A Musical Biofeedback System for Balance and Gait Rehabilitation in Hemiparetic Stroke Patients

Designing Intuitive, Relevant and Flexible Interaction Concepts for the Clinical Environment

Master Thesis

Prithvi Ravi Kantan

Aalborg University
Department of Architecture, Design and Media Technology

This document was prepared in LaTeX using the AAU Report Template. Figures were created using draw.io and MATLAB.

Sound and Music Computing
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
A Musical Biofeedback System for Balance and Gait Rehabilitation in Hemiparetic Stroke Patients

Theme:
Extended Master Thesis (50 ECTS)

Project Period:
Autumn 2019 - Spring 2020

Project Group:
Prithvi Ravi Kantan

Participant(s):
Prithvi Ravi Kantan

Supervisor(s):
Sofia Dahl, Erika Spaich

Copies: 1

Page Numbers: 207

Date of Completion:
May 27, 2020

Abstract:

Balance and gait rehabilitation are critical to the recovery of motor function in hemiparetic stroke patients. Musical biofeedback has been shown to hold great rehabilitative potential due to its emotional appeal, capacity to induce and facilitate bodily movement and documented therapeutic benefits. This thesis aimed to investigate the types of user-tailored musical biofeedback interactions and strategies most applicable to common rehabilitation protocols. A prototype application based on wireless inertial sensors was built and iteratively evaluated over three development cycles in collaboration with patients and clinicians. Results showed that the developed interactions tailored to static balance, dynamic balance, sit-to-stand and gait may be clinically useful and usable with a number of stroke patient subgroups, promoting autonomy and augmenting conventional training. Future studies must systematically investigate short/long term physical and psychological effects.

Contents

Preface

This report was written in Spring 2020 in connection to a Master Thesis project undertaken over the 9^{th} and 10^{th} semesters of the Sound and Music Computing Msc. program at Aalborg University, Copenhagen. It was a continuation of project work from the previous two semesters of the program. The project was supervised by Assoc. Prof. Sofia Dahl (Department of Architecture, Design and Media Technology, AAU) as well as Assoc. Prof. Erika Spaich (Department of Health Science and Technology, AAU). Work from two course mini-projects was integrated into the project; The hardware component was developed as part of the Prototyping and Fabrication course project, while the initial sonification strategy set was designed as part of the Research in Sound and Music Computing course project (both in the 9^{th} Semester).

This project has been a lengthy process and undergone several phases, benefiting from support, guidance and direction from numerous sources on its course to completion. I would first like to thank my supervisor Sofia Dahl for her major role in this journey, which included setting up inter-department collaborations and arrangements, regular remote supervision, sincere, consistent and exemplary pedagogy as well as psychological support at many difficult junctures over the past year. I would also like to thank Erika Spaich for graciously agreeing to take the time and effort to co-supervise me, helping me settle in and work comfortably at the HST department in Aalborg, connecting me with relevant clinicians, sharp and precise guidance on navigating the unfamiliar clinical world and building a project with real-life potential. Next, I would like to thank Helle Rovsing Møller Jørgensen for her irreplaceable role in testing the project with real patients, and her incredible energy and enthusiasm in planning test dates, recruiting patients and conducting training sessions, all while patiently tolerating my Danish. I would then like to express my gratitude towards Peter Skotte, Jens Munk Clemmensen, Thomas Kristensen and Markus Löchtefeld at the AAU CREATE compus for their enthusiasm, creativity and skill in helping me build a robust hardware prototype, as well as John Hansen and Daniel Overholt for helping with software troubleshooting. I am also grateful to Naja Jensen and Laura Petrini for taking the time to meet with me at different stages and discuss qualitative evaluation methods. Next, I would like to thank the individuals who took part in the various stages of project evaluation, beginning with the 13 stroke patients at Neuroenhed Nord, Frederikshavn who showed the incredible kindness and patience to participate in interviews and device trials. For their participation in expert interviews, I thank Daniel Skak-Mazhari Jensen (UCN Aalborg), Abhishek Srivastava, Geeson David, Ashish Kasbe (Kokilaben Dhirubai Ambani Hospital, Mumbai), Hanne Pallesen, Gunhild Mo Hansen (Hammel Neurocenter), Kristian Aagaard, Katarina Guldager Skov and Mariann Borg Sømoe (Neuroenhed Nord, Frederikshavn). I thank the ten music producers who took part in the online survey. I would like to thank Mikael Skov and the steering committee of the Tech Talent program for designing and arranging such a wonderful series of learning activities, and for financing my participation in international conferences, all of which were invaluable learning experiences. I also thank my SMC coordinators for their continued support, encouragement and guidance throughout the course of my education. I am grateful to my friend Chetali Khadye for helping me proofread the final chapters. Last but not least, I would like to thank my parents Ravi Ramakantan and Priti Ravi for their steadfast and unwavering support, my friends in Mumbai, Copenhagen and Aalborg as well as my girlfriend Janhvi Sampat for helping me stay steady through all of it.

Aalborg University, May 27, 2020

Prithvik

Prithvi Ravi Kantan
<pkanta18@student.aau.dk>

Chapter 1

Introduction

Stroke is a leading age-related cause of death and disability worldwide. Resulting from damage to the central nervous system from vascular causes, it typically leads to hemiparesis or one-sided weakness of ranging severity in survivors [86]. The impact of stroke is far reaching at both a personal and societal level. Home confinement, dependence on others and reduced social contact are potentially devastating in terms of quality of life. Moreover, it has been estimated that in Denmark itself, there are presently 93,000 individuals living post-stroke, and annually 15,000 new stroke cases and 3600 stroke deaths [93]. Stroke also causes annual production losses of DKK 2,630 million and 600 early retirements per year [93]. Stroke survivors exhibit deficits that are both cognitive and physical in nature, generally requiring clinical rehabilitation in the period following the cerebrovascular event [87]. The physical deficits commonly manifest themselves as disturbances to balance and gait, both of which are critical to safe and independent daily functioning. The rehabilitation process invokes the plasticity of the brain, meaning its ability to rewire itself after trauma to regain function [87]. Rehabilitation is of paramount importance in the recovery of premorbid ability [63]. Recovery is usually not complete, with most survivors experiencing lifelong consequences. However, it has been established the adherence to rehabilitation is a key determinant of eventual clinical outcomes among patients [27].

In recent years, technology has taken on a greater role in both balance and gait rehabilitation, be it in the form of robotic applications [25], electrical muscle stimulation [45] or biofeedback [31]. This can be ascribed to multiple factors such as advances in affordable computer power as well as the advent and proliferation of lightweight and inexpensive motion-sensing technology [57]. The latter may, for instance, be based on force sensing or inertial measurement. It engenders intriguing possibilities in rehabilitation due to its abilities to capture and document mild disability (which is not otherwise visually apparent), monitor patients in non-clinical environments and provide them with immediate feedback through various sensory modalities [36]. This is called biofeedback, and serves to make physiological information available to conscious experience to allow for greater self-awareness and modification of bodily states where necessary [46]. Biofeedback is classified based on the type of bodily information being perceptualized [31], and the biomechanical variety (based on bodily kinematics) is of greatest relevance to stroke rehabilitation [46]. Indeed, different biofeedback designs and philosophies have been implemented and applied across multiple sensory modalities (visual, auditory, haptic) in both balance and gait rehabilitation, with short term physical effects that are often superior to conventional rehabilitation techniques [107, 91].

While the visual modality was predominant in early biofeedback applications, several researchers have made the case for the auditory and haptic media, citing advantages in terms of portability, cost, human reaction time and reduction of visual burden [57, 21]. The auditory modality has several other advantages in terms of temporal resolution and ability to process multiple data streams [43]. Auditory biofeedback thus involves the conversion of measured bodily information into a completely artificial or psychoacoustically designed sonic representation. By definition, it can thus be seen as a specific case of interactive sonification, where data relations are converted into auditory relations in real-time [35]. Depending on how the interaction in question is designed, the auditory information provides either continuous/discrete guidance or serves as a tool to guide error correction [46]. Auditory biofeedback has been shown to be effective in rehabilitation applications where sensory feedback information is important to physical performance (e.g. balance, gait), especially when the patient is deprived of one or more of the typical feedback channels (somatosensory, visual or vestibular) [21].

Although interactive sonification as auditory guidance has shown laboratory success, it has failed to attain widespread adoption [73, 68] or be integrated into common clinical protocols [91], both of which have a multitude of probable causes. One of these is a lack of focus on aesthetics in sonic interaction design [73], which can often lead to poor user satisfaction levels, fatigue and frustration from prolonged use [16]. This is particularly sub-optimal when designing technology for physically and cognitively frail patients who are undergoing a lengthy process that demands perseverance and patience. A solution is replacing the typically used simple auditory feedback designs by a universally appreciated stimulus such as music as the substrate for provision of biofeedback. Music, with its established socio-cultural appeal and emotional value, has well-known benefits in the exercise domain [44]. When used as a biofeedback medium, it has the capacity to motivate, monitor and modify bodily movement through an array of cortical mechanisms [59]. But perhaps most importantly, decades of research in the discipline of *neurologic music therapy* have documented direct therapeutic benefits of music across multiple dimensions in physical rehabilitation [99].

Auditory rhythms have been successfully used in the rehabilitation of rhythmic physical activities such as walking, where the rhythm serves as a cue for movement planning, execution and optimization [99]. This is called *Rhythmic Auditory Stimulation (RAS)* and has been shown to address several stroke-specific gait deficits [95]. The other musical dimensions of melody, harmony and dynamics can be used to provide temporal, spatial and force cues to train movement gestalts in another process called *Patterned Sensory Enhancement (PSE)* [99]. Functional movement patterns can regained by utilizing musical instruments in *Therapeutic Musical Instrument Performance (TIMP)*, which can further be reinforced by rhythmic patterning [99]. Modern technology can be used in music therapy as well, in the form of electronic MIDI-based instruments and digital audio workstations to create patient-tailored rehabilitation settings with the vast possibilities enabled by the digital domain [99]. Combining the psychological and therapeutic benefits of music with the portability, versatility and movement modification capability of auditory biofeedback enables powerful mediation of human behavior [52]. The use of musical biofeedback has not been widespread, but applications exist in gait (D-Jogger [64]), cardiovascular biofeedback (MoBeat [102]) and machine-based workout [29].

Designing musical biofeedback applications for stroke patients is a difficult proposition due to the diversity of the patient group and challenges experienced as part of interdisciplinary research [52]. These patients are usually elderly, and may face challenges when using musical biofeedback owing to perceptual, cognitive, attention and physical deficits. Therefore, it may only be a subset of stroke patients that are able to effectively use such technology, which must be designed to adapt to individual abilities and consistently provide feedback that is relevant, timely, intuitive and meaningful. The capability of such technology to provide clinicians with useful information about patient performance is an added benefit. To the best of the author's knowledge, there is no existing research on interaction design principles pertaining to musical biofeedback in stroke rehabilitation, and the present study attempts to address this gap. The goal is to target important conventional rehabilitation activities (static balance, dynamic balance, sit-to-stand and gait), build and assess technological movement-music interactions to augment these activities from the patient's and clinician's perspective. This is carried out through the user-centered development and evaluation of a prototype application capable of measuring relevant movement quantities for each activity, and converting these into meaningful musical biofeedback. The core tenets of the development are guided by existing literature, while evaluation is carried out in the form of real testing with patients, expert interviews with clinicians and system technical testing.

Initial Problem Statement: What types of musical biofeedback interactions can be effectively utilized in balance and gait rehabilitation of hemiparetic stroke patients?

Chapter 2 examines the relevant body of research and builds a theoretical foundation for the present study. Chapter 3 formulates the final research question. Chapter 4 synthesizes past research, discusses key early facets of system design and explains the methods used in the study. The subsequent chapters provide a detailed treatment of iterative development and evaluation. Findings through the course of the study are finally analyzed, synthesized and summarized in Chapters 10 and 11.

Chapter 2

Related Research

This chapter reviews scientific work relevant to the initial research question, beginning with an overview of stroke. Fundamentals of gait and balance will be briefly covered, followed by their post-stroke impairments and rehabilitation. Next, we will look at the role of modern technology as well as the potential of biofeedback, specifically auditory. Next, the case will be made for musical biofeedback as a powerful form of mediation technology, and music-based interventions in neurological therapy will be touched upon, followed by an appraisal of relevant musical biofeedback studies.

2.1 Stroke and Rehabilitation

2.1.1 Overview

Stroke is typically characterized as a "neurological deficit attributed to an acute focal injury of the central nervous system (CNS) by a vascular cause, including cerebral infarction, intracerebral hemorrhage, and subarachnoid hemorrhage, and is a major cause of disability and death worldwide" [86]. Impairments such as hemiparesis, incoordination and spasticity are the most common motor deficits post-stroke [86].

2.1.2 Balance - Fundamentals and Post-Stroke Impairments

The main functions of the postural control system are to build up posture against gravity, ensure its maintenance and fix the position and orientation of body segments that serve as a reference frame [61]. Multisensory inputs (visual, vestibular, proprioceptive and cutaneous) contribute to orienting the postural segments w.r.t. one another and the external world [61]. The largest balance degradation extents are seen with the deprivation of somatosensory information, followed by vestibular and visual [67]. The information from all three sources is redundant, which is crucial in the eventuality that one or more source is missing. The ability to manage such situations depends on *sensory integration*, or the ability of the CNS to evaluate and assess sensory information to form an internal representation of the environment [21].

Stroke patients usually exhibit characteristics such as abnormal muscle tone, abnormal movement control and dyscoordination between motor strategies [69, 90, 4]. The trunk is involved bilaterally in stroke patients, with trunk muscle function deterioration affecting proximal control [94]. These factors contribute to stroke patients typically experiencing problems in maintaining both static and dynamic balance. In addition, the *Sit-to-stand (STS)* transition, a fundamental pre-requisite to daily activities [9], is also commonly compromised [76]. Ordinarily, it involves the coordinated movement of the trunk and lower limbs, which have specific muscular activation patterns that result in the center of mass being transitioned from a relatively wide base of support in sitting to a smaller one in standing [83]. Stroke patients with hemiparesis often show a lack of coordination between hip and knee displacements at the end of STS, as seen from a kinematic analysis based on angular displacement and velocity data [2]. They exhibit abnormal muscle activation patterns [9] and generally exhibit longer rise times, which could be an indicator of fall risk [14].

2.1.3 Gait - Fundamentals and Post-Stroke Impairments

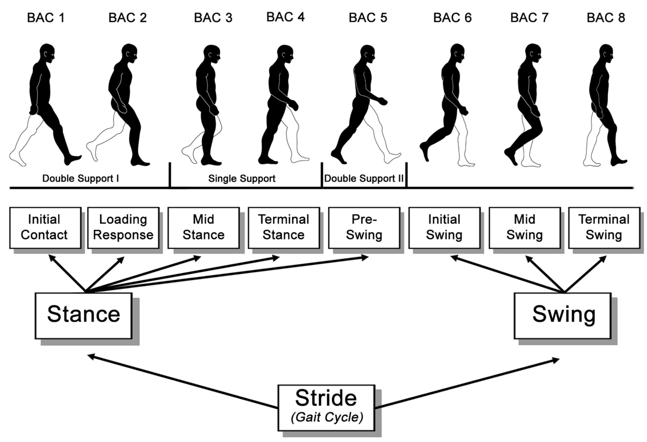

Figure 2.1: An illustration of the human gait cycle provided in [92] (licensed for free use).

Walking is a phenomenon that healthy individuals take for granted, but constitutes an extremely complex process of neuromuscular control [53]. Activation of muscles across the body in a certain spatiotemporal pattern is required to ensure appropriate joint positions to support and advance the body weight through the different gait phases [53]. It can be described as a progression of alternating weight-bearing limbs, with the body's center of gravity displacement viewed as the end result of all muscle forces acting on the body [53]. A more detailed account of kinematic determinants and neural control of normal gait is given in [53]. Basic stepping patterns are generated in the spinal cord, while fine walking control involves several brain regions [20].

A depiction of the gait cycle is given in Fig. 2.1. The basic unit for gait cycles is a *stride*, a sequence in which each foot alternates between ground contact (stance phase) and non-contact (swing phase) [99]. The stance phase accounts for about 60% of a normal gait cycle. A gait cycle is complete when each foot has completed its stance and swing phase, when the starting foot hits the ground again [99]. In each stride, there are two occasions when both feet are in contact with the ground (double support time) which collectively account for about 20% of the gait cycle. This is the most stable portion of the cycle, and thus tends to be longer in abnormal gait patterns [99]. A *step*, on the other hand, is measured from the time one foot hits the ground to the time the other hits the ground. The number of steps per minute is called *cadence* [99].

Hemiplegia is an important contributing factor to reduced gait performance post-stroke [6]. Stroke patients usually have a shortened stance phase and prolonged swing phase on the paretic side, with sub-normal walking speed and stride length [75]. Stroke survivors use fewer groups of co-excited muscles (modules) in the paretic limb while walking as compared to normal controls, causing them to walk more slowly and demonstrate more asymmetry [85]. A full spectrum of abnormality is seen clinically, depending on the level of muscle weakness, severity of spasticity, compensatory mechanisms and interactions between these [53]. Among these, spasticity and muscle weakness are most common and pose the most severe challenges for patient care [66]. Patients are classified as Fast, Moderate, Slow-Extended and Slow-Flexed walkers and exhibit impairments such as a lack of heel rise during terminal stance, excessive knee and hip flexion in mid-stance, hip hiking, leg circumduction and abnormal trunk leaning [66].

2.1.4 Rehabilitation

Brain plasticity is a broad term for the brain's ability to adapt to environmental pressure, experiences and challenges, including brain damage [42]. It occurs at many levels and in the case of stroke is facilitated by post stroke rehabilitative interventions [42]. Rehabilitation is essential for stroke survivors to recover mobility and function so as to live independently, participate in their community and experience fewer secondary complications [34]. Physical activity and exercise have been established to benefit stroke patients in terms of walking ability and balance [63]. Current literature suggests that ideal exercise intervention for stroke survivors includes individually-customized combinations of gait, balance and aerobic activities that are appropriate for the patient's level of impairment [63].

Standard motor rehabilitation includes neurofacilitation techniques, task-specific training and task-oriented training [87]. They encompass several approaches and focus on different aspects of motor retraining. The intensity of training varies considerably across patients, typically ranging in duration from 30-60 minutes per day early after stroke and tending to decrease with time [87]. The rehabilitation period varies depending on the degree of impairment and functional deficits [87]. Recovery has been observed to be most rapid in the first month post-stroke, slowing in subsequent months and plateauing by 6 months of rehabilitation, after completing which approximately 50-60% still experience some degree of motor impairment [34] and dependency [63]. Good outcomes are strongly associated with motivation and engagement from the side of the patient, and setting individual goals may be helpful in this regard [48]. Cognitive function and attention have also been identified as key determinant factors [62]. We now examine specifics of gait and balance in the context of stroke rehabilitation.

Balance Rehabilitation

Balance control is obtained through a unique combination of systems, and correspondingly requires task-specific complex rehabilitation [56]. The systematic review [56] examines a large number of studies of balance training on acute and sub-acute stroke patients, which focus on static and dynamic balance in both one-on-one and group-based therapy settings - concluding that there is moderate evidence of physical improvements with training. Trunk control and sitting balance are considered key predictors of functional outcome and hospital stay post-stroke [101], with several reviews concluding that trunk training improves them both (as listed in [18]). Aside from finding strong evidence that trunk training improves trunk control, sitting/standing balance and mobility, Van Criekinge et al [18] found strong carry-over effects among them, strengthening the notion that proximal stability is a prerequisite for distal mobility. The effect of STS rehabilitation has been reviewed [76] and there is moderate evidence that it improves STS duration and weight-bearing symmetry.

Gait Rehabilitation

Lower-limb rehabilitation programs in sub-acute patients mainly focus on gait training [41]. Classic rehabilitation techniques can be classified as neurophysical and motor learning [77], the latter of which includes modern techniques such as robotic rehabilitation, functional electrical stimulation and brain-computer interfaces. In neurophysical techniques, the patient acts as a relatively passive recipient while motor learning techniques entail active patient involvement [77]. Although there is insufficient evidence to state that one approach is more effective than the other, the combination of different strategies seems to be more effective than over-ground gait training alone [6].

2.2 Biofeedback

In a biofeedback system, a person's bodily functions, parameters and states are sensed, processed and relevant results are fed back to the person through one or more human senses. The person attempts to act upon this feedback to modify the respective functions, parameters or states in a desired fashion [46]. With recent scientific advances in inexpensive wireless sensor technology, there are now many types of sensors capable of quickly and accurately capturing body motion. These facilitate the process of providing immediate biofeedback to focus attention and enhance performance as a form of sensory substitution. Additionally, they allow the measurement and storage of mobility metrics in daily life not restricted to clinical settings. General factors to consider are monitoring accuracy, sensitivity to impairments and ease of use for therapists, physicians and patients [36].

Giggins et al. (2013) [31] categorize biofeedback systems as being either *physiological (PBF)* or *biomechanical biofeedback (BMBF)*. The former measures physiological systems such as the neuromuscular, respiratory and cardiovascular systems, while the latter entails measurements of movement, postural control and force. The application developed in this study is a BMBF system, and the discussion will exclusively focus on this system category. In rehabilitation, the motor learning focuses on reinstating natural movement patterns after injury [31]. Inertial sensors, force plates, electrogoniometers, pressure biofeedback units and camera-based systems can all be used to provide BMBF [31].

2.2.1 BMBF in Rehabilitation

Additional sensory information (i.e. biofeedback) on an individual's own motion may improve movement performance by serving as a substitute for the typical channels (somatosensory, vestibular, visual) in the nervous system's sensorimotor integration [107]. Movement performance improvements through biofeedback may be caused by sensory *reweighing* processes, in which the relative dependence of the central nervous system on different senses in sensorimotor integration is altered [37]. Biofeedback systems for posture and mobility in older populations are reviewed in [107]. There were indications for larger improvements after training balance, gait and sit-to-stand transfers with biofeedback than without it.

Many researchers have used contemporary technology such as force plates and motion capture systems to track center-of-pressure or center-of-mass of patients, providing feedback if its position exceeds a pre-defined range or dead-zone [57]. Although these devices are effective, their non-portability limits their applicability to indoor use only. This requires patients to visit hospitals or laboratories for therapy, leading to low adherence and impeding its use in real-world environments [57]. Ma et al. [57] also reviewed multiple studies that developed devices using wearable movement monitors including inertial sensors and force sensors to measure postural sway/tilt and ground reaction force respectively. They summarized that wearable sensors had the advantages of sufficient accuracy, low cost and portability, allowing them to become balance aids in daily life and replace conventional clinical instruments [57].

In the studies examined by the review [57], inertial motion sensors were used to measure postural sway or lower limb joint coordinations in the mediolateral (MLat) or anteroposterior (APos) planes during standing and walking. The majority of reviewed studies showed improvements in both static and dynamic balance, with a general trend that inertial sensors enhanced static balance, and plantar force sensors enhanced dynamic balance and were more suited to gait. Moreover, studies reviewed by Dozza [21] indicated that practicing static tasks had little potential to transfer performance improvements to dynamic tasks [65] and vice versa. This implies that two different biofeedback therapies, aimed to static and dynamic balance respectively may be necessary [21].

Stanton et al. [91] reviewed studies of biofeedback in lower limb activities among stroke patients, and concluded that biofeedback is more effective than usual therapy and could be used widely in rehabilitation, although long term learning effects remained unclear. The interventions used in their reviewed studies provided feedback on EMG activity, linear gait parameters and joint angles [91]. Biofeedback was visual, auditory or a combination thereof. Their assessment was a moderate effect of biofeedback in lower limb activities including walking, suggesting that information from biofeedback is can supplement therapist communication and promote autonomy [91]. Tate et al. [96] performed a more general review of biofeedback in gait retraining, and found moderate to large effects of kinematic, temporospatial and kinetic biofeedback relative to usual therapy. A highlighted limitation of the reviewed studies was a lack of long term retention testing.

Ma et al. [57] further suggested that conventional visual biofeedback could be discarded in favor of auditory or tactile biofeedback for the sake of portability. The sensors were placed in the lower back region near the location of the center-of-mass, or the shank/thigh. Static and/or dynamic balance were focused on, and the biofeedback modality was visual, auditory, vibrotactile or electrotactile. Balance evaluation was carried out on an *immediate* basis by both instrumented and non-instrumented tests, also summarized in [57]. As also pointed out by Dozza [21], new design trends are indeed moving in the direction of auditory and tactile biofeedback using inertial sensors with the intent of producing cost-effective and portable systems for balance training. Additional reasons favoring this direction are non-reliance on expensive and cumbersome monitors, power cabling and the fact that inertial sensors are one thousand times cheaper (and smaller) than force plates [21].

2.2.2 Designing Biofeedback Systems

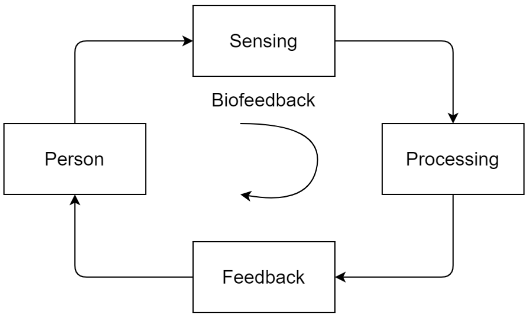

Figure 2.2: A high-level biofeedback loop inspired from the schematic given in [46]. The 'person' block refers to the cognitive processing and measurable actions performed by the biofeedback user within the loop.

Figure 2.2 shows the biofeedback operation as a cyclic process where the user (person) is part of the loop containing the other functional blocks of the system. The loop is thus closed if the user understands the feedback and acts as intended [46].

Success Criteria: For the feedback loop to be closed, the following conditions must be met [46]:

- Bodily parameter sensing is possible and available with sufficient accuracy.
- Relevant feedback information is computable.
- Appropriate feedback type is used.
- Feedback timing is suitable.
- Feedback is understandable to the user.
- Cognitive load of user processing feedback information is not too high.

Design Considerations: Biofeedback system design typically faces a number of challenges. As per Dozza [21], key among these is the determination of the variable to be fed back, and this should depend on the motor control mechanism, training task and therapeutic goal [70]. The presence of more than one relevant parameter to control and/or feed back during the task is another consideration. A task-oriented system should be able to feed back all relevant information to the user without being overwhelming or distracting [21]. Determining how to combine various types of information into one variable without being too cognitively demanding is paramount [21], and this is particularly relevant for stroke patients. Other challenges include designing a feedback representation that is easy to understand and learn, and does not interfere with task performance [43].

Assessment Considerations: Specifically addressing balance biofeedback, Dozza [21] highlights the importance of quantifying learning, retention and transfer effects resulting from training, in addition to simply immediate task performance effects. Depending on underlying pathology and other factors, some individuals may benefit from biofeedback more than others. Moreover, the extent of performance improvement due to biofeedback alone is hard to assess (as spontaneous learning also occurs simply by repetition of a task). A solution to this is the use of randomized controlled trials [21].

Implications for Current Study: Examining the success criteria in context with the available sensing technologies [36, 46], feedback modalities [21] and effect studies [91, 57], it is clear that the criteria of sensing accuracy, feedback type, timing, comprehensibility, cost-effectiveness and portability can be well accounted for by an auditory (or haptic) biofeedback system based on wearable sensors. The conversion of data relations into sound relations is called *sonification* [35], a process which adds expressive qualities to processes that otherwise lack the ability to be heard [43]. **By definition, auditory biofeedback is a form of real-time sonification categorized as *interactive sonification* [35], ultimately serving as a form of *auditory guidance***, and so we briefly discuss pertinent aspects of both these research disciplines.

2.3 Auditory Biofeedback as Interactive Sonification

2.3.1 Principles of Sonification

Interactive sonification is the subset of sonification defined as "the discipline of data exploration by interactively manipulating the data's transformation into sound" [35]. It represents the disciplinary overlap between the research disciplines of Human-Computer Interaction (HCI) and Sonification as a whole [35] (see Fig. 2.3). In sonification itself, the key aspects studied are the data transformation technique, the algorithmic implementation and the interaction itself. In biofeedback applications, (physiological) data features are mapped onto acoustical parameters of sonic events, which is termed as *parameter mapping sonification*. Interaction-wise, the considerations are similar to the success criteria of biofeedback applications in general [46, 35], which makes sense, considering that biofeedback in general and interac-

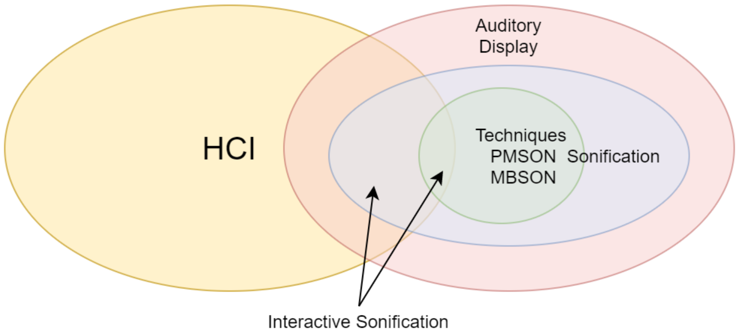

Figure 2.3: The disciplinary overlap between HCI and sonification, inspired from [35]

tive sonification in particular are intimately associated with control loops. Control loops with sound are critical to worldly experience, as they provide information about the environment and synchronize with visual and tactile assessments of objects [35]. The act of making sound may be satisfying to humans because they are in a very tightly responsive control loop, where actions are initiated and constant results are achieved [35].

A central challenge in sonification design is mapping data onto representational acoustic variables, with attention given to effectively conveying the intended message to the listener [35]. This is primarily determined by the auditory dimension mapping, polarity, scaling, concurrent presentation of data streams, aesthetics, training and perceptual/cognitive ability of the listener (requires special attention in cognitively impaired target populations such as stroke patients) [35]. In parameter mapping sonification, the potentially high dimensionality of data and acoustic variables affords a large design space, and effective designs are a compromise between intuitive, pleasant and precise sonic representations [35]. Careful data preparation, mapping function choices and mapping topology selection are crucial considerations in parameter mapping sonification design [35].

2.3.2 Principles of Auditory Guidance

Auditory biofeedback systems provide a user with information on bodily states or movements with the express purpose of *guiding* the user towards desired states or goals. Therefore, a brief background on auditory guidance is highly relevant. As per Sanz et al. [82], sonification designs for guidance purposes may be classified as either *psychoacoustic* or *artificial*, depending on the data representation technique used. The former leverages natural discrimination abilities based on spatial parameters which makes it more intuitive and natural. The latter maps data attributes to perceptual characteristics of sound such as pitch, loudness and timbre, which affords superior task accuracy but entails a longer learning process [73]. An example of artificial sonification is Chiari et al. [15], who used an IMU (inertial measurement unit) to convert horizontal trunk accelerations into 2D directional auditory feedback by manipulating the frequency, level and stereo balance of sinusoidal tones. Mapping functions (linear, exponential, sigmoid) were chosen to suit each respective auditory dimension. They too found balance improvements in healthy subjects [15].

Parseihian et al. [73] address several key aspects of guidance sonification design with a focus on 1D guidance tasks. A central concept is the dissociation of the data domain from the mapped auditory dimension. This means that rather than directly mapping data to the auditory dimension in real-time, the instantaneous data value is first compared to a 'target' value representing the *desired* system state, followed by normalizing by the maximum data value [73]. This converts the data into a dimensionless quantity that is 0 when the target is matched and 1 at the maximum distance from the target. This facilitates applications with multiple sonified variables having different units, scales and scaling function requirements [73]. A good example is Costantini et al. [17], who developed and tested an IMU-based biofeedback system that measured MLat and APos trunk inclination and sonified deviations from a stable central mean position with discrete auditory warnings. While not precisely the same, they converted 2D projections of trunk position into discrete auditory feedback zones and essentially sonified the distance of the trunk projection from the *target* upright zone [17]. The sounds used were simple - filtered and modulated noise. Although they only performed short-term evaluations with normal subjects, they found significant improvements in different conditions of sensory deprivation [17].

Parseihian et al. [72, 73, 74] characterize the information-conveying power of an auditory dimension in terms of its ability to guide the user to the target as quickly as possible, as accurately as possible and without passing the target. They also propose a taxonomy of guidance strategies:

- Basic Strategies: The auditory dimension value is a direct function of distance from target. The effectiveness of these strategies is constrained by human perceptual limits specific to each dimension.

- Strategies with Reference: These include a sound reference corresponding to the target, enabling target distance estimation without exploring the whole space. The reference may also be implicit, as with dimensions like harmonicity and synchronicity.

- Strategies with Reference and Zoom: These aim to increase precision around the target and reduce identification time by adding a 'zoom' effect, created by strategy duplication in multiple frequency bands [73].

Parseihian et al. [72] compared these different types of strategies in a 1-D guidance task, testing their efficacy in terms of guidance speed, accuracy and target overshoot. In the absence of a reference, basic strategies elicited the most overshoots, while accuracy correlated well with JND (just-noticeable difference) values of the tested auditory dimensions - pitch and tempo afforded greater accuracy than loudness and brightness [72]. Reference and zoom strategies maximized accuracy while minimizing target overshoots. They highlighted that while some strategies may provide precise guidance in a 1D task [73], they may be disruptive when combined with another strategy in a 2D or 3D task. The perceptual effects of combining several auditory streams must be considered in guidance design.

2.3.3 Sonification Aesthetics

A tendency of auditory guidance applications to use relatively simple sounds and auditory dimensions has significant implications for the real-world use of interactive sonification, especially auditory biofeedback. Parseihian et al. [73] observed that auditory guidance fails to find its place in commercial applications despite laboratory promise, and cited the lack of aestheticism in sonification design as a likely cause. Specifically, they attribute user rejection to auditory fatigue caused by the sounds used, or a lack of correspondence to the users' taste. Indeed, the notion of user satisfaction has been neglected in guidance aid research. As user tastes are diverse, Parseihian et al. [73] suggest the use of *morphocons* to convey information through the sonic *evolution* along auditory dimensions, rather than the actual sounds themselves. This sound-agnostic temporal manipulation makes it feasible to satisfy individual aesthetic preferences, and allow seamless switching among sound palettes without major changes in cognitive load or learning time [73]. The way in which sonic evolution conveys information has a large design space; relevant sound parameters may be manipulated either as an effect applied to real sounds or a control parameter to sound synthesis [73].

However, the topic of aesthetics in sonification is also a source of tension in a field that has been traditionally scientific in nature. This centers around the ambiguity introduced into data when codified using an aesthetic approach (e.g. music) [16]. More 'functional' signals (e.g. sine waves and noise) and simple auditory dimensions (e.g. pitch, loudness) have been opted for, owing to a preference for unambiguous data [5]. Neuhoff [68] advocates the 'bifurcation' of sonification into distinct (but not mutually exclusive) paths of either artistic or scientific sonification. Artistic sonification would focus on more aesthetic aspects of sonic representation, giving a sense of the underlying data but not always preserving precise data relations [68]. The other track of 'empirical' sonification favors auditory dimensions where individual differences are smallest and perceptual interactions are minimized [68].

The influence of aesthetics on user experience of an application is examined from an HCI perspective by MacDonald [3]. Several reviewed studies found evidence of a firm relationship between aesthetics and usability, with high correlations between these two types of ratings - leading to the conclusion that 'what is beautiful is usable'. A study [19] even found that users seemed to disregard usability problems when aesthetics were rated highly. The conclusion was that the aesthetic appeal of a system mediates the perceived usability and usefulness in a complicated fashion, providing evidence that these constructs comprise both pragmatic and hedonic elements [3]. These studies highlight the importance of aesthetics in HCI in general as well as interactive sonification and auditory biofeedback in particular. The present study must pay close attention to interaction aesthetics by choosing a distinct design path and leveraging principles such as the use of morphocons to address the aesthetic preferences of a diverse population.

2.3.4 Sonification Research in Balance and Gait

We now examine some relevant interactive sonification applications in balance and gait. Dozza et al. [22] used simple auditory dimensions in a force plate-based study with patients having bilateral vestibular loss, and found significant reductions in postural sway. In a separate study, Dozza et al. [21] found that direction specificity of audio biofeedback reduced postural sway and increased the frequency of postural corrections in the direction of the biofeedback. They discovered that the optimal mapping function for trunk sway to auditory v/s visual biofeedback is different - sigmoid for audio and linear for visual, indicating that each modality may encourage a different type of postural sway strategy [23]. Engardt et al. [24] assessed the long term effects of auditory biofeedback training during sit-to-stand transfers for hemiparetic stroke patients. They found short term improvements in body weight distribution, but re-tests done a few years later revealed significant worsening [24]. They discussed how instantaneous feedback could cause reliance and be detrimental to learning in the long term. They explained that this could partially be due to a fixation on the feedback rather than the development of a robust internal representation for loading the paretic leg [24]. They suggested that auditory biofeedback be incorporated into therapy with 'limited frequency', with 'booster' sessions over longer periods of time [24].

There are also corresponding examples in gait rehabilitation. As reviewed in [88], instrumented footwear-based [58] interactive sonification systems have been used with Parkinson's Disease patients. These are essentially shoes with sensors that collect information, triggering auditory cueing stimuli to inform about the user's current state. Rodger et al. [84] tested systems using synthesized walking sounds to enhance gait coordination in PD, and their results displayed an effect on step length variability. Torres et al. [100] introduced an IMU-based system and prescribed a number of movement-sound couplings, such as fixed movement thresholds to trigger discrete auditory feedback or modulate continuous auditory feedback to name a few [100]. Example interactions targeting gait were based on ankle dorsiflexion, knee hyperextension and leg speed variability. Bresin et al. [11]

built and evaluated a system for the expressive sonification of footsteps, finding that harder sonic textures tended to promote aggressive walking patterns and vice versa.

To summarize, several auditory biofeedback studies in balance rehabilitation have shown significant reductions in postural sway, whether provided on the basis of IMU or force plate data. Direction specificity has been shown to be beneficial, and an optimal mapping function shape has also been suggested [23]. Studies that gauge long term retention are, however, generally lacking. As for gait, there is evidence for the efficacy of instrumented footwear and synthesized walking sounds, as well as suggestions for possible interactions. It is clear that IMU-based systems can be used to capture important measures of both gait and balance for biofeedback purposes. Another general tendency among the reviewed studies was that of using artificial sonification paradigms with simple synthesized sounds, indicating on one hand that these mappings can be effectively learned by patients and pointing to the inherent aesthetics problem on the other.

2.4 Music in Rehabilitation - The Case for Musical Sonification

The main assertion of this section is that music not only addresses the issue of aesthetics in sonification but also serves as an effective biofeedback medium due to its positive physical and psychological effects during exercise (Karageorghis et al. [44]). More importantly, the evolving field of neurologic music therapy has demonstrated that music can effectively be applied in the field of rehabilitation as well.

2.4.1 Why Musical Sonification?

Maes et al. [59] put forward the hypothesis that music is a highly convenient way to present biofeedback on physiological processes, motor kinematic or kinetic processes and performance parameter output. They present the core functions of music in the biofeedback context as the 3MO model - namely to *motivate, monitor and modify* movement towards specific goals based on reinforcement learning processes [59].

Motivation Motivation is imperative in situations that demand high endurance and perseverance, such as balance and gait rehabilitation [59]. The premise is that sonification (biofeedback) can take advantage of the strong motivational qualities inherent to people's interactions with music. A known phenomenon is the ability of music to induce physical movement through arousal and motor resonance mechanisms [59]. Music has also been found to affect the limbic system and, in turn, emotional states [40], although this is tied to personal traits, preferences, familiarity and autobiographical memory [47]. However, certain surface features of sound such as tempo, consonance, mode and texture have been found to affect musical responses more or less universally [104] and may be used for biofeedback purposes. Park et al. [71] investigated how emotional states influenced forward gait based on familiarity with the music selection. Using consonant and dissonant versions of both familiar and unfamiliar music to participants, they found that familiarity with music interacted with emotional responses to influence gait kinematics. Gait velocity was significantly greater in the familiar-consonant condition relative to familiar-dissonant. However, this difference was *not* observed between the unfamiliar-consonant and unfamiliar-dissonant conditions [71]. This indicates that familiarity is an important mediator of motivation in the face of pleasant or unpleasant modifications to music.

Monitoring Maes et al. [59] outline strategies for multilayer sonification to monitor physiological and kinematic parameters. The first leverages human auditory parallel processing ability by assigning different layers of auditory feedback to different physiological processes to increase awareness of their interplay in relation to performed output. They give the example of an orchestration of muscle synergies being sonified as a well-organized auditory 'symphony' [59]. The second strategy relies on perceptual fusion effects, wherein different auditory layers blend into a single auditory perceptual object. The focus here is on the cumulative outcome of all physiological processes, instead of the explicit contribution of each. The idea is that optimal coordination of processes should lead to a pleasing auditory outcome and vice versa [59]. The third strategy is related to temporal periodicity-based fusion. Music often contains repeated patterns with integer-related periodicities, which can be related as phase-locked oscillators [59]. Several physiological and motor patterns also exhibit a periodic nature, suggesting that music may assist in synchronizing biological oscillators at different periods [59].

Modification Maes et al. [59] summarize two ways in which eventual modification of motor behavior may occur. The first is guided by reasoning, and requires that the learner has an explicit representation of the target behavior, to which ongoing behavior can be compared. Learning comprises minimizing the error between ongoing and target behavior [59]. The ability to modify behavior is thus directly governed by the ability to monitor, and this is the typical approach in motor learning. The second approach is based on reinforcement learning and does not require that the learner has an explicit representation of the target behavior [59]. Pleasant and rewarding states promoted by music would then serve as attractors of motor behavior, operating based on mechanisms of brainstem-driven reward and predictive processing-driven reward [59]. A prime example is auditory-motor synchronization, which depends on the ability to predict the time at which a musical beat is to occur. Successful prediction can lead to strong feelings of pleasure and control or ‘agency’ [59]. These benefits have been leveraged in technological applications such as the D-Jogger [64] and IM Gait Mate [39].

Novelty, Surprise and Expressiveness From current learning theories, music that is too repetitive, simple or conventional will not sustain reward responses [59]. Dopamine (responsible for feelings of reward) is maximally released when the uncertainty of a reward outcome is maximum, and vice versa [26]. This is relevant both in the realms of music composition and biofeedback systems, where it is necessary to include elements of surprise and novelty to support learning, self-regulation and motivation [59]. Expressiveness, too is an important affordance of music in that it affords expressive responses to it. For instance, activating and relaxing expressions in music [51] have been shown to influence walking velocity.

With these benefits of music, it is appropriate the musical stimulus is constantly present throughout the course of an exercise. Biofeedback can be provided by manipulating this music in a continuous or discrete fashion as the case may be. Aesthetically, a stimulative [44] or activating [12] expression appears most suitable, and this has implications for biofeedback system design. Using pre-existing music is advantageous in that it can be catered to patient preferences and its playback is computationally inexpensive. However, its artifact-free manipulation is complex and there is limited fine system control over its constituent musical elements. Real-time synthesized music conversely affords straightforward control over not only its rhythm, tempo and pitch but also individual instrument tracks and their sequencing, synthesis and effect parameters. This allows the creation of interesting and powerful biofeedback interactions.

2.4.2 Neurologic Music Therapy

Neurologic music therapy (NMT) is defined as "the therapeutic application of music to cognitive, affective, sensory, language, and motor dysfunctions due to disease or injury to the human nervous system" [99]. It is based on neuroscientific models of music perception and production, as well as the influence of music on changes in non-musical brain and behavior function. An advantage of using music in therapy for the elderly is that music is painless, non-intrusive, easily accessible and cost-effective [52]. Treatment techniques are adapted to suit the patient’s needs, and directed towards non-musical therapeutic goals. Translational biomedical research in music has led to the development of clusters of evidence showing the effectiveness of certain interventions, which were later classified into a system of about 20 techniques that make up NMT, such as Rhythmic Auditory Stimulation, Patterned Sensory Enhancement and Therapeutic Musical Instrument Performance (see Appendix A). Music is processed by the brain in a highly distributed fashion from spinal and subcortical areas to cognitive and motor control centers. The theoretical models of NMT are fundamentally based on an understanding of music perception processes. NMT techniques and principles are compatible with both traditional and new concepts of motor rehabilitation grounded in motor learning rules [99].

2.4.3 Musical Biofeedback Research

Studies employing musical biofeedback are rare, and some relevant ones are reviewed here - although these are not necessarily stroke rehabilitation-related. Some studies performed simple manipulations of existing music pieces, for example by adding noise [8] or adjusting audio quality [33] to sonify respiratory rate. Lorenzoni et al. [55] sonified running cadence compliance through the addition of noise to pre-selected songs of the participants' preferred genre. They found that the feedback was capable of altering running cadence significantly better than verbal instructions. In a pilot study, Schedel et al. [89] tested the capability of rhythmic distortion, timbral distortion and white noise added to the preferred music of Parkinson's Disease patients. They found that the patients could perceive these distortions and utilize this information for error correction with similar speed and accuracy to healthy peers. The D-Jogger [64] sonifies detected step cadence by synchronizing pre-existing music to detected gait patterns through digital signal processing. The tempo and phase manipulations make it possible to walk/run in time with the beat. Sensors measure gait timing, and tempo-appropriate music is selected [64]. A phase vocoder then adjusts the tempo and phase of the music to match the gait timing. Interaction with the D-Jogger is founding to have a strong rewarding effect as the synchronization provides energizing and satisfying feelings of agency [52]. The 'Jymmin' system developed by Fritz et al. [29] provides musical biofeedback when interacting with fitness machines, by mapping movements of the machine to parameters of effects acting on electronic dance music loops. These included band pass filters and pitch-shifters in Ableton Live. On experimentally comparing the use of this system with passive music listening during exercise, they found that using the system reduced perceived exertion [29], pain [28] and improved mood [30].

Other studies employed real-time synthesis approaches, through which it is easier to exercise finer control over sonic parameters of music. Gorgas et al. [78] mapped gait characteristics to musical notes, which led to improvements in cadence and velocity in Parkinson's Disease patients. Bergstrom et al. [7] tested the efficacy of music as a sonification signal in arousal modulation. Their design sonified measured heart rate through changes in music tempo and amplitude, and their evaluation compared this design with plain music listening and a simple sine pitch sonification. They concluded that the effects of musical sonification for arousal modulation were superior to those of music alone, and as effective as the sine sonification. They noted that the musical design had the added advantages of drawing attention and providing variety in the feedback signal, which would serve to reduce auditory fatigue [7]. Yu [106] sonified heart rate as speed, emphasis and inter-beat delay in arpeggio chords and note pairs respectively. While the system was found to be effective at its purpose, participants, however, found the biofeedback stressful possibly due to the sonification design strategy and unfamiliarity of the audio forms [106]. The author emphasized that the simplicity of the musical biofeedback was not comparable to the richness of a properly composed piece of music, which may have also led to tiredness among participants [106]. The 'moBeat' system developed by van der Vlist et al. [102] provided heat-rate biofeedback during a cycling exercise by supplying pedalling-synchronized synthetic music and giving feedback on training intensity compliance through altering the richness of musical layers. In addition, synthetic tones were given outside the compliance zone to direct the user on cycling speed. The system was found to be comparable to a reference system in terms of compliance [102]. They found that the music provided a natural distraction away from the exercise itself as seen through fewer distress cues, in addition to eliciting greater motivation overall. The authors did, however, stress the importance of providing preferred music to the users [102].

2.5 Defining an Interaction in the Present Context

Although simple auditory biofeedback has been researched with stroke patients, there lacks a design framework for musical biofeedback applications catering to this group. In other words, it is neither known how movement-music interactions can best be designed to fit into existing gait and balance training regimes, nor how effective these interactions are from a therapeutic standpoint. The former is a necessary precondition for the latter.

It is important to define the term 'interaction' more specifically for the purposes of the present study, as the term can refer to entities that hold distinct meanings depend on the purpose and context. The way in which the term is construed will influence the notion of what constitutes a good interaction and in turn, the thought process underlying interaction design [38]. In an essay, Hornbæk et al. [38] discussed the various ways in which human-computer interaction can be defined, formulating interaction as dialogue, transmission (of a message over a noisy channel), tool use, embodiment, experience and optimal behavior. Glancing at key phenomena, constructs and good interaction characteristics of each of them, it is clear that they have much in common and that most interactions can be seen as a combination. A musical biofeedback system for stroke patients, for instance can closely related to the following subset of these terms [38]:

1. **(Patient Perspective)** A control system that "interactively minimizes (move-

ment) error against a reference (state)".

2. **(Patient Perspective)** An "ongoing stream of expectations, feelings and memories", or an experience.
3. **(Clinician Perspective)** "a sender (patient) sending a message (movement information) over a noisy channel".
4. **(Patient Perspective)** "Acting and being in situations of a material and social world", or embodiment.

As the theoretical foundation of the present study constitutes biofeedback loops, auditory guidance, aesthetics and musical experiences for therapeutic purposes, the first three definitions seem to fit it readily. However, the embodied perspective of lived experiences cannot be neglected due to its profound impact on user experience, particularly for a target group with cognitive impairments. Acknowledging this, the majority of interaction design is anticipated to be performed from a third-person perspective as a collaborative design process is not feasible at present. Therefore, 'interactions' henceforth imply the first three definitions, with "rapid and stable convergence to target state", "satisfaction of psychological needs and motivation" and "maximum throughput of information" respectively constituting a good interaction [38], although the patient perspective is of primary importance.

2.6 Own Past Work

In previous work [79], we developed a proof-of-concept application which synthesized a multitrack ensemble of instrumental electronic music. Gait was captured using single bilateral force sensors, and temporal deviations were directly sonified as unpleasant modifications to the music such as noise, disturbance notes and ring modulation effects. The music was pre-programmed and basic, and it was not possible to customize mappings or adjust the system to individual abilities. Evaluation showed that the sonifications were hard to perceive. A second more elaborate proof-of-concept system [80] was developed with a series of foot-switches. This time, the music was generated in a pseudo-random manner, but still basic and electronic-sounding. Temporal gait deviations were sonified by modifying energetic qualities of the instruments designed according to motor-mimetic embodiment theories. The gait parameter mappings were individual-baseline specific and customizable. Pilot tests showed that the sonifications were easily perceptible for young, healthy individuals, but that the foot switches and physical hardware were cumbersome and sometimes uncomfortable or restrictive [80]. In general, the music also received poor aesthetic ratings. Clinical tests with real patients were not conducted in either study, and the hardware prototypes were generally fragile in the face of wear-and-tear.

Chapter 3

Problem Analysis

At this point, we define the broad goal of this research as the design, development and evaluation of a biomechanical biofeedback application that provides real-time functional kinematic feedback through the medium of music, with particular focus on the development of movement-music interactions suitable to therapy.

3.1 Outcomes, Delimitation and Final Framing

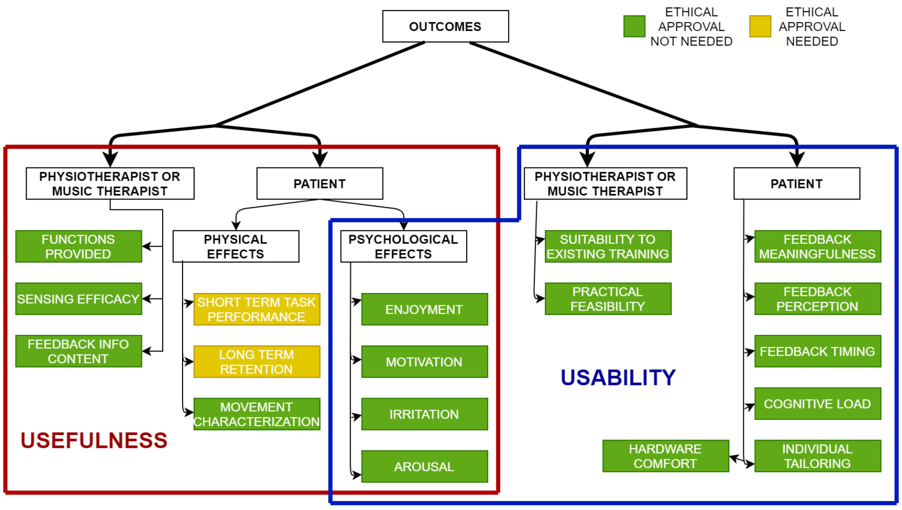

Figure 3.1: A tree diagram of the problem framing, with the various outcomes classified on the basis of usability and usefulness from patient and therapist perspectives.

The evaluation of the application will focus on the assessment of usefulness and usability from the perspective of the patient as well as the clinician (physiotherapist or music therapist). The success criteria of the application can be formulated by reframing the basic biofeedback success criteria from [46] using HCI concepts of *usefulness* and *usability* [3] to accommodate the perspectives of both patients and clinicians. These two terms have much in common but a key distinction. Usefulness is defined as "the extent to which a system's functions allow users to complete a set of tasks and achieve specific goals in a particular context of use". Usability, on the other hand, pertains to whether the system does so with "effectiveness, efficiency and satisfaction" [3].

On the basis of these definitions, the success criteria of the biofeedback application from both perspectives are re-framed in the current context and classified into usefulness and usability criteria, as depicted in Fig. 3.1. Most of these criteria are either self-explanatory or have been discussed, with the exception of usefulness criteria from the therapist perspective. These primarily relate to whether or not the application is capable of sensing all movement behaviors relevant to the activity (including phenomena that are hard to perceive visually), and making them explicit through the musical feedback. Psychological effects on patients can be seen on one hand as usefulness criteria as they are positive outcomes resulting directly from the interactions, and on the other as usability criteria, as they facilitate the achievement of superior task performance and movement quality. If this diagram were to be condensed into a problem statement as per PICO(T) guidelines, it would appear as follows:

Original Problem Statement: How does the application of a user-tailored musical biofeedback system impact physical movement parameters and subjective experience in balance and gait rehabilitation of hemiparetic stroke patients? How useful is the auditioning of movement phenomena to a clinician?

