university of groningen / Faculty of Science and Engineering / Biomedical Engineering

Advancing knee rehabilitation through integration of virtual reality and haptic systems

María Fossati Estévez
S-5362970

Discrete Technology and Production Automation (DTPA)

Period: 05/02/2024 - 21/06/2024

Master's Project

1st Examiner: Dr. E. Wilhelm, Discrete Technology and Production Automation
2st Examiner: Dr. J.M. Hijmans, Faculty of Medical Sciences

July 4, 2024

Contents

Page

Abstract **5**

1 Introduction **6**
1.1 Knee arthoplasty 6
1.2 Virtual reality in rehabilitation 6
1.2.1 Knee rehabilitation games at RUG 7
1.3 Tactile stimulation for rehabilitation 8
1.4 Research goal 8
1.5 Haptic systems 9

2 Materials & Methods **11**
2.1 Study of different technologies as potential haptic feedback systems 11
2.1.1 Ultrahaptics 11
2.1.2 Vibration 13
2.1.3 Peltier elements 16
2.2 Design of Vibration Motor haptic feedback system 17
2.2.1 Requirements and wishes 17
2.2.2 Final system Vibration motors 17
2.2.3 Effect on muscle activation 21
2.3 Study of effect of vibration on knee extension task 22
2.3.1 Study design 22
2.3.2 Statistical Analysis 23

3 Results **25**
3.1 Ultrahaptic sensations on knee 25
3.2 Peltier components test results 25
3.3 EMG with and without vibration 26
3.3.1 Knee extensors 26
3.3.2 Muscles in viccinity of vibration device 27
3.3.3 During playing 28
3.4 RCT results 30
3.4.1 Game scores 30
3.4.2 Immersion Questionnaire 34

4 Discussion **35**
4.1 Ultrahaptics as haptic sensation 35
4.2 Peltier as haptic sensation 35
4.3 Vibration system as haptic sensation 36
4.4 Randomised control trial study 37

5 Conclusions **39**

Bibliography **41**

A Arduino Code for pwm control of 6 motors 46

B Arduino Code for 5 modes of constant ’on’ vibration intervals of 8 motors 47

C Arduino Code for 5 modes of constant ’off’ vibration intervals of 8 motors 49

D Arduino Code for 5 modes of constant ’on+off’ vibration intervals of 8 motors 51

E Arduino Code for 5 modes of constant vibration of 8 motors at different amplitudes 53

F Arduino Code for 5 modes of periodic vibration of 8 motors at different amplitudes 55

G Motor pinnout in final knee sleeve design 57

H Schematic of circuit for motor control with Arduino Nano BLE 58

I Final Motor Control Arduino Code, Slave/Peripheral device 59

J Final Motor Control Arduino Code, Master/Central device 63

K Results of Ultrahaptic tests 66

Acknowledgments

I would like to express my gratitude to my supervisor, Elisabeth, for giving me the opportunity to work on this project, and for her constant support, guidance, and trust until its completion. I also want to send a special thanks to the technicians of the lab. Especially Simon for his expertise in designing the electronics, and Hayden for his assistance with soldering. Your technical support was crucial to this project. I am also very grateful to the PhD students in the research group. In particular to Yiling, for allowing me to work on her game and always being available to answer my questions, and to Luis Felipe, for his willingness to help at multiple stages of the project. Lastly, I also want to thank the other students in the lab for creating a wonderful work environment. In particular, Catalina and Gigi, your support has been invaluable.
Thank you all for your contributions and encouragement.

Abstract

Rehabilitation is critical for restoring mobility and strength post-injury or surgery. However, the repetitive and tedious nature of traditional therapies often leads to poor adherence and sub-optimal outcomes. In particular, patients post-knee arthroplasty surgery do not consistently experience relief from chronic pain or restoration of proprioceptive sense. This project explores the integration of haptic feedback in a virtual reality (VR) rehabilitation game for knee extension training, aiming to increase engagement, enhance motor learning, and potentially aid in chronic pain treatment. Various haptic feedback mechanisms, including vibration motors, Peltier elements, and Ultrahaptics arrays, were evaluated. A knee sleeve equipped with vibration motors was developed and tested for its ability to provide continuous tactile stimulation. Preliminary EMG analysis indicated no adverse effects on muscle activation. A 3-day control study with healthy participants showed that the vibration motors offered a consistent, perceivable sensation around the knee. Initial data from five participants revealed significant improvement in both groups after training with the device, suggesting motor learning due to the game. Furthermore, a significant difference in the test group's performance over the control group suggests the potential of haptic feedback to improve rehabilitation outcomes. Continuation of the study with a larger sample size and further research in patient populations is recommended to confirm these findings and explore the broader application of this technology.

