked to and control musical parameters of a sound production unit[8]. SWAM instruments provide a customizable MIDI mapping interface, allowing users to assign these and various other parameters to different MIDI messages (CC, Aftertouch, NRPN) and tailor the response curves.

3.2 The NITH Framework

The NITH[5] framework is a comprehensive open-source software and hardware ecosystem specifically designed to facilitate the rapid development and customization of accessible human-computer interaction systems for users who have limited or no use of their hands and feet. It aims to address the diverse needs of these users by supporting a wide array of interaction channels available from the neck upwards, such as head movements, facial gestures, gaze tracking, and breath control. All of them are defined through a comprehensive list of parameters that can be detected sensors. NITH framework components can be downloaded and used freely. Extensive documentation, examples, and GitHub repositories links are collected in NITH's website[6]

At its core, NITH emphasizes modularity, simplicity (adhering to the KISS[7] principle), affordability, and openness to community-driven improvement. The ecosystem comprises several key interdependent components. The relevant ones for HeadBower implementation include:

- `NITHlibrary`: The main software library, implemented in C# with .NET 8, providing a consistent interface for parsing data from NITH-compatible

[4] Parameters may change in more recent implementations of the SWAM instruments.
[5] NITH acronym is the result of a joke/play on words, and stands for "Non-Invasive Trans-Humanism"
[6] NITH framework website: https://purl.org/nith-framework
[7] Keep It Simple, Stupid.

sensors and wrappers. It handles data manipulation, filtering, calibration, and supports a behavior-based design for defining interaction logic.
- `NITHsensors`: A collection of easily constructible, open-source hardware sensor designs (DIY-AT devices) like the `NITHheadTracker` or `NITHbreathSensor`, designed for detecting various hands-free interaction parameters[8].
- `NITHwrappers`: Software components, such as the `NITHwebcamWrapper`, that interface with existing commercial peripherals (e.g., webcams) or third-party software, converting their output into the NITH framework's standard communication protocol.
- `NITHdmis`: A specialized extension of `NITHlibrary` providing additional tools tailored for the development of ADMIs.
- `NITHemulation`: A module for emulating mouse and keyboard inputs through sensors and wrappers.

A key aspect of NITH is its unified handling of data from these diverse sources. All components communicate using a standardized, plain-text communication protocol (`NITHprotocol`), which allows, for example, an application to receive head tracking data (e.g., `head_pos_yaw` from either a dedicated IMU-based `NITHheadTracker` or from a webcam via the `NITHwebcamWrapper` (both described below) without requiring significant changes to the application's core logic, and enabling source switching while the application is running. This flexibility is vital for tailoring interfaces to individual user needs and capabilities. Specifically, sensor data is represented as key-value pairs representing the actual sensor readings (e.g., `head_pos_yaw=21.6`), where values can be single numbers, booleans, strings, or a triplet indicating minimum, current, and maximum values for ranged parameters.

The core NITH libraries, including `NITHlibrary` and its extensions, are currently implemented in C# using the .NET 8 framework. This choice supports cross-platform compatibility, allowing applications to potentially run on various operating systems, such as Windows, macOS, and Linux. However, specific modules like `NITHemulation`, depend on Windows Presentation Foundation (WPF) framework[9], thus making them compatible only with Windows operating systems.

3.3 Hardware and Wrapped Peripherals

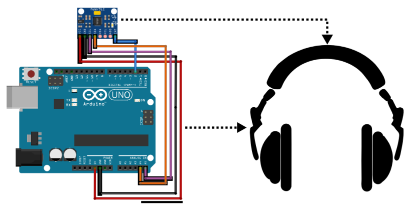
As mentioned, the NITH framework provides input data for HeadBower through custom-built Do-It-Yourself (DIY) hardware sensors[10], known as `NITHsensors`, and software `NITHwrappers`. HeadBower utilizes the following:

[8] Full list is available in the aforementioned NITH framework website: https://purl.org/nith-framework
[9] Windows Presentation Foundation documentation website: https://learn.microsoft.com/en-us/dotnet/desktop/wpf/
[10] All sensors and wrappers' DIY instructions and/or related software are available for download on NITH framework website: https://purl.org/nith-framework

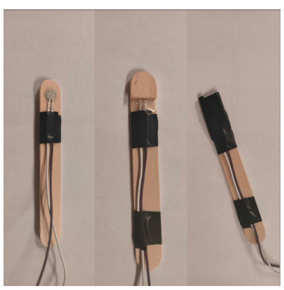
NITHheadTracker. A wearable NITHsensor, typically an Arduino with an Inertial Measurement Unit (IMU) like the BNO055 (Fig. 1a-b). It detects head orientation (yaw, pitch, roll) and angular accelerations. HeadBower primarily maps yaw to bow sweep/velocity (for intensity), and roll/pitch to modulations (e.g., vibrato). Accelerations can further refine intensity control.