And what music interaction schemes and feedback strategies are most suited to common training activities in terms of meaningfulness, perceptibility, timing, individual tailoring, cognitive load, relevance and practical feasibility?

Necessary Delimitation: In Fig. 3.1, criteria are also categorized on the basis of whether ethical approval is required to evaluate them or not. Enquiries to the Region Nordjylland Ethical Committee helped clarify that the majority of criteria would *not* require official ethical approval in order to be evaluated, with the exception of **physical effect measurements**, both short and long term. Although an ethical application was framed, it could not be submitted to the committee for approval within the time-frame of this project, but will be submitted to allow future work to proceed. Another constraint was the fact that the project did not have the financial support to enlist the assistance of physiotherapists for extended periods of time for testing. All evaluation therefore had to be conducted in limited time-frames with small numbers of patients (5-7 in a single day per iteration). In the light of these restrictions, the scope of the research was modified to exclude effect measurements altogether, and focus exclusively on the design and development of intuitive, relevant and flexible interactions for gait and balance training. The remaining success criteria were left intact, and the final problem statement is reformulated as follows:

In balance and gait rehabilitation of hemiparetic stroke patients, what music interaction schemes and musical biofeedback strategies are most suited to common training activities, in terms of subjective experience, meaningfulness, perceptibility, timing, individual tailoring, cognitive load, relevance and practical feasibility? How useful is the auditioning of movement phenomena from a clinician's perspective?

Chapter 4

Methods

We now define a set of broad requirements that the developed technology must fulfill:

- Generation of suitable and user-customizable musical stimuli.
- Non-invasive, lightweight and comfortable movement sensing hardware capable of capturing the required kinematic data for biofeedback purposes.
- An available set of intuitive and perceptually salient musical feedback strategies.
- Real-time mechanisms for relevant kinematic parameter calculation from the raw sensor data.
- Flexible and user-customizable mapping from kinematic parameter domain to auditory feedback domain.

4.1 Research Methodology

Fulfilling the above requirements will involve facing domain-specific challenges related to designing music technologies for healthcare, particularly when elderly people are involved who are primarily non-musicians. Problems with budgets, ethical constraints, logistics and healthcare system structure, variability in the abilities of patients, as well as the stigma experienced by prospective participants of being approached as patients are challenges to this form of research [52]. These unfamiliar technologies can also bring ethical concerns and confidence issues, which must be addressed [52]. For these reasons, a *participatory* approach to this research appears most suitable.

As reviewed by [54], old people have traditionally been categorized as research 'subjects', pointing to an imbalance of power between them and the researcher. Participatory research (relevant when the research is conducted in collaboration with the group being studied) ethically values the capabilities of the elderly and advances their autonomy, allowing them to appraise project relevance and increasing the adoption of research outcomes [54]. Elderly individuals generally participate in research with the intention of giving, as well as social participation to combat loneliness. Recruitment and retention of participants over the course of a study can be challenging, and it is important for researchers to be respectful, flexible and appreciative of the diversity among the elderly [54]. A more detailed treatment of the matter can be perused in [54].

Lesaffre et. al. [52] explain how participatory *user-centered studies* are gradually seen as the staple research methodology for music-based mediation technology. This is, in brief, "a joint activity of a cross-disciplinary team of stakeholders that cooperate throughout the entire research procedure" [52]. The design process is guided by principles of participatory design, meaning stakeholders with different areas of expertise have a deciding vote in the design process. Target patient groups must be narrowed down to patients who enjoy music, are sensitive to music reward experiences, have a positive advantage towards new technology, have adequate motor skills and so forth [52].

The methodology of the current study is firmly rooted in this philosophy; it is carried out in an iterative manner over a total of three design and development cycles (shown in Figure 4.2). Stakeholders such as stroke patients and clinicians were enlisted during evaluation in all these iterations, with the exception of the third one where patients could not be accessed due to the COVID-19 situation in Denmark.

4.2 Methods Used in Current Study

4.2.1 Terminology Clarification

For the sake of brevity, certain repeating terms will henceforth be shortened and an explanation is provided in Fig. 4.1.

4.2.2 Design and Implementation Philosophy

Musical Stimulus

As reviewed, musical biofeedback studies have used either pre-existing music or real-time synthesized stimuli. Preexisting music is advantageous in that it can be catered to patient preferences and its playback is computationally inexpensive. However, its artifact-free manipulation is complex and there is limited fine system control over its constituent musical elements. Real-time synthesized music conversely affords straightforward control over not only its rhythm, tempo and pitch but also individual instrument tracks and their sequencing, synthesis and effect

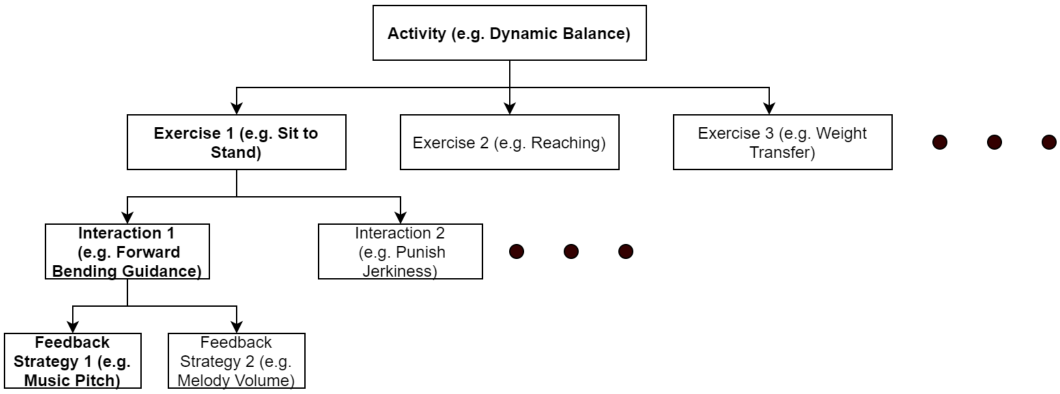

Figure 4.1: A tree diagram depicting the use of terminology to describe the sub-components of each type of therapy. Each activity targets a specific bodily ability, with multiple exercises aimed at rehabilitating various aspects of that ability. Each exercise may be augmented using a number of movement-music interactions, which in turn may employ one of several feedback strategies to convey movement information through sound in a specific way.

parameters. This allows the creation of interesting and powerful interactions. Additional control over the density of the ensemble is desirable in TIMP applications [99], where stimuli must cater to cognitive and attention deficits of patients. Although past research has warned against use of stimuli that are too simple, it is feasible to synthesize rich ensembles in real-time using modern computers.

A caveat of the synthesis approach is the ability of the patient to choose the musical stimulus; the importance of familiarity in inducing emotional responses has already been discussed, particularly for a fragile group such as stroke patients. The system must therefore be able to synthesize music encoded in digital symbolic notation formats (e.g. MIDI) which would allow selected music to be encoded in advance and reproduced when needed. The synthesis system must correspondingly allow encoded music to be reproduced in a selection of music styles to cater to individual genre-based preferences. Provisions for expressive and novel variations in the reproduction must also be made. Although this adds a preparation step and the synthesized version may never optimally match the original music, I argue that with sufficient refinement, the benefits of real-time synthesis from an interaction perspective can outweigh this downside.

Biofeedback System

While compact user architectures with local processing are most convenient, there are several challenges related to processing and software limitations as well as the need for rigorous interface usability testing with a population that may not be comfortable using technology. As the goal of this research is the exploration of *interactions*, it would be ideal not to have to tackle the above challenges at the present stage. An instructor-based architecture with remote processing by a powerful computer eliminates the processing challenge. To be clear, the term *instructor* in this context would normally mean a physiotherapist, but the present goal is only to create an interface to test interactions **in collaboration with** a physiotherapist, who will *not* directly operate the technology at this stage.

With the difficulties experienced using force sensors and foot-switches in our own previous studies [79, 80], the chosen approach here is that of IMU units with multiple axes of measurement (both accelerometer and gyroscope), capable of wireless transmission at a sufficient rate. The inertial data obtained from these sensors can be processed in different ways to obtain an array of parameters related to orientation, movement quality, quantity and timing. For instance, trunk inclination, sway velocity, jerk quotients, foot swing and heel-strike impact can be readily computed. An appropriate IMU product must be chosen, and a robust and safe mounting mechanism must be developed for different parts of the body. The next consideration is whether to use existing digital audio software for system control (e.g. Ableton Live used by [102]) or to develop a stand-alone application from scratch. Using existing software can save development time considerably, as certain key functionalities are already covered such as external controller interfacing, audio synthesis, effect manipulation and mixing. On the other hand, the complex processing of IMU measurements, data transformation prior to feedback mapping and precise design of feedback behaviors can be difficult to achieve through such software. The chosen approach is therefore to build a software application from the ground up. The added advantage of this approach is that many developed functional elements can be reused when porting the system to other platforms (e.g. mobile) in future research.

Musical Feedback and Mapping

Philosophy: The musical feedback in the current application broadly aims to provide the patient with the following in a context-dependent manner:

- Concurrent feedback on position and movement quality.
- Cyclic feedback on movement periodicity.
- Auditory cues for movement (rhythmic or contextual).

The musical feedback would generally classify as *artificial* sonification [82], as there is no strict psychoacoustic correspondence between the sonification and the exercise. The challenge lies in designing feedback that is clearly perceptible and intuitive to the patient in context with the ongoing exercise, both of which are critical to closing the biofeedback loop. A useful starting point comes from Maes et. al. [59], that is the notion of mapping desirable movement behaviors to pleasant auditory states and vice versa. While this is apparently simple, subjectivity in the perception of pleasant auditory states combined with cognitive deficits interfering in this meaning-making process are foreseeable obstacles.

Years of music listening experience means that most individuals have a robust internal schema of how music 'ought' to sound, in terms of consonance, rhythmicity, harmonicity, internal synchronicity and timbre. Universal agreement about the phenomenon of dissonance, for instance, has already been discussed. This internal schema can serve as an *implicit reference* for pleasant auditory states, while degradations applied to any of the above musical dimensions can represent unpleasant states. These degradations will be effected by manipulating synthesis, envelope and effect parameters to achieve specific perceptual results. This is the musical equivalent of the *strategies with reference* defined by Parseihian et. al. [73] and fits well with the philosophy proposed by Maes et. al. [59]. Of paramount importance is that the unpleasant states are distinct enough from pleasant states to be unambiguously perceived and understood by individuals who are elderly and cognitively impaired.

Meaning-Making Through Music: Due to the temporal and spectral complexity of music, musical sonification is not likely to provide the same level of *precision* in data representation as, for instance, pitch and tempo manipulations of simple waves. But what must be stressed is that it is not intended for the sonification to be used in this way. As it is, the auditory system is not as sensitive to small differences as the visual system [68] and it is perhaps futile to attempt to provide pinpoint positional information through musical sonification. It is more expedient to use music to assist the patient in interpreting the **meaning** of their movements, so he/she can take corrective action if necessary. This would fall firmly under Neuhoff's definition of *artistic* sonification [68].

On one hand, this could be seen as reinforcement learning as per Maes et. al. [59] as the patient is directly encouraged in the direction of desirable behaviors. But on the other hand, they have an implicit reference for how desirable behavior should sound, and attaining that sound can involve a reasoning process described by motor learning theory [46]. Borrowing from principles of PSE [99], the meaning-making process is facilitated by intuitive movement-music mappings. Sonification need not necessarily be restricted to either pleasant or unpleasant states. Non-rhythmic movement cues, for instance can be given through 'neutral-sounding' momentary artifacts in the music (with neither positive nor negative intended meaning).

Flexible Mapping Framework: Conveying kinematic 'meaning' to patients through musical dimensions requires that the technology is first able to understand whether a measured quantity constitutes a desired behavior pattern or not. This is further complicated by the fact that different activities and exercises entail different behavior types, and stroke patients have differing abilities and achievable therapeutic goals. This indicates that a direct and rigid mapping between kinematic parameters and musical dimensions is unlikely to suit the needs of the diverse patient population. Moreover, the goal of building and testing interactions requires that all kinematic parameters and musical dimensions are mutually mappable, which is a challenge as they have different ranges and perceptual scaling factors.

The solution to both these problems is a flexible parameter mapping framework, where the data domain is dissociated from the audio domain by data normalization and sonification of measured movement error, as prescribed by Parseihian et. al. [73]. A useful tweak to this system is to allow the system to work in error correction mode as well (relative to a threshold) [46]. Thus, all measured kinematic parameters will be transformed from their own value ranges to a normalized range whose extremes represent desired and undesired behavior respectively. Hence, all auditory feedback strategies can be designed with appropriate mapping functions for compatibility with this system.

Feedback Dimensionality: A final consideration is the feedback dimensionality, meaning how many kinematic parameters can be mapped to auditory dimensions. While multidimensional mappings are possible in music [59], Parseihian et. al. [73] warn about detrimental interactions between auditory streams and dimensions. The choice adopted through the entirety of this project is **1-D** mapping, where a single kinematic parameter is mapped to a parameter in the auditory domain. This is primarily done to reduce cognitive load on patients who may have all manner of attention deficits, perceptual impediments and cognitive difficulty. The system will be built to allow scalability to multiple dimensions, but the interaction foundations will be laid in a 1-D architecture.

4.2.3 Evaluation Procedures

Every iteration undergoes one or more evaluation procedures at its end. The evaluation data are systematically documented in the form of sensor logs, A/V recordings and interview data for analysis.

Participatory Studies with Patients: Groups of stroke patients admitted to Neuroenhed Nord, Region Nordjylland were approached to help evaluate the first and second iterations. The evaluation activities were structured interviews and actual prototype trials respectively, and 6-7 patients took part in each case. The purpose of these was to assess the developed interactions in real-life training scenarios. In general, sub-acute patients were chosen, ethical guidelines were respected, and participants were informed about the activities a week in advance, with written consent obtained from each of them.

Expert Interviews with Clinicians: Expert interviews were conducted with music therapists and physiotherapists at each stage. The purpose of these interviews was to showcase the developed interactions and gauge the main outcome measures in Chapter 3. The interactions were demonstrated either in real life (Iteration 1, 2), or through high quality video recordings (Iteration 3). These interviews formed the primary evaluation procedure in Iteration 3, where five physiotherapists and two music therapists were interviewed remotely. All interviews were recorded with the participants' permission. The transcriptions were coded by a single coder (author) by an inductive approach into a hierarchical coding scheme.

Technical Testing: This was a round of tests conducted on the application (Iteration 3) to assess key technical parameters of the system, such as latency, computational performance and sensor range. Overall system latency was measured during different interactions by frame-wise video and audio analysis. Computational performance was measured as processor time, and logged using Windows benchmarking software. Sensor range was measured in terms of the percentage of received IMU data packets in a short time-frame.

Online Survey - Music Production: A brief 15 minute online survey was prepared and shared with music producers, so as to obtain feedback on the production values of the system-synthesized music (final iteration only). Questions were posed based on stylistic and aesthetic choices, as well as audio-specific aspects such as frequency and dynamic balance. Five synthesized styles of music were chosen and assessed in this manner.

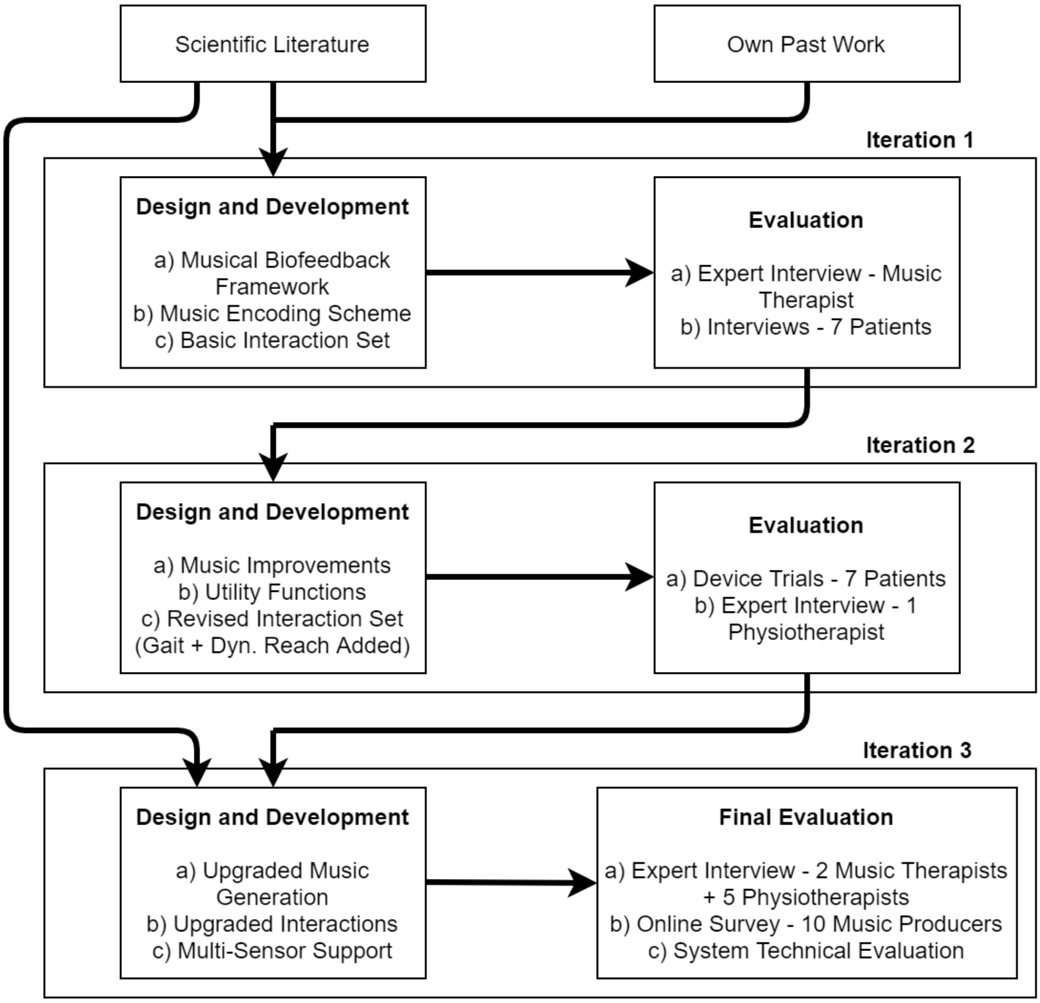

Figure 4.2: A flowchart depicting the design, development and evaluation methods used across all three iterations of the present study.

Chapter 5

Iteration 1

5.1 Aims

The first iteration aims to design and implement a fundamental framework that contains the key components of a musical biofeedback system, including a rudimentary set of balance training interactions. From an implementation standpoint, the aim is to achieve the following functionality in real time:

- Wireless inertial sensing
- IMU signal analysis
- 1-D sonification
- Basic set of auditory feedback strategies
- Music generation framework
- Lightweight music encoding framework
- Tempo/rhythm (groove) manipulation

5.2 Design

5.2.1 Overview - System Structure

The system has a distributed structure [46], wherein the processing and monitoring are performed at a 'remote' location, a laptop in this case, while sensors (and possibly feedback actuators) are mounted on the user. The system design is simply illustrated in Fig. 5.1, depicting a lightweight wireless IMU attached to the trunk of the user. A laptop application acts as the hub for inertial data reception, processing, music synthesis and sonification. It connects to a loudspeaker or pair of headphones which acts as the feedback actuator.

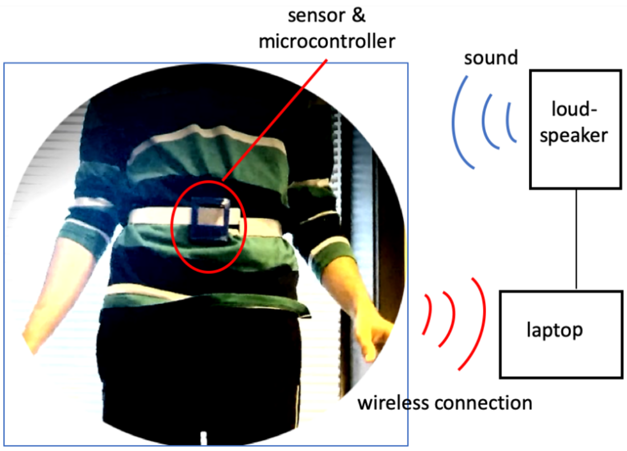

Figure 5.1: A simplified depiction of the system, showing a wireless IMU unit mounted to the trunk of a patient, as well as the information transmission and reception channels.

5.2.2 Sonic Interaction Set - v1

As a starting point, only static and dynamic balance training contexts are considered in this first iteration. From literature, static balance training exercises include maintaining a position of equilibrium under a variety of conditions (sitting, standing, eyes closed, etc.). An example of a dynamic training exercise on the other hand is trunk bending (to pick up or place a distant object for instance). With the current system, an array of sonic interaction concepts with suitable feedback strategies is conceived and short-listed for evaluation. The interaction concepts are as tabulated in Table 5.1 and explained below. Note that the measured movement parameters and feedback strategies are explained later in this chapter.

- **Static Balance - Posture Feedback - Negative:** The objective of static balance exercises is to help maintain upright posture. This feedback concept is provide negative feedback if upright posture is lost, whose intensity is directly proportional to the loss of posture. Posture can be measured in terms of MLat/APos trunk inclination angle. The goal of the patient is thus to keep their trunk inclination angle *below* a target threshold, failing which, negative reinforcement is provided through the music. The music itself sounds completely 'normal' while the patient remains within the target threshold. Appropriate music feedback strategies are the ones which result in unambiguous degradation of music quality.

- **Dynamic Reaching - Posture Feedback - Positive:** The objective of reaching exercises is to attain a certain magnitude of angular trunk bending en route to possibly performing a functional task such as grabbing or touching an object. This bending may either be in the APos or MLat direction, and can similarly

Sr. No	Activity	Interaction	Movement Parameter	Auditory Feedback Strategy
1	Static Balance	Trunk Orientation - Negative Reinforcement of Postural Deviations	Absolute MLat/APos Orientation Angle	- Melody Ring Modulator - Melody Distortion - Percussion Synchronicity
2	Dynamic Balance	Trunk Orientation - Positive Reinforcement of Trunk Bending	Absolute MLat/APos Orientation Angle	- Melody Loudness - Melody Brightness - Melody WahWah
3	Static/Dynamic Balance	Negative Reinforcement - Movement Jerkiness	Scalar Jerk	- Melody Tonic - Pitched Wave Disturbance - Noise Disturbance

Table 5.1: An overview of the activity-wise interaction possibilities in Iteration 1, along with relevant MP and AP choices in each case. Listed AP's are only tested examples, and do not cover all possibilities.

be measured using the corresponding inclination angle. The key difference in this case is that exceeding the target threshold is a *desired* behavior in this training context, and will result in *positive feedback*. Although one possibility is to provide negative feedback under the threshold and normal sounding music above (exactly opposite to the previous concept), it is not fair to punish a patient for sitting upright as the upright phase is a valid part of the reaching exercise. While it does make sense to provide normal sounding music above the target, suitable feedback strategies below the target are those that are more 'neutral'.

- **Static/Dynamic Balance - Jerkiness Feedback - Negative:** The third and final interaction concept involves giving direct negative feedback if the measured jerkiness during any training exceeds the target threshold. Movement quality during training is of great importance, and sudden jerky movements or movement phases can indicate instability. Jerkiness is a quality that is hard for a therapist to give real-time feedback on (or even see) due to its rapid nature, but the IMU can sense it with ease. The goal of the exercising patient is to keep their measured scalar jerk under the custom-defined target threshold, failing which rapid negative feedback will be provided for the duration of the jerky movement and proportional to jerk magnitude. The most appropriate feedback strategies are found to be the ones that by their nature sound most 'jerky', and correspond directly to the underlying movement.

Most initial design choices have been discussed and motivated in Chapter 3; the movement sensing approach of choice is that of wireless IMU-based measurement. A choice of movement measures is offered, ranging from orientation to sway velocity and jerk. The movement-music mapping is one-dimensional and keeping with the normalized target error framework described by Parseihian et. al. [73], with any movement-feedback mapping strategy made possible in the user interface. Desired movement behaviors can be changed in real-time, allowing the biofeedback to function in error-correction mode as well [46]. The remainder of this section focuses on the design philosophy for music representation and generation.

5.2.3 Music Generation

For a coordinated and balanced music synthesis output, the following broad elements are necessary:

- Multi-Instrument Music Structure Data: Information required to synthesize a specific music piece. May be stored within the application or loaded from external files, or a combination.
- Music Sequencing: Temporally organized musical structure information for multiple instruments.
- Music Synthesis: Architecture to realize multitrack musical structure information as synthesized mono or stereo audio tracks, including melodic or percussive instruments.
- Music Mix and Master: The output of the individual synthesizers must be given individual dynamics processing, equalization, gain adjustment and stereo panning, followed by stereo mixdown and master processing, yielding a single stereo output.

Musical Structure Representation The overall evolution of a piece of music is the sum of the individual evolutions of each of its constituent instruments. Of course, the real-life performance of any musical instrument would possess multiple dimensions which would need to be considered for accurate performance capture and reproduction. Symbolic representations such as MIDI are used in modern music production to represent instrumental performances, and can be input to virtual instruments to simulate these performances with an appreciable degree of realism. Due to its simplicity and lightweight structure, MIDI is used as the primary inspiration for the **custom lightweight notation** used in the present iteration.

MIDI captures polyphonic musical performances in the form of a sequence of messages that are ordered as they temporally appear. The timeline is discretized in a tempo-dependent fashion, into 'ticks' that equally divide intervals of musical time with fine granularity (typically 960 ticks per quarter note). Every MIDI message has a timestamp represented in ticks, and the varied message types relate to note onset/offset, articulation and synthesizer control. Speaking specifically of note onset/offset, these are contained in the interval between 'note-on' and their corresponding 'note-off' messages, which specify note number (0-127 range) and note velocity (0-127 range). It is possible to encode information pertaining to both percussion and melody instruments in MIDI.

Novel Simplified Scheme: Simply building a MIDI library would ordinarily be a very time-consuming task as a large amount of multitrack information would need to be encoded for each song, in addition to the need for a software framework for the reading and tempo-accurate playback of multitrack MIDI files. The use of readymade MIDI files from free websites is also less-than-ideal, considering the lack of standardization found among such files in terms of tracks, voicing and detail. Hence, there is a need to design and implement a simpler and faster method for the representation of music pieces. A new simplified scheme is developed, termed as *Compact Music Representation (CMR)*. The specifics of the representation design are given in Appendix B. A *separate* dedicated JUCE application is implemented for the encoding of music pieces in this scheme, discussed in greater detail in the next section. Music files are stored in the CSV format (<3 KB) that can be loaded for real-time playback.

5.3 Implementation

The software requirements for this iteration are fulfilled in the form of two separate standalone Windows applications:

- Music Encoder Application: This facilitates the creation of CMR files. It allows the user to audition musical passages, modify existing files and export CSV files in CMR format for audio reproduction by the biofeedback application.

- Biofeedback Application: This is where sensor data reception, processing and feedback generation are collectively carried out. It has a user interface for music selection, playback and biofeedback control.

Both applications are built in C++ using the JUCE [1] programming environment with Microsoft Visual Studio 2017. JUCE is chosen for its wide selection of classes and libraries that expedite the creation of user interfaces as well as sensor data reception, accurately timed callbacks and advanced data handling. For audio

[1] JUCE. url: https://juce.com/

Figure 5.2: The M5Stack Grey Device (courtesy Adafruit)

synthesis, both applications use FAUST, a domain-specific functional audio programming language with its own libraries for efficient audio synthesis, filtering and routing. FAUST is interfaced with JUCE by using the Faust2Api command in the FAUST compiler, which compiles FAUST .dsp scripts as JUCE-compatible DspFaust classes in C++. The applications are developed as VST plugin projects in JUCE, which also create standalone applications on compilation.

5.3.1 Movement Sensing

The movement sensing functionality was implemented as part of the Prototyping and Fabrication course project in the 9th Semester.

Hardware Description

The sensing hardware must fulfill the following requirements:

- Low-latency wireless transmission
- Ease of mounting and portability
- Sufficient battery life to last the duration of a training session.
- Low cost
- Scalability (no. of sensors)

With these in mind, the chosen sensing device is the M5Stack Grey [2]. With an ESP32 core, it carries a multi-axis IMU sensor (MPU9250+BMM15). The ESP32 has a hybrid Bluetooth/Wi-Fi chip programmable in the Arduino programming environment. The M5Stack device is relatively small (54x54x20mm), and has a 320x240 LCD screen, a small loudspeaker and three programmable buttons. The

[2]ESP32 GREY Development Kit with 9Axis Sensor. URL: https://m5stack.com/ products/grey-development-core.

factory-shipped device is powered by a 150 maH LiPo battery built into its plastic casing. Figure 5.2 depicts the device itself, and other technical details are available on the M5Stack website. The device features a 3D accelerometer, gyroscope and magnetometer, but only ACC and GYR are captured and transmitted for present purposes.

Mounting: For robust sensing, the M5Stack device must be firmly affixed to a specific area of the human body (e.g. lower limb, back). A silicone rubber housing was designed and fabricated for the M5Stack dimensions, including a slit for the insertion of a 25mm wide strap. **Credit for building the silicone housing is due in its entirety to the CREATE laboratory staff at Aalborg University**. Multiple straps were made from velcro to fasten the housing to the body, and this is illustrated in Figure 5.3.

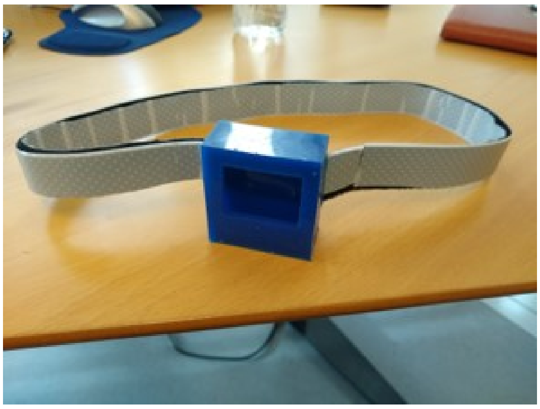

Figure 5.3: Silicone Mount with Velcro Strap.

Microcontroller Programming

The M5Stack device must perform the following tasks:

- Capture and digitize inertial readings, each along 3 axes at an adequate sampling rate.
- Encapsulate the captured values in easy transmissible and destination-readable data packets.
- Transmit the packets wirelessly with minimal latency and data loss.
- Allow device monitoring of wireless connection status, battery level and charge status.

Data Capture The Arduino code is organized into two main functions - the *setup* function which is called once when the ESP32 is switched on, and the *loop* function which is executed in an infinite loop after setup. Time delays can be added into the loop function to make it such that the function is executed at a desired rate. The M5Stack device and IMU are initialized in the *setup* function, and 6 axes of information (Acc X,Y,Z and Gyr X,Y,Z) are read at a rate of 100Hz in the *loop* function by adding a delay of 10ms at the end of every loop execution. This sampling rate balances the tradeoff between latency (max 10ms) and computational/battery load.

Data Packet Creation and Transmission The most convenient transmission protocol for the IMU data is the *OSC (Open Sound Control) protocol*, transmitted using UDP over a WiFi network to a fixed remote port that the biofeedback application listens to (port initialized in the *OSCReceiverUDP_Sensor* class written in JUCE). The *OSCMessage* class for Arduino provides a ready method to package the six ready IMU data values in a single OSC message at every sample interval in a floating-point format.

The WiFi capabilities of the ESP32 provide the ideal mode of data transmission due to its high bandwidth and range. The *Wifi* and *WiFiUDP* classes are included in the Arduino code. The client computer creates a secure WiFi network, whose *SSID, password, IP address and remote port* are initialized in the ESP32 program memory as global variables. Connection success is displayed on the LCD screen, and failure results in continuous retries at 500 ms intervals. Once connection is successful, the OSC data packets assembled at every sampling interval are transmitted over the network as part of the *loop* callback.

Battery Life and Status Monitoring Despite the various power conservation measures applied in the code, it was found that the onboard LiPo battery lasted only approximately 19 minutes from full charge to full discharge. For testing as well as real-life use, this duration was unacceptably low and a solution was needed. The on-board battery (150 mAH) was replaced with a more powerful one. As the M5Stack pin connections are not used, the pin connector board inside the enclosure was removed to create room for a new LiPo battery connected directly to the ESP32 battery socket using a 2-pin JST connector. The new battery is a 1000 mAH LIP553450.0 [3], with dimensions that just about allow it to be squeezed into the enclosure without deformation. Upon testing, the new battery was shown to last well over 90 minutes without fully discharging. The enclosure is closed using insulation tape.

[3] Lithium-Ion Polymer Batteri - 3.7V 1000mAh. url: https://minielektro.dk/ lithium-ion-polymer-batteri-3-7v-1000mah-lip553450.html

The screen, speaker and buttons of the M5stack are all used to help monitor the device status. The screen is kept off by default by configuring its brightness to minimum, thereby preserving battery power and prolonging use time. Update of battery and charging information is only done once every second, as opposed to unnecessarily and wastefully running these parts of the code during every loop. The included *Wire* class allows access to battery status at a resolution of five charge levels (0, 25, 50, 75, 100 percent) in addition to charging status. If a change in battery level is detected (increase while charging or decrease while discharging), the speaker plays a short beep tone as a warning. If the battery is fully charged, the board automatically disables charging mode to minimize fire risks.

Multiple ESP32 sensors for simultaneous sensing can be added by configuring each sensor to transmit to a separate UDP port, and by creating multiple *OSCReceiverUDP_Sensor* objects in the biofeedback application to independently listen to each of these known ports within its own sensing callback. The inbuilt WiFi adapter of the present laptop allows eight simultaneous connections. We now turn back to the JUCE applications.

5.3.2 List of C++ Classes

Both the music encoder and biofeedback applications are structured as a combination of C++ classes, each responsible for a specific component of the net functionality. The following classes are implemented in the first iteration, and the music encoder application only uses a subset of these:

- **GaitSonificationAudioProcessor:** Central class containing synchronous callbacks for music clocking, sensor data receivers, movement analysis, music info mapping and sonification computation/mapping.

- **GaitSonificationAudioProcessorEditor:** Responsible for creating and maintaining the user interface, as well as configuring it to map to the aforementioned central class.

- **DspFaust:** Obtained from Faust2Api, responsible for all audio synthesis and mixing. Contains optimized DSP code, and routes a stereo audio stream to the Windows Audio Engine.

- **Sequencer:** Handles musical timekeeping, accesses stored musical information for each track of the ensemble and returns it to *GaitSonificationAudioProcessor* at regularly timed, tempo-dependent intervals.

- **PercPatternLibrary:** Contains temporally-organized groove and rhythm information for all percussion instruments, as well as triggering patterns for common melody instruments.

- **MusicInfoRead:** Reads music CSV files and dynamically stores the information they contain, for access by *Sequencer*.
- **FaustStrings:** Helper class containing strings and sub-strings to build the mapping addresses of all external *DspFaust* controls, including music info, FX parameters and sonification parameters.
- **AudioParamInfo:** Helper class containing details of all auditory feedback strategies, including their names, mapping function orders, polarities and smoothing filter information, if any.
- **MixerSettings:** Helper class containing track-wise gain, equalizer settings and compressor settings, as well as functions to fetch these values from their respective matrices.
- **OSCReceiverUDP_Sensor:** Responsible for receiving, pre-processing and storing new OSC messages from the IMU over UDP.
- **GaitParamInfo:** Helper class containing metadata of all movement parameters including their names, value ranges, target values, desired behaviors and tolerance percentages.
- **GaitAnalysis:** Accepts IMU ACC/GYR samples, computes and stores user-selected movement parameters.
- **SoniMappingCompute:** Maps the measured movement parameter value to the audio parameter domain in real-time based on the target value/range, desired behavior and mapping function.
- **BiQuad:** Second order biquadratic filter with configurable Butterworth HPF or LPF coefficients. Allows detection of signal maxima, minima and value crossings in either direction.
- **CSVReader:** Helper class to read information from music CSV files and return these values to *MusicInfoRead*.

We begin by examining the implementation of the music encoder application, as its principles are key to understanding the working of the music sequencing functionality in the biofeedback application.

5.3.3 Music Encoder Application

The music encoder provides a relatively quick and simple method to create CMR-compatible CSV files that can be decoded, sequenced and synthesized by the biofeedback application. It reuses a simplified subset of the key functional components of the biofeedback application, namely the following classes:

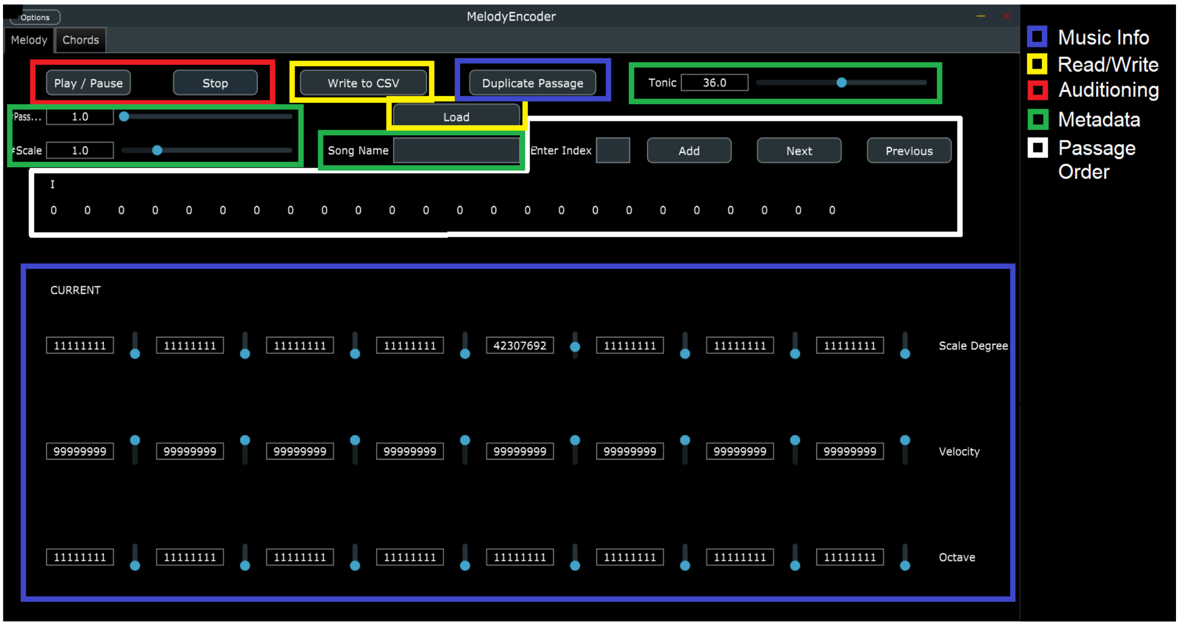

Figure 5.4: Music Encoder Interface.

- *Sequencer, DspFaust* (simplified) - for real-time auditioning
- *MusicInfoRead, CSVReader* - for file reading, writing and memory storage
- *GaitSonificationAudioProcessor* - clocking component for sequencer triggering is reused in *MelodyEncoderAudioProcessor*, the corresponding central class here.

The interface allows the user to accomplish the following:

- **Passage Encoding:** Encode melody and chord information for upto 5 four-bar passages.
- **Read/Write CSV:** Save the encoded data as a CSV file readable by the biofeedback application. Import, edit, preview and save a CSV file of the same type.
- **Auditioning:** Pause and play a real-time preview of the currently encoded data.
- **File Metadata:** Configure music metadata such as name, scale and tonic.
- **Passage Order:** Create a 24 passage order in which the passage numbers sequentially appear.

The layout of the user interface is illustrated in Fig. 5.4. The implementation of each functionality component is described in Appendix B.

5.3.4 Biofeedback Application

Application Structure Overview

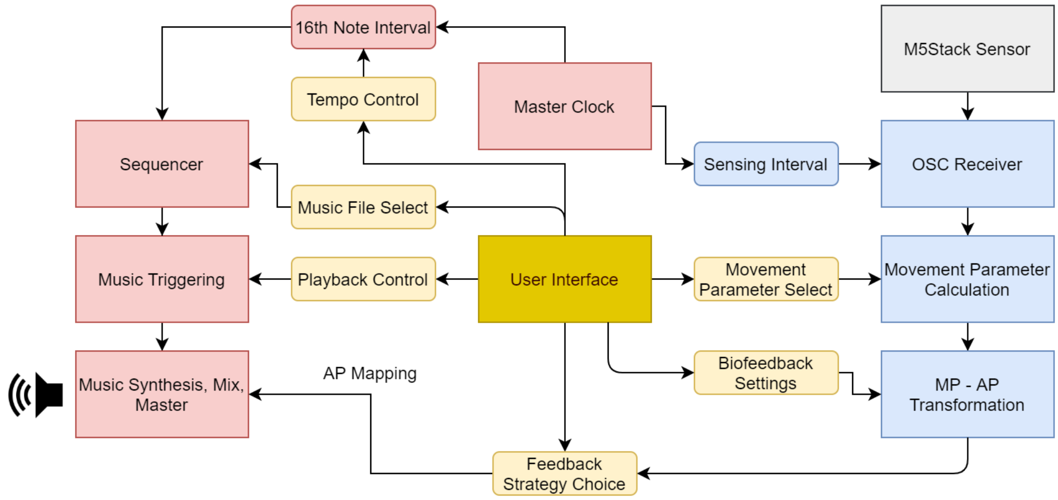

Figure 5.5: High level schematic of the biofeedback application functionality.

The biofeedback application encapsulates and coordinates functional elements ranging from sensor data reception to movement parameter computation and music synthesis in real time. Figure 5.5 depicts the structure of the application at a high level. As shown, there are two functional *tracks*, representing operations related to sensor data processing and music synthesis respectively, although both are controlled by a common timing reference or *Master Clock*. The user interface, on the other hand, branches out to nearly every functional block to provide control over key elements of most functions. It is organized into two *tabs* using the JUCE *TabbedComponent* class, one of which houses music playback-related controls, while the other contains biofeedback parameter controls.

5.3.5 Real-time Callback Structure

The real-time operation of the application is achieved by using periodic callback functions governed a stable timing reference, during which different operations are performed at appropriate intervals. Fig. 5.6 depicts the functional flow of the callback, contextualizing the various C++ classes with the functionalities shown in Fig. 5.5. It is further explained below. The biofeedback application must perform three activities in real-time:

- **Music Synthesis:** The real-time sequencing, audio rendering and playback of a stored music file at the configured tempo in the selected rhythm.

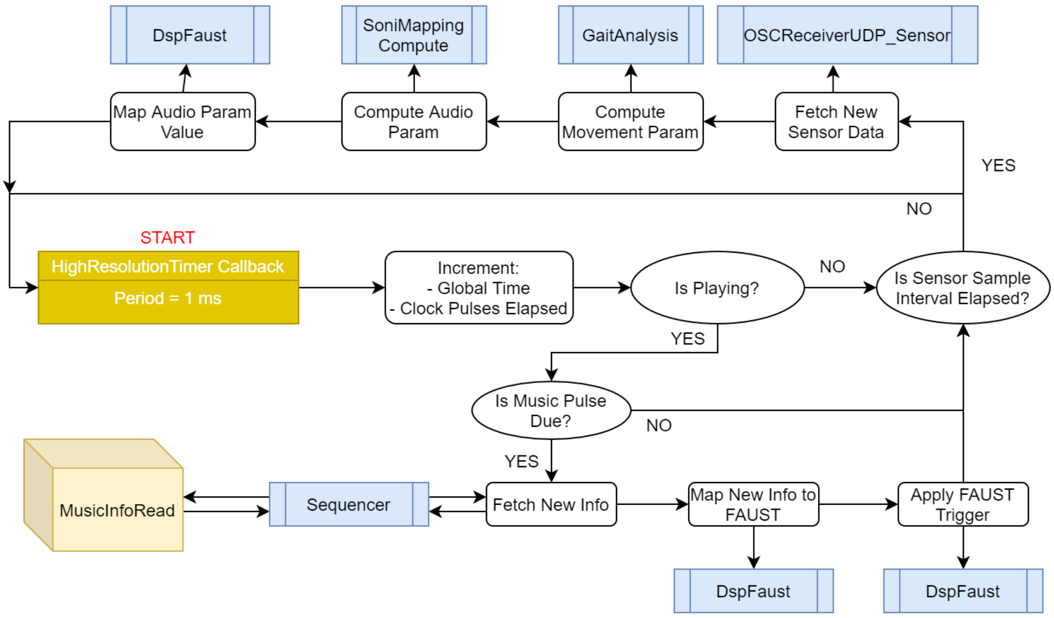

Figure 5.6: High-level schematic of real-time system operation.

- **Movement Measurement and Feedback Computation:** The periodic reception of IMU data, calculation of selected *movement parameter (MP)* and mapping of MP values to *audio parameter (AP)* values to control the selected feedback strategy.
- **UI Update:** Repopulation of UI elements on tab change and update of MP labels and other monitoring elements.

These processes have inherently different rates. IMU measurement occurs at 100 Hz, and consequently MP and AP calculation need not occur faster. On the other hand, music synthesis, processing and playback occur at high sampling rates (48 KHz chosen in this case). As for music information sequencing and mapping, the musical time resolution of the CMR is the straight sixteenth note. For tempi ranging from 60-150 BPM, this implies a sequencing rate between 4 Hz and 10 Hz. UI update need not be very rapid or precisely timed, and rate of 25 Hz is chosen. Given the relative independence of most processes from one another, it is possible to have some level of asynchronous operation as long as thread safety is ensured. Ultimately, the distribution of real-time callbacks is decided based on feasibility and necessary timing precision.

The main audio callback has high performance requirements due to the high sample rate and the demands of multitrack synthesis and processing. *DspFaust* creates its own audio callback which cannot directly be accessed or interrupted by its JUCE parent object (*GaitSonificationAudioProcessor*). Of the remaining low-rate operations, the ones demanding timing precision are music sequencing, sensor reception, MP/AP calculation and mapping. Inaccurate timing would cause noticeable rhythm deviations in music sequencing and triggering. Timing errors in sensor and feedback processing would lead to temporal distortion of the MP measurements and, in turn, the auditory feedback. These operations are therefore grouped together to be handled by a single precise callback. The standard JUCE libraries contain classes for timer callbacks, namely 'Timer' and 'HighResolutionTimer'. Both these classes allow callbacks to a virtual C++ method at a chosen frequency [4] [5], but differ in temporal accuracy. The web reference states that the *Timer* object cannot be expected to be accurate to more than 10-20 ms, which in a worst-case scenario would translate to noticeable rhythmic deviations. On the other hand, the *HighResolutionTimer* object is far more accurate as it uses a dedicated thread. It is, however, computationally more expensive, but given the importance of timing accuracy, it is the tool of choice for music sequencing and sensing. It runs in *GaitSonificationAudioProcessor*.

Even though the sequencing and sensor-related rates are low (4-10 Hz and 100Hz respectively), the floating-point tempo range means that the two rates cannot be expected to be integrally related to each other, which has implications for the choice of callback frequency. This is tackled by configuring the callback to run a much higher frequency than either rate (1000 Hz or 1ms interval chosen), and calling sequencing and sensor-related sub-callbacks at submultiples of the main callback frequency. As the sensor callback frequency is an exact submultiple of 1000 Hz, no timing precision is lost. Although this is not the case for the music sequencing rate, the maximum timing error corresponds to a single callback interval (1 ms) which is practically negligible.

UI update is not as time-sensitive and some amount of timing inaccuracy is tolerable as long as there is no perceptible lag or latency while using the interface. A simple *Timer* object is used to create a 25 Hz callback in the UI class *GaitSonificationAudioProcessorEditor*. While it operates asynchronously, and is not as accurate as the *HighResolutionTimer*, it is computationally light and found to serve its purpose. The frequency of 25 Hz ensures that UI update does not suffer from visible choppiness or sluggishness, and allows for smooth data visualizations in future iterations.

5.3.6 Music Playback Controls

This part of the interface allows the loading of music files and control of playback parameters (see Fig. 5.7). They all influence some element of the clocking, sequencing or audio functionality, such as play/pause, tempo, percussion rhythm (groove)

[4] Timer Class Reference. url: https://docs.juce.com/master/classTimer. html

[5] HighResolutionTimer Class Reference. url: https://docs.juce.com/master/ classHighResolutionTimer.html.

Figure 5.7: Playback controls as they appear in the JUCE UI.

and muting of individual instruments. Each of these is explained in greater detail in Appendix C.

5.3.7 Music Sequencing

Music Clocking

The music clocking mechanism is responsible for:

- Tracking the precise sixteenth note interval with millisecond accuracy using the master clock callback.
- Providing temporal cues for sequencer update and music information fetching.
- Providing a triggering signal to *DspFaust* in a reliable and time-accurate fashion.

The sixteenth note interval at a given tempo may be computed as follows:

$$Interval_{ms} = \frac{(0.25 \cdot Tempo_{BPM})}{60000}$$

Clocking is input to *DspFaust* via a virtual *button* control, whose output is 1 when the button is pressed down, and 0 otherwise. By ‘pressing’ the button at the sixteenth note interval calculated by JUCE, a tempo-governed impulse train is generated as a timing reference within FAUST. The clocking works as follows:

- USER BEGINS PLAYBACK: Music clock triggered at time zero.
- The expected timestamp for the next expected pulse is calculated from the 16th note interval at the set tempo.
- Elapsed time incremented by 1ms within each main callback, and compared to the expected pulse timestamp.
- If Elapsed Time > Expected Pulse Timestamp, then the sequencer is updated, new music information for that musical timestamp is fetched, and the music clock button of *DspFaust* is virtually pressed down.

- If the music clock button has been down for > 70ms, it is then virtually released. The 70ms duration was found to be practically necessary to ensure that the pulse was consistently detected by FAUST.

Thus, the music clock output in *DspFaust* is a rectangular wave that is "HIGH" for 70ms and low for the remainder of the cycle. The maximum possible configurable tempo is 150 BPM which translates to 100ms per sixteenth note. Even in this boundary case, a clock-press time of 70ms is comfortably feasible without risking "HIGH" phases from consecutive clock pulses overlapping with one another and eliminating the required rising edges for envelope triggering.

The real-time sequencing of new musical information to be mapped to *DspFaust* occurs at discrete intervals, triggered periodically by the rising edge of the sixteenth-note clock pulse registered in *GaitSonificationAudioProcessor*. The ultimate goal of the sequencing functionality is to register the temporal cues provided by the clock pulse, organize them into musical time, regularly fetch the new note information for every single track (instrument) in the ensemble and map all this information to *DspFaust* before finally triggering all their envelopes simultaneously with the new information. This set of operations must be performed within a single millisecond (master clock callback duration). The bulk of the process is handled by the *Sequencer* class, which:

- contains numerous counters at the sixteenth note, beat, bar and measure level to keep track of musical time.
- has access to the *MusicInfoRead* object that contains the melody and chord information derived from the loaded CMR file.
- has a member *PercPatternLibrary* object, which contains numerous percussive patterns and variations corresponding to all available rhythm (groove) types and organized at the sixteenth note level.
- has a main function dedicated to fetching different types of information on a track-wise basis.
- has helper functions for the necessary decoding (pre-processing) of melody information.

We now look at the sequence of sub-operations that takes place in the callback when a music clock pulse is due. The first of these is the simultaneous update of musical time counters at all levels. This is best explained visually, and depicted in Fig. 5.8. As is apparent, the input of a new music clock pulse triggers the update of multiple counters responsible for different sequencing functions, mainly:

- within-bar and within-measure sixteenth note counters, which are used as vector indices to fetch percussion and melody note information respectively.

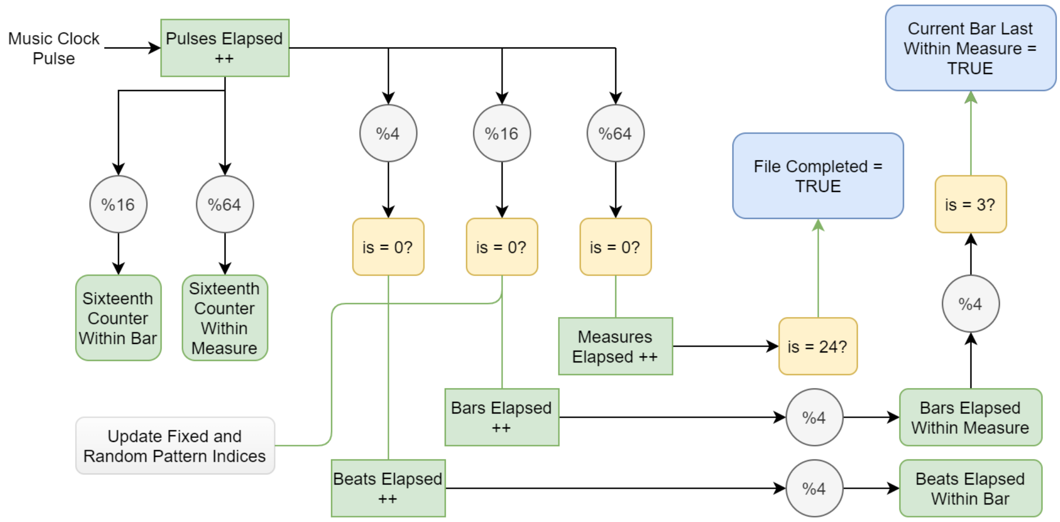

Figure 5.8: Musical counter update flow at every musical clock pulse interval.

