A PHYSICAL THERAPY SYSTEM FOR ENCOURAGING SPECIFIC MOTION IN WRIST REHABILITATION EXERCISES

A Thesis
Presented to
The Academic Faculty

by

Brittney A English

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2018

A PHYSICAL THERAPY SYSTEM FOR ENCOURAGING SPECIFIC MOTION IN WRIST REHABILITATION EXERCISES

Approved by:

Professor Ayanna M Howard, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Linda Wills
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 7 August 2018

To my family, my parents, brother, and my niece.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Ayanna Howard, for her guidance, direction, and mentorship. I would like to thank my thesis committee, Dr. David Anderson, Dr. Irfan Essa, Dr. Aaron Lanterman, and Dr. Linda Wills, for their time and input. I would also like to thank my family, my parents, brother, and niece, for their love and support. Finally, I would like to thank my HumAnS labmates, friends, and Georgia Tech colleagues for their friendship, support, and feedback on my project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

To aid the readers as they traverse this thesis, Figure 1 shows the chapter dependencies of this work.

The goal of physical therapeutic exercises is to increase proficiency of a motor skill. Physical therapeutic exercises are commonly prescribed to individuals with motor disabilities. During the physical therapy process, individuals will usually practice once a week with the assistance of a clinician and six days a week in isolation. When practicing exercises in the presence of a skilled clinician, an individual receives several benefits including: (1) real-time feedback on accuracy of motions; (2) real-time adaptations to an exercise plan that accommodates the client's skill level and performance; (3) social interactions that increase participant engagement; and (4) positive feedback that increases morale. These benefits are not realized by the client when practicing exercises in isolation, causing clients to struggle to comply with therapeutic regimens at home [11, 26, 37, 41, 47].

The purpose of this research is to increase user outcomes through personalized training sessions that have been created via artificial intelligence to best fit the needs of the end user. The system that we have developed will assist clients with the at-home portion of physical therapy by mimicking the four previously mentioned benefits that a clinician provides during therapeutic exercise sessions. First, we designed a passive exoskeleton with a rehabilitation gaming suite that encourages therapeutic motions. Then, we verified its ability to increase engagement while completing therapeutic exercises. Next, we verified the ability of our system to encourage accurate

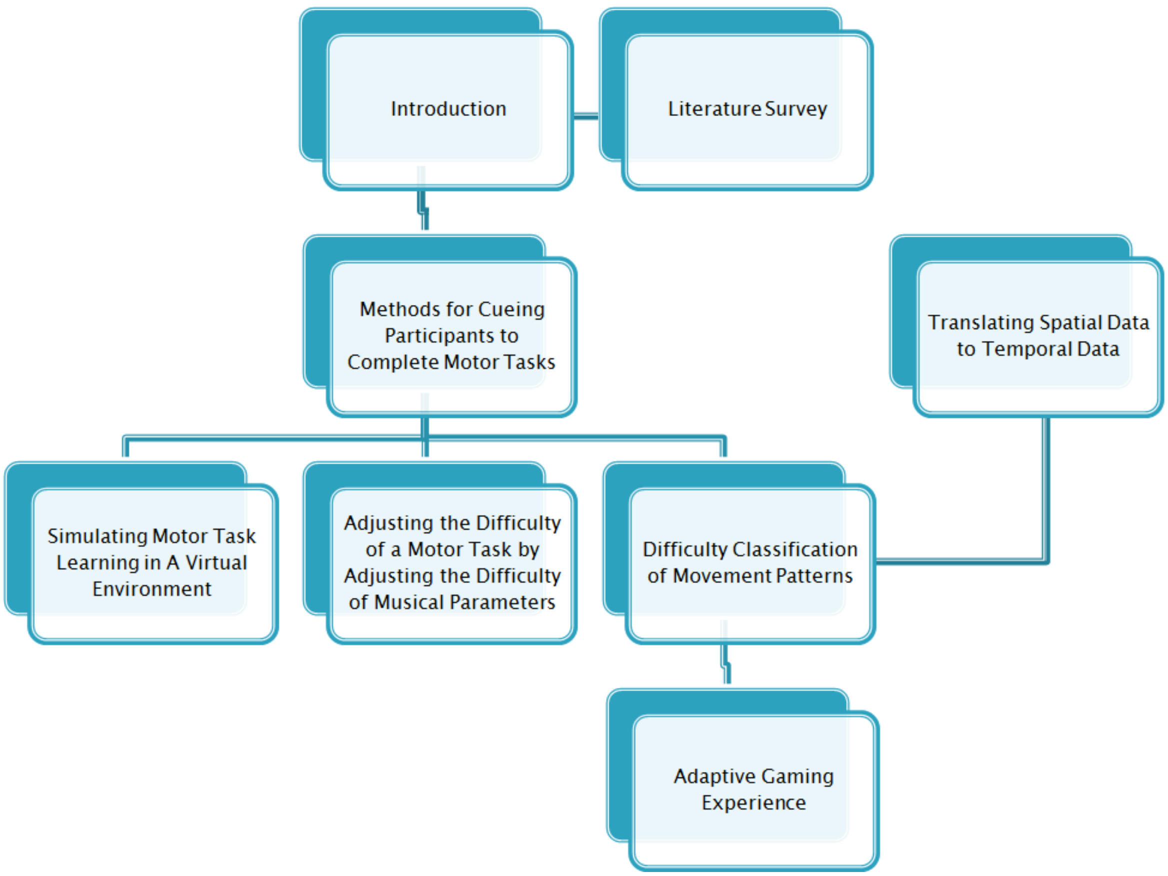


Figure 1: A visual representation of the chapter dependency.

therapeutic motions. Once the games were optimized, we created a virtual environment and validated that the system could be simulated virtually. Then, in a virtual environment, we adjusted the difficulty of the motor task via adjusting the difficulty of musical parameters, per music theory definitions. Finally, we used machine learning techniques to classify the task difficulty of similar, commercially available games and used these classifications to automatically adapt the difficulty of our game. The final product was validated with our target audience (elderly adults and stroke survivors) and we found that the adaptations were effective in promoting improved learning of the motor task. Furthermore, the target audience reported enjoying the system, even more than younger adults had in previous experiments.

CHAPTER II

LITERATURE SURVEY

2.1 Motor Learning for Stroke Survivors

Physical therapy is a common treatment for the rehabilitation of hemiparesis, or the weakness of one side of the body. Stroke is a common cause of hemiparesis [21]. Patients with stroke-induced hemiparesis may have reduced muscular strength to a vital body part, e.g. their dominant hand. However, through physical therapy exercises, patients can regain strength and improve their ability to use their dominant hand when performing daily activities. Unfortunately, physical therapy, in general, is a painful process that patients do not enjoy participating in. Furthermore, the attitude of the patient directly correlates to their compliance and success during physical therapy sessions [41]. Rehabilitation studies have shown that motivating and empowering patients by providing them with the perception of control can expedite the achievement of the patient's rehabilitation goals [26, 47]. Providing patients with positive feedback promotes morale and empowerment [11].

Last year, roughly 795,000 Americans suffered from stroke. This statistic is forecasted to increase. In America, stroke is also the leading cause of long-term disability [21]. Roughly half of these stroke survivors still suffer from hemiparesis six months after their strokes, and roughly 30% of stroke survivors were treated with outpatient rehabilitation [21]. A recent study found that about one third of stroke patients in rehabilitation hospitals are poor participators as ranked on the Pittsburgh Rehabilitation Participation Scale [37]. Thus, improvements to outpatient stroke rehabilitation will benefit a large portion of our population. About one third of stroke victims suffer from depression after their stroke [21]. Additionally, the limitations caused by reduced wrist and hand movements are a key factor associated with reduced perception of quality of life [8]. Thus, stroke survivors will greatly benefit from morale boosting physical therapy.

In its current form, physical therapy is not enjoyable, causing patients to lack diligence in their participation at home [41]. There is a knowledge gap when it comes to making patients more accountable for their therapy. Not only do patients skip at home therapy due to lack of motivation [26], but, using traditional rehabilitation therapy, clinicians are unable to monitor patient progress automatically while the patients are in their homes. Therefore, in these traditional scenarios, clinicians typically do not know whether or not a patient has been participating in therapy until they come in for their next visit. Even when the patient comes into the office, the clinician has few strategies for assessing patient diligence. The clinician will ask the patient how often they participate in therapy, but patients may not answer honestly. The clinician can also assess the progress of the patient and estimate frequency of patient therapy sessions. However, a diligent patient's morale can be decreased if the clinician makes a low estimate of their frequency of participation due to slow progress with therapy [11].

A solution to these problems is to develop an adaptive therapy gaming system that will monitor the frequency, duration, and physical motions of the patients during their at-home therapy sessions and appropriately challenge users during all points in their therapy sessions. Rehabilitation gaming systems have been shown to increase motivation [9] while maintaining similar treatment effectiveness [4, 7].

2.2 Music and Motor Learning

The challenge point framework asserts that learning may be increased for difficult tasks by providing the learner with a model of the task (i.e. an auditory or timing model) during practice. However, such a model reduces learning for easy tasks due to learners being provided with unnecessary amounts of information [23]. This framework has been validated in a variety of studies. A study by van Vugt and Tillmann showed that a tap sequence could be learned more quickly and accurately by allowing participants to practice with tapping beat cues corresponding to the times that a tap occurred in the sequence being learned, as compared to practicing in silence or in the presence of randomly timed auditory beats [43]. A study by Aluru et al. on stroke survivors with chronic hemiparesis suggests that different types of auditory stimulations can be effective during the different stages of recovery. During early stages of recovery, when a stroke survivor is suffering from spastic paresis, a metronome beat was shown to increase wrist extension and muscle co-activation. During mid stages of recovery, when a stroke survivor is suffering from spastic co-contraction, silence was shown to increase wrist extension but reduce co-activation. During the late stages of recovery, when a stroke survivor is suffering from minimal paresis, minimal gains were made regardless of the auditory stimulus [2]. To discover which auditory cues were most effective for assisting motor skill learning, Vinken et al. used kinematic-acoustical mapping to associate seven different auditory cue schemes to six everyday upper limb actions. In this study, the different auditory schemes did not affect the learning of the task, suggesting that any type of music can be used to teach any type of motor skill [44]. In another study by Butler and James, participants were asked to create sounds with novel musical objects. Then, they were asked to identify each object by the sound it produced. For the duration of this study, fMRI data was collected. This data showed that the functional connectivity between visual- and motor-related processing regions was enhanced during the presentation of actively learned audiovisual associations [5].

2.3 Adaptive Video Games

Rehabilitation video games have been used to assist the rehabilitation process. Several studies have shown that adaptive game environments promote sustained improvements and high user morale [6, 10, 32]. Ma et. al. used adaptations to allow for their upper limb motor rehabilitation system to be used with a population of stroke survivors who had a diverse set of performance capabilities. User performance acted as an input to adapt the game to the appropriate difficulty level for each user. Users reported increased motivation while exercising with the system [32]. Cameirao et. al. created an adaptive task-oriented training system for upper extremity rehabilitation for acute stroke survivors. This system required users to operate virtual arms in a virtual reality environment where participants would move the virtual arms to complete tasks that translated to real world activities used in daily life. Once again, user performance was inputted into the adaptation function to alter the difficulty of the game to best fit the need of each user. The system was evaluated with 14 stroke survivors and the results suggested that the system promoted motor skill improvements and benefitted the users' performance with daily life activities [6]. Duff et. al. created an adaptive mixed reality rehabilitation training system which used audio feedback to cue users to perform desired movements. The system was used for stroke survivors and was tested with three chronic stroke patients. These patients improved their reaching movements, suggesting that mixed reality environments allow for easier translation of skills learned in the virtual world into the real world [10].

Several studies have been also conducted to validate a variety of methods for adapting rehabilitation games in order to best meet the needs of the users. In a study by Basteris et. al., a lead-lag based assessment was used to adapt a game played with a robotic hand rehabilitation system. In this game, the user was required to hit targets at specific times. If the user hit the target too soon, they were considered to lead the target, but, if the user hit the target too late, they were lagging. When the user lagged 40% of the time, the speed was reduced. When the user was leading 80% of the time, the speed was increased. This adaptation showed a significantly larger amount of movement repetitions performed by the subjects during each exercise session [3]. In another study by Panarese et. al., a statistical model was used during game play to increase the exercise difficulty when the user had overcome a predefined target. This algorithm was able to successfully track patient progress during the rehabilitation process of 18 chronic stroke survivors, thus allowing the automation of maintaining a consistent challenge level throughout a user's rehabilitation [36]. Another study, performed by Zimmerli et. al., used a Fitts' Law algorithm to adjust the difficulty of a game. Their game required 10 stroke survivors to move between two points on a screen in a specific amount of time. Slower times were considered easier and faster times were considered more difficult. This study showed that the Fitts' Law model can be used match a desired difficulty with the capabilities of the user, allowing for more productive practice sessions [46].

2.4 Summary

In this section, we discussed several research efforts including the problems with motor learning in stroke survivors, how music can be helpful in increasing motor learning, and how adaptive video games can be useful for appropriately challenging stroke survivors. We observed that the adaptive rehabilitation game studies used simple algorithms to adapt their game cues and focused mainly on the current performance of the user. In this research, we aim to create a more sophisticated adaptive algorithm that includes both current performance and historical data in order to better encourage users towards an expedited recovery.

CHAPTER III

METHODS FOR CUEING PARTICIPANTS TO COMPLETE MOTOR TASKS

3.1 Motivation

Existing robotic therapy systems for rehabilitation of motor function for stroke survivors are designed to engage patients through the use of interactive video games. These systems can also monitor the time spent in therapy and patient success in real time and provide feedback via game scores. This information can also be sent to clinicians in order to allow the clinicians to accurately track patient diligence and progress. Existing systems are only useful for a short period of recovery, as the task types and difficulties are static. However, an adaptive stimulus could be useful for a larger duration of the recovery process. It is hypothesized that adaptive gaming systems can increase engagement and motivate patients to participate in longer therapy sessions and that adaptive gaming systems can promote more productive exercise practice.

3.2 Design of a Passive Therapy Device with Rehabilitation Gaming Suite

We hypothesize that we can develop methodologies for rehabilitation using robotic interventions that promote engagement and encourage expedited learning for individuals learning a new motor task [14, 18]. Based on past research that showed the effectiveness of coupling assistive robotic devices with adaptive video game augmentation, we determined that, for our research, we would focus on developing a passive therapy device with a rehabilitation gaming suite that promotes engagement and productivity in at-home therapeutic wrist exercises. Our first step in proving this hypothesis is to develop a platform that can be used in a series of validation experiments. One component of our research was the development of a suite of fun and engaging rehabilitation games that could facilitate therapeutic wrist exercises. The game framework was designed such that various intervention protocols, correlated with different wrist exercises, could easily be programmed by a clinician. This framework is necessary because it allows each game to benefit users at all stages of therapy. Immediately after a stroke, users are most severely impaired; they have a limited Range of Motion (ROM), have reduced strength, move slowly, and may have shaky movements [21]. As they participate in therapy, their ROM increases, their strength improves, and they are able to move more quickly and steadily. Allowing the clinician to adjust the games allows for the clinician to pick exercises that focus on a single aspect for improvement (i.e. ROM by encouraging the user to move a greater distance or strength by encouraging the user to hold an extended position) as well as to adjust the games to practice at a desired difficulty level.

In order to enable these games to facilitate motor learning, a passive therapy device acts as a game controller [14, 18]. Our first prototype of the system utilized an existing robotic arm exoskeleton, called the HandMentor [42]. However, this existing exoskeleton did not fit the needs of our system because it came in a single size and was large and heavy due to its function for actuating the user's wrist – a capability that we do not intend to use in our designs. In order to circumvent these limitations and fit the needs of our research, we designed an easily scalable, 3D printed passive therapy device that can fit users of any size. Current robotic exoskeletons are one-sized-fits all and designed for the average adult male. 3D printing our own passive therapy device allows our technology to be utilized by end users of all shapes and sizes, including people who are too small for commercially available systems such as petite women and children as well as people who are too large for existing systems. For placement, the user simply straps our device onto their arm via three strips of

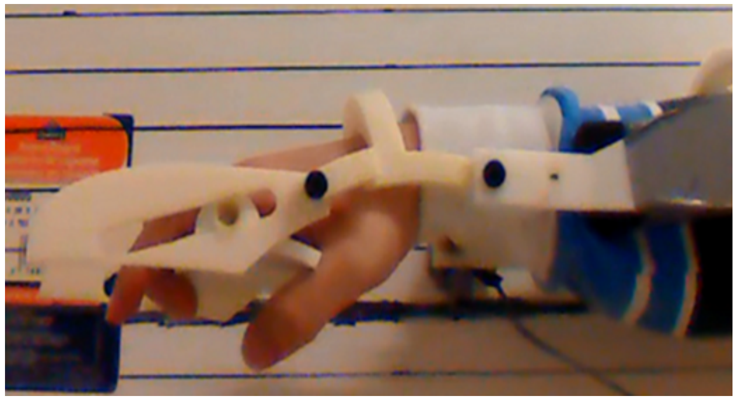


Figure 2: The scalable robotic passive therapy device.

velcro – one above the finger bed, one below the wrist, and one below the arm, as seen in Figure 2. The passive therapy device is also padded with foam for comfort.

The full range of motion of the patient's wrist is captured by a potentiometer that is located at the wrist joint of the passive therapy device and transferred via Bluetooth as a raw input to the interactive game system. The patient's wrist movements are then translated into game commands that enable user control. Currently, there is no known passive therapy device for the arm with this capability available in the public sector. Creating a scalable passive therapy device allows for this research to provide a gaming rehabilitation systems to a larger population of individuals.

The software goal of this design included creating a variety of engaging games that enhance compliance by embedding adjustable options into the game design to engage participants at all levels of recovery. As shown in Figure 3, two games, RoboBlaster and RoboRockNRoll, were created for this purpose. RoboBlaster requires the user to control a spaceship and destroy asteroids by shooting them with lasers. RoboRock-NRoll requires the user to control a pick and catch music notes that correspond to popular songs that are being played during the gaming session. Both games couple the wrist's effective range of motion and speed in a unified manner which correlates with specific therapy interventions [14, 18].

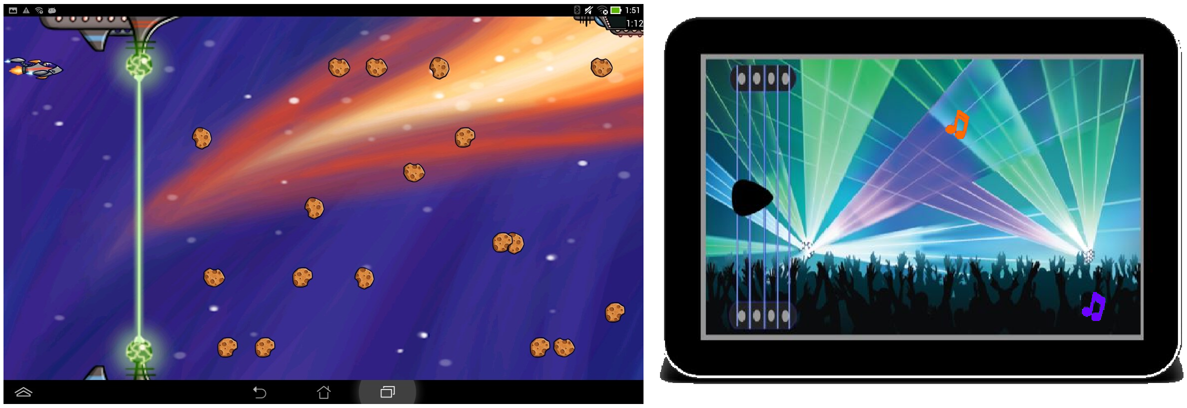


Figure 3: RoboBlaster rehabilitation game (right) and RoboRockNRoll rehabilitation game (left).

3.3 Increasing Engagement

Lack of participation in physical therapy is known to be a significant factor hindering the recovery of stroke survivors [11, 26, 37, 41, 47]. Increasing engagement and motivation for users to complete physical therapeutic exercise is of utmost importance for improving quality of life of stroke survivors [8]. Therefore, through the coupling of interactive games with a passive therapy device, we designed our rehabilitation system to be fun and engaging. We hypothesized that interacting with the system would be more enjoyable than doing exercises without the aid of the system and that people would be willing to complete more exercises with the assistance of a game than they would unaided. Thus, for our first system validation experiment, we designed an experiment to verify the system's ability to increase engagement in therapeutic motions and discourage boredom while completing these exercises [13]. We define engagement by the user's desire to participate in the task, which is measured by the amount of time that the user reports being willing to participate in the task. We define boredom by the user's desire to stop the task, which is measured by the time that the user reports wanting to end the task.

3.3.1 Experimental Setup

For this study, 14 able-bodied participants between the ages of 14 and 35 were selected. Eight of the participants were male and six were female. The participants were asked to complete two tasks, wrist exercise using the passive therapy device integrated with a rehabilitation game and wrist exercise using the passive therapy device without the game [13]. Half of the participants were randomly selected to complete the exercise task with the games first and the other half completed the exercise task without the games first. The participants were instructed to end the exercise period when they became bored with the task. The time that each participant elected to spend performing each task was recorded.

Prior to beginning either task, the participants put on the first prototype of our system built from the Hand Mentor arm brace [42] and were told that they would be exercising with the arm brace. Immediately following the completion of both tasks, the participants were asked to complete a survey that posed questions about the enjoyment and boredom during the two exercise tasks.

3.3.1.1 Exercise Task with Interactive Rehabilitation Games

For the exercise task involving the interactive rehabilitation games, participants used the Hand Mentor controller to play the RoboBlaster game. A script was read to the participants prior to the initiation of the task. This script informed the participants that they would be playing RoboBlaster using the Hand Mentor as a controller. They were informed of the game play instructions and the goal of the game. They were also told to stop and inform the researcher when they became bored with the task. The following script was used to provide these instructions:

> *"For your (first/second) task, you will be playing JetBoy using the Hand-Mentor as a controller. Your goal is to destroy the asteroids. Your ship continuously fires. To move your ship up and down, move your wrist the*

corresponding direction. You may stop when you become bored with this task."