NITHbiteSensor. A NITHsensor measuring bite pressure (Fig. 1c-d), constructed with a Force-Sensitive Resistor (FSR) and an Arduino. Its primary output, `teeth_press` (normalized bite force), is used experimentally for nuanced bow pressure control.

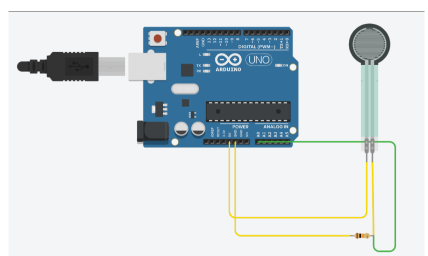
(a) NITHheadTracker (physical)



(b) NITHheadTracker (diagram)



(c) NITHbiteSensor (physical)



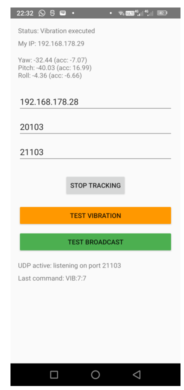
(d) NITHbiteSensor (diagram)

Fig. 1: NITH Hardware Sensors: (a) NITHheadTracker physical device, (b) its wiring diagram. (c) NITHbiteSensor physical device, (d) its wiring diagram.

NITHwebcamWrapper. A NITHwrapper using a standard webcam and Google's MediaPipe for real-time facial feature and head pose extraction (Fig. 2a). Relevant outputs for HeadBower include `mouth_ape` (mouth aperture for bow pressure modulation) and estimated head orientation (yaw, pitch, roll), providing a less precise alternative to the `NITHheadTracker`. An updated version integrates the `eyetrax`[11] library for webcam-based gaze coordinates (`gaze_x`, `gaze_y`), offering another navigation option.

NITHphoneWrapper. The `NITHphoneWrapper`[12] (Fig. 2b), an Android app, serves dually as a head tracking sensor and haptic feedback receiver. It uses the smartphone's motion sensors (rotation vector, gyroscope) for high-frequency sampling of head orientation (yaw, pitch, roll) and angular acceleration. Data, formatted per NITHprotocol (Sec. 3.2), is sent to a PC via UDP over LAN, with user-configurable IP/port. It also listens on a configurable UDP port for vibration commands (e.g., `VIB:duration:amplitude`), activating the phone's vibrator. The UI shows network/tracking status, sensor readings, and received commands. This component is crucial for HeadBower's haptic feedback, providing tactile cues for musical actions.

(a) NITHwebcamWrapper



(b) NITHphoneWrapper

Fig. 2: NITH Wrapper Interfaces: (a) NITHwebcamWrapper output (facial landmarks, mouth aperture, head pose). (b) NITHphoneWrapper Android UI (sensor readings, network config, haptic tests).

3.4 Eye Tracking Options

HeadBower's gaze-based note selection is flexible. The NITH framework offers support via: (1) `NITHbeamWrapper` for *Beam Eye Tracker*[13], a proprietary webcam-based software; and (2) `NITHwebcamWrapper` using the *Eyetrax* library (Sec. 3.3).

HeadBower also interfaces with any commercial eye tracker (e.g., Tobii, GazePoint) or gaze software controlling the system's mouse cursor. Its mouse-aware visual interface hides the system cursor to prevent distractions [10]. While

[11] Eyetrax repository on GitHub: https://github.com/ck-zhang/EyeTrax
[12] NITHphoneWrapper repository on GitHub: https://github.com/LIMUNIMI/NITHphoneWrapper
[13] Beam Eye Tracker website: https://beam.eyeware.tech/

webcam-based gaze tracking is accessible, dedicated hardware eye trackers (often IR-based) typically offer superior accuracy for nuanced control.

4 HeadBower Implementation

The system is developed in C# using the .NET 8 framework and Windows Presentation Foundation (WPF) for its graphical user interface, and operates within the broader NITH framework, leveraging its modularity and support for diverse interaction channels. HeadBower functions as a MIDI controller, specifically designed to explore expressive control of physically modeled synthesizers like the SWAM Viola by Audio Modeling (see Sec. 3.1), though its MIDI output can be remapped to control other MIDI-compatible sound sources. This focus on expressive control addresses a key challenge in ADMI development, where nuanced performance capabilities are sought [8]. A demo video of the instrument is available on the HeadBower GitHub repository, in the *Demo* folder[14]. In the following sections, we detail its implementation and the rationale behind its design choices.

4.1 User Interface Overview

The graphical user interface (GUI) of HeadBower (Fig. 3) supports the ongoing interaction and mapping experiments. The GUI provides the essential visual context for note selection through two distinct layouts and offers real-time feedback on various interaction parameters. Each on-screen button corresponds to a playable musical note. To aid interaction, the gaze point "snaps" to the nearest button, a technique akin to a "discrete cursor" which can be less distracting than a continuous one, as continuous cursors may introduce visual clutter especially given the nature of gaze data [10]. Furthermore, these buttons feature graphical cues, such as white text displaying note names, acting as "visual hooks" to help the user maintain stable fixation on the button's center [10]. The current GUI allows users to switch between the available note layouts and adjust basic operational settings such as sensor calibration, and can be entirely operated through gaze interaction and blinking actions (e.g., a long eye closure will trigger activation of settings buttons).