1 Introduction

1.1 Knee arthoplasty

Knee arthoplasty is a surgery that aims to resurface a knee damaged by arthritis, by replacing the joint with an implant [1]. It is a very common and successful procedure performed up to 2.5 million times a year in Europe [2] with 90-95% cases showing favourable outcomes within 10 years [3]. After such surgery, timely and effective rehabilitation is critical to ensure the patient retains full mobility and strength. It is however in question which rehabilitation program yields the most optimal results [4].

Even if the surgery is in general quite successful, there are some issues that do not get fully resolved with the surgery. For example, reduced proprioceptive sense of their knee [5] is a very common problem in arthritis patients. However, the knee replacement surgery aimed to treat the arthritis only results in slight improvement of proprioception, reaching values still lower than in healthy populations [6]. As the sense that lets us perceive the location, movement, and action of parts of the body [7], proprioception of the knee is vital for safe locomotion, and alterations on it cause functional problems. Even though physiotherapy is recommended to improve this issue, not one specific physiotherapy protocol specifically tackles it.

Moreover, ailments arise after the surgery in some patients. The most common example of this is chronic pain in the replaced joint. Defined as persistent pain after surgery that prevails 2-3 months after the procedure and has no other discernible cause [8], research suggests that up to 44% of patients suffer from it [9]. Even if very prevalent, the exact mechanisms behind this pain are still not fully understood. Some research suggests that the damage due to both the arthritis, and the surgery itself might result in changes in the way the nervous system processes pain and touch [10, 11]. Physiotherapy interventions that specifically tackle this issue are severely underesearched, with all current approaches showing similar and poor results at reducing the pain [12]. In any case, further research is needed to come up with effective ways of tackling, and potentially reducing this pain.

1.2 Virtual reality in rehabilitation

In the recent years, there has been a surge in virtual reality games being used in rehabilitation practices. These games require patients to perform specific motor exercises to fulfill the in-game tasks. The entertaining and goal-focused nature of the game make this alternative to traditional therapy more engaging and motivating to the patient. Another potential benefit of this rehabilitation method is the possibility of exercising from home, since the addition of sensors can ensure that the exercises are correctly performed in a safe and controlled manner. The therapist can even access the data from the home therapy sessions and analyse the improvement, so progress can be better assessed [13].

Research has shown that different modes of virtual reality therapy are indeed effective for reducing knee pain and improving knee function, dynamic balance, mobility and strength after knee arthoplasty surgeries. This method of rehabilitation also shows high satisfaction, adherence and motivation from the patients [14]. While promising, there is still no clear evidence of whether these games are more effective than traditional therapies. Therefore, this newer practices still need further research to fully understand and leverage their capabilities [15].

1.2.1 Knee rehabilitation games at RUG

A group of PhD students, working under Dr Wilhem at the DPTA department at RUG university are working on developing immersive virtual reality games specifically for aiding knee rehabilitation [13]. They have currently developed two games at two levels of increased difficulty, both aiming to re-train correct knee extension. The first game allows the patient to play while seated to work on knee extension without weight bearing; and the second works the loaded knee by performing squats. Both levels use Inertial Measurement Unit (IMU) sensors to measure the knee angle, which controls the functionalities of the game. Moreover, electromyography (EMG) sensors around the knee measure muscle activation to ensure that the exercise is being performed correctly. These measurements can also be be used for data analysis and allow for further research on knee rehabilitation itself.

One of the games, specifically the one focusing on knee extension, will be used in this project. It aims to improve proprioception and control of the knee extension by asking the participant to control the degree of bending of the knee in accordance to instructions shown on screen. The user interface of the exergame is shown in Figure 1. In this game, fishes appear randomly in 4 different trails which represent 4 knee bending degrees, 90°, 60°, 30°, 0º. The knee angle of the player controls the hook, which shows up on screen at the height related to the degree of knee bending of the user, measured in real time. Therefore, when the player's knee bending is within the range of a trail, the hook will show in the related trail. The target angle is indicated by the swimming fish, which appears at one of the 4 trails, and moves from one side of the screen to the other. The fish in the trails relating to 30º and 90º of knee bending appear at the right side of the skin and swim towards the left. Alternatively, the fish in the trails relating to 0º and 60º of knee bending swim from left to right. The trail that each fish appears at is randomised.