- measure, bar and beat counters, which keep track of the progress of a music piece both at a local and global level.
- fixed and random pattern indices, which are primarily used to add a degree of unpredictability to percussive patterns and are updated at the end of every bar.

The next step is to use this counter information to fetch melody and percussion information, but we first discuss how this information is organized and stored for periodic access. Table 5.2 depicts the sources of musical information necessary for all eight instrument tracks. As seen, the information is obtained from either the CMR CSV file or a class called *PercPatternLibrary*, and the sequences may represent either a single bar or four bars. There are also instrument tracks that reuse information from other instruments, for example the chord synths and bassline. As a rule, information that specifically pertains to a piece of music (e.g. melody or chord progressions) comes from the music CSV, while information that is directly dependent on the chosen rhythm or groove is stored in *PercPatternLibrary*. The latter includes the triggering velocity of the bassline and high chord synths for instance, which are additional tracks that derive frequency-related information from the music CSV. We address percussion and melody information separately as follows:

Track	Instrument Name	Information Type	Source	Bars
1	Bass Drum	Velocity	PercPatternLibrary	1
2	Snare Drum	Velocity	PercPatternLibrary	1
3	Hi-Hat	Velocity	PercPatternLibrary	1
4	Main Chord Synth	Velocity	MusicInfoRead – Music CSV	4
		Root Note Degree	MusicInfoRead – Music CSV	4
		Chord Type	MusicInfoRead – Music CSV	4
5	Bassline	Velocity	PercPatternLibrary	1
		Root Note Degree	***Same as Main Chord Synth***	4
6	Main Melody Synth	Velocity	MusicInfoRead – Music CSV	4
		Note Degree	MusicInfoRead – Music CSV	4
		Note Octave	MusicInfoRead – Music CSV	4
7	High Chord Synth	Velocity	PercPatternLibrary	1
		Root Note Degree	***Same as Main Chord Synth***	4
		Chord Type	***Same as Main Chord Synth***	4
8	Crash Cymbal	Velocity	PercPatternLibrary	1

Table 5.2: The various types of musical information, their sources and lengths.

Percussion Information

As alluded to earlier, it is possible for the user to cycle between a choice of rhythm grooves (e.g. Dance, Reggaeton, March, etc.) in real-time. These rhythms are all stored in arrays of velocity information at the sixteenth note level in *PercPatternLibrary*, of which the *Sequencer* object has a member instance. Velocity information for each percussion instrument - bass drum, snare drum, hi hat and crash cymbal, is stored in matrices with ordered rows corresponding to each rhythm. The rhythmic measure is one bar (4 beats or 16 sixteenth notes) in length, and thus four repetitions of a percussive beat occur through the duration of every 4-bar melody passage. This could conceivably lead to a percussion pattern that sounds very static and repetitive, and provisions are made to introduce elements of unpredictability and evolution as a passage progresses.

This is achieved by dividing the temporally organized percussion information into two components - a fixed base component for each instrument, which captures the essence of the rhythm with no variations or flourishes, and a variable component that 1) becomes progressively more 'busy' as a passage proceeds and 2) introduces unpredictability as it is randomly chosen from a set of fixed possibilities. As the snare drum and hi-hat are typically the most expressive percussive elements, the variable component applies only to them. Appropriate variable components are encoded and stored for the different rhythm types, in the form of different 'pools' for Bar 1, 2, 3 and 4 in a melody passage, which are increasingly

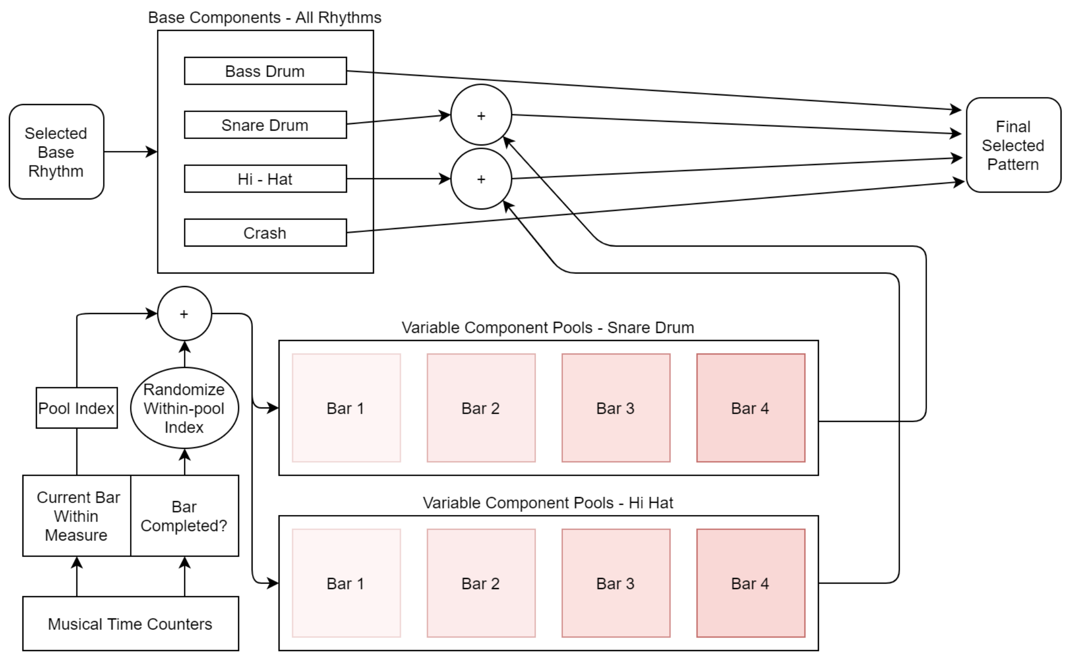

Figure 5.9: Simplified illustration of how the base and variable components of a percussive rhythm are combined to yield an evolving rhythmic pattern.

'busy' musically, and this results in rhythms that build in intensity towards the end of the passage and resolve with a drum fill. The selected variable component indices are randomized at the end of every bar. The essentials of the process are illustrated in Fig. 5.9. As far as the matrices for velocity information go, they are simply arrays containing integers between 0 and 9 representing velocity. For example, the following array represents a quarter note pattern, possibly a dance style bass drum. (each element represents a sixteenth note).

9 0 0 0 **9** 0 0 0 **9** 0 0 0 **9** 0 0 0

A simple drum fill (perhaps played on the snare drum at the end of the melody passage) could be encoded in the Bar 4 snare drum variable component pool as:

9 0 0 0 **9** 0 0 0 **3** 4 5 6 **7** 8 9 9

Melody Information

The eight digit codes from the CSV file are loaded by *MusicInfoRead*, following which the individual digits of each code are separated and stored in 5×64 matrices. These store the different information types for each of the five passages. This information can now be accessed using the within-measure sixteenth counter of *Sequencer*. No further processing is applied at this stage.

Information Fetching and Mapping

After the counters are updated, melody and percussion information is fetched track-wise from the respective sources by using, as the case may be, the within-bar or within-measure sixteenth note counters as array read indices. The information is then mapped, all at once, to the respective music information sliders of *DspFaust*. After this is done, the *Music Clock* FAUST button control is virtually set to 1 as the global envelope triggering cue for individual music synthesizers in FAUST.

5.3.8 Audio Synthesis and Mixing

Overview The generation of a multitrack music ensemble is realized as a FAUST *dsp* script, which is compiled for JUCE using FAUST2Api, which yields the *DspFaust* C++ class. This class handles the synthesis of both the music ensemble (8 independent stereo tracks) and its manipulation by a set of auditory feedback strategies. The audio playback occurs at a fixed sample rate of 48 KHz. Its FAUST UI controls are automatically manipulated by the *GaitSonificationAudioProcessor* class in real-time (e.g. music/AP controls) or at the start of playback (e.g. mixer settings). Thus, many sets of controls are required, and these are divided into FAUST *tabs* that are functionally segregated as follows:

- Track-wise music information (Note degrees, velocities, etc.).
- AP value sliders for each feedback strategy.
- Track-wise equalizer section.
- Track-wise compressor section.
- Master section (volume, track mute, 2 band EQ)

Ensemble Description: Figure 5.10 visually depicts the stereo image, showing the panning locations of elements within the ensemble. Most critical elements are center-panned to ensure compatibility with mono playback systems, although the supporting chord synths are processed to artificially increase their stereo width. Supporting percussion tracks are panned to opposite locations as well, and the addition of reverberation augments the overall sense of space in the mix.

Music Information Preprocessing

The information mapped to *DspFaust* require pre-processing before use, and this is described in the following paragraphs:

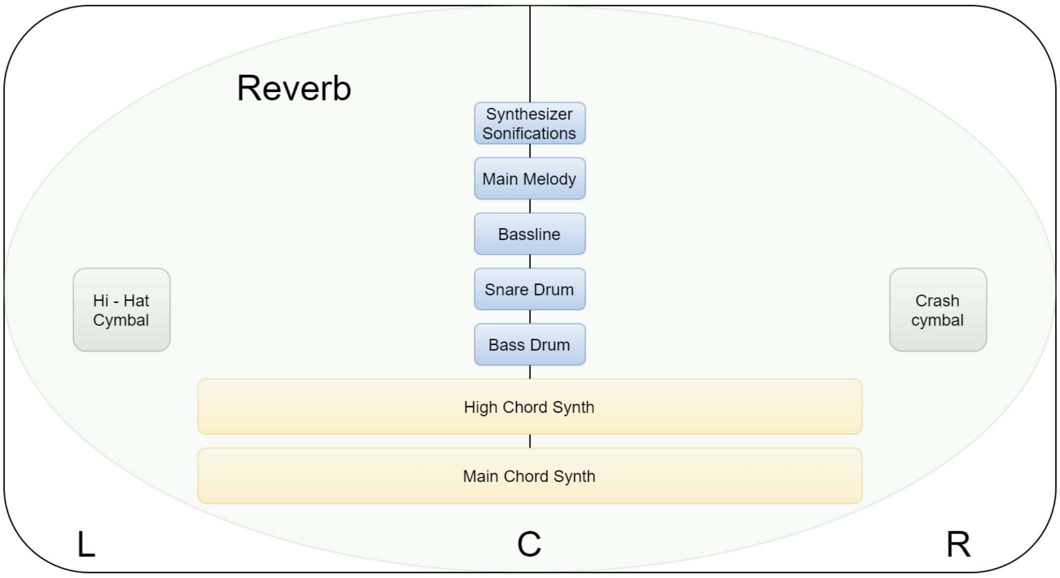

Figure 5.10: A visual representation of the stereo image of the synthesized music and how it is built.

Envelope Triggering: As the CMR file only contains velocity data for each 16th note, triggering instants must be derived and there is no additional information about note duration. ADSR-based amplitude envelopes are typically used by virtual instruments, but the D and S components are not usable here. Therefore, all synthesized instruments use variants of a simple 'AR' type envelopes, which can be linear or exponential depending on the required timbre. FAUST *en.ar* (linear) and *en.are* (exponential) envelopes are used. AR envelopes receive trigger inputs, which work by initiating the attack phase when rising from zero, and the release phase when falling to zero.

Using the velocity signal from the respective instrument velocity sliders would not directly work as envelope triggers as the release would never be triggered if adjacent sixteenth note intervals had non-zero values. However, this is solved by multiplying the velocity signal of each track with the impulsive master clock signal. This yields a velocity-scaled impulse train with no impulse when the velocity signal is zero. This is then used as a 'cooked' envelope trigger for the amplitude envelope of every track.

Velocity-Based Envelope Amplitude: The main function of the velocity signals is to control articulation intensity of the synthesized instruments. In real instruments, the *velocity* with which an instrument is physically played will impact not only the resulting loudness but also the spectrotemporal characteristics of the emerging waveform, due to nonlinear aspects of the instrument's physics. This can be replicated to a large degree in digital instruments as well, but in the current iteration,

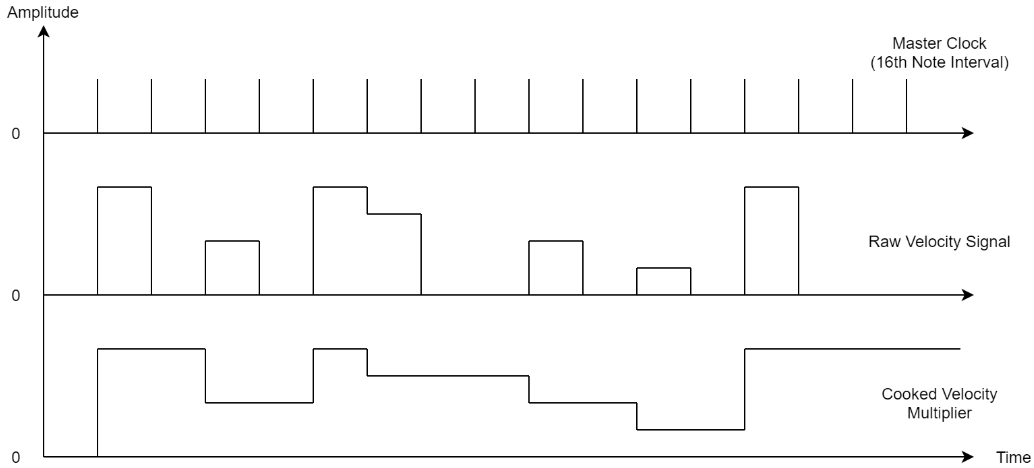

Figure 5.11: Transformation of the raw velocity signal from JUCE to its sampled version which is used as an envelope multiplier.

only loudness is varied in a simple, linear fashion that is proportional to the value of the velocity signal. Although this seems like a simple matter of *multiplying* the velocity signal with the synthesized waveform, that would cause problems when the decay time of the instrument *exceeds* the sixteenth note interval. For example, if an 8th note velocity pattern is encoded '90709050' then the release phase of the instrument will abruptly jump to zero after a sixteenth note interval, creating undesirable clicks and unnatural-sounding instrument release phases.

To fix this problem, the velocity signal is preprocessed by passing it through the FAUST sample-and-hold function *ba.SAndH*, which samples the velocity signal only at instants where the master clock pulse and velocity signal are simultaneously nonzero. The output is normalized to the 0-1 range and used as a envelope amplitude multiplier for its respective instrument. Fig. 5.11 illustrates the velocity pre-processing.

Melody Instruments - Frequency Calculation: Separate pitch signals corresponding to each monophonic or polyphonic voice must be generated from the mapped CMR information. In the case of the former (main melody synth, bassline), the frequency information available is the tonic, scale code, the scale degree of the current note and the octave offset relative to the tonic. In the case of polyphonic instruments (main chord synth, high chord synth), the the scale degree refers to the root note of the chord while the octave offset is replaced by a code representing the chord type. In both cases, the first task is to convert these representations into a format usable by a synthesizer oscillator - frequency values in Hertz (f0). Hence, there must be a f0 'signal' corresponding to every voice of every pitched instrument serving as a control signal for its respective oscillator. For both chord tracks, the total number of voices is kept constant at 4 to be able to reproduce chordal harmony, and all voices are triggered simultaneously by the respective track trigger signal. Thus, there is no possibility of arpeggios in this iteration. **The detailed procedure of how f0 values in Hz are computed from CMR information is provided in Appendix B**.

Uncorrelated Noise Sources for Individual Instruments

All percussive sounds in this iteration are synthesized using techniques common to electronic music synthesis, which commonly include the use of filtered and envelope-controlled noise. FAUST has a standard library function *no.noise* that can be used for this purpose, but the noise it generates is only *pseudo-random*, meaning that its signal characteristics approach that of white Gaussian noise. But it is actually generated in a deterministic manner, and multiple instances of the function running simultaneously all generate identical signals. This can create problems if multiple instruments are to be synthesized using them, as they are triggered at different times and filtered in very different ways in order to achieve their respective desired timbres. As all signals are correlated, different filters on each instrument will impose different phase delay characteristics on each instrument, causing unpredictable phase cancellation when the instruments coincide temporally. To prevent this, the instruments must use separate noise sources that are inherently uncorrelated to one another. FAUST also provides a function to serve this purpose, namely *no.multinoise(n)* which provides *n* uncorrelated noise signals in parallel outputs. These are all routed in parallel to the synthesis algorithms of each percussion instrument, thus mitigating the problem of phase cancellation. **Details of individual instrument synthesis algorithms are provided in Appendix C.**

Tempo-Dependent Synthesis Parameters

Although the sequencing rate is handled outside *DspFaust*, it is also necessary for certain synthesis parameters to scale appropriately with the music tempo. Examples include the time-constants of delay-based effects, which would not maintain synchronization without adjusting to tempo changes. However, even synthesis and envelope parameters of instruments would benefit from these adjustments. The possible range of tempos is wide (60 - 150 BPM), and having constant envelopes across this range would lead to single instrument envelopes overlapping in time at the upper tempo extreme and exhibiting silent gaps at the lower extreme, neither of which is desirable. Chord tracks that exhibit multiple notes playing simultaneously would sound more pleasant at slower tempos if the onsets of their notes were not perfectly synchronized, perhaps as a natural 'strum'. Certain sonification strategies are also tempo-dependent, but these are discussed separately. The music-specific dependencies are as follows:

- **Percussion Release Times:** The envelope release constants of the bass drum, snare drum and hi-hat cymbal are multiplied by a constant factor that increases their release time as the tempo decreases, slowing their decay and giving the sounds more perceptible 'body' to suit the slower tempos. The release factor is 1 at 120 BPM and higher, and below 120 BPM it is:

$$mult_{rel} = 1 + 1.5 \cdot (120 - tempo)/40$$

  This is directly tied to the input tempo from JUCE, and updates the synthesis algorithms in real time.

- **Reverb Decay Time:** A single reverberation effect is used to create a sense of space, and works on multiple instruments. Its decay time is a function of tempo, and is discussed further in the next subsection.

- **High Chord Note Delay - Strum Effect:** The high chord synthesizer sends the stereo signals of each constituent note to separate short delays prior to mixing them, which delay the notes by different amounts at tempos below 120 BPM. The delays proportionally increase as the tempo becomes less, creating a transition from a *percussive* chord sound to a more gentle, strummed sound. The delay times in seconds for each note of the chord (below 120 BPM) are as follows:

$$\begin{aligned} d_{n1} &= 0 \\ d_{n2} &= 0.015 \cdot (120 - tempo)/60 \\ d_{n3} &= 0.030 \cdot (120 - tempo)/60 \\ d_{n4} &= 0.045 \cdot (120 - tempo)/60 \end{aligned}$$

- **Bassline Release Time:** The bassline goes from a sharp and 'snappy' envelope at high tempos to a slower and smoother envelope at lower tempos, which, aside from being more suitable, fills out the upper bass register and adds thickness to the overall mix. The release time in seconds below 120 BPM is calculated as:

$$rel_{bassline} = 0.6 + (120 - tempo) \cdot 0.06$$

- **Echo Time:** The main melody and high chord synths are processed using dotted echo effects, whose echo time is a function of the tempo, calculated below. Note that the beat interval in samples is calculated from the tempo using the standard FAUST function *ba.tempo*:

$$
\begin{aligned}
beatTime_{samples} &= ba.tempo(tempo) \\
melDelay_{samples}(L) &= 0.75 \cdot beatTime_{samples} \\
melDelay_{samples}(R) &= 1.5 \cdot beatTime_{samples} \\
hiChordDelay_{samples} &= 0.75 \cdot beatTime_{samples}
\end{aligned}
$$

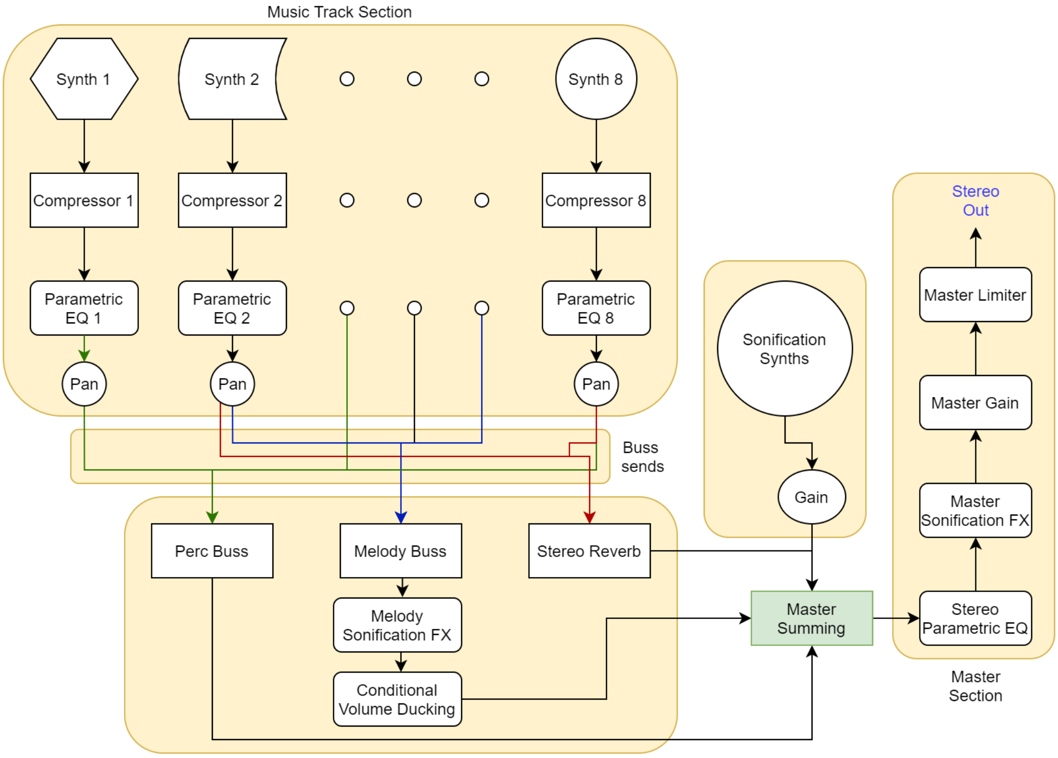

Figure 5.12: High level audio mixing schematic.

Mix and Master: The synthesized individual audio tracks must be balanced in context with one another, panned to appropriate locations in the stereo field and mixed down to a single stereo pair at an appropriate signal level for final dynamics processing prior to output. The entire mixing process is inspired by conventional practice in music mixing, wherein most audio tracks receive some form of dynamic range compression and equalization in that order, prior to master summing. Fig. 5.12 is a broad depiction of the signal routing and processing chain. As shown, each of the eight synthesized tracks undergo individual channel processing in the form of:

- **Channel Compressor:** This is a simple dynamic range compressor identical to [80], implemented as a variable gain amplifier with modifiable threshold

Figure 5.13: Channel compressor tab UI as seen in the FAUST web editor.

and ratio, controlled by a FAUST envelope follower with configurable time constants (attack, release). Individual compressors for each track are created in parallel using FAUST *hgroup, vgroup and tgroup* primitives, and the web editor UI rendering can be seen in Figure 5.13. These FAUST controls are not included in the JUCE UI because the synthesis methods and signal levels are known, so the parameter values can be hard-coded at compile time. Trackwise settings are stored in matrices in the C++ class *mixerSettings* and mapped to the respective FAUST control addresses from *FaustStrings* when playback is initiated.

Figure 5.14: Parametric EQ tab UI as seen in the FAUST web editor.

- **Parametric EQ:** Each track is processed using a fully parametric equalizer, to modify the synthesized sounds in the frequency domain and eliminate undesirable resonances, low-frequency rumble and clashes with other instruments. The custom-written parametric EQ function comprises a highpass filter, a lowpass filter and two constant-Q peaking filters, all of which are available as part of the standard FAUST libraries. All filters are second order IIR, and provide control over cutoff/center frequency, Q and gain (for peaking filters). Thus, each EQ has a combined total of 10 configurable parameters, as illustrated in Fig. 5.14. The controls are similarly organized in FAUST using same grouping primitives, and track-wise values are stored in *mixerSettings* and initialized on playback in similar fashion to the compressor settings.

Buss Processing: The eight tracks are first pre-summed into three groups, namely the percussion tracks, melody tracks and a global reverb send.

- Percussion Buss: Consists of the *bass drum, snare drum, hi-hat and crash cymbal* tracks. This group proceeds to master summing with no further processing.
- Melody Buss: Consists of the main chord synth, bassline, main melody and high chord synth. This buss is processed by the *melody FX-based sonification* sequence, followed by conditional volume ducking by 24 dB that occurs if selected synthesizer-based sonification strategies are active.
- Reverb Buss: Four tracks (snare drum, hi-hat, main melody and high chord synth) are separately summed and sent to a stereo reverb effect from the standard FAUST libraries, called re.zita_rev1stereo. This is a feedback delay network-based reverberator function with several arguments, that are set as follows:
  - Reverb Pre-Delay: 10 ms
  - DC/Midrange Crossover Frequency: 100 Hz
  - Frequency above which $RT60_{mid}$ is halved: 2000 Hz
  - RT60 @0 Hz: 1 sec
  - RT60 @midrange: This parameter is a tempo-dependent parameter which increases as the tempo becomes slower as per the following equation:

    $$RT60_{mid}(sec) = 0.3 + (150 - tempo(BPM)) \cdot 0.06$$

    This creates a greater sense of space at slow tempos, which is commonly seen in popular music.

  The reverb output is bandlimited from 200 Hz - 10000 Hz by a second order Butterworth bandpass filter function from the standard FAUST libraries.

The three busses are combined into a single master stereo pair to which a 2-band parametric EQ can be applied (neutral by default) followed by master gain and a non-linked stereo-limiter with 10:1 ratio, 1ms attack and 50ms release. Fully mixed and master song examples from the system may be found in Media Links at 1.1.9 and 1.1.10.

5.3.9 Movement Parameter Computation

The raw IMU signals (3 Acc + 3 Gyr) are smoothed after reception using 2nd order Butterworth lowpass filters at 5 Hz. They are then processed within the *GaitAnalysis* class to yield a variety of MP's, of which one can be selected for real-time sonification. In the sensor callback of the main class, the pre-processed IMU values are passed to the *compute* function in *GaitAnalysis*, which relays them to the required MP computation function. In this iteration, the following MP's are computed for a trunk-mounted IMU:

Mediolateral, Anteroposterior Orientation Angles from Vertical

The angular displacement of the upper body from the vertical in the mediolateral axis (*MLat* - roll angle) or anteroposterior axis (*APos* - pitch angle) is calculated from the raw Acc and Gyr signals using a complementary filter. This filter provides a means to combine accelerometer and gyroscope data to compute accurate orientation angles without the conceptual and computational complexity of a Kalman filter [103]. It was implemented with the help of two online tutorials [81] [1], and is explained as follows. Acc and Gyr readings can both be used to determine the orientation of an object, and they do it differently. A gyroscope does this by integrating angular velocity over time, while an accelerometer readings are used to determine the position of the gravity vector [81]. While this holds theoretically, neither method is good enough by itself to provide accurate enough orientation estimates.

As an accelerometer measures all active forces on the IMU, it will capture more than just the gravity vector, and its readings will be affected by device movement and vibration/mechanical noise [1]. Thus, the accelerometer readings are only reliable in the *long term* [81]. A gyroscope, on the other hand, is not as susceptible to external forces but the velocity integration over time causes the measurement to drift. Hence, its measurements are only reliable in the *short term* [81]. The simplest way to describe a complementary filter is as a 'best-of-both-worlds', as it relies on the gyroscope in the short term and the accelerometer in the long term. It is implemented based on the procedure detailed in [1], which will briefly be described here. First, the triaxial accelerometer readings are stored in a vector:

$$R_{acc} = [R_{accX}, R_{accY}, R_{accZ}]$$

As the length of the true gravity vector is 1 (times g), R_{acc} is normalized by first calculating its magnitude:

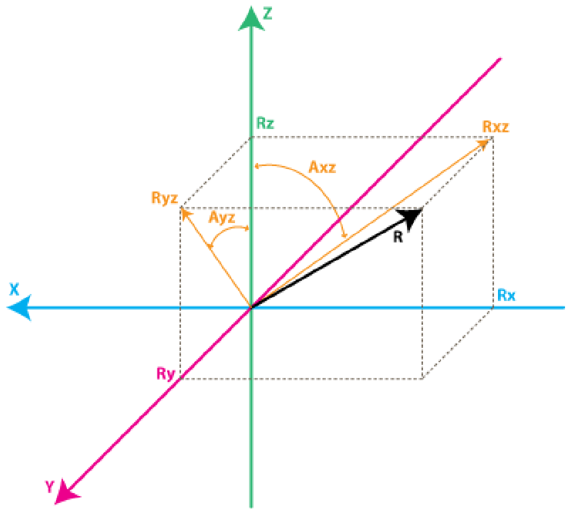

Figure 5.15: Vector R in the 3D plane, and its corresponding angle. Figure courtesy of [1]

$$|R_{acc}| = \sqrt{(R_{AccX})^2 + (R_{AccY})^2 + (R_{AccZ})^2}$$

R_{acc} is then divided by its magnitude to yield its normalized version.

$$R_{acc}(normalized) = \left[\frac{R_{accX}}{|R_{acc}|}, \frac{R_{accY}}{|R_{acc}|}, \frac{R_{accZ}}{|R_{acc}|}\right]$$

A new acceleration estimation vector R_{est} is introduced, which is the main quantity to be computed by the filter:

$$R_{est} = [R_{estX}, R_{estY}, R_{estZ}]$$

R_{est} is initialized at time zero with the values of R_{acc}:

$$R_{est}[0] = R_{acc}[0]$$

R_{acc} receives new measurements at intervals corresponding to the sampling interval t (10 ms), and new estimates $R_{est}[0]$, $R_{est}[1]$, etc. are issued accordingly. At an arbitrary time n, the filter has $R_{est}[n-1]$ (the previous estimate) and $R_{acc}[n]$. As shown in Figure 5.15, rotation about the Y axis is represented by ∠ Axz and by knowing $R_{est}[n-1]$, ∠ Axz[n-1] can be calculated as:

$$\angle Axz[n-1] = arctan2(R_{estX}[n-1], R_{estZ}[n-1])$$

arctan2 is a function that takes two arguments instead of one, and returns an angle in the range $(-\pi, \pi)$ instead of $(-\pi/2, \pi/2)$. The Y reading of the gyroscope measures the rate of change of ∠Axz ($Rate_{\angle Axz}$), and the present ∠Axz and ∠Ayz can be calculated as:

$$\angle Axz[n] = \angle Axz[n] + Rate_{\angle Axz}[n] \cdot t$$
$$\angle Ayz[n] = \angle Ayz[n] + Rate_{\angle Ayz}[n] \cdot t$$

These two angles correspond to the main measured quantities, namely MLat and APos trunk angular displacements respectively. Their absolute values are stored, as direction is not considered in later steps. For precision, $Rate_{\angle Axz}$ is averaged across two samples as (same for $Rate_{\angle Ayz}$):

$$RateAvg_{\angle Axz}[n] = 0.5 \cdot (Rate_{\angle Axz}[n] + Rate_{\angle Axz}[n-1])$$

The calculation of R_{est} is now discussed. The vector R_{gyr} is an intermediate vector used to update R_{est} based on gyroscope readings.

$$R_{gyr} = [R_{gyrX}, R_{gyrY}, R_{gyrZ}]$$

Its rectangular components are derived in [1] as:

$$R_{gyrX} = \frac{1}{\sqrt{1 + cot^2(\angle Axz[n]]) \cdot sec^2(\angle Ayz[n]]))}}$$
$$R_{gyrY} = \frac{sin(\angle Ayz[n]]))}{\sqrt{1 + cos^2(\angle Ayz[n]]) \cdot tan^2(\angle Axz[n]]))}}$$
$$R_{gyrZ} = Sign(R_{estZ}[n-1]) \cdot \sqrt{(1 - (R_{gyrX})^2 - (R_{gyrY})^2)}$$

where Sign is 1 if $R_{estZ}[n-1]$ is positive and -1 if negative.
$R_{est}[n]$ is then calculated as:

$$R_{est}[n] = \frac{R_{acc} + R_{gyr} \cdot w_{gyr}}{1 + w_{gyr}}$$

where w_{gyr} is the weight given to the gyroscope reading in the estimation. The value of w_{gyr} is set at 5. The calculated R_{est} vector serves as the basis for angle calculation during the *next* sample period ($R_{est}[n]$ is the same as $R_{est}[(n+1)-1]$). To counteract gyroscope drift, R_{est} is reinitialized to R_{acc} at an empirically determined interval of every 10 seconds.

RMS MLat, APos Acceleration:

Acceleration of the sensor in the MLat or APos direction results from trunk movement. The corresponding axis readings are first high-pass filtered to remove any DC component of the gravity vector that may be present in these signals. Two cascaded second order Butterworth high-pass filters with cutoff frequency 0.2 Hz are used. The RMS value of each filtered signal is then calculated over a 50ms window (5 samples) and stored.

Scalar Jerk:

Jerk represents the rate of change of movement acceleration, and describes movement smoothness [49]. Similar to RMS acceleration, the acceleration readings from all three axes are high-pass filtered using the same cascaded Butterworth filters, and differentiated to obtain their individual jerk values:

$$Jerk_x[n] = Acc_x[n] - Acc_x[n-1]$$
$$Jerk_y[n] = Acc_y[n] - Acc_y[n-1]$$
$$Jerk_z[n] = Acc_z[n] - Acc_z[n-1]$$

The scalar jerk value is simply obtained as:

$$Jerk_{scalar}[n] = \sqrt{(Jerk_x[n])^2 + (Jerk_y[n])^2 + (Jerk_z[n])^2}$$

MLat, APos Angular Sway Velocity:

These are simply the pre-processed gyroscope readings about the Z and X axes respectively.

5.3.10 Audio Parameter Calculation and Mapping

For sonification purposes, the selected MP is normalized to a fixed range as conceptualized by Parseihian et. al. [73], depending on configured system behavior. The AP calculation is carried out by a function in the *SoniMappingCompute* class, which takes multiple arguments related to the configured feedback settings and returns the AP value for mapping to *DspFaust*. TIt is always in the range of 0 - 1 irrespective of the MP range or feedback settings, and an increase from 0 - 1 always implies an increase in auditory feedback intensity. *SoniMappingCompute* is versatile in that it supports different user behaviors in terms of MP range, polarity and function scaling.

AP Calculation Input Arguments

- **Desired Behavior:** Defines whether the MP value is to be maintained by the patient *below* a certain value, *equal to* the value or *greater than* the value. Default behaviors are stored in an array in the *gaitParamInfo* class for each MP, but are modifiable in real time in the user interface.

- **Target Value:** This is the pivotal value that precisely determines the system response; in case of *less than* or *greater than* behaviors, it is the threshold that the patient must not cross, and in case of *equal to* behavior it is the MP value the patient must match. The target always lies between the minimum and maximum MP bounds, and default values are stored in an array in *gaitParamInfo*, although it can be modified in real-time in the user interface.

- **MP Range:** This is the difference between the maximum and minimum possible MP values, default values of which are stored in *gaitParamInfo* for each MP and unmodifiable.
- **Tolerance Bandwidth:** This is the percentage of the MP range that the user is allowed to deviate about the target in either direction without triggering auditory feedback. It applies more to the *equal to* behavior than the others. Default tolerance bandwidths are stored in *audioParamInfo* for each feedback strategy, and are unmodifiable by the user. They are set to zero by default, but can be modified if a use case strategy arises for them.
- **Mapping Function Order:** This is a constant (integer or decimal) that determines the MP-AP mapping function shape. The use of a decimal number order allows a wide range of shapes, which is important as different auditory dimensions exhibit different perceptual scaling. Default function orders are defined in *audioParamInfo*, and are unmodifiable by the user.
- **Number of Quantization Levels:** This integer number represents the number of discrete levels the computed AP will be quantized to, in case a particular feedback strategy benefits from quantization. If it is not desired, this argument is set to zero. It is stored in *audioParamInfo* for each AP, and cannot be modified by the user.
- **AP Smoothing Filter Cutoff Frequency:** The computed AP function can be smoothed in time prior to mapping if necessary using a 2nd order IIR Butterworth lowpass filter. The cutoff frequency for each feedback strategy is stored in *audioParamInfo* and cannot be modified in real-time. If filtering is not desired, it is set to 100.
- **Present MP Value:** The final argument is the freshly computed MP value that is to be transformed to an AP value. It is stored after computation in *GaitAnalysis*, accessed and passed to *SoniMappingCompute* for AP computation.

AP Computation

With the provided information, it is now possible to compute the AP value. This is done differently for each desired behavior, although there are some common steps. The manner in which intermediate MP values are mapped depends on the mapping function order, a simple example of which is a linear function as depicted in Fig. 5.16. As shown, the AP value is zero when the MP is at the target value or conforms to the desired behavior, and increases towards 1 in its violation. As the target can be modified in real-time, the shape of the mapping plot must also adapt on the fly. The computation comprises the following steps:

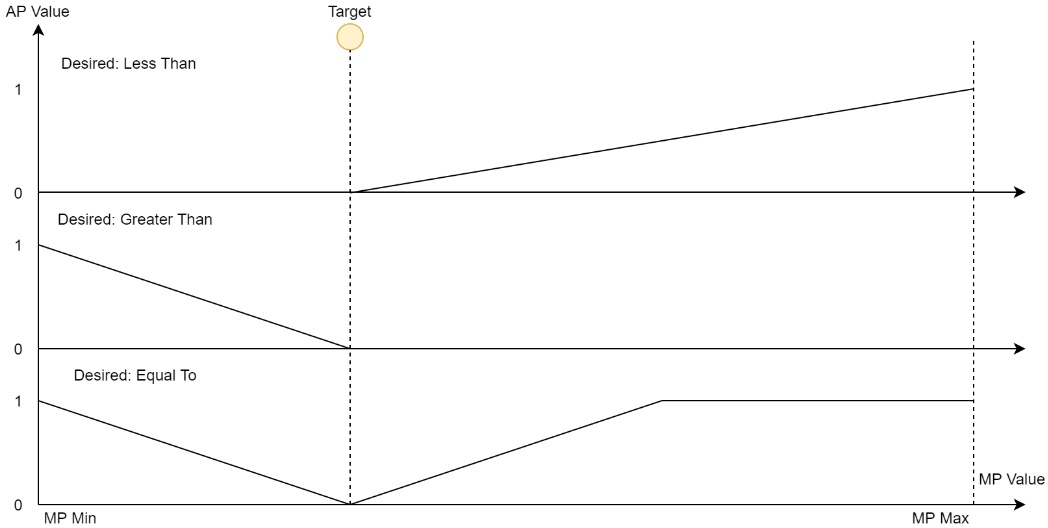

Figure 5.16: The MP-AP mapping curves for an arbitrary target value with the different desired behaviors and a linear mapping function example.

- **Calculate MP sub-range SR_{MP} from target (T) and full range (R). This is the size of the MP interval over which the AP varies between 0 and 1, and depends on desired behavior:**
  - $SR_{MP}(lessThan) = R - T$
  - $SR_{MP}(greaterThan) = T$
  - $SR_{MP}(equalTo) = T$
- Calculate absolute normalized difference between present MP value $MP_{present}$ and T as:

$$Error = \frac{abs(MP_{present} - T)}{SR_{MP}}$$

- If desired behavior is *less-than* and $MP_{present}$ < T, or desired behavior is *greater-than* and $MP_{present}$ > T, then set *Error* to zero.
- Apply mapping function *order* to get preliminary AP value as:

$$AP_{pre} = (Error)^{order}$$

  As AP_{pre} generally ranges between 0 and 1, applying an order greater than 1 will lead to more gradual AP transitions close to the target value and steeper at the extremities, while orders less than 1 will have the opposite effect.

- AP_{pre} can then be smoothed using a 2nd order IIR Butterworth lowpass filter at a specific cutoff frequency for each feedback strategy. This step is ignored if the cutoff frequency argument value is 100.
- The output is finally quantized to the desired number of levels (no quantization if the number of levels is zero), bounded between 0 and 1 as a failsafe and returned to *GaitSonificationAudioProcessor*.
- The parameter address for the currently active AP is fetched from the helper class *FaustStrings* and mapped to *DspFaust* using the *setParamValue* function.

5.3.11 Movement Sonification Strategies for Real-time Movement Feedback

With the real-time music synthesis approach used here, there are three principal ways of manipulating the final audio output for the provision of real-time movement feedback:

- **Music Synthesis Parameter Sonification:** Feedback provided by manipulating the synthesis parameters of the musical instruments, which may include temporal or frequency characteristics of the generated waveforms.
- **Audio Effect Parameter Sonification:** Feedback provided by varying the intensity/spectro-temporal parameters of audio processors applied to individual synthesized music tracks or sub-mixes.
- **Additional Synthesizer Sonification:** Feedback is provided by synthesizing and mixing new audio entities to the music output when certain AP conditions are met, and manipulating them in accordance with the mapped AP. These synthesizers are added to the remainder of the music busses at master summing time.

To recap, each feedback strategy is JUCE-controlled by a separate AP in the normalized range 0-1. Feedback intensity for all strategies is maximum when the AP = 1. For the purpose of this discussion, we represent the corresponding AP value of the strategy with the variable x. All designs are intended for use as concurrent feedback channels, and numerical factors are set experimentally for optimal perceptual scaling. The strategies under each category are described in detail in this section. *Note that these strategies were initially developed and tested as part of the Research in Sound and Music Computing miniproject in Semester 9.*

Music Synthesis Parameter Sonification

- **Melody Tonic:** (demo: 1.2.1 in Media Links) x is mapped as a multiplicative factor to all f0 calculations of the melody instruments, such that their pitch increases in direct proportion to x.

$$f0_{sonified} = f0_{original} \cdot (1 + 5x)$$

- **Melody Envelope Release:** (demo: 1.2.2 in Media Links) If x exceeds 0.01, the individual melody instrument waveforms are multiplied by an additional FAUST AR-type envelope triggered at the same instant as the instrument itself. The attack of this envelope is fixed at 0.001 sec, and the release of the envelope is inversely proportional to x. This has the effect of making the melody instruments sound more impulsive or 'staccato' as x increases.

$$Rel_{sec} = 3 - 2.9 \cdot x$$

Audio Effect Parameter Sonification

- **Melody Brightness:** (demo: 1.2.3 in Media Links) x is mapped to the cut-off frequency of a resonant low pass filter on the master mix, such that the cutoff frequency is maximum when $x = 0$ and reduces with increasing x. The filter has a variable Q-factor which decreases as x increases to prevent lower midrange frequencies in the music from becoming increasingly loud and unpleasant as they overlap with the filter resonance.

$$fc = 20000 - x \cdot (19800)$$
$$Q = 4.0 - 3.3 \cdot x^2$$

- **Melody Loudness:** x is mapped to the gain control of the melody instrument submix, such that increasing x results in the melody elements becoming softer, leaving percussion elements intact. The mapping is logarithmic.

$$melGain_{dB} = -80 \cdot x$$

- **Instrument Synchronicity:** (demo: 1.2.4 in Media Links) All instruments are processed immediately after synthesis through separate delay buffers, where they are subjected to a delay time d that is directly proportional to x. The delay time scaling factors for every instrument are defined such that they share no common factors, leading to the instruments becoming more and more desynchronized as x increases. To prevent pitch modulations resulting from continuously varying delay times, x is sampled and held every 0.5 second to discrete the evolution of the distance variable. The delay in samples for each track is calculated as follows:

$$d(trackIdx) = d_{Max}(trackIdx) \cdot x_{sampled}$$

- **Melody Ring Modulation:** (demo: 1.2.5 in Media Links) This strategy linearly maps x to the mix ratio control of a ring modulator effect patched as a parallel send from the melody submix. The modulation frequency is configured to the note six semitones above the tonic, the most dissonant interval in the scale. This yields the most dissonant sum and difference frequencies in the modulation product. Increasing x increases the ring modulation mix, which is zero at x = 0 and 100% at x = 1. The output is calculated as follows:

$$Prod_{RM} = melBuss_{pre} \cdot sin(2\pi f_{mod} t)$$
$$melBuss_{final} = melBuss_{pre} \cdot (1 - x) + Prod_{RM} \cdot x$$

- **Melody Distortion:** (demo: 1.2.6 in Media Links) x is linearly mapped to the mix control of a FAUST cubic distortion effect patched as a parallel send from the melody submix, whose output is a distorted product of the melody signal. Increasing x results in the melody instruments becoming more distorted, while they are undistorted at x = 0.

- **Melody Phaser:** x is mapped to the depth of a duplicated mono phaser effect from the standard FAUST libraries, configured to have 10 notches of 800 Hz width and a frequency spacing factor of 1.5. The effect is applied to the melody buss, at a sinusoidal LFO frequency determined as *tempo/120* Hz.

- **Melody Wah Wah:** (demo: 1.2.7 in Media Links) In this custom function, x is mapped to the gain of a modulated peaking filter. A sinusoidal LFO at *tempo/60* Hz modulates the effect, which oscillates between 250 Hz and 5000 Hz at the LFO frequency and operates on the melody buss. An output gain adjustment compensates for the loudness increase resulting from the peaking boost.

Additional Synthesizer Sonification

- **Pitched Disturbance:** (demo: 1.2.8 in Media Links) x is mapped to the frequency of a sawtooth wave, which is determined as:

  $$f_{saw}(Hz) = 250 + 4750 \cdot x^2$$

  x also controls the level of the sawtooth wave. If x exceeds 0.01, the melody submix is automatically *ducked* by 24 dB and the sawtooth wave is ‘turned on’. The percussion submix is left as-is. The audible result depends upon the mapped MP, and may either be a slow-moving pitched disturbance or a rapidly fluctuating glitch-like sound interrupting the music.

- **Filtered Noise Disturbance:** x is mapped to the cutoff frequency of a resonant lowpass filter acting on an independent white noise source as follows:

$$fc_{noise} = 1000 + 19000 \cdot x^2$$

Identical to the pitched disturbance, the melody submix is ducked and the filtered noise is only turned on if x exceeds 0.01. The audible result may be either a relatively stationary noise disturbance or a rapid 'whooshing' sound depending on the mapped MP.

5.4 Evaluation

5.4.1 Expert Interview - Music Therapist

The first iteration was also evaluated by means of an expert interview with a qualified music therapist in Aalborg. The purpose of the interview was to demonstrate the key functional components developed thus far, and gain insight into the key areas of novelty, potential use-scenarios and practical considerations.

Setup and Procedure

The interview was conducted in English in a small, quiet room at the UCN campus, Aalborg East and lasted 60 minutes. The music synthesis, manipulation and user interface were demonstrated directly on a Dell laptop using the inbuilt speakers, and the movement-based sonic interactions were demonstrated through high-quality videos showcasing each interaction. Questions related to each topic were posed in a structured fashion, and the answers were audio-recorded with permission and transcribed later. Key points in response to each question were compiled from the transcriptions and summarized in the results.

Results

(Music synthesis, mixing and tempo/rhythm capabilities demonstrated) **Q. What are your comments on the generated music and the current tailoring capabilities?** It was generally OK to listen to, but one observation was that even though different songs were demonstrated, they all had the same 'soundscape' and sounded similar to each other. It is possible that some users feel it is not true to the original. But on the other hand, there is no existing technology to compare this to, and the soundscape problem does not have an easy solution. However, the two most important controls - *tempo* and *rhythm* are available. In clinical reality, all a therapist would do is cycle through rhythms, find out what the patient likes and adjust the tempo thereafter. How the patient perceives the music will depend upon how it is presented to them, and how they wish to experience the music in that context. They could either wish for an *authentic* experience like being at a concert or simply music that can help them in rehabilitation, which does not need to recreate the original song. But with this kind of synthesized music, the emotional impact is different, due to which it may not be able to invoke the same memory-based and emotion mechanisms that the original song would. It still, however, presents the song in a recognizable fashion. The ability of the patient to process the stimuli would depend on their cognitive deficits, and the possibility of modifying the music by muting instruments would be helpful here. Organizing all these different sounds may be difficult if a patient has some form of conscious disorder. It would help to know the musical taste of the patient in advance of a training session.

Q. (Demo - Static Sitting Posture - Melody Ring Modulation Feedback) Comments on pleasantness, meaningfulness, perceptibility? The auditory feedback is perceptible, and it is possible to picture the position of the patient's trunk without looking at the video. It could, however, be difficult to follow for a patient who has attention deficits. Using such feedback would be easier in a *therapist + patient* setting, where the therapist can dial in the feedback settings, draw the attention of the patient to the auditory feedback and show them how to correctly modify their posture.

Q. (Demo - Static Sitting Posture - ML Angle - Percussion Synchronicity Feedback) Comments on pleasantness, meaningfulness, perceptibility? The feedback was annoying, and less pleasant than the previous. As a patient, it may not be enjoyable to have to be attentive to a stimulus that repeatedly tells you that you did something wrong.

Q. (Demo - Dynamic Reaching - ML Angle - Music Brightness Feedback) Comments on pleasantness, meaningfulness, perceptibility? This strategy should be easy to perceive and follow for patients with minimal attention deficits and moderate cognitive deficits. It is advantageous in that it simply provides feedback through changing the sound 'quality' and not explicitly 'right' or 'wrong'.

Q. (Demo - Dynamic Reaching - ML Angle - Melody Envelope Release) Comments on pleasantness, meaningfulness, perceptibility? This feedback was perceptually subtle. The difference between the normal and 'staccato' sound is obvious for a musician but not necessarily for a lay listener.

Q. (Demo - Dynamic Reaching - Scalar Jerk - Pitched Disturbance) Comments on pleasantness, meaningfulness, perceptibility? This feedback is very interesting due to its instantaneous nature, and the way in which it provides negative feedback in a game-like fashion. It is funny and could also be used in playful learning settings, and is most applicable in my own music therapy work.

Q. (Demo - Static Standing - ML Angle - Melody Distortion) Comments on pleasantness, meaningfulness, perceptibility? I have not personally engaged in this particular training exercise, although it is valid in principle. The feedback is easy to hear, although it could possibly be annoying if listened to for prolonged periods.

Q. (Demo - Static Standing - Scalar Jerk - Melody Tonic) Comments on pleasantness, meaningfulness, perceptibility? This feedback, along with the other jerk-based interactions are the most novel presented. I have never seen a system that only responds to jerky movements, under certain conditions. The jerky auditory feedback is easily noticeable, and intuitive as it matches the jerky movement. One consideration is that the sudden feedback should not trigger another jerky movement or series of jerky movements in response.

Q. (Demo - Dynamic Reaching - ML Angle - Melody Wah Wah) Comments on pleasantness, meaningfulness, perceptibility? This feedback is perceptible and not annoying, which is appropriate as it matches a sideways reach, which should not trigger negative feedback. However, this could potentially be annoying to listen to after a period of time.

Q. Can you generally comment on the aspects of the application that you feel are worthwhile to pursue in a future version? This is a capable and versatile application, and it makes sense to focus on aspects that are relevant to rehabilitation, both occupational and physical therapy. The interactions focusing on jerk were the most interesting and novel, and clinical trials will be the main determining factor of the utility of the application. It is important for the therapy to be intuitive to the therapist; RAS has been found by me to be very intuitive, while TIMP (therapeutic musical instrument performance) has not. It makes sense to focus on maybe a couple of exercises that match what a group of patients is doing, and then experiment with the possible kinds of interactions. This would also depend on the collaborating therapist, who must always feel comfortable with and aware of how the application supplements the training. If this is ensured, the patient will also be more open minded and likely to like the session, thus making the 'relations' in the room very important.

Q. As a music therapist, how do you view the potential and utility of this application? At this point, it is not possible with the application but recording movement data would be a major advantage. Ideally, I would like to give the patient equipment that they can use on their own, while it records them. Music therapists are only present for a minimal part of their week (2-3 hours) and every bit of exercise possible at home would be good. Although real instruments are sometimes preferable to give patients guidance and/or feedback, they cannot provide it for aspects such as jerkiness, which is where such an application can be applied.

Q. Are there any other practical aspects that must be considered when developing a future version? These depend on the location of the rehabilitation training. Some facilities conduct individual training, and some do it in groups (e.g. lower/upper extremity). If it is difficult for a patient to participate in group training, they receive individual training. This application would also need to be used individually, as it would be distracting to listen to multiple sets of musical stimuli. Depending on the setting, the auditory feedback may need to be provided through either speakers or headphones. Overall, the main goal at this stage should be to target a very specific group of patients and develop further with them in mind, as such an application can be very good for patients who are able to use it.

Q. Can you comment on the use of RAS in stroke rehabilitation in Denmark? RAS is recommended in the clinical guidelines provided by the highest health authority in Denmark (Sundhedsstyrelsen), but not systematically used. However, physiotherapists use similar 'intuitive' ideas like singing while training, and RAS playlists are used on some occasions. Its use is growing, but it is still not common in systematic healthcare or included in Music Therapy education curricula.

5.4.2 Interviews - Sub-Acute Stroke Patients and Physiotherapist

The participatory study comprised a set of interviews with key stakeholders, specifically sub-acute stroke patients and an accompanying physiotherapist. These were all conducted on the premises of Neuroenhed Nord, Regionshospital Nordjylland, Frederikshavn. The purpose of the study was primarily to assess the prototype from the patient's perspective - I sought to gauge their subjective impressions of the rhythms, musical stimuli and feedback strategies. The interviews were conducted using a set of prepared audio and video demonstrations, where the synthesized music and interactions were showcased.

Participants

A total of 7 sub-acute stroke patients (1 woman) admitted at Neuroenhed Nord, Frederikshavn volunteered themselves for the interview. Exact data was not obtainable but they were all above 50 years of age and recent sufferers. They were cognitively and physically impaired to differing degrees, but all able to understand the purpose of the interview and respond coherently to the questions and tasks posed during its course.