4.2 Visual Layouts for Note Selection

HeadBower employs gaze tracking for the selection of musical notes, offering users two alternative visual layouts. A foundational design principle across both layouts is the implementation of a *hybrid interaction* model [8]. This model is specifically chosen to mitigate the well-known "Midas Touch problem" [22, 29], where unintended activations can occur simply by looking at an interface element [49, 50]. In HeadBower, while note selection is performed via gaze, the

[14] HeadBower repository on GitHub: https://github.com/LIMUNIMI/HeadBower

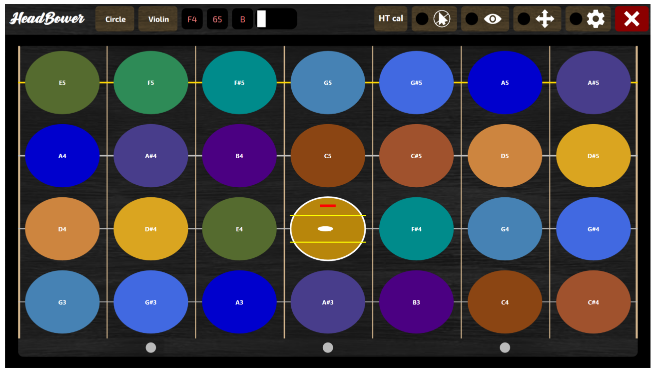


Fig. 3: HeadBower GUI with Violin Neck Layout. Top: main controls; Center: note buttons with overlay on active note (E4) showing bowing indicators.

actual sounding of the note (activation) is contingent upon a subsequent, deliberate head movement—specifically, a detected change in the "bowing" direction (detailed in Sec. 4.3). This decoupling of selection from activation is crucial for preventing inadvertent note triggers and fostering more intentional and controlled musical performance.

Demo videos for both the layouts are available in the *Demo* subfolder of the HeadBower GitHub repository[15].

Violin Neck Layout This layout, rendered within a dedicated canvas area in the GUI (Fig. 3), emulates a traditional violin fingerboard, and it's designed to: (a) be intuitive for users with prior violin experience; (b) enable knowledge transfer from previous violin repertoire. Notes are arranged on virtual strings and at corresponding positions, providing a visual metaphor that may be familiar to users with prior musical experience on string instruments. This familiarity can potentially lower cognitive load and enhance the "transparency" of the instrument's note organization [16]. A note button, when highlighted by the user's gaze, becomes armed for playing and is subsequently triggered by the physical head-bowing gesture. An interactive overlay can also furnish additional visual feedback directly onto this layout, reflecting the nuances of the bowing action.

Circle Layout The Circle Layout introduces an alternative approach where note keys are displayed in a circular arrangement on a canvas. This layout strategy directly addresses the Midas Touch problem by employing a form of dynamic "keys displacement" [10] designed to prevent the gaze path from unintentionally crossing intermediate, non-target keys when moving between notes—a common issue with linear layouts (such as a piano keyboard metaphor [10]). The layout

[15] HeadBower repository on GitHub: https://github.com/LIMUNIMI/HeadBower

incorporates a dual dynamic pattern to enhance selection accuracy and minimize interference: when the user gazes at a particular key, that key gradually moves towards the center of the screen. Concurrently, to prevent the entire interface from drifting, all keys are gently repositioned back towards the overall screen center. This dynamic behavior is intentionally designed to leverage the eyes' natural "smooth pursuit" capability [29]. Such "auto-scrolling," previously implemented in instruments like Netytar [13] and Netychords [12], is governed by exponentially decaying moving average smoothing filters. It not only aims to create a clearer focus on the intended target but can also improve detection accuracy by keeping the interaction predominantly in the central region of the eye tracker's view, where gaze detection performance is often optimal [20]. This design seeks to entirely eliminate interference from keys along the gaze path between targets. The demo video mentioned in the beginning of this section illustrates this strategy in action.

4.3 Head Interaction and Other Channels

The core of HeadBower's functionality lies in its mapping of performer actions to musical expression. This is primarily managed by a dedicated behavior module within the NITH framework, which processes data from various NITH sensors—including a head tracker and, optionally, sensors for mouth aperture and teeth pressure—and translates these inputs into MIDI messages (Note On/Off, CCs, Pitch Bend) to control a SWAM or similar strings synthesizer. The overarching goal is to achieve a high degree of expressive control, a recognized challenge in ADMI design [8].