3.3.1.2 Exercise Task without Interactive Rehabilitation Games

For the exercise task that did not involve the interactive rehabilitation games, participants completed an exercise regimen that involved moving their wrist up and down while wearing the Hand Mentor. A script was read to the participants prior to the initiation of the task. This script informed the participants that they would be exercising with the arm brace by moving their wrist up and down. They were instructed to stop and inform the researcher when they became bored with the task. The following script was used to provide these instructions:

> *"For your (first/second) task, you will be exercising with the HandMentor. To complete this exercise regiment, move your wrist up and down inside the arm brace. You may stop when you become bored with this task."*

3.3.2 Results

3.3.2.1 Duration

The average amount of time that the participants spent exercising is shown in the right image in Figure 4. As depicted, participants spent 222 seconds exercising with the rehabilitation game and 59 seconds exercising without, on average. This difference equates to the participants spending roughly four times longer exercising with the rehabilitation game than without the rehabilitation game. While the standard deviation was high for both groups, 127.2 seconds and 41.1 seconds, respectively, an unpaired t-test resulted in a P value less than 0.0001, which asserts a statistical significance in the difference between these values. Therefore, participants spent significantly more time exercising when the regimen was accompanied with a rehabilitation game than they did without.

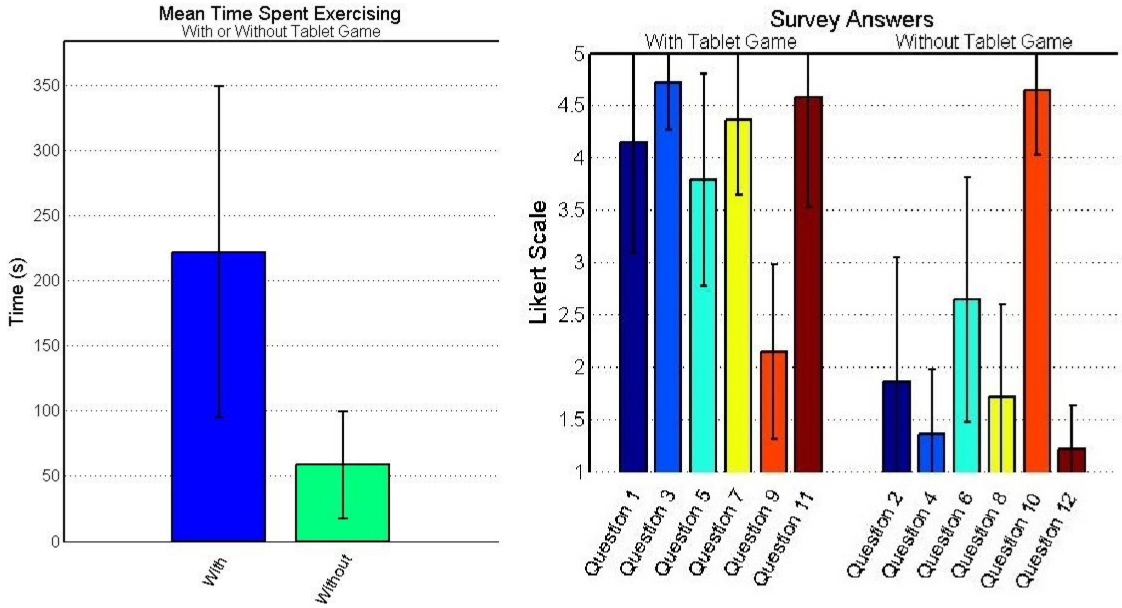


Figure 4: Engagement experiment results: mean time spent exercising with and without the rehabilitation game (right) and mean responses to survey questions (left).

3.3.2.2 Engagement

Immediately following the completion of both exercise tasks, the participants were asked to fill out a survey, where they ranked a series of statements using a five-point Likert scale with selections ranging from one to five. On the Likert scale, one corresponds to strongly disagree, two to disagree, three to neutral, four to agree, and five to strongly agree. The statements that the participants were asked to respond to are listed in Table 1.

The averages and standard deviations of the participants' responses are shown in the left image in Figure 4. From the participants' responses, the participants experienced significantly more enjoyment and engagement when exercising with the rehabilitation game. While participants gave high enjoyment and engagement scores for sessions that were accompanied by the rehabilitation game, they unanimously agreed that they did not enjoy the exercise experience without the rehabilitation game. Participants also experienced significantly less boredom while exercising with

Table 1: Survey Questions Presented to Participants

Number	Question
1	I enjoyed the exercise session WITH the tablet game.
2	I enjoyed the exercise session WITHOUT the tablet game.
3	Exercising was MORE enjoyable WITH the tablet game.
4	Exercise was MORE enjoyable WITHOUT the tablet game.
5	I felt that my exercise session WITH the tablet game was productive.
6	I felt that my exercise session WITHOUT the tablet game was productive.
7	I felt engaged while exercising WITH the tablet game.
8	I felt engaged while exercising WITHOUT the tablet game.
9	I felt bored while exercising WITH the tablet game.
10	I felt bored while exercising WITHOUT the tablet game.
11	If I had to exercise my wrist for one hour a day, every day, I would rather complete all of these exercises WITH the tablet game.
12	If I had to exercise my wrist for one hour a day, every day, I would rather complete all of these exercises WITHOUT the tablet game.

the rehabilitation game as compared to traditional rehabilitation exercises. The participants unanimously agreed that if they were required to exercise their wrists for an hour a day, as is a normal requirement for patients in stroke therapy [19], that they would prefer to do so by playing the rehabilitation game. The participants also felt that their exercise with the rehabilitation game was more productive than their exercise without, which would encourage them to participate in therapy longer and more frequently. All question pairs were significantly different with p-values < 0.05.

3.3.3 Conclusions

The results of this study show that participants prefer playing rehabilitation games with the passive therapy device more than they do exercising with traditional rehabilitation methods. On average, participants spent approximately four times longer playing the rehabilitation game before becoming bored than they did with traditional exercises. From their responses to the retrospective survey, the participants experienced significantly more enjoyment and engagement when exercising with the rehabilitation game. They also experienced significantly less boredom. The participants unanimously agreed that if they were required to exercise their wrists for an hour a day, as is a normal requirement for patients in stroke therapy [19], that they would prefer to do so by playing the rehabilitation game. Additionally, many users complained of the HandMentor exoskeleton being heavy and/or not fitting well. Based on the outcomes, we were motivated to use our customized 3D arm for future studies.

3.4 Encouraging Specific Motions

When designing a system to facilitate therapeutic motions, it is important to verify that the system actually encourages users to complete the motions precisely and accurately, in order to ensure that time spent using the system directly translates to time spent diligently completing therapeutic activities. Thus, an experiment was designed to verify that the rehabilitation game cues encouraged users to accurately complete commonly prescribed wrist therapeutic motions that are tested by the Fugl-Meyer assessment [12], a movement examination that physicians routinely use to assess the recovery of stroke patients [20]. As shown in Figure 5, three of the Fugl-Meyer wrist motions that this experiment encourages participants to complete are (1) alternating between maximum dorsiflexion and maximum volar flexion, (2) holding their wrist in a strong and stable position of their maximum dorsiflexion, and (3) holding their wrist in the a strong and stable position of their maximum volar flexion [20].

3.4.1 Experimental Setup

Fifteen able-bodied participants between the ages of 14 and 35 completed this experiment. The inclusion criteria for this study was healthy individuals. Eleven of the participants were male and four were female. Each participant completed six levels of the game, each with a different algorithm to encourage a specific motion pattern

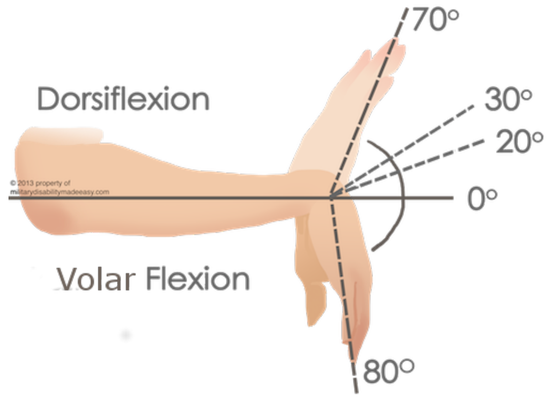


Figure 5: Wrist alternating between maximum dorsiflexion and maximum volar flexion.

[12]. The participants also completed a control session, where they were instructed to exercise without the rehabilitation game. The positions of their wrists were recorded during all of the sessions, including the control session. The participants completed the seven tasks (six levels and control) in a random order. The length of time that the participants spent in each task was one minute and twelve seconds. Actual paths and encouraged paths were compared using correlation coefficients calculated using corrcoef, a normalized covariance MATLAB function.

3.4.1.1 Walking Algorithm

The walking algorithm launches asteroids in a triangle wave represented by Equation 1, where t_{launch} is a list of natural numbers that represents the times when a target is launched. In this equation, the function used to calculate the position of the target is known as *Target(time)*. This algorithm is designed to encourage a slower frequency oscillation with a large amplitude. It encourages slow and controlled wrist motions with equal time spent in maximum dorsiflexion and maximum volar flexion positions. During the walking algorithm experimental sessions, each participant was presented with encouragement for five downward and four upward motions. The walking algorithm, shown in the right image in Figure 6, was designed to encourage this smooth, alternating motion.

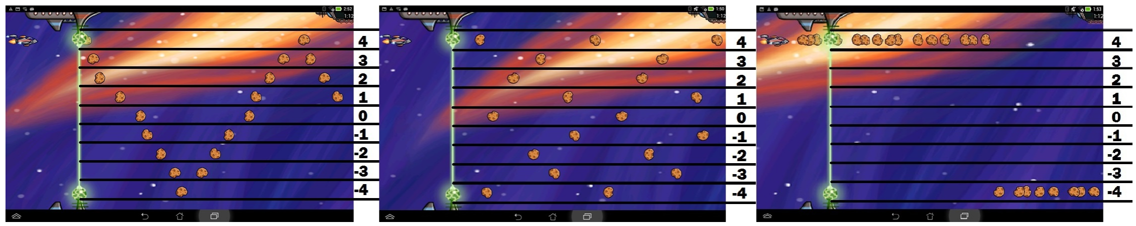


Figure 6: Three RoboBlaster levels with lanes and lane numbers overlaid: (1) Walking algorithm (right), (2) Halfing algorithm (center), and (3) Hold Stretches (left).

$$Target(t_{launch}) = 8/\pi * sin^{-1}(cos(\pi/8 * t_{launch})) \quad (1)$$

3.4.1.2 Halfing Algorithm

As seen in the center image in Figure 6, the halfing algorithm launches asteroids in the pattern described by Equation 2, where t_{launch} is a list of natural numbers that represents the times when a target is launched. This algorithm is designed to encourage oscillation that vary in difficulty. First, the patient must reach to the maximum dorsiflexions and volar flexions. Then, the participant creates an oscillation that has an amplitude of half of their maximum range. Next, the required oscillation is 75% of their full range. Finally, the participant must complete an oscillation that is 25% of the full range. The participant repeats this oscillation pattern for the entirety of the gaming session. During the halfing algorithm experimental sessions, each participant was presented with encouragement for nine oscillations from the minimum to the maximum of their range, eight oscillations of 75% of their range, eight oscillations of 50% of their range, and eight oscillations of 25% of their range. This algorithm was designed to encourage the portion of the Fugl-Meyer test that requires alternating motions between maximum dorsiflexion and maximum volar flexion [20]. However, the variation of the levels of difficulty is designed to encourage growth and allow for partial success for participants who have not yet achieved high flexion in either direction.

$$Target(t_{launch}) = \begin{cases} 4, & \text{if } t_{launch} \bmod 9 = 0 \\ -4, & \text{if } t_{launch} \bmod 9 = 1 \\ 0, & \text{if } t_{launch} \bmod 9 = 2 \\ 2, & \text{if } t_{launch} \bmod 9 = 3 \\ -2, & \text{if } t_{launch} \bmod 9 = 4 \\ 3, & \text{if } t_{launch} \bmod 9 = 5 \\ -3, & \text{if } t_{launch} \bmod 9 = 6 \\ 1, & \text{if } t_{launch} \bmod 9 = 7 \\ -1, & \text{if } t_{launch} \bmod 9 = 8 \end{cases} \tag{2}$$

3.4.1.3 Hold Stretches Algorithm

The hold stretches algorithm launches asteroids in a square wave pattern that is described in Equation 3, where t_{launch} is a list of natural numbers that represents the times when a target is launched. This algorithm is designed to encourage participants to reach their maximum dorsiflexion position and then hold the stretch. Then, the participant must reach their maximum volar flexion and hold the stretch. This process repeats. During the hold stretches algorithm experimental sessions, each participant was presented with encouragement for holding three stretches in their maximum dorsiflexion and three stretches in their maximum volar flexion. For this test, the wrist is held at approximately 15 ° dorsiflexion while a slight amount of resistance is added [20]. The hold stretches algorithm, shown in the left image in Figure 6, was designed to allow for practice of the wrist stability portion of the Fugl-Meyer test. This algorithm encourages a much higher maximum, of greater than 52.5 °. Since this system currently does not allow for resistance to be applied, a larger angle is used, to utilize naturally occurring resistance from tendons stretched to their maximum potential.

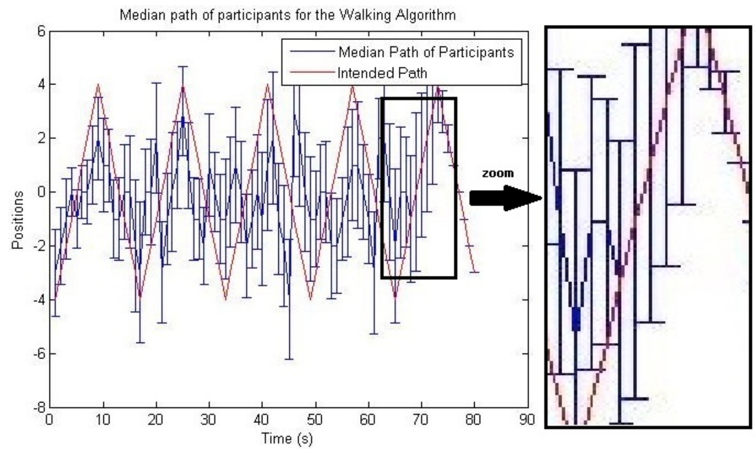


Figure 7: Difference from intended path for the walking algorithm.

The increase of the encouraged angle could be changed to fit the maximum range of each participant.

$$Target(t_{launch}) = 4 * (-1)^{floor(t_{launch}/15)} \tag{3}$$

3.4.2 Path Approximations

The differences between participants' actual paths and the intended paths were graphed for the walking and hold stretches algorithms, shown in Figures 7 and 8, respectively. A trend that is learned from these comparisons is that the participants aligned most accurately with the encouraged path during the beginning of the game. The correlation coefficients between the intended paths and the median actual paths for the walking and hold stretches algorithms were found to be 0.44 and 0.75, respectively. At the beginning of each session, the actual path aligns more closely with the encouraged path and the standard deviations between the participants' movements tend to be smaller. As time progresses, the average real path of the participants becomes more erratic and the standard deviation becomes larger.

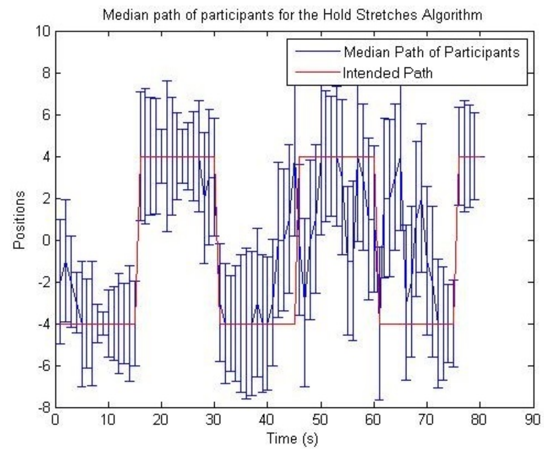


Figure 8: Difference from intended path for the hold stretches algorithm.

3.4.3 Aliasing

Algorithms that involve frequent, rapid changes between asteroid locations did not encourage frequent, rapid movements from the participants. Instead, an aliasing of the intended path occurred, as it does in an under-sampled signal. The path that the participants followed better fits a path of the algorithm with a lower frequency. Each participant followed a signal with a different frequency. For example, one of the participants followed a signal that was best fit by a 300% increase of the period of the intended path. The right image in Figure 9 shows a participant's path plotted with the non-aliased version of the intended signal, while the left image in Figure 9 shows a participant's path plotted with an aliased signal with a period of 300% of the intended period. The correlation coefficients for the non-aliased and aliased paths were found to be 0.14 and 0.41, respectively. From these figures, it can be seen that, the aliased version is a better fit than the non-aliased version.

3.4.4 Discussion

The results of this healthy pilot study suggest that not only can the therapy system change how a participant moves, but can also encourage specific motions designed to

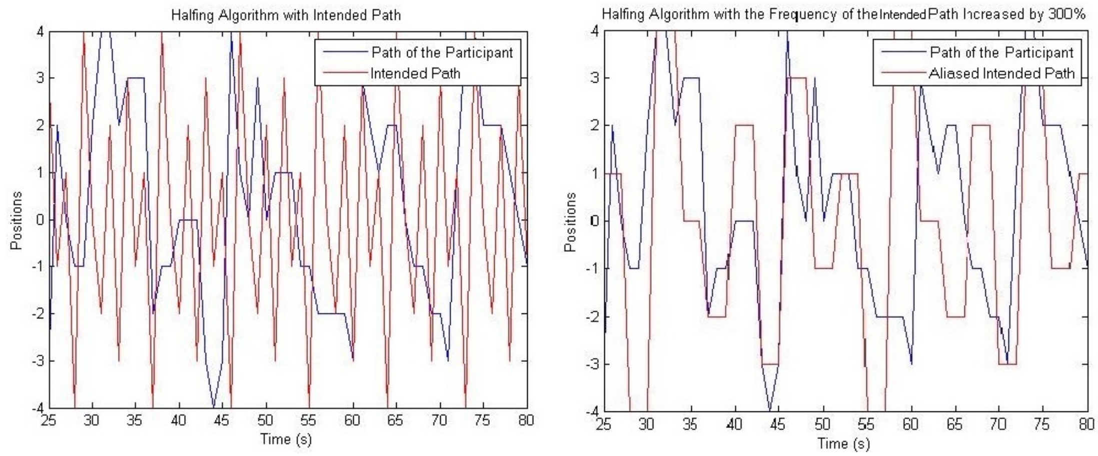


Figure 9: The halfing algorithm plotted with the intended path (right) and the halfing algorithm plotted with a 300% frequency increase of the intended path (left).

mimic therapeutic interventions. This study also exposed two important considerations, frequency of targets and game speed, that need to be taken into account in order to encourage participants to follow the encouraged path precisely.

3.4.4.1 Path Approximations

During data analysis, it was discovered that participants followed the encouraged paths closely at the beginning of each of their sessions. However, as time progressed, their paths became less predictable and the standard deviations between these paths became larger. This trend occurs because of an error in game design, as the lasers that are fired by the ship move slowly and are fired at a faster rate than asteroids are launched. As a result, residual lasers from a previous target will remain on the screen after the target asteroid has been destroyed. As the game progresses, these residual lasers begin to destroy asteroids as they appear, removing the encouragement for the participants to move to the wrist position that corresponds to the asteroid's lane placement. The effects of this become prevalent at roughly 40 seconds into the session. The correlation coefficients between the intended paths and the median actual paths for the first 40 seconds of the walking and hold stretches algorithms were found to be 0.50 and 0.98, respectively.

3.4.4.2 Aliasing

It was also discovered that an aliasing effect occurs between the actual path participants followed and the encouraged path when the encouraged path presented frequent, rapid changes between targets. This suggests that participants prefer slow, smooth paths to quick, harsh motions. In levels with rapid fluctuations of targets, each of the participants exhibited this trend with a different frequency of the intended path. This suggests that each individual has a maximum preferred speed at which they feel most comfortable moving. This maximum preferred speed appears to be a different speed for each individual. When the encouraged path's speed exceeded the participants' maximum preferred speed, the participants would alias the encouraged path to their maximum preferred speed.

3.4.5 Conclusions

The motivation of this project is to use the system to facilitate therapy sessions for stroke patients with a passive therapy device and gaming environment. Showing that our system is capable of encouraging motions that mimic therapeutic interventions was a preliminary task for producing a novel robotic wrist rehabilitation system that integrates the strengths of three of the most favorable rehabilitation strategies for post-stroke rehabilitation of hand function.

3.5 Musical Cues

Based on our current results with our rehabilitation system, we desire to find ways to improve the learning process such that learning is optimized for each individual. As such, we looked at the impact that musical cues have on motor learning. Studies suggest that musical cues may help users learn more quickly and/or retain motor skills [43, 2, 44, 5]. In this pilot study, we wished to examine whether prior knowledge of musical cues can be used to expedite the learning of a motor skill.

3.5.1 Experimental Setup

We conducted a pilot study to determine whether prior knowledge of musical cues could be used to expedite the learning process [17]. In this experiment, 21 able-bodied participants were asked to complete a 90 second exercise session. To complete this experiment, a level of the RoboRockNRoll video game was created with music notes that correspond to the popular nursery rhyme and song, "London Bridges." During the 90 second exercise period, participants completed three iterations of the "London Bridges" pattern in the video game, with each iteration presented as a 30 second section. Each participant completed three sections during this experiment. During sections 1 and 3 of the exercise period, all participants completed their exercises without any sound cues. During section 2, 11 participants were presented with music cueing – notes of varying frequencies that were presented at times that corresponded with visual cue appearances; three were presented with a tapping beat cue – a clapping sound of a constant frequency that was presented at times that correspond with visual cue appearances; and seven were exclusively presented with the visual cues and no sounds. As shown in Table 2, the participants were divided into five subgroups for analysis. (Note: the group that was presented with musical cues was further separated into three subgroups.) The participants' accuracy in hitting the notes and jerkiness of motions were compared between these groups. Upon completion of the 90 second exercise period, all participants were asked if they could identify the song, in order for the researchers to discover if the participants were applying prior knowledge about the song to this exercise period.