Setup All interviews were conducted in a small and quiet office room at Neuroenhed Nord, Frederikshavn. The audiovisual material was played back using the default media player application on a Dell laptop connected to a Logitech loudspeaker placed approximately one meter from the patient at a comfortable listening volume. As the interviews were conducted in Danish and I am a non-native speaker, a Danish physiotherapist was present during all interviews to assist with communication when necessary.

Procedure

- The first part of the interview comprised a **rhythm tapping exercise**, where system-generated recordings of different types of rhythms of varied tempo and complexity were played back to the patient. Some clips were percussion-only and some of them had melody instruments along with the percussion. A CMR library containing encoded versions of several known English and Danish pop songs was used. The patients were asked to try and tap along in time with the rhythms, and comment on their experience afterwards. Specifically, they were asked how they felt tapping along to the music and what they thought of the prospect of exercising to this type of music. A total of twelve clips were played back in this manner.

- The second part of the interview was a **video session**, where high quality video clips of the various sonic interactions with the device (same as with the music therapist) were shown to the patient. The patient was briefed that the music was under the control of the body, and that certain types of movements would lead to changes in the sound. After watching the each video, the patient was asked whether they could understand the movement-music connection, whether it was easy to perceive, and how they felt about the way it sounded. They were also asked to generally comment on the music. A total of eight videos were shown to each patient, and the complete interview lasted between 20 and 30 minutes. After all patients were interviewed, the accompanying physiotherapist shared inputs and suggestions based on the videos. All responses were recorded and later transcribed manually and translated to English.

Interview Results

It must be noted that the patients had varying degrees of lucidity and expressiveness, so it was not possible to obtain equally detailed responses from all of them. Key points from the transcripts are summarized here (patients referred to as P1-7 where necessary):

- Rhythm-Tapping Exercise: All patients (P1-7) were able to follow and tap along with the march beat, but found the reggaeton and slow rock rhythms harder to follow. An exception was P3, who found the dance rhythm to be the hardest. P5 commented that it was more "fun" when the tempo was faster, and when melody instruments were added to the percussion. P5 added that it was more fun when a song was reproduced using different rhythms than the original. P4 commented that drum fills made it harder to follow the rhythm. P1 mentioned that it was harder to tap along to a rhythm using the non-dominant hand when the dominant hand was weakened, and that many interacting rhythms made the music difficult to tap to (although not difficult to understand). P1 added that the activity was more interesting with multiple rhythms, as one could choose which one they wanted to follow.

- Impressions of Music: The patients had diverse subjective judgements of the music. P1 felt the music was pleasant for those who like "synthesizer music", personally stating a preference for traditional and rock music, although adding that the synthesized music was "nothing dangerous" and would be "OK for training purposes". P2, a rock fan, too felt that it was "OK" but "lacked something" particularly when reproducing familiar artists such as Kim Larsen and Pink Floyd. P3 admitted to being "not critical" of the music used, as its purpose was to help with training and that music helped one "feel less alone". P4 admitted to liking the music, and P5, a classical music fan, expressed an appreciation of the idea that music could mean more in life (a training tool) than before. P6 claimed to like all kinds of music and was similarly indifferent to the type used here. P7, on the other hand, was very discerning of the way his favorite songs (Kim Larsen and Pink Floyd) sounded in their synthesized form, commenting that they "were missing something", "sounded too dark" and "didn't sound right". He said that the music needed to sound nice for training purposes. However, most patients were able to recognize clips of popular Danish songs that they had heard before.

- Sonic Interaction Videos: Once explained and demonstrated, the interactions between the body and the music were generally understood by the majority of participants, and found to be perceivable. One exception was P2, who was unable to see how the body controlled the music in most videos. For the others, the melody ring modulation effect, distortion, and synchronicity strategies were found easy to perceive (P1, P3, P6, P7), brightness and wah-wah to a lesser extent. With the exception of brightness, most felt that these modes of feedback would not be enjoyable to listen to for long periods of time. The jerkiness feedback videos were also generally easy to perceive (P1, P2, P3, P4, P6) and found to be both funny and irritating. P1 said that the feedback would require him to concentrate, but that it would be good to receive such auditory warnings because movements are unconscious. P3 stated that the feedback was generally easier to hear when he knew what to listen for, and that it gradually became easier to hear with repetition. P4 felt that the feedback could be useful when losing balance as compared to verbal instructions. P7 felt in general that most feedback strategies were "awful" to listen to, and caused him to "tune out" the sound altogether.

- Comments from Accompanying Physiotherapist: The physiotherapist offered inputs, both on the prototype and new interaction ideas, summarized as follows:
  - The auditory feedback strategies need to be much more perceptually salient, so as to be understood even by patients with hearing and cognitive impairments.
  - There should be a clear demarcation between auditory punishment and reward, where the former can be used to provide feedback on deviations (such as in static balance) and the latter can be used in goal-oriented tasks (such as dynamic reaching)
  - The jerk feedback can potentially be used in the sit-to-stand exercise, where patients have a tendency to make unsafe jerky movements (such as falling back down onto a chair while sitting).
  - A useful addition would be to provide feedback on heel-strike during gait, either in terms of timing or in terms of heel strike quality (as patients have a tendency to place their feet incorrectly).

5.5 Discussion and Reflection

The first iteration of system design and development largely focused on the creation of a hardware and software framework with the core functional elements of a musical biofeedback system. Appropriate IMU sensing hardware was developed, along with a basic set of movement analysis algorithms. A data-independent 1D sonification structure was realized, with provisions to easily alter its behavior in a user- and activity-specific manner. Robust and flexible music sequencing as well as computationally efficient multitrack music synthesis was designed and implemented. Provisions for individual music adjustments - tempo, rhythm and ensemble density were made. Basic sets of sonification strategies and sonic interaction paradigms were created. A lightweight scheme and application for music structure encoding was also built. All the various parallel operations (sensing, analysis, sequencing, synthesis, sonification) were combined in a thread-safe, crash-free standalone biofeedback application. The stated aims at the outset of this development cycle were broadly fulfilled. This section now reflects more deeply on several key aspects, and discusses findings from the evaluation protocols to pave the way for the next iterations.

Concerning inertial sensing and transmission, sensor scalability and low packet drops were stated as important criteria. While the OSC framework is easily scalable in terms of adding more sensors, the biofeedback application does not have a framework to handle multiple sensors at this stage. This must be added in future iterations. Specifics of UDP transmission efficiency in terms of packet drops were not probed either. Filtering at the receiver end most likely hides the effect of packet drops from the user for the most part, but this must be explored in future iterations to find out whether the transmission algorithms require modification. Specifically, it should be possible to record logs of receiver activity to calculate the number of received packets in a given time-frame. On the subject of IMU signal analysis, the set of movement parameters is quite simplistic, consisting mostly of measures that are easy to directly obtain from raw IMU readings without advanced error correction. Additionally, none of these can directly be used with even basic gait measurement, except perhaps trunk angles while walking. Future iterations must add to this list of MP's to create more and better-defined sonic interaction possibilities. In the 1D Sonification framework, the target value is static by default. Dynamic tasks could benefit from a dynamic target trajectory, and that is not easy to realize using the current interface. Future iterations must integrate mechanisms for automatic target manipulation in order to create target trajectories for dynamic activities. The sonic interaction possibilities must be expanded to formally include walking activities, and tested with real patients in future iterations to assess their merits. The CMR scheme for music was successful at recognizably encoding simple songs, but several clear limitations are already apparent - the fixed time signature, lack of triplet intervals and necessity to stick to scale-defined music intervals and avoid scale and tonic modulations. Passage-wise scale and tonic settings could alleviate these problems.

Another problem area is the user interface of the biofeedback application. Although interface usability from the therapist's perspective is not important at this stage, certain shortcomings make the interface very difficult for even me (the developer and operator) to use in a fast-paced real life setting. The interface gives very little feedback on whether the sensors are successfully connected to the WiFi network, the music playback status, biofeedback settings and mixer settings. Provisions must be made to make critical aspects of system status easily apparent, so that easy troubleshooting is possible when necessary. This could be through the addition of periodic status checks and visual elements to indicate system status. Calibration of system parameters and logging of a training session are not possible either at the current stage, and must be added as well in future iterations. Logging is important to be able to record and analyze patient performance (as per music therapist) and to calculate physical effects of the biofeedback in future studies. The music therapist also mentioned that a good approach would be to target specific training activities and design interactions for them. The user interface must be reorganized to facilitate activity-wise division of MP's and AP's.

Expectedly, the synthesized music elicited diverse reactions from the music therapist and the patients. While most patients said they did not mind the synthesized music while exercising although they had different music preferences, there were a few who felt the music was lacking in some of the songs they recognized. This is possibly attributable to the simplicity of the music synthesis methods, and the same sonic palette used for all songs. This sentiment was echoed by the music therapist as well, who commented on how all songs sounded similar. The interviews reinforced the role played by subjectivity among patients, so future iterations must explore the attitudes of patients towards this form of synthesized music. The choice of synthesized music over pre-recorded music appears warranted at this stage; the music therapist emphasized the value of being able to manipulate the ensemble rhythm, tempo and density. Patients had differing abilities to follow and synchronize to rhythms, making the available flexibility a useful feature. Future iterations must explore the expansion of the sonic palette, through differences in musical structure, instrument textures and mixer settings. As far as the sonification strategies are concerned, it is positive that the patients were able to perceive these, but as the physiotherapist stated, they must be more perceptually salient to cater to cognitively less-capable patients. Strategies such as melody release time and phaser were found to be perceptually subtle and can be dropped. The synchronicity feedback was generally found very annoying, and can also be dropped. The current strategies are simple manipulations of single synthesis or effect parameters. Their perceptual impact can be magnified by creating composite strategies (one-many mappings) or replicating them across frequency bands (reference and zoom [73]). Future iterations must explore both these possibilities. A clear demarcation must be made between auditory reward and punishment, and strategies must be defined specifically for different activities based on this. The jerk-based strategies must be tested with sit-to-stand exercises, and interactions must be developed specifically for gait.

In summary, the next objective is to further explore the utility of musical biofeedback interactions in real tests with stroke patients. Such tests are the only way to understand the practicalities involved in introducing such interactions into training, as well as to get a more robust functional assessment of the system. Before this can be done, the system must undergo several improvements as discussed above. The next iteration will focus on addressing several of the uncovered issues, and its ultimate goal will be to test a refined set of interactions on real patients.

Chapter 6

Iteration 2

6.1 Aims

The second iteration seeks to prepare the application for use in gait and balance training with real stroke patients. This involves addressing findings from the first iteration, adding new *movement parameter (MP)* options, improving existing feedback strategies or *audio parameters (AP's)* and interaction concepts, and developing important utility functions. The main goals are summarized below:

- Addition of new MP measures of heel strike and heel strike periodicity.
- Implementation of functionality for time-varying targets to be used in dynamic tasks.
- Tweaks to music sequencing and mixing.
- Organization of sonic interaction interface into controls pertaining to static balance, dynamic balance, sit to stand and gait.
- Improvement of sonification strategy set to promote perceptual salience, and clearly justified classification of strategies to be used within each exercise.
- Addition of interface functionality for storage (sensor, system logging) and real-time MP data visualization.
- Addition of utility functions such as sensor state and music progress monitoring, as well as a simplified sound mixer for environment-specific adjustments.

The second iteration was evaluated through brief tests conducted with a small group of real stroke patients with ranging impairment levels. These were carried out in collaboration with a physiotherapist, who was later interviewed. This chapter proceeds with a detailed treatment of new design and implementation decisions, before addressing the evaluation protocol.

6.2 Sonic Interaction Set - v2

The initial interaction concepts from Iteration 1 are revised and broadened on the basis of the feedback obtained from the expert interviews from Iteration 1. New concepts targeting gait and dynamic reaching exercises are implemented using the new MPs and dynamic target options respectively. Old concepts are also improved with superior sonification strategies, and a clear boundary is drawn between strategies to be used for auditory punishment and reward. A system calibration phase (as previously described) should be applied in each case to tailor the relevant parameters to the patient prior to the exercise. The interaction concepts as of Iteration 2 are described below:

- **Static Balance - Posture Feedback - Punishment:** This is identical to the concept in Iteration 1, with the goal of the patient to keep their trunk inclination within the target threshold, in the *mediolateral (MLat)* or *anteroposterior (APos)* plane. The only difference is that the simple auditory feedback strategies from Iteration 1 are replaced by the *composite* punishment strategies newly developed (described later in chapter).
- **Dynamic Reaching - Posture Feedback - Reward:** The new functionality to modulate the target with a customizable periodic waveform can be used to guide trunk inclination in MLat/APos planes during upper body reaching exercises, and provide auditory reward to the compliant user. The purpose is to encourage periodic repetitions of a trunk movement. This can be achieved by setting the *desired behavior* to *equal to* with an appropriate target trajectory shape and sufficient dynamic error tolerance encourages the patient to rhythmically follow the trajectory to maintain the quality of the sound. This would involve reaching out to the maximum inclination and coming back to mean position at specific times, with movement and rest intervals determined by the rhythm and timing mode. If following a precise value is too difficult, the desired behavior can be set as *greater than* and the peak target inclination can be set relatively low, so the patient only has to *exceed* a minimum threshold at all times rather than match a precise angular trajectory. The auditory reward strategies (brightness, instrumentation, complexity) lend themselves most suitably to this concept.
- **Sit to Stand - Jerkiness Feedback - Punishment:** This is identical to the Iteration 1 concept, only that it is applied specifically to the sit-to-stand exercise.

- **Walking - Periodicity Feedback - Punishment:** The idea of this concept is to encourage the patient to maintain a stride duration that matches the tempo of the music, with the auditory feedback punishing deviations in stride periodicity in a proportional manner. The music tempo and tolerable stride time coefficient of variation are calibrated beforehand, and the chosen MP is the *Stride Periodicity Parameter* (explained later in this chapter). Any punishment strategy can be used with this concept, but the most perceptually salient and appropriate ones are found to be the melody pitch modulation, pitched disturbance and noise disturbance from Iteration 1.

- **Walking - Synchronicity Feedback - Reward:** This concept effectively sonifies every heel strike event as a triggered snare drum sample, providing direct musical agency and rewarding synchronous stepping with a snare drum that plays in synchrony with the remainder of the ensemble. Thus, this concept also focuses on stride periodicity but phase is critical as the sound is only 'rewarding' if it plays at the correct time within the ongoing rhythm. The *step trigger* parameter is the appropriate MP for this concept, while the *snare drum trigger* is the suitable AP (both explained later in this chapter).

6.3 Design and Implementation

6.3.1 MP Additions - Heel Strike

In this iteration, two heel-strike based MP's are computed with the goal of providing cyclic feedback on stride periodicity and temporal phase matching. Both depend upon the timely detection of unilateral heel-strikes, which is achieved using a simple angle-based detection algorithm on the signal from an anterior thigh-mounted IMU (M5Stack Grey device). The detection algorithm is based on lower limb angular swing and its directional changes when heel (foot) strike occurs. Thus, the IMU information used by the algorithm is the *gyroscope X* reading. *Backward* leg swings (relative to direction of locomotion) register as *positive* angular velocities, and vice versa (depicted in Figure 6.1). Assuming a smooth signal, the heel strike could be assumed to occur at the instant where the angular velocity transitions from negative (forward limb movement) to positive (backward limb movement).

In practice, the gyroscope signal obtained even during normal walking is far from smooth, and requires lowpass filtering to detect important phases with any degree of accuracy. The detection algorithm used here employs a second order Butterworth lowpass filter with an empirically determined cutoff frequency of 1 Hz and Q-factor of 0.7 to obtain the required smoothness in the gyroscope signal. However, this incurs a large phase shift, which translates to detection delays. To get a quantitative idea, the phase shift at the cutoff frequency of a second order But-

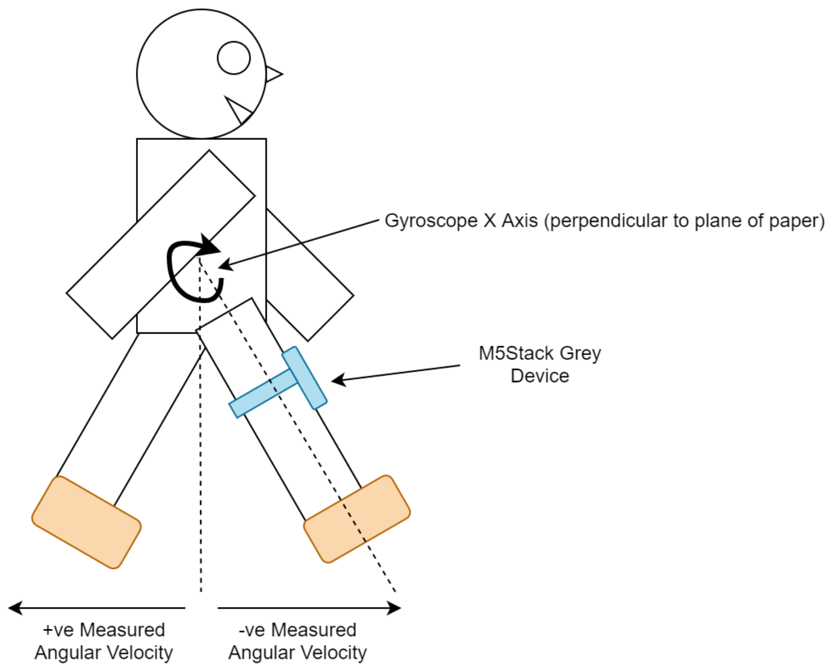

Figure 6.1: Depiction of IMU sensor positioning and detected angular velocity polarity for heel strike measurement.

terworth lowpass filter is $\Pi/4$ or 90 degrees, increasing at higher frequencies and vice versa. At a cutoff frequency of 1 Hz (also close to the normal stride frequency in human walking), this phase shift corresponds to a delay of approximately 250 ms. This, by itself, not only exceeds the auditory reaction time [46] but is also an unacceptable delay especially if impulsive biofeedback sounds are involved.

The heel strike detection must be immediate, and this problem is remedied by implementing a simple predictive model. The model assumes that the heel will strike the ground shortly after the limb passes the *vertical* on its way forward. As mentioned, the filter delay is frequency/cadence-dependent and correspondingly, the model predicts the timing of the heel strike event based on the detected cadence over the last five steps. In practice, the limb crossing the vertical on its way forward can be registered as the instant at which its angular velocity is maximum (assuming simple harmonic motion). In our case this translates to the local *minimum* of the filtered gyroscope X signal as explained earlier. If this local minimum is directly recorded as a heel strike instant, it is now found to be too early. Therefore, an additional *cadence-dependent* time delay must be added to bridge this difference. This is applied to the smoothed gyroscope signal using a simple circular buffer. The delay time is a function of the detected cadence over the past two strides, calculated by an empirically determined formula as follows:

$$cadence = 120/tMean_{stride}$$
$$delay_{ms} = offset + 90 \cdot (max((100 - cadence), 0) \cdot 0.03)^{0.7}$$

The *offset* variable is by default zero, but can be increased in the user interface if found to be necessary. As the formula shows, the required extra delay time increases non-linearly as the detected cadence reduces. This makes sense as the filter phase delay incurred also reduces with reducing cadence, while the interval between limb vertical crossing and heel strike is greater. This was found to yield good timing results in normal walking self-tests, once a regular cadence was established. Of course, there are greater errors in detection timing when a new stride has a significantly different duration than expected, or if heel strike is deliberately delayed after the limb crosses the vertical on the way forward.

The heel strike detection is used to compute one of two MP's related to stride periodicity and stride phase respectively, and these serve as specialized MP's that can be mapped to the available choice of feedback strategies, but they are designed with only certain AP's in mind. **These MP's are calculated such that their bounds and polarities fit the AP mapping rule** (zero value for normal-sounding music and 1 for maximum feedback intensity) and therefore do not need further processing in *SoniMappingCompute*. The *target* MP value in this case can be set to zero (desired behavior: less-than, mapping function: linear). The computed AP value thus matches the MP value.

Stride Periodicity Parameter

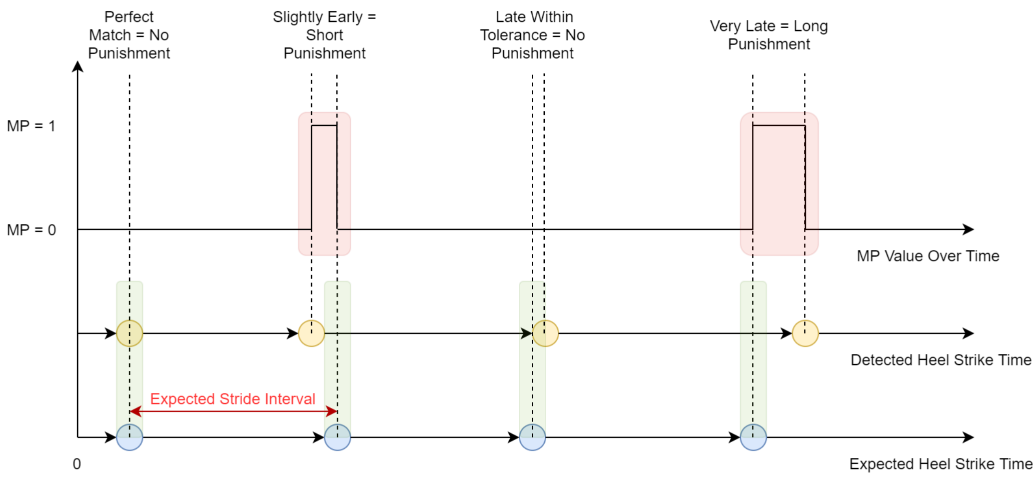

Figure 6.2: Stride periodicity MP calculation in a straight walking condition, showing how the MP is derived from detected and expected heel strike timestamps.

The first specialized MP is a time series that aims to provide feedback on stride periodicity relative to the music tempo. This is essentially achieved by recording the timestamp of a heel strike, computing the *expected* timestamp of the next heel strike of the same foot based on the tempo, and then providing negative feedback of duration proportional to the size of the time discrepancy between the expected timestamp and the detected timestamp.

$$t_{expected}[n] = t_{detected}[n-1] + 2 \cdot interval_{qtr/half}$$

A period tolerance parameter is also present, allowing a certain percentage deviation from perfect periodicity without punishment. This detected timestamp, in turn, will serve as the basis for the calculation of the subsequent expected timestamp and so forth. Thus, this qualifies as cyclic feedback in which feedback is given at the end of a periodic or cyclic movement. The parameter can be configured to target a typical *quarter note* stepping rhythm or a slower *half note* rhythm depending on the needs of the patient. It must be noted that this parameter does not aim to consider the gait phase relative to the music phase, making it suitable purely for period matching. Fig. 6.2 illustrates this calculation. Note that if the expected timestamp is crossed without a detected step, the MP value stays glued to 1 until a step is detected and it is reset to zero again.

Turning: A second LPF processes the gyroscope Y signal (vertical axis) to check if the person is turning. If the instantaneous filtered angular velocity about the vertical axis exceeds 60 degrees per second (empirical threshold), the person is deemed to be turning, in which case the MP remains zero and no feedback is given. This prevents auditory punishment when a patient is turning, as stride periodicity is hard to maintain during the turning phase and ought not to be punished.

Step Trigger Parameter

The second specialized MP is straightforward, simply converting detected heel strikes into a pulse signal whose rising edge can be directly used to trigger either a percussion instrument or some other transient feedback burst. The MP value is 0.5 by default and is set to 1 for a duration of 50 ms every time a heel strike is detected. The default value of 0.5 tells *DspFaust* not to play the normal sequencer-mapped pattern of that percussion instrument, and to *only* trigger it with heel strikes.

6.3.2 Biofeedback Framework Upgrades

Exercise Modes

With an increasing number of available MP's and AP's, as well as system parameters that are activity-specific, the interface quickly becomes cluttered and confusing. Navigation of settings also becomes more time-consuming due to the elements on screen. To address this, the interface controls, MP's and AP's are divided into five exercise modes - Testing, Static Balance, Dynamic Balance, Jerk Punishment and Walking. The lists of available MP's and AP's are divided among these Exercise Modes, such that only those relevant to the chosen activity are made available for use. This reduces the size of the MP and AP drop down lists. The MP calibration button text and functionality (later in this section) also change depending on the chosen exercise mode. As of now, only the Walking exercise has dedicated UI parameters (e.g. Stride time error tolerance) and these are only visible when Walking Exercise Mode is chosen. Similar controls for other activities can be added in the future if necessary. The Testing mode contains all controls and all MP/APs, and as the name suggests, is used for experimentation purposes.

Dynamic Target Modulation

The goal of dynamic target modulation is to address a key shortcoming of the sonification framework in Iteration 1 - difficulties in configuring the system for dynamic tasks. The principle is to automate target trajectories in a rhythmic fashion, and is based on existing neurologic music therapy principles related to PSE and TIMP [99]. These encourage repetition in motor learning by the rhythmic practice of originally non-rhythmic bodily movements. We use the example of dynamic trunk bending for a reaching task. To rhythmically train trunk bending, music can be used as a temporal cue. This concept is extended to include musical feedback, which must provide information on:

- whether the patient's movement reaches the desired amplitude.
- whether the patient's movement followed the desired temporal trajectory.
- whether the patient was able to return to the rest position in a timely manner.

This is made possible by augmenting the MP-AP mapping functionality to enable periodic modulation of the previously static target MP value, changing the target *value* to a target *trajectory*. Patient compliance with the target trajectory can in turn be calculated by continuously calculating the movement error and converting this error into feedback. As it may be difficult to precisely follow an abstract trajectory, allowances must be made to tolerate a certain amount of error. Functionality is added to modulate the target in real-time, while the rest of the AP calculation remains intact. The target modulation function is part of *SoniMapping-Compute*, and requires the following parameters:

- Peak Target Value
- Target Function Order
- Music Tempo

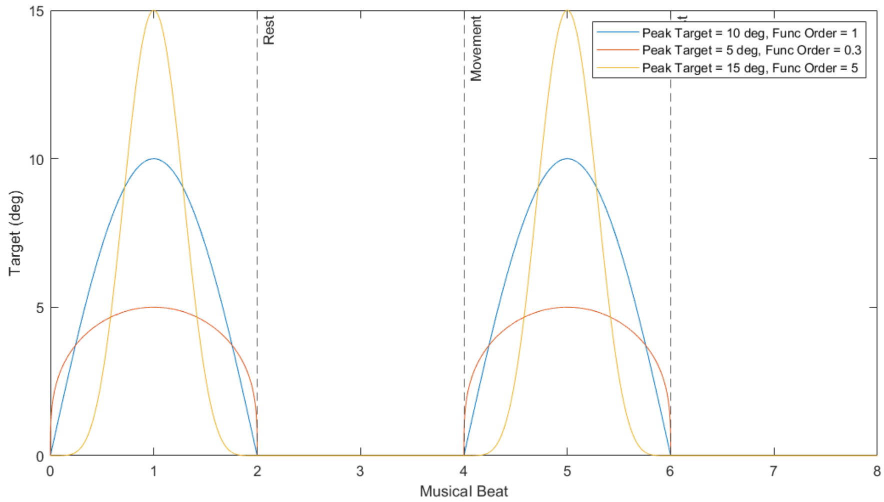

Figure 6.3: An illustration of how a target inclination angle threshold is rhythmically modulated by different parameter settings, resulting in different desired movement trajectories. All of these are divided into movement and rest periods of equal duration that depend on the tempo of the music, which thus serves as a temporal cue for movement.

- Timing Mode

The target modulation function is a half-wave rectified sine phasor raised to the function order. This is chosen because the sine phasor is the most easily achievable periodic function and the exponent order allows it to assume a wide range of shapes depending on the required movement pattern. It would, however, not be able to achieve sigmoid-esque characteristics. It is unknown at this stage what function shape would be most suited to an action such as reaching. The frequency of the phasor depends on the music tempo and timing mode and adapts to it in real-time. In *normal time* mode, a musical *bar* is divided into two beats of movement and two beats of rest, while these durations are doubled in *half time* mode. The phase of the phasor is automatically adjusted so that its cycle start and end coincide respectively with the beginning and end of a musical measure, so that the patient receives reliable phase cues even when the tempo is changed. Examples of modulation functions in normal time mode are shown in Fig. 6.3. These parameters can be set in the user interface, which also has an error tolerance percentage slider mapped to the tolerance bandwidth used in AP calculation. The target modulation can be used with all three desired behavior modes, leading to a versatile set of interaction possibilities. The key configuration challenges are finding an appropriate movement tempo/timing mode, peak movement target and movement function order.

Feedback Calibration

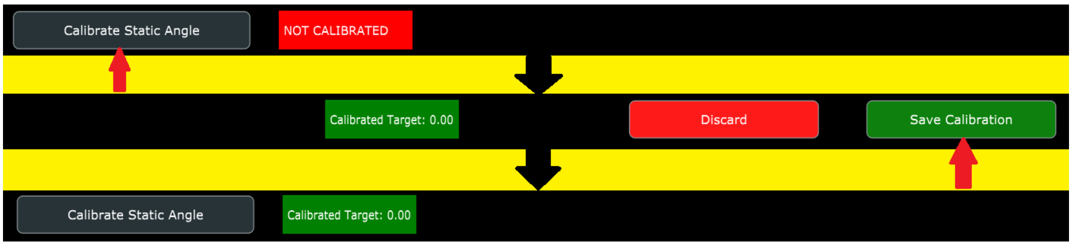

Figure 6.4: Feedback Calibration Controls.

The feedback calibration functionality automates the process of setting system parameters based on measured movement performance. Such process can be quite elaborate depending on the MP that is to be calibrated, and only a basic version is implemented in this iteration. For all MP's, calibration can be initiated, saved and discarded using the same three buttons as shown in Fig. 6.4. Depending on the presently selected Exercise Mode and active MP, the button text and presses are handled differently. The *GaitAnalysis* class stores calibration status (calibrated/not calibrated) and calibrated target values for each MP, so previously calibrated values can be recalled even if the selected movement parameter is changed in the UI. As is visible in Fig. 6.4, calibration status and values are visible on the user interface for the selected MP.

When the process is initiated, the calibration method is first chosen based on the exercise mode. For static/dynamic balance and sit/stand, the target threshold value is set as the global maximum of the angle/jerk parameter measured during the calibration period. For heel strike/walking, the calibrated quantity is the average stride duration (over the last five measured strides) and its coefficient of variation. This measured quantity rapidly updates in real-time and displays on the UI as shown in Fig. 6.4. The user has the choice of storing or discarding this calibrated value when the process is complete. If the user saves the calibrated value, then the 'target' slider is set to this value in case of static/dynamic balance or sit/stand exercises. In the case of heel strike, the cadence is calculated from the calibrated stride time and the music tempo is set to this value. The stride duration tolerance parameter is set to the measured coefficient of variation.

6.3.3 Music Functionality Upgrades

Main Chord Track - Individual Note Velocity Control

In Iteration 1, the simultaneous triggering of all four notes in the main chord track was controlled by a single velocity value obtained from the music CSV. This made it impossible to play arpeggios or other interesting patterns that would mesh with the underlying percussive rhythm. A modification is made in this iteration to address this, namely replacing the single velocity control for the entire track by four discrete controls, corresponding to each note of the chord. This is realized as four separate FAUST sliders, and four different velocity signals and triggers, that are combined with the corresponding frequency information and used to generate four audio signals corresponding to each note. As for the velocity information, it is no longer encoded into the music CSV file, instead it is predefined in *PercPatternLibrary* as a three-dimensional matrix that stores velocity patterns for each chord note, at each temporal location within a bar, for each rhythm (Dance, March, etc.). They are defined in a fashion that was found appropriate for each rhythm, and the values are fetched by the *Sequencer* at every sixteenth note pulse interval as usual. This also has the added advantage of streamlining the music encoding process, as the user no longer needs to encode chord velocity information. An audio demo showcasing the different arpeggio patterns (4 bars of each) can be found in 2.1.1 of Media Links *(clickable)*. Version 2 of the CMR Music Encoder is described separately in this section.

Rhythm-Specific Track Gain Offsets

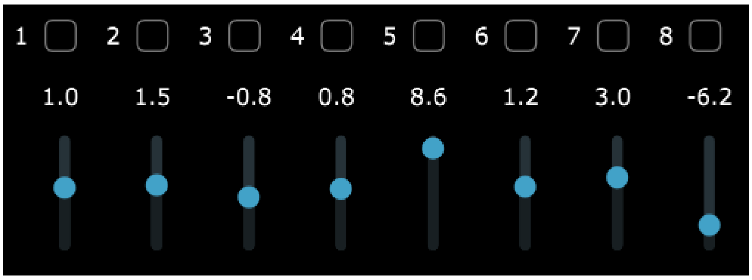

Figure 6.5: UI for manual real-time modification of track-wise gain offsets in dB, located below the respective track mute buttons.

There is a wide range of 4/4 rhythms possible so far, from march beats to rock and dance beats. Due to differences in sonic expectation in all these distinct styles, it is not possible to find a set of mixer settings that provides optimal balance for all styles. Ideally, all rhythms should have their own set of track gain, EQ and compression settings. This is addressed in the present iteration, although EQ and compression settings are still kept constant across rhythms as the instruments do not change. Track gain settings are pre-configured to change when a new rhythm is selected. This is achieved by adding a new matrix to *MixerSettings*, which stores gain *offsets* for each of the eight tracks, for each rhythm. These offsets are added to the original gain value for each track, and the final gain is calculated and mapped to *DspFaust* every time the rhythm is changed in the UI. The offsets range from -10 to +10 dB, which is found to be a sufficient range for making inter-rhythm adjustments.

The gain offsets matrix is pre-populated with values by a manual mixing process for each rhythm to find optimal level balances, but can also be adjusted in real time if the situation demands it. This could be necessary if a patient is unable to hear a certain instrument, or more generally if the output is deemed to sound sub-optimal for a particular song or in a particular environment. An array of 8 JUCE sliders is added below the respective track mute buttons to change and map the offset value for each track in real time. This modifies the values of the matrix in *MixerSettings* directly, and modified settings are thus saved for that rhythm even if it is changed. The sliders themselves are also 'motorized' to snap to the current gain offset values every time the rhythm is changed. Based on the new track gain settings, EQ and compressor settings are also modified to provide the best balance across rhythms. The track mixer UI is depicted in Fig. 6.5.

Master EQ

Figure 6.6: Master EQ sliders to the right of the master gain slider.

A 2-band semi-parametric master EQ is added to the UI, just to the right of the master gain slider. It consists of two peaking filters, with sliders to control their center frequency and gain. They are realized in FAUST using a standard library function for an IIR constant-Q peaking filter, which acts on both channels of the stereo output before the master limiter. They are set to zero gain by default, and have a constant unchangeable Q-factor of 0.7. They are provided in case a frequency range needs to be accentuated or attenuated in a particular acoustic environment. The master EQ settings are depicted in Fig. 6.6.

Song Progress Bar and Remaining Duration

Due to the repetitive nature of the encoded music passages, it can often be difficult to judge the progress of a song, and how close it is to completion. This is important as the operator of the application must be aware of when there is an imminent need to load a new song file. For this purpose, a visual progress indicator as well as a label indicating remaining time are implemented in the music control tab in JUCE.

Progress Bar: A JUCE *ProgressBar* [1] object is the main UI element used here. It is initialized in *GaitSonificationAudioProcessorEditor* (UI class) to monitor a song progress variable in *GaitSonificationAudioProcessor*, which is updated during every timer callback (at 1 ms intervals). JUCE handles automatic update of the progress bar when the monitored variable value changes. The progress is calculated from elapsed sixteenth note pulses. As every music CSV contains a music piece of the same length (24 four-bar measures), the total number of sixteenth note pulses is always 1536, and song progress is obtained as a fraction of this. The progress bar is large in size and colour coded as it crosses 25%, 50% and 75%, so it can be easily seen from longer distances from the screen.

Remaining Time: Remaining time in the song is displayed using a JUCE Label just below the progress bar. The calculation of remaining time is not the same as it would be for an audio file, as the music tempo can be changed in real time, and the label must reflect remaining time at the *current* tempo. This is simply handled by calculating the *total duration* of the piece every time the tempo slider is altered, and computing remaining time from the already available song progress when the progress bar is updated.

Music Encoder - V2

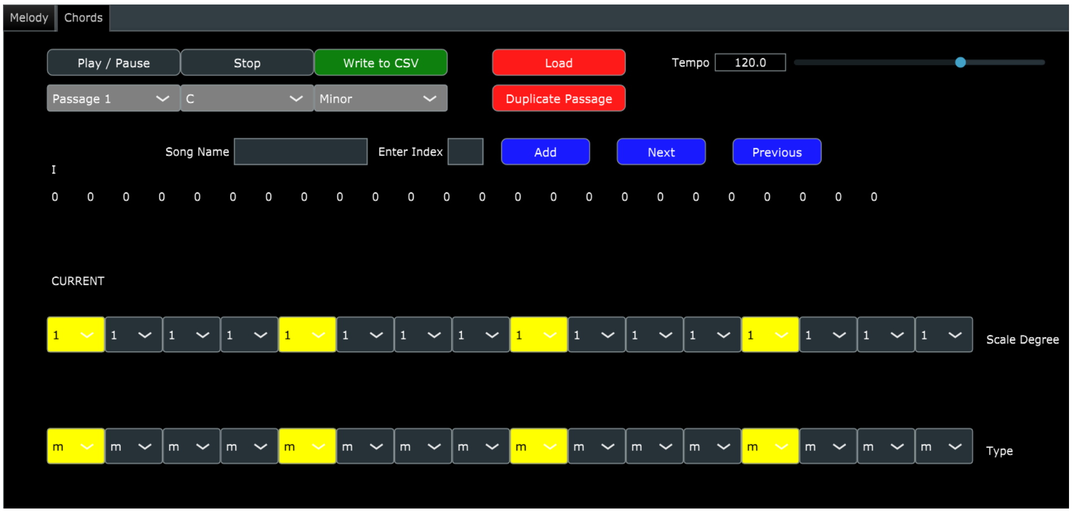

Figure 6.7: The revamped interface of the Music Encoder Application (V2), showing the chord tab. The chord velocity sliders are removed, and the others are replaced by drop down lists to streamline the encoding process. The other elements are also overhauled and color-coded.

A second version of the CMR music encoder is built with a revamped look and

[1]ProgressBar Class Reference. url: https://docs.juce.com/master/classProgressBar. html

feel, as well as several functional modifications. It is now possible for each of the five passages to have different tonics and scales, so as to allow more flexibility in the musical structures encoded. Buttons are colour-coded based on functionality and sliders are replaced with drop-down lists where possible. The melody encoding sliders are left intact, but the chord sliders are overhauled. As the individual chord note velocities are now coded in a rhythm-specific mannner in *PercPatternLibrary*, these values are no longer encodable, and the scale degree and chord type sliders are replaced by drop down lists representing each *beat*, colour coded with yellow representing a new bar. This substantially reduces the amount of manual typing, making the encoding process faster. Also, the chord types are no longer coded as numbers but as meaningful letters such as 'M' for major, 'm' for minor, 'M7' for major seventh and so forth. All other functional aspects are intact.

6.3.4 General Utility Functions

Sensor Connection Status Monitor

A stable wireless connection between the M5Stack sensor and the biofeedback application is critical for proper operation, and it is therefore important to be able to monitor the status of this connection during use. The M5Stack screen indicates when a successful connection is made, but does not give any form of warning if this connection is lost for any reason. This can cause serious setbacks to user experience during training if the auditory feedback no longer reflects the actions of the patient, and there must therefore be a mechanism to inform the instructor in the event of a connection loss. This is achieved by means of a permanently visible JUCE label on the user interface that reflects connection status in real-time.

The principle employed here is that when the connection is active, the *OSCReceiver_UDP* object will be receiving regular OSC messages at an ideal rate of 100 Hz (messages/sec). Hence, a counter variable is added to the *OSCReceiver_UDP* class, which is incremented each time a new OSC message is received. Ideally, this counter should be checked by the main HighResolutionTimer callback at the sensor sampling rate to check for unit increment compared to the previous sample interval. If there is a difference between previous and current counter values, the connection status would be set to TRUE, and if the values are the same, it would be FALSE. But this regularity of checking cannot be applied to a UDP connection, which is known to frequently drop packets. Checking at every sample interval was found to lead to the connection status flag intermittently switching between TRUE and FALSE whenever packets were dropped. This is undesirable as the connection is, strictly speaking, active despite the packet drops. To mitigate this intermittent flag switching, the received packet count is checked every *two seconds* so as to ignore minor packet drops that would be filtered out in any case at the OSC pre-processing step. The interface label, in turn, is updated at the interface refresh callback frequency. The two second waiting period between message count checks does add a maximum latency of two seconds to the label update when connection status changes, but this is more tolerable than the rapid connection status switching which would otherwise be experienced.

Data Logging

This allows the instructor/operator to log a training session in real-time. If the sensor connection status is TRUE, the RECORD button becomes active and can be pressed. When pressed, the application creates a new directory in the application folder, named using the active movement parameter and the present time/date stamp. Within this folder, two CSV files are created, and populated with new data during the sensor callback:

- **Raw IMU Data:** This file stores the raw, unfiltered IMU signals as received from the M5Stack. Each set of simultaneous ACC (3 axes) and GYR (3 axes) values is stored in a single horizontal row, and subsequent values are appended as subsequent rows, so each column corresponds to one axis of measurement.

- **Full Log:** This file stores logs detailed information during every sensor callback, including the recording timestamp, calculated movement parameter, music tempo, active rhythm index, target type (static/dynamic), target value, desired behavior and music playback status.

The two files are filled synchronously, so their corresponding rows represent the same instant in time and can be accurately plotted together for comparison.

Real-time Visualizer

Although the primary mode of movement feedback is auditory, a visual component in the interface can pose several benefits both to the interaction designer and the therapist/instructor conducting the training:

- Allow them to keep track of measured patient performance when the patient is using headphones and the feedback is not audible to all.

- Provide clear performance information to therapists who are not yet accustomed to drawing movement inferences from the movement-audio mapping.

- Provides a simultaneous visual reference to the interaction designer when a feedback strategy or interaction paradigm is being tested, making it clear whether the interaction behaves optimally, or if the mapping function needs to be modified.

Figure 6.8: Real-time Movement Parameter Visualizer. The numbers (0 and 25 in this case) represent the MP value range. The red 'C' box represents the present MP value along this range. The T boxes represent the target value or range depending on the desired behavior.

As all interactions are one-dimensional (one MP mapped to one AP), a compact yet sufficient visualization strategy would be to indicate the MP value as a point on a horizontal line, whose ends represent the minimum and maximum values for that MP. The target value/range based on desired movement behavior can be represented on a parallel line. This is the approach followed here, and the implemented visualizer is shown in Fig. 6.8.

The implementation method itself is relatively crude and basic. The maximum and minimum bounds of the MP are fetched from *GaitParamInfo* and depicted at the left and right extremes of the designated space. In between these, the current value 'C' and target value/range 'T' are visualized as rectangular blocks. These are simply realized as JUCE labels with red and green/blue/grey background colours respectively, based on the desired movement behavior. If the visualizer is enabled, the interface callback continuously monitors the target values, desired movement behaviors and current values and updates the visualizer at the callback rate of 25 Hz. If the target value and/or behavior is modified, the 'T' label is updated by modifying its X position and width as shown in Fig. 6.8. If the target is dynamic, the automatic movements of the target/target range also reflect smoothly in the visualizer. If the current MP value changes, the X position of the 'C' label is modified accordingly in real-time. This approach is found to work well in concurrent feedback scenarios without adding any noticeable computational overhead.

6.3.5 Movement Sonification Strategies for Real-time Feedback - Set 2

The feedback received during the evaluation of Iteration 1 is used to design a new set of sonification strategies for real-time movement feedback, addressing the key issues of perceptual salience and a clear conceptualization of 'reward' and 'punishment'. The perceptual salience issue is addressed by reinforcing the existing one-to-one AP mappings (one AP value to one musical instrument/effect) with the use of *composite mappings*. These can be understood as one-to-many mappings where each AP simultaneously maps to separate synthesis or effect parameters to create an overall sonification that is more perceptually salient by virtue of manipulating the music signal in multiple ways instead of just one. Strategies are clearly classified as reward-based or punishment-based. The general principle of auditory *punishment* is that the music is *degraded* by negatively manipulating its melodic, harmonic and timbral components. On the other hand, auditory *reward* leverages the implicit reference that the user has for normal-sounding music, providing a full ensemble free of manipulation to reward desired performance while simply stripping away its individual elements, fullness or complexity when this is not the case.

In addition to the three sonification types used in Iteration 1 (music synth parameter-based, audio effect-based and additional synthesizer-based), we introduce a fourth type in Iteration 2, namely *Sequencer-based*. As the name suggests, these sonification strategies are applied at the sequencing stage by manipulating the very information that is responsible for triggering and controlling music synthesis. Although only one such strategy is developed for this iteration, it is certainly a powerful way of manipulating a music ensemble due to the breadth of the potential design space. As in Iteration 1, each strategy is controlled by JUCE using a separate AP slider with a normalized value range x = 0-1, where 1 corresponds to maximum feedback intensity.

- **Melody Detune - Frequency Distortion:** (demo: 2.2.1 in Media Links) This punishment strategy is composed of two manipulations of synthesis parameters, where x is mapped as follows:
  - *Main Melody Vibrato Intensity:* x controls the depth control of a sine LFO acting on the calculated fundamental frequency signal of the main melody. Hence, increasing x leads to a vibrato effect of increasing intensity, to the point where the melody is detuned no longer recognizable. It is realized as follows (code edited for clarity):

```
//Calculate LFO Value
soniVibratoLFO = 1 + x * os.osc(tempo/15) * 0.5;
//Calculate F0 signal
 fundamentalCooked = fundamental * soniVibratoLFO;
```

- *Chord Note Frequency Distortion Factor:* As x increases to 1, the frequencies of the individual notes of the chord tracks are offset by different multiplicative factors, progressively adding inharmonicity to the sound. For the four notes of the chord, the factors are all 1 when $x = 0$, and when $x = 1$, they are 1.05, 0.97, 1.31 and 0.73 respectively. The factors vary linearly as x.

The minimum value of x is kept at 0.0101 instead of 0. This is done because the small amount of detuning at this x value was found to sound more pleasant and lively than the unprocessed signals, and therefore made the norm.

- **Cartoon Effect:** (demo: 2.2.2 in Media Links) This punishment strategy aims to convert the music ensemble into an unpleasant sonic caricature of itself as x increases. This is a 'hybrid' strategy as it combines an audio effect-based strategy (band pass filtering) with an additional synthesizer-based strategy (melody caricature) This is a composite strategy that simultaneously maps x in the following ways:

  - *Drum Filtering:* The drum tracks (bass drum, snare drum, hi-hat) are processed using second-order FAUST resonant bandpass filters at the end of their ordinary signal chains, with center frequencies 400, 1000 and 10000 Hz respectively and a Q factor of 20. The input signal is split into dry and wet paths, and if x exceeds 0.01, the wet gain is given by

$$gain(dB) = -20 + 20 \cdot x$$

    The dry gain is simply (1 - linear wet gain) and is thus controlled by x. In addition, the center frequency of each filter is modulated by a sinusoidal LFO at an arbitrary frequency of (tempo/97) Hz. The modulation depth (and resulting frequency deviation) is modulated by x as

$$fc = fc_{orig} \cdot (1 + 0.8 \cdot x \cdot sin(2\pi \cdot (tempo/97)) \cdot t)$$

  - *Original Ensemble Level:* The original melody instruments are softened as x increases, such that:

$$gain(dB) = -20 \cdot x$$

  - *Detuned Synth Melody Level:* A detuned caricature of the original melody is added as x increases, using a FAUST filtered, envelope-controlled and frequency-modulated sawtooth oscillator. The following modified code snippet realizes this:

```
//Filter Cutoff Calculation
Soni_P2_PeakFreq = 10000 * (0.01 + x) ...
* (Soni_P2_Env) + 100 : si.smoo;
//Envelope Calculation
Soni_P2_Env = pow(en.ar(0.001,release,masterClock*mv_present),2)...
  with {release = beatTime * 2 / (1 + 2 * x);};
//Putting it all together
Soni_P2_PulseMelody = 2.3 * os.sawtooth(freqCooked) * Soni_P2_Env: ...
fi.resonlp(Soni_P2_PeakFreq,3,1) : _*(gain_SoniP2) : getPanFunction(0)
  with {
  //Gain Calculation
  gain_SoniP2 = - 20 + 20 * x : ba.db2linear : *(x > 0.01);
  //Fundamental Frequency Calculation
freqCooked = melody_fundamental_freq * (1 + x * 0.4 * os.osc(0.2));
};
```

The combination of all these elements converts the original music ensemble into its cartoonish caricature as x increases.

- **Melody Ring Modulator V2:** (demo: 2.2.3 in Media Links) This punishment strategy reinforces the ring modulator strategy from Iteration 1, by adding an extra manipulation, specifically of the melody note frequencies. Normally, the computed f0 values of the main melody and chord tracks remain constant at least until the next note/chord arrives. In this composite strategy, this frequency signal is additionally 'cooked' by multiplying it with a factor that causes it to slowly dip as the amplitude envelope gets softer:

$$f0_{cooked} = f0 \cdot (1 - x \cdot (1.0001 - envValue))$$

This creates a 'drooping effect' and reinforces the sense of detuning. The factor 1.0001 is used to prevent the frequency from becoming zero when the envelope dies away completely, which would make the oscillator unstable. Strictly speaking, this sonification strategy is also a 'hybrid' strategy as one component acts as an audio effect (ring modulation) while the other manipulates a synthesis parameter directly (pitch drooping).

- **Instrumentation:** (demo: 2.2.4 in Media Links) This *reward* strategy works by reducing the gain of all instruments except the main melody and bass drum as x increases from 0 to 1. The reward is achieved at $x = 0$, where the entire ensemble plays normally at full volume. The gain of the other tracks is reduced as follows:

$$gain(dB) = -80 + 80 \cdot x$$

- **Brightness:** (demo: 2.2.5 in Media Links) This is identical to Iteration 1, but designated as a *reward* strategy.

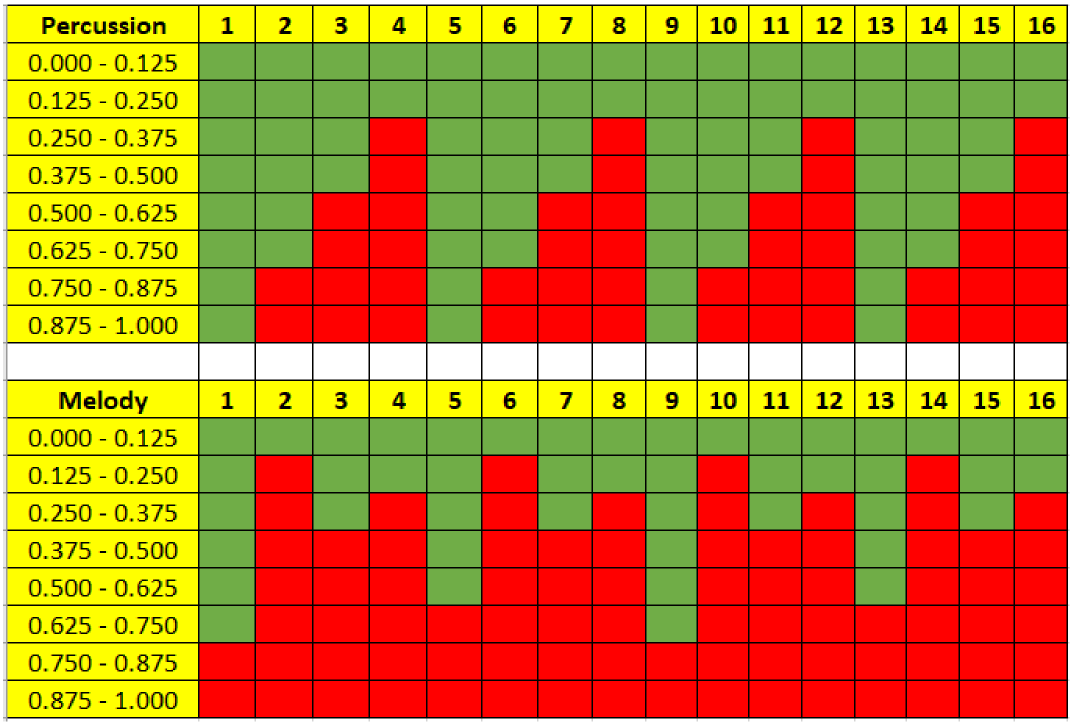

Figure 6.9: Depiction of the musical complexity strategy in different value ranges of x (left column). Red squares indicate temporal locations within a bar at which instrument articulation velocity is multiplied by zero, and green represent locations where it is multiplied by 1.

- **Musical Complexity:** (demo: 2.2.6 in Media Links) This *reward* strategy works directly at the *sequencer* level on the fetched *instrument articulation velocity* values of all tracks prior to mapping them to *DspFaust*. As x increases, this strategy multiplies the fetched velocities within a bar by zero at a progressively increasing number of sixteenth note locations, essentially stopping the instrument from playing at those locations and simplifying its 'musical part'. The reward is realized at $x = 0$, where the entire ensemble plays at full complexity. This is shown in Fig. 6.9, where it is clearly visible how the complexity is made to reduce as x increases. Green squares depict sixteenth note locations where the instrument plays normally, and red squares depict locations where articulation velocity is multiplied by zero. As x increases (leftmost column), the proportion of red cells in a bar increases for both melody and percussion instruments. Complexity reduction patterns are different for melody and percussion instruments as shown.