The primary interaction metaphor maps head movements to the nuanced actions of a violin bow, employing various mapping strategies as defined by McGlynn [35] and discussed in the context of HeaDMIs [8]:

- **Bowing Motion and Intensity**: Horizontal head rotation (yaw) simulates the sweep of the bow. The velocity of this rotation, derived from filtered sensor data, is mapped to control the intensity of the synthesized note, typically via MIDI Expression (CC11). A faster yaw movement results in a louder, more intense sound. This is an "Excitation" mapping strategy [8], where the rate of change of a physical parameter directly influences a musical one. Such a mapping aligns with the ecological principle that the acoustic energy of an instrument should reflect the muscular energy injected by the performer [21], aiming for a more natural and engaging interaction.
- **Note Articulation (On/Off)**: Changes in the direction of the horizontal head rotation trigger MIDI Note On and Note Off events, serving as a "Trigger" mapping strategy [8]. This detection of directional change provides a clear, discrete event analogous to a violinist changing bow direction. A brief, configurable pause between strokes is incorporated to allow for natural musical phrasing and to prevent unintended rapid re-triggering, catering to varying user capabilities and expressive intentions, a key aspect of adaptive ADMI design [8]. When a new Note On is triggered for the currently gaze-selected pitch, the previously sounding note is sent a Note Off message.

- **Expressive Modulations**: Vertical head movements (along the pitch or roll axis) are mapped to various expressive MIDI parameters, selectable by the user. These can include MIDI Pitch Bend for subtle pitch inflections (a "Note level" control [8]), MIDI Modulation (CC1) for vibrato effects (a "Timbral level" control [8]), or another MIDI CC (e.g., CC9 for "BowHeight") to control parameters like bow pressure or other timbral qualities. This use of an orthogonal head movement axis expands the "Degrees of Freedom" available from a single input modality [2], enriching the expressive palette, which is crucial for performance-oriented ADMIs [37]. A neutral dead zone around the central head position is often employed to prevent unintended modulation from minor, involuntary movements, thereby enhancing control stability [8].
- **Supplementary Controls for Bow Pressure**: To further enhance expressive dimensionality, HeadBower explores supplementary channels for finer articulatory control:
  - Mouth aperture, detected by the NITHwebcamWrapper, can be mapped to MIDI CCs controlling bow pressure or other intensity-related parameters. This leverages a residual motor capability often available (Mouth-lip movements channel [8]) and can be conceptualized as a "Continuous range" mapping [8].
  - An experimental mapping integrates the NITHbiteSensor: teeth pressure can activate the "bow on string" state, with the pressure level potentially modulating bow pressure intensity. Releasing the bite could simulate lifting the "bow." This again utilizes a distinct physical channel (Teeth channel [8]) for nuanced control.

The translation of raw sensor data into MIDI messages involves crucial processing steps, including filtering for signal smoothing and data mapping to ensure smooth, musically responsive, and controllable output.

The mapping strategies were developed based on an ecological approach to instrument design, aiming to create an intuitive link between physical effort and sonic output. The head-to-bow metaphor was chosen as it mimics the continuous, expressive gestures of violin playing. This reflects a principle of energy correlation (Effort-to-Output) named in [8]: the physical energy the performer puts in is directly related to the sonic energy that comes out.

Overlay The overlay functionality (visible in both Fig. 3 and Fig. 4) provides immediate visual "extrinsic feedback" [28] related to the performer's bowing actions. It consists of a white ellipsis (representing the current bow position, moving horizontally in response to head yaw), a short red line (reflecting head movement on the pitch or roll axis, moving vertically), and two golden horizontal parallel segments. These segments act as visual thresholds; when the red line crosses them, various playing effects (like pitch bend or vibrato, depending on the current mode) are triggered. This design choice enhances the "transparency" of the mapping [16] from the performer's viewpoint, clarifying the relationship between their head movements and the resulting sonic modulations, which can aid in motor learning and control precision. Strategically, the overlay is positioned directly over the currently gazed-at note button. This allows the user to maintain their primary visual focus on the target note while simultaneously receiving feedback on their expressive gestures.

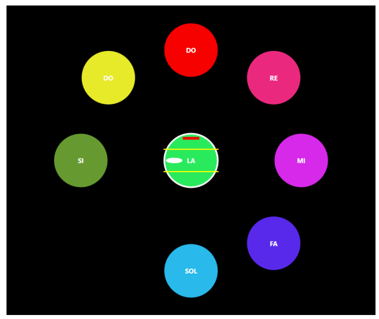


Fig. 4: HeadBower Circle Layout with Midas Touch mitigation. Central active note (LA) shows visual feedback overlay (bow position, modulation, thresholds).

4.4 Haptic Feedback Implementation

To address the common lack of rich tactile sensation in digital musical interactions [39], and specifically in many ADMIs [18], HeadBower integrates haptic feedback via a smartphone running the NITHphoneWrapper application. The velocity of the head-yaw gesture, which maps to bowing intensity (e.g. MIDI CC11), is translated to UDP commands (formatted as `VIB:duration:amplitude`) sent to the NITHphoneWrapper. These messages are received to trigger vibrations on the smartphone, with the intensity and duration of these vibrations proportional to the musical intensity generated by the performer's bowing gestures. This aims to restore a degree of "embodiment" and enhance the performer's sense of connection and engagement with the digital instrument [21], especially critical in a touch-less ADMI context. The smartphone is meant to be placed in any comfortable position on the user's body which the user has tactile sensation on (e.g., strapped to chest or arm, or to the neck), which can be adjusted and experimented based on individual preferences and capabilities.