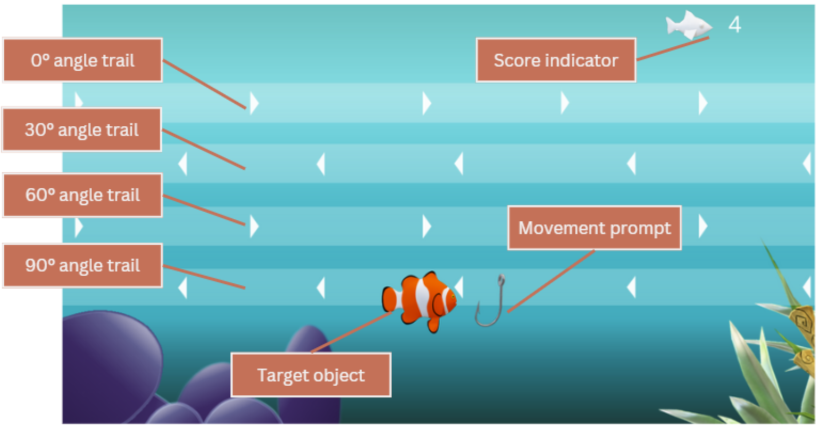


Figure 1: Interface of the OKC 'fishing' exergame

The goal of the player is to move the hook to the row in which the fish swims and hold the respective position for some time to be defined depending on desired game difficulty, then the fish will be "hooked". The exact range that is considered to be the correct position can be changed, to also increase or decrease the difficulty of the game, with smaller ranges requiring from better knee control from the player. The player needs to hook the fish before it gets to the end of the trail, with the speed that the fish swims at being another variable that can affect game difficulty. The score is indicated by the fish count displayed in the upper right corner. When the knee deflects inward or outward beyond the permissible range, the hook will become transparent and cannot catch fishes.

1.3 Tactile stimulation for rehabilitation

Including tactile sensory stimulation during rehabilitation practices could have significant benefits. One big benefit can be deducted while studying the impact of tactile stimulation in motor learning. Motor learning is a highly complex phenomena involving both psycho-motor and cognitive aspects that happens when training our body to perform new motor skills [16], and thus a very important mechanism during rehabilitation. The addition of tactile feedback during motor learning activities has been studied and has shown some positive results. It has been used to improve immediate performance of a motor task [17] and in its long term retention [18]. Moreover, it has shown to enhance acquisition of fine motor skills [19], and facilitate their transfer from dominant to nondomninant hand [20]. In addition, it has resulted in accelerated psychomotor training [21]. In research utilising haptic feedback as a mean for motor learning, the feedback is almost exclusively related to performance error during the task [17, 20]. Two opposite approaches have been studied, error reduction and error augmentation. The former uses the feedback to actively help the player reduce the performance error, the latter uses the feedback to augment the perception of small errors. Both approaches have been linked to improved motor learning, but error augmentation shows more positive results in research [22].

Similarly, training with tactile feedback has been linked with improved proprioception [23, 24]. The proprioceptive sense informs of precision and accuracy during movement [25], and is therefore necessary to maintain motor control during exercise [26]. Therefore, an improved proprioception should translate to improved rehabilitation results, while also directly targeting the issue of reduced proprioceptive sense in arthritis patients [5].

In a different note, tactile feedback also shows promising results regarding the improvement of chronic pain. Research on the lower back shows that combining traditional rehabilitation exercises with sensory stimulation could have positive effects on dealing with chronic pain. The research relies on the theory that chronic pain is caused by the nervous system no longer processing sensory information normally after the surgical procedure. It therefore presents the hypothesis the inclusion of controlled tactile stimulation during the rehabilitation stage can help retrain the nervous system to interpret the stimulus as touch instead of pain. A new protocol, namely 'sensorimotor retraining' was presented and tested on patients with lower back pain and shows positive results in significantly reducing pain at least in the short term [27].

Another interesting area in which touch stimulation shows compelling results is specifically within virtual reality games. Some papers show that the inclusion of haptic feedback inside these virtual scenarios increases the immersion felt by the user [28]. This can be paired with the theory that increased immersion in VR environments could have positive results in motor performance [29].