- **Snare Drum Trigger:** (demo: 2.2.8 in Media Links) This is a separate triggering parameter for the snare drum, meant to be used with the *heel strike trigger* MP to trigger the drum sound while walking. This can be considered a reward strategy in that timing a heel-strike correctly while walking will result in the snare drum playing in time with the rest of the musical ensemble. When this strategy is inactive, $x = 0$ and when $x < 0.49$, the snare drum is programmed to play as usual based on the articulation velocity values periodically mapped to it by the *Sequencer*. But when the heel strike trigger MP is selected, its default value (x) is 0.5, telling the snare drum track to ignore incoming sequencer information and wait for the user to trigger it manually (by walking). The snare drum is triggered whenever there is a non-zero difference between successive samples of x (ideally a detected heel strike).
- **Pitched Disturbance, Filtered Noise Disturbance, Melody Tonic:** These are identical to Iteration 1, but designated as *punishment* strategies.

6.4 Evaluation with Stroke Patients

The evaluation of the second iteration marked the first time the application was used by real stroke patients. The aim of this participatory study was to test the developed interactions and feedback strategies with patients in a real-life training scenario, and obtain physiotherapist feedback on the same through a brief interview. The study was also conducted on the premises of Neuroenhed Nord, Frederikshavn over the course of a single day.

6.4.1 Participants

Six sub-acute stroke patients (2 women) admitted at Neuroenhed Nord, Frederikshavn were recruited for the study. This was done by the accompanying physiotherapist, and they were provided with a detailed information sheet about its purpose, maximum duration and the training exercises they would be undertaking. Exact data was not obtainable but all patients were above 50 years of age. The patients had differing degrees of physical and cognitive impairment, on the basis of which they were assigned exercises by the physiotherapist, who conducted the training for all of them.

6.4.2 Experimental Setup

The study was conducted in a broad corridor at Neuroenhed Nord, Frederikshavn. The biofeedback application was run on a Dell laptop, whose audio output jack was connected to a Logitech loudspeaker placed on the same table as the laptop. The playback volume was not measured, but set to a moderately high level to allow

No.	Name	Num. Patients	Auditory Feedback Strategies	Recorded MP
A1	Static Sitting	1	Melody Detune - Freq. Distortion, Cartoon Effect, Melody Ring Modulator V2	MLat Trunk Inclination
A2a	Dynamic Reach - Dynamic Target	1	Brightness, Musical Complexity, Instrumentation	MLat Trunk Inclination
A2b	Dynamic Reach - Static Target	1	Melody Detune - Freq. Distortion, Cartoon Effect, Melody Ring Modulator V2	MLat Trunk Inclination
A3	Sit to Stand	5	Pitched Disturbance, Melody Pitch	Scalar Jerk
A4	Rhythmic Walking	4	S1 - Noise Disturbance, S2 - Snare Drum Trigger	Stride Time, Raw Gyroscope Signal

Table 6.1: A summary of training activities, feedback strategies used and recorded measures obtained from the Iteration 2 evaluation with stroke patients.

walking patients to clearly hear it. The sitting-oriented exercises were performed on a standard training bench with adjustable height, while walking was performed in a short segment of the corridor close to the loudspeaker.

6.4.3 Procedure

When they arrived at the study location, their signed consent was obtained and it was reiterated that their participation was completely voluntary, and that they could withdraw at any time without needing to justify it, and without impacting present or future rights to treatment. All communication with patients was done in Danish, and the collaborating physiotherapist assisted when necessary. Activity-wise procedures were as follows:

A1 - Static Sitting: The patient was made to sit upright on a training bench without back support, and the therapist set arm reaching tasks that were to be performed without inclining the trunk, and the patient was reminded to "make the music sound good at all times". The tasks involved both forward and sideways reaching, and lasted about 5 minutes.

A2a - Sitting Dynamic Reach - Dynamic Target: The patient was asked to incline their trunk sideways and touch an object - in rhythm with the music, and the working principle was shown to the patient. During the task, the patient was given visual cues initially to help understand when to reach out and come back to mean position. The activity lasted six minutes.

A2b - Sitting Dynamic Reach - Static Target: The goal of the patient was to (at his own pace) reach out repeatedly by inclining their trunk and exceed that threshold, whereupon the music would sound normal and pleasant. ML angles below the threshold were punished with unpleasant auditory feedback.

A3 - Sit to Stand: The patients were instructed to stand up and sit down repeatedly on a training bench in a fashion that was as balanced and careful as possible. They were instructed that they would hear disturbances if their movements were too rough or jerky. The activity lasted 3-6 minutes based on the endurance of the patients. The jerk threshold was manually adjusted on a case-by-case basis with direction from the physiotherapist.

A4 - Rhythmic Walking: The initial music tempo was set low (about 80 beats/min) and the rhythm was set to a simple march beat. For both strategies, the patient was asked to walk in time with the music. For S1, they were told that failure to do so would lead to unpleasant noises in the music. For S2, they were told that their foot would act as a drum sound in the music and their goal was to make that drum play at the correct time. The activity lasted 6-8 minutes for each patient.

6.4.4 Observations

A1 - Static Sitting: It transpired that the patient had better static balance than anticipated by the therapist, and was able to sit upright when not performing a task. The target inclination threshold angle was therefore set relatively low (about 2 degrees on either side). The patient was able to perform the set tasks without falling over, and perceive all auditory feedback strategies. He showed increased arousal during the session, even moving his body to the music (which led to negative auditory feedback being triggered in some instances). Overall, the trunk tilt measurement mechanisms functioned as expected.

A2a - Sitting Dynamic Reach - Dynamic Target: This task proved difficult for the patient. Although the patient understood the task when it was demonstrated, he was unable to time his movements to the musical rhythm, both in terms of period and phase. The auditory feedback also proved uniformly difficult for him to perceive. Visual cues were therefore given, but he was not able to meet a target maximum inclination of 16 degrees, only managing about half that angle, as shown in Fig. 6.10.

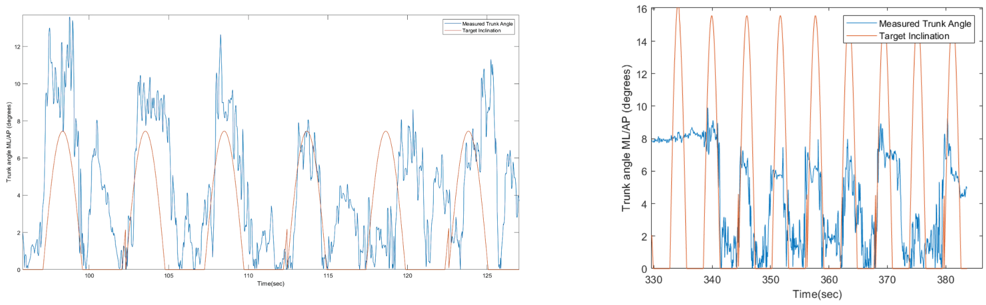

Figure 6.10: A2a - Sitting Dynamic Reach - Dynamic Target. LEFT: The patient was unable to spontaneously match the target trunk angle rhythm, as evidenced in the phase mismatch between red (target) and blue (measured) curves especially in the right of the graph. Note also the blue peaks in between red peaks. RIGHT: When visual cues for timing were also provided, the patient was able to better match the rhythm, but then unable to reach the target angle of 16 degrees.

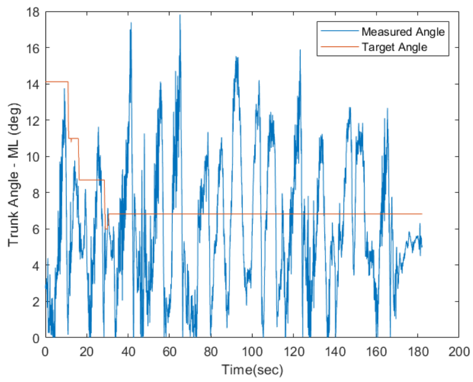

Figure 6.11: A2b - Sitting Dynamic Reach - Static Target. The target was static (horizontal red line) and the patient (blue curve) was consistently able to meet and exceed it in the absence of the rhythmic constraint.

A2b - Sitting Dynamic Reach - Static Target: Having a musical background, the patient was able to both perform the task and clearly hear all the changes in the music in response to his movements. The configured target trunk angle proved manageable, as shown in Fig. 6.11.

A3 - Sit to Stand: The measurement mechanisms functioned as designed. Patients were clearly able to hear the auditory feedback, and adjust their movements accordingly. The majority stated that they enjoyed the activity and the manner in which they could hear their movements. Two of the four patients had difficulties standing up from a sitting position and ended up falling back down on numerous occasions while attempting to stand. This necessitated adjustments in seat height.

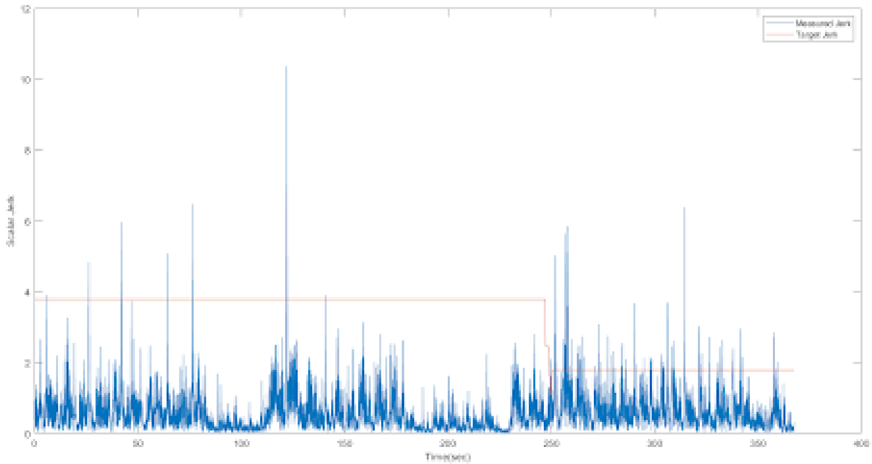

Figure 6.12: A3 - Sit to Stand. Plot of measured jerk (blue) and target jerk threshold (red) over a 5 minute sit-to-stand training session. Barring sporadic spikes, this patient was able to keep the blue curve below the red line.

On some occasions, auditory feedback was triggered even when the patients felt their movements were smooth and balanced, which puzzled them. This was likely due to the jerk threshold being configured too low in these instances. Jerk plots from a 5 minute exercise excerpt are shown in Fig. 6.12.

A4 - Rhythmic Walking: Step detection and triggering worked best when the sensor was connected to the unaffected limb. In a few instances when the paretic limb was used, there was a observed increase in false step detection and incorrect feedback. In a small proportion of instances, the drum sound was triggered slightly before foot contact, or during quiet standing. All patients could hear and understand the auditory feedback. All patients were able to follow the rhythm and match the musical periodicity with their strides (see Fig. 6.13), but P4 stated difficulties in being able to hear the rhythm clearly and it had to be changed and adjusted. Most patients had periods of good and poor synchronization, depending on concentration, distance from the loudspeaker and the need to turn. Verbal cues (1-2-3-4) were often necessary to guide them back into proper rhythm. P4 had significant step-time asymmetry, and stride period matching was challenging for her. 2 of 4 patients increased cadence noticeably during the trial, and the tempo of the music had to be manually adjusted. The majority of the patients enjoyed the task, as well as the musical agency and control they had while performing it, especially when triggering the snare drum with their heel-strikes. The lack of space caused patients to have to turn repeatedly, which increased the fall risk and triggered negative auditory feedback despite the turn-detecting mechanism.

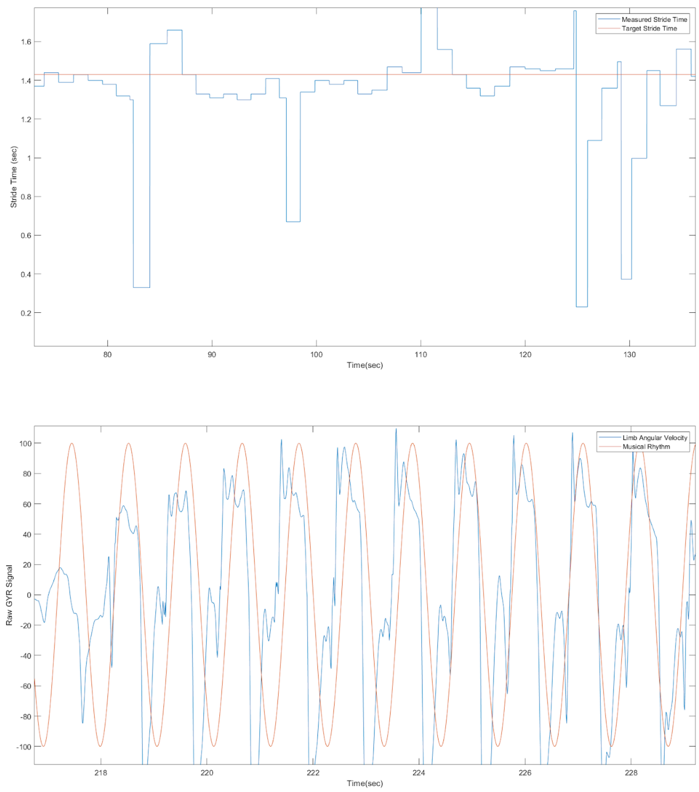

Figure 6.13: A4 - Rhythmic Walking. TOP: Observed stride time (blue) v/s ideal stride time (red) in one trial. Good matching is seen, except sporadic downward spikes attributable to shuffled steps during turning. BOTTOM: Plot of measured limb angular velocity(blue) v/s a sinusoid at the frequency of the music tempo *(tempo/60)*. This is an example of good period matching; the sinusoid does not represent the *phase* of the musical beat cycle, so it is unclear precisely how well the patient was able to match rhythm phase.

Interview with Collaborating Physiotherapist

After testing with patients was complete, the physiotherapist had several general and specific inputs. These are summarized topic-wise as follows:

Perceptual Salience of Feedback: The first input was that the difference between sounds representing "good" and "bad" movements needed to be "much clearer", and that in the current version, it was too "blurred". She emphasized that the patients that had been tested that day did not have severe cognitive impairments, and were very attentive and that one could therefore "get away" with that level of perceptual salience. For example, both the underlying rhythm and the triggered drum during the walking exercise needed to be more emphatic and "fire" the patients up more, compared to the present sound.

Interactions: She expressed that the concepts behind the interactions made sense, but that there was room for improvement. The existing rhythmic dynamic reach, for example, would not be practically feasible for many patients whose cognitive impairments include rhythm-following and complex tasks, although the idea of rewarding "the perfect reach" was still worth pursuing, albeit differently. To this end, she suggested that dynamic reaching be conducted with a static target and/or without the rhythmic constraint. This could be used with tasks that are designed to make a patient incline their trunk and keep it in that position. Moving on to the next interaction, she felt that the jerky sit-to-stand feedback made perfect sense in the training context, as did the concept of rewarding heel strikes that maintained the right rhythm. She especially felt positively about the former, because the jerky audio feedback was "rugged" and "irritating" when it needed to be. On the latter, however, she explained that it was important to measure *both* feet and not just stride duration from one foot. Capturing only one foot would not account for asymmetry, an important functional gait quality indicator for stroke patients that could easily go undetected without punishment even in correct stride rhythms.

Using Supplementary Cues: One important input the therapist had about conducting the training with the patients was the use of extra human cues (e.g. verbal timing cues along with music) to help guide them in the early stages of learning the interactions. She said that while it was acceptable in some circumstances, it would make more sense while testing to allow the patients to rely *solely* on the stimuli generated by the system, as would be the case in any future randomized control trials. That said, she added that she observed that all patients liked the training and were physically aroused by the music, which is why she felt music would be a "fantastic medium to get access to their brains".

Music: On the topic of the music, she admitted that it did not sound very nice and requested that it be improved. She underlined that if the device was one day meant to be used on a widespread basis, the music would "simply have to be better". She conceded, however, that none of the patients seemed to have been "put-off" by it, but personally expressed that the current system output was "not music". She expressed that familiarity was important and recalled one of the patients who had recognized the song ‘Country Roads’ during training and immediately taken to the exercise with more positivity. Despite this, she said that getting prior information on patients’ preferences would be difficult, and that it would make more practical sense to have a library of songs prepared in advance, from which one could be chosen at random while testing with patients.

Practicality: On the subject of practicality, the therapist did not feel that the setup process was too time-consuming nor did she think any of the patients felt any kind of discomfort while wearing the sensor apparatus.

6.5 Discussion and Reflection

The broad goal of Iteration 2 was to bring the application to a functional level where it could be used with real stroke patients during their balance and gait training. This involved the addition of gait-related MP’s, dynamic target modulation, music functionality improvements, enhanced sonification strategies and the addition of utility functions for sensor status tracking, data visualization and music control. Generally speaking, these goals were all achieved during the course of Iteration 2. The heel strike MP’s largely functioned in a satisfactory manner in tests with real patients. The framework for dynamic target trajectories with modifiable shapes was also realized. The music was improved over Iteration 1 in terms of rhythm-specific chord arpeggios, mixer settings and improvements in overall sonic balance. The CMR representation was also upgraded to allow more flexibility in terms of melodic and harmonic content. A new set of sonification strategies was implemented, which was a perceptual improvement over the previous, but still insufficient in some regards. The CSV logging functionality functioned as intended, providing an easy way to store patient performance and reproduce plots of various activities. The MP data visualizer was helpful in debugging program flaws and monitoring patient performance in real time. This facilitated the task of setting movement targets (especially for sit-to-stand jerk). The sensor status indicator proved useful quickly detecting instances of sensor disconnection.

Although it is positive that the concepts of all interactions were judged by the therapist to be well-founded, it is clear that all interactions have substantial room for improvement at this juncture. Regarding upright static posture and dynamic reaching, a limitation of this iteration was that they could not be tested on sufficient patients. The primary complaint was that the auditory feedback was too 'blurred', or not perceptually salient enough for severely affected patients. This is similar to the feedback obtained in Iteration 1, persisting despite converting the punishment strategies into composite ones (one-many mappings). A likely reason for the perceived 'blurriness' could be that the mapping function is continuous, causing difficulties in perceiving small feedback changes (e.g. ring modulation depth). A second shortcoming of the static balance interaction is that feedback can only be given for MLat *or* APos orientation, and that the angle MP's are absolute values (meaning directional feedback is impossible). Both dimensions are equally important to posture, and a single MP must capture them to be effectively used for static balance training. All these issues can be collectively addressed by designing an MP based on the concept used in [17], where the 2D MLat-APos plane is divided into six feedback zones.

The dynamic target modulation proved practically infeasible, although it was only tested on a single patient. This was possibly due to the complexity of the task, which included rhythmic trunk bending along an abstract trajectory. Nevertheless, the therapist did acknowledge the merits of the underlying concept - rewarding the perfect reaching motion. Moreover, the interaction converts this motion to a rhythmic one, which can lend itself to TIMP and PSE [99] if properly realized. The shape of the function must be made more flexible in order to accommodate different movement trajectories; at this time it is not known whether the exponentially raised sine phasor would suffice in this regard, so further tests with well-defined movements (e.g. reaching) are necessary. The patient was generally unable to time his movements and reach the desired angles, leading to the *reward* feedback being ambiguous and hard to perceive. Making the interaction more intuitive is key to solving these problems, and this could be achieved through visual cues from the therapist, auditory cues to initiate and terminate movements and more meaningful feedback strategies. This must be addressed in future iterations. The jerk feedback interaction was found to work well in sit-to-stand exercises, and the feedback strategies were judged to be appropriate. Future iterations can still improve these further.

The gait interactions too showed promise; most patients were able to easily understand the tasks and found them enjoyable, especially the drum-triggering interaction. The cadence increases in some cases were remarkable too, although this could compromise safety due to insufficient balance. Here too, there is potential for improvement. While the heel strike detection functioned acceptably during phases of straight walking at quasi-steady cadences when detecting the non-paretic limb, false detections were observably higher when turning, varying cadence and when the sensor was attached to the paretic limb. Imprecisions in heel strike detection ultimately serve to confuse the patient, degrading user experience and potential gain from using the system. Additionally, the unilateral sensing was judged by the therapist to be less-than-ideal, due to its inability to capture asymmetric patterns. In future iterations, the heel strike sensing must be upgraded to be bilateral, with at least two sensors instead of one. An algorithm change to an impact detection-based scheme would also reduce the incidence of early or false detections. The auditory feedback also needs to be made more salient, particularly the triggered drum sound. A final issue was that patients had a tendency to lose rhythm, which could either be attributed to attention deficits or a lack of rhythm salience in the music. The latter is a matter of concern, and must be addressed in future iterations.

The final topic of discussion is the synthesized music, a matter that drew criticism from the therapist. While the focus of development has been the interactions, the sonic and expressive qualities of the music itself have received relatively little attention to this point. Despite this, none of the patients complained about the way it sounded but the therapist insisted that it needed to be better in order for the interactions to be enjoyable. The shortcomings of the music can be attributed to several factors including the simplicity of CMR, its inability to encode complex song structures and the monotony and crudeness of the music synthesis algorithms. Specific shortcomings are the limited number of passages, fixed total length, fixed scales, low resolution velocity quantization, lack of timing flexibility and lack of expressive information encoding. Using MIDI notation can address the issues related to CMR, although several architecture changes are necessary for MIDI to be usable within the current framework. Variety can be added to the sonic textures of the music by broadening the sonic palette to include several types of synthesized sounds that suit the different available rhythms. The distinct rhythms can thus be programmed to have different instrumentation to make the music less monotonous and better suit individual preferences. So far, the instruments have been completely synthesized from basic waveforms. The integration of a framework to use high-quality audio samples could dramatically improve sound quality, reduce computational load and make greater variety easily possible. A revamped architecture for music encoding, sequencing and synthesis is a central focus of Iteration 3.

Chapter 7

Iteration 3

7.1 Aims

The third and final iteration of this study aims to make significant upgrades across the application, guided both by literature and findings from past evaluations, particularly the second. The main objectives are listed below:

- A generic framework for the simultaneous connection, reception, processing and logging of signals from multiple M5Stack sensors.
- User-flexible functionality for discrete static balance feedback based on two-dimensional trunk position (MLat/APos).
- Upgrade of gait-related MP's to accommodate bilateral sensors and provide improved heel-strike detection.
- Upgrade of music encoding functionality to allow the use of either multitrack MIDI files or the original CMR representation. Applies to both melody and percussion.
- Upgrade of music synthesis functionality to create multiple sonic styles with a combination of instrumentation variants and mixer settings.
- Added sonic interaction possibilities for sit-to-stand and dynamic balance.
- Updated set of sonification strategies for enhanced perceptual salience.

7.2 Design and Implementation

Figure 7.1 is a high level context diagram depicting the overall system structure of the final version (post-Iteration 3) and highlighting key changes relative to Iteration 2. Of particular note is that the system now accommodates upto three

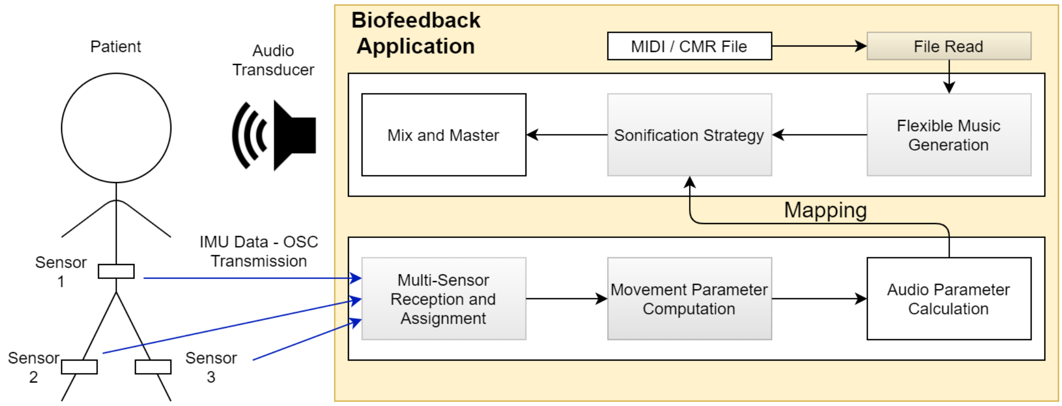

Figure 7.1: A high level schematic of the final system version developed in Iteration 3. The grey boxes indicate the functionalities that undergo a significant overhaul since Iteration 2. Note that the IMU transmission and reception system is now a *multi-sensor* setup, which will be further explained in this chapter.

simultaneous wireless sensors and significantly overhauls multiple functionality areas including movement parameter (MP) computation, music generation and sonification. First, an overview of the final possible sonic interactions is provided, followed by the remainder of current version upgrades.

7.2.1 Sonic Interaction Set - v3

The final set of sonic interactions is discussed here, with specifics outlined in Table 7.1.

- Static Upright Balance - Trunk Orientation - Negative Reinforcement: (video demo: 3.2.1 in Media Links) The principle is to reward good upright posture with normal pleasant-sounding music, while discouraging postural deviations through proportional disturbances in the music. The 2D Projection MP in combination with any punishment strategy is most suited. Zone size adjustments and calibration can be used to tailor the interaction to different individual abilities.

- Dynamic Trunk Control - Trunk Orientation - Positive Reinforcement: (video demo: 3.2.2 in Media Links) The ability to move the 2D projection reference frame in real-time can be used to train dynamic trunk control. There are two interaction possibilities, both based on the task of finding Zone A and maintaining trunk position (and thereby pleasant music qualities):

  - Find and hold position: The therapist can offset the reference frame, moving the center of Zone A to a non-upright position (see Fig. 7.4) The

Sr. No	Activity	Interaction	Movement Parameter	Auditory Feedback Strategy
1	Static Upright Balance	Trunk Orientation - Negative Reinforcement of Postural Deviations	Trunk Angle - 2D Projection	- Cartoon Effect - Melody Ring Modulator - Music Stop - Ambulance Siren - Melody Detune
2	Dynamic Trunk Control	Trunk Orientation - Positive Reinforcement of 1) Finding and holding position 2) Following dynamic trajectory	Trunk Angle - 2D Projection	- Instrumentation - Music Stop - Melody Detune etc.
3	Sit-to-Stand	Negative Reinforcement - Movement Jerkiness	Scalar Jerk	- Melody Tonic - Pitched Wave Disturbance - Noise Disturbance
4	Sit-to-Stand	Movement Cues - Stand or Sit	AP Trunk Angle - STS Cue Trigger	- Bell Cue - Wah Wah
5	Rhythmic Gait	1) Negative Reinforcement - Gait Period 2) Positive Reinforcement - Gait Phase	1) Step Periodicity Feature 2) HS Trigger	- Pitched Wave Disturbance - Drum Trigger

Table 7.1: An overview of the activity-wise interaction possibilities in Iteration 3, along with relevant MP and AP choices in each case. Listed AP's are only tested examples, and do not cover all possibilities.

goal of the patient is to move their trunk, find and maintain this desired position using the music feedback.

  – Follow dynamic trajectory: The therapist *continuously* moves the reference frame in the front-back or left-right direction. This essentially creates a desired *trajectory* that the patient must follow to maintain pleasant qualities of the music. Such an interaction can be audio-only or can accompany a visual display (the visualizer).

- Sit-to-Stand - Jerkiness - Negative Reinforcement: (video demo: 3.2.3 in Media Links) This is the same as the corresponding interaction in previous iterations, only with augmented feedback strategies for greater perceptual salience.

- Sit-to-Stand - AP Trunk Angle - Movement Cues: (video demo: 3.2.4 in Media Links) This provides the patient with a sit/stand cue based on anterior trunk bend angle. This combines the STS Cue Trigger MP with a preferably *neutral* sounding feedback strategy as a cue to sit or stand. The cue may either be transient (bell) or steady-state. Cueing angles can be adjusted to suit the patient's needs.

- Rhythmic Gait - Periodicity and Phase Feedback: (video demo: 3.2.5 in Media Links) Same interactions as in Iteration 2, only using the improved bilateral heel strike detection MPs, converting stride time feedback to step time feedback.

7.2.2 Movement Sonification Strategies for Real-time Feedback - Set 3

Based on feedback obtained during the evaluation of Iteration 2, and in accordance with the goal of discretizing feedback intensity, a final strategy set is designed and developed. It consists of strategies that are either completely new or revisions of existing ones. To recap, all AP's take values between 0 and 1, where 0 represents zero sonification intensity and 1 represents the maximum. Some strategies are designed to have discrete levels (even if controlled by a continuous MP quantity), and continuous strategies can be discretized if controlled by discrete MP's. The fixed 0-1 range ensures that the two strategy types are usually compatible with both continuous and discrete MP's, and the general philosophy in this iteration is to stick to discrete MP's. Discrete MP levels (upto six in this iteration) are typically coded at pre-defined AP values between 0 and 1, and decoded by the active sonification strategy. Note that certain continuous strategies from Iteration 2 such as *Brightness, Melody Detune - Frequency Distortion, and Cartoon Effect* are retained but used in a discrete fashion. They are categorized on the basis of the feedback level design, which governs their possible use-cases. Note that certain new MP's mentioned here are explained in more detail later in the chapter.

Revisions to Original Continuous Strategies

Synthesis-based Sonifications - Augmentation: Three strategies from Iteration 1 *(Melody Tonic, Pitched Disturbance and Filtered Noise Disturbance)* are revised for greater perceptual impact. While their core operations remain the same, their AP signals are simultaneously mapped to the depth control of a modulated stereo delay on the *drum buss* (not including the bass drum due to its importance as a timing cue). Normally, the modulation depth of this delay is zero which results in a normal drum sound. But if the AP value rapidly changes (as it does when providing jerk-based feedback), the drum timing and pitch get rapidly scrambled by the delay changes. When this effect is combined with the rapid pitch, noise and tonal modulations from the original strategies, the perceptual impact of these strategies is considerably increased.

Melody Ring Modulator - V3: To increase perceptual salience, the melody ring modulator strategy is revised to integrate principles of *Reference and Zoom* suggested by Parseihian et. al. [73]. This essentially involves replicating the sonification strategy across several frequency bands. For the ring modulator, such a replication can be realized by replacing the sine modulator with a waveform that has far more harmonic content - a square wave at the tritone frequency relative to the tonic. The new square wave modulator is pre-processed through an AP-dependent 2nd order low-pass filter (fc = 500 Hz @ AP = 0, fc = 4000 Hz @ AP = 1), meaning that the modulator has progressively increasing harmonic content as the AP increases, magnifying the perceptual impact of the ring modulator effect considerably.

Two Feedback Levels - On and Off

Music Stop: The idea of this discrete strategy is to stop the music from playing if the AP value is non-zero. This works not by muting all tracks, but by multiplying all track trigger signals in FAUST by zero. The effect of this is that the instrument sounds and reverb/echo tails trail away naturally rather than abruptly cutting out. This is more agreeable to listen to, while still perceptible as a stoppage in the music. Note that this does NOT pause the JUCE sequencing operation, so the music does not continue from the exact point it stopped at, when the AP returns to zero.

Drum Trigger - Augmentation: This new drum trigger AP encodes separate triggers for *two* drum tracks, the bass drum and the snare drum in the form of signal impulses with different heights. When this AP is enabled, its default value is 0.5, which tells the respective drum tracks to ignore inputs from the sequencer. The bass drum is triggered by AP spikes at a height of 0.7, and the snare drum at 0.8.

The FAUST *ba.impulsify* function is used to convert these inputs into sharp triggering impulses for the sample players. The samples triggered by this AP are *always* the max velocity samples, and are further amplified by 3dB relative to the regular sequencer-triggered samples to make them more perceptually salient. Note that the triggered samples depend on the *chosen instrument variant for each drum*, and are thus different for each rhythm and user-customizable.

Bell Trigger: This AP signal triggers a bell sound at its rising edges. It is primarily meant for use with the STS Cue trigger MP, but can be used with any MP that has rising edges. The *ba.impulsify* function in FAUST is used to convert the AP signal into triggering impulses for a FAUST *pm.churchbell* physical model. The advantage of using a physical model rather than a sample is that the sound is different each time it is triggered, making for a more organic and natural listening experience. The downside is a slight increase in computational load.

Six Feedback Levels

These strategies are exclusively intended for use with the *Trunk Angle - 2D Projection Zone* MP that also has six discrete feedback zones (A-F). The feedback levels 1-6 correspond to A-F respectively, meaning that feedback intensity increases from Level 1 to Level 6, with directional feedback at Levels 5 and 6 as they represent the zones to the left and right of Zone A.

Ambulance Siren: The principle of this strategy is to provide normal music at Level 1, and replace it by an increasingly intense siren-like sound from Levels 2-6. A siren sound is simulated using a frequency-modulated triangle wave with a sine modulator whose frequency depends on the feedback level. The modulation width remains constant across levels, keeping the character of the siren sound uniform. Increasing the frequency of the siren as the feedback level increases intends to convey an increasing sense of 'urgency'. The scheme is described as follows:

- Level 1: No feedback (normal music)
- Level 2: Music ducked by 50 dB.
- Level 3: Music level ducked, low frequency siren @ 0.15 Hz, center panned.
- Level 4: Music level ducked, medium frequency siren @ 1.4 Hz, center panned.
- Level 5: Music level ducked, high frequency siren at 1.9 Hz, left panned.
- Level 6: Music level ducked, high frequency siren at 1.9 Hz, right panned.

Instrumentation: The idea of this strategy is to provide a full ensemble at Level 1 and strip instruments away from Levels 2-6. This is done by simply muting their busses in a programmed manner, described below:

- Level 1: No feedback (normal music)
- Level 2: Main melody track muted.
- Level 3: All melody and chord tracks muted.
- Level 4: All tracks muted except bass drum (center panned).
- Level 5: All tracks muted except hi-hat (left panned).
- Level 6: All tracks muted except snare drum (right panned).

7.2.3 Multi-sensor Setup

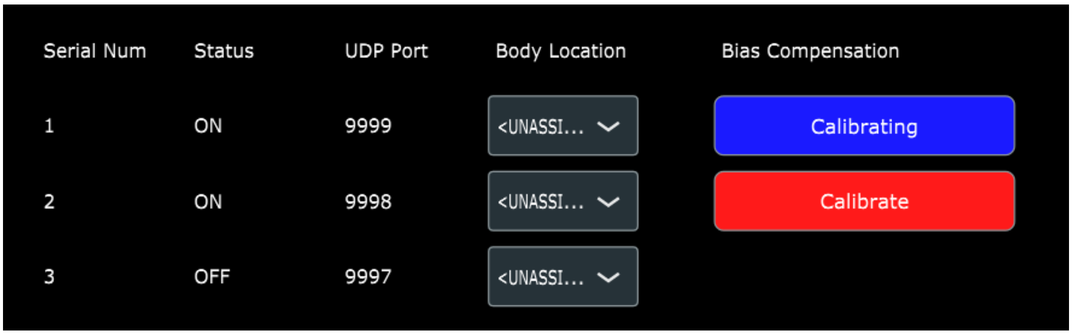

Figure 7.2: Multi-sensor setup section in *Peripherals* tab of biofeedback application. The status of each sensor is updated in real-time and buttons appear to calibrate bias when sensors are active.

Sensor Assignment and Modification of MP Computation

In the second iteration, the sensor-reception framework was written such that IMU data from only one M5Stack sensor could be handled. The MP computation function in *GaitAnalysis* executed regardless of whether the sensor was active or not. Backup sensors had to all be programmed to transmit to the same UDP port, making the measurement of more than one body part difficult. The current iteration introduces of a flexible architecture to allow upto three M5Stack sensors to transmit to dedicated ports. They can be assigned in real-time to one of three locations on the body (trunk, left foot or right foot) or kept unassigned. This is done using a dedicated section in a separate *Peripherals* tab of the UI, depicted in Fig. 7.2. When a sensor is detected to be active, a button appears next to it to allow IMU bias calibration. We now discuss the details of how the setup works.

An additional class *SensorInfo* (refer attached code) is implemented and instantiated in *GaitAnalysis*. This class contains information pertaining to all three sensors, including UDP ports, OSC message headers, online status and body location assignment. Aside from this, a matrix is added to *gaitParamInfo*, to store the required sensor locations for each MP (e.g. a trunk sensor will suffice for measuring static balance, both left and right foot sensors required for gait heel strike). The primary change is how the MP computation function in *GaitAnalysis* now works. At the beginning, it first checks whether there are sensors *online and assigned to all required body locations* for the chosen MP. If this is so, the computation function will proceed to the subroutine for that MP, passing IMU data arrays from the assigned body parts in correct order. The label that simply showed whether one sensor was online in the previous iteration now shows whether the required sensors for the selected MP are online and assigned to the required body parts. If for some reason the required sensors are either offline, unassigned or both, the MP is not computed and the aforementioned label reflects this. Another addition is the ability to separately calculate and compensate for bias in each active sensor. A button to carry out one-time bias compensation is made visible when a sensor comes online, and is explained in more detail below.

IMU Bias Calibration

This iteration introduces functionality for one-time IMU bias compensation. Sensor bias is defined as the average sensor output at zero sensor input [46]. Typical IMU bias can introduce large amounts of errors into calculations involving temporal integration, for e.g. calculating position from acceleration. Bias drifts over time under the influence of several factors, particularly temperature. One-time bias compensation, but has a time-limited effect and only works well for operations operating in stable environments. Periodic compensation is carried out either at regular intervals or on a need basis, for instance after every significant change in IMU temperature [46].

The complementary filter responsible for orientation calculation integrates gyroscope readings to update accelerometer estimates. Gyroscope bias can thus introduce small inaccuracies into the calculated orientation, which can thus benefit from bias compensation. Future measurements of integrated quantities such as velocity will also become far more accurate through bias compensation. The calibration algorithm works on the assumption that the M5Stack is placed in a stationary lying position with the screen side facing upwards. When the calibration process is initiated for one of the sensors, a function in *OSCReceiver_UDPSensor* is called, which records a running mean value for each IMU axis reading of that sensor for a duration of 10 seconds. The button changes to indicate that the calibration process is in progress, and changes colour to tell the user it is complete. The gravitation vector is maintained intact, and mean bias values are stored for each axis. These bias values are subtracted from all new IMU readings. The buttons can be simply be pressed again to recalibrate bias.

An addition was made to the raw sensor logging functionality, specifically to record whether a new OSC packet was received in the most recent sample period. This was achieved by checking for increments in the received packet count during every sensor callback at the receiver end. This is useful to evaluate the wireless transmission channel.

Sensor Sampling Interval Correction

A preliminary analysis of the new raw sensor logs found that with the M5Stack device pausing for 10 ms intervals between loop iterations, there was a large deficit of registered packets at the receiver end. Over 10 second intervals, the percentage of receiver callbacks at which an OSC packet was consistently just under 70%. Missed packets were seen to occur both on a regular (one of every five receiver callbacks) and irregular basis (upto 10 successive receiver callback intervals without a packet received). The irregular UDP packet losses could be attributed either to intermittent WiFi interference or signal strength issues. The regular component was found to be caused by the *delay* function used in Arduino to pause between consecutive transmission loops, due to the non-zero time taken for each loop execution. This led to a longer *true* sampling interval, while the receiver callback operated at a 100 Hz frequency. Although the previous two iterations operated in this way, the effects were likely not felt due to the filtering at the receiver end smoothing out the dropouts.

The sampling interval issue was solved by replacing the *delay(ms)* function in Arduino by the *millis()* function which keeps tracks of *total* time since the program was started, and transmits at fixed predetermined intervals. This change was found to compensate for the loop execution time, and resulted in an improvement in receiver callback packet reception to approximately 89% at a sample interval of 10 ms. The irregular component due to interference and signal strength is more unpredictable, but was solved by simply increasing the transmission rate to a sample rate of 125 Hz while keeping the receiver callback running at 100 Hz. This led to an increase in packet reception to over 98% at close range (< 1m). This was retained as the final approach for this iteration.

7.2.4 MP Additions

Trunk Angle - 2D Projection Zone

Principle: The main purpose of this new MP is to convert the continuous-valued trunk angle measurements into a discrete form, which can easily lend itself to sharply-defined discrete auditory feedback. Secondly, it aims to combine angle measurements in the MLat and APos planes to provide a 2D representation of

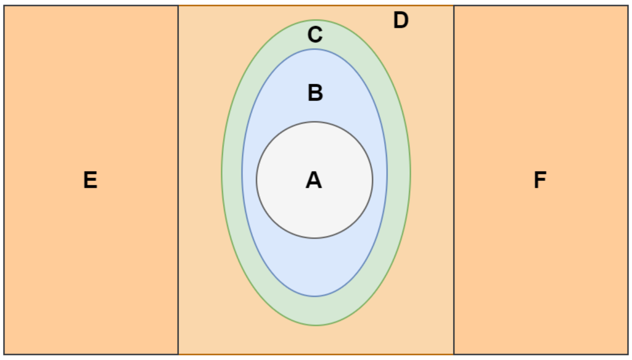

Figure 7.3: A schematic of the trunk angle projection zones, inspired from [17]. It is a top perspective, with the MLat direction along the horizontal axis and the APos direction along the vertical. The center of Zone A corresponds to the upright position.

trunk position that can be used for both static and dynamic balance tasks. The design is closely based on the work of Costantini et al. [17]. In this the human body is modelled as an inverted pendulum, orthogonally projecting the pitch and roll angles of the trunk on the floor. Six discrete zones are designated, where the trunk projection point can lie at any given time (shown in Fig. 7.3. The origin of this plane (rest position) can be calibrated. The goal is to provide distinct feedback in each zone, implying "safety" (Zone A), low-level, mid-level and high-level warnings (Zones B-F). These zones were circular, elliptical and rectangular in shape [17].

Present Adaptation: *The principle of feedback zones used by Costantini et. al. [17] as well as the general shape parameters of the zones are replicated exactly in this MP.* Thus, the calculated MLat and APos angles are input to a function in *GaitAnalysis* in real-time to check which of the six zones the trunk is currently in. The stored MP value is thus a discrete coded value between 0 and 1 that represents the zone index (1 for Zone A, increasing from B-F). Thus, a number of adaptations are made to the original zone scheme:

- Zones can be stretched in both planes to cater to patients with diverse abilities.
- The zone reference frame can be moved in real-time *in both planes* to allow this discrete zone scheme to be used in dynamic trunk activities.
- Zones can be merged together to reduce the number of effective feedback levels, to help patients with perceptual difficulties.

Finally, the reference frame can be calibrated using the functionality from Iteration 2. All these tasks are streamlined into a custom visual interface. This interface provides a real-time 2D visualization of trunk position and present zone, as well

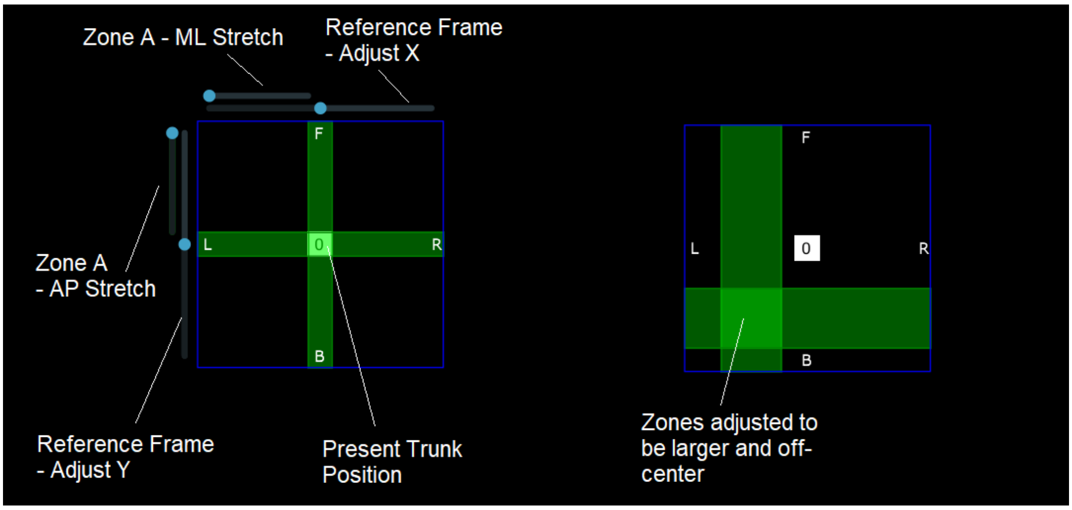

Figure 7.4: The visual interface created in JUCE. Sliders are used to adjust zone size and position parameters. The overlap between the green rectangles is a rectangle tangential to Zone A, while other zones are not shown. The number in the white rectangle indicates the present zone (1-6 for A-F, 0 when inactive).

as controls to stretch the zone shapes in both planes and move the reference frame, all in real-time with the auditory feedback active. The interface is shown and explained in Fig. 7.4. The visualizer is made in JUCE using *Labels*, by manipulating label position, height and width in real-time using the UI callback when it is active. The slider-controlled vertical and horizontal green rectangles respectively indicate the width and height of Zone A, and their intersection is thus a square whose sides are all tangential to Zone A (a circle or ellipse). Note that applying a stretch factor in either plane (front-back or left-right) will proportionally stretch all zone shapes, although the visualizer does not show this. A drop-down list is also provided to merge feedback zones together for steeper or more gradual feedback intensity transitions (feedback slope). Depending on how the various parameters are set, this MP can be used to provide feedback in static sitting, standing or trunk control activities.

AP Trunk Angle - STS Cue Trigger

This MP has the specific purpose of triggering movement cues during the STS activity. It is based on the principle that the act of standing or sitting involves a particular amount of forward trunk flex [9]. The goal of this MP is thus to monitor whether the patient is sitting or standing, and respond in some manner when a sit/stand-specific forward trunk angle threshold is crossed. The M5Stack sensor must therefore be attached to the lower back and assigned to the trunk region, and AP orientation is calculated in the usual fashion. The calculation of this cue

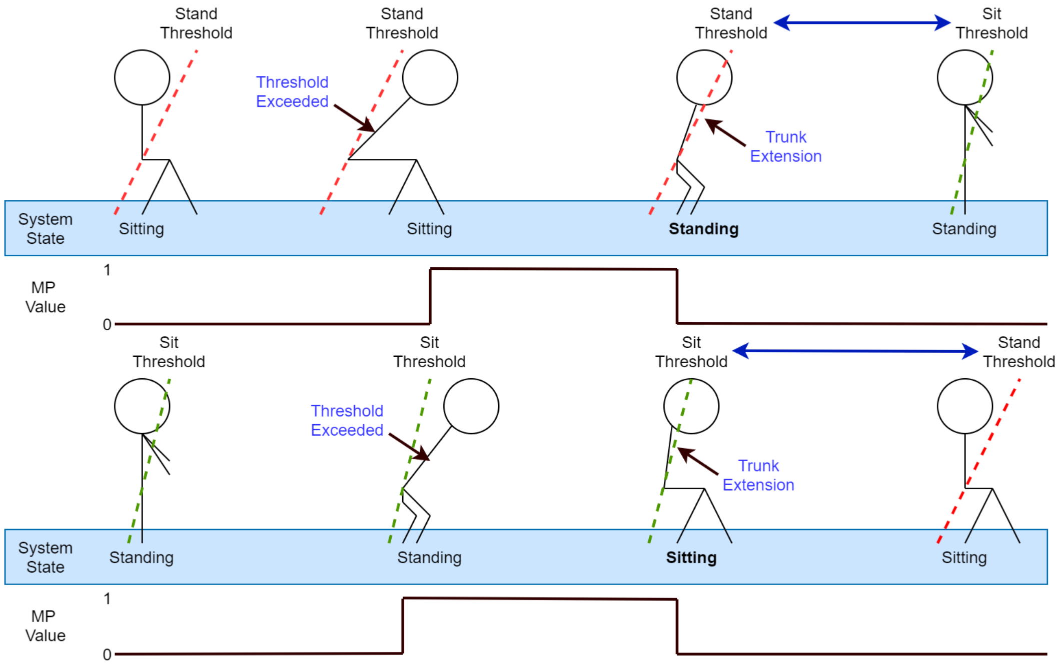

Figure 7.5: Schematic of the STS Cue Trigger MP computation. The MP transitions from 0 to 1 when the cueing angle threshold is crossed, and back to zero when the APos angle goes back under the threshold.

trigger MP begins by smoothing the AP orientation time-series with a 2nd order Butterworth lowpass filter with cutoff frequency at 3 Hz. Cueing angle thresholds for sitting and standing can be set in the user interface when the exercise mode is set to Sit-to-Stand. There is also a toggle-button to initially tell the system whether the patient is sitting or standing. The MP value is zero by default. When the patient flexes his/her trunk forward and the filtered AP angle exceeds the relevant threshold, the MP value is set to 1. It stays at 1 until the trunk is extended and AP angle goes back below the same threshold angle, after which the system assumes that the patient has completed the action and the MP goes back to 0. At this point, the system state flips from sitting to standing or vice versa and the relevant threshold is chosen. This is illustrated in Fig. 7.5. An auditory cue can thus operate in two ways in response to this MP - a cueing sound can be triggered by the rising edge of the MP, or a cueing *state* (e.g. sound effect) can be enabled during the time when the MP value is 1.

Bilateral HS Detection - Impact-Based

The heel-strike detection mechanism in Iteration 3 is changed from unilateral and gyroscope-based to bilateral and accelerometer-based. This means that two separate M5Stack devices are used, transmitting to separate UDP ports and assigned to the respective lower-limb locations through the multi-sensor interface. This shift also necessitates a change in sensor location from the thigh region (just above knee) to the shank region (just above the ankle) and the lengths of the velcro straps are adjusted accordingly. This detection mechanism is far simpler than the gyroscope-based one, and works on the principle that there are large spikes in foot acceleration when the heel strikes the ground. *Note that the gyroscope-based turning detection is still used.* All three accelerometer axes are filtered through fourth order Butterworth high-pass filters with cutoff frequency 0.2 Hz to remove the gravity vector, after which the acceleration norm is calculated as:

$$Acc_{norm} = \sqrt{Acc_X^2 + Acc_Y^2 + Acc_Z^2}$$

Several measures are taken to minimize false heel strike detection. The acceleration norm is constantly monitored for both sensors when the walking activity begins. If the norm for either foot sensor exceeds a threshold (modifiable in the user interface when the Exercise Mode is set to Walking), a heel strike for that foot is registered, and a flag is set to indicate that the next detected heel-strike must come from the opposite foot. All threshold crossings from the same foot until an opposite foot detection are treated as false detections and ignored. When the opposite foot strikes, the flag is reversed and the process repeats itself. Additionally, every valid detection has a time-out period (empirically set at 80% of the music beat period) where all opposite detections are also ignored. This prevents conducted force from one heel strike from triggering the opposite sensor. This algorithm was found to work well in self-tests, although the detection threshold typically requires adjustment for different walking styles (higher for faster or more forceful walking).

The previously described stride periodicity and heel-strike trigger MP's used in Iteration 2 are adapted for bilateral sensor use with the new detection algorithm. The stride periodicity feature is converted to a step periodicity feature, with the reference interval by default being *one* beat interval instead of two in normal time (two in half time). This makes it possible to capture step time asymmetry and, in turn, provide feedback on it. The heel trigger MP is modified to distinguish between left and right heelstrike events by creating impulses of different heights (0.7 and 0.8 for L and R respectively) so they can be recognized in *DspFaust* and used to trigger different drums.

7.2.5 Music Representation Upgrade - Multitrack MIDI

The solution to the previously discussed shortcomings of the CMR music representation proposed is the addition of MIDI support for melody and percussion. The approach used in this iteration is to adapt the sequencing routines to allow either CSV or MIDI playback modes and use the same file-picker to load files of either type. If a valid MIDI file is selected, the contents of the file are loaded entirely in memory prior to playback, and a special set of functions is written to track musical time and handle new MIDI events. Percussion MIDI for the different rhythms, on the other hand, is pre-encoded and stored in a local folder and populated into lists when the application is run.

Melody Tracks

Complete melody information (bassline, chords, main melody) is stored in the form of multitrack (Type 1) MIDI notation files, which can now be loaded by the biofeedback application.

File Creation in REAPER First off, the MIDI files must be created in a certain manner with no more than three tracks, which respectively correspond to the main melody, main chord synth and bassline. Note that the main chord synth and bassline root-notes are now independent of one another. An easy method to create these files is using a digital audio workstation (Cockos REAPER used in this case). Three tracks can be created in REAPER and MIDI objects can be drawn on each one, corresponding to the contents of each of the three mentioned tracks. MIDI data can easily be duplicated, transposed, time-stretched, humanized and previewed in REAPER and this makes the encoding process far simpler, yet more powerful. Once completed, the file can be exported as a multitrack (Type 1) MIDI file in the REAPER file menu. Information on the time signature and rhythm type (straight/shuffle) of the music piece must be encoded into the file name, so the appropriate rhythms can be selected for it when the file is loaded by the biofeedback application.

File Reading in Biofeedback Application A toggle button is provided to choose between MIDI and CMR file mode. Multiple JUCE classes are used in the file reading process. If MIDI mode is selected, the *FileChooser* object used for file browsing will only show '.mid' files instead of '.csv'. The entire MIDI read process is incorporated into the custom *MusicInfoRead* class written for CSV reading. A JUCE *MidiFile* object is created as a member of *MusicInfoRead*. A custom helper class *MidiTrack* is also written to store MIDI information pertaining to each track. *MidiTrack* is included in *MusicInfoRead* and an empty *MidiTrack* array with 4 elements is created as a member (refer attached code).