4.5 Gaze Tracking Integration

As previously outlined in Sec. 3.4, HeadBower is designed to support various gaze tracking inputs for note selection. This includes dedicated hardware eye trackers as well as webcam-based solutions facilitated by the NITH framework (e.g., NITHbeamWrapper interfacing with Beam Eye Tracker, or the NITHwebcamWrapper with its integrated `eyetrax` library). Regardless of the specific source, HeadBower interprets the gaze input as mouse cursor coordinates. This is a pragmatic approach allowing the application's GUI elements, particularly the note buttons in both visual layouts, to respond to standard mouse-over events, which are effectively driven by the user's gaze point when an eye tracker with mouse emulation is active[16]. To maintain a clean visual field and minimize distractions, the system's mouse cursor is hidden.

A full overview of HeadBower's mapping strategy and interaction model is illustrated in Fig. 5.

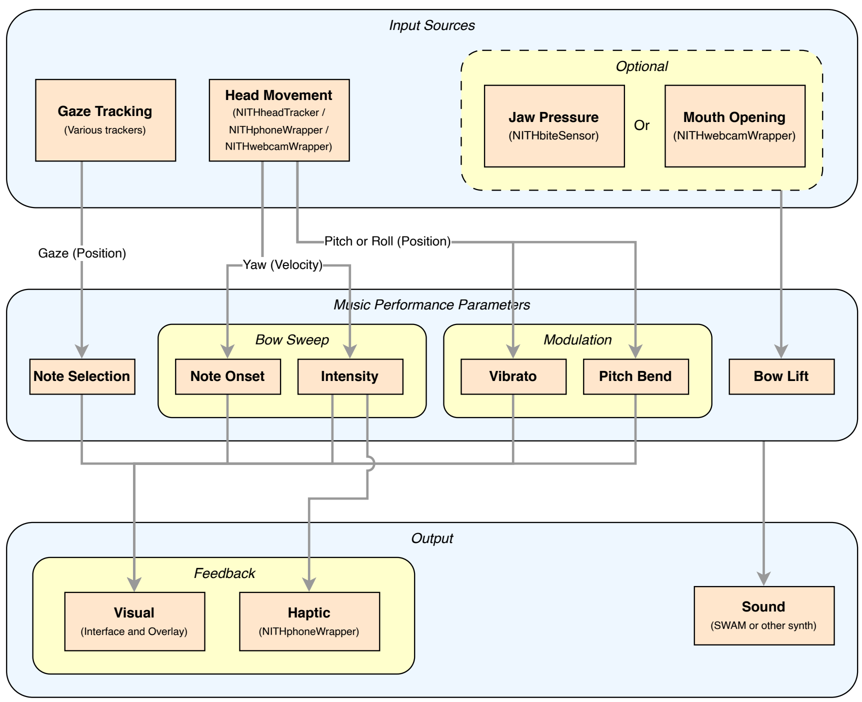


Fig. 5: Overview of the HeadBower interaction model and mapping strategies.

5 Conclusions and Future Work

This paper has presented HeadBower, an accessible music interface exploring expressive violin control for individuals with quadriplegia. HeadBower translates head movements into nuanced bowing dynamics (sweep, intensity, note articulation, and modulations like vibrato/pitch bend), augmented by supplementary

[16] Any HW eye tracker capable to directly control mouse positioning is thus employable.

sensors (e.g., jaw/mouth input) for finer bow pressure control. Note selection utilizes two gaze-tracking layouts designed to mitigate the "Midas Touch" problem, including a novel circular dynamic paradigm. A key contribution is the experimental integration of haptic feedback via smartphone vibration to enhance tactile sensation. Currently, HeadBower serves as an experimental platform for these mapping strategies rather than a finalized instrument.

The primary limitation is the lack of formal user evaluation. Future work must prioritize comprehensive studies with the target population to validate HeadBower's effectiveness, usability, and expressive potential. These studies will investigate, in particular: (RQ1) the efficacy of the head-to-bow mapping for emulating violin dynamics; (RQ2) the impact of smartphone-delivered haptics on engagement and expressiveness; (RQ3) the usability of combined gaze, head, and supplementary sensor inputs; and (RQ4) the comparative performance and user preference for the Violin Neck versus Circle layouts, especially concerning Midas Touch mitigation and cognitive load. Possible physical fatigue from the involved gestures will also be assessed. Evaluation will employ mixed methods: quantitative performance tasks [7] (measuring accuracy in timing, intensity, modulation) and qualitative approaches [4] (observations, think-aloud protocols, questionnaires like adapted MPX-Q [42]) to assess perceived control, comfort, and experience. Longitudinal studies will be invaluable for observing learning curves (especially for the complex head-bowing paradigm) and refining the system based on sustained user feedback.