1.4 Research goal

In this context this masters projects aims to study the capabilities of different haptic systems as integrated tactile feedback within the knee rehabilitation games being developed at the DPTA department. It will in particular focus on adding haptic feedback to the fishing game that focuses on knee extension.

The project will start with the design of multiple haptic systems, using as starting point the technologies available in the laboratory. Then the capabilities of each technology will be leveraged to choose the most optimal one for perceivable sensations that provide intuitive feedback. The chosen haptic technology will be integrated in game where it will provide real time feedback for the player. Finally, the designed haptic game will be tested in-game on healthy participants to analyse its potential on motor learning both short and long term when compared to a non-haptic version of the game. The immersive capabilties of both haptic and non-haptic version will also be compared.

This project first aims to solve the following design question:
- Can a haptic device be developed and integrated within the existing knee extension game aimed at knee rehabilitation, providing real time in-game feedback perceptible for the user?

It also aims to answer the following research questions regarding the effect of haptic feedback during the trained leg extension task:

- Does training with the developed exergame improve the performance of the knee extension task during gaming? Measured by analysing the improvement of
  - in game score
  - reaction time from the appearance of the target until correct position is reached

  during the training session
- Does the improvement in performance during training with the exergame persit through multiple training days? Measured by checking game performance markers before and after each of the training sessions.
- Does the addition of haptics improve the perceived immersion of the player during gaming per measurement of the Immersive Experience Questionnaire [30]?

1.5 Haptic systems

The word 'haptic' means 'pertaining to sense of touch'. Then, haptic technology can be defined as the science of applying touch sensation and control to interact with computer developed applications [31] One application of this type of technology is as feedback in virtual reality scenarios, where diverse types of haptic technologies could be used to give the user some sort of tactile sensation. Three examples of said technologies are disc or coin vibration motors, Peltier elements, and Ultrasonic arrays. These three techonologies are specially relevant to the project given their availability at the DPTA lab, and thus, used as a starting point for the design task.

Disc or coin vibration motors are small devices, generally around 9-10 mm in diameter. When subjected to certain voltage they generate a vibration that can be sensed when in direct contact with the skin. Their wide availability, simple working principle and low price, make them very good choices for haptic sensations [32, 33].

A Peltier element is a device that transfers heat from one side to the other when a current is applied to it. This happens due to the Peltier effect, which occurs when a junction of two dissimilar conductors allows current to flow through, causing a generation or abosorption of heat at said junction. Depending on the material combination and current flow, the heat is liberated or removed, causing either heating or cooling down. Peltier elements are engineered under this principle, and controlling the current flows through them allows controlling the temperature that each side of the element reaches [34]. Peltier elements have multiple applications, often being used as coolers, but their ability to absorbe and generate heat energy by simply controlling the applied current also makes them suitable as haptic components. The change heat transfer of one of the surfaces can be used as sensory feedback, either in direct contact with the skin, or by heating some other medium.[35]

Ultrasonic waves can be used to generate haptic sensations mid air. The devices able to do this consist of an array of ultrasound emitting speakers, that emit ultrasound waves at higher frequency than the human ear can perceive. Control of each speaker in the array allows for control of the point in the 3D space that the waves coincide at. That point is called focal point, and the combined force from the ultrasound waves creates enough pressure for mechanoreceptors on the skin to be activated, which results in a feeling similar to touch [36]. The technology was developed to work on the hands, where mechanoreceptors are highly concentrated [37]. However, other parts of the body show a lower distribution of mechanoreceptors, which can make the technology less effective. A recent research studied the possibility of getting a good level of sensation in the forearm, and potentially other areas of hairy skin. A promising methodology, consist on placing the focal point 3 cm away from the target, and has shown to be effective at inducing some level of sensation. [38]

2 Materials & Methods

2.1 Study of different technologies as potential haptic feedback systems

Different technologies readily available in the lab were first studied as potential technologies to add haptic feedback to the knee extension game. A list of requirements was developed, that the systems would need to fulfil in order to be usable as a haptic sensation in the game.