3.5.2 Results and Discussion

3.5.2.1 Song Recognition

Of the 11 participants who were presented with a music intervention, four correctly identified the song, four identified the song incorrectly, and three did not know what

Table 2: Description of Participant Groups

Group Identifier	Description
1	Participants were presented with musical intervention and correctly identified the song.
2	Participants were presented with musical intervention and incorrectly identified the song.
3	Participants were presented with musical intervention and did provide a guess as to which song was played.
4	Participants were presented with a tapping beat intervention.
5	Control - Participants were presented with a silent intervention.

the song could be. These participants were divided into the three subgroups, Group 1-3, as described above for analysis and shown in Table 2. None of the 10 participants in the tapping beat or control(no sound) groups could identify a specific song, which was a valid assessment since no song was provided for these two groups.

3.5.2.2 Lead and Lag

A lead/lag score was calculated each time a music note target was collected by the user during game play. Lead is defined as the amount of time that a user intends to hit a note prior to the time that the theoretical path anticipated that the user would hit the note. Lag is defined as the amount of time after the theoretical hit time that the user hit the note. The lead/lag score provides insight as to whether the user is anticipating note appearances or simply reacting to game cues. A lead/lag score that tends more towards leading suggests that the user has learned the path and knows where a future target will appear, before the game prompts the user with this information. A lead/lag score that tends more towards lagging suggests that the user has not yet learned the movement pattern and is relying heavily on game cues to successfully complete the exercise motions.

This lead/lag score was used to quantify the amount of lead/lag at each point of interest, which we define as the point on the actual path that participants traveled in which they intended to hit the target. First, we developed an algorithm to automatically extract these points of interest as shown in Equation 4. The position extrema in between the time that the previous note appeared and the time that the next note appears is used to determine these points, as seen in the blue circle in Figure 10. If there was no extrema in this time period, the closest point on the actual path that had the same y-value was selected, as seen in the red circle in Figure 10. Once these points of interest were determined, they were compared to the points where the notes appeared on the screen, which are shown as black and red dots, respectively, in Figure 10. As shown in Equation 5, the lead/lag score was calculated by calculating the difference in x-values. If the x-value of the point of interest continued past the intended point, this resulted in a negative score and was associated to lagging. However, if the x-value of point of interest occurred before the x-value of the intended point, this was a positive score and was associated to leading.

$$\begin{aligned}
&\textit{if (extrema exists)} \\
&\qquad POI = extrema(P_a(T_p), P_a(T_n)) \\
&\textit{else} \\
&\qquad POI = P_a(T(y == P_t(T_c)))
\end{aligned} \tag{4}$$

where POI = point of interest, P_a *= actual path,* P_t *= theoretical path,* T_c *= the time that the current target appears,* T_p *= the time that the previous target appears, and* T_n *= the time that the next target appears*

$$LeadLag = T_c.x - POI.x \tag{5}$$

where $T_c.x$ *= the x value of the current target and POI.x = the x value of the point of interest*

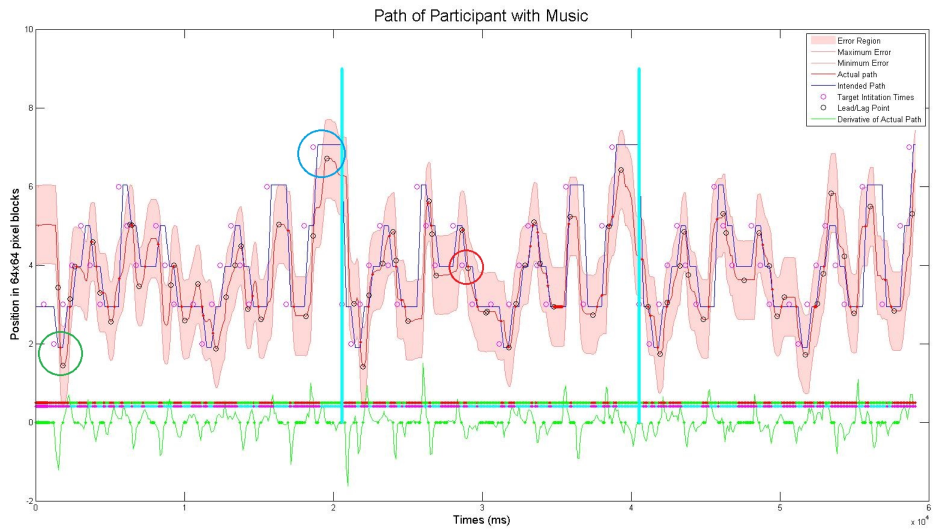


Figure 10: Graph of actual and theoretical paths with target points and points of interest.

Table 3 shows the results for the mean lead/lag scores for participants in all five groups across all three sections of game play. As shown, the lead/lag scores prior to and during the interventions were not significantly different from one another for any group. When you look at the trends across the sections of game play, the groups that received musical intervention (groups 1-3), show a trend of decreasing lag. The largest decrease across the 3 sections was in group 1. After the intervention, the lead/lag scores for the group of participants who received musical intervention and correctly identified the song, was significantly less negative than the music not identified, beat, and no sound groups. This trend suggests that once participants were able to correctly identify the song, they were able to rely on their prior knowledge of where they will be expected to move next in order to anticipate the movement quicker.

3.5.2.3 Overshoot and Undershoot

An overshoot/undershoot score was also calculated each time a music note target was collected by the user during game play. An overshoot is defined as the amount of distance that a user continues past the target, while an undershoot is defined as the

Table 3: Table of Mean Lead/Lag Scores for the Groups

Mean LeadLag	Section 1	Section 2	Section 3	Average for all Sections
Group 1	-677.9	-590.3	-555.4	-608.6
Group 2	-682.5	-620.4	-602.6	-635.6
Group 3	-673.9	-671.1	-654.4	-666.6
Group 4	-711.1	-672.3	-704.2	-695.8
Group 5	-664.3	-580.9	-637.4	-627.4

amount of distance that a user stops prior to reaching the target. In Figure 10, the points inside of the blue circle are an example of an undershot point and the points inside of the green circle are an example of an overshot point. This score was calculated by comparing the points where the notes appeared on the screen to the points of interest calculated in Equation 4. As shown in Equation 6, the overshoot/undershoot score was calculated by the difference in y-values. A positive value is associated to overshooting, while a negative value is associated to undershooting.

$$
\begin{aligned}
&OverUnder = P_a(T_c).y - POI.y \\
&\textit{if}(P_a(T_p) > P_a(T_c)) \\
&\qquad overscore = overscore - OverUnder \\
&else \\
&\qquad underscore = underscore + OverUnder
\end{aligned} \tag{6}
$$

where $T_c.y$ = *the y value of the current target,* $POI.y$ = *the y value of the point of interest,* P_a = *actual path,* T_c = *the time that the current target appears, and* T_p = *the time that the previous target appears*

As shown in Table 4, the over/undershoot scores for the musical groups (1-3), regardless of participant recognition of the song, were significantly smaller than those of the silent group during the intervention, with p-values less than 0.01 for all musical

Table 4: Table of Mean Overshoot/Undershoot Scores for the Groups

Mean OverUnder	Section 1	Section 2	Section 3	Average for all Sections
Group 1	-0.0326	0.1789	0.2548	0.1320
Group 2	0.1457	0.2435	0.2716	0.2195
Group 3	0.2634	0.2652	0.3159	0.2810
Group 4	-0.00955	0.05208	0.07088	0.03734
Group 5	0.3887	0.7961	0.5459	0.5773

groups. However, no statistically significant trend exists before or after the intervention. This trend suggests that musical cues may be used to focus a user's motions and encourage them to hit a target more precisely. However, these benefits do not appear to carry over into practice sessions after the musical cues are removed. This is contrary to the results derived from many studies that use auditory cues, so this aspect of the research needs further investigation in future work [2, 5, 43, 44].

3.5.2.4 Jerkiness

The derivative of each participant's actual path was calculated, as shown by the green line in Figure 10. As shown in Equation 7, the maximum absolute value of the derivative between each point of note initiation and each point of interest was calculated. This value is considered the jerkiness factor for each point. Jerkiness is defined as the peak speed that the participant moves towards each target.

$$jerkiness = max(abs(P_a'(T_{app}), P_a'(T_{poi}))) \tag{7}$$

where P_a' = *the derivative of the actual path,* T_{app} = *the time that the current target appears,* T_{poi} = *the time associated to the point of interest*

As shown in Table 5, the average jerkiness varied drastically from participant to participant. Therefore, no trend immerged prior to, during, or after the intervention for any of the different groups. Since the participants were so different from one

Table 5: Table of Mean Jerkiness Scores for the Groups

Mean Jerkiness	Section 1	Section 2	Section 3	Average for all Sections
Group 1	0.52474	0.561361	0.51002	0.532351
Group 2	0.676433	0.887858	0.713825	0.760014
Group 3	0.640842	0.755425	0.564765	0.65493
Group 4	0.317708	0.389757	0.345788	0.351159
Group 5	0.975446	1.103237	1.082492	1.05332

another, this study will need to be repeated with more participants in order to discover whether any trends exist for jerkiness.

3.5.3 Conclusions

This study shows that musical cues may be able to encourage more precise aiming at targets when the music is present. Precision when it comes to timing appears to be retained after the removal of musical cues. However, improvements in precision for the y-positions of the targets do not appear to persist beyond the removal of musical cues based on our current implementation. This study also shows that existing knowledge of a song can be used to encourage users to anticipate motions and therefore learn a motor task more quickly.

CHAPTER IV

SIMULATING MOTOR TASK LEARNING IN A VIRTUAL ENVIRONMENT

Now that we have created a rehabilitation gaming system with a passive arm rehabilitation device as a controller, we wish to create a virtual simulation for the same game. In this virtual environment, we wish to teach similar tasks and to validate this environment as being an effective simulation of the physical environment. Creating and validating such a virtual environment would allow the researchers to have a simplified testing process and thus, allow us to perfect our system in a virtual environment and then port the final results back to the physical environment for a final validation.

Testing in a completely virtual environment can be a simplified process for researchers. Virtual experiments allow for researchers to be required to set up less equipment as well as to be freed from the worry of damage to physical systems. In addition, virtual experiments offer the researchers an opportunity to conduct online experiments, which can be much more time efficient than in person experiments. The reduction of cost and ease of usability are two aspects of virtual simulations that make such techniques popular among researchers [31]. However, in order to validate that testing in a virtual environment is a valid way of simulating a physical interaction, we must compare the results between a test in the physical environment and one in a virtual environment. If the results are consistent between platforms, we can assume that the virtual environment is a similar enough approximation of our physical environment to assume that conclusions from results of a virtual environment are translatable to the physical environment.

In this chapter, we begin to analyze the learning results of our users as they learn the task via calculating and plotting their learning curves. We begin to explore various methods for presenting task information and their effects of the learning of the motor task.

4.1 Learning Curves

When teaching motor tasks to individuals, the individual's performance can be plotting along a graph. For normal tasks, these graphs follow a curve trend. When the individual is a novice, their performance is very poor. As they learn, the performance improves. At first, performance gains occur very quickly. However, as the performance improves, the gains slow until a platue is reached. This traditional curve is referred to as a learning curve. While the shape of the curve is consistent, the parameters (such as the starting performance, slope of the curve, and ending platue) differ greatly between users and tasks [22].

When teaching motor tasks, practice is considered the single most important factor for permanent ability improvement for a motor skill. The challenge-point framework exstensively outlines how practice difficulty effects learning outcomes for various motor tasks. It states: (1) learning cannot occur in the absence of task information and performance feedback, (2) learning will occur more slowly if the individual is presented with too much or too little task information and performance feedback, and (3) for learning to occur optimally, the individual must be presented with the optimal amount of task information and performance feedback – however, this optimal amount differs as a function of the skill level of the individual [22].

4.2 Virtual Environment

A virtual environment was created to simulate playing the RoboRockNRoll game, shown in Figure 3, with our physical system. This virtual game was used in all of the experiments described in this chapter. To play the developed virtual game, participants were asked to watch a tapping sequence and then recreate it on their

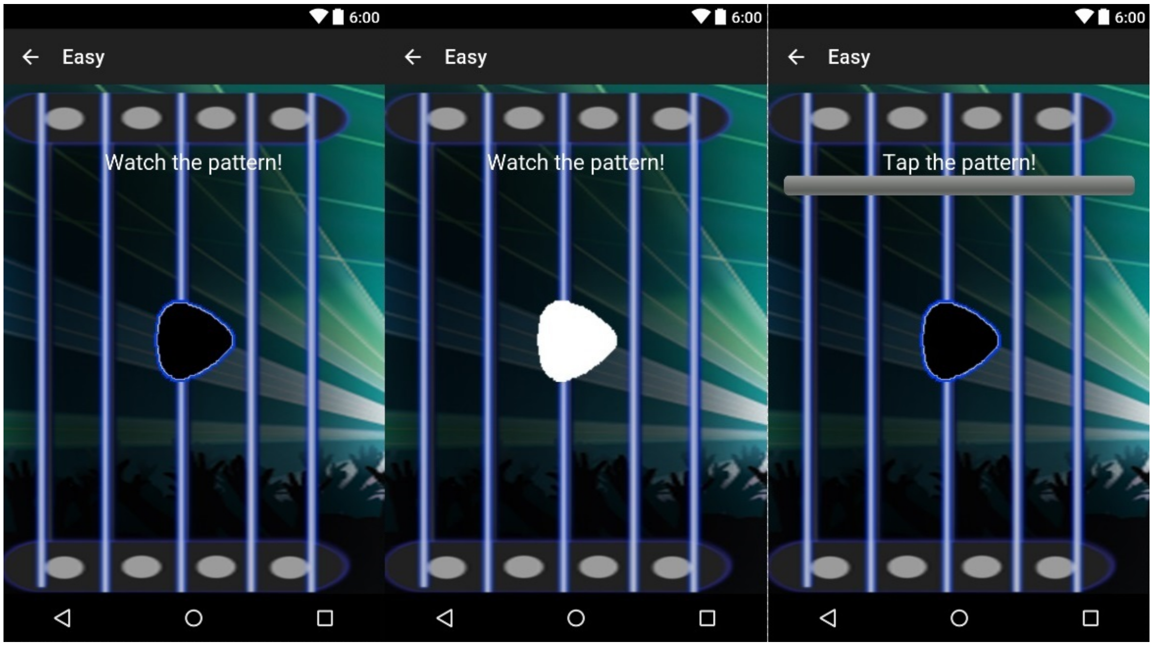


Figure 11: Screenshots from the tapping game. The left screenshot shows the game during the watching portion of the interaction when the pick is black. The center screenshots shows the pick flashing white during the watching portion of the interaction. The right screenshot shows the game during the recreation part of the interaction.

own. During the watching portion of this interaction, a black guitar pick overlaid on guitar strings would flash white when a tap event was supposed to occur. During the recreation portion of this interaction, a progress bar would appear on the top of the screen to show the participants the amount of time they had to tap the sequence. However, the guitar pick would be black and would not flash. Figure 11 shows screenshots of the game during the watching and recreation portions of the interaction. They repeated this watch and recreate process four more times, for a total of five sessions of watching the tapping sequence and five sessions of recreating the tapping sequence. In this game, the tapping sequence that was presented to the users began with for slow taps followed by eight faster taps. The faster taps had a frequency that was double the slower taps. Figure 12 depicts sheet music corresponding to the tapping sequence taught in this game.

Upon completing the five sessions of tapping, participants were asked to complete

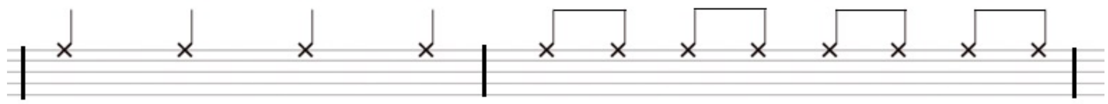


Figure 12: The sheet music of the tapping sequence taught in our app game.

a short survey. As part of this survey, they were asked their age, gender, whether or not they heard sounds during the gaming experience, and to rate the easiness and frustration of the game on a scale of 1 to 5, where 1 corresponded to easiest/least frustrating and 5 corresponded to hardest/most frustrating. Figure 13 shows a screenshot of this post-game survey.

4.3 Validating that Learning Trends Observed in the Physical Environment Translate to the Virtual Environment

In the "Musical Cues" section of the "Preliminary Results" chapter of this thesis, we observed that musical cues could be could be used to teach a motor task better than practicing in silence. In this experiment, we aim to validate that this learning trend will be observed in a virtual environment as well [16].

4.3.1 Experimental Setup

4.3.1.1 Virtual Therapy Game

We conducted a virtual study on Amazon's Mechanical Turk. 106 participants completed this experiment. 38 participants were female, 55 were male, and 14 preferred not to answer this question. The age of participants ranged from 19 to 54. The average age of participants was 30.53 and the standard deviation was 7.94. The data from 25 participants was thrown out due to the participants having technical difficulties with the game or for not following instructions. At the end of the game, the participants were shown a screen stating that their data was being uploaded. When the upload was complete, a screen was shown saying that they could close out of the game. Participants that closed the game prior to this screen had results that

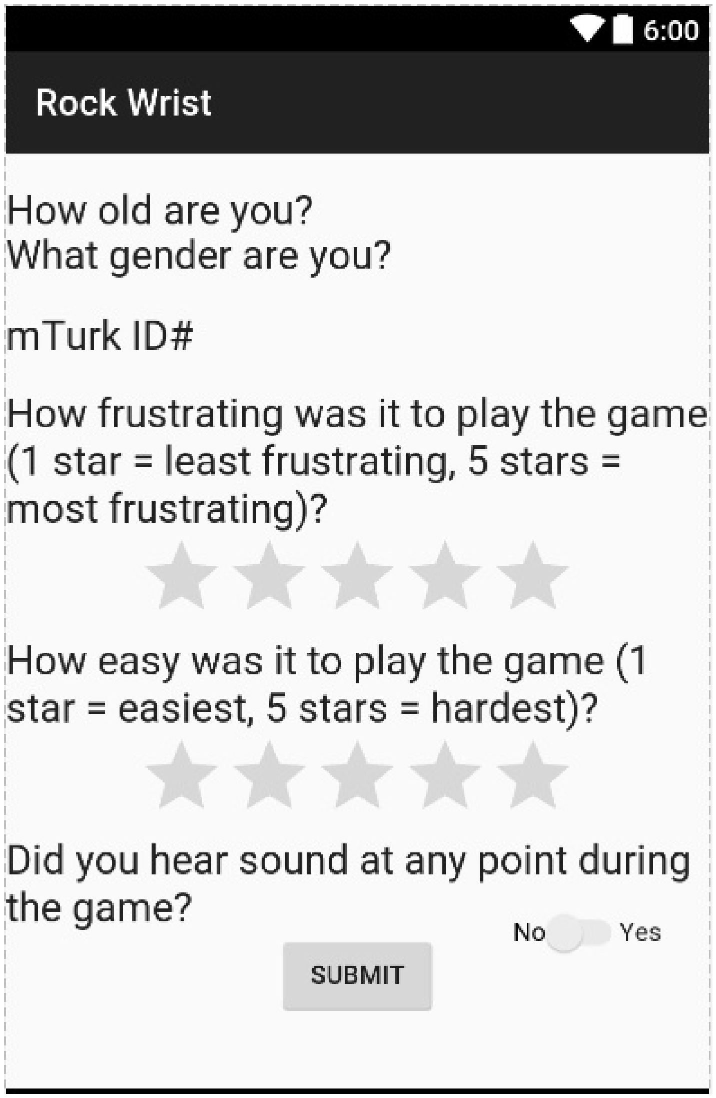


Figure 13: A screenshot of the post-game survey.

were partially submitted or not submitted to us. These participants were considered to have technical difficulties. Participants that did not attempt to create a tapping sequence for one or more recreation sessions were considered to have not followed directions. The data from the remaining 81 participants was analyzed.

Participants were randomly assigned to one of two test groups. In the experimental group, tapping sounds were played during the watching phase of the interaction when the guitar pick flashed white. In the control group, no sounds were played at any point in time during the interaction [16].

4.3.1.2 Experiment Using a Robotic Wrist Rehabilitation System

In order to validate that our results would translate to trends that could also be observed during robotic rehabilitation sessions, we recreated this experiment using a robotic wrist rehabilitation system [16]. The therapy game used for this experiment was identical to the game used in the virtual environment. However, the movement methods used for interaction with the game were different. Instead of tapping the screen, as the users did in the virtual environment, users were asked to interact with the game by explicitly moving their wrist to control a robotic arm exoskeleton, as shown in Figure 2. The exoskeleton functions by detecting the full range of motion of the user's wrist via a potentiometer, located at the wrist joint of the robotic exoskeleton therapy device that measures the wrist angle. This information is then transmitted from the exoskeleton as a raw input to the therapy game via Bluetooth. The user's wrist movements are then translated into game commands that enable user control.

Participants were asked to make downward drumming motions while wearing the exoskeleton, as if they were tapping a beat on a table. The exoskeleton detected the lowest peak in their downward wrist flexion and identified this as the moment in which a tap occurred. Eleven participants completed the experiment in the physical environment (5 female, 6 male). The age of participants for this experiment ranged from 17 to 36, with a mean of 25 and a standard deviation of 5.23.

4.3.2 Performance Measures

During the reconstruction phase of the interaction, the participants' screen taps were monitored. For data analysis, the tapping sequence was separated into two sections: the slower section and the faster section. We define these sections as follows:

4.3.2.1 Slower Section

The slower section is defined as the portion of the experiment in which four slow taps are presented to participants. This is shown as the first four notes in Figure 12.

4.3.2.2 Faster Section

The faster section is defined as the portion of the experiment in which eight taps, played at twice the frequency of the slow taps, is presented to participants. This is shown as the last eight notes in Figure 12.

To evaluate performance, screen taps were analyzed and separated into four performance measures: (1) slower portion note count, (2) faster portion note count, (3) slower portion average period, and (4) faster portion average period.