Preliminary observations suggest that the head-bowing metaphor, coupled with haptics, can be an intuitive and engaging expression strategy, though some practice is required to master the coordination of gaze and head movements. While the violin neck layout can be promising for knowledge and repertoire transfer possibilities, the Circle Layout is promising for reducing unintended activations, achieving fast execution speeds without the need for detection filtering. Jaw sensing appears particularly stable and reliable, without the need for recalibration.

Future development will focus on several areas. The haptic feedback system requires refinement and users' feedback to optimize vibration patterns and their correlation with musical events, potentially developing a richer cue library and customization. Enhancing user-friendliness is crucial, including interfaces for remapping parameters, tuning sensor sensitivities, adjusting calibrations (e.g., dead zones, response curves), and managing personalized presets. The Circle Layout needs expansion for a wider chromatic range. Finally, creating targeted tutorials, exercises, and adapted musical pieces will support user learning and exploration of these interaction paradigms.

Our commitment to an open-source approach for both software and hardware aims to make these experimental tools and findings accessible, fostering broader contributions to accessible digital music.

Acknowledgements

The authors acknowledge the use of AI Large Language Model tools to support the preparation and revision of this manuscript. We would also like to express our gratitude to AudioModeling for providing us academic access to their SWAM software instruments.

References

1. Bailey, S., Scott, A., Wright, H., Symonds, I.M., Ng, K.: Eye.Breathe.Music: Creating music through minimal movement. In: Proc. Conf. Electronic Visualisation and the Arts (EVA 2010). pp. 254–258. ScienceOpen, London, UK (Jul 2010)
2. Birnbaum, D., Fiebrink, R., Malloch, J., Wanderley, M.M.: Towards a Dimension Space for Musical Devices. In: Proc. Int. Conf. on New Interfaces for Musical Expression (NIME'05). pp. 192–195. Vancouver, BC, Canada (2005)
3. Bosch, A., Stauffer, E.S., Nickel, V.L.: Incomplete Traumatic Quadriplegia: A Ten-Year Review. JAMA **216**(3), 473–478 (Apr 1971). https://doi.org/10.1001/jama.1971.03180290049006
4. Bottarelli, F., Davanzo, N., Presti, G., Avanzini, F.: DJeye: Towards an Accessible Gaze-Based Musical Interface for Quadriplegic DJs. In: Proceedings of the 2023 Sound and Music Computing Conference. Stockholm, Sweden (Jun 2023)
5. Costa-Giomi, E.: Effects of Three Years of Piano Instruction on Children's Academic Achievement, School Performance and Self-Esteem. Psychology of Music **32**(2), 139–152 (Apr 2004). https://doi.org/10.1177/0305735604041491
6. Davanzo, N., Avanzini, F.: A Dimension Space for the Evaluation of Accessible Digital Musical Instruments. In: Proc. 20th Int. Conf. on New Interfaces for Musical Expression (NIME '20). p. 7. NIME '20, Online repository, Royal Birmingham Conservatoire (Jul 2020). https://doi.org/10.5281/ZENODO.4813326
7. Davanzo, N., Avanzini, F.: Experimental Evaluation of Three Interaction Channels for Accessible Digital Musical Instruments. In: Proc. '20 Int. Conf. on Computers Helping People With Special Needs. pp. 437–445. Springer, Cham, Online Conf. (Sep 2020)
8. Davanzo, N., Avanzini, F.: Hands-Free Accessible Digital Musical Instruments: Conceptual Framework, Challenges, and Perspectives. IEEE Access **8**, 163975–163995 (2020). https://doi.org/10.1109/ACCESS.2020.3019978
9. Davanzo, N., Avanzini, F.: Resin: A Vocal Tract Resonances and Head Based Accessible Digital Musical Instrument. In: Proceedings of the 2021 AudioMostly Conf. Trento, Italy (online conf.) (Sep 2021). https://doi.org/10.1145/3478384.3478403
10. Davanzo, N., Avanzini, F.: Design Concepts for Gaze-Based Digital Musical Instruments. In: Proceedings of the 2022 Sound and Music Computing Conference. pp. 477–483. Zenodo, Saint-Etiénne, France (Jun 2022)
11. Davanzo, N., Avanzini, F., Ludovico, L.A., Moreno, D., Moreira, A., Tymoshchuk, O., Azevedo, J., Marques, C.: A Case Study on Netychords: Crafting Accessible Digital Musical Instrument Interaction for a Special Needs Scenario. In: Da Silva, H.P., Cipresso, P. (eds.) Computer-Human Interaction Research and Applications, vol. 1996, pp. 353–372. Springer Nature Switzerland, Cham (2023)
12. Davanzo, N., De Filippis, M., Avanzini, F.: Netychords: An Accessible Digital Musical Instrument for playing chords using gaze and head movements. In: In Proc. '21 Int. Conf. on Computer- Human Interaction Research and Applications (CHIRA '21). p. 8. SciTePress, Online conf. (2021)