When a MIDI file is chosen, a new function *loadMidiFile* in *MusicInfoRead* is called. The MIDI file name is searched for specific sub-strings related to the rhythm/time signature (e.g. "3by4"), and the appropriate list of rhythms is made available for *this music piece*, depending on the information found (more on this in the next subsection). The file path is used to create and initialize a JUCE *File* object, from which a JUCE *FileInputStream* object is, in turn, created and initialized. If the

FileInputStream is successfully created, the empty *MidiFile* object calls its *readFrom* method to read the file contents. JUCE methods are used to fetch and store the number of tracks in the MIDI file as well as its time format (number of ticks per quarter note). At this point, an empty JUCE *MidiMessage* object is created. Next, the MIDI message information for all tracks must be carried out. The procedure for each track is outlined as follows:

- The MIDI info matrix of the respective *MidiTrack* object is flushed with zeros.
- The track is read from the JUCE *MidiFile* object, yielding a *MidiMessageSequence* pointer reflecting the track information.
- JUCE functions fetch the number of MIDI events in the track as well as the final message timestamp.
- With the number of events and a *MidiMessageSequence* for the track in hand, the following process obtains and stores the MIDI messages in the *MidiTrack* object. This is conducted for *each event* in the sequence:
  - A pointer to the event is fetched using the *getEventPointer* method and stored in a JUCE *MidiEventHolder* object, from which the actual MIDI message can finally be accessed.
  - A check is performed using JUCE methods to find out whether the MIDI message is 'Note On' or 'Note Off', depending on which it is assigned an integer code 1 or 2. The event is stored at its index position in the main *Info* matrix in its respective *MidiTrack* object as a matrix row along with its note number, velocity and timestamp.
  - The event counter is incremented, and the operation is complete when all events have been stored.
- The above procedure is repeated until all events from all tracks are read and stored in simple C++ matrices in the *MidiTrack* object array, which can now be accessed with ease by the sequencer.

Percussion Loops

Velocity and timing information for different rhythms in different categories is locally stored in the form of single-track (Type 0) MIDI notation files, which are automatically pre-populated when the application is started up.

File Creation in REAPER MIDI files are created for each rhythm, with a single track containing a MIDI item no longer than four bars (one melodic measure). The four percussion components (bass drum, snare drum, hi-hat and crash cymbal) are represented by pre-defined note numbers. These numbers are chosen to coincide with standard MIDI drum conventions (as used by VST instruments like EZDrummer 2) so that the rhythms can be previewed in real time before they are exported. The freedom of MIDI allows rhythms to have more complex timing variations, shorter note intervals and timing humanization. The files must be exported to a particular directory with a specific naming convention, with the name of the rhythm and its time signature/timing mode mentioned in the file name.

File Reading in Biofeedback Application The loading of MIDI percussion loops is carried out at application startup. Using functions of the JUCE *File* class, the defined directory containing the grooves is searched for MIDI files. The filename of every file found is checked for sub-strings related to name and timing information, and the file is categorized accordingly. The information in the file is then loaded in an identical fashion to the melody files, to an element of an array of modified *MidiTrack* objects that have smaller MIDI info matrices. In this fashion, all files are loaded and categorized into straight 4/4, triplet 4/4, or 3/4 time. Note that the application does *not* know how many rhythms to expect and loads all available files (maximum 30), automatically populating the lists of available rhythms in each timing category (refer attached code).

Clocking and Sequencing Modifications for MIDI Playback

With melody/harmony encoding now possible and readable in MIDI, the sequencer is modified to play this information back while still supporting the old CMR format. The CMR data need only be fetched at sixteenth note pulse instants, while MIDI events can occur at any time in a file and must be continuously monitored. The CMR format encodes note number information and handles voicing implicitly, while it must be manually handled in MIDI. Sixteenth note triggering of CMR data simply depends on the tempo-dependent inter-pulse interval in *milliseconds*, while the timing metric in MIDI is *ticks*, whose temporal density can vary among MIDI files. The operator can switch between MIDI and CMR modes with a Toggle Button on the Music Playback Control Tab of the application, depending on which files of each type may be browsed for and loaded.

Clocking in MIDI Mode: The key idea of the MIDI clocking is to find the number of *ticks per quarter note* from the MIDI file header, and use this along with the configured tempo to compute the number of MIDI *ticks per millisecond*. This can be used to increment an *elapsed MIDI ticks* counter, which can in turn be applied to handle MIDI events in the melody and percussion files at tempo-dependent rates for correct playback.

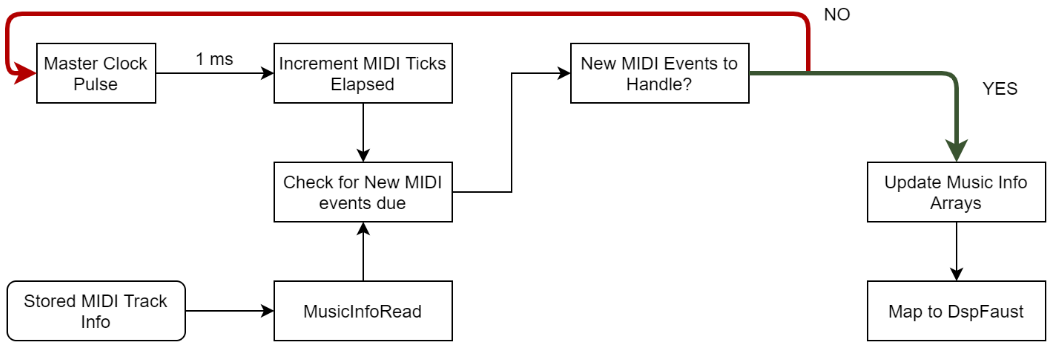

Figure 7.6: Simplified flow diagram of MIDI event handling for a single instrument track.

Real-time Event Handling The first step after a MIDI music file is loaded is to check its time signature and timing mode, and modify the list of available rhythms in the UI accordingly. The first rhythm of each list is chosen by default, and the timing mode is set to straight 4/4 if the MIDI file has no timing information in its filename. A custom function is added to the *Sequencer* class to fetch and handle music information from the *MidiTrack* matrices in real-time. This is relatively complex compared to simply fetching indexed integers from the *MusicInfoRead* matrices as the following must be considered:

- MIDI events do not necessarily occur only at 16th note pulses (due to note off events, humanization, etc.) and new events must therefore be checked for at the finest time resolution possible in the main callback (1ms).
- Depending on the tempo, the number of elapsed MIDI ticks from the clock may not exactly coincide with the timestamps in the loaded MIDI files, due to rounding errors and limited floating point precision.
- Multiple simultaneous events may need to be handled.

If MIDI mode is enabled, the main callback requests the sequencer function to check for new MIDI messages at every clock interval, and handles new messages by modifying the content of the instrument information arrays. Here is a step-by-step explanation of how the function works for melody information (refer attached code):

- The main callback calls the sequencer function, providing as arguments the track index, number of voices, MIDI ticks elapsed, MIDI ticks per millisecond and a pointer to the music info array to be modified. A flag indicating whether to ignore velocity information is also provided, in case of chord and bassline tracks which have their own style-defined velocity patterns from *PercPatternLibrary*.

- The function deduces the 'present interval' as *(MIDI Ticks Elapsed - MIDI Ticks per ms)*. It then iterates between the next unhandled MIDI event and the last MIDI event in the *MidiTrack matrix* for that track, and calculates the number of MIDI events lying in the present interval that need to be handled.
- If there are no events to handle, the function returns a Boolean *false*.
- For each of the events to be handled within the present interval, the function does the following:
  - If the event is a *Note On*, the respective music info array is updated with the MIDI key number and velocity (depending on velocity flag status for the track). Music info arrays are of differing length depending on the number of voices (e.g. one voice for main melody but four for chords) and every new note on event is written cyclically at the next array index so simultaneous events do not overwrite each other. Velocity, originally an integer between 0 and 127, is normalized as a floating point number between 0 and 10.
  - If the event is a *Note Off*, it is ignored if the velocity flag is false. If it is true, the velocity at the respective voice location in the info array is set to zero.
- The number of MIDI events handled is incremented, and the function returns a Boolean *true*.

This new event check is carried out for the main melody, chord and bassline tracks. If new events were handled in any of them during a callback, then the updated array values are immediately mapped to their respective FAUST controls after key-number restriction (if applicable to the track). The process for MIDI percussion event handling is nearly identical (refer attached code). The key difference is that the percussion file is *looped* for the duration of the melody file, by resetting the elapsed MIDI ticks to zero at the end of every four bars. The note number of every event in the selected MIDI rhythm is used to map the event to the intended *DspFaust* control. A simplified schematic of the process is shown in Figure 7.6.

MIDI-related Modifications - Song Progress Bar and Remaining Time

The addition of the MIDI mode also means that the song time displays need to be handled accordingly. In MIDI, the file length is not fixed like in CMR mode, so the song progress cannot be calculated the same way. However, the timestamp of the *final MIDI event* in the file is obtained when the file is first read, and essentially represents the duration of the file in ticks. The number of elapsed MIDI ticks is simply divided by the total duration in ticks to yield the song progress. As elapsed ticks update themselves in every timer callback as opposed to only sixteenth note intervals, the progress bar updates in a smoother fashion than in CMR mode. A similar logic is employed for remaining time, which is derived from the final timestamp, number of ticks presently elapsed and the tempo-dependent number of ticks per millisecond. The use of double precision variables here minimizes rounding errors.

7.2.6 CMR Playback Upgrade - Timing Modes

Triplet and shuffle timing are also provided in CMR *(Compact Music Representation)* mode, through modifications made to the counter incrementing function of the *Sequencer*. First, two toggle buttons are provided on the music playback control tab, to set flags that enable or disable triplet timing mode and 3/4 mode respectively.

Triplet Timing: When triplet timing is enabled, the *fourth* sixteenth note in every quarter note is *skipped*. This results in a perceived transformation from a quarter note comprising *four straight sixteenth notes* to *three triplet eighths* at a tempo that is 1.33 (4/3) times higher. The tempo is internally slowed by a factor of 0.75 to compensate for this increase. As the fourth sixteenth note position of a beat is relatively seldom played except in syncopated grooves, the originally straight 4/4 rhythms are transformed quite seamlessly into triplet 4/4 rhythms. In CMR mode, the same holds for melody content, and triplet mode can be switched on and off during playback without any loss of synchronization as all instruments refer to the same counters.

3/4 Time: This is relatively simple - the fourth beat of every bar is skipped by the sequencer. The results are not as seamless as with triplet timing, as the drum fills in the final beat of a bar get truncated, but the results are still musically valid.

7.2.7 Music Generation Upgrades

Integration of Pre-Rendered Drum Samples

The functionality for the integration of drum samples is implemented for a number of reasons. First, it is easy to modify or change drum sounds by simply replacing audio files. Secondly, recorded real drums can sound far more realistic than their artificial simulations at a fraction of the computational cost. Third, the use of multiple drum samples per percussion track affords far more articulation possibilities.

Natively, FAUST allows local audio files to be used by *dsp* scripts using the *soundfile* primitive, which allows multiple mono or multichannel local sound files to be loaded into memory. Their paths must be specified in a specific syntax, and a read index addresses the sample index of the file to control its resulting audio stream, which can be processed just like any other signal in FAUST. The *-soundfile* command must be used while compiling the script using Faust2Api in Ubuntu, to include the necessary libraries for audio reading and playback. Although this is simple in principle, the *-soundfile* command was found to be incompatible with the Faust2Api *-juce* command at the time of initial testing, and the resulting *DspFaust* class failed to compile on C++. This was reported to members of the core FAUST development team, who updated the FAUST compiler itself to fix the issue. After 2-3 rounds of back-and-forth testing, the *-soundfile* containing *DspFaust* class successfully compiled in Visual Studio. [1]

With the FAUST functionality working, the next step was to craft the appropriate percussion samples for each instrument, in three variants. This was done in REAPER, using licensed copies of VSTi's Toontrack EZDrummer 2 and Spectrasonics Omnisphere 2 as high quality sources of real and electronic drum sounds respectively. Any default ambience effects such as reverberation or stereo width enhancement were disabled, and the VSTi outputs were serially treated with dynamics and spectral processing using stock REAPER plugins ReaComp and ReaEQ respectively. Although the exact processing parameters varied among the chosen source sounds, the general practice was to apply slow-attack, slow-release compression to emphasize drum transients and filter inaudible low frequencies and undesired resonances.

Three sample variants were thus created for the bass drum, snare drum, hi-hat and crash cymbals. For the bass drum and crash cymbal, only one drum sample file was created per variant. But for the more expressive snare drum and hi-hat cymbal, three percussion samples were created per variant to create articulation possibilities in the 1-9 velocity range. This range is further subdivided to trigger one of the three samples from velocity 1-3, 4-6 and 7-9 at proportional loudness levels. The maximum loudness in this case occurs at multiples of three, and the custom velocity function handles this in a linear fashion. Drum Samples for the three velocity sub-ranges were varied appropriately in terms of envelope characteristics and loudness. All percussion samples are mono files and either 1 or 2 seconds in length depending on the length of the sound. The samples are rendered at a sample rate of 48 KHz, which is the same as the sample rate used for the remainder of audio synthesis. At this time, no measures have been taken to handle multiple sampling rates. They are all faded out to zero, so that there is no DC offset present when the read index reaches and remains at the final sample value. An example of the bass drum is taken here, where two drum samples are loaded into FAUST as in the following FAUST code:

```
K_FILES = soundfile("K_SMPL[url:{'D:\\GaitSonification\\Drum Sample
```

[1] At the time of writing, however, there is still a bug in the FAUST parser that prevents '\\' in file paths from being retained in the *DspFaust* class, and these must be input manually into the file prior to final C++ compilation.

```
s\\Final Library\\K_V1.wav'; 'D:\\GaitSonification\\Drum Samples\\
Final Library\\K_V2.wav'}]",1) : !,!,_;
```

The two files are then isolated into playable functions as follows, where *i* is the file read index:

```
K_SMPL_V1(i) =   0,i : K_FILES;
K_SMPL_V2(i) =   1,i : K_FILES;
```

Sample playback is achieved by manipulating the read index of the audio sample corresponding to the selected instrument variant. When the instrument is triggered, the read index must snap to zero and increment at every audio sample interval until the zero sample value at end of the file is reached. This is practically achieved by modulating the read index of the file using a standard FAUST function *ba.countup*, a counter which resets to zero when its trigger argument is 1, and increments to a preset maximum at a rate equal to the audio sampling rate. Thus, if the drum trigger from the sequencer is used to trigger the *countup* function and the file length is used as the preset maximum, the file can be played back in real-time when triggered and processed by the subsequent signal chain of the track. Thus, a *samplePlayer* function is written to play a file from a *fileFunc* like K_FILES above:

```
samplePlayer(fileFunc,trigger) = fileFunc(ba.countup(96000,trigger));
```

And finally, this is used to play back the drum sample as follows (TRG_K is the drum triggering signal):

```
kick_V1 = samplePlayer(K_SMPL_V1,TRG_K);
kick_V2 = samplePlayer(K_SMPL_V2,TRG_K);
```

[2]

Instrument Variants and Music Style Presets

The second iteration expanded upon the original idea of providing multiple percussive rhythms by adding rhythm-specific chord arpeggio/bassline patterns and mixer gain settings. This is further enriched by creating different instrumentation configurations for each rhythm, so that the rhythm options differ from each other not only in terms of the percussive pattern, but also the *style* of music. As there are

[2]It must be noted that the *soundfile* functionality used here is at an experimental stage in FAUST and that several problems exist, mainly that loading too many files causes unpredictable playback problems. Also, the defined file paths are absolute and thus require the files to be present at those exact locations on the local computer. It is hoped that future versions of the FAUST compiler will address some of these issues, but the current implementation provides a working solution in the current development.

several rhythms (10 - 13), it is not possible for the FAUST compiler to handle eight unique synthesis methods for each rhythm. Instead, a simple structure is created that allows up to three instrument *variants* for each of the eight tracks. The variants are designed such that:

- their timbres fit the musical role of the track.
- for a single track, they have enough timbral diversity to be used in different music styles.
- it is possible to create multiple sonically balanced combinations of variants to simulate a palette of music styles.
- percussion tracks are sample-based and melody tracks are synthesized.
- all instrument variants across tracks are computationally light.

Assignment to Rhythms and Real-time Manipulation The single synthesis algorithm per track is replaced by three per track, of which one is selected at run-time depending on the selected variant for that track. Variant selection is dynamic, and a set of numerical entry controls is added for JUCE access that allow the selected variant for every instrument to be changed in real-time. The *ba.selectn* function is used to select between variant audio signals, and is optimized such that only the selected variant is synthesized. The remainder of the signal chain after the variant signal selection is identical, hence effect-based or additional synthesizers sonification strategies are not affected.

Every rhythm is assigned a preset configuration of instrument variants for all eight tracks, which was honed by ear and is stored in *mixerSettings*. In addition to this, an array of drop down *ComboBox* objects is added to the music control tab, serially mapped to the variant selector controls of *DspFaust* and allows the user to change individual instrument variants in real-time. The drop-down lists for each track are named appropriately to facilitate the task of locating specific timbres or samples. Each variant has separate EQ, compression, gain and note limit settings, all of which are fine-tuned by ear. Every time a rhythm is toggled, the variant choices, mixer settings and UI controls are automatically updated (refer attached code). Details of the synthesis techniques and sample used for each instrument variant are provided in Appendix C.

Chapter 8

System Technical Evaluation

The first part of the evaluation process focuses exclusively on various important technical parameters of the system, namely effective sensor range, biofeedback loop delay and computational performance. This chapter presents an array of experiments to assess each of these parameters, and the obtained results are discussed in Chapter 10.

Test Setup: In all experiments, the biofeedback application was run on a Dell Inspiron 15 7000 Windows laptop with an i7 processor and 8 GB RAM running at 1.8 GHz (4 logical cores). A USB-connected Focusrite 18i8 audio interface was used for audio output, which was auditioned using a wired Logitech speaker.

8.1 Sensor Range

The purpose of this test was to study the effect of sensor distance on packet reception efficiency in an indoor environment. The percentage of received OSC packets in a short time interval serves as a good indicator of useful sensor range, providing valuable reference information when using the equipment in real-life environments (e.g. large wards). It is acknowledged that WiFi interference and other signal obstacles vary considerably among buildings and indoor locations, but this evaluation was restricted to a single indoor environment where all factors but sensor distance were kept constant for the experiment duration.

8.1.1 Experiment

Setup: The biofeedback application was setup on a Dell laptop placed in the corner of a large furnished room. The laptop itself received its internet connection from a mobile phone hotspot. This connection was shared with a single M5Stack

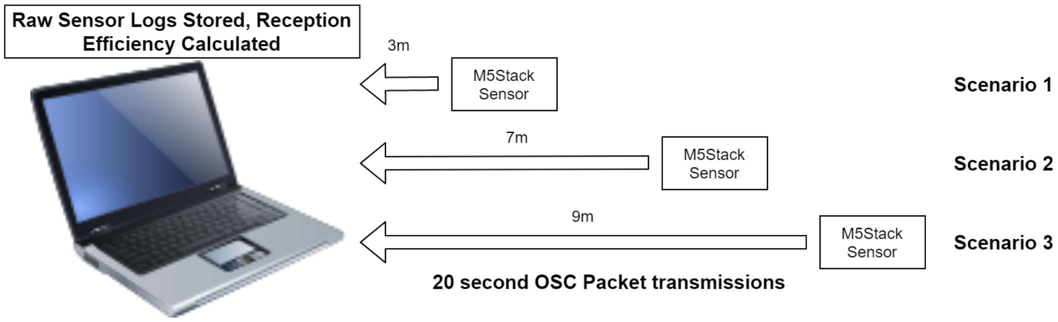

Figure 8.1: Depiction of the range measurement procedure.

sensor, which transmitted data packets over it at a sampling rate of 125 Hz, while the receiver callback in the application operated at 100 Hz.

Procedure: Three sensor distances were chosen based on the ergonomics of the selected room:

- **Scenario 1:** 3 meters, direct line-of-sight between sensor and laptop.
- **Scenario 2:** 7 meters, direct line-of-sight between sensor and laptop.
- **Scenario 3:** 9 meters direct distance, but with a wall corner impeding the direct line-of-sight. (WORST CASE SCENARIO)

The sensor was placed on a chair at these measured distances, and raw sensor data logs were stored over a 20 second duration for each location. The procedure is shown in Fig. 8.1.

8.1.2 Results

The percentage of receiver callbacks with new OSC packets was calculated from the logs for each distance. The results are as follows:

- **Scenario 1:** 96.35%
- **Scenario 2:** 96.10%
- **Scenario 3:** 82.5%

8.2 Biofeedback Loop Delay

The purpose of this series of tests was to measure the loop delay of the system (time interval between movement and its corresponding auditory feedback) during each of the main interaction types (angle-based, jerk-based and heel-strike based). Caveats of this testing must be highlighted in advance. The first is the level of obtainable measurement accuracy. A challenge is that in most interactions, the auditory feedback is merged with the music ensemble, which means that there can be significant masking between feedback and music, making it difficult to pinpoint the exact feedback onset instant, even through waveform or spectrogram inspection. The second is that two of the three analyses were video-based for finding movement onsets, and the video framerate was only 23.91 fps, due to a lack of higher quality equipment. A framewise analysis could therefore lead to a maximum positive measurement error of 41.8 ms in these cases. Identifying the precise onsets of jerky movements and exact moments of angle threshold crossing from video is also subject to imprecision due to an inability to identically replicate such movements or visually inspect angle thresholds. Next, these measurements only yield the *total* loop delay and not the delays incurred by individual operations (wireless sensor transmission, MP computation, AP mapping and audio output buffer). Lastly, the temporal comparison between movement/audio is done with audio captured from the WASAPI driver (Windows Audio Stack), and thus does not take into account the sound propagation delays from the computer audio output to the patient. All measurements were performed on myself in a small square room.

8.2.1 Experiment

Experimental Setup: The setups for heel strike, jerk and trunk angle feedback latency measurement were distinct. Audio feedback strategies with the most salient perceptual onsets were chosen in each case to improve the accuracy of onset instant identification during data analysis.

- **Heel Strike:** The heel strike drum trigger sonification strategy was chosen in the biofeedback application for this measurement, with two separate foot-mounted M5Stack sensors. A handheld mobile recorder was used to record the sound of the feet physically striking the floor, while a simultaneous mono recorder track in REAPER was set up to capture the triggered drum sounds.

- **Jerk/Trunk Angle:** The pitched disturbance sonification strategy was used for jerk, and the discrete projection zone-based ring modulator strategy was used for trunk angle. In both cases, a single M5Stack sensor was mounted to my lower back region, and a mobile camera was used to capture trunk movement videos from the frontal direction. A similar recorder was set up in REAPER to capture the application audio output. The mediolateral trunk angle threshold was kept at 1 degree in either direction, and the jerk target threshold was kept at the lowest possible value to prevent rest triggering.

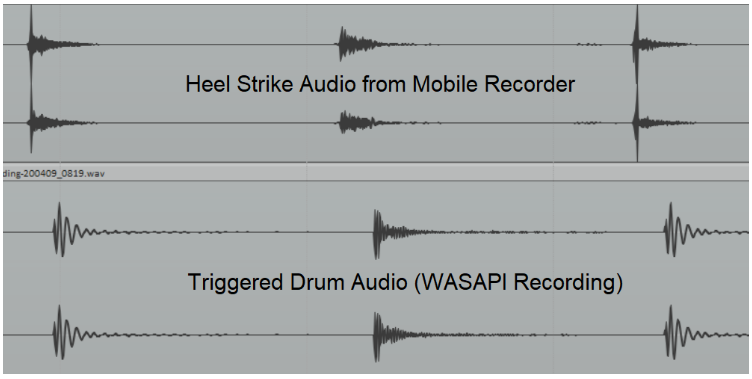

Figure 8.2: Direct heel-strike audio and triggered drum biofeedback tracks in REAPER. The tab-to-transient function was used to locate the exact onset timestamps and calculate average loop delay over 19 repetitions.

Procedure: The REAPER recorder and phone recorder simultaneously recorded a sine burst to synchronize biofeedback and phone recordings for subsequent analysis. The individual procedures are as follows:

- **Heel Strike:** The feedback loudspeaker was thereafter muted, and 19 steps were taken around the room, while the phone recorder captured physical heelstrike events as audio signals, and the REAPER recorder captured the triggered drum output of the biofeedback application.
- **Jerk:** The feedback loudspeaker was thereafter muted, and 13 repetitions of sudden jerky movements separated by still pauses were recorded by the camera. Simultaneously, the biofeedback application output was recorded in REAPER.
- **Trunk Angle:** The REAPER recorder and phone recorder simultaneously recorded a sine burst for recording synchronization purposes. The feedback loudspeaker was thereafter muted, and 14 repetitions of sideways trunk tilts separated by upright rests were recorded by the camera. Simultaneously, the biofeedback application output was recorded in REAPER.

Data Analysis: In all three cases, movement onsets and feedback onsets were manually annotated from the respective audio and video recordings, after which corresponding onsets were compared and averaged to get mean and standard deviation loop delay measures.

- **Heel Strike:** Corresponding physical step onsets and triggered drum onsets were both identified using the tab-to-transient feature in REAPER. This is shown in Fig. 8.2.

- **Jerk:** Jerky movement onsets in the video recording were identified by a frame-by-frame analysis. As jerky movements were separated by still pauses, frames containing the initiation of jerky movement were easy to identify visually. In the biofeedback output recording, the onset instants of auditory feedback were identified both by audition and waveform inspection.

- **Trunk Angle:** Trunk tilt initiation instants were identified by a manual frame-by-frame analysis. As trunk tilts were separated by upright pauses, frames containing tilt initiations were easy to identify visually, although it was not possible to identify the mediolateral angle threshold crossing instants. As the threshold was low (1 degree), a minimal delay between movement initiation and threshold crossing was assumed. In the biofeedback output recording, the onset instants of auditory feedback were identified both by audition and waveform inspection.

8.2.2 Results

The results of the data analysis are shown in Table 8.1. Heel strike and jerk show nearly identical loop delay values, but trunk angle delay measurements are considerably higher, with greater variance.

Movement Parameter Type	# Repetitions	Loop Delay (ms)
Heel Strike	19	93 (48)
Jerk	14	93 (37)
Trunk Angle	13	300 (90)

Table 8.1: The results of the loop delay data analysis, with delay in each case shown in milliseconds as Mean (STD value).

8.3 Computational Load

The computational load incurred by the biofeedback application during an activity is an indicator of the efficiency of the overall program. This is important in determining the types of computer systems the application is capable of running on in real-time, as well as the potential for adding more layers of complexity to the music synthesis and sonification structure. For such an evaluation, it is logical to record the load of the application in the most computationally 'stressful' situation possible. It was difficult to predict or calculate what application configuration would correspond to this, due to the number of parameters and variables involved. Therefore, the overall computational load testing was done in three steps - first, CPU load in a number of music-only scenarios was measured. Next, CPU load was measured during sensor reception, MP calculation and logging for a number of MPs *in the absence of music*. Finally, the most computationally heavy scenarios from both the above tests were combined, and the overall computational load was measured.

8.3.1 Measuring Computational Load

A standard metric for computational load of a program is % Processor Time. For Windows, this is defined as " the percentage of elapsed time that the processor spends to execute a non-idle thread" [1], in this case the biofeedback application. It is expressed relative to the total available processing capability, i.e. over a baseline of 100% $\times$ no. of logical cores. It was measured in this case using the Windows Performance Monitor (PerfMon), where it is possible to monitor a Windows process and log recorded processor time at 1 sec intervals (fastest possible).

A similar procedure was followed for each test scenario. The application was configured to the desired condition, and PerfMon logging was begun. To compensate for the low processor polling rate and accommodate spikes in processor usage, logs were recorded for a duration of 100 seconds (= 100 entries) in each case. Mean and standard deviation values for processor time were computed in MATLAB.

8.3.2 Test Scenarios and Results

Music-Only

In these scenarios, the IMU sensor was not connected to the application. Music was played back in several of the main rhythm styles (to accommodate synthesis of different instrument variants) in CMR as well as MIDI mode at the tempo extremes (60 BPM and 150 BPM). The test scenarios are listed as follows:

1. Breakbeat - 60 BPM - MIDI
2. Breakbeat - 60 BPM - CMR
3. Breakbeat - 150 BPM - MIDI
4. Breakbeat - 150 BPM - CMR
5. Dance - 60 BPM - MIDI
6. Dance - 60 BPM - CMR
7. Dance - 150 BPM - MIDI
8. Dance - 150 BPM - CMR

[1] Windows Performance Counters Explained. url: http://www.appadmintools.com/documents/windows-performance-counters-explained/

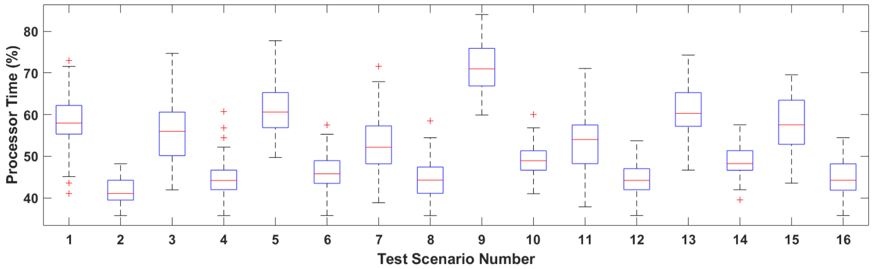

Figure 8.3: Box plot of % Processor Time logs for all 16 test scenarios (the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol.).

9. March 8th Note - 60 BPM - MIDI
10. March 8th Note - 60 BPM - CMR
11. March 8th Note - 150 BPM - MIDI
12. March 8th Note - 150 BPM - CMR
13. Slow Rock - 60 BPM - MIDI
14. Slow Rock - 60 BPM - CMR
15. Slow Rock - 150 BPM - MIDI
16. Slow Rock - 150 BPM - CMR

The recorded logs were imported into MATLAB and the time series of processor time (100 seconds each) were compared using a box plot, shown in Fig. 8.3. Some immediate observations can be made. For all four tested styles, the CMR scenarios show lower % Processor Time than their MIDI counterparts. Additionally, the majority of 60 BPM scenarios showed slightly *higher* mean values than their corresponding 150 BPM scenarios. This is unusual, considering that a higher tempo means the sequencing routines must be executed at higher rates. A third observation is that the scenarios showed considerable variability in second-to-second CPU time values. This is expected as the execution of routines does not occur at a uniform rate, but varies depending on the instantaneous phase of the music. The most 'stressful' is Number 9 (71.13 ± 5.53 %).

MP Measurement + MP/IMU Logging-Only

% Processor Time was next measured in the absence of music, while receiving OSC Data (either one or two sensors), calculating different MPs and logging the

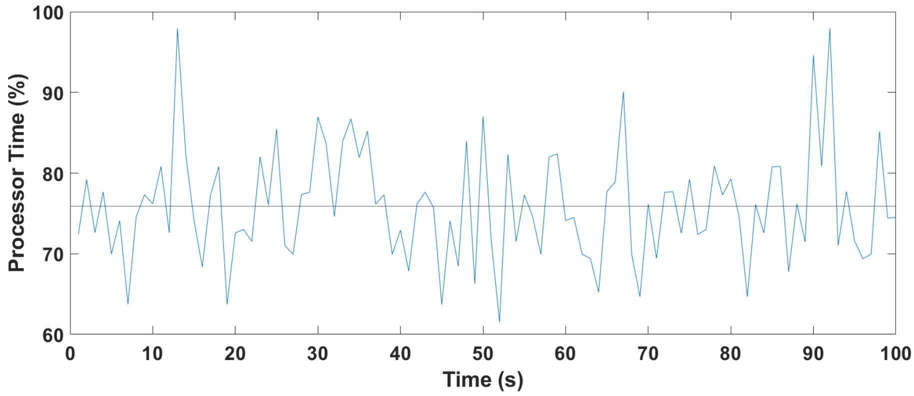

Figure 8.4: Time series of logged % Processor Time during the final computational worst case scenario. The horizontal red line depicts the mean value.

IMU readings and MP values to file. PerfMon logs (100 seconds) were similarly recorded and analyzed, and the results from three scenarios are as follows:

1. **1 Sensor - Trunk Angle - 2D Projection:** 1.23 ± 2.13 %
2. **2 Sensors - Step Periodicity Feature:** 1.35 ± 1.75 %
3. **1 Sensor - Scalar Jerk:** 7.97 ± 2.31 %

It is apparent that Scalar Jerk was the most computationally 'stressful' test scenario of the three. Additionally, it is clear that the non-music related functions of the application are computationally far less demanding than music sequencing and synthesis. This is an expected result, considering that music sequencing (in MIDI mode) and synthesis happen at far higher sample rates.

Final Worse-Case

Scenario 9 from the Music-only tests was combined with Scenario 3 from the Measurement-only tests to create a computational *worse-case* scenario. For this test, % Processor Time was recorded as usual and the Windows Task Manager CPU usage peak as well as memory usage of the process was also monitored. The results over 100 seconds are provided here. Fig. 8.4 depicts the time series of measured % Processor Time.

- % Processor Time: 75.88 ± 6.96 % (Peak: 97.97 %)
- Peak CPU Usage (Task Manager): 26.8%
- Memory Usage: 150.4 MB

The graph shows spiking values with several peaks considerably higher than the mean value. With four logical cores, the Peak % Processor Time and Peak CPU Usage correspond well with each other, although not perfectly as they poll the processor at different instants. No audio dropouts were observed at any moment.

Chapter 9

Final Evaluation

This chapter covers the evaluation of the final sonic interactions and the synthesized music. The former was assessed through structured expert interviews and the latter through an online survey. Findings from both these evaluation procedures as well as the system technical evaluation are ultimately discussed in Chapter 10.

9.1 Expert Interviews - Clinicians

Due to the COVID-19 restrictions in Denmark during the planned final evaluation period with patients, these tests had to be cancelled. Instead, the interactions were evaluated through a series of expert interviews with music therapists and physiotherapists, all conducted remotely. The goal of these interviews was to obtain an expert assessment of the final set of developed sonic interactions, tackling the majority of the aspects that constitute the final problem formulation.

9.1.1 Participants

A total of seven experienced clinicians comprising five neurorehabilitation physiotherapists and two music therapists volunteered themselves for the interview. Of these, two of them had participated in previous iteration evaluations, and the remaining were approached for the first time, either independently or through AAU contacts. In terms of affiliations, one was attached to Neuroenhed Nord, Region Nordjylland, two to Hammel Neurocenter, Region Midtjylland, three to Kokilaben Dhirubai Ambani Hospital, Mumbai (India) and one to UCN.

9.1.2 Setup

All interviews were conducted over video-conferencing platforms such as Skype or Pexip Infinity Connect. Participants with a common affiliation were interviewed in groups, and all participants were hence covered over four sessions. The interactions were demonstrated through a series of videos shared with the participants as YouTube links (see 3.2.1 - 3.2.5 in Media Links). The interviews were recorded with the permission of the participants using REAPER.

9.1.3 Procedure

All participants were provided with a brief information sheet beforehand, with details of the project and the interview questions [1]. At the beginning of the interview, the structure was explained to the participants and they were requested to begin by watching the first sonic interaction video using a pair of headphones. When this was completed, they answered interaction-specific questions (listed in Appendix D) and this process was repeated for all five interaction videos in a fixed order (same as Table 7.1). When the interviews were not one-on-one, the participants preferred to systematically divide the questions among themselves based on their individual areas of expertise.

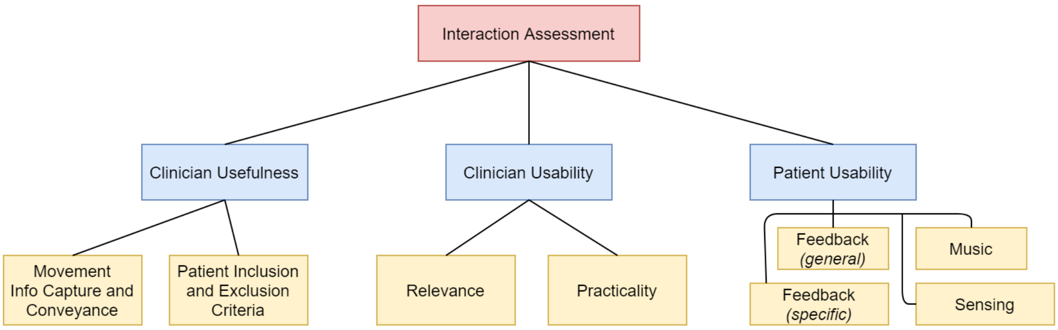

Figure 9.1: A depiction of the hierarchical coding scheme used to analyze the interview data for each of the five interactions. Feedback (general) refers to overall assessments of the auditory feedback (e.g. timeliness, meaningfulness) while Feedback (specific) refers to assessments of individual strategies.

9.1.4 Data Analysis

The interview recordings from REAPER were first transcribed (partially manually and with the help of DeIC Konch [2]). The transcriptions were then coded by an inductive approach into a hierarchical coding scheme, illustrated in Figure 9.1. The three top level categories were *Clinician Usefulness, Clinician Usability and Patient Usability* (refer to Chapter 3 for the definitions of these categories). Within these, codes were assigned to sub-categories based on relevance. Themes were identified

[1] Expert Interview Info Sheet. https://docs.google.com/document/d/1d6xWIIaIwsDlf-NhUBszqJzxnvC9M$_{y}wlfrMFPY4m0/edit?usp = sharing$.

[2] Konch Transcription Platform. https://www.deic.dk/da/konch

based on code incidence for each of the five sonic interaction videos, and these are summarized in the next subsection.

9.1.5 Results

The interview results are analyzed for each of the top-level categories of the coding hierarchy, specifically pertaining to each sonic interaction *(Abbreviations Used: SB = Static Upright Balance, DB = Dynamic Trunk Control, STS-Jerk = Sit-to-Stand (Movement Jerkiness), STS-Angle Cue = Sit-to-Stand (Movement Cues), Gait = Rhythmic Gait)*

Clinician Usefulness

Patient Inclusion and Exclusion Criteria

- Static Upright Balance (SB): The participants expressed that in general, patients with auditory perceptual difficulties and severe cognitive impairments would not be suitable for this form of biofeedback. One stated that candidates for musical biofeedback were those who are "unable to use tactile information from a therapist" to train balance, and those who are "motivated by music". Another mentioned that severely affected patients in particular could be easily confused by the feedback, which then "could be a minus", and that patients would "need some cognitive ability" to make use of it. Suggested target patient types were those with trunk stability problems, arm paralysis or neglect causing trunk tilt to the contralateral (paretic) side. Participants felt that this could potentially be used across physical impairment levels, ranging from acute/severe to moderate, although it would be less relevant as the patient's condition improves (e.g. if he/she begins walking).

- Dynamic Trunk Control (DB): Most participants felt that patients with trunk stability issues would be a suitable target group, but with better performance and flexibility than those that would be treated with the static balance interaction (SB), as dynamic balance exercises are typically more complex. One participant stated that sub-optimal spatial abilities would be clear exclusion criterion. In terms of physical impairment groups, mild to moderate groups were stated to be more likely to benefit from this interaction as "acute patients with hardly any power would not be able to do the tasks" and as such, sub-acute or chronic patients could also be targeted.

- Sit-to-Stand - Movement Jerkiness (STS-Jerk): Participants generally felt that this would be most suited to "high-level" patients with the ability to independently sit and stand, as it would be difficult for moderately or severely impaired individuals to carry out the challenging STS weight transfer along with paying attention to the interaction. An example target group could be chronic stroke patients who have learnt the basic STS action "but would like to improve their movement quality".

- Sit-to-Stand - Movement Cues (STS-Angle Cue): Participants generally felt that patients needed to have good trunk stability in order to benefit from this interaction. One mentioned that "a patient progressing from acute to moderate, adequate trunk control, sitting control, little dynamic control is a perfect candidate – just above acute is very good candidate". Another stated that it would be the same category as the STS-Jerk interaction with the ability to stand unsupported and fine motor control, although patients with memory problems would need to be excluded.
- Rhythmic Gait (Gait): Participants felt that inclusion criteria would depend not only on impairment severity but also on the location of the infarction/bleed. One provided a detailed explanation of how the typical effects of *cortical* strokes are stiffness and weakness, resulting in step time asymmetry and deviation from a straight line. The motor cortex plans movements and the basal ganglia coordinate their execution with the cerebellum. He continued that "RAS is commonly used in extrapyramidal conditions where power is normal but control is lacking. In cortical strokes, control is not lacking and the movement pattern is because of weakness and spasticity. The walking pattern can usually not be changed much with RAS like in Parkinson's Disease. Subcortical strokes with bleeding or infarcts of basal ganglia mimic Parkinson's Disease. Those patients are relevant to some extent but RAS will not work the same way for stroke patients as it does for Parkinson's Disease, broadly per se." Another participant concurred, mentioning that patients suffering from cerebellar or lower-brainstem bleeds, and thus having more cognitive ability but challenges with coordination, would be suitable candidates. Two participants expressed the concern that patients lacking rhythm-finding abilities would not be able to benefit from this interaction. In terms of physical impairment level, one participant stated that this would be "very attractive" to use with mild-moderately impaired patients who were at the stage of trying to stabilize gait aspects, a "useful and critical period where gait becomes more autonomous".

Movement Info Capture and Conveyance

- SB: The participants generally felt that the sensing system was able to capture relevant movement patterns away from the upright position both in the 3D space and the MLat and APos planes. They also felt that the biofeedback was effective at conveying this information to a therapist, but did not generally feel they received any new information about patient performance that vision would not provide. However, one participant mentioned that the information was more "exact" due to the fine nature of the measurements.

- DB: Here too, participants felt that the system effectively captured and conveyed the pertinent movement patterns owing to "the flexibility of the system and the feedback". While one participant did not feel that a therapist would receive additional information from the auditory feedback, another mentioned that they "did get a movement sense from the feedback, which would be useful to know if the patient is about to reach the target position or has overshot it", while stressing that it was more important for the patient to receive this information than the therapist.

- STS-Jerk: Responses were more varied for this interaction. One participant felt that jerky movements were reliably sensed, but that the system seemed not to distinguish these from movements that were merely "fast or rapid". Multiple participants felt that the auditory feedback provided extra information that was not available visually, noting that the music captured subtleties of jerkiness better than the eye did, and that this information could be clinically useful in determining the stage of the movement at which patients tended to exhibit jerkiness. They felt that this could be added to the clinical information already available about the patient, and that the objective measurement of jerkiness could be useful in monitoring patient progress. However, one participant noted that this interaction did not capture trunk bend, a crucial STS variable. Others felt that although the knowledge of jerk was no doubt useful to obtain, they were unsure of whether the best course of action would be to directly provide feedback on it, or who might benefit from something of this kind. One participant explained that due to weakness, it is common for patients to use "trick" or "compensatory" movements in order to stand, and therapists are often required to use "momentum" to help patients stand, which can result in unavoidably jerky STS transitions. Negative feedback here would be confusing or discouraging to these patients.

- STS-Angle Cue: Participants generally felt that the sensing system was able to effectively capture the forward trunk bending motion. One highlighted that this could be very useful due to the importance of trunk bend to the STS transition and the usual inability of patients to gauge the optimal bend angle for standing and sitting, leading to them either falling backward when sitting or crashing back down when trying to stand. This participant stated that lower limb strength, trunk strength, patient balance, patient height and surface height were the main factors determining optimal trunk bend angles. Another mentioned that a therapeutic consideration is the strategy used by the patient in standing (hip or trunk), both of which are accounted for by the sensor. As far as conveying information to the therapist is concerned, one participant mentioned that the biofeedback did this effectively, although it was more important for the patient to receive this information.

- Gait: Participants felt that the system was effectively able to capture cadence and step patterns even in patients not having proper heelstrikes, although one pointed out that gait is indeed multidimensional. Another expressed that the emphasis of the interaction on heelstrikes would be good in focusing the attention of patients who tended to land on the forefoot or side of the foot. Two felt that this information was effectively conveyed to the therapist through the biofeedback, which provided extra information ("hearing the relation between the steps") whose acquisition would otherwise entail the recording and analysis of video footage.

Clinician Usability

Relevance to Existing Therapy Protocols

- SB: Participants generally felt that the interaction would fit well and could easily be integrated into both existing occupational therapy and physiotherapy protocols, with one also pointing out that the geometric system (circular/elliptical zones) corresponded well with what was used in regular training. A recurring theme in participant responses was an inclination towards using this interaction to provide continuous feedback while doing other tasks such as standing, sitting upright in one's room watching TV, reading the newspaper or bedside sitting during meals. Another possibility is in goal-oriented tasks with the purpose of monitoring trunk orientation while the patient does something with their legs or holds a training ball. It could also be used in occupational therapy tasks such as peeling apples or wiping a table. A participant also stated that this could be adopted in gait training by physiotherapists using the "Bobath" approach [32]. Autonomy was stated by participants as an advantage in that it allows the patient to take charge in training their own static balance autonomously of the therapist, either alone or in groups. One participant compared this to existing dynamic posturography technology that uses visual feedback (e.g. Balance Master) and highlighted that this would be advantageous in terms of cost and portability while providing similar benefits.

- DB: There was good agreement that the interaction would fit well with existing protocols, given the abundance of goal-oriented reaching-related exercises in conventional therapy. One participant stated that its applicability could be widened by attaching the sensor to the upper limb or neck regions. Several felt that the concept could be adapted to create "fun training" scenarios, for example if the therapist too wears a sensor and conducts a "follow-me" exercise where the patient must mimic the movements of the therapist or if a visual component were to be added. One participant also stated that this too could aid patient autonomy in training by allowing the exercises to be done in groups with greater independence from the therapist.

- STS-Jerk: Participants similarly felt that this interaction would fit well with existing STS protocols and could easily be integrated. The participating therapist who had accompanied previous testing expressed that smoothness is important in all movements and indicates coordination; therapists want "timely and smooth setting in of muscles", relevant for STS. She also mentioned that this could be another "fun training" scenario, recalling the amusement of patients from Iteration 2 testing. Two participants mentioned that this interaction principle could also be used in a different task - providing feedback on hand or finger jerkiness while writing or drawing, as writing tremors are common among stroke patients.

- STS-Angle Cue: There was general agreement that this interaction would fit STS protocols, is relevant from an everyday therapeutic standpoint and could certainly be implemented in practice. One participant explained that a common practice is to have patients to sit with one side against a wall having a vertical stripe that serves as a trunk bend STS cue, and that it would be interesting to see how this form of biofeedback would work instead of that. Another mentioned the possible of greater patient autonomy as this would allow the patient to train STS anywhere independently, such as a chair in their own room. Adaptations of the interaction for other therapeutic purposes were also suggested, such as visual neglect, where a similar cueing sound could be provided when a particular visual angle is scanned. It could similarly provide arm or shoulder angle biofeedback when training the upper limb (e.g. reaching exercises).

- Gait: Participants felt that this interaction would be useful to existing protocols (depending on whether the physiotherapist follows Bobath or evidence-based approaches). One stated that patients at all stages needed to train gait rhythm, and that bodyweight-supported systems could make such training possible even from very early stages of gait rehabilitation. Another (a music therapist) expressed the possibility of using the drum trigger interaction to provide PSE (patterned sensory enhancement) even in pre-gait training when the patient sits still and tries to raise his/her feet.

Practicality

- SB: A participant mentioned that a therapist would have to be mindful of and account for a short set-up time where individual-specific biofeedback parameters would have to be adjusted. Another asked whether it would be possible for the entire interface to be made available on a mobile device for easier and more portable use, and whether biofeedback settings for a patient could be saved for future recall.

- DB: One participant stated safety as a key concern here; the simultaneous need for a therapist to operate an interface while ensuring that the patient does not fall could lead to a usability issue, as therapists generally prefer to have their hands free. Solutions such as a body-mounted remote control or sensor (follow-me training) could address this issue. Participants agreed that in most cases, patient supervision would be necessary to prevent injury. Concerning the interaction, one participant mentioned that diagonal movements (MLat + APos) are an important focus of training, while the available controls only allow target zone manipulation in one plane at a time. A topic of disagreement was whether the interaction can be used without showing patients a visual interface of the target zone. Some felt it was necessary, while others felt that the combined sensory information may cognitively overwhelm the patient.
- STS-Jerk: Safety concerns and the need for supervision were similarly mentioned by one participant for this interaction, as the possibility of patients falling back down on the seat could be dangerous. Another mentioned that the jerk interaction might not be feasible in cases where the therapist uses "momentum" to help the patient stand, which would result in avoidable rapid and jerky movements and trigger negative feedback in an unfair manner.
- STS-Angle Cue: There were no further comments on practicality regarding this interaction.
- Gait: One participant stressed patient safety as an important concern, saying that this type of gait training is tough as patients must attend to the music rhythm and can get carried away in trying to follow the music even though they lack the balance to safely do so - it is thus safer to use bodyweight-supported systems. Another participant discussed practical problems with treadmill training due to the complexity of having to set a treadmill speed appropriate for the cadence and step length of the patient. A third mentioned the possibilities of using different musical meters to train distinct gait patterns, be they purely bipedal or with a cane/walker.

Patient Usability

- SB: Pertaining to the music feedback in general, participants agreed that it was clearly perceptible, provided in a timely manner and that the individual zone size adjustments would account for the inherent variability in patient impairments. One participant elaborated that "most stroke patients eligible for balance training will have enough auditory comprehension, except those with global aphasia who won't be taken for training – so patients who are amenable will be able to perceive the feedback". There was also agreement that the feedback made sense in terms of the action that caused it and would be intuitive for the patient as long as the therapist explicitly mentioned the goal of the interaction. Cognitive load, on the other hand would depend on several factors, explained by one participant in terms of age, fatigue and cognitive ability. He stated that for example, a 60 year old stroke patient without dementia would be able to manage the interaction but a more elderly patient with dementia might not. Another mentioned that it would be up to the therapist to choose patients for whom this interaction would be most appropriate.

The individual suitability of feedback strategies was also generally agreed upon. Several stated that they felt the ambulance strategy was "good" in that it managed to be clearly perceptible without being overly annoying, and that the directional feedback was helpful. One participant mentioned that while the melody distortion strategy was clear, it was very annoying and likely to put the patient off but testing would be necessary to gauge this. On the other hand, the cartoon effect was less annoying but more perceptually "blurred". Another mentioned that the feedback needed to be "somewhat annoying but not *so* annoying" as they were. A third expressed not liking the use of the word "punishment" as he felt the notion of punishing a stroke patient in any way was unethical.

- DB: In this case, participants agreed that the optimal music feedback strategy would vary among individuals depending on their perceptual ability, making it important for therapists to be able to choose between strategies. For example, two participants felt that the *Instrumentation* strategy was harder to perceive than the other clearer ones, meaning it would be more suited to "the higher end of patients". In general, they agreed that the feedback was provided in a timely manner, but differed on their preferred feedback strategy. One participant favored the *Music Stop* strategy, saying that the *Melody Detune - Frequency Distortion* was too annoying. Another favored the latter, as the former "did not reflect mild perturbations". Participants also agreed that the feedback was sensible and that the possibility to adjust the target zone size was very good. One participant mentioned that while this could be a "nice refreshing auditory exercise" for some patients, the cognitive load would depend on individual spatial abilities as some can find it difficult to orient themselves, but that it was otherwise generally "OK".

- STS-Jerk: Participants agreed that the auditory feedback is suitably adjustable, timely, sensible, and intuitive "smooth movement = smooth music - goes very well in the brain" as long as the therapist gives explicit instructions, although it would need to be tested. One appreciated the idea of providing positive reinforcement through pleasant sounding music, while another appreciated making this training more interactive as patients typically have to do hundreds of STS repetitions during their rehabilitation. While some felt the feedback was sufficiently clear, one felt that the feedback was "too fast" and that the *Noise Disturbance (Scratch)* strategy was harder to perceive. Others favored the *Melody Tonic (Pitch)* strategy, saying that the *Noise Disturbance* was too annoying. In terms of patient cognitive load, one stated that it would depend on the awareness the patient has of their own jerky movements (falling down, for instance), saying the feedback would only be beneficial if patients lacked this awareness, otherwise it might create a "high" cognitive load.

- STS-Angle Cue: Participants agreed that while the feedback principle made sense, the choice of strategy would need to be individualized. Even in terms of their preferred strategy (*Bell* v/s *Wah Wah*), participants disagreed considerably. Some favored the bell, citing reasons such as "it is an all-or-none movement so a single cue makes sense", "the bell provides a clear signal indicating the time to act", "the wah wah may not make sense to all" and "wah wah is annoying". Others favored the wah wah out of personal preference. Some participants did feel that the bell may be too soft for some patients to perceive along with the music, and asked whether it might be possible to control its volume level or simply amplify it. One participant felt that the cue "should be facilitating and not complex" it would be most intuitive for the bell cue to be replaced by a human voice saying "Up" or "Down". Otherwise, participants agreed that the feedback was timely, the individual adjustments were sufficient and the cognitive load would be "OK" as long as the cues are perceptible.

- Gait: Between the two showcased interactions (*Pitched Disturbance Punishment* and *Foot Drum Trigger*), participants generally favored the latter as "patients will always look for" positive reinforcement, which is "facilitating" and "supports motor learning", although it would need to be tested. Participants felt the feedback was clear, timely, suitably adjustable and intuitive, although some felt the *Pitched Disturbance Punishment* may be excessively annoying, especially if a patient must turn around in a small space or has rhythm-finding difficulties. Another felt that the difference between the left and right foot drum was not very large. While one felt the cognitive load would be manageable due to the intuitiveness of the feedback in general, another mentioned that some patients may lose track and need to be regularly reminded to pay attention to the feedback.