4.3.2.3 Note Count

For this performance measure, the number of taps during the slower and faster sections were recorded. To achieve perfect performance, a user must tap four times during the slower section and eight times during the faster section. For this performance measure, the note counts at time t=0 were initialized to 0. As seen in Equation 13, the note count (which is represented as N_{note}) is the summation of the occurrences that the user inputs a tap (which is represented as $note(t)$).

$$N_{note} = \sum note(t) \tag{8}$$

4.3.2.4 Average Period

The average period, or the inverse of the frequency, that each participant tapped was calculated for the slower and faster sections. A perfect performer would have a period of 1250ms in the slower section (to match the period that the visual cues were presented to the users during the slower section) and a period of 625ms in the faster section (to match the period that the visual cues were presented to the users during the faster section) . For this performance measure, the average period at time t=0 was initialized to 0 for both the slower and faster sections. As seen in Equation 14, the average period (which is represented as T_{note}) was calculated by dividing the difference between the time that the last note in the section was tapped (which is represented as $t_{lastnote}$) and the time that the first note in the section was tapped (which is represented as $t_{firstnote}$) by the note count for the section (which is represented by N_{note}).

$$T_{note} = \frac{t_{lastnote} - t_{firstnote}}{N_{note}} \tag{9}$$

4.3.2.5 Learning Curves

For each participant and each performance metric, we calculate a learning curve that best fits the data. There exist a variety of shapes of learning curves. However, in this context of learning, the performance measures can be described as showing large amounts of improvements early in learning. Then, the performance measures begins to level off and approach a horizontal asymptote. This type of learning curve is best fit by the power law learning curve [30, 35]. Thus, we calculate a power law learning curve that fits to each of the performance metrics for each of the participants. The exact formula used to calculate the best−first learning curve can be seen in Equation 15, where P is the estimated performance, a is the estimated performance of the 0th trial (i.e. time = 0), N_{trial} is the number of trials, and b is the log−log slope of the curve.

$$P = a \times (N_{trial})^{-b} \tag{10}$$

4.3.3 Results and Discussions

4.3.3.1 Learning Curve Calculations

For each participant, the error for each performance metric was graphed for each trial. Error was computed by calculating the difference between actual performance and perfect performance, as defined in the Performance Measures section. For each performance metric, an exponential curve was fit to the data points. The best−fit exponential curve represents the learning curve for that participant associated with each performance metric. To find the best fit exponential curve, we used the power law learning curve formula, as seen in Equation 15. The correlation coefficient between the learning curve and the set of data points for each metric was also calculated. If the correlation coefficient was similar enough to the data that it had a p−value $<$ 0.1, the curve was considered to be a fair representation of the data. Otherwise, we determined that, in that instance, learning did not occur.

4.3.3.2 Calculating Expertise

An expertise value was calculated for each performance metric for each participant, using Equation 16 and the calculated learning curves. For the initial trial at time=0, all participants were considered to have equal knowledge, corresponding to an initial baseline performance level of 0. When the error between actual and perfect performance converged to zero, the participants were considered to have mastered 100% of the performance metric. We considered the participants to be experts at the performance metric when the learning curve estimated that they would perform at 90% of mastery of the performance metric. 90% was chosen since this associates to the highest letter grade that is possible to achieve on the education grading scale [38]. As

Table 6: Statistical Analysis for Each Performance Metric for Participant who used the Virtual Environment

Performance Metric	Sound Mean	Sound STD	Silent Mean	Silent STD	p-value
Slow Note Count	0.6156	0.2932	0.4655	0.2627	0.0126
Slow Average Period	0.2910	0.1864	0.2283	0.1687	0.0243
Fast Note Count	0.4009	0.2578	0.2936	0.1438	0.0187
Fast Average Period	0.2953	0.1890	0.2012	0.1457	0.0055

seen in Equation 16, the number of trials to achieve this expert level was calculated by taking the inverse of $1 + N_t$, where N_t is the number of trials until an expert level is achieved. Using this equation, the expertise of each participant was calculated by taking the inverse of the number of trials needed to perform at an expert level.

$$Expertise = \frac{1}{1 + N_t} \tag{11}$$

4.3.3.3 Virtual Environment

As shown in Table 6, the means and standard deviations for the expertise of each participant were calculated for each performance metric. As seen in the sound mean and silent mean columns, the average expertise was higher for the sound group as compared to the silent group for all performance metric. Based on this quantity, the sound group learned all performance metrics more quickly than the silent group. The p−values, shown in the right most column, show a statistically significant difference with p−values < 0.05 for all performance metrics.

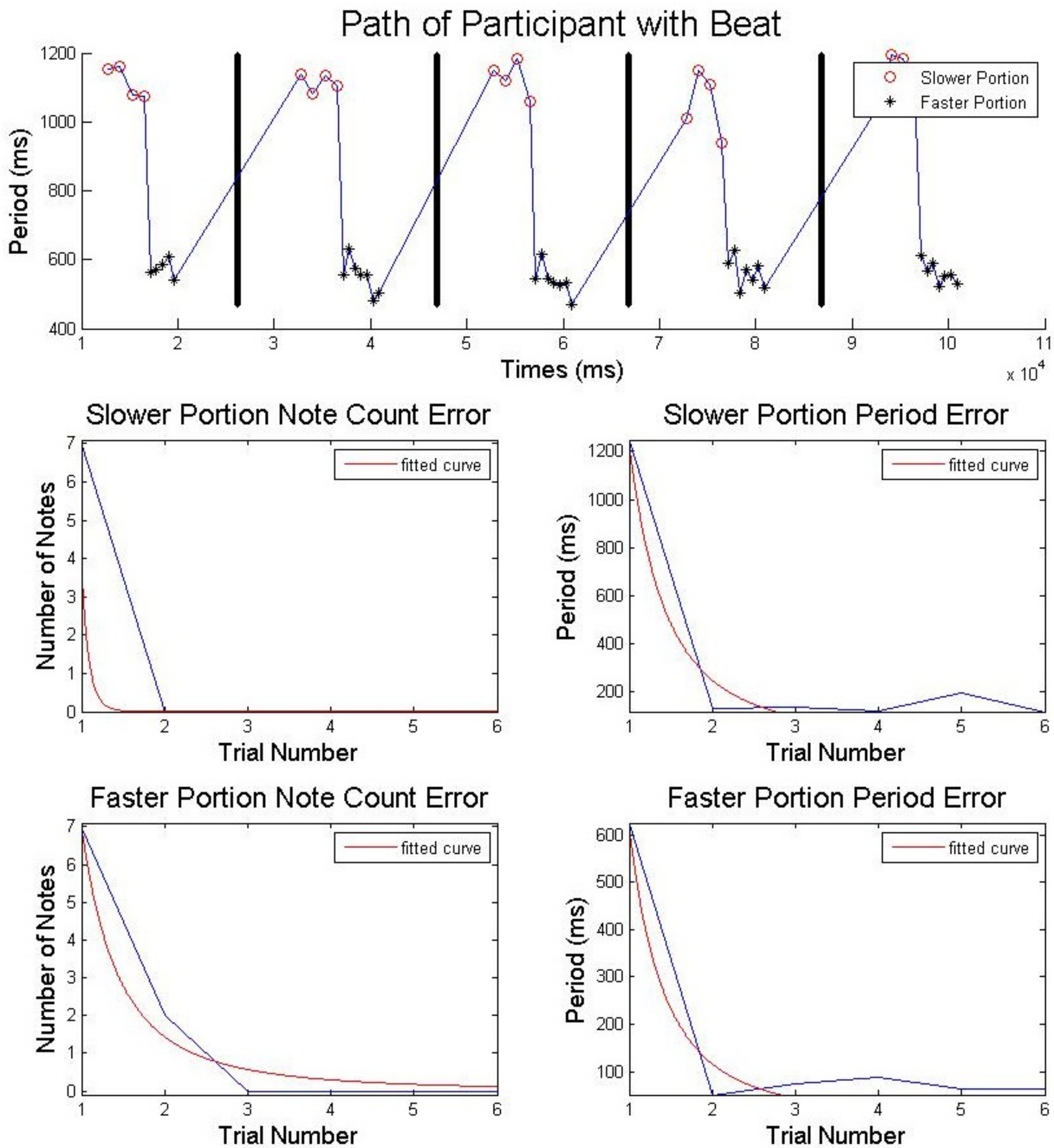


Figure 14: A participant's data and learning curves.

Table 7: Statistical Analysis for Each Performance Metric for Participants who used the Physical Environment

Performance Metric	Sound Mean	Sound STD	Silent Mean	Silent STD
Slow Note Count	0.4227	0.3080	0.3611	0.2989
Slow Average Period	0.2450	0.0661	0.1799	0.1360
Fast Note Count	0.2023	0.1717	0.1308	0.0928
Fast Average Period	0.4142	0.1548	0.2705	0.1489

4.3.3.4 Physical Environment

As shown in Table 7, the trends of having an average expertise for the sound group as compared to the silent group continued in the physical environment for all performance metric. However, these trends were not shown to be statistically significant due to a small sample size of participants.

4.3.3.5 Calculating the Correlation between the Physical and Virtual Environments for Musical Learning Trends

To show that the learning experience created in the physical environment is correlated with the learning experience that we created in the virtual environment, the concordance correlation coefficient for the expertise datasets for both environments was calculated using Lin's concordance. By definition, Lin's concordance assesses the degree of equivalence between data collection methods, allowing the researcher to compare a new method to a validated method. The equation used to calculate the concordance coefficient can be seen in Equation 12, where ρ_c represents the concordance correlation coefficient, ρ represents the correlation coefficient between the datasets, μ_{exo} and σ_{exo} respectively represent the mean and standard deviation for the dataset of participants who used the rehabilitation robotic exoskeleton to learn the pattern, and μ_{touch} and σ_{touch} respectively represent the mean and standard deviation for the dataset of participants who used the touch screen game to learn the pattern

[28, 29, 34].

$$\rho_c = \frac{2\rho\sigma_{exo}\sigma_{touch}}{\sigma_{exo}^2 + \sigma_{touch}^2 + (\mu_{exo} - \mu_{touch})^2} \tag{12}$$

In addition to calculating the concordance correlation coefficients between the datasets, the concordance was also calculated between datasets that were normalized for user interface difficulty. When playing a touch screen game on a user's personal android device, the user is already familiar with the user interface of the game. The users have ample experience touching buttons on their android devices and, thus, can immediately focus on learning the task without needing to learn a new user interface. However, when playing the game using the rehabilitation robotic exoskeleton control interface, users had to learn the interface while they were learning the task. Thus, we shifted the estimated number of trials before each participant was considered an expert at the performance metric, N_t, by -1 in order to remove user interface differences by allowing the exoskeleton users 1 trial to focus on learning the interface. For all shifted datasets, we set the minimum number of trials that a user could learn the task to 0, so a participant could not be shifted to requiring negative trials to learn the task.

The concordance coefficients for the normalized and actual datasets for each performance metric are shown in Table 8. By definition, a value of $\rho_c \geq 0.90$ shows correlation between methods used for data collection [34]. Before normalizing the data, 3 of the 4 parameters for the silent group and no parameters for the sound group showed correlation between user interfaces. After normalization, 3 of the 4 learning parameters for the sound group and 1 parameter for the silent group showed correlation. These correlations suggest that the virtual environment was an adequate simulation for evaluating motor learning that occurred with able$-$bodied participants who were interacting with our rehabilitation game. However, these trends also suggest that the sound group was more effected by the new user interface than the silent

Table 8: Concordance Coefficients between the Virtual Environment Dataset and a Normalized for Difficulty Exoskeleton Dataset

Performance Metric	Sound ρ_c	Silent ρ_c	Scaled Sound ρ_c	Scaled Silent ρ_c
Slow Note Count	0.78	0.92	0.90	0.88
Slow Average Period	0.73	0.96	0.94	0.90
Fast Note Count	0.64	0.56	0.90	0.75
Fast Average Period	0.72	0.93	0.35	0.62

group and that not all learning parameters were effected equally by the difficulty of the user interface. We hypothesize that this occurred because the users appeared to master the exoskeleton user interface about half way through the reconstruction phase of the first trial. Therefore, the parameters in the slow section were effected. However, in the faster section, the average period performance metric was unaffected by the users learning the exoskeleton interface while learning the task while the note count performance metric was effected. This occurred because learning the exoskeleton interface caused users to take longer for the slower section and run out of time during the faster section, preventing them from inputting enough notes.

4.3.4 Conclusions

Since participants learned performance measures significantly faster in the sound group than the no sound group, we conclude that audio-visual cues help participants learn tasks more quickly than visual cues alone in our virtual environment. Thus, we have been able to observe this trend in both physical systems and virtual environments.

4.3.4.1 Virtual Simulations of Rehabilitation Robotic Systems

In this experiment, we compared user performance of a task using a virtual environment to that of a physical environment using concordance coefficients. Our results showed high correlation after shifting the physical environment to normalize for interface difficulty. We conclude that our results were an adequate simulation for evaluating motor learning that occurred with able−bodied participants who were interacting with our rehabilitation game.

4.3.4.2 Multisensory Stimulus for Teaching Timing Synchronicity

Timing synchronicity is of direct relevance to the domain of motor function rehabilitation, where timing can a key attribute of many tasks that are learned. For example, the timing of leg movements is crucial for maintaining balance when walking and the timing and the sequence of forces applied to the object are important when picking up objects in order to maintain control of the object. In this experiment, we explored methods for teaching timing synchronicity effectively. Although the literature shows mixed reviews concerning the effectiveness of auditory cues on motor learning, our experiment indicates that using auditory cues, in addition to visual cues, may enhance the quality of timing with respect to a motor task. Since participants learned performance measures significantly faster in the sound group than the no sound group, we conclude that audio−visual cues help participants learn timing tasks more quickly than visual cues alone. This suggests that multisensory visuo−auditory cues are effective in the case of our timing synchronicity tasks. The limitations of this approach is that audio−visual cues may not be appropriate in all settings, which may explain why the literature review reported that multisensory stimulus was not always the most effective method for teaching motor learning tasks. While our results showed user benefits from multisensory stimulus, visuo−auditory may not always be most effective for teaching rehabilitation tasks. Thus, when considering which type of cues to use, one must consider the relevance of these cues to the skills being taught as well as the performance metrics for measuring said skills. For example, in the case of our experiment, we used auditory cues expedite the learning process of learning timing associated with wrist movements. Since timing is a skill that has auditory relevance, our results showed that auditory cues had a positive effect on learning. Thus, in the case of teaching timing synchronicity, our study suggests that multisensory visuo−auditory cues may be more effective than visual cues alone.

CHAPTER V

ADJUSTING THE DIFFICULTY OF A MOTOR TASK BY ADJUSTING THE DIFFICULTY OF MUSICAL PARAMETERS OF THE TASK

The intended application of our research is to develop a rehabilitation gaming experience in which the difficulty of the task is adjustable, allowing for the user to be most optimally challenged and thus learn the motor task in an optimal time line. To achieve this goal, we focused on developing a model that defines the correlation between task type and motor difficulty.

5.1 Music Theory and Rehabilitation

Many studies have been conducted on the effects of musical cues and sequences as a means of teaching motor tasks. However, when teaching, it is important to keep participants at the appropriate level of challenge so they learn most quickly. The challenge point framework asserts that learning may be increased for difficult tasks by providing the learner with a model of the task (i.e. an auditory or timing model) during practice [23]. This framework has been validated in a variety of studies. In a study by van Vugt and Tillmann, it was shown that a tap sequence could be learned more quickly and accurately by allowing participants to practice with tapping beat cues that corresponded to the times that a tap occurred in the sequence being learned, as compared to practicing in silence or in the presence of randomly timed auditory beats [43]. A study by Aluru et al. on stroke survivors with chronic hemiparesis suggests that different types of auditory stimulations can be effective during the different stages of recovery. During early stages of recovery, when a stroke survivor is suffering from spastic paresis, a metronome beat was shown to increase wrist extension and muscle co-activation. During mid stages of recovery, when a stroke survivor is suffering from spastic co-contraction, silence was shown to increase wrist extension but reduce co-activation. During the late stages of recovery, when a stroke survivor is suffering from minimal paresis, minimal gains were made regardless of the auditory stimulus [2]. To discover which auditory cues were most effective for assisting motor skill learning, Vinken et al. used kinematic-acoustical mapping to associate seven different auditory cue schemes to six everyday upper limb actions. In this study, the different auditory schemes did not affect the learning of the task, suggesting that any type of music can be used to teach any type of motor skill [44]. In another study by Butler and James, participants were asked to create sounds with novel musical objects. Then, they were asked to identify each object by the sound it produced. This data showed that the functional connectivity between visual- and motor-related processing regions was enhanced during the presentation of actively learned audiovisual associations [5]. These studies suggest that musical cues may help users learning more quickly and/or retain motor skills [43, 2, 44, 5].

5.2 Methodology

5.2.1 Musical Difficulty Definitions

In order to adjust task difficulty, first we must understand the parameters which need to be adjusted. Sebastien et. al. studied music theory and musical difficulty classifications to develop seven criteria that contribute to song classifications. Once this framework was completed, Sebastien et. al. validated these criteria and used machine learning techniques to automate the classification of instrumental sheet music [40]. The seven criteria that they deemed crucial for determining the difficulty of sheet music were: (1) playing speed, (2) fingering, (3) hand displacement, (4) polyphony, (5) harmony, (6) irregular rhythm, and (7) length. The definitions of these musical difficulty criteria are described in the following subsections.

5.2.1.1 Playing Speed

The playing speed is determined by how fast the music notes must be played. It correlates to the required velocity that the fingers must move to correctly play the piece of music. Musical parameters such as the tempo and the shortest significant note value determine the quickness of the playing speed. The tempo is a numerical value that is presented on a piece of sheet music that tells the musician the number of beats per minute a piece of music is intended to be played at. (For a more detailed description of the tempo, refer to Chapter 6 of this thesis.) The shortest significant note value refers to the occurrence of large groups of certain types of notes. For example, there are note types such as whole notes (which occupy the duration of 4 beats), half notes (which occupy the duration of 2 beats), quarter notes (which occupy the duration of 1 beat), eighth notes (which occupy the duration of 1/2 of a beat), assuming music in a 4/4 signature. So, higher tempos correlate to higher playing speed as well as large occurrences of shorter notes (i.e. eighth notes).

The musical difficulty parameter is relevant to sheet music for all instruments and is also relevant in context of our rehabilitation game.

5.2.1.2 Fingering

Fingering refers to the sequence of finger and hand positions required to correctly execute the sheet music. Some notes are played with the thumb, index finger, middle finger, etc. in isolation and some require multiple fingers to be used simultaneously. The difficulty of fingerings is largely dependent on the type of instrument that is being played. Thus, Sebastien et. al., C. Lin, and A. Kasimi et. al. used cost functions to determine fingering difficulties [40, 27, 25].

Fingering is a parameter that is relevant to classifying the difficulty of sheet music for all instruments. However, the classifications of difficulty are unique to each instrument. This parameter is not relevant to our rehabilitation game, since we use wrist position instead of fingers to activate notes.

5.2.1.3 Hand Displacement

The hand displacement criteria is the amount of distance that you need to move your hands or fingers between notes in order to correctly hit all of the notes. The difficulty associated to hand displacement is dependent on the amount of time that the musician has to move between the notes. The larger the distance a musician must move, the greater the difficulty. Also, the less time the musician has to complete such a movement, the greater the difficulty.

Hand displacement is a criteria that is relevant for all instruments. However, the criteria used for assessing this parameter is unique for each instrument. Hand displacement is also relevant for our rehabilitation game.

5.2.1.4 Polyphony

Polyphony, by definition, occurs when multiple notes are played at the same time. The polyphonic difficulty is determined by the number of notes that are being playing during a single instance. More notes being played simultaneously associates to a higher polyphonic difficulty and fewer notes associates to a lower polyphonic difficulty.

Polyphony is only a relevant criteria for musical instruments that allow for multiple notes to be played at the same time, such as the piano and string instruments like the violin and guitar. It is not relevant for our rehabilitation game, since a user's arm can only be in a single position at a time.

5.2.1.5 Harmony

Harmony is defined as the ratio notes that have a tone that differs from the piece's main tonality. Notes with a different tone are defined to be notes that are altered by an indication of a sharp symbol ($\sharp$) or a flat symbol ($\flat$). Sheet music that has a higher ratio of notes with different tones is defined to be more difficulty than sheet music that has a lower ratio of notes with different tones.

The criteria harmony is relevant to all musical instruments. However, it is not relevant to our rehabilitation game, because we have a simplified selection of musical notes that are available to the user to play. Thus, our game does not include options for sharp notes and flat notes. Our game contains a total of 9 note options, which is a much smaller range and therefore simpler than musical instruments. For example, a piano is capable of playing 88 different musical notes - 52 in the most common tonality and 36 sharps/flats.

5.2.1.6 Irregular Rhythm

Irregular rhythm is defined as rhythms that differ from the rhythm that is denoted by the time signature. Time signatures are numerical values in sheet music that define what type of note is considered a beat. A beat is defined to be the basic unit of time within the measure (more details about these concepts can be found in chapter 6 of this thesis). Notes which are the same type as the note that is defined to be a beat by the time signature are considered to be regular. All others are considered to be irregular. The more different these notes are from the beat, the more difficult they are considered to be. Also, notes that have odd number relations to the beat unit of time are considered to be more difficult than those that have an even number relation to the beat unit of time. For example, a note that is 1/2 of a beat is considered easier to a note that is 1/3 of a beat.

Irregular rhythms are relevant criteria in determining the difficulty of sheet music for all musical instruments, as well as for our rehabilitation game.

5.2.1.7 Length

The difficulty criteria of length is the duration of time that it takes to completely execute a piece of sheet music. Longer sheet music is considered to be more difficult than shorter sheet music. This parameter is relevant for sheet music for all musical instruments and is also relevant for our rehabilitation game.

5.2.2 Calculating Musical Difficulty for Rehabilitation Game

We created a version of the RoboRockNRoll game that allowed for us to adjust the difficulty of tapping sequences using the difficulty definitions set forth by musical theory. We adjusted the difficulties of speed, length, and irregular rhythm in order to adjust the song difficulties. While the trends of the effects of adjusting these difficulty parameters on musical difficulty are defined, a formula that expresses such effects is not defined within music theory. Also, many of these parameters have unique properties that are dependent upon what instrument is being played. The relationships between these variables and difficulty has been discovered for many popular musical instruments by Sebastien et. al. using machine learning techniques. However, since our rehabilitation game, RoboRockNRoll, acts as a unique, simplified instrument, a model for the trade-offs between criteria for this situation is not yet defined. In this section, we aimed to conduct user studies of our rehabilitation game to determine the mathematical relationship between three musical parameters (speed, length, and rhythm irregularity) and the difficulty of our game. In these experiments, our focus was on developing a model that defines the correlation between task type and motor difficulty [15].