13. Davanzo, N., Dondi, P., Mosconi, M., Porta, M.: Playing music with the eyes through an isomorphic interface. In: Proc. of the Workshop on Communication by Gaze Interaction. pp. 1–5. ACM Press, Warsaw, Poland (2018). https://doi.org/10.1145/3206343.3206350
14. Davanzo, N., Valente, L., Ludovico, L.A., Avanzini, F.: Kiroll: A Gaze-Based Instrument for Quadriplegic Musicians Based on the Context-Switching Paradigm. In: Proc. '23 AudioMostly Conference. ACM, Edinburgh, Scotland (Aug 2023)
15. Dobrian, C., Koppelman, D.: The E in NIME: Musical Expression with New Computer Interfaces. In: Proceedings of the International Conference on New Interfaces for Musical Expression. pp. 277–282 (2006). https://doi.org/10.5281/zenodo.1176893
16. Fels, S., Gadd, A., Mulder, A.: Mapping transparency through metaphor: Towards more expressive musical instruments. Organised Sound **7**(2), 109–126 (2002)
17. Feng, W., Sameki, M., Betke, M.: Exploration of Assistive Technologies Used by People with Quadriplegia Caused by Degenerative Neurological Diseases. International Journal of Human–Computer Interaction **34**(9), 834–844 (Sep 2018). https://doi.org/10.1080/10447318.2017.1395572
18. Frid, E.: Accessible Digital Musical Instruments—A Review of Musical Interfaces in Inclusive Music Practice. Multimodal Technologies and Interaction **3**(3), 57 (Jul 2019). https://doi.org/10.3390/mti3030057
19. Frid, E.: Diverse Sounds : Enabling Inclusive Sonic Interaction. Ph.D. thesis, KTH Royal Institute of Technology, Kollegiesalen, Brinellvägen 8, Stockholm (Jan 2020)
20. Holmqvist, K., Nyström, M., Mulvey, F.: Eye tracker data quality: What it is and how to measure it. In: Proceedings of the Symposium on Eye Tracking Research and Applications. pp. 45–52. ETRA '12, Association for Computing Machinery, New York, NY, USA (Mar 2012). https://doi.org/10.1145/2168556.2168563
21. Hunt, A., Wanderley, M.M., Paradis, M.: The Importance of Parameter Mapping in Electronic Instrument Design. Journal of New Music Research **32**(4), 429–440 (Dec 2003). https://doi.org/10.1076/jnmr.32.4.429.18853
22. Jacob, R.J.K.: Eye tracking in advanced interface design. In: Virtual Environments and Advanced Interface Design, pp. 258–288. Oxford University Press, Inc., USA (1995)
23. Jafar, M.R., Nagesh, D.S.: Literature review on assistive devices available for quadriplegic people: Indian context. Disability and Rehabilitation: Assistive Technology **18**(6), 929–941 (Aug 2023). https://doi.org/10.1080/17483107.2021.1938708
24. Jaschke, A.C., Eggermont, L.H., Honing, H., Scherder, E.J.: Music education and its effect on intellectual abilities in children: A systematic review. Reviews in the Neurosciences **24**(6), 665–675 (2013). https://doi.org/10.1515/revneuro-2013-0023
25. Jourdan, T., Caramiaux, B.: Machine Learning for Musical Expression: A Systematic Literature Review. In: Proceedings of the International Conference on New Interfaces for Musical Expression. pp. 319–331 (2023). https://doi.org/10.5281/zenodo.11189198
26. Lubet, A.: Music, Disability, and Society. Temple University Press (2011)
27. Lyons, M.J., Haehnel, M., Tetsutani, N.: Designing, Playing, and Performing with a Vision-based Mouth Interface. In: Proc. 3rd Conf. on New Interfaces for Musical Expression. pp. 116–121. NIME '03, NIME, McGill University, Montreal, Canada (2003)
28. Magill, R.A., Anderson, D.I.: Motor Learning and Control: Concepts and Applications, vol. 11. McGraw-Hill New York, 11 edn. (2017)