- Music: Several participants highlighted the importance of subjectivity in terms of patient music preference and history of music consumption. One participant (a physiotherapist) felt it was enjoyable, but could not comment on its motivational value. Another (a music therapist) expressed that while the music was recognizable and engaging in that sense, it still sounded like computer music - some patients might not be bothered by alterations to their favorite music but for others, it is not that enjoyable. He speculated that this type of music would induce different affect and emotion than regular music while having a similar effect in terms of movement kinematics. While he appreciated the difficulty of the trade-off between control over music parameters and fidelity to the original, he concluded that it was simply "not the same". He also questioned the role of the music in, for instance, the STS Trunk Angle Cue interaction.

  Another participant expressed that while music taste was highly subjective, music with a "low pulse" (slow tempo) might be more suitable to activities such as static balance or STS so as to prevent it from spontaneously inducing movement. Similarly, slow tempo music might also be suitable to gait rehabilitation where patients walk with a low cadence. Two participants who were familiar with previous iterations appreciated the variety added by introducing multiple music styles, saying it "was not as monotonous as last time", "it is important to have variety" and that it "absolutely does sound better than it did before".

- Wireless Sensor: Participants agreed that it would be quite straightforward to strap on and that it would not add a significant practical overhead. One participant (accompanying physiotherapist from Iteration 2) expressed potential problems with using velcro as the strap material as it could potentially damage certain types of clothing materials (such as those used in sweaters) and suggested using elastic straps instead. She continued that hygiene could be a pertinent issue as well if the same sensors were to be used with multiple patients, and that they would have to be presented to the hygiene nurses for more details. Lastly, she asked whether the sensor could be "sewn into garments" or similar as we had experienced minor issues last time with foot sensors falling down mid-trial. Another participant mentioned that it was advantageous to be able to strap the sensor to the outside of clothing and not onto the skin directly, as some patients can find it "too close" or "annoying" when sensors are applied directly to the skin.

9.2 Survey - Music Production Quality

This evaluation aimed to obtain an expert assessment of the synthesized music, as well as suggestions for improvement. The main goal here was to evaluate the different rhythms (and sonic styles) in terms of musical aesthetic qualities and sonic production values. The target group comprised music producers and audio engineers, who were asked to provide ratings, impressions and suggestions on several recordings of system-generated music styles.

9.2.1 Participants

Ten music producers (one woman) participated in the survey. They were approached on social media, but the majority were from Aalborg University's Sound and Music Computing alumni community. They ranged in age from 25 to 37 years (mean = 27.1). In terms of skill level, five of them self-reported to be amateur producers, three intermediate and two advanced.

9.2.2 Setup and Procedure

The evaluation was conducted through an online survey (taking approximately 15 minutes). The participants were not informed about the purpose of the music and what it was meant to be used for. Synthesized clips of three MIDI-encoded songs in five rhythm styles (Dance, Reggaeton, Waltz, Breakbeat and Slow Rock) were showcased in a fixed order, with separate clips for each style showcasing the percussion section and the full ensemble. Questions about groove, expressiveness, production and mix quality were asked in a fixed order. The questions were in the form of 7-point scale ratings, as well as short/long subjective text responses.

9.2.3 Results

The 7-point scale responses are aggregated and shown in Table 9.1. It is apparent that there is generally good agreement among the respondents (from the relatively low standard deviation in ratings). Focusing more closely on different aspects of the clips, some response tendencies are seen across rhythm styles. The participants agreed that the rhythms had prominent pulse and groove characteristics (S1), but also that some were static and repetitive (S2) and lacked sufficient variations (disagreement to S3). A consistent comment was that the interaction between instrument rhythms was not always optimal. They also tended to agree that the main melody sounded computerized and lifeless (S4) and lacked a rich, interesting timbre (disagreement to S5). One repeated suggestion was to use a "different patch" which was "less staccato" with "more organic timbres" or from "sample libraries". A second consistent suggestion was to use LFO-based modulation and have notes of varying duration. In the majority of styles, the positive role of the

Statement #	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**	**S12**	**S13**
Focus	*Rhythm*			*Main Melody*			*Supporting Instruments*		*Arrangement and Mix*				
Dance	5.7 (0.9)	4.8 (1.8)	3.0 (1.3)	4.8 (1.5)	4.1 (1.5)	3.0 (1.6)	4.9 (1.3)	2.5 (1.2)	5.4 (1.2)	3.2 (1.4)	3.7 (1.6)	3.9 (1.4)	5.3 (0.8)
Waltz	4.9 (1.1)	3.4 (2.1)	4.3 (1.4)	4.0 (1.6)	4.4 (1.2)	3.5 (1.4)	4.7 (1.1)	2.8 (1.3)	4.2 (1.7)	3.8 (1.3)	3.3 (1.1)	3.9 (1.7)	4.2 (1.1)
Slow Rock	5.5 (1.1)	3.5 (1.2)	4.2 (1.5)	4.9 (1.9)	4.5 (1.6)	3.6 (1.7)	4.3 (1.4)	2.7 (0.7)	4.1 (1.8)	2.9 (1.6)	4.1 (2.1)	5.1 (1.2)	4.8 (1.4)
Breakbeat	5.1 (1.1)	4.9 (1.4)	3.2 (1.0)	4.6 (1.8)	3.9 (1.4)	3.3 (1.8)	3.3 (1.1)	4.1 (1.6)	3.6 (1.6)	4.8 (1.3)	3.4 (1.8)	3.8 (1.3)	4.0 (1.1)
Reggaeton	5.3 (1.7)	4.0 (1.8)	3.6 (1.2)	4.7 (1.9)	3.1 (1.9)	3.2 (1.8)	4.1 (2.0)	4.0 (1.8)	4.1 (2.0)	3.0 (1.6)	4.2 (1.8)	5.1 (1.0)	4.4 (1.7)

Table 9.1: Subjective ratings obtained for each rhythm style in the form of Mean (STD). The scale was from 1 (Strongly Disagree) to 7 (Strongly Agree). Green column headers indicate *positive* statements and red headers indicate *negative* ones. S1: The synthesized rhythm has a prominent pulse and strong groove. S2: The synthesized rhythm is overly static and repetitive. S3: The synthesized rhythm has the appropriate quantity of expressive and unpredictable variations. S4: The main melody sounds computerized and lifeless. S5: The main melody is emotionally expressive. S6: The main melody has a rich and interesting timbre. S7: The supporting instruments add richness and harmonic depth to the music. S8: The supporting instruments are of little to no musical value. S9: The instruments interact well with each other to create a full sounding arrangement. S10: The individual instruments clash with each other, causing the mix to sound muddy. S11: The individual and combined instrument timbres are appropriate for the recreation of this style of music. S12: The overall mix is optimally balanced in terms of musical dynamics and sonic 'punch'. S13: The overall mix is optimally balanced in terms of frequency spectrum (separation between instruments/overall tonality)

supporting instruments was agreed upon (S7, disagreement to S8). Pertaining to the arrangement and mix, the ratings were more divided among the rhythm styles due to the diversity in instrumentation, tempo and song choices among the clips. There was, however, mild agreement that the mixes were optimally balanced in terms of frequency spectrum (S13). Style-specific comments and suggestions are now presented:

Dance When asked to describe this style in a single word, some respondents recognized it as 'Eurodance' and 'Disco' but others used adjectives such as 'Pungent', 'Basic' and 'Garageband'. Several suggestions were made regarding the rhythm, such as using sidechain compression, a better bass drum sample, stronger pulse and more variations in the pattern. Pertaining to the melody, suggestions for stereo manipulation, sustain, richness and complexity were made.

Waltz Some adjectives used here were 'Folkish' and 'Calm' but also '8 Bit'. A key problem repeatedly stated here related to the interacting rhythms, specifically that the snare drum and melody patterns did not fit, the percussion needed to be brought forward, and that the patterns needed to be edited to convey the tempo. Production-wise, one respondent suggested reducing the robotic timing precision, while others suggested focusing on the sense of space, panning and using bowed-string samples for the accompaniment.

Slow Rock On one hand, there were adjectives such as 'Mellow', 'Pop' and 'Ballad', but on the other, '8 bit' and 'Advertisement'. Several suggestions were made related to the tone of the 'guitar' track, from aligning its rhythm to the percussion to filtering it for better frequency balance or using samples instead. Production-wise, a common suggestion was to enhance the bass, while others were to create a common sense of space, dual-track the guitar and better manage the interacting rhythms.

Breakbeat Adjectives for this style were generally negative, such as 'Rigid', 'Electroclash', 'Robotic' and '8 bit'. Rhythm-based suggestions included better fit between the various drum timbres, more variations and better "connectedness". In terms of production, the comments were that there were too many competing rhythms, a repetitive and unclear bass voice and lack of overall harmony.

Reggaeton Some adjectives used here were 'Hawaii', 'Pop' and 'Euroreggae'. Rhythm-related comments were that there were too many staccato elements and overlapping rhythms. Production-wise, several comments stated a lack of low end, an unusual combination of instruments and clashes among them. Some suggestions included octave-shifting, filtering and more organic textures.

Chapter 10

Discussion

In this study, a real-time flexible musical biofeedback system was conceptualized, designed and developed over the course of three iterations. Through dialogue with clinicians and reviewing past literature, a series of interactions targeting numerous stroke-relevant training activities was developed and honed. A software framework for the user-customizable generation of instrumental music was also designed and developed. This was made to work in tight and stable co-ordination with other biomechanical biofeedback functions such as movement sensing, movement feature computation, mapping, visualization and logging. The technology was evaluated after each iteration through a combination of expert interviews with therapists, structured activities with patients, surveys and various technical tests. Overall, the evaluations showed that the interactions are relevant, practical and applicable to real clinical use, with most musical feedback strategies being intuitive and easy to perceive, albeit with room for feasible improvement. Each of these evaluation procedures focused on a subset of the multi-faceted problem formulation given in Chapter 3. This chapter discusses the findings of the overall study in context with relevant past research and the final problem statement.

10.1 General Discussion - Interactions

We now proceed with a topic-wise discussion *(Abbreviations Used: SB = Static Upright Balance, DB = Dynamic Trunk Control, STS-Jerk = Sit-to-Stand (Movement Jerkiness), STS-Angle Cue = Sit-to-Stand (Movement Cues), Gait = Rhythmic Gait)*:

10.1.1 Clinician Usefulness

Target Subpopulations

The developed interactions cater to therapeutic needs of varied complexity, ranging from the relatively simple act of sitting upright to the complex act of walking. As the evaluation procedures showed, it therefore makes sense that these **interactions would be clinically useful with different sub-populations of stroke patients depending on their level of physical and cognitive impairment**. The usefulness was seen not only in the diversity of patients enlisted for Iteration 2 testing, but also in the expert interview responses pertaining to suitable patient groups for each interaction. Overall, the clinicians agreed that all interactions would be useful when applied to the appropriate patient types. Some of their stated pre-requisites such as having sufficient auditory perceptual and cognitive ability (and preferably being motivated by music) are in line with literature [52, 62] and apply fairly uniformly to all interactions. Future studies must take these into account when enlisting patients for testing. Besides these general criteria, the expert interview yielded detailed interaction-wise potential target group characteristics. As it is known that certain types of individuals may benefit more than others from biofeedback [21], this information will most certainly serve as a valuable starting point in framing inclusion and exclusion criteria. An important next step in future studies would be to frame these criteria in terms of established measures of balance and gait quality (as summarized in [57]). This will entail further clinician input and pilot testing.

In terms of the required level of trunk stability and general motor ability to engage in the interactions, the clinicians estimated that it would increase as we go from SB to DB, STS-Jerk, STS-Angle Cue and Gait. This corresponds well with the increasing complexity of the respective training, but there are also activity-specific criteria to consider aside from simply strength and stability. In DB, for instance, spatial comprehension abilities are essential to being able to move and hold the trunk in a desired position or follow a trajectory. Particularly in gait, two clinicians stated that the location of the bleed/infarct in the brain would be an important factor in determining the relevance of the rhythm-based interactions. This was not considered in the present study as most of the referenced RAS-Stroke literature [60] [99] [95] does not make any location-specific RAS efficacy distinctions, while there have been positive findings exclusively with cerebellar stroke patients [105]. Pilot tests and further clinician input will be necessary to determine the importance of stroke location as an inclusion criterion.

Movement Information Capture and Conveyance

The clinicians agreed that in the majority of interactions, **the sensing system was able to capture relevant movement patterns**. Broadly, this points to the suitability of wearable inertial sensors and aligns well with past research [57]. In SB and DB, the 2D geometric system (inspired by Costantini et. al. [17]) was judged to effectively capture trunk inclination in the horizontal plane. Integrating the MLat and APos angles into a single measure was thus an upgrade over Iteration 2 where only one of these angles could be sonified at a time. The clinicians felt similarly about the STS-Angle Cue interaction as it measures a similar quantity (APos Angle). This indicates that the orientation measurement algorithm (complementary filter based on [1, 81]) provides sufficiently accurate trunk angle measurements, at least when the rest of the body is stationary. Future studies should test the trunk angle measurement algorithm during dynamic conditions such as walking.

For the gait interactions, the clinicians felt that **the bilateral sensing system was able to effectively capture the temporal stepping patterns**, and the impact-based detection algorithm would also allow the interaction to work for patients lacking a proper heel-strike. This is also an upgrade over Iteration 2, where heel strike measurement was unilateral and thus unable to capture asymmetrical stepping patterns. However, it must be noted that the lack of sensing discrimination in terms of which part of the foot first contacts the ground makes it difficult to measure and provide feedback on foot placement patterns. A combined system using plantar force sensors (as reviewed in [57]) could address this problem. Measures of gait aside from merely rhythm-based ones must be added in future studies. Suggested parameters for sonification are ankle dorsiflexion, knee hyperextension and leg speed variability as described by Torres et. al. [100]. For STS-Jerk, while clinicians did feel that the system was able to capture jerky movements, one point of feedback indicated that it did not distinguish these from simply *fast* movements. This could be attributed to the manner in which the scalar jerk parameter is computed (norm of 3D differentiated acceleration), and to the fact that fast movements contain phases of high trunk acceleration along one or more principal axes. Future studies should investigate measures that better differentiate true jerkiness from velocity.

In SB, **the clinicians generally did not feel that the biofeedback gave them *new* movement information unavailable visually**. This could be attributed to the ostensibly large amplitude and low frequency of postural deviations in stroke patients, both easy to perceive through vision. As pointed out, the ability to hear target zone approach and overshoot could possibly be more useful in the case of DB. In a contrasting interaction focused on deviations of low amplitude and high frequency (i.e. STS-Jerk), they felt not only that the biofeedback provided extra information but also that **this information could be clinically useful in identifying problematic movement phases and monitoring patient progress**. This could be attributed to the superior temporal resolution of the auditory system [43] as well as the perceptual *amplification* of the jerkiness phenomenon through salient sonification strategies. A possible issue is that despite this extra information, it may not always be suitable to *directly* provide feedback on jerkiness, especially for individuals who cannot avoid it. Therefore, it may be more suitable for jerkiness information to be stored for future clinician reference than sonified during the training session. Future studies must ascertain the approach that works best for different types of patients.

For STS-Angle Cue, the clinicians did not express receiving any extra information, which makes sense as the sonification is primarily meant to serve as a cue for the *patient*. For gait on the other hand, clinicians felt that **the ability to hear temporal step patterns could be a promising alternative to tedious video analysis**. This could be ascribed to A) The perceptual amplification of small timing discrepancies through salient sonification strategies and B) The ability to directly compare patient foot strikes (drum hits) to a simultaneous timing reference (remaining music). When looking at these interactions (STS-Jerk, Gait) as *information transmission* from the clinician perspective, the expert assessments indicate that they satisfy the good interaction criterion of "information throughput" [38].

10.1.2 Clinician Usability

Clinical Relevance

A positive finding across clinician responses was the assessment that **all interactions were both clinically relevant and could easily be integrated into existing physiotherapy protocols**. This could be because the interactions were all developed either based on past literature or directly based on inputs from clinical stakeholders. Other encouraging notions were that some of the interactions could foster greater patient **autonomy** (SB, DB, STS-Angle Cue) in training and even be **fun** (DB, STS-Jerk). It must be noted, however, that the application itself must evolve considerably for the autonomy potential of its interactions to be realized. A conceivable step in this direction could be the development of a streamlined mobile application with a simple interface that can be used by a patient or relative. There must be a selection of predefined interactions that adjust automatically to suit the patient. The eventual deployment and practical integration of these interactions must be centered around promoting patient autonomy. The question of whether the training interactions are fun would likely be highly subjective and depend greatly on the patient's state of mind, attitude and receptiveness [52]. Enjoying the training would likely improve adherence and future studies should pay close attention to the manner in which patients respond to each interaction, and optimize their design to maximize the "fun" quotient. This could enhance attention and motivation, and in turn eventual physical outcomes [48]. Music-based neural reward mechanisms are likely relevant to this [59, 26], and should also be referenced more closely in future designs.

A recurring theme was **the possibility of integrating the interactions into *other* training tasks or targeting different body parts**. This indicates that the interaction principles are both therapeutically sound and versatile in terms of the variables fed back [21]. For instance, the large number of possible use-case scenarios mentioned for SB would point to the general importance of maintaining upright posture in a wide range of training settings. Potential benefits from the sensory substitution provided by the biofeedback [107] would make the interaction highly relevant in all these settings. This could in turn lead to beneficial effects beyond just static balance, due to the influence of proximal stability on distal mobility [18]. The positive comparison made by one clinician between this system and visual dynamic posturography is line with the wearable sensor advantages discussed by Dozza [21] and Ma et al. [57]. Having a separate dynamic balance interaction (DB) is line with the need to separate static and dynamic balance training [21] The suggested adaptation of the DB interaction to other body parts (upper limb, neck, shoulder) shows the general rehabilitative relevance of maintaining a position or following a trajectory. The idea of the patient ‘following’ the movements of a therapist (who also wears a sensor) to obtain auditory rewards is also highly interesting, and must be explored as part of future work. The developed multi-sensor assignment architecture will likely facilitate this process. Although a basic 1D dynamic target modulation functionality was realized in Iteration 2, it was limited in trajectory shapes and not properly tested. Future studies must generalize this to the 2D (MLat-APos) plane, make more trajectory shapes possible and test the interaction in a TIMP-based protocol to promote learning through rhythmic repetition where feasible [99].

The overall importance of movement smoothness makes **the STS-Jerk interaction not only relevant but also adaptable to writing or drawing tasks in upper limb rehabilitation**. Future studies must explore the feasibility of this with real patients, and the same goes for the STS-Angle Cue for visual neglect. The relevance of the STS-Angle Cue is justified by the importance of forward trunk bending in the STS transition [9]. The stated relevance of the Gait interactions to train patients even at a very early stage (pre-gait or bodyweight-supported training) is worth exploring, although it will need to be ascertained whether the gait detection and feedback function optimally in these settings. The option of time signature selection (in both MIDI and CMR modes) makes the gait interactions usable with ternary gait patterns as well (e.g. with a cane or walker support as mentioned by one clinician), although the corresponding movement parameters and auditory feedback strategies will have to be adapted to temporally accommodate a third ‘foot’.

Clinical Practicality

Patient safety and the need for supervision were often stated by the clinicians as important practical considerations, particularly for the interactions involving larger and more complex movements. This is highly relevant as safety concerns can damage the potential of the interactions to promote patient autonomy in training. Safety must be kept in mind when defining target groups for each interaction, and future studies must frame precautions and safety guidelines to minimize the physical risk to the patient engaging with the system. This especially applies to gait training where (as a clinician stated) the unbridled enthusiasm of a patient could lead to untimely injury. In Iteration 2 testing as well, the therapist needed to constantly position herself close to the patients and monitor them despite the fact that they were (and generally are [41]) all sub-acute. Bodyweight-supported systems can be beneficial here. Another goal must be **to design future systems such that the clinician is able to work with the patient in a hands-free manner** and is not encumbered by having to continuously operate an interface - particularly relevant to the DB interaction. Having to set up a computer application (like at present) could possibly be cumbersome. 'Follow-me' training scenarios or automatic target trajectory modulation (including diagonal trajectories) are both promising in this regard. Patient-specific setup time and complexity must both be minimized through the design of intuitive interfaces, possibly mobile or tablet-based. Allowing the therapist to save patient-specific settings for future recall could be helpful here, as one of the clinicians suggested. For usability in group-based training, it should be possible to provide patients with biofeedback through headphones, although signal splitting should be possible if a therapist would like to monitor the feedback in a one-on-one session. Volume levels must be strictly regulated, and open-back headphones should be used so that verbal communication is still possible with the patient over the music.

The last point of discussion here is the wireless sensor. Over the course of the three iterations, the **hardware prototype (M5Stack + silicone mount) worked as intended with no accidents, damage or mishaps**. In Iteration 3, the sensory battery life was sufficient for up to three brief training sessions and the mounting apparatus worked well in most situations, except on a few occasions when the thigh mounted sensor tended to slip down. Clinicians also felt that there would not be any significant practical overhead related to mounting the sensors. This embodies several known merits of wearable inertial sensors [57], which can be used in future studies in similar fashion, but certain factors must be considered, as explained by one of the clinicians. Hygiene considerations must be understood in more detail and implemented based on the guidelines in Denmark. This could be difficult as the M5Stack sensor cannot easily be washed, but more information must be obtained to come to a solution. The same would apply to plantar force sensors if future studies include these. The velcro strap must be replaced by an elastic strap to alleviate concerns related to clothing damage, tight fitting as well as hygiene. Possibilities of smaller and more robust 'smart-sensors' must also be investigated on a regular basis.

10.1.3 Patient Usability

This study focused almost exclusively on usability metrics pertaining to the auditory biofeedback (closely in line with the success criteria defined in [46]), as well as other constituents of user experience such as feedback aesthetics and the generated music.

Musical Feedback

Feedback perception and comprehension are critical to biofeedback success, and the user's ability to react to biofeedback depend greatly on its meaningfulness, timely delivery, perceptual salience and the cognitive load of processing it [46]. The underlying premise of musical biofeedback design in this study was the principle of rewarding desired bodily states with pleasant auditory states and vice versa [59]. Desired bodily states were defined in terms of customizable *target* values of movement parameters (MP) and a flexible data-independent 1-D sonification framework was designed as explained by Parseihian et. al. (2014) [73]. Up to Iteration 2, MP's were converted to continuous-valued audio parameters (AP) which controlled the intensities of varied sonification strategies. These strategies worked by providing pleasant full-sounding music as an implicit reward reference and 'punishing' undesired movement behavior by degrading acoustic characteristics such as consonance, instrumentation, musical complexity and synchronicity. Acoustic dimensions were combined into composite strategies for increased perceptual salience. In Iteration 3, the underlying feedback philosophy was carried forward but the continuous-valued MP's and AP's were mostly replaced by discrete ones to perceptually highlight changes in feedback intensity. Existing strategies were enhanced and new ones were added to suit the updated interactions. We now discuss the results of the final evaluation specifically pertaining to the feedback. It must be borne in mind that drawing any concrete conclusions on the feedback will require pilot testing with patients.

Meaningfulness, Timing, Cognitive Load and Adjustability: Despite the application being an example of fairly complex artificial sonification [82], the clinicians felt that **most of the feedback was both meaningful and sensible in context with the action that caused it**. This shows that the principle of basing feedback strategies on the implicit reference [73] of pleasant-sounding music is well founded, as well as the decision to keep the sonification framework one-dimensional. This aligns with the success criteria of interactive sonification applications, where it is important to effectively convey a message to the user in a manner that is easy to understand, learn and does not interfere with the task [35]. This was particularly the case for the directional negative feedback in SB, the pitch modulation in STS-Jerk and the *Foot Drum Trigger* interaction in Gait. This could be ascribed to the explicit relationship between the underlying movement phenomena and generated feedback sounds used in these cases (e.g. jerky movement = jerky pitch modulations, or lean left = ambulance sound in left ear). The latter directly leverages the known benefits of direction specificity of auditory biofeedback [21]. This was not as much the case for DB and STS-Angle Cue, possibly because the information (trunk position, standing cues) was coded as music in a more indirect manner (number of musical instruments, bell sound respectively) which could necessitate a learning phase. A suggestion received was to use voice cues such as ‘Up’ or ‘Down’. While this would certainly add clarity, it could also disturb the flow of the music in a far more conspicuous manner than a more musical bell/wah wah sound. This is a prime example of the known trade-off between pleasantness and intuitiveness (as well as precision) [35] or the "ambiguity added to data when codifying it using a heavily aesthetic approach" [16], a recurring theme in the assessment of the musical feedback strategies. While more utilitarian sounds such as voice instructions would be more intuitive on an immediate basis, aesthetically codified cues could be a better candidate in the long run even if a short learning period is needed. Similarly for gait, although feedback through synthesized walking sounds [84] may initially be more intuitive than triggered drums, the latter interaction could eventually provide the user with a more engaging and enjoyable experience. Nevertheless, there is likely much subjectivity among patients with regard to this, so a varied choice of feedback strategy types would be ideal for testing purposes.

For all interactions, **the clinicians perceived the auditory feedback to be provided in a timely manner**. For the most timing-sensitive ones (STS-Jerk and Gait), the measured feedback loop delay (refer Chapter 8) was about 90 ms on average, well below typical human auditory reaction times [46]. Despite the higher measured loop delay for the angle-based interactions (300 ms mean), the feedback was not perceived to be late. The high value of the measured result could be attributed to inaccuracies in loop delay measurement and lack of a visual angle reference to judge feedback timing. The precursors of the angle-based interactions were tested with patients in Iteration 2 as well, and there was no evidence that the feedback was provided too late. These findings indicate good temporal performance of the sensing, processing and feedback generation.

The clinicians expressed that **cognitive load is more subjective, with age, fatigue, general cognitive ability, memory and spatial abilities acting as key factors**. Each of the interactions will need pilot testing in order to better understand the factors affecting cognitive load associated with them, especially DB where a visual interface could possibly be necessary as a supplementary sensory stream. There were generally no problems with excessive cognitive load during Iteration 2 testing, but most of the volunteering patients were sub-acute and not representative of more severely impaired members of the population. For example, multiple clinicians warned that the gait interaction would not be usable with patients having poor rhythm-following ability. Future studies should address this by adding the option of adapting music tempo to measured patient cadence, like in the D-Jogger [64]. As far as individual adjustments to the interaction are concerned, a positive general assessment from the clinicians was that the **adjustment possibilities would be sufficient in tailoring the interactions to the diverse patient population**. This is in line with Iteration 2, where the adjustments were versatile enough to cater to each of the enlisted patients. Minor suggestions such as being able to increase the volume of the bell sound can easily be integrated in future studies.

Perceptual Salience and Aesthetics: The preceding discussion brings us to perhaps the most central aspects of the musical biofeedback, and how they interact with one another. At the end of three iterations, the time spent making incremental upgrades to the feedback strategy set was certainly vindicated by the assessments of the clinicians. With some exceptions *(Cartoon Effect in SB, No. of Musical Instruments in DB, Noise Disturbance in STS-Jerk and the Bell in STS-Angle Cue)*, **the feedback strategies were unanimously judged to be sufficiently clear for use with the target patient group**. This indicates that a discrete feedback mapping scheme may be more appropriate than a continuous scheme for some interactions (SB, DB, Gait). That said, the clinicians did mention that it would be ideal to have a set of feedback choices available to suit patients with varied perceptual ability, and the developed system is capable of providing this flexibility.

Auditory feedback aesthetics is of great importance, particularly when the goal is the design of motivating and satisfying experiences [38] for a sensitive target group such as stroke patients. Past research shows the strong connection between aesthetics and usability [3, 19] and points out the lack of aesthetic considerations in interactive sonification research [68, 16, 73] including balance biofeedback [15]. Even existing studies involving music-based feedback provide (or encourage the provision of) movement information through the addition of noise and distortion to the music [55, 89]. In the interviews conducted as part of Iteration 1 evaluation, **several patients found such feedback strategies to be clearly audible but considerably annoying**. In the final expert interviews, **clinicians tended to favor auditory feedback strategies where patients were given positive reinforcement rather than negative** (e.g. a preference of *Foot Drum Trigger* to *Pitched Disturbance* - "more facilitating"). One clinician took severe exception to the use of the term 'auditory *punishment*' as he considered the notion of **punishing patients in any way very wrong from an ethical standpoint**. A better term for future use would perhaps be *'negative feedback'*, which is line with the view of the interaction acting as a control system minimizing error [38]. Strategies such as the *Melody Ring Modulator V3*, *Melody Detune - Frequency Distortion* and *Noise Disturbance* were found to be particularly unfavorable by multiple clinicians, while they found less aggressive ones such as *Melody Tonic*, *Music Stop*, *Ambulance* and *No. of Musical Instruments* to be more suitable.

Although this is subject to individual preference and taste, certain inferences can cautiously be made at this stage. Keeping in mind that the application aims to come under Neuhoff's categorization of *artistic sonification* [68], the *clearest* feedback strategy is *not* necessarily the most appropriate. The feedback must aid the meaning-making process during training, while the music must inject the training environment with motivation and positivity. However, the latter can be undone if the feedback is overly unpleasant and discouraging. The balance between clear feedback and pleasant feedback is therefore a very fine one, and **the feedback need not be any clearer than is absolutely necessary** for performing the interaction. The clinician assessments provide insight into how to design good strategies; the favored strategies were either A) closely tied to the music generation itself (foot drum trigger, music stop) or B) a possible source of humor (melody tonic jerky feedback, ambulance) - "funny training" as some put it (and seen in Iterations 1 and 2 with STS-Jerk). The strategies that added deliberate artificial-sounding static disturbances (pitched/noise disturbance) to the music or purposefully corrupted its sound (ring modulators or melody detuning) tended to be deemed annoying or unpleasant.

The optimal philosophy may thus be to design strategies **that work by creatively manipulating the music generation process itself and introduce humorous elements** where possible to ensure that the patient always receives positive, or at the very least funny reinforcement rather than a 'scolding'. The main design challenge then lies in creating intuitive and meaningful music-centric strategies that are pleasant yet perceptible. An exemplar is the *Foot Drum Trigger* strategy, which intuitively sonifies an impulsive action (foot strike) as an impulsive sound (drum) in a manner that ties in seamlessly with the remaining music and is perceptually salient. The patients in Iteration 2 enjoyed this interaction as well even in its unilateral format. A good example of a less successful strategy is the now-discarded *Melody Release Time* strategy in Iteration 1, which aimed to be pleasant but was not easily perceptible or intuitive. This design challenge is a prime example of the trade-off among precision, pleasantness and intuitiveness [35].

Music Aesthetics

The generated music is discussed in greater detail in the next section, but some important takeaways from the expert interviews are first addressed here as part of patient usability. In Iterations 1 and 2, there were **considerable individual differences among the patients both in terms of their rhythm-following ability, responses to the synthesized music and how much the music quality mattered** to them during training. While some were less critical of the clearly synthetic sound, certain individuals did not appreciate the computerized versions of songs they knew and recognized, and tended to feel that something was lacking in them. The overall verdict at the end of Iteration 2 was that the music simply needed to be better, and Iteration 3 introduced MIDI support, drum samples, instrument variants and new synthesis methods to address this requirement. While it was still judged to sound like computer music, multiple previously involved clinicians felt that aside from being easily recognizable, it undoubtedly sounded both better and less monotonous due to the introduction of sonic variety. Some interesting new inputs about preferably using a slow music tempo in certain types of training (SB, STS-Jerk and Gait) were also obtained. Indeed, increased arousal and spontaneous bodily movement in response to uptempo music is well documented in literature in terms of motor resonance mechanisms [59] and was seen in Iteration 2 testing as well. Stroke patients walk with lower cadence than unimpaired humans, and the music must be designed with this in mind. Overall, future music designs must take training activity-specific considerations into account. The ability to alter music tempo, rhythm and sonic style will certain go a long way in addressing much of the inherent individual differences among patients.

It must be stressed that the main limitation of the final evaluation was the shortage of trials with real patients; the interactions were defined, built and tested from a third-person perspective. Due to the circumstances, patient tests could only be conducted during Iteration 2, and with relatively few individuals for a short duration apiece. The final interaction set was evaluated through expert interviews and which, despite their merits, cannot replace real-life testing. Even the expert interviews themselves were conducted remotely using video footage; live meetings and demonstrations with fewer time constraints would have been ideal. It will only be possible to obtain an accurate estimate of interaction usefulness and usability through more extensive tests in real clinical environments with the appropriate patient groups for each interaction. Future studies must also take the first-person (embodied) perspective into design consideration [38].

10.2 Generated Music

The efficient encoding and synthesis of instrumental music, while not the primary focus of the study, has been an ongoing process across iterations with regular upgrades and improvements. Even though the music is only meant for training purposes, past musical biofeedback research has shown that the quality of the music can have an impact on user experience [106]. While the emotional effects elicited by music are highly subjective [47], tailoring the present music to the preferences of users is critical in order to reap the known benefits of music during exercise [44] [71] and therapy [52]. While the majority of evaluation procedures revealed that the system-generated music still has numerous shortcomings, the main merits must be acknowledged at this stage. The biofeedback application, as of Iteration 3, **is able to generate music in parallel with its other functions, and does so in a stable, thread-safe and computationally efficient manner**. For the encoding and reproduction of musical structures, a custom compact representation was devised and implemented (CMR), which despite several limitations proved to work well for simple song structures. These limitations were addressed by the integration of MIDI support for both melody and percussion passages, and the final version supports both formats. The MIDI implementation is complete with looping of percussion patterns and tempo/track-dependent sequencing. CMR is more computationally efficient and supports randomized drum pattern variations, but MIDI is more flexible and intuitive in its encoding process. Support for different time signatures and timing modes (straight, triplet) was also added. This makes it possible to quickly encode and reproduce songs that are familiar to a patient in advance of a training session. As it takes musical knowledge to encode music, and it may not always be possible to obtain patient preference information in time, future studies may benefit from having a library of demographic-appropriate music pieces encoded and ready. Possible copyright issues related to these must be anticipated and resolved in advance.

In terms of music synthesis and mixing, the final version provides **considerable flexibility**. Any piece of encoded music can be reproduced in a number of rhythmic music styles that combine a common set of instrument roles in different sonic palettes at a wide range of tempos. This is in line with the *morphocon*-based sonification design philosophy described in [73], which provides feedback through changes in the sound *evolution* rather than the sound itself. Each instrument role can be reproduced in three sonic variants that include both sample-based and synthesized sounds. The decision to use FAUST was fruitful as the available audio function libraries substantially sped up the prototyping and testing process. The synthesis methods used are simple, yet versatile in terms of the timbres they can produce (e.g. FM synthesis). Custom functions to generate both monophonic and polyphonic content in a sonically consistent manner irrespective of tonic and key are in place. The instruments are discretely processed by channel equalizers, compressors, faders and master processing to create a balanced mix at the output. Each instrument variant and music style has a pre-defined set of mixing settings which can be modified in real-time in the application. Overall, this framework can be easily modified to accommodate new sounds and settings and directly used in future studies.

The online survey conducted with music producers provided a useful appraisal of the generated music, and helped in identifying several possible areas of improvement, all of which can be addressed by future studies. Although most felt the rhythms had **prominent grooves**, they found a **lack of expressive variations**. This could be attributed to the MIDI percussion grooves lacking a random component (like Iterations 1 and 2 had). Novelty and unpredictability are important determinants of music-driven reward [59] and must be addressed in future studies, along with the problems with interacting rhythms among instruments. Several issues were identified with the main melody synth, which substitutes for the vocal track of encoded songs. Participants generally felt that **all variants lacked expressiveness, timbral richness and was too 'staccato'**. The exploration of superior synthesis techniques, high quality samples and LFO modulation of envelope decay in future studies will help addressing these issues. At this time, FAUST does not provide stable support for a large number of imported audio samples, so alternatives may have to be explored in case this functionality does not receive the necessary troubleshooting.

Many participants felt that the arrangement sounded muddy in certain styles, was **lacking in spectral fullness or contained "unusual combinations of instruments"**. This could be attributed to the limited number of instrument variants to choose from in designing each of the styles, which should be expanded in future studies. Other aspects of the arrangement such as **octave separation, timing swing and chord voicing** must also be paid attention to. Mix-related issues were also identified, specifically that the instruments sounded "disjointed" due to a **lack of common space and problems with spectral and dynamic balance**. While more uniform reverb processing can address the former, the spectral and dynamics-related problems can be ascribed to A) the lack of powerful mix processing tools like multi-band compressors, tape saturation and analog-modelled mix buss compressors like most mix engineers use and B) the lack of a proper mixing setup or environment. The bulk of mixing was done on a pair of studio headphones and reference mono speaker, which is sub-optimal at best. A limitation of this evaluation method was that the participants were young and differed from the target group in terms of both perceptual ability and musical sensibilities. Future studies must add DSP functionality for more powerful mix processing and carry out the mixing process in an appropriate environment using better equipment.

10.3 Developed System

10.3.1 Design and Implementation Choices

Several aspects of the developed hardware and software system are retrospectively discussed. A pivotal decision was to develop a JUCE/FAUST standalone application rather than using a digital audio workstation directly as previous studies have done [102]. This was similar to my past work [79] [80] and while it added extra software development work in terms of UI design, music encoding, reading and playing, **it brought all aspects of functionality together in a flexible, portable and usable package**. JUCE and FAUST brought with them a wide range of ready functionality related to user interface design, data reading, handling, and storage, timed callbacks, audio synthesis and audio processing in a thread-safe and developer-friendly environment, making this combination ideal for prototyping. The decision to use an ESP32 board (M5Stack Grey) with OSC transmission not only made the IMU data easily readable in JUCE but also made it easy to scale up the number of sensors in Iteration 3 and add other peripherals in the future. These include remote control devices or other novel control surfaces which may be necessary for the hands-free use of certain interactions like DB.

The developed MP measurement, AP selection and mapping interface made it easy to test different combinations together in a reasonably data-independent manner, successfully implementing and augmenting the philosophy put forth by Parseihian et al. (2014) [73]. Not only did this make several biofeedback modes possible (guidance, error correction, dynamic trajectories), but it **also allowed more complex MP's *(STS Angle Cue, 2D Trunk Projection Zones and Step Periodicity Feature)* exist and function within this very scheme**. The affordances of JUCE UI elements also made it possible to shape the geometric static balance zone scheme of Costantini et al. [17] into a versatile user-customizable interaction mechanic, complete with real-time visualization. The architecture of the *GaitAnalysis* class and the user interface make the addition of new MP's a straightforward task in future versions. A limitation of the sonification mapping framework is that while it allows a range of desired behaviors and mapping function shapes, it does not support a true sigmoid curve, the optimal shape for auditory feedback mapping in balance training [21] (although it can be approximated). Future studies must make this addition.

10.3.2 Technical Performance

This section focuses on the findings of the experiments covered in Chapter 8.

Sensor Range: The effective transmission range of M5Stack sensors was measured in terms of the percentage of receiver sensor callback intervals where new OSC data was received. The documenting and logging functionality was thorough and robust. The results of the experiment show that **the sensor is capable of very healthy transmission even at a considerable distance from the receiver laptop in a large room**. Even introducing a large obstacle (room corner) only dropped the receiver packet percentage to 82%. Bear in mind that prior to Iteration 3 it was <70%, with no perceptible loss in performance due to smoothing at the receiver end. The tolerable packet drop percentage depends on the time critical nature of the movement information to be captured, and worsening transmission effectiveness would conceivably have a greater effect on gait and jerk-based interactions. The main takeaway is that the sensor range should suffice in most real-life situations barring excessive WiFi interference. The limitation of the experiment is that the range analysis could not be conducted in a real-life large training ward-like location at even greater distances. Future studies must explore the effect of transmission efficiency reductions on the feasibility of all interactions, so as to gauge maximum effective sensor range for each.

Biofeedback Loop Delay: System-wide loop delay was measured for gait, jerk and angle-based interactions using synchronized audio and video analysis respectively. The results have already been discussed earlier under Patient Usability, wherein **the feedback was generally found to be timely** during evaluation. Possible sources of measurement inaccuracy are A) The low video framerate (23.91 FPS) that could cause upto 42 ms worth of positive error and B) that the audio output was recorded directly from the sound card and thus does not take into account hardware output latency or propagation delays C) Lack of threshold angle reference in the measurement of trunk angle loop delay. Future measurement protocols must address these flaws and attempt to determine incurred delay at each individual processing step so as to identify and remedy possible system design flaws.

Computational Performance: % Processor Time and CPU Usage were used to measure the performance of the JUCE application, and results of the evaluation showed that **the application runs in real-time very efficiently even when the most computationally stressful functional elements are active**. The preliminary tests showed that CMR was consistently more efficient than MIDI mode, and an unexpected finding was that playing the music at 60 BPM tended to be *more* CPU-intensive than 150 BPM. Differences between music styles could be attributed to the synthesis algorithms involved. A stress-test was conducted over 100 seconds by combining the most computationally intense music and MP measurement configurations, and the net CPU usage peaked at 26%. The main limitation of the performance measurement was that CPU polling was done at the low frequency of 1 Hz, causing CPU spikes to potentially be missed, although it is hoped that the long test duration of 100 seconds would have compensated for this loss of fine-grained information. CPU spikes were indeed seen, but no audio dropouts occurred at any point. Future studies should also gauge the effect of background CPU tasks on application performance.

The memory usage was also relatively low (150 MB). The separate preliminary tests showed that the bulk of the computational load can be attributed to the music synthesis and processing, which makes sense given its high sample rate and relative computational complexity owing to multiple audio tracks. The FAUST compiler optimizations can certainly be credited with the obtained level of efficiency. Although the testing was done on a a fairly powerful i7 laptop, the findings have several positive implications; the synthesis and processing algorithms have processing headroom to allow substantial upgrade, allowing superior sound generation algorithms and sonification strategies to be implemented. The JUCE application can include faster visualizations and more biofeedback controls. A larger number of high-quality sample files can be loaded into memory to improve the music quality. Much of the existing functionality can be ported to mobile platforms such as phones or tablets in future studies.

In summary, the methods used during the final evaluation were the best substitutes to true user-centered testing, that could be designed and arranged in the

COVID-19 situation. Ideally, the final interactions would have been tested in a manner resembling Iteration 2, with a larger number of more diverse patients and with a more controlled, organized procedure including structured interviews. The music quality would have been evaluated with experts in a more suitable demographic. Even expert interviews would have been conducted in person over longer durations, so as to allow the clinicians to try using the technology themselves and provide input, as opposed to simply watching videos. While the present procedures have provided valuable information, it is hoped that there will be opportunities to follow them up with future studies conducted in a more ideal fashion. Only through rigorous pilot testing, effect testing and user experience evaluation will the technology reach the level of maturity required for widespread augmentation of stroke rehabilitation protocols.

Chapter 11

Conclusion

This research aimed to investigate music interaction schemes and musical biofeedback strategies that could be applied to common training activities in balance and gait rehabilitation of hemiplegic stroke patients. The study was based on a theoretical foundation of biomechanical biofeedback with wearable sensors, interactive sonification, musical biofeedback and neurologic music therapy. Through a user-centered design process, a prototype application was developed and honed over three iterations, all of which were evaluated in collaboration with relevant stakeholders. Through this prototype, a series of interactions targeting static balance, dynamic balance, sit-to-stand transitions and walking were made possible in a patient-tailored fashion. Flexible schemes for music encoding, sequencing and synthesis were also built. The findings of the study indicate that the developed interactions have the potential to be useful to several subsets of the stroke patient population at different stages of recovery, and suitable inclusion and exclusion criteria are paramount. The developed sensing system was generally found to be able to capture movement patterns relevant to each activity. In some cases (sit-to-stand jerkiness and gait), clinicians have the potential to gain information about patient movement quality that would not be readily available visually.

The interactions were also found to be relevant to existing training protocols and relatively easy to integrate in a way that boosts patient autonomy and promotes fun during training. Several interaction principles were judged to have potential in other training activities than originally intended. Practically, the importance of patient safety while training will entail scrutiny by the responsible clinician during most interactions involving movement. The developed sensing hardware was not found to add a significant practical overhead, although hygiene considerations must be made in future studies. Through the study, the musical biofeedback strategies evolved from simplistic 1-D continuous-valued manipulations of auditory dimensions to a number of composite and/or discrete feedback strategies with enhanced perceptual salience. The feedback was generally found to be meaningful, intuitive, timely and sufficiently adjustable to suit patients with varied abilities. Patient cognitive load, on the other hand, is likely to vary among individuals. Although the majority of final feedback strategies were found to be easy to perceive, a subset of these were deemed excessively annoying as forms of negative feedback. Clinicians expressed a preference for strategies promoting positive reinforcement, considering the sensitivity of the patient group. This presents future studies with the sonification design challenge of managing the fine balance of precision, pleasantness and intuitiveness. Feedback unpleasantness could potentially be alleviated by introducing humorous auditory entities.

The quality of the synthesized music was identified as an issue during the evaluation of early iterations, and the final version made several improvements in the form of support for MIDI and high quality audio samples. Although this was found to address several of the problems related to monotony and sound quality, a survey conducted among music producers identified several potential areas for improvement, related to arrangement choices and mixing decisions. A technical system evaluation found that the wireless sensing system has acceptable range, the biofeedback loop delay is sufficiently low, and the application is computationally efficient. Overall, a large portion of the developed functionality can be carried forward, and many of the identified improvements can be implemented with relatively minor changes. Issues related to interface usability and portability from the clinical standpoint, however, will necessitate significant design and testing in its own right. An immediate follow-up goal is to design a set of facilitating positive reinforcement strategies based on the findings of this study for each interaction, and hone them through iterative focus-group testing. Future studies should concentrate on evaluating the interactions in the clinical environment in a randomized controlled manner to systematically investigate physical and psychological effects, both short and long-term. It is hoped that the seamless integration of music technology into the clinical environment can help this patient group, both through improved physical outcomes and the augmentation of training to a more engaging and enjoyable experience.

Bibliography

[1] *A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications*. URL: http://www.starlino.com/imu_guide.html.

[2] L Ada and P Westwood. "A kinematic analysis of recovery of the ability to stand up following stroke". In: *Australian Journal of Physiotherapy* 38.2 (1992), pp. 135 –142. ISSN: 0004-9514. DOI: https://doi.org/10.1016/S0004-9514(14)60558-4.

[3] Michael E. Atwood and Craig M. MacDonald. "Understanding usefulness in human-computer interaction to enhance user experience evaluation". PhD thesis. 2012.

[4] Fibio Badke. "Balance Deficits in Patients with Hemiplegia: Considerations for Assessment and Treatment". In: *Balance: proceedings of the American Physical Therapy Association forum.* (1990).

[5] Mark Ballora. "Sonification, Science and Popular Music: In search of the 'wow'". In: *Organised Sound* 19.1 (2014), 30–40. DOI: 10.1017/S1355771813000381.

[6] Juan Manuel Belda-Lois et al. "Rehabilitation of gait after stroke: A review towards a top-down approach". English. In: *Journal of NeuroEngineering and Rehabilitation* 8.1 (2011). ISSN: 1743-0003. DOI: 10.1186/1743-0003-8-66.

[7] Ilias Bergstrom et al. "Using Music as a Signal for Biofeedback." In: *International journal of psychophysiology : official journal of the International Organization of Psychophysiology* 93 (Apr. 2013). DOI: 10.1016/j.ijpsycho.2013.04.013.

[8] Rhushabh Bhandari et al. "Music-Based Respiratory Biofeedback in Visually-Demanding Tasks". In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME 2015. Baton Rouge, Louisiana, USA: The School of Music, the Center for Computation, and Technology (CCT), Louisiana State University, 2015, 78–82. ISBN: 9780692495476.

[9] Amira Boukadida et al. "Determinants of sit-to-stand tasks in individuals with hemiparesis post stroke: A review". In: *Annals of Physical and Rehabilitation Medicine* 58.3 (2015), pp. 167 –172. ISSN: 1877-0657. DOI: https://doi.org/10.1016/j.rehab.2015.04.007.

[10] J. Bradt et al. "Music therapy for acquired brain injury". In: *Cochrane Database of Systematic Reviews* 7 (2010). ISSN: 1465-1858. DOI: 10.1002/14651858.CD006787.pub2.

[11] Roberto Bresin et al. "Expressive sonification of footstep sounds". In: *Proceedings of the Interaction Sonification workshop (ISon)*. May 2010.

[12] Jeska Buhmann et al. "Spontaneous Velocity Effect of Musical Expression on Self-Paced Walking". In: *PloS one* 11 (May 2016), e0154414.

[13] Cathrin Bütefisch et al. "Repetitive training of isolated movements improves the outcome of motor rehabilitation of the centrally paretic hand". In: *Journal of the Neurological Sciences* 130.1 (May 1995), pp. 59–68. ISSN: 0022510X. DOI: 10.1016/0022-510X(95)00003-K. URL: https://linkinghub.elsevier.com/retrieve/pii/0022510X9500003K (visited on 04/01/2020).

[14] Pao-Tsai Cheng et al. "The sit-to-stand movement in stroke patients and its correlation with falling". en. In: *Archives of Physical Medicine and Rehabilitation* 79.9 (Sept. 1998), pp. 1043–1046. ISSN: 00039993. DOI: 10.1016/S0003-9993(98)90168-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S000399939890168X (visited on 04/04/2020).

[15] Lorenzo Chiari et al. "Audio-Biofeedback for Balance Improvement: An Accelerometry-Based System". In: *IEEE transactions on bio-medical engineering* 52 (Jan. 2006), pp. 2108–11. DOI: 10.1109/TBME.2005.857673.

[16] Stuart Duncan Haffenden Cornejo. "Towards Ecological, Embodied and User-Centric Design in Auditory Display". en. In: *Proceedings of the 24th International Conference on Auditory Display - ICAD 2018*. Houghton, Michigan: The International Community for Auditory Display, June 2018, pp. 191–196. ISBN: 978-0-9670904-5-0. DOI: 10.21785/icad2018.016. URL: http://hdl.handle.net/1853/60078 (visited on 01/07/2020).

[17] Giovanni Costantini et al. "Towards the enhancement of body standing balance recovery by means of a wireless audio-biofeedback system". In: *Medical engineering & physics* 54 (2018), pp. 74–81.

[18] Tamaya Van Criekinge et al. "The effectiveness of trunk training on trunk control, sitting and standing balance and mobility post-stroke: a systematic review and meta-analysis". In: *Clinical Rehabilitation* 33.6 (2019). PMID: 30791703, pp. 992–1002. DOI: 10.1177/0269215519830159. eprint: https://doi.org/10.1177/0269215519830159. URL: https://doi.org/10.1177/0269215519830159.

[19] Antonella De Angeli et al. "Interaction, usability and aesthetics: what influences users' preferences?" In: Jan. 2006. DOI: 10.1145/1142405.1142446.

[20] Volker Dietz. "Interaction between central programs and afferent input in the control of posture and locomotion". In: *Journal of Biomechanics* 29.7 (1996), pp. 841 –844. ISSN: 0021-9290. DOI: https://doi.org/10.1016/0021-9290(95)00175-1. URL: http://www.sciencedirect.com/science/article/pii/0021929095001751.

[21] Marco Dozza. "Biofeedback systems for human postural control". PhD thesis. Alma Mater Studiorum, Università di Bologna, 2007.

[22] Marco Dozza, Lorenzo Chiari, and Fay Horak. "Audio-Biofeedback Improves Balance in Patients With Bilateral Vestibular Loss". In: *Archives of physical medicine and rehabilitation* 86 (Aug. 2005), pp. 1401–3. DOI: 10.1016/j.apmr.2004.12.036.

[23] Marco Dozza et al. "Effects of Linear versus Sigmoid Coding of Visual or Audio Biofeedback for the Control of Upright Stance". In: *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* 14 (Jan. 2007), pp. 505–12. DOI: 10.1109/TNSRE.2006.886732.

[24] M Engardt. "Long-term effects of auditory feedback training on relearned symmetrical body weight distribution in stroke patients. A follow-up study". In: *Scandinavian journal of rehabilitation medicine* 26.2 (1994), 65—69. ISSN: 0036-5505. URL: http://europepmc.org/abstract/MED/7939494.

[25] Susan E Fasoli, Hermano I Krebs, and Neville Hogan. "Robotic technology and stroke rehabilitation: translating research into practice". In: *Topics in stroke Rehabilitation* 11.4 (2004), pp. 11–19.

[26] Christopher D. Fiorillo, Philippe N. Tobler, and Wolfram Schultz. "Discrete Coding of Reward Probability and Uncertainty by Dopamine Neurons". In: *Science* 299.5614 (2003), pp. 1898–1902.

[27] Rebecca Forkan et al. "Exercise adherence following physical therapy intervention in older adults with impaired balance". In: *Physical therapy* 86.3 (2006), pp. 401–410.

[28] Thomas Fritz et al. "Musical Agency During Physical Exercise Decreases Pain". In: *Frontiers in Psychology*. 2017.

[29] Thomas Fritz et al. "Musical Agency Reduces Perceived Exertion During Strenuous Physical Performance". In: *Proceedings of the National Academy of Sciences of the United States of America* 110 (Oct. 2013). DOI: 10.1073/pnas.1217252110.

[30] Thomas Fritz et al. "Musical Feedback During Exercise Machine Workout Enhances Mood". In: *Frontiers in Psychology* 4 (Dec. 2013), p. 921. DOI: 10.3389/fpsyg.2013.00921.

[31] Oonagh Giggins, Ulrik McCarthy Persson, and Brian Caulfield. "Biofeedback in rehabilitation". In: *Journal of neuroengineering and rehabilitation* 10 (June 2013), p. 60. DOI: 10.1186/1743-0003-10-60.

[32] Julie Vaughan Graham et al. "The Bobath concept in contemporary clinical practice". In: *Topics in stroke rehabilitation* 16.1 (2009), pp. 57–68.