5.3 Experimental Setup

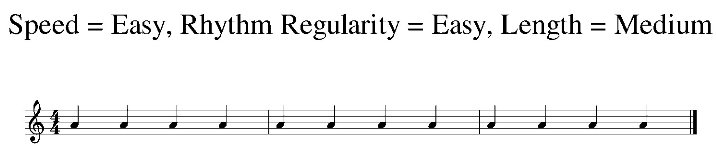
We conducted a virtual experiment on Amazon's Mechanical Turk [15]. In this study, participants downloaded the virtual game shown in Figure 11, played a short gaming app, and completed a post−game survey. 208 participants completed this experiment. 47 participants were female, 144 were male, and 17 preferred not to answer this question. The age of participants ranged from 18 to 74. The average age of participants was 29.93 and the standard deviation was 8.35.

Table 9: Difficulty Definitions

Musical Parameter	Difficulty		
	Easy	Medium	Hard
Speed	Quarter Notes	Eighth Notes	Sixteenth Notes
Length	2 stanzas	3 stanzas	4 stanzas
Rhythm	1	1/2	2/3

Participants were randomly assigned to one of seven test groups, each of which represents a different task difficulty. In order to validate the effects of adjusting musical parameters on the overall task difficulty, three musical parameters were adjusted in this experiment. The three parameters selected were speed, length, and rhythm. Speed is defined as how quickly the music notes appear. Length is defined as the number of measures required to complete the tapping pattern. Rhythm is defined as the ratio of the speed of the music notes in the first half of the song as compared to the second half of the song. Table 9 shows the definitions of each difficulty level tests for each of the difficulty parameters. In the control group, speed, length, and rhythm were all easy difficulty and was represented by Group #1, as seen in Table 10. For each of the six experimental groups, one of the musical parameters was adjusted while keeping the other two at an easy difficulty. As seen in Table 10, the test groups were as follows: (2) easy speed, medium length, and easy rhythm, (3) easy speed, hard length, and easy rhythm, (4) medium speed, easy length, and easy rhythm, (5) hard speed, easy length, and easy rhythm, (6) easy speed, easy length, and medium rhythm, and (7) easy speed, easy length, and hard rhythm. The sheet music corresponding to each test group can be seen in Figure 15.

(a) Group #1 (Control)



(b) Group #2

(c) Group #3

(d) Group #4

(e) Group #5

(f) Group #6

(g) Group #7

Figure 15: The sheet music that represents the tapping sequence played by participants in groups #1-#7.

Table 10: Group Definitions

Group#	Speed	Length	Rhythm
1	Easy	Easy	Easy
2	Easy	Medium	Easy
3	Easy	Hard	Easy
4	Medium	Easy	Easy
5	Hard	Easy	Easy
6	Easy	Easy	Medium
7	Easy	Easy	Hard

5.4 Performance Measures

During the reconstruction phase of the interaction, the participants' screen taps were monitored. Since groups 6 and 7 had a switch in the types of notes presented, the tapping sequence for these groups was separated into two sections: the slower section and the faster section. We define these sections as follows:

5.4.0.1 Slower Section

The slower section is defined as the first portion of the experiment in which four slow taps were presented to participants. This is shown as the first four notes in Figures 15.f and 15.g.

5.4.0.2 Faster Section

The faster section is defined as the second portion of the experiment in which six or eight taps, played at a faster frequency that the first portion, was presented to participants. This is shown as the last six or eight notes in Figures 15.g and 15.f, respectively.

To evaluate performance, screen taps were analyzed and separated into two performance measures: (1) note count and (2) average period.

5.4.1 Note Count

For this performance measure, the number of taps was recorded. To achieve perfect performance, a user must tap the patterns depicted in the sheet music shown in Figure 15. For this performance measure, the note counts at time t=0 were initialized to 0. As seen in Equation 13, the note count (which is represented as N_{note}) is the summation of the occurrences that the user inputs a tap (which is represented as $note(t)$). In the case of groups 6 and 7, the note count was calculated separately for the first and second halves of the tapping sequence.

$$N_{note} = \sum note(t) \tag{13}$$

5.4.2 Average Period

The average period, or the inverse of the frequency, that each participant tapped was calculated for each participant. A perfect performer would have a period of 1250ms for quarter notes, 625ms for eighth notes, 312.5ms for sixteenth notes, and 833.33ms for quarter triplets. For this performance measure, the average period at time t=0 was initialized to 0. As seen in Equation 14, the average period (which is represented as T_{note}) was calculated by dividing the difference between the time that the last note in the section was tapped (which is represented as $t_{lastnote}$) and the time that the first note in the section was tapped (which is represented as $t_{firstnote}$) by the note count for the section (which is represented by N_{note}). In the case of groups 6 and 7, the average period was calculated separately for the first and second halves of the tapping sequence.

$$T_{note} = \frac{t_{lastnote} - t_{firstnote}}{N_{note}} \tag{14}$$

5.4.3 Learning Curves

For each participant and each performance metric, we calculate a learning curve that best fits the data. There exist a variety of shapes of learning curves. However, in this context of learning, the performance measures can be described as showing large amounts of improvements early in learning. Then, the performance measures begin to level off and approach a horizontal asymptote. This type of learning curve is best fit by the power law learning curve [30, 35]. Thus, we calculate a power law learning curve that fits to each of the performance metrics for each of the participants. The exact formula used to calculate the best−first learning curve can be seen in Equation 15, where P is the estimated performance, a is the estimated performance of the 0th trial (i.e. time = 0), N_{trial} is the number of trials, and b is the log−log slope of the curve.

$$P = a \times (N_{trial})^{-b} \tag{15}$$

5.5 Data Analysis

5.5.1 Learning Curve Calculations

For each participant, the error for each performance metric was graphed for each trial. Error was computed by calculating the difference between actual performance and perfect performance, as defined in the Performance Measures section. For each performance metric, an exponential curve was fit to the data points. The best−fit exponential curve represents the learning curve for that participant associated with each performance metric. To find the best fit exponential curve, we used the power law learning curve formula, as seen in Equation 15. The correlation coefficient between the learning curve and the set of data points for each metric was also calculated. If the correlation coefficient was similar enough to the data that it had a p−value < 0.1, the curve was considered to be a fair representation of the data. Otherwise, we determined that, in that instance, learning did not occur.

5.5.2 Calculating Expertise

An expertise value was calculated for each performance metric for each participant, using Equation 16 and the calculated learning curves. For the initial trial at time=0, all participants were considered to have equal knowledge, corresponding to an initial baseline performance level of 0. When the error between actual and perfect performance converged to zero, the participants were considered to have mastered 100% of the performance metric. We considered the participants to be experts at the performance metric when the learning curve estimated that they would perform at 90% of mastery of the performance metric. 90% was chosen since this associates to the highest letter grade that is possible to achieve on the education grading scale [38]. As seen in Equation 16, the number of trials to achieve this expert level was calculated by taking the inverse of $1 + N_t$, where N_t is the number of trials until an expert level is achieved. Using this equation, the expertise of each participant was calculated by taking the inverse of the number of trials needed to perform at an expert level.

$$Expertise = \frac{1}{1 + N_t} \tag{16}$$

5.6 Results

As shown in Table 11, the means and standard deviations for the expertise of each participant were calculated for each performance metric. Although the groups are not large enough to have statistically significant groups, trends emerge. In groups 1-3, we adjusted the length of the tapping pattern in an attempt to make the tapping task more difficult. Group 1 represents the easiest length, group 2 the medium length, and group 3 the hard length. However, in the context of our simple experiment, the length changes did not appear to have an effect on the users' ability to learn the task.

In groups 1, 4, and 5, we adjusted the speed of the notes in an attempt to change the difficulty of the tapping sequence. Groups 1, 4, and 5 represent easy, medium,

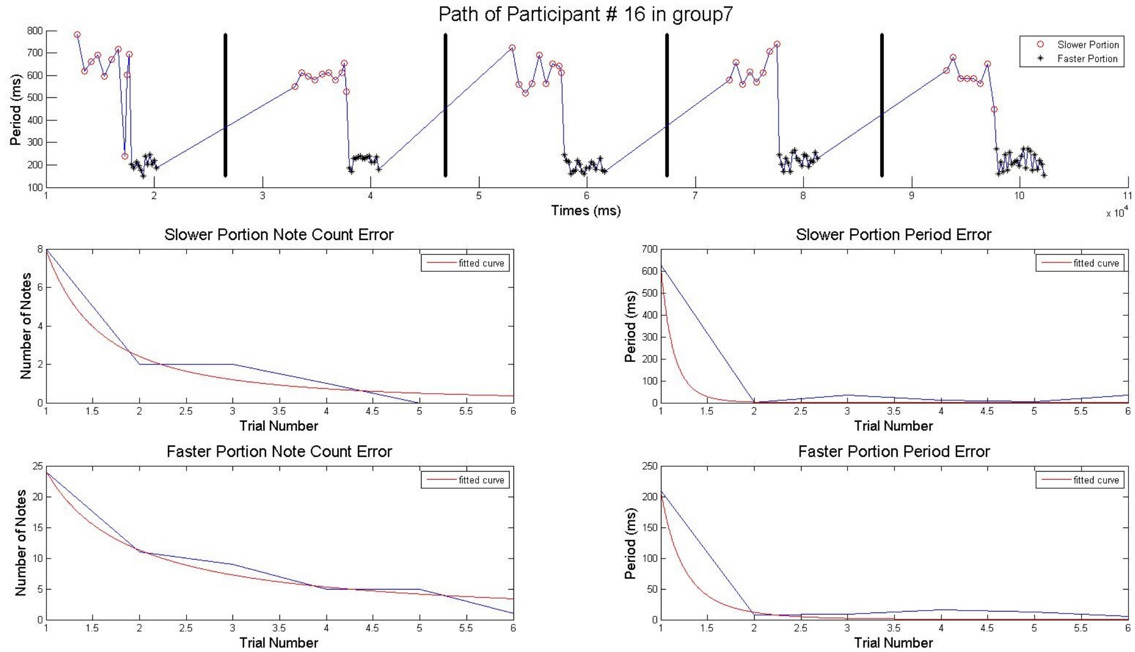


Figure 16: A participant's data and learning curves.

and hard speeds respectively. When looking at the learning trends of participants in these groups, as shown in Table 11, it is noticed that participants learn the note count performance measure more quickly when the task is easier and more slowly when the task is harder. However, it is noticed that the opposite trend is true of the average period performance measure. The easier the task is, the more slowly participants learn the average period, and the more difficult the task is, the more quickly participants learn the average period.

In groups 1, 6, and 7, we kept the first half of the tapping sequence the same but shifted the rhythm of the second half of the song. Groups 1, 6, and 7 represent easy, medium, and hard rhythm shifts, respectively. When looking at the learning trends shown in Table 11, we notice that the same trends are seen when adjusting the difficulty of the rhythm as were seen when adjusting the difficulty of the speed.

In order to show statistical significance to these trends, super-classes of tapping sequences that presented the users with easy, medium, and hard difficulty were created. The easy super-class contains groups 1, 2, and 3, and the slower portion of

Table 11: Means, Standard Deviations, and Number of Included Participants for Each Performance Metric for Each Test Group

Group #	Slow Note Count			Slow Average Period		
	Mean	Std Dev	# Participants	Mean	Std Dev	# Participants
1	0.3945	0.2882	15	0.2847	0.2056	13
2	0.3331	0.2055	23	0.2810	0.1589	21
3	0.4477	0.2461	22	0.2968	0.1890	20
4	0.2397	0.2363	29	0.3174	0.1698	24
5	0.1386	0.1386	20	0.3754	0.1726	18
6	0.4576	0.4576	24	0.3222	0.1994	25
7	0.3259	0.3259	25	0.2265	0.1683	24
Group #	**Fast Note Count**			**Fast Average Period**		
	Mean	Std Dev	# Participants	Mean	Std Dev	# Participants
6	0.1779	0.1344	24	0.3869	0.2053	24
7	0.1011	0.0675	22	0.4197	0.1527	23

groups 6 and 7. Although we attempted to adjust the difficulty in groups 2 and 3, they were considered easy since the users did not appear to be effected by the adjustments made to the length of the tapping sequence. The medium super-class contains groups 4 and 6 and the hard super-class contains the faster portion of groups 5 and 7. Table 12 shows the means, standard deviations, and number of data points for each of these super-classes. A t-test was performed to compare the medium and hard super-classes to the easy super-class and the resulting p-values can also be found in table 12. The trends seen in the data from the individual groups are consistent with the trends seen in the data in this super-classes. In addition, these super-class trends are all statistically significant with p-values < 0.05.

5.7 Discussion

In our experiment, expertise values were calculated that correspond to the speed at which the participants were able to learn the task. This expertise value was calculated for two performance metrics: (1) note count - how quickly the participant learned how many notes were in the sequence and (2) average period - how quickly the participant

Table 12: Statistical Analysis for Each Performance Metric With Participants Combined into Easy, Medium, and Hard Super-Classes

	Note Count			
Group	Mean	Std Dev	# Points	p-value
Easy Super-Class	0.3904	0.2706	109	—
Medium Super-Class	0.2117	0.1968	53	0.0001
Hard Super-Class	0.1189	0.1292	42	0.0001
	Average Period			
Group	Mean	Std Dev	# Points	p-value
Easy Super-Class	0.2818	0.2181	103	—
Medium Super-Class	0.3521	0.1292	48	0.0403
Hard Super-Class	0.4002	0.1618	41	0.002

learned the correct period to tap the sequence. When the difficulty of the tapping sequence was adjusted, participants learned the note count performance metric more quickly for easier sequences and more slowly for harder sequences. However, the opposite was true of the average period performance metric. Participants learned the average period performance metric more quickly for harder tasks and more slowly for easier tasks.

The difference in the learning trends could be explained by these performance metrics measuring different aspects of learning. The average period metric measures motor learning, since tapping at a specific frequency is a motor function, while the note count performance metric measures cognitive learning, since counting the number of taps is a cognitive function. Our experiment suggests that in context of teaching a simple tapping sequence, participants are multitasking cognitive and motor aspects of the task. The simpler the task, the quicker participants learn cognitive aspects of the task and the more difficult the task, the quicker participants learn motor aspects of the task. When multitasking different types of learning, a performance trade-off is seen. However, the increased neural activation and cortical networking in the motor-planning region of the brain when participants are performing more difficult tasks as compared to easier tasks and the increased cognitive demands in regionally specific activation [39] may have caused this increased performance in the average period performance metric for the more difficult tasks. Since the main focus of this is the motor task, participants may be prioritizing the motor task over the cognitive task, allowing them to reap the benefits of this increased brain activity for the aspect of the task that the participant is focusing on.

5.8 Conclusions

This experiment discusses an approach for teaching multitasking aspects of a tapping task. The results from this study suggest that in the context of teaching a simple tapping sequence to healthy subjects, participants are multitasking cognitive and motor aspects of the task. The simpler the task, the quicker participants learn cognitive aspects of the task and the more difficult the task, the quicker participants learn motor aspects of the task. While this experiment was performed using able-bodied participants, the benefits of multitasking on learning specific aspects of a task may be able to be transferred to a stroke survivor population in context of a therapeutic setting. Therefore, our study suggests that if a therapist were wanting to emphasize cognitive aspects of a task and have their client show greater improvements with the cognitive portions of the task, they should encourage their client to practice simpler tasks. However, if they would like to see greater benefits with the motor portions of the task, they should encourage their clients to practice more difficult tasks. If these benefits could be transferred to therapy, an expedited recovery could be realized by stroke survivors, allowing them to regain motor functions, and therefore daily living activities, more quickly.

CHAPTER VI

TRANSLATING SPATIAL DATA TO TEMPORAL DATA

A goal of this research is to classify the difficulty of a task. Adjusting task difficulty via adjusting musical theory difficulty criteria did not produce consistent user performance, as seen in chapter 5 of this thesis. Thus, in this chapter, we are exploring a different approach for classifying task difficulty. Having a knowledge of which movement patterns associate to what difficulty level will allow us to adjust our game to be more and less difficult as needed. However, we cannot make such adjustments without correctly classifying the difficulty levels of a variety of patterns. In our context, we would like to classify the difficulty of movement tasks within the RoboRockNRoll rehabilitation game, shown in Figure 3.

Classification can be easily achieved by machine learning algorithms. However, such algorithms require large datasets, which are difficult and time consuming for researchers to collect themselves with user data of our system. Thus, we have pulled user data from a similar, popular, commercially available game called Guitar Hero. Guitar Hero can be played on PlayStation 2, PlayStation 3, Xbox 360, Wii, Microsoft Windows, and Mac OS X platforms [1]. An example of guitar hero can be seen in Figure 17 and the controller used to play the game can be seen in Figure 18. When comparing the Guitar Hero game with our RoboRockNRoll rehabilitation game, many similarities exist. Both games involve notes traversing the screen and approaching specific area. In both games, the notes must be selected at the correct time in order to succeed at gaining the points associated with the music notes. However, there are a few differences. For example, in the Guitar Hero game, music notes move vertically towards a horizontal bar, and in the RoboRockNRoll game, music notes

Figure 17: Guitar Hero 3 [33].

move horizontally towards a vertical bar. The note shapes are also slightly different. However, the most notable differences are in the controller. In Guitar Hero, the user presses fret board buttons (shown in Figure 18) to select the correct note color and press the strum button (shown in Figure 18) at the correct time. In RoboRockNRoll, the passive wrist rehabilitation device shown in Figure 2 is used to control the game. The users move their wrists up and down, reaching the desired location at the correct time to collide a guitar pick with the musical note. Also, in our rehabilitation game, there are 9 horizontal positions that notes can occur at, but only one can be presented at a time. In the Guitar Hero game, there are only 5 vertical positions that note can occur at, but multiples can be presented to the users at the same time.

In order to classify the difficulty of patterns from Guitar Hero and translate these difficulty classifications to our rehabilitation game, we first need to obtain the game patterns within the Guitar Hero game. A representation of these patterns is available online in the form of fan created sheet music posted to a wiki fan page. Fans of the Guitar Hero video game can create sheet music for each level of Guitar Hero,

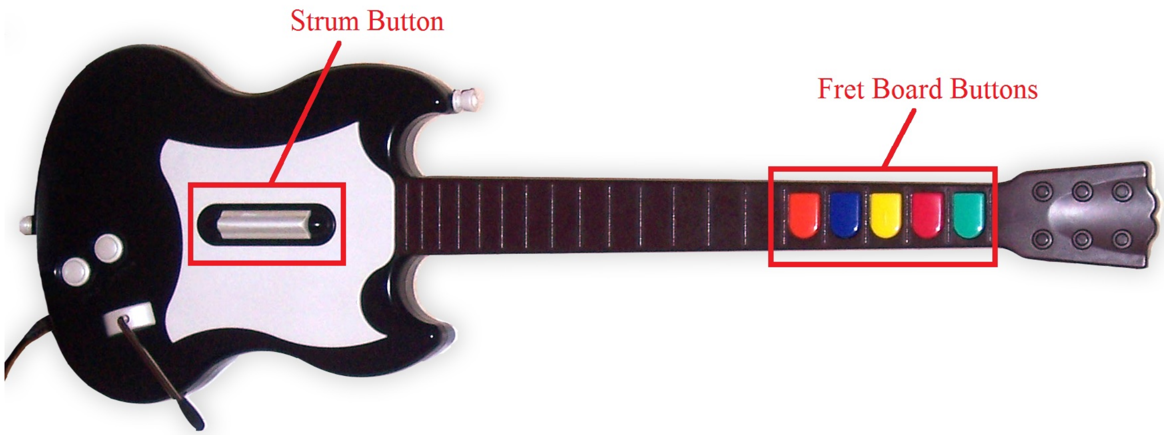


Figure 18: Guitar Hero guitar game controller [45].

share them on this website, and validate existing sheet music to promote accuracy. We downloaded these sheet music files from the website and used computer vision techniques to translate the sheet music into a temporal representation of each button press. This chapter describes the process that we used to translate the images to a temporal representation of the pattern. Once we have created these patterns, we will be able to use machine learning techniques to classify pattern difficulty from features within the patterns and, thus, adapt our pattern difficulty within our rehabilitation game. Figure 19 shows an example of the Guitar hHro sheet music. Music signatures, tempo, measure bars, and notes must be detected and analyzed in order to translate an image of sheet music, like the one shown in Figure 19.

6.1 Encoding Sheet Music

In order to analyze the properties of the musical patterns presented to users in Guitar Hero, we must first process the data and convert it to a useable format. We are interested in reading the sheet music images of each song, as a musician would do as they play their instrument. The key pieces of information that we need to read in order to understand the meaning of each pattern are (1) measure bars, (2) music

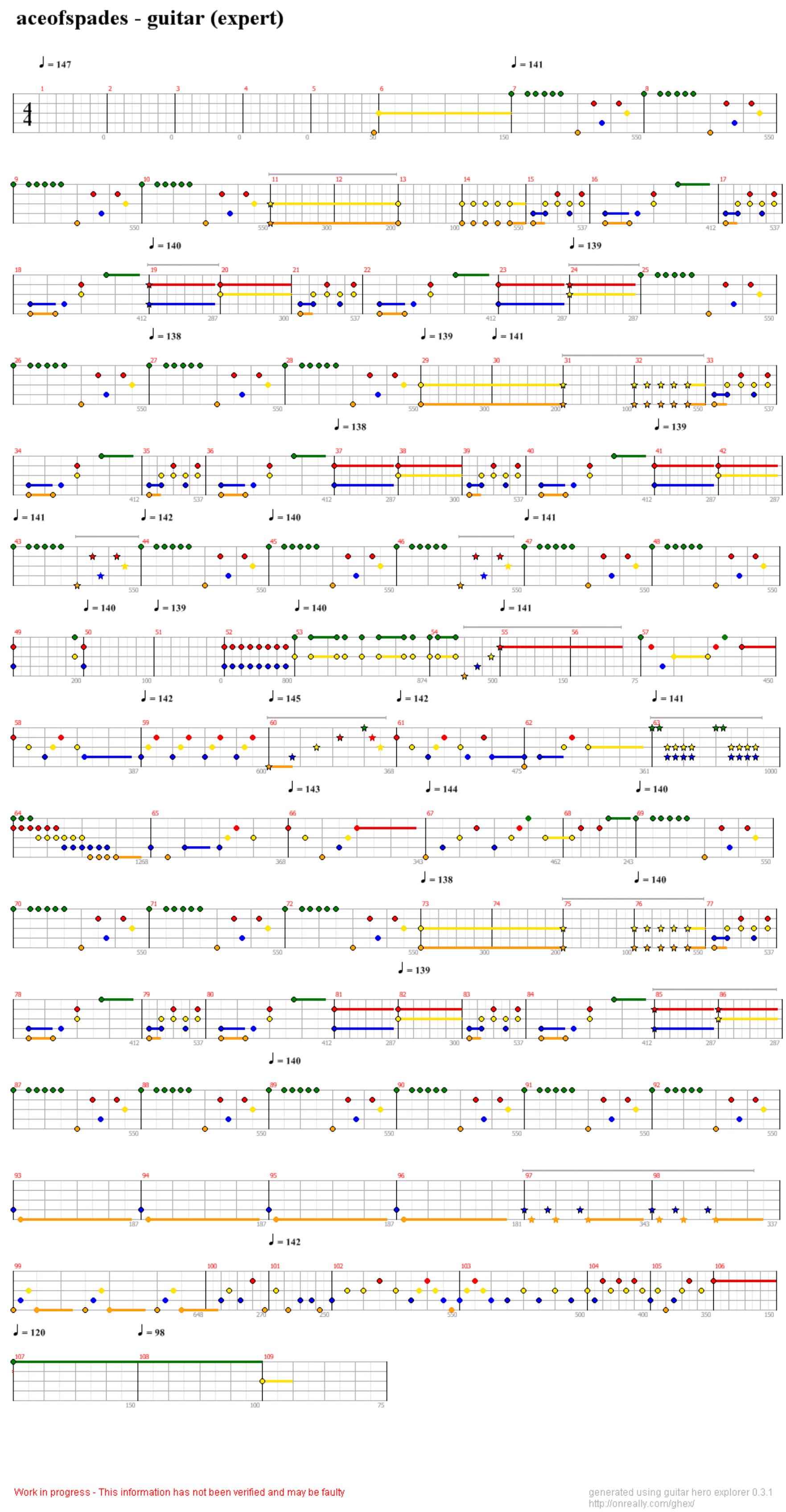


Figure 19: Sheet music representation of the expert difficulty level of the "Ace of Spades" song [24].