- The system should be able to induce perceivable sensations around the knee of the participant consistently, to ensure that the player receives the desired feedback while playing
- The system should be able to induce the sensations while the player extends their knee in the game, as that movement would be performed while playing the game
- The sensation induced by the system should be adjustable in either intensity or some other scalable way, in order to allow for a more continuous feedback
- The changes in sensation should occur fast, allowing for smooth non delayed feedback while playing
- The induced sensations should not cause discomfort in 90% of healthy volunteers
- The system should be safe to use

2.1.1 Ultrahaptics

The STRATOS Inspire ultrasonic haptic device by Ultraleap [39] was first studied as a potential technology to add haptic feedback to the knee extension game. This system is designed to generate a sensation similar to touch and thus more directly translatable to real life sensations.

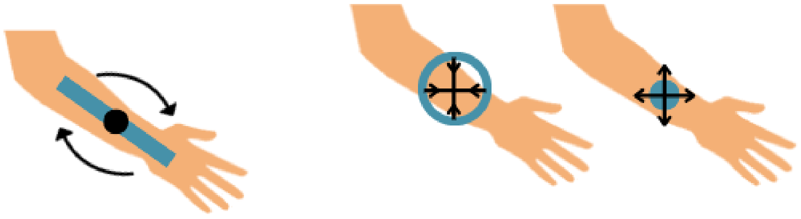
Haptic sensations were generated using the Unity package on the SDK V3 Software, provided by Ultraleap [40]. The sensations could simulate different shapes, visible on the computer screen, that were translated to the coordinates in the array of the device, which in turn generated ultrasonic waves in the desired position. Some of the sensations represented static shapes, say a circle or square, with changeable dimensions; while others allowed for some movement at a desired speed, like a rotating line, or a pulsating circle.

All sensations generated an easily perceiveable sensation on the palm of the hand of the researcher. The palm of the hand is a part of the body with an abundance of mechanoreceptors, which makes the sensations easily perceivable. However, this project requires haptic sensation around the knee, which has significantly less mechanoreceptors and thus causes the sensations to not be easily perceived in the area. Previous research dwelled in this issue and developed a protocol to increase sensation in hairy skin by placing the focal point of the sensation 3 cm away from the target area [38]. A similar approach was tested in this project. Different sensations were triggered mid air, and the forearm of the participant placed 3 cm away. Due to not having ethical approval for these test, they were carried out on two volunteers from the developing team. The pre-fabricated sensations with the parameters shown in Table 1 were tested to study the potential of the Ultrahaptic system. A diagram showing the shape of the sensations around the forearm can be found in Figure 2.

Sensation name	Size parameters	Value	Movement	Frequency range
Circle	Radius	0.035	none	NA
Line	Start	-0.05	none	NA
	Stop	0.5		
Scan	Length	0.01	Back and forth through y direction	Range: 0.5-10
	Travel distance	0.1		
Open	Minimum Radius	0.005	Breathing: Open+close	Range: 0.5-2
	Maximum Radius	0.035		
Rotor	Length	0.12	Rotation	Range: 1-10

Table 1: Tested pre fabricated sensations and parameters

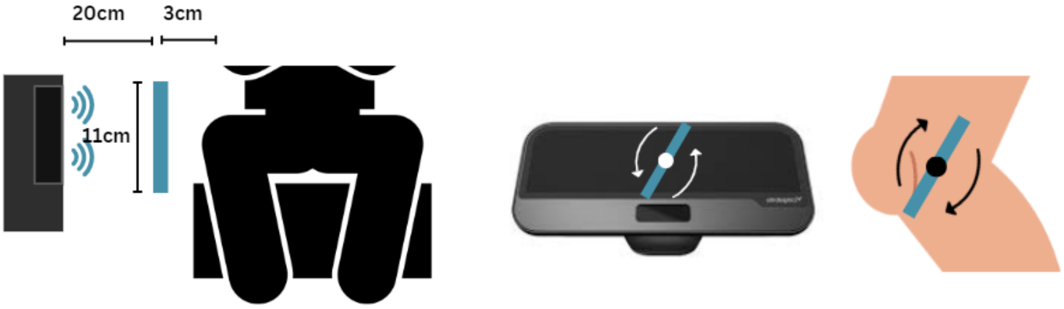
a line sensation **b** circle sensation **c** scan sensation



d rotor sensation **e** open-close sensation

Figure 2: Different sensation as they would be felt on the forearm

After testing that placing the focal point of the ultrasonic sensation 3 cm away from the skin in the forearm increased the perceived sensation in the area, a set up was developed to test whether the sensation was also