[33] Jason Harris et al. "Sonic Respiration: Controlling Respiration Rate through Auditory Biofeedback". In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, 2383–2388. ISBN: 9781450324748. DOI: 10.1145/2559206.2581233.

[34] Henk Hendricks et al. "Motor Recovery after Stroke: A systematic Review of the Literature". In: *Archives of physical medicine and rehabilitation* 83 (Dec. 2002), pp. 1629–37. DOI: 10.1053/apmr.2002.35473.

[35] Thomas Hermann, Andy Hunt, and John G Neuhoff. *The Sonification Handbook*. Logos Verlag Berlin, 2011.

[36] Fay Horak, Laurie King, and Martina Mancini. "Role of Body-Worn Movement Monitor Technology for Balance and Gait Rehabilitation". In: *Physical therapy* 95 (Dec. 2014). DOI: 10.2522/ptj.20140253.

[37] Fay B. Horak, Charlotte L. Shupert, and Alar Mirka. "Components of postural dyscontrol in the elderly: A review". In: *Neurobiology of Aging* 10.6 (1989), pp. 727 –738. ISSN: 0197-4580. DOI: https://doi.org/10.1016/0197-4580(89)90010-9. URL: http://www.sciencedirect.com/science/article/pii/0197458089900109.

[38] Kasper Hornbæk and Antti Oulasvirta. "What Is Interaction?" English. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. null ; Conference date: 06-05-2017 Through 11-05-2017. Association for Computing Machinery, 2017, pp. 5040–5052. DOI: 10.1145/3025453.3025765.

[39] *IM Gait Mate*. URL: http://www.interactivemetronome.com/IMW/IMPublic/products.aspx.

[40] Anne J. Blood and Robert Zatorre. "Intensely Pleasurable Responses to Music Correlate With Activity in Brain Regions Implicated in Reward and Emotion". In: *Proceedings of the National Academy of Sciences of the United States of America* 98 (Oct. 2001), pp. 11818–23. DOI: 10.1073/pnas.191355898.

[41] Diane Jette et al. "Physical Therapy Interventions for Patients With Stroke in Inpatient Rehabilitation Facilities". In: *Physical therapy* 85 (Apr. 2005), pp. 238–48. DOI: 10.1093/ptj/85.3.238.

[42] B. B. Johansson. "Current trends in stroke rehabilitation. A review with focus on brain plasticity". In: *Acta Neurologica Scandinavica* 123.3 (2011), pp. 147–159. DOI: 10.1111/j.1600-0404.2010.01417.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1600-0404.2010.01417.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0404.2010.01417.x.

[43] Emil Jovanov et al. "Tactical audio and acoustic rendering in biomedical applications". In: *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society* 3 (July 1999), pp. 109–18. DOI: 10.1109/4233.767086.

[44] Costas Karageorghis and David-Lee Priest. "Music in the exercise domain: A review and synthesis (Part II)". In: *International review of sport and exercise psychology* 5 (Mar. 2012), pp. 67–84. DOI: 10.1080/1750984X.2011.631027.

[45] Teresa J Kimberley et al. "Electrical stimulation driving functional improvements and cortical changes in subjects with stroke". In: *Experimental Brain Research* 154.4 (2004), pp. 450–460.

[46] Anton Kos and Anton Umek. In: *Biomechanical Biofeedback*. Sept. 2018. ISBN: 978-3-319-91348-3. DOI: 10.1007/978-3-319-91349-0_2.

[47] Gunter Kreutz et al. "Using music to induce emotions: Influences of musical preference and absorption". In: *Psychology of Music - PSYCHOL MUSIC* 36 (2007), pp. 101–126. DOI: 10.1177/0305735607082623.

[48] Peter Langhorne, Julie Bernhardt, and Gert Kwakkel. "Stroke rehabilitation". In: *The Lancet* 377 (May 2011), pp. 1693–1702. DOI: 10.1016/S0140-6736(11)60325-5.

[49] Caroline Larboulette and Sylvie Gibet. "A review of computable expressive descriptors of human motion". In: *Proceedings of the 2nd International Workshop on Movement and Computing*. 2015, pp. 21–28.

[50] Soon Hyun Lee, Kyoung Jin Lee, and Chang Ho Song. "Effects of Rhythmic Auditory Stimulation (RAS) on Gait Ability and Symmetry after Stroke". In: *Journal of Physical Therapy Science* 24.4 (2012), pp. 311–314. DOI: 10.1589/jpts.24.311.

[51] Marc Leman et al. "Activating and Relaxing Music Entrains the Speed of Beat Synchronized Walking". en. In: *PLoS ONE* 8.7 (July 2013). Ed. by Joel Snyder. ISSN: 1932-6203.

[52] Micheline Lesaffre. "Investigating Embodied Music Cognition for Health and Well-Being". In: *Springer Handbook of Systematic Musicology*. Jan. 2018, pp. 779–791.

[53] Sheng Li, Gerard E. Francisco, and Ping Zhou. "Post-stroke Hemiplegic Gait: New Perspective and Insights". In: *Front. Physiol.* 2018.

[54] Pranee Liamputtong. "Participatory Qualitative Research Methodologies in Health". In: 55 City Road, London: SAGE Publications Ltd, Apr. 2020. DOI: 10.4135/9781473919945. URL: https://methods.sagepub.com/book/participatory-qualitative-research-methodologies-in-health.

[55] Valerio Lorenzoni et al. "A Biofeedback Music-Sonification System for Gait Retraining". In: *Proceedings of the 5th International Conference on Movement and Computing*. MOCO '18. Genoa, Italy: Association for Computing Machinery, 2018. ISBN: 9781450365048. DOI: 10.1145/3212721.3212843. URL: https://doi-org.zorac.aub.aau.dk/10.1145/3212721.3212843.

[56] Anat Lubetzky-Vilnai and Deborah Kartin. "The Effect of Balance Training on Balance Performance in Individuals Poststroke: A Systematic Review". In: *Journal of neurologic physical therapy : JNPT* 34 (Sept. 2010), pp. 127–37. DOI: 10.1097/NPT.0b013e3181ef764d.

[57] Christina Ma et al. "Balance Improvement Effects of Biofeedback Systems with State-of-the-Art Wearable Sensors: A Systematic Review". In: *Sensors* 16 (Mar. 2016), p. 434. DOI: 10.3390/s16040434.

[58] Justyna Maculewicz, Lise busk kofoed, and Stefania Serafin. "A Technological Review of the Instrumented Footwear for Rehabilitation with a Focus on Parkinson's Disease Patients". In: *Frontiers in Neurology* 7 (Jan. 2016). DOI: 10.3389/fneur.2016.00001.

[59] Pieter-Jan Maes, Jeska Buhmann, and Marc Leman. "3MO: A Model for Music-Based Biofeedback". In: *Frontiers in Neuroscience* 1 (Dec. 2016).

[60] Stefan Mainka et al. "The Use of Rhythmic Auditory Stimulation to Optimize Treadmill Training for Stroke Patients: A Randomized Controlled Trial". In: *Frontiers in Neurology* 9 (2018), p. 755. ISSN: 1664-2295. URL: https://www.frontiersin.org/article/10.3389/fneur.2018.00755.

[61] Jean Massion. "Postural control system". In: *Current opinion in neurobiology* 4 (Jan. 1995), pp. 877–87. DOI: 10.1016/0959-4388(94)90137-6.

[62] Joan Mcdowd et al. "Attentional Abilities and Functional Outcomes Following Stroke". In: *The journals of gerontology. Series B, Psychological sciences and social sciences* 58 (Feb. 2003), P45–53. DOI: 10.1093/geronb/58.1.P45.

[63] Glen E. Gresham M.D, William B. Stason, and Pamela W. Duncan. *Post-Stroke Rehabilitation*. en. DIANE Publishing, Aug. 2004. ISBN: 978-0-7881-4144-7.

[64] Bart Moens et al. "Encouraging Spontaneous Synchronisation with D-Jogger, an Adaptive Music Player That Aligns Movement and Music". In: *PLOS ONE* 9 (Dec. 2014).

[65] JD Moreland, MA Thomson, and AR Fuoco. "Electromyographic biofeedback to improve lower extremity function after stroke: a meta-analysis". eng. In: *Archives of physical medicine and rehabilitation* 79.2 (Feb. 1998), pp. 134–140. ISSN: 0003-9993. DOI: 10.1016/s0003-9993(98)90289-1. URL: http://europepmc.org/abstract/MED/9473993.

[66] Sara Mulroy et al. "Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke". In: *Gait posture* 18 (Sept. 2003), pp. 114–25. DOI: 10.1016/S0966-6362(02)00165-0.

[67] Lewis M. Nashner, Franklin O. Black, and C Wall. "Adaptation to altered support and visual conditions during stance: patients with vestibular deficits." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 2 5 (1982), pp. 536–44.

[68] John G. Neuhoff. "Is Sonification Doomed to Fail?" en. In: *Proceedings of the 25th International Conference on Auditory Display (ICAD 2019)*. Newcastle upon Tyne: Department of Computer and Information Sciences, Northumbria University, June 2019, pp. 327–330. ISBN: 978-0-9670904-6-7. DOI: 10.21785/icad2019.069. URL: http://hdl.handle.net/1853/61531 (visited on 12/14/2019).

[69] Susan Niam et al. "Balance and physical impairments after stroke". In: *Archives of physical medicine and rehabilitation* 80 (Nov. 1999), pp. 1227–33. DOI: 10.1016/S0003-9993(99)90020-5.

[70] Deborah S Nichols. "Balance Retraining After Stroke Using Force Platform Biofeedback". In: *Physical Therapy* 77.5 (1997), pp. 553–558. ISSN: 0031-9023. DOI: 10.1093/ptj/77.5.553. URL: https://doi.org/10.1093/ptj/77.5.553.

[71] Kyoung Shin Park et al. "Emotional states influence forward gait during music listening based on familiarity with music selections". In: *Human Movement Science* 66 (Mar. 2019), pp. 53–62. DOI: 10.1016/j.humov.2019.03.004.

[72] Gaëtan Parseihian et al. "Comparison and Evaluation of Sonification Strategies for Guidance Tasks". In: *IEEE Transactions on Multimedia* 18.4 (2016), pp. 674–686. ISSN: 1941-0077. DOI: 10.1109/TMM.2016.2531978.

[73] Gaëtan Parseihian et al. "The Process of Sonification Design for Guidance Tasks". In: *Wi:Journal of Mobile Media*. Locus Sonus Issue 9.2 (2015). URL: https://halshs.archives-ouvertes.fr/halshs-01230638.

[74] Gaëtan Parseihian et al. "Exploring the usability of sound strategies for guiding task: toward a generalization of sonification design". In: (Oct. 2013).

[75] Jacquelin Perry and Judith Burnfield. *Gait Analysis: Normal and Pathological Function*. Feb. 2010. ISBN: 978-1556427664.

[76] A. Pollock et al. "Physical rehabilitation approaches for the recovery of function and mobility following stroke". In: *Cochrane Database of Systematic Reviews* 4 (2014). ISSN: 1465-1858. DOI: 10.1002/14651858.CD001920.pub3.

[77] Alex Pollock et al. "Physiotherapy treatment approaches for the recovery of postural control and lower limb function following stroke". In: *Cochrane database of systematic reviews (Online)* 1 (Feb. 2007), p. CD001920. DOI: 10.1002/14651858.CD001920.pub2.

[78] Anna-Maria Raberger et al. "Short-Term Effects of Real-Time Auditory Display (Sonification) on Gait Parameters in People with Parkinson's Disease—A Pilot Study". In: *Biosystems and Biorobotics* 15 (Oct. 2016), pp. 855–859.

[79] Prithvi Ravi Kantan and Sofia Dahl. "An Interactive Music Synthesizer for Gait Training in Neurorehabilitation". In: *Proceedings of the 16th Sound Music Computing Conference SMC 2019*. Proceedings of the Sound and Music Computing Conference. Sound and Music Computing Network, 2019, pp. 159–166. DOI: 10.5281/zenodo.3249297.

[80] Prithvi Ravi Kantan and Sofia Dahl. "Communicating Gait Performance Through Musical Energy: Towards an Intuitive Biofeedback System for Neurorehabilitation". In: *Combined proceedings of the Nordic Sound and Music Computing Conference 2019 and the Interactive Sonification Workshop 2019*. KTH Royal Institute of Technology, 2019, pp. 107–114.

[81] *Reading a IMU Without Kalman: The Complementary Filter*. URL: https://www.pieter-jan.com/node/11.

[82] Pablo Revuelta Sanz et al. "Scenes into sounds: a taxonomy of image sonification methods for mobility applications". In: *Journal of the Audio Engineering Society* 62 (Apr. 2014), pp. 161–171. DOI: 10.17743/jaes.2014.0009.

[83] Patrick Riley, David Krebs, and Rita Popat. "Biomechanical analysis of failed sit-to-stand". In: *IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society* 5 (Jan. 1998), pp. 353–9. DOI: 10.1109/86.650289.

[84] Matthew Rodger, William Young, and Cathy Craig. "Synthesis of Walking Sounds for Alleviating Gait Disturbances in Parkinson's Disease". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society* 22 (Oct. 2013). DOI: 10.1109/TNSRE.2013.2285410.

[85] Rebecca L. Routson, Steven A. Kautz, and Richard R. Neptune. "Modular organization across changing task demands in healthy and poststroke gait". In: *Physiological reports*. 2014.

[86] Ralph Sacco et al. "An Updated Definition of Stroke for the 21st Century A Statement for Healthcare Professionals From the American Heart Association/American Stroke Association". In: *Stroke; a journal of cerebral circulation* 44 (May 2013). DOI: 10.1161/STR.0b013e318296aeca.

[87] Judith Schaechter. "Motor rehabilitation and brain plasticity after hemiparetic stroke". In: *Progress in neurobiology* 73 (June 2004), pp. 61–72. DOI: 10.1016/j.pneurobio.2004.04.001.

[88] Nina Schaffert et al. "A Review on the Relationship Between Sound and Movement in Sports and Rehabilitation". In: *Frontiers in Psychology* 10 (2019), p. 244.

[89] Margaret Schedel et al. "Interactive sonification of gait: realtime biofeedback for people with Parkinson's disease". In: *Interactive Sonification Workshop*. 2016.

[90] Anne Shumway-Cook 1947. *Motor control : theory and practical applications*. Second edition. Philadelphia : Lippincott Williams & Wilkins. URL: https://search.library.wisc.edu/catalog/999918167702121.

[91] Rosalyn Stanton et al. "Biofeedback improves performance in lower limb activities more than usual therapy in people following stroke: a systematic review". In: *Journal of Physiotherapy* 63.1 (2017), pp. 11 –16. ISSN: 1836-9553. DOI: https://doi.org/10.1016/j.jphys.2016.11.006. URL: http://www.sciencedirect.com/science/article/pii/S1836955316300881.

[92] Tino Stöckel et al. "The mental representation of the human gait in young and older adults". In: *Frontiers in Psychology* 6 (2015), p. 943. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2015.00943. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2015.00943.

[93] Sundhedsstyrelsen. *Sygdomsbyrden i Danmark*. 2015. URL: https://www.sst.dk/da/sygdom-ogbehandling/~/media/00C6825B11BD46F9B064536C6E7DFBA0.ashx.

[94] Karthikbabu Suruliraj et al. "A review on assessment and treatment of the trunk in stroke A need or luxury". In: *Neural Regeneration Research* 7 (Sept. 2012), pp. 1974–1977. DOI: 10.3969/j.issn.1673-5374.2012.25.008.

[95] Thomas J. Schmitz [edited by] Susan B. O'Sullivan. *Physical rehabilitation*. Fifth edition. Philadelphia : F.A. Davis, 2007.

[96] Jeremiah J. Tate and Clare E. Milner. "Real-Time Kinematic, Temporospatial, and Kinetic Biofeedback During Gait Retraining in Patients: A Systematic Review". In: *Physical Therapy* 90.8 (Aug. 2010), pp. 1123–1134. ISSN: 0031-9023. eprint: https://academic.oup.com/ptj/article-pdf/90/8/1123/31662101/ptj1123.pdf. URL: https://doi.org/10.2522/ptj.20080281.

[97] M.H Thaut, G.C McIntosh, and R.R Rice. "Rhythmic facilitation of gait training in hemiparetic stroke rehabilitation". In: *Journal of the Neurological Sciences* 151.2 (Oct. 1997), pp. 207–212. ISSN: 0022-510X. URL: https://doi.org/10.1016/S0022-510X(97)00146-9 (visited on 04/01/2020).

[98] Michael Thaut. "Rhythm, music, and the brain: Scientific foundations and clinical applications". In: *Rhythm, Music, and the Brain: Scientific Foundations and Clinical Applications* (Jan. 2005), pp. 1–247. DOI: 10.4324/9780203958827.

[99] Michael H. Thaut and Volker Hoemberg, eds. *Handbook of Neurologic Music Therapy.* New York, NY, US: Oxford University Press, 2014. ISBN: 978-0-19-969546-1 (Hardcover).

[100] Andrés Villa Torres, Viktoria Kluckner, and Karmen Franinovic. "Development of a sonification method to enhance gait rehabilitation". In: *Proceedings of ISon 2013, 4th Interactive Sonification Workshop*. 2013.

[101] Geert Verheyden et al. "Trunk performance after stroke and the relationship with balance, gait and functional ability". In: *Clinical Rehabilitation* 20.5 (2006). PMID: 16774097, pp. 451–458. DOI: 10.1191/0269215505cr955oa. eprint: https://doi.org/10.1191/0269215505cr955oa. URL: https://doi.org/10.1191/0269215505cr955oa.

[102] Bram van der Vlist, Christoph Bartneck, and Sebastian Mäueler. "moBeat: Using Interactive Music to Guide and Motivate Users During Aerobic Exercising". In: *Applied Psychophysiology and Biofeedback* 36.2 (June 2011), pp. 135–145. ISSN: 1573-3270. URL: https://doi.org/10.1007/s10484-011-9149-y.

[103] Greg Welch and Gary Bishop. "An Introduction to the Kalman Filter". In: *Proc. Siggraph Course* 8 (Jan. 2006).

[104] Felix Weninger et al. "On the Acoustics of Emotion in Audio: What Speech, Music, and Sound have in Common". In: *Frontiers in psychology* 4 (2013), p. 292. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00292. URL: https://europepmc.org/articles/PMC3664314.

[105] Rachel L Wright et al. "Metronome cueing of walking reduces gait variability after a cerebellar stroke". In: *Frontiers in neurology* 7 (2016), p. 84.

[106] Bin Yu. "Designing Biofeedback for Managing Stress". PhD thesis. May 2018. DOI: 10.13140/RG.2.2.24670.95049.

[107] Agnes Zijlstra et al. "Biofeedback for training balance and mobility tasks in older populations: A systematic review". In: *Journal of neuroengineering and rehabilitation* 7 (Dec. 2010), p. 58. DOI: 10.1186/1743-0003-7-58.

Appendix A

Neurologic Music Therapy Techniques: An Overview

This appendix provides an overview (mainly based on [99]) of neurologic music therapy techniques relevant to the present study.

A.1 Rhythmic Auditory Stimulation (RAS)

RAS is "a neurologic technique used to facilitate the rehabilitation, development, and maintenance of movements that are intrinsically and biologically rhythmical" [99]. The gait deficits specific to stroke patients have been discussed, and RAS can address several of these [95]. RAS may be instrumental in improving gait velocity, cadence, stride length and gait asymmetry [10]. RAS works on four neurological principles. *Rhythm entrainment* is the ability of the motor system to couple with the auditory system and drive movement patterns. *Priming* is the ability of an auditory cue to stimulate motor neuron recruitment at the spinal cord level, thereby entraining muscle activation patterns in the legs during gait [99]. *Cueing of the movement period* is a principle based on evidence that rhythmic motor synchronization is based on period entrainment rather than phase entrainment [97]. *Step-wise limit cycle entrainment* is the process of entraining a patient's limit cycle (frequency at which gait functions optimally) through a stepwise progression to modulate cadence to approximate premorbid frequencies [99]. RAS guidelines for stroke patients include an emphasis on evenness and symmetry, as well as a focus on heel strike, equitable weightbearing and improvement in gait velocity not at the expense of gait quality [99]. RAS has been found to improve gait velocity, symmetry and cadence in chronic stroke patients [50]. It has also shown higher efficacy in treadmill training than standard approaches at restoring functional gait [60].

A.2 Patterned Sensory Enhancement (PSE)

PSE is "a technique that uses the rhythmic, melodic, harmonic, and dynamic acoustical elements of music to provide temporal, spatial, and force cues for movements which reflect functional movements of activities of daily living, or the fundamental motor patterns underlying these activities" [99]. It is applied to movements that are non-rhythmic by using music gestalts to train the performance of movement gestalts [99]. Its goals are increase physical strength, balance, posture and functional skills [98]. In PSE exercises, a musical pattern supporting the spatial, temporal and force aspects of a movement are repeated to shape and facilitate the movement over time. It can similarly be used to used to train functional movement sequences with a variety of neurological and geriatric populations, with numerous examples of relevant motion exercises listed in [99].

A.3 Therapeutic Musical Instrument Performance (TIMP)

TIMP "utilizes musical instruments to help patients to exercise impaired motor function and regain functional patterns of movement" [99]. It helps overcome unhealthy compensation strategies while enhancing strength, endurance and movement control. It targets some of the typical deficits resulting from stroke, such as weakness, spasticity, limb paresis, etc. Rhythmic patterning of movement leverages the advantages of rhythmic cueing and facilitates repetition, which is key to motor rehabilitation [13]. Engaging with music in this way also facilitates additional supportive benefits such as enhanced motivation and positive emotional states, creating feelings of accomplishment, collaboration and enhanced motivation to work on therapeutic goals [99]. Percussion instruments are most accessible as they can be simply played by non-musicians. If songs are used, they should be structurally simple as well as familiar to the patient. Patients with attention problems may not be able to sing along and perform the instrument at the same time, in which case a single repetitive melody may be better. If the exercises are done in groups, the patients should be compatible in terms of rehabilitation needs and endurance. Sessions should begin with a warm-up followed by the actual exercise [99].

Appendix B

Compact Music Representation

The custom music representation scheme compactly encodes music pieces as an ordered sequence of musical passages containing a melody and an underlying chord progression. We now delve into the specifics of the representation.

B.1 Design

Streamlining Decisions: The representation is inspired by MIDI notation, but is stripped down in several ways for simplicity and speed of encoding.

- **Time:** The time signature is always 4/4, to suit the music to binary rhythmic activities such as walking and to simplify the representation.
- **Track Count:** The total number of instrument tracks is always fixed at eight and comprises the essential percussive and melodic elements present in popular music.
- **Common Information:** All music information except melodic and harmonic structures is kept common among all music pieces, and coded into the music sequencing functionality of the *biofeedback application*.
- **Fixed Passage Length and Number:** A piece of music is represented with a limited number of passages of fixed musical length, in a specific sequence. This allows rapid encoding and ordering of passages on a single interface screen.
- **Compactness:** Note and velocity information for each sixteenth note interval is separately represented by a *single digit*. Such a compact representation allows longer musical passages to be displayed and modified on one interface screen and stored inexpensively.

Passage Number, Length and Order: The maximum number of passages in a music piece is **five**. Each passage is **four bars/measures** in length and represented at the sixteenth note level. Thus, every passage has 64 temporal locations. The passages can be ordered in any fashion and the total music piece is a combination of 24 passage repetitions.

Note and Velocity Information: While MIDI represents this information through note on and note off messages with timestamps, the simplified representation captures melody and chord information as a sequence of digit values at each temporal location throughout a passage. Note numbers in MIDI are converted to single digits by allowing the user to specify a global **tonic** note number and a passage specific **scale** (major/minor...). Note values at every temporal location are thus completely contained by two digit streams - **scale degree** and **note octave** relative to the tonic. Note velocity on the other hand is simply a 0-9 value at each temporal location that specifies the triggering intensity of the corresponding scale degree. A similar scheme is employed for chord progressions, wherein the scale degree digit captures the root note of the chord. Note octave is replaced by a **chord type** digit, which serves as a code for a choice of chord types, including major, minor, dominant, etc. Chord velocity functions in the same way as melody velocity. Note that velocity only represents note/chord triggering at that temporal location and *not* note duration. Thus, only note onset is captured unlike MIDI, where notes can be pressed for long or short periods. Thus, each passage is represented in its entirety by six 64 digit sequences, three for melody information and three for chord information. An entire song, on the other hand, is represented by five such passages and scale codes, a global tonic, song name and passage order. Of course, it is acknowledged that such a minimalist representation has significant limitations in being able to represent a piece of music with fidelity. It must be stressed that the purpose of the representation is not to exactly capture and reproduce the original piece of music, but to enable the production of a recognizable and enjoyable instrumental version for training and biofeedback purposes.

B.2 Implementation

The functional schematic of the music encoder is shown in Fig. B.1.

Passage Encoding: The interface is organized into two tabs pertaining to *Melody* and *Chord* information encoding (see Fig. 5.4), using the JUCE *TabbedComponent* class [1]. The current passage being encoded is set using the 'Passage Number' JUCE slider, whose range is between 1 and 5. Each passage is 4 bars in length,

[1] TabbedComponent Class Reference. url: https://docs.juce.com/master/classTabbedComponent.html.

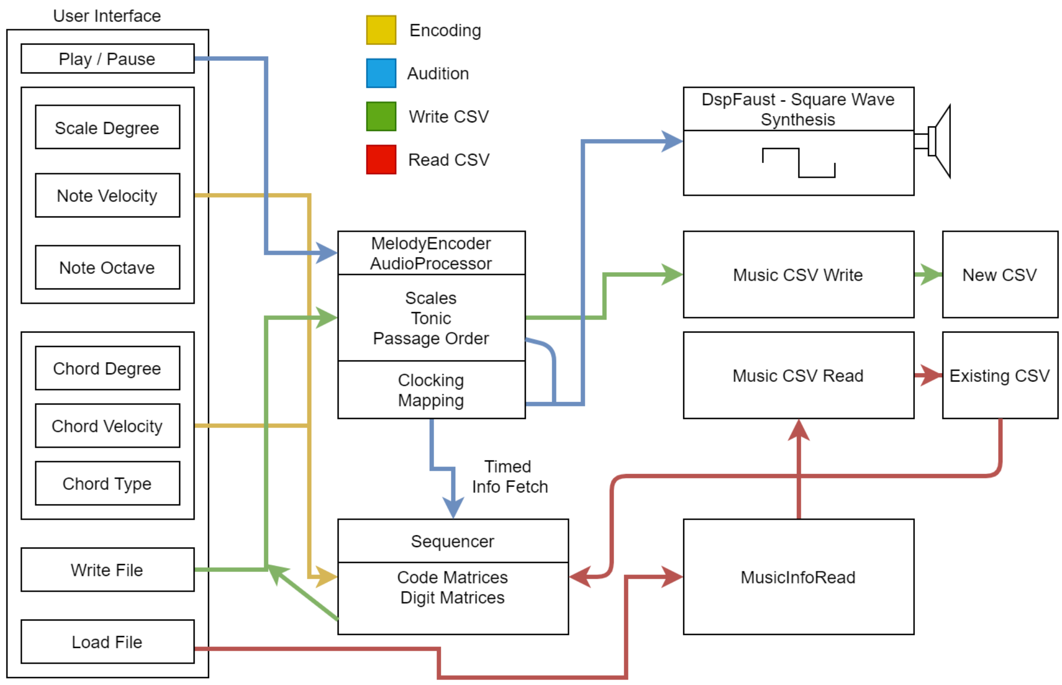

Figure B.1: Music Encoder Functional Schematic.

represented by single digits at the sixteenth note level by three sets of information streams. For the *Melody* component, these are *'Scale Degree', 'Velocity' and 'Note Octave'*, and for the *Chord* component these are *'Scale Degree', 'Velocity' and 'Chord Type'*. Each information stream is thus represented by 64 digits.

To make these 64 digits visible and easily editable on a single interface screen, they are divided into 8 eight digit *integer codes* that can be modified using arrays of JUCE sliders. Thus, each integer code represents a period of two quarter notes within the four bar passage. The sliders themselves do not have the resolution to allow precise mouse dragging for specific codes, so the slider text boxes are kept visible so that the user can type the codes in manually, as shown in Fig. 5.4. Slider value changes trigger lambda functions that store the newly entered value in the correct element position of its corresponding matrix in the *Sequencer* class. There are six 5×8 *code matrices* corresponding to 8 integer codes for each of the five passages, for three melody/chord information types. The same three arrays of sliders are used for both melody and chord information, but they map to different matrices depending on the active tab. After being stored in these matrices, the integer codes are split back into single digits and stored in a corresponding set of 5 $\times$ 64 *digit matrices*, also in the *Sequencer* class. The code matrices are used for CSV storage purposes, while the digit matrices are used for real-time auditioning. The interface is dynamic, meaning that the sliders and boxes change to reflect present stored values when the user switches between passage indices. A button on the interface allows the user to duplicate the present passage in its entirety to the next index, saving time if the difference between successive passages is not large.

File Metadata: The interface allows the user to fill in important metadata such as song name in a JUCE TextEditor [2], song tonic (note number) and a scale code for each passage (e.g. 0 = major scale, 1 = natural minor scale, etc.).

Passage Order: The passages can be ordered by their indices to define a 24 passage-long sequence in which they are desired to play back. As shown in Fig. 5.4, this sequence is displayed just above the slider arrays and using the 'Next' and 'Previous' buttons, the 'I' cursor can be moved to the desired sequence position and a passage index (1-5) can be entered in the attached TextEditor and stored with the 'Add' button. The sequence and 'I' cursor are displayed using JUCE Labels [3], and the order array is stored in *MelodyEncoderAudioProcessor*.

Read/Write CSV: The UI allows the user to store the encoded passages as a music CSV, following which a function in *MelodyEncoderAudioProcessor* stores the integer code matrices and metadata in a CSV file that can be read and synthesized by the biofeedback application. Conversely, the 'Load' button opens a JUCE FileChooser [4] that allows a CMR-compatible CSV to be imported into the encoder application for preview and modification. Loading a file automatically updates the code and digit matrices so the data in the loaded file reflects in the encoding interface.

Auditioning: The application allows the user to audition a passage in real-time after codes are entered and stored. Clicking the play/pause button toggles the playback of the selected passage in an endless loop. The clocking mechanism for this process is identical to that used in the biofeedback application, and music info for playback is fetched from the digit matrices which are addressed using the within-passage sixteenth note counter in *Sequencer*. The values are regularly mapped to a simplified version of *DspFaust* at every clocking interval, which reproduces the melody and chord tracks using simple square wave oscillators. The tempo is kept fixed at 120 BPM. The UI has a a special label 'CURRENT' which moves from left to right in synchronous fashion with the playback as a crude substitute to a progress bar.

[2]TextEditor Class Reference. url: https://docs.juce.com/master/classTextEditor. html.

[3]Label Class Reference. url: https://docs.juce.com/master/classLabel. html

[4]FileChooser Class Reference. url: https://docs.juce.com/master/classFileChooser. html.

B.3 Melody/Chord F0 Calculation

Moving now to the biofeedback application, the integer codes encoded in the CMR format are converted to their MIDI note equivalent in the *Sequencer* and then to a fundamental frequency value (f0) in Hz in *DspFaust*. We now discuss how the f0 signals are computed for melody instrument synthesis control. In both cases, the f0 signal is calculated by first computing the MIDI note number from the available CMR information *in JUCE*. This note number is then mapped to *DspFaust* and then converted to Hz using FAUST library function *ba.midiKeytoHz* for audio synthesis. The note number is computed in the *Sequencer* class at every music clock interval (16th note). The following information is stored in the form of look-up matrices in *Sequencer*:

- **Scale Intervals:** This matrix stores multiple music scales in its rows (e.g. major, minor, harmonic minor, etc.) while its columns represent degrees of the scale as the number of semitone intervals between that degree and the tonic/root note. For example, this row represents the major scale:

  $$-1, 0, 2, 4, 5, 7, 9, 11, 12, 14$$

  Array index 1 (second element) represents the tonic, while array index 8 (ninth element) represents a 12 semitone interval or an octave. Depending on the active scale and present note degree, a semitone interval relative to the tonic can be fetched from this matrix.

- **Chord Intervals:** While the root note of a chord can be represented by the scale degree, the semitone intervals between the individual notes of the chord are determined by the chord type (e.g. major, minor, dominant 7th, etc.). As the number of voices is fixed, the semitone intervals for each note in the chord, relative to the root note, are stored for all chord types as follows:

  short chord_2nds[8] = 4, 3, 4, 3, 4, 4, 3, 7;

  short chord_3rds[8] = 7, 7, 7, 7, 7, 7, 7, 12;

  short chord_4ths[8] = 12, 12, 11, 10, 10, 14, 14, 19;

  As is evident, the *second* note usually represents an interval corresponding to a major or minor third. The *third* note is generally the perfect fifth and the *fourth* is either the octave note, major/minor 7th or 9th. The intervals in each array are ordered such that selecting a particular chord type index will yield the note interval combination corresponding to the chord represented by the index. E.g. major chord will take the first element of each array yielding semitone intervals 0 (root note), 4 (major third), 7 (perfect fifth), and 12 (octave).

Using this tables, the MIDI key numbers are computed as follows:

- **Monophonic (Melody, Bassline):** The preliminary MIDI note number for the monophonic tracks is calculated from the *activeNoteDegree* and *octave* as follows:

```
int activeInterval = scales[activeScale][activeNoteDegree];
int preliminaryMidiValue = tonic + 12 * octave + activeInterval;
```

  activeInterval converts the scale degree into the number of semitones above the tonic, and this is used to compute the preliminary MIDI note number for that sixteenth note duration.

- **Polyphonic (Low Chord, High Chord):** The MIDI key numbers for all four notes of the chord track are calculated as follows:

```
int activeDegree = scales[activeScale][chordDegree];
short incs[3] = { 0 };
incs[0] = chord_2nds[chordType];
incs[1] = chord_3rds[chordType];
incs[2] = chord_4ths[chordType];
```

  The root note is calculated here just like in the monophonic case, and the increments are added to yield the preliminary MIDI note numbers for for that chord.

Note Value Restriction: It may be noted that the term *preliminary* is used for all derived MIDI note numbers, and this is because there is one final processing step. The computed note numbers depend on the tonic note number as well as scale and chord interval values. Both these factors can take a relatively wide range of values, resulting in the respective tracks occupying a potentially wide fundamental frequency range depending on how the music was originally encoded. This can cause several problems:

- Synthesis algorithms may not be capable of generating uniform timbres in different note registers.
- Equalizer settings may not be equally suitable if f0 ranges are not known. For example, a parametric boost at 100 Hz with 2 octave bandwidth to the bassline track may be completely ineffective if its fundamental frequency never goes below 220 Hz.

- Instruments may unpredictably and uncontrollably mask one another in the frequency domain if they are play, or have too many common spectral components in the same register.
- The *fullness* of the ensemble may be compromised if there is no spectral information in important frequency ranges such as the upper bass (100-200 Hz), lower mid-range (200-500 Hz) and high mid-range (2 KHz - 5 KHz).

The simplest solution is to restrict the preliminary note values of all melody tracks into certain bounds after computation. This ensures that the instruments remain within registers where they sound best, do not clash spectrally and result in a full-sounding and balanced ensemble. The principle is that instruments are assigned certain *MIDI note limits*, and the preliminary MIDI note number is checked against the limits for that track when they are computed. If the note number lies within the limits, it remains as-is. If higher than the upper limit, a function determines *how many octaves* higher the note is, as transposes it down until it is within the note limits. Similarly, the note is transposed up if it is below the lower limit. As notes across octaves are musically equivalent, it does not alter the essence of the note or chord. Hence, the unpredictability resulting from the tonic value and note octaves is accounted for. The limits themselves are selected empirically based on what works best with the chosen synthesis methods, but the general principle is to separate instruments into different registers where their timbres fit best with one another. The note limits for each melody instrument are as follows:

- Bassline: 24 - 35
- Main Melody Synth: 10 - 80
- Main Chord Synth: 51 - 71
- High Chord Synth: 63 - 77

Appendix C

Music Generation

C.1 Iteration 1

C.1.1 Music Playback Controls

The working of the music playback controls depicted in Figure 5.7 is explained as follows:

- **Play/Pause Toggle, Stop:** JUCE *TextButton* objects are used for playback control. These controls act directly on the *DspFaust* object, starting and stopping it when playback is initiated and terminated respectively. Pausing the music also stops *DspFaust*, but the song progress is maintained as the musical time counters in *Sequencer* (next subsection) maintain their states. When the Stop button is pressed, the counters are all reset to zero, thus these buttons function similarly to any audio player. Music clocking and sequencing only occur when playback is enabled, hence freeing up computational power when it is disabled.

- **Tempo Slider:** Sets the tempo of the music in BPM. When the value is changed, its lambda function resets the 16th note interval of the music clock in milliseconds to manipulate the frequency with which pulses are triggered, speeding up or slowing down the entire music sequencing process and subsequent music playback without any artifacts. The tempo in BPM is also mapped to *DspFaust* to make tempo-appropriate synthesis and effect adjustments, such as instrument envelopes, echo effect times and so on.

- **Tap Tempo:** This provides an easier tempo-setting method, as it is easy for the operator of the application to click the mouse in synchronization with a periodic movement (e.g. gait) and have the music set its tempo accordingly, in comparison to precisely setting a slider to the correct value. The functionality works by timing the duration between two successive button presses and considering that as one *beat* duration as it is the interval most commonly tapped. From this beat interval, the tempo is derived and the slider is set accordingly.

- **Load File:** Playback is not possible unless a CMR CSV is loaded. Pressing the Load button opens a JUCE *FileChooser* that is used to browse the local computer for CSV files. When a valid file is loaded, its absolute path is passed from the FileChooser to the *MusicInfoRead* object in *Sequencer*, which uses the helper class *CSVReader* to load all the metadata and encoded music passages into its matrices and begins playback automatically. The functionality used for file loading is identical to that used in the encoder application.

- **Track Mute Buttons:** An array of JUCE *ToggleButton* objects is used to control the density of the music ensemble by muting different instruments. The purpose of this is to enable the therapist/instructor to manipulate the individual instruments based on the demands of a situation. For example, a patient may find eight simultaneous instruments over-stimulating, or it may not be possible to pay attention to a particular element in the presence of other distracting elements, or it may be easier to deduce the rhythm in a sparse arrangement. These *ToggleButtons* map directly to the toggle buttons in the master section of *DspFaust* (discussed in a later subsection) which mute the respective tracks.

- **Change Rhythm:** It is possible for the instructor/therapist to select a suitable percussive rhythm for a song/session from a selection of about ten. This does not affect the melody tracks, and only changes the pattern played by the drums. The rhythms range from simple dance grooves to march beats, reggaeton and rock beats. They vary in rhythmic complexity, and can be adjusted to suit patients with varying degrees of cognitive and rhythm-following ability. The control works by incremented the selected rhythm in the *Sequencer*, which fetches velocity information for all percussive instruments from the appropriate matrix locations in *PercPatternLibrary*, the class where all rhythm-related information is stored. This is explained in subsequent subsections.

C.1.2 Music Synthesis

With the goal of computationally light music synthesis firmly in mind, the instruments are generated in FAUST using methods that are algorithmically and principally simple, combining subtractive synthesis, FM synthesis, simple waveshapes and Karplus Strong string synthesis. The algorithms are inspired by classic synthesizers such as the Roland 808, 909 and configurations demonstrated by music producers on Youtube. This process is streamlined by writing common FAUST functions that can produce different timbres from different input arguments. For example, the bass drum and the snare drum are created by distinct configurations of sine waves and filtered white noise. The individual synthesis methods are discussed as follows:

- **Bass Drum:** (audio demo: 1.1.1 in Media Links) This is realized by a dedicated percussion synthesis function, as a combination of filtered white noise (separately sourced as discussed earlier) and a sine wave whose frequency is modulated by its amplitude envelope. The idea is that the sine component contains the bulk of the low frequency content of the drum and has a slow decay, while the noise component captures the beater at the outset of every hit and decays quickly. The noise and sine components thus have different envelope functions, although these are simultaneously triggered. Both these envelopes are FAUST *en.are* functions with a constant attack time of 0.001 sec. The noise and sine release times are function arguments, experimentally adjusted to 0.04 sec and 0.26 sec respectively. The attack and release phases are exponential in shape, suitable for snappy-sounding percussion. The instantaneous frequency of the sine component is simply its envelope multiplied by the peak frequency (225 Hz). The waveforms are mixed at different amplitudes (sine gain exceeds noise gain by 13 dB), and this sub-mix is duplicated to a parallel buss with a FAUST cubic soft clipper effect *(en.cubicnl)* with gain 0.5. This distorted component is mixed back at a level 33 dB below the sine component, and this sum is passed through 2nd order resonant low-pass and high-pass filters with cutoff frequencies 20 Hz and 3500 Hz respectively. The result is a monophonic bass drum simulation.

- **Snare Drum:** (audio demo: 1.1.2 in Media Links) The same function is used as for the bass drum but with different parameters. The noise and sine release times are 0.23 sec and 0.24 sec respectively. The peak sine frequency is 200 Hz, and the two components are mixed at the same level. The distorted component is mixed in 19 dB lower, and the final low-pass and high-pass cutoff frequencies are 100 Hz and 15000 Hz respectively.

- **Hi-hat Cymbal:** (audio demo: 1.1.3 in Media Links) The hi-hat cymbal is synthesized by combining an inharmonic FM component with a filtered noise component, with the idea of creating a treble-dominant high frequency composite waveform with a metallic timbre. The noise component is simply white noise (separately sourced as discussed above). The FM component is generated using the FAUST *sy.fm* function, which has a carrier at 14903 Hz modulated by sine waves at 20715 and 72502 Hz (aliased components) at modulation indices of 8516 and 8516 respectively. The two components are summed at equal levels, and the sum is multiplied by the overall envelope, a FAUST *en.are* function with attack 0.001 sec and release 0.18 sec for open and

0.54 sec for closed hi-hat simulations (velocity signal > and < 5 respectively). The last processing step is a high pass filter with cutoff frequency 7000 Hz, and the result is a monophonic hi-hat simulation which is panned to the left of the stereo image.

- **Main Chord Synth:** (audio demo: 1.1.4 in Media Links) This comprises four notes/voices that are triggered simultaneously by a single velocity signal. Their f0 values are computed from the MIDI key values from JUCE using the *ba.midiKeytoHz* library function. The instrument choice is a piano simulation using filtered rectangular waves with different duty cycles. We first discuss the synthesis of a single note. It is not possible to directly synthesize these using FAUST functions, so the standard *ba.pulsen* impulse train function with variable impulse length is used to simulate different duty cycles. A custom helper function takes wave frequency and duty cycle percent as arguments, converting them into the appropriate rectangular wave. Using this helper function, the single note piano synthesis function generates three rectangular waves with duty cycles 10%, 33% and 66%. These are summed and multiplied by the amplitude envelope, a FAUST *en.ar* envelope with attack time 0.001 sec and release time 10 sec, raised to the power 6 to change its shape from linear to exponential. The result is passed through a second order Butterworth low-pass filter *fi.lowpass* whose cutoff frequency is envelope controlled, and calculated as follows:

```
freqEnv = en.arfe(0.001,1.6,0.4,trigger) : si.smooth(ba.tau2pole(0.0001));
cutoff = (freqEnv + 0.01) * 4000 * freq / 600 * (1 - min(freq,1000)/2000)
: limit(20,20000);
```

  The frequency envelope is a FAUST *en.arfe* function which is the same as the exponential envelope but the final value converges to the final value (f = 0.4 in this case) instead of zero. This allows control over the cutoff frequency of the filter as the note dies away. The result of this envelope is a filter that is 'open' at the note onset but closes as the note progresses. The output of the function is a single mono piano note. To create a greater sense of stereo width, two such mono notes with slightly different frequency (0.5 Hz offset) for each note f0, and panned L and R respectively. This is a common 'pseudo-stereo' creation practice with good phase performance when the playback is collapsed to mono. An identical procedure is followed for all four notes of the synth, which are summed into a stereo signal.

- **Bassline:** (audio demo: 1.1.5 in Media Links) The bassline is generated using FM synthesis. Its f0 value is sent to the synthesis function along with the trigger and velocity. The *sy.fm* function is used for synthesis, with a sine carrier at f0 and 5 sine modulators, each at a frequency three times the last (the first at $3 \cdot \mathrm{f0}$). The modulation indices are controlled by the amplitude envelope, a square root *en.ar* with a = 0.001 sec and r = 0.6 sec above 120 BPM. A parallel distortion buss is created using *ef.cubicnl* with gain 0.7, which is mixed in at a level of -24 dB. The output is a mono bassline which is center-panned.

- **Main Melody Synth:** (audio demo: 1.1.6 in Media Links) This captures the main melodic motif of the music piece, and is generated using the same piano simulation as each note of the main chord synth, only with the addition of a custom dotted quarter-note echo effect. The synth itself is mono and center-panned but the echo effect has echo times that are different on the left and right channels.

- **High Chord Synth:** (audio demo: 1.1.7 in Media Links) This is principally similar to the main chord synth with the same number of simultaneous notes and stereo width increase, only different in that it is synthesized using a custom Karplus-Strong string simulator in a higher register. It is implemented using a FAUST fractional delay with 4th order Lagrange interpolation and a simple 2-sample FIR averaging filter in the unity gain feedback loop. The note fundamental frequency is pre-multiplied by an empirical constant 1.0116 to compensate for the frequency reduction incurred by the phase delay of the feedback filter, and the delay length is set accordingly. The model is excited using white noise passed through an *en.ar* envelope with a = 0.001 sec and r = 0.001 sec, controlled by the track envelope trigger. Each of the four notes is synthesized in this way, and the same pseudo-stereo method is applied as in the main chord synth.

- **Crash Cymbal:** (audio demo: 1.1.8 in Media Links) The crash cymbal is synthesized in a nearly identical fashion to the hi-hat with a few key differences. Its release time is more gradual at 1.8 sec. The FM carrier is at 2100 Hz, and the modulators are at 6500 and 9543 Hz respectively. The FM level is 1 dB lower than the noise level, and the track is panned to the right of the stereo image.

C.2 Iteration 3 - Individual Variant Samples/Synthesis

In the final iteration, each instrument has three sonic variants. Drum tracks are sample-based and melody tracks are synthesized. All sample-based sounds are created in REAPER, manually processed and gain-adjusted prior to export.

- **Bass Drum:** (audio demos: 3.1.1 in Media Links)

  - *Variant 1:* An acoustic bass drum sample from Toontrack EZDrummer 2 - timbrally mellow.
  - *Variant 2:* An electronic bass drum sample synthesized using Spectrasonics Omnisphere 2.
  - *Variant 3:* Similar to Variant 1 but processed differently for a thicker and more aggressive sound.

- **Snare Drum:** (audio demos: 3.1.2 in Media Links) This is sample-based, with three different sounds at different hit intensities used to cover the entire velocity range of each variant. They have distinct envelope shapes and spectral evolution, and allow greater expressiveness and realism compared to only one sample.
  - *Variant 1:* An acoustic snare drum from EZDrummer 2. The transient is emphasized for percussive effect.
  - *Variant 2:* An electronic clap sound from Omnisphere 2.
  - *Variant 3:* A different acoustic snare drum from EZDrummer 2 with the 'body' of the drum sound emphasized through processing.

- **Hi-Hat Cymbal Role:** (audio demos: 3.1.3 in Media Links) In addition to three samples per variant the samples also include variety with respect to open-closed, bell-edge etc.. These articulations are triggered by dividing the velocity range into three parts (1-3, 4-6, 7-9) and treating each part as its own full velocity range.
  - *Variant 1:* Acoustic hi-hat from EZDrummer 2. Articulations include closed hi-hat, hi-hat pedal and open hi-hat.
  - *Variant 2:* Ride cymbal from EZDrummer 2. Articulations include regular hits at different intensities and bell hits.
  - *Variant 3:* Marimba physical model from FAUST (pm.marimba), with f0 at 800 Hz, maximum strike sharpness and an upper cutoff frequency of 5000 Hz.

- **Main Chord Synth:** (audio demos: 3.1.4 in Media Links) The stereo widening technique used in Iteration 1 and 2 (frequency-offset oscillators hard-panned left and right) is replaced by a simpler and more traditional strategy which pans the four notes to different locations across the stereo image. This is superior to the previous strategy as it ensures perfect mono-compatibility without the 'chorusing' effect that the frequency offset introduced when the channels were summed to mono. Tempo-based temporal envelope stretching is done in all cases.

- *Variant 1:* Same as Iteration 1 and 2.
- *Variant 2:* This is a mellow xylophone-like timbre generated using a versatile custom FM synthesizer function. This function has separate envelopes for the modulation index and the amplitude envelope, with complete control over all time parameters. The modulation index envelope has fixed and variable component to allow precise control over spectral evolution, and the amplitude envelope shape can be chosen from an array of choices. The modulator function is a triangle wave at the chosen frequency multiple of the sine carrier.
- *Variant 3:* This is a clean electric guitar timbre simulated using a FAUST library function that simulates the Casio CZ resonant trapezoidal oscillator. It has a constant resonant frequency factor relative to f0. The amplitude envelope decay is exponential in shape.

- **Bassline:** (audio demos: 3.1.5 in Media Links) Tempo-based temporal envelope stretching is done in all cases.
  - *Variant 1:* Same as Iteration 1 and 2.
  - *Variant 2:* The same custom FM synth function mentioned above, with parameters adjusted to create a clean mellow bass sound. Its note value range is adjusted so it plays one register above the typical bass registers, which is more suitable for march beats.
  - *Variant 3:* Custom FM synth with parameters adjusted to create a punchy distorted bass timbre suitable for dance music.

- **Main Melody Synth:** (audio demos: 3.1.6 in Media Links) Tempo-based temporal envelope stretching is done in all cases. Dotted stereo echo effect is used in all three cases.
  - *Variant 1:* Uses the same trapezoidal oscillator function from FAUST as mentioned before, but with the resonant frequency factor modulated by the amplitude envelope to create a more distinct timbral evolution.
  - *Variant 2:* Similar xylophone timbre to Variant 2 of the main chord synth.
  - *Variant 3:* Same custom FM function with distinct modulation and envelope parameters to simulate a 'trumpet'-like timbre, which is generally more aggressive and bright-sounding than Variant 2, with more prominent upper mid-range frequency components.

- **High Chord Synth:** (audio demos: 3.1.7 in Media Links) This track uses identical stereo widening to Iteration 1 and 2.
  - *Variant 1:* Same as Iteration 1 and 2.

  - *Variant 2:* Warm 'pad' sound obtained using triangle waves for each note with *no amplitude envelope*.
  - *Variant 3:* Bright pad sound obtained using sawtooth waves with no amplitude envelope.

- **Crash Cymbal:** (audio demos: 3.1.8 in Media Links) The variants are three different samples from EzDrummer 2.

Appendix D

Interview Questionnaires

This appendix contains the questionnaires used in the expert interviews and music producer survey conducted as part of the final evaluation.

D.1 Expert Interview

After watching each of the five sonic interaction videos, the participants were interviewed in a structured manner. The questions were tightly related to the final problem formulation (Chapter 3), and were posed as follows:

D.1.1 Therapist Perspective - Usefulness

- Does the sensing system effectively capture the movement patterns relevant to this training activity?
- Does the music feedback effectively convey patient movement information to the therapist during training?
- Does the music feedback convey useful information about the patient that would not be available by conventional means (e.g. vision)?
- To which specific impairment level of patients would this interaction be most relevant and useful?

D.1.2 Therapist Perspective - Usability

- Is this interaction suitable/adaptable to existing training protocols?
- Please comment on the feasibility of this interaction from a real-life practical standpoint.
- Are there other training situations where such an interaction could be used?

D.1.3 Patient Perspective - Usability

- Does the feedback make sense in the context of the action that caused it (intuitiveness, meaningfulness)?
- Would the feedback be easy for the typical patient to perceive?
- Is the feedback given in a timely manner (not too early/too late compared to the movement)?
- Would the patient's cognitive load be too high while doing the task?
- Would the individual system adjustments help in covering the range of patient disabilities? What more can be adjusted to suit individuals?

D.1.4 Patient Perspective - Usability - At End

- Is the music enjoyable and motivating?
- Would the hardware be comfortable to wear?

D.2 Music Producer Survey

D.2.1 Personal Information

- Please enter your age.
- Please enter your gender.
- How would you describe your expertise as a music producer? *(Never Produced Music / Amateur / Intermediate / Advanced)*

D.2.2 Style-wise 7 Point Scale Ratings (1 = Strongly Disagree, 7 = Strongly Agree)

- S1. The synthesized rhythm has a prominent pulse and strong groove.
- S2. The synthesized rhythm is overly static and repetitive.
- S3. The synthesized rhythm has the appropriate quantity of expressive and unpredictable variations.
- S4. The main melody sounds computerized and lifeless.
- S5. The main melody is emotionally expressive.
- S6. The main melody has a rich and interesting timbre.

- S7. The supporting instruments add richness and harmonic depth to the music.
- S8. The supporting instruments are of little to no musical value.
- S9. The instruments interact well with each other to create a full sounding arrangement.
- S10. The individual instruments clash with each other, causing the mix to sound muddy.
- S11. The individual and combined instrument timbres are appropriate for the recreation of this style of music.
- S12. The overall mix is optimally balanced in terms of musical dynamics and sonic 'punch'.
- S13. The overall mix is optimally balanced in terms of frequency spectrum (separation between instruments/overall tonality).

D.2.3 Style-wise Short/Long Answers

- Please describe the synthesized style of music in ONE word.
- Please list two or three key production improvements you would make to the overall music if it were in your hands.