Table 13: Musical Term Definitions

Number	Musical Term	Definition
1	Measure Bar	A segment of time corresponding to a specific number of beats.
2	Note	A pitch and duration of sound.
3	Signature	A notation to specify how many beats are contained within a single measure and which note value is equivalent to a single beat.
4	Tempo	The speed at which a section of music is played.
5	Beat	The basic unit of time within a measure.

notes, (3) signatures, and (4) tempo. These key terms are defined in Table 13. In this section, we will describe how we detect and process each of these pieces of information.

6.1.1 Detecting Measure Bars

When music is written, it is partitioned into smaller groups so that it is more manageable for the music reader to follow and interpret, in a similar fashion to the way words are partitioned into sentences in writing to help the reader interpret the meaning of the ideas. These measures are indicated by the darkest vertical lines as well as the small red numbers above the line of music near the darkest vertical lines. In the Guitar Hero sheet music, there are also many grey lines that indicate the sections of different beats. These sub-measure lines are somewhat common in expressions of guitar chords, as they emulate markings of a guitar's fret board. However, they are uncommon in normal sheet music. So, for this computer vision image processing, we ignore them to allow our code to be more useful across the majority of sheet music.

The measure bars indicate a separation between two measures. This means that the area between two measure bars constitutes one measure and that the number of beats indicated by the signature should exist between them with an equal spacing between each of these beats. In the case of notes that are faster than the beat, a combination of notes that is equivalent to a single beat should be spaced accordingly such that the group of faster notes takes up the same amount of space as a single beat. Measures can be different sizes, depending on how complex the pattern of notes are in that measure. Those that have no notes or only notes that make up a single beat will be smaller, since there are fewer notes to print in that measure. Measures that have many fast notes that are faster than a beat will be larger, as there is more space needed to fit all of the notes. However, within the measure, the spacing will be consistent, meaning that the distance between notes indicates how fast the notes should be played. Many examples of different measure sizes can be seen in Figure 19. For example, on the first line, measures 2-5 contain no notes and have a smaller distance between them than measures 7 and 8, which contain a complex pattern of notes. Despite the differences in physical lengths, we know that each measure indicates the same amount of time as every other measure, given the same tempo and signature.

Another case that we have to note for signature bars is the case when a signature is included in the measure. This can be seen in measure 1 in the top left hand corner of Figure 19. In this case, the measure area starts at the light grey vertical line directly following the signature value (which has a corresponding red measure number) instead of at the darkest black line at the beginning of the line. This allows for room to write the signature without interfering with the placement of musical notes that follow. However, as we process these images, this case must be accounted for.

A pseudocode representation of the code that was written to detect measure bars can be seen in Figure 20. This process involves 2 steps: (1) detecting the lines of the measure and (2) detecting the numbers of the measure.

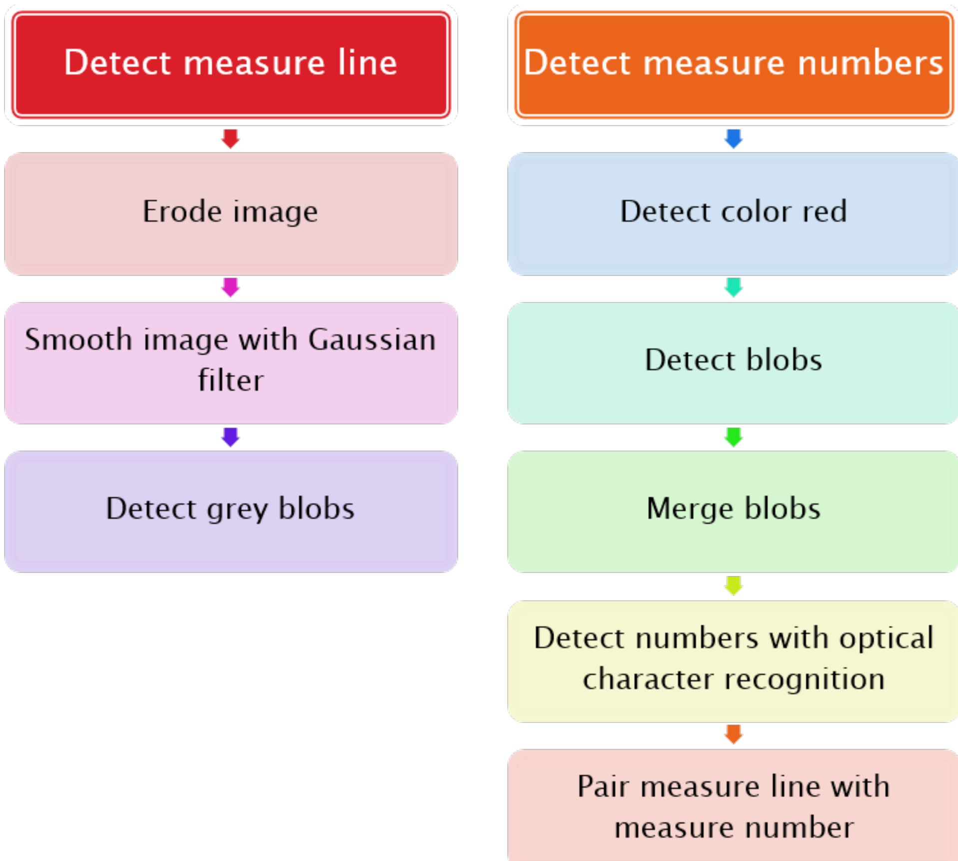


Figure 20: Flow chart showing the processed used to detection of measure bars in the Guitar Hero sheet music.

First the lines of the measure bars are detected. In this process, we create a binary black image by isolating the black pixels from the rest of the image. However, this image contains more than just measure bars. It also contains the other black objects, which include tempos, signature numbers, the outlines of the notes, and the small numbers beneath each measure bar. Next, we erode the image, using Equation 17, in order to thicken all of these components, making the measure bar lines easier to detect and read. The output of Equation 17 gives us the set of pixel locations z, where the structuring element translated to location z overlaps only with the foreground pixels of A, thus resulting in a set of pixels that outlines each of our blobs. Next, we smooth the image using a Gaussian smoothing filter, shown in Equation 18, in order to fill in any holes in the blobs that were caused by the existence of colored music notes overlapping the measure bar line whose existence creates a gap in the black vertical line. Next, we detect the blobs by looking for contiguous regions in the image. These blobs are then filtered to isolate blobs with sizes and proportions similar to those of measure bars.

$$A \ominus B = \{z | B_z \subseteq A\} \tag{17}$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{18}$$

Second, we detect the measure numbers. In this process, as shown in the pseudocode in Figure 20, we first create an image of all of the red pixels by simply isolating all of the red pixels and creating a new image from them. Next, we detect all of the blobs in the red image. These blobs are then merged to allow for nearby blobs (for example, the 2 numbers contained within a 2-digit number) to become a single blob. Next, optical character recognition is utilized to identify whether or not a blob is a number. All blobs that contain a number are considered to be measure numbers and all others are thrown out.

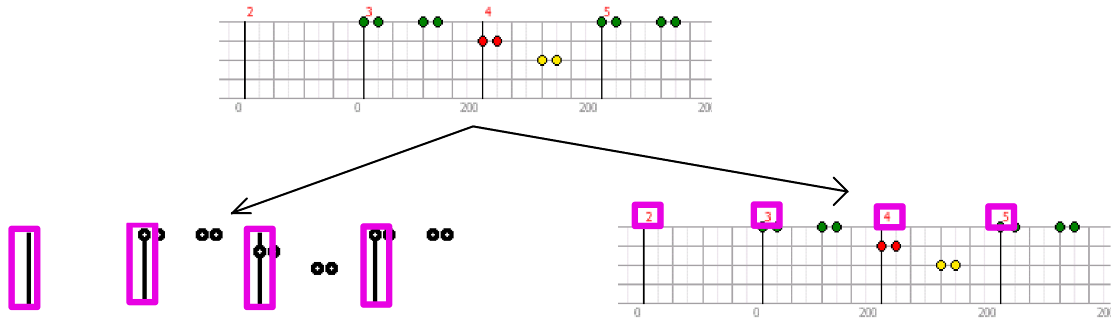


Figure 21: Detection of measures in the Guitar Hero sheet music. The top shows the input image and bottom images show the output images, with the left bottom image showing the output for detecting measure lines and the right bottom image showing the output for detecting measure numbers. Within the output images, the magenta squares surrounding the measure bars (on the right) and the measure numbers (on the left) show the measure bar and measure number blobs that have been correctly detected. These items are then paired to identify the existence of a measure bar.

Finally, we pair the detected measure lines with measure numbers. An example of the input and output images from this process can be seen in Figure 21. The top image in Figure 21 shows an input section of sheet music. The 2 lower images show the outputs of our 2 branches of psuedocode. In the right branch, the measure bars are detected. In the left branch, measure numbers are found. When a measure number occurs at the top of a detected measure bar, they are paired and determined to be a correctly identified measure bar. Any instance where a measure bar and number do not occur near each other are not considered to be measures. Such an instance could occur when a measure bar is falsely detected when 3 or 4 musical note outlines occurred at the same instance. This blob of circle outlines can be falsely identified as a measure bar. However, there is never a measure number above such an occurrence and, thus, it is correctly identified as being something that is not a measure bar.

6.1.2 Detecting Notes

Another piece of information that we need to extract from the sheet music in order to allow us to understand the musical pattern is the music notes themselves. In normal sheet music, music notes are placed on the horizontal lines. The lengths of time that the notes are played are indicated by their shape and the tone that they represent is indicated by their vertical position. However, Guitar Hero sheet music is slightly different from the traditional representation from music. In the Guitar Hero sheet music, like the sheet music shown in Figure 19, these notes are either a circle or star shape. However, the shape does not represent the duration of time that the note is played, but represents a game state that provides extra points instead. Thus, shape is an irrelevant piece of information for translating the image into temporal information. Each button pressed is the same duration of time, which is a single instance of pressing the button. In the case that a button must be held for longer than an instance, there is a line tailing the initial button press indicator. An example of this can be seen in the final line of sheet music in Figure 19. In this line, a green line that spans 2 measures can be observed. This indicates that the user would press the button at the beginning of the first measure of that line and hold that button down for the duration of time that corresponds to 2 measures. Also unlike traditional music, Guitar Hero sheet music has colored music note indicators that match the color that coordinates with the color of the corresponding button on the game controller (as shown in Figure 18).

The musical note features are the simplest features to extract, because of their vibrant colors that are only seen within the note features in these images. The pseudocode for detecting musical notes can be seen in Figure 22. First, we create a series of images that isolate the colors of the notes. One image is created for green, one for red, one for yellow, one for blue, and one for orange. These images are then merged into a binary image that contains only these colors. Next, a blob detection algorithm is run to detect all of the blobs in the image. These blobs are then filtered to only include blobs that are similar shape and proportion to a musical note. Figure 23 shows an example of an input section of the sheet music (the top image in the figure) and an output image where all of the music notes are highlighted with a magenta

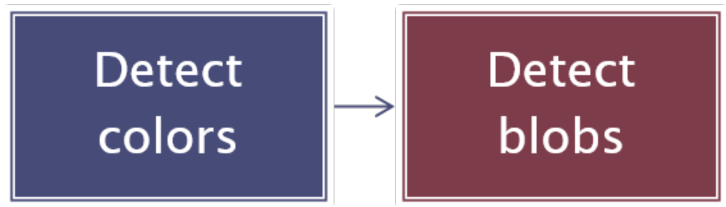


Figure 22: Flow chart showing the processed used to detection of notes in the Guitar Hero sheet music.

square (the bottom image in the figure).

6.1.3 Detecting Signatures

Time signatures are a set of two numbers, one on top of the other, that are placed at the beginning of a measure. The bottom number indicates the type of note that constitutes a single beat in the following measures. Examples of types of notes that could represent a beat are quarter notes (which is represented by a 4) or eighth notes (which is represented by an 8). In Guitar Hero notation, a quarter note would appear on the four darker grey lines within a measure with no notes in between. The two red notes in measure 68 of Figure 19 are examples of quarter notes. All of the notes in measure 104 in Figure 19 are examples of eighth notes, which occur when a musical note appears on every grey line in the measure.

An example of a time signature can be seen in the first measure in the upper left hand corner of Figure 19. In this example, the signature is 4 over 4. The bottom number is a 4, indicating that a quarter note constitutes a beat, while the top number indicates how many beats will be in each of the following measures. Since the top number in this example is a 4, there will be 4 beats (or quarter notes) in each measure.

Pseudocode summarizing the process used to detect time signatures can be found

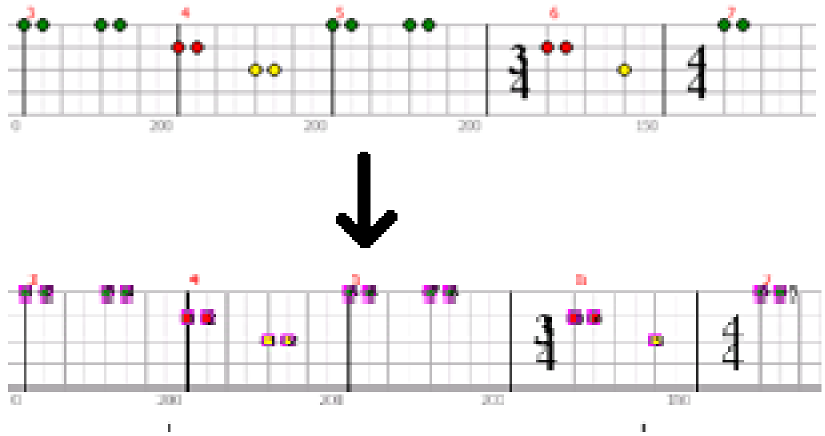


Figure 23: Detection of notes in the Guitar Hero sheet music. The top shows the input image and bottom shows the output image. Within the output image, the magenta squares surrounding each note show the note blobs that have been correctly detected as for our software.

in Figure 24. Since the signatures values are drawn in black lettering in all of our images, we first implement a simple color filter on the image to extract all black pixels from the image. The resulting image contains all objects drawn in black, which includes the measure bars, signature numbers, the outlines of the notes, and the small numbers beneath each measure bar. So, we continue to process the image to isolate the signature values from all of these other types of musical objects. Next, we erode the image, using Equation 17, in order to thicken all of these components, making the signature numbers that are of interest easier to detect and read. The output of Equation 17 gives us the set of pixel locations z, where the structuring element translated to location z overlaps only with the foreground pixels of A, thus resulting in a set of pixels that outlines each of our blobs. Next, we smooth the image using a Gaussian smoothing filter, shown in Equation 18, in order to fill in any holes in the blobs that were created by the blob containing pixels whose values were slightly outside of the range of our color filter. Next, we detect the blobs by looking for contiguous regions in the image. These blobs are then filtered to isolate blobs with sizes and proportions similar to those of signature values. Finally, the blobs are

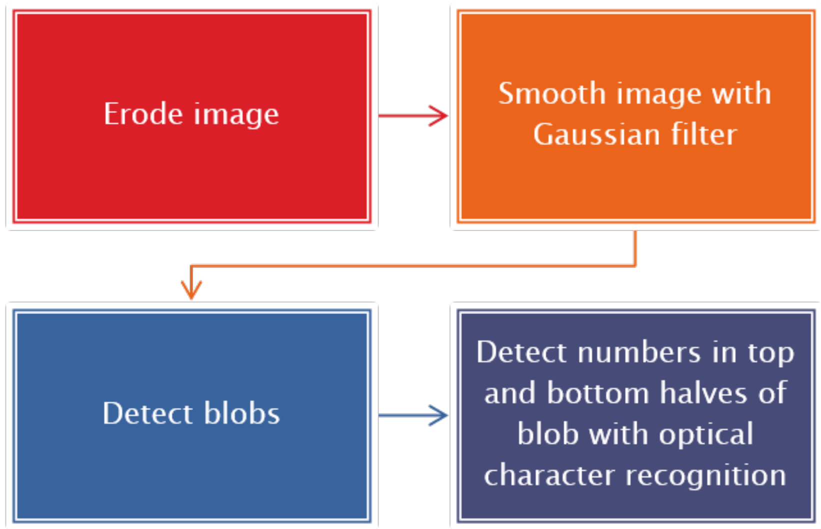


Figure 24: Flow chart showing the processed used to detection of signatures in the Guitar Hero sheet music.

divided into half and optical character recognition is used on the top and bottom halves of the image to detect the numerical value of the blob. Examples of the input and output images for signature detection can be seen in Figure 27.

6.1.4 Detecting Tempo

The final piece of information that we need to gather in order to understand the temporal pattern of the music is the tempo. Tempo corresponds to the number of beats that are heard per minute. The tempo is indicated by the image of a note which is followed by an equals sign which is followed by a number. The meaning of the word "beat" is determined from the signature, as discussed in the "Detecting Signatures" section in this chapter. The tempo indicator is placed above the measure that it pertains to and its value continues to dictate the speed of the music until a new tempo is displayed. In Figure 19, there are several examples of tempo changes. Two examples can be seen above the first line of music. The first example shows a value of 147, meaning that the music should be played at a speed of 147 quarter note beats per minute, or that a quarter note would last approximately 0.408 seconds. The music is played at this speed for 6 measures. At the 7th measure, a new signature

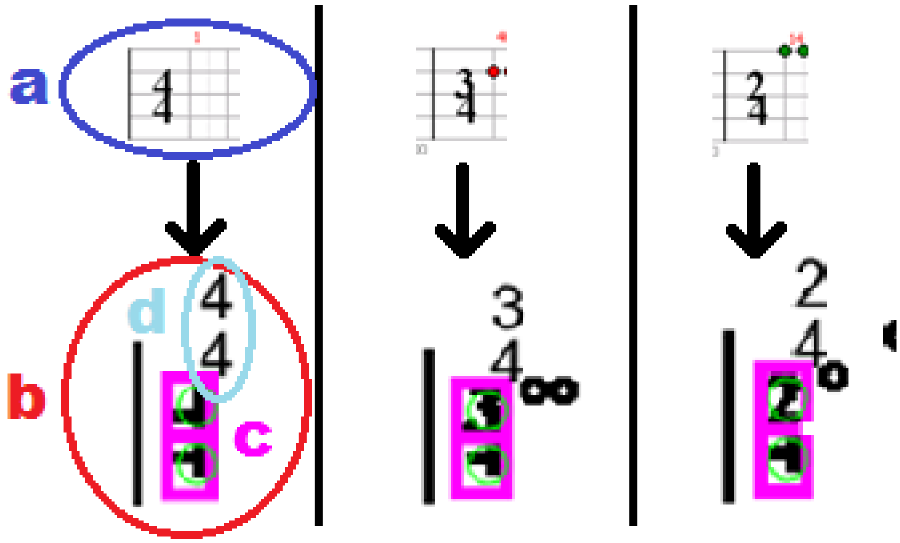


Figure 25: Detection of signatures in the Guitar Hero sheet music. The images in the first column are labeled such that 'a' shows the input image and 'b' shows the output image. Within 'b,' the magenta squares labeled 'c' show the blobs detected as possible signature values with their detected numerical value labels shown in 'd.'

value of 141 is displayed. This means that starting at the 7th measure, the music is played at 141 beats per minute, meaning that the music is slowed slightly such that each quarter note now lasts approximately 0.426 seconds. This new speed continues until another tempo is displayed.