29. Majaranta, P., Bulling, A.: Eye Tracking and Eye-Based Human–Computer Interaction. In: Fairclough, S.H., Gilleade, K. (eds.) Advances in Physiological Computing, pp. 39–65. Human–Computer Interaction Series, Springer, London (2014)
30. Majnemer, A., Shevell, M., Hall, N., Poulin, C., Law, M.: Developmental and Functional Abilities in Children With Cerebral Palsy as Related to Pattern and Level of Motor Function. Journal of Child Neurology **25**(10), 1236–1241 (Oct 2010). https://doi.org/10.1177/0883073810363175
31. Man, D.W.K., Wong, M.S.L.: Evaluation of Computer-Access Solutions for Students With Quadriplegic Athetoid Cerebral Palsy. The American Journal of Occupational Therapy **61**(3), 355–364 (May 2007). https://doi.org/10.5014/ajot.61.3.355
32. Mandanici, M., Bergamino, G., Valente, S.: The I-Ork project: A technological approach to inclusive music making and therapy. Frontiers in Education **10** (May 2025). https://doi.org/10.3389/feduc.2025.1552302
33. Marshall, M.T.: Physical Interface Design for Digital Musical Instruments. Ph.D. thesis, McGill University, Montreal, QC, Canada (Apr 2008)
34. Matossian, V., Gehlhaar, R.: Human Instruments: Accessible Musical Instruments for People with Varied Physical Ability. Annual Review of Cybertherapy and Telemedicine **219**, 202–207 (2015)
35. McGlynn, P.: Interaction Design for Digital Musical Instruments. Ph.D. thesis, National University of Ireland, Maynooth (May 2014)
36. McMillan, A., Morreale, F.: Occupational Therapy Methods in the Design of Accessible Musical Instruments. In: Proceedings of the International Conference on New Interfaces for Musical Expression. pp. 339–345 (2024). https://doi.org/10.5281/zenodo.13904874
37. McPherson, A., Morreale, F., Harrison, J.: Musical Instruments for Novices: Comparing NIME, HCI and Crowdfunding Approaches. In: Holland, S., Mudd, T., Wilkie-McKenna, K., McPherson, A., Wanderley, M.M. (eds.) New Directions in Music and Human-Computer Interaction, pp. 179–212. Springer Series on Cultural Computing, Springer International Publishing, Cham (2019)
38. Meng, L.F., Li, T.Y., Chu, C.N., Chen, M.C., Chang, S.C.H., Chou, A.M., Yang, T., Hui, C.C., Chiao, K.A., Lin, Y.L., Weng, P.t., Shih, Y.c., Lu, T.y., Yeh, N.h.: Applications of computer access approach to persons with quadriplegics. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) Computers Helping People with Special Needs. pp. 857–864. Springer, Berlin, Heidelberg (2004)
39. Miranda, E.R., Wanderley, M.: New Digital Musical Instruments: Control And Interaction Beyond the Keyboard. A-R Editions, Inc., Middleton, Wis, 1st edition edn. (Jul 2006)
40. Munsat, T.L., Andres, P.L., Finison, L., Conlon, T., Thibodeau, L.: The natural history of motoneuron loss in amyotrophic lateral sclerosis. Neurology **38**(3), 409–413 (Mar 1988). https://doi.org/10.1212/wnl.38.3.409
41. My Breath My Music Foundation: The Magic Flute – The Head Sets the Tone. https://mybreathmymusic.com/en/magic-flute (2020)
42. Schmid, G.M.: Evaluating the Experiential Quality of Musical Instruments. Springer Fachmedien, Wiesbaden (2017). https://doi.org/10.1007/978-3-658-18420-9
43. Sears, A., Young, M., Feng, J.: Physical disabilities and computing technologies: An analysis of impairments. In: Sears, A., Jacko, J.A. (eds.) The Human-Computer Interaction Handbook, chap. 42, pp. 829–852. CRC Press, United States, 2 edn. (2008)

44. Smith, L., Lyons, F., Bridges, B., Casey, R.: WithFeelVR: The spatial and textural affordances of VR as a mapping strategy for an accessible digital musical instrument. In: Proceedings of the International Computer Music Conference. pp. 124–131 (Dec 2022)
45. Stensæth, K.: “Musical co-creation”? Exploring health-promoting potentials on the use of musical and interactive tangibles for families with children with disabilities. International journal of qualitative studies on health and well-being **8**, 20704 (Aug 2013). https://doi.org/10.3402/qhw.v8i0.20704
46. United Nations: Universal Declaration of Human Rights. https://www.un.org/en/universal-declaration-human-rights/index.html (Oct 2015)
47. Vamvakousis, Z.: Digital Musical Instruments for People with Physical Disabilities. Ph.D. thesis, Universitat Pompeu Fabra Barcelona (2016)
48. Vamvakousis, Z., Ramirez, R.: The EyeHarp: A Gaze-Controlled Digital Musical Instrument. Frontiers in Psychology **7**, article 906 (2016). https://doi.org/10.3389/fpsyg.2016.00906
49. Velichkovsky, B., A. Rumyantsev, M., A. Morozov, M.: New Solution to the Midas Touch Problem: Identification of Visual Commands Via Extraction of Focal Fixations. Procedia Computer Science **39** (Dec 2014). https://doi.org/10.1016/j.procs.2014.11.012
50. Velichkovsky, B., Sprenger, A., Unema, P.: Towards gaze-mediated interaction: Collecting solutions of the “Midas touch problem”. In: Howard, S., Hammond, J., Lindgaard, G. (eds.) Human-Computer Interaction INTERACT ’97: IFIP TC13 International Conference on Human-Computer Interaction, 14th–18th July 1997, Sydney, Australia, pp. 509–516. IFIP — The International Federation for Information Processing, Springer US, Boston, MA (1997)
51. Ward, A.: The development of a Modular Accessible Musical Instrument Technology Toolkit using action research. Frontiers in Computer Science **5** (Oct 2023). https://doi.org/10.3389/fcomp.2023.1113078