Figure 26 depicts a flowchart of the pseudocode that summarizes the software that was created to detect these musical tempos. In this process, we first created an image that contained only the black and dark grey pixels from the original image, since the tempo values are always black, but some of the pixels in the letters can also be captured in this image due to their shade of grey. The resulting image contains several black and dark grey objects including measure bars, some bars that divide the measures, signature numbers, the outlines of the notes, and the small numbers beneath each measure bar. Thus, more processing must be done on this image to isolate the tempo values. First, a blob detection found all regions with contiguous regions in the image. These blobs were then merged with nearby neighbor blobs in

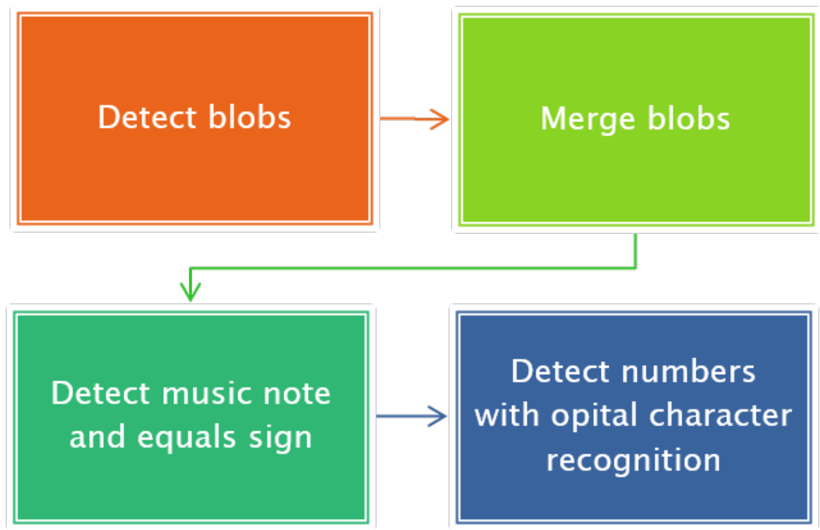


Figure 26: Flow chart showing the processed used to detection of tempos in the Guitar Hero sheet music.

order to allow numbers and symbols that were next to one another to be grouped. Next, optical character recognition was used on the blob groups. Groups that began with a music note symbol and an equals sign were determined to be tempos and all other blob groups were discarded. Finally, optical character recognition was used to detect the numerical values that followed the musical note and equals sign, and the resulting number is determined to be the tempo value. Figure 26 shows examples of detected tempos.

6.2 Translating Spatial Information to Temporal Information

Once all of the timing signatures, tempos, and musical notes are detected, using the process described in section 6.1 of this thesis, their positions and values are recorded. From this information, we can calculate the time value that each pixel in the image represents. This section of the thesis will describe the process of calculating these time values.

The timing signatures are used to detect how many notes should be occurring within each measure. The tempo values are used to detect the speed of each note.

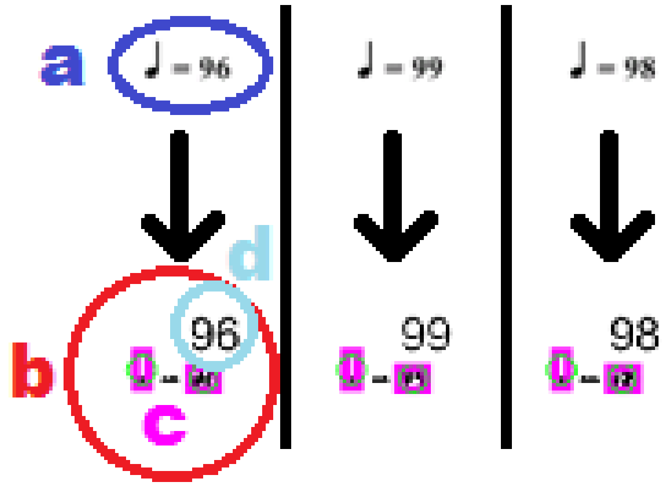


Figure 27: Detection of tempos in the Guitar Hero sheet music. The images in the first column are labeled such that 'a' shows the input image and 'b' shows the output image. Within 'b,' the magenta squares labeled 'c' show the blobs detected as possible tempo values with their detected numerical tempo value labels shown in 'd.'

From these two values, we can determine how to divide the space within a single measure. Two formulas are used to calculate the time at which each music note is played. First, we calculate the timing value of each measure, as seen in Equation 19. In this formula, t represents time (in seconds), and m, c, S, and T represent measure, current, Signature, and Tempo, respectively. From this formula, we can see that the time (in seconds) of the current measure is calculated by adding the time (in seconds) of the previous measure to the 60 times the Signature value divided by the Tempo value. The Signature value has the units of beats per measure and the Tempo has a units of beats per minute. In this formula, we divide the Tempo by 60 to change the units from beats per minute to beats per second. The unit of beats in the Signature variable cancels with the units of beats in the Tempo variable, leaving us with units of seconds per measure. Since we are calculating the timing of a single measure, we multiply this value by 1 measure to leave us with only units of seconds.

$$t(m_c) = t(m_{c-1}) + \frac{60 \times S}{T} \tag{19}$$

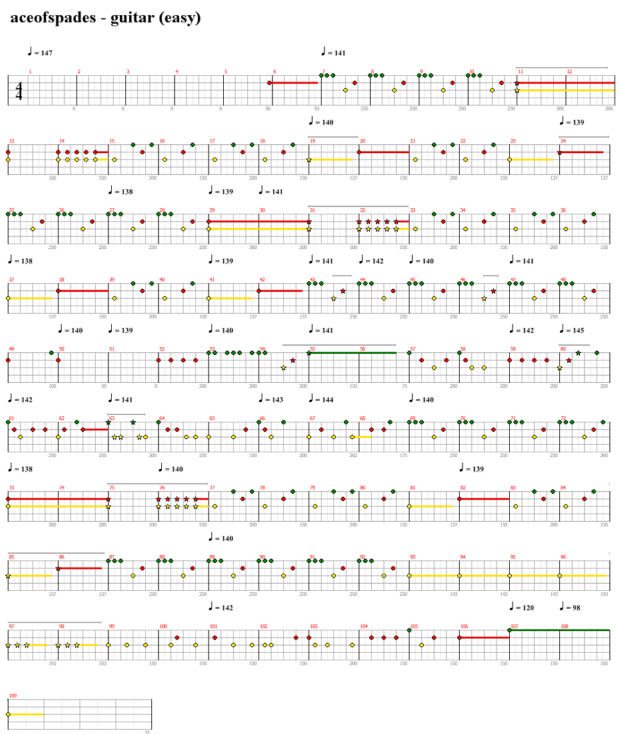
To calculate the time (in seconds) of each note that occurs in the musical pattern, we use the formula shown in Equation 20. In this equation, t represents time (in seconds) and n, m, c, x, T, and S represent note, measure, current, x-distance, Tempo, and Signature, respectively. From this formula, we calculate the time difference from the start of our current measure and the position of the note of interest and then add that value to the start time of our current measure to give us the timing value of the musical note. This time difference is calculated by dividing the x-position difference of the music note and the start of the measure by the total amount of time (in seconds) in the current measure. The total amount of time in the current measure is calculated by multiplying the number of beats per second (represented by T/60) by the x-distance in the measure dividing by the timing Signature.

$$t(n) = t(m_c) + \frac{x(n) - x(m_c)}{\frac{T}{60} \times \frac{x(m_c) - x(m_{c+1})}{S}} \tag{20}$$

For example, when we process the top line in Figure 19, we detect 8 measures, 1 signature, 2 tempos, and several musical notes in measures 6-8. First, we begin within measure 1 and look at the features in there. Spatially, the first feature that is detected is the signature, which is determined to be 4 over 4. We initialize our algorithm with this value. As we move to the right in the image, we detect our first tempo value, which is determined to be 147 beats per minute. From this information, we know that there are 4 beats per measure and that each beat is equates to $\frac{60\frac{seconds}{minute}}{147\frac{beats}{minute}}$ or 0.4082 seconds. Thus, each measure represents $4 \times 0.4082 = 1.6327$ seconds. Since no musical notes occur in the first measure, we know that the song starts with 1.6327 seconds of silence. Since there are no notes, tempo changes, or signature changes in measures 2-5, we add 4 more measures of silence. Thus, the song starts with $5 \times 4 \times 0.4082 = 8.1633$ seconds of silence. In measure 6, we detect a yellow note that is held for the duration of the song. That musical note is begins at the time associated to tbe beginning on measure 5, 8.1633 seconds. The musical note ends at

$8.1633 + 4 \times 0.4082 = 9.7960$ seconds. At the beginning of measure 6, we detect a new tempo change. Now, the music is operating at 141 beats per second, meaning that each beat in the measure is $\frac{60 \frac{seconds}{minute}}{141 \frac{beats}{minute}}$ or 0.4255 seconds and that each measure is $4 \times 0.4255 = 1.7021$ seconds. We detect a series of green notes. The first one occurs at the beginning of measure 6, which corresponds to 9.7960 seconds. The second note occurs at $\frac{1}{8}$ of the measure, the next at $\frac{3}{16}$, $\frac{1}{4}$, $\frac{5}{16}$, and $\frac{3}{8}$. Thus, we add the green notes to the pattern at times $9.7960 + \frac{1}{8} \times 1.7021$, $9.7960 + \frac{3}{16} \times 1.7021$, $9.7960 + \frac{1}{4} \times 1.7021$, $9.7960 + \frac{5}{16} \times 1.7021$, and $9.7960 + \frac{3}{8} \times 1.7021$, or times of 10.0088, 10.1151, 10.2215, 10.3279, and 10.4343 seconds, respectively. We continue this process of calculating the time value of each note by detecting where it occurs in the measure and calculating what time value is associated to that position. Through this process, we detect and add the following notes to our pattern: an orange note at $9.7960 + \frac{1}{2} \times 1.7021 = 10.6471$, a red note at $9.7960 + \frac{5}{8} \times 1.7021 = 10.8598$, a blue note at $9.7960 + \frac{11}{16} \times 1.7021 = 10.9661$, a red note at $9.7960 + \frac{13}{16} \times 1.7021 = 11.1790$, a yellow note at $9.7960 + \frac{7}{8} \times 1.7021 = 11.2853$. Next, we detect the beginning on measure 8, which occurs at $9.7960 + 1.7021 = 11.4981$ Measure 8 is a repeat of the pattern in measure 7 and we are able to detect 6 green notes at time values 11.4981, 11.7109, 11.8172, 11.9236, 12.0300, and 12.1364, an orange note at 12.3492, a red note at 12.5619, a blue note at 12.6683, a red note at 12.8810, and a yellow note at 12.9874. This process can be continued for all of the lines of music within a song and our output is a text file that contains the type of feature at the time at which that feature occurs. Figure 28 shows an example of an input sheet music image and a portion of the output text file.

From these calculations, we are able to create a timing representation of all of the notes that occur in each song that will allow us to mathematically process each of these patterns. Features from these songs were extracted and fed into a machine learning algorithm to classify the difficulty of each pattern. The results of the classifications




```
New Measure: #1, Start Time: 0.000000
New Signature: 4/4
New Tempo: 147
New Measure: #2, Start Time: 1.632653
New Measure: #3, Start Time: 3.265306
New Measure: #4, Start Time: 4.897959
New Measure: #5, Start Time: 6.530612
Time: 8.061144, Note: RED, P/R: 1
New Measure: #6, Start Time: 8.163265
Time: 9.681243, Note: RED, P/R: 0
Time: 9.794737, Note: GREEN, P/R: 1
New Measure: #7, Start Time: 9.795918
New Tempo: 141
Time: 9.988738, Note: GREEN, P/R: 1
Time: 10.204353, Note: GREEN, P/R: 1
Time: 10.635584, Note: YELLOW, P/R: 1
Time: 10.959007, Note: RED, P/R: 1
New Measure: #8, Start Time: 11.498046
Time: 11.498046, Note: GREEN, P/R: 1
Time: 11.691959, Note: GREEN, P/R: 1
Time: 11.907419, Note: GREEN, P/R: 1
Time: 12.338337, Note: YELLOW, P/R: 1
Time: 12.661526, Note: RED, P/R: 1
New Measure: #9, Start Time: 13.200174
Time: 13.200174, Note: GREEN, P/R: 1
Time: 13.394087, Note: GREEN, P/R: 1
Time: 13.609546, Note: GREEN, P/R: 1
Time: 14.040465, Note: YELLOW, P/R: 1
Time: 14.363653, Note: RED, P/R: 1
New Measure: #10, Start Time: 14.902301
Time: 14.902301, Note: GREEN, P/R: 1
Time: 15.096215, Note: GREEN, P/R: 1
Time: 15.311674, Note: GREEN, P/R: 1
Time: 15.742592, Note: YELLOW, P/R: 1
Time: 16.065781, Note: RED, P/R: 1
New Measure: #11, Start Time: 16.604429
Time: 16.604429, Note: RED, P/R: 1
Time: 16.604429, Note: YELLOW, P/R: 1
New Measure: #12, Start Time: 18.306557
```

Figure 28: An input sheet music image and a portion of the output text file showing the detected features and the time value associated to their occurrence.

of these patterns will be discussed in the next chapter.

CHAPTER VII

DIFFICULTY CLASSIFICATIONS OF MOVEMENT PATTERNS

One of the greatest benefits of practicing therapeutic exercises with a clinician is that the clinician is able to give their clients real-time feedback. The clinician is able to access the client's engagement as well as performance and adjust the task in order to increase engagement during the session as well as overall performance. However, when clients practice their therapeutic motions at home, they do not have any means of receiving feedback. Thus, clients regularly struggle with maintaining high levels of engagement and performance when practicing in isolation during at home therapeutic exercise sessions. Technologies exist to aid clients with the at home portion of their therapy; however, the technologies that are currently on the market do not possess the capability to automatically give real-time feedback to the users.

Thus, a goal of this research is to develop a scheme for classifying the difficulty of game patterns that relies on mathematical principals to replace the commonly used, tradition, and subjective scheme that relies on the"gut feelings" of individuals. To implement such a method, we extract features of each of the Guitar Hero game patterns. Next, a Linear Discriminant Analysis (LDA) classification method was used to define cluster regions for each difficulty label.

From the difficulty classifications, we will be able to quantitatively adjust the difficulty of our game based on the user's performance in order to promote the higher levels of user engagement, and thus, expedite the learning process in a similar way to how a clinician promotes faster task learning in their clients.

7.1 *Features Extracted from Game Patterns*

7.1.1 Time Difference Between Notes

The time difference between notes, in seconds, was calculated between each musical note in the pattern. The values were sorted from smallest distance to largest distance, the distribution of these values was plotted, and the area under the curve calculated, as shown in Equation 21. Next the values of the time difference were calculated in 0.1 second bins, as shown in Equation 22. The percentage of instances in each bin is calculated by dividing the value by the total amount, as seen in Equation 23. The percentage values for each of the 11 bin values and the area under the curve are 12 features that were used to train the LDA classification algorithm.

$$Area = \int_0^N sorted\left(d_x\left(\frac{x}{N}\right)\right)dx \tag{21}$$

$$d_{timetotals}(n) = \begin{cases} d_{time0}++ & \text{if } d < 0.1sec \\ d_{time1}++ & \text{if } d \geq 0.1sec \quad \& \quad d < 0.2sec \\ d_{time2}++ & \text{if } d \geq 0.2sec \quad \& \quad d < 0.3sec \\ d_{time3}++ & \text{if } d \geq 0.3sec \quad \& \quad d < 0.4sec \\ d_{time4}++ & \text{if } d \geq 0.4sec \quad \& \quad d < 0.5sec \\ d_{time5}++ & \text{if } d \geq 0.5sec \quad \& \quad d < 0.6sec \\ d_{time6}++ & \text{if } d \geq 0.6sec \quad \& \quad d < 0.7sec \\ d_{time7}++ & \text{if } d \geq 0.7sec \quad \& \quad d < 0.8sec \\ d_{time8}++ & \text{if } d \geq 0.8sec \quad \& \quad d < 0.9sec \\ d_{time9}++ & \text{if } d \geq 0.9sec \quad \& \quad d < 1sec \\ d_{time10}++ & \text{if } d \geq 1sec \end{cases} \tag{22}$$

$$d_{perc}(n) = \frac{d_{time(n)}}{\sum_{i=0}^{10} d_{time(i)}} \tag{23}$$

7.1.2 Note Position Difference

The position difference, in number of lines, was calculated between each musical note in the pattern. For example, if a green note appears on the top row of the measure followed by a red note on the second row, the position difference between these notes is 1. If there were multiple notes in a single instance, the position difference between all notes in these 2 instances was calculated and the sum was reported as the position difference. For example, if a green (top row) and yellow (third row) notes occurred simultaneously followed by a red (second row) and blue (fourth row), the position difference between the green and red would be calculated to be 1. Then, the position difference between the yellow and blue would be calculated to be 1. These values would be added together and the sum position difference for this instance would be 2. The percentage of instances with position change values of 0, 1, 2, 3, 4, 5, and >5 were calculated, as shown in formulas Equations 24 and 25. The mean and standard deviation of the distribution of position differences between note instances was also calculated, as shown in Equations 26 and 27, respectively. Also, the values of the position differences were sorted from smallest to largest, their distribution was plotted, and the area under the curve was calculated, as shown in Equation 21. The area under the curve, mean, standard deviation, and 6 values of the percentage of instances in each bin were used as additional features in the LDA classifier.

$$d_{positiontotals}(n) = \begin{cases} d_{position0} ++ & \text{if } change == 0 \\ d_{position1} ++ & \text{if } change == 1 \\ d_{position2} ++ & \text{if } change == 2 \\ d_{position3} ++ & \text{if } change == 3 \\ d_{position4} ++ & \text{if } change == 4 \\ d_{position5} ++ & \text{if } change == 5 \\ d_{position6} ++ & \text{if } change < 5 \end{cases} \tag{24}$$

$$d_{perc}(n) = \frac{d_{position(n)}}{\sum_{i=0}^{6} d_{position(i)}} \tag{25}$$

$$\overline{x} = \frac{\sum_{n=0}^{N} x_n}{N} \tag{26}$$

$$s = \sqrt{\frac{\sum_{n=0}^{N} (x_n - \overline{x})^2}{N-1}} \tag{27}$$

7.1.3 Number of Buttons Pressed for Each Instance

The number of notes that occurred in each time instance was calculated. For example, an instance that has a single blue note would have a value of 1 while an instance that has a green and blue note occurring simultaneously would have a value of 2. The percentage of instances with a single note and the percentage of instances with greater than one note were calculated. The area under the curve of the distribution of button presses per instance was also calculated, using Equation 21. When multiple button presses occurred at the same instance, the distance between these musical notes was calculated. For example, if a green (first line) note and a blue (fourth line) note occurred in the same instance, the distance reported would be 3. The mean and standard deviation for these distances were calculated, using Equations 26 and 27, respectively. The percentage button presses involving a single button, percentage of button presses involving multiple buttons, mean of the distance between buttons in a single instance, and standard deviation of the distance between buttons in a single instance were used as features in the LDA classifier.

7.1.4 Percentage of Notes

In the game pattern, there are 5 possible y-positions for the musical notes that correspond to 5 possible buttons that can be pressed on the game controller. These buttons can be pressed individually or multiple at a time. Thus, there are large numbers of possible combinations of time dependent button press sequences. These 5 possible y-positions correspond to specific colors and rows in the measure. The green buttons occur on the first row, red on the second, yellow on the third, blue on the fourth, and orange on the fifth. The number of musical notes in each position was calculated and divided by the total number of notes, resulting in percentage values for each note position, as shown in Equations 28 and 29. Also, the area under the curve for the distribution of notes of each color was calculated, as shown in Equation 21. The value of the area under the curve and the 5 percentage values were added as features in the LDA classification algorithm.

$$color_{totals}(n) = \begin{cases} c_0 ++ & \text{if } color == green \\ c_1 ++ & \text{if } color == red \\ c_2 ++ & \text{if } color == yellow \\ c_3 ++ & \text{if } color == blue \\ c_4 ++ & \text{if } color == orange \end{cases} \tag{28}$$

$$c_{perc}(n) = \frac{c_n}{\sum_{i=0}^{5} c_i} \tag{29}$$

7.1.5 Number of Notes in a Second

Each pattern was divided into 5 second intervals. For each 5 second interval, the number of notes occurring in that time interval was calculated. These values were sorted and the area under the curve, mean, and standard deviation were calculated, using Equations 21, 26, and 27, respectively. The fastest speed in a 5 second interval was also calculated. The values were divided into bins and the percentage of values within each bin was calculated, as shown in Equations 30 and 31. The area under the curve, mean, standard deviation, fastest speed, and percentage of instances in each bin were added as additional features in the LDA classification algorithm.

$$speed_{totals}(n) = \begin{cases} s_0 ++ & \text{if } speed < 3bps \\ s_1 ++ & \text{if } speed \geq 3bps \quad \& \quad speed < 6bps \\ s_2 ++ & \text{if } speed \geq 6bps \quad \& \quad speed < 9bps \\ s_3 ++ & \text{if } speed \geq 9bps \quad \& \quad speed < 12bps \\ s_4 ++ & \text{if } speed \geq 12bps \quad \& \quad speed < 15bps \\ s_5 ++ & \text{if } speed \geq 15bps \quad \& \quad speed < 18bps \\ s_6 ++ & \text{if } speed \geq 18bps \quad \& \quad speed < 21bps \\ s_7 ++ & \text{if } speed \geq 21bps \quad \& \quad speed < 24bps \\ s_8 ++ & \text{if } speed \geq 24bps \quad \& \quad speed < 27bps \\ s_9 ++ & \text{if } speed \geq 27bps \end{cases} \tag{30}$$

$$speed_{perc}(n) = \frac{s_n}{\sum_{i=0}^{9} s_i} \tag{31}$$

7.1.6 Lengths of Held Notes

Each held note in the pattern was detected and the amount of time the note was held was calculated. These time values were sorted and the area under the curve, mean, and standard deviation were calculated, using Equations 21, 26, and 27, respectively. The longest hold was also calculated. The hold values were divided into bins and the percentage of values within each bin was calculated, as shown in Equations 32 and 33. The area under the curve, mean, standard deviation, longest hold, and percentage of instances in each bin were also added as additional features in the LDA classification algorithm.

$$holdtime_{totals}(n) = \begin{cases} h_0 + + & \text{if } hold < 1sec \\ h_1 + + & \text{if } hold \geq 1sec \quad \& \quad hold < 3sec \\ h_2 + + & \text{if } hold \geq 3sec \end{cases} \tag{32}$$

$$hold_{perc}(n) = \frac{h_n}{\sum_{i=0}^{2} h_i} \tag{33}$$

7.2 Classification Accuracies

The features for each pattern were input into the LDA classification algorithm in 2 different passes with 2 different sets of labels. First, the inputs were labeled with "easy," "medium," and "hard." The ground truth for these labels was given by the Guitar Hero video game. All patterns from the game defined "easy" difficulty were given a label of "easy," all game defined "medium" were given a difficulty of "medium," and all game defined "hard" and "expert" difficulties were combined into a super-class and given the "hard" label. Patterns from the Guitar Hero games, Guitar Hero 1, Guitar Hero 2, Guitar Hero 3, and Guitar Hero 80's were used only. This subset of games were chosen because they had user voted sub-labels for each pattern in addition to game defined "easy," "medium," and "hard," labels. In order to increase the number of patterns in our training set, the pattern from each song was split in half and it was assumed that both halves of the songs would have the same difficulty that matched the given labels. The LDA was used to classify this data with 10-fold cross validation

Table 14: Confusion Matrix for Easy, Medium, and Hard Labels

	True Easy	True Medium	True Hard
Classified Easy	96.98%	1.53%	1.49%
Classified Medium	4.41%	94.35%	1.24%
Classified Hard	0.00%	4.55%	95.45%

and the results are displayed in the confusion matrix shown in Table 14. The results of this algorithm show that we can successfully classify all labels with 94% accuracy or better.

Next, the data was passed through the LDA separately to classify 3 subcategories of difficulty for each group. These subcategories were defined by user-ranked song difficulties created from user voting on Guitar Hero fan forums. In the forums, fans voted to rank songs in order from easiest to hardest. Since this was a continuous shift in difficulty with no defining lines between adjacent patterns and uncertain difficulty, the ranked patterns were divided into 5 sections. We classified the 1st, 3rd, and 5th sections, throwing out the 2nd and 4th sections. The easiest, middle, and hardest songs were correctly classified with 62.79%, 40.30%, and 64.07% accuracy, respectively. However, the LDA was capable of classifying the 1st and 5th sections without the 3rd section much more successfully. In the "easy" category, the easiest and hardest songs were correctly classified with 80.25% and 73.18% accuracy, respectively. In the "medium" category and the "hard" category, the easiest and hardest songs were correctly classified with 81.78%, 82.83%, 80.48%, and 82.25% accuracy, respectively. When the data was combined and the LDA classified the easiest and hardest songs for all pattern difficulties simultaneously, it correctly classified the easiest and hardest songs with 77.65% and 81.50% accuracy, respectively.

7.3 Labeling 30 Second Patterns

Finally, the song patterns were divided into 30-second time intervals. The features were extracted from each of these patterns and they were run through both passes of the LDA to classify their difficulty and a sub-category. The resulting output allows for us to have 9 different difficulty values associated to 30-second patterns. We selected 5 30-second patterns at each label to use in our final system. In order to reduce error, we only selected 30-second patterns from full patterns that were correctly labeled by the LDA. These patterns were placed in a look-up table in our game. As the game adapts and adjusts difficulty, it determines the desired difficulty for the participant to play. Then, it looks up a 30-second pattern that has been defined to be the correct difficulty and displays it to the user.

CHAPTER VIII

ADAPTIVE GAMING EXPERIENCE

One goal of this project is to create an adaptive video game experience that promotes accelerated task learning. To meet this goal, we used an LDA machine learning algorithm to classify 30-second long game patterns into nine unique difficulty levels. The video game assesses user performance every 30 seconds. If users perform with 90% accuracy or better, they are considered to have mastered that difficulty and the game will increase the difficulty level. If users perform with a 70% accuracy or worse, users are considered to be too greatly challenged and the game decreases difficulty.

8.1 Experimental Setup

We are conducted a final experiment with elderly adults to verify our hypothesis. We recruited 19 elderly adults to participate in this experiment. The age range of our participants was 52 to 86 with a mean of 77.4 and a standard deviation of 9.97. Two of our test participants were stroke survivors and three had previously participated in physical therapy for non-stroke related impairments. In this experiment, users were asked to complete three sessions of game play on three different days. Due to scheduling conflicts, not all participants were able to complete all three test sessions. We had six participants complete all three sessions, four complete two sessions, and nine complete one session. At the beginning and end of each session, participants completed a pre- and post-test respectively. The pre- and post-tests were the same patterns and were comprised of a 30-second middle easy difficulty, 30-second middle medium difficulty, and a 30-second middle hard difficulty. Between the pre- and post-tests, participants were asked to play the game for six 30-second intervals. Participants were divided into three test groups. In the control group, participants were

Table 15: Accuracy Score Improvements Between Pre- and Post-Tests

	Group A	Group B	Group C
Easy	5.85%	8.72%	13.58%
Medium	0.18%	4.78%	8.27%
Hard	-1.40%	6.23%	2.93%

asked to complete all six of their 30-second intervals with the middle medium difficulty. In experimental group A, participants were asked to complete the first two 30-second intervals at the middle medium difficulty and then the game will adapt for the following four sessions. In experimental group B, participants were asked to complete the first two 30-second intervals at the difficulty corresponding to the highest difficulty that they performed 70% - 90% accuracy during the pre-test. The following four 30-second intervals were adaptive for experimental group B.

8.2 Results

8.2.1 Accuracy

The accuracy score improvements for the easy, medium, and hard sections of the pre- and post-tests for each test group for each session were compared. Table 15 shows the results of the learning improvements for this experiment. In the easy, medium, and hard categories, a trend is seen that participants in the adaptive groups (B and C) had greater amounts of learning during each session when compared to the non-adaptive group (A).

8.2.2 Lead/Lag

The path that the game targets encouraged was compared to the path that was actually followed by each participant. From this plot, a lead/lag score can be calculated by subtracting the difference in the x-value of the actual path and the intended path, as seen in Equations 4 and 5. As defined in Section 3.5.2.2 of this thesis, lead refers

Table 16: Improvements Between Pre- and Post-Tests

Parameter	Control Group	Adaptive Groups
Lead/Lag	-26.79s	+15.48s
Jerkiness	-3.37%	-6.72%
Over/Undershoot	-0.39pixel	+0.22pixel

to the amount of time that a user intends to hit a note prior to the time that the theoretical path anticipated that user would hit the note. Lag is defined as the amount of time after the theoretical hit time that the user hit the note.

As seen in Table 16, when we compared the lead/lag score improvements gained between the pretest and posttest, we found that participants in the adaptive groups (groups B and C) had an average gain of 15.48 more seconds spent leading, while the control group (group A) had an average loss of 26.79 more seconds lagging. These results suggest that participants in the adaptive groups were anticipating the notes and responding more quickly during the posttest than they had done during the pretest, while the participants in the control group were being more reactive. Thus, the participants in the adaptive games reduced their reaction times and learned to anticipate movements, while the control group experienced some losses in these categories.

8.2.3 Jerkiness

The movement paths completed by the users were also analyzed for jerkiness. Jerkiness is calculated taking the derivative of the participants' movement paths and then finding the peak values between target notes , as shown in Equation 7. As described in section 3.5.2.4 of this thesis, jerkiness is defined as the peak speed that a participant moves towards each target. Having a higher jerkiness suggests that a participant is reacting and having to move more quickly in order to reach the target in time, while a lower jerkiness value suggests that the participant in anticipating the next target and moving at a consistent, controlled, and appropriate speed to meet said target.

As shown in Table 16, the average jerkiness decreased between the pretest and posttest for all groups. This suggests that the users were moving more consistently and in more controlled manners during the posttest when compared to the pretest. However, participants in the adaptive groups experienced higher gains than those in the control with jerkiness reductions of 6.72% and 3.37%, respectively.

8.2.4 Overshoot/Undershoot

The overshoot/undershoot score was calculated for each musical note target that each user was presented with. This score is calculated by determining the difference in y-values in the actual path and the intended path, as seen in Equation 6. In section 3.5.2.3, we defined overshoot to be the distance a user continues past the target, while undershoot is defined as the distance that a user stops prior to reaching the target. A participant who is completing the task better would have smaller absolute values of overshoot and undershoot, while the direction (overshoot vs undershoot) is not a relevant performance metric. Thus, we took the absolute value of these overshoot and undershoot scores to assess our users' performances.

Table 16 shows that, in the control group, users had larger overshoot/undershoot absolute value scores in the posttest, when compared to the pretest. They were less accurate to the y-position by an average of 0.39 pixels. Meanwhile, the adaptive group participants made an average gain of 0.22 pixels more accurate in the y-position during their posttest, when compared to the pretest. This suggests that users in the adaptive groups learned to hit the y-positions more accurately than those in the control group.

8.3 Participant Survey

Upon completing the experiment, participants were asked to answer survey questions about their opinions of the gaming experience. The specific questions asked to the participants are shown in Table 17 and the responses are shown in Figure 29. While

Table 17: Survey Questions Presented to Participants

Number	Question
1	I enjoyed the exercise session WITH the tablet game.
3	I felt that my exercise session WITH the tablet game was productive.
4	I felt engaged while exercising WITH the tablet game.
5	I felt bored while exercising WITH the tablet game.
6	I motivated while exercising WITH the tablet game.
7	I frustrated while exercising WITH the tablet game.
8	I MORE motivated while exercising WITH the tablet game that I would if I had to exercise WITHOUT the tablet game.
9	If I had to exercise my wrist for one hour a day, every day, I would rather complete all of these exercises WITH the tablet game.

the responses were not different between the adaptive and non-adaptive test groups in this experiment, the results are higher than responses from a previous, non-adaptive gaming experience that was tested with participants aged 14-35. The results of their surveys can be seen in Figure 4. This suggests that elderly participants may enjoy this rehabilitation experience greater than younger participants. During the testing sessions, many of the participants were laughing, expressing how much better they felt this was than other, traditional treatments that they may have received in the past, and reminiscing about how it reminded them of arcade games like Pacman and Donkey Kong that they used to play in their younger years. The younger group may have been biased against enjoying the experience as much because they have more experience with higher complex games and therefore may not enjoy simple games as much. The younger population may also not fully understand the tediousness of doing traditional exercise sessions as alternative to playing a rehabilitation game.

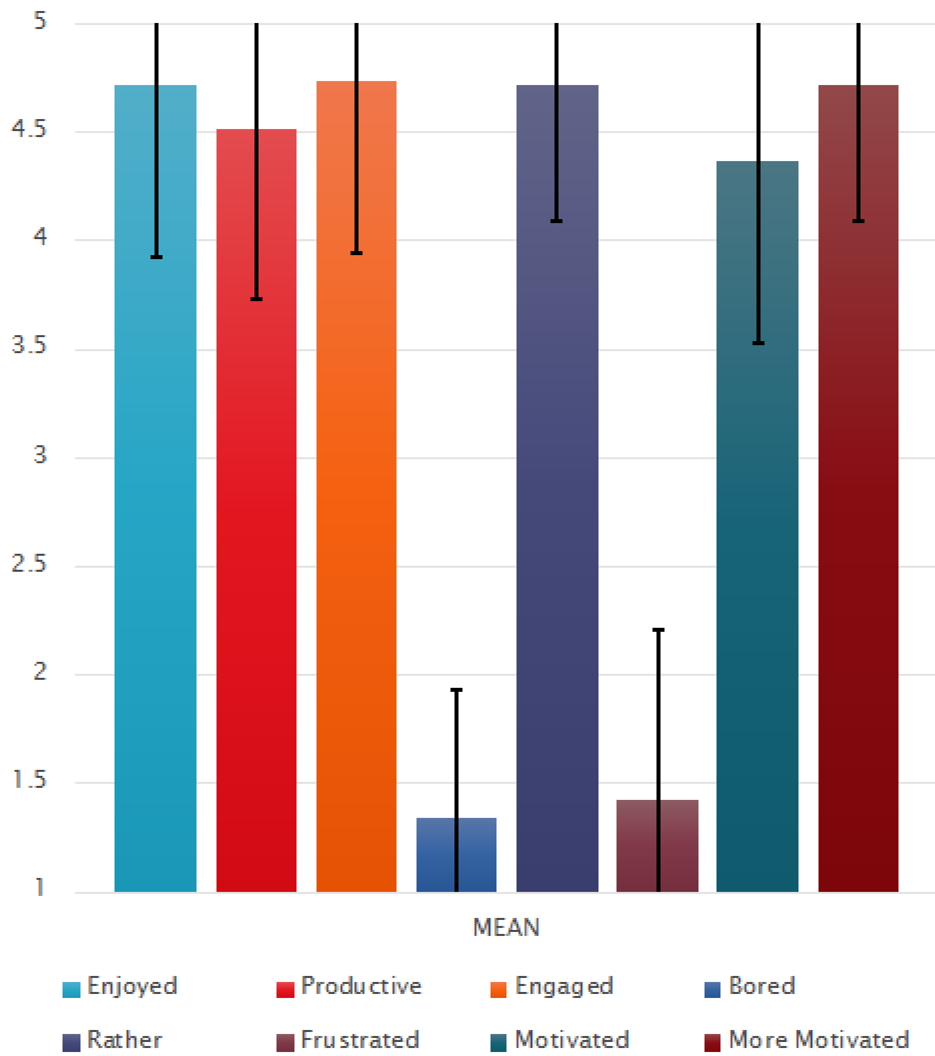


Figure 29: Survey results from our final experiment.

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

In this research project, we were able to develop a passive physical therapy system for assisting stroke survivors with at-home therapy sessions. We were able to show that this system could encourage users to follow specific motions. A goal of the project was to allow for the game to provide feedback in the form of adjusting the difficulty to best fit the needs of the user, and thus promote an optimal rate of learning. We were able to develop a methodology for classifying task difficulty, using machine learning techniques and a database collected from video game data from a similar, commercially available, popular video game whose sole purpose was for entertainment. Once we were able to classify the task difficulty, we were able to use this information to add adaptations to our rehabilitation game. We conducted a final validation experiment of the system with elderly adults and stroke survivors, and the results showed that the algorithm effectively promoted increased learning during game sessions. In addition, the elderly adults reported that they highly enjoyed using the system even more than younger adults had in previous experiments.

This results of this thesis have all been obtained from short term studies. Thus, future works for this project that would greatly increase the impact of the results should involve conducting a long term study with more participants. Such a study would allow us to validate that the learning trends that were seen in this thesis would continue over several months, which is the length of time that stroke survivors traditionally participate in physical therapy.

In addition, the process laid out in this thesis for classifying task difficulty could be translated to other spaces. For the future work of this project, we could collect large video game datasets and analyze them to determine difficulty of other types of tasks including movement or cognitive tasks. It would be interesting to see if these methods could be used on puzzle games, for example, to determine what types of puzzles are more difficult than others. Such a task difficulty classification scheme could be useful for creating better physical therapy games, as well as to generate appropriately challenging educational games for children, given their skill level.

REFERENCES

[1] Activision, "Guitar hero."

[2] Aluru, V. *et al.*, "Effect of auditory constraints on motor performance depends on stage of recovery post-stroke," *Frontiers In Neurology*, vol. 5, pp. 1–13, 2014.

[3] Basteris, A. *et al.*, "Lag-lead based assessment and adaptation of exercise speed for stroke survivors," *Robotics And Autonomous Systems*, vol. 73, pp. 144–154, 2004.

[4] Brutsch, K. *et al.*, "Influence of virtual reality soccer game on walking performance in robotic assisted gait training for children," *Journal of NeuroEngineering and Rehabilitation*, vol. 7, p. 15, 2010.

[5] Butler, A. J. *et al.*, "Active learning of novel sound-producing objects: Motor reactivation and enhancement of visuo-motor connectivity," *Journal Of Cognitive Neuroscience*, vol. 25, no. 2, pp. 203–218, 2013.

[6] Cameirao, M. S. *et al.*, "Using a multi-task adaptive vr system for upper limb rehabilitation in the acute phase of stroke," *Proceedings of the International Conference on Virtual Rehabilitation (ICVR08)*, pp. 2–7, 2008.

[7] Cioi, D. *et al.*, "Ankle control and strength training for children with cerebral palsy using the rutgers ankle cp," *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–6, 2011.

[8] Clarke, P. *et al.*, "Quality of life following stroke: Negotiating disability, identity, and resources.," *Journal of Applied Gerontology*, vol. 24, no. 4, pp. 319–336, 2005.

[9] Colombo, R. *et al.*, "Design strategies to improve patient motivation during robot-aided rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 4, p. 3, 2007.

[10] Duff, M. *et al.*, "An adaptive mixed reality training system for stroke rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 5, pp. 531–541, 2010.

[11] Dunn, D. S. *et al.*, "Prospects for a positive psychology of rehabilitation.," *Rehabilitation Psychology*, vol. 50, no. 3, pp. 305 – 311, 2005.

[12] English, B. and Howard, A., "Encouraging specific intervention motions via a robotic system for rehabilitation of hand function: A healthy pilot study," *IEEE Symposium Series on Computation Intelligence*, 2014.

[13] English, B. and Howard, A., "Engagement study of an integrated rehabilitation robotic tablet-based gaming system," *IEEE Workshop on Advanced Robotics and its Social Impacts*, 2014.

[14] English, B. and Howard, A., "An adaptive robotic tablet gaming system for post-stroke hand function rehabilitation," *ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, 2015.

[15] English, B. and Howard, A., "The effects of adjusting task difficulty on learning motor and cognitive aspects of a multitasking task," *2017 IEEE Symposium Series on Computational Intelligence,*, 2017.

[16] English, B. and Howard, A., "The effects of auditory and visual cues on timing synchronicity for robotic rehabilitation," *IEEE International Conference on Rehabilitation Robotics*, 2017.

[17] English, B. and Howard, A., "The effects of musical cues on motor learning using a robotic wrist rehabilitation system," *IEEE Workshop on Advanced Robotics and its Social Impacts*, 2017.

[18] English, B., Hunter, J., Tuck, J., and Tzintzarov, G., "Designing a scalable robotic exoskeleton and tablet gaming suite for hand function rehabilitation," *Rehabilitation Engineering and Assistive Technology Society of North America Annual Conference*, 2015.

[19] Foley, N. *et al.*, "Inpatient rehabilitation following stroke: Amount of therapy received and associations with functional recovery," *Disability and Rehabilitation*, vol. 34, no. 25, pp. 2132–2138, 2012.

[20] Fugl-Meyer, A. *et al.*, "Does stretching induce lasting increases in joint ROM? a systematic review.," *Scandinavian Journal Of Rehabilitation Medicine*, vol. 7, no. 1, pp. 13 – 31, 1975.

[21] Go, A. S. *et al.*, "Heart disease and stroke statistics–2014 update: a report from the American Heart Association.," *Circulation*, vol. 129, no. 3, pp. 399 – 410, 2014.

[22] Guadagnoli, M. and Timothy, D., "Challenge point: A framework for conceptualizing the effects of various practice conditions in motor learning," *Journal Of Motor Behavior*, vol. 36, no. 2, pp. 212–224, 2004.

[23] Guadagnoli, M. A. *et al.*, "Challenge point: A framework for conceptualizing the effects of various practice conditions in motor learning," *Journal Of Motor Behavior*, vol. 26, no. 2, pp. 212–224, 2004.

[24] JJCirri, "aceofspades - guitar (expert)."

[25] Kasimi, A., Nichols, E., and Raphael, C., "A simple algorithm for automatic generation of polyphonic piano fingerings," *ISMIR 2007 Eighth Int. Conf. on Music Inf. Retr. Proc.*, 2007.

[26] Lenze, E. J. *et al.*, "The pittsburgh rehabilitation participation scale: reliability and validity of a clinician-rated measure of participation in acute rehabilitation," *Archives of Physical Medicine and Rehabilitation*, vol. 85, no. 3, pp. 380 – 384, 2004.

[27] Lin, C. and Liu, D., "An intelligent virtual piano tutor," *Proceedings of VRCIA 2006*, pp. 353–356, 2006.

[28] Lin, L., "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, pp. 255–268, 1989.

[29] Lin, L., "A note on the concordance correlation coefficient," *Biometrics*, vol. 56, pp. 324–325, 2000.

[30] Logan, G., "Toward an instance of theory of automatization," *Psychological Review*, vol. 95, pp. 492–527, 1988.

[31] Ma, M., Jain, L., and Anderson, P., *Virtual, Augmented Reality and Serious Games for Healthcare 1.* Springer, 2014.

[32] Ma, M. *et al.*, "Adaptive virtual reality games for rehabilitation of motor disorders," *Universal Access in Human-Computer Interaction. Ambient Interaction Lecture Notes in Computer Science*, vol. 4555, p. 681690, 2007.

[33] MadGamer, "Guitar hero 3 - all bosses 5 stars - expert guitar (60 fps)," 2012.

[34] McBride, G., "A proposal for strength-of-agreement criteria for lin's concordance correlation coefficient," *NIWA Client Report: HAM2005-062*, 2005.

[35] Newell, A. and Rosenbloom, P., "Mechanisms of skill acquisition and the law of practice," *Cognitive skills and their acquisition*, p. 155, 1981.

[36] Panarese, A. *et al.*, "Tracking motor improvement at the subtask level during robot-aided neuro rehabilitation of stroke patients," *Neuro Rehabilitation Neural Repair*, vol. 26, no. 7, pp. 822–833, 2012.

[37] Paolucci, S. *et al.*, "Impact of participation on rehabilitation results: a multivariate study," *European journal of physical and rehabilitation medicine*, vol. 48, no. 3, pp. 455 – 466, 2012.

[38] National Assessment of Educational Progress, "How is grade point average calculated?," 2011.

[39] Rietschel, J. and others, "Cerebral-cortical networking and activation increase as a function of cognitive-motor task difficulty," *Biological Psychology*, vol. 90, no. 2, pp. 127–133, 2012.

[40] Sebastien, V., Ralambondrainy, H., Sebastien, O., and Conruyt, N., "Score analyzer: Automatically determining scores difficulty level for instrumental e-learning," *13th International Society for Music Information Retrieval Conference*, 2012.

[41] Sluijs, E. *et al.*, "Correlates of exercise compliance in physical therapy.," *Physical Therapy*, vol. 73, no. 11, pp. 771–786, 1993.

[42] Valentine, G. *et al.*, "Hand mentor prototyping and interface guide, rev. 5.." November 2013.

[43] van Vugt, F. T. *et al.*, "Auditory feedback in error-based learning of motor regularity," *Brain Research*, vol. 1606, pp. 54–67, 2015.

[44] Vinken, P. M. *et al.*, "Auditory coding of human movement kinematics," *Multisensory Research*, vol. 26, no. 6, pp. 533–552, 2013.

[45] Y2kcrazyjoker4, "Guitar hero contoller horizontal," 2008.

[46] Zimmerli, L. *et al.*, "Validation of a mechanism to balance exercise difficulty in robot-assisted upper extremity rehabilitation after stroke," *Journal of Neuroengineering and Rehabilitation*, vol. 9, p. 6, 2012.

[47] Zimmerman, M. *et al.*, "Empowerment theory for rehabilitation research: Conceptual and methodological issues.," *Rehabilitation Psychology*, vol. 43, no. 1, pp. 3–16, 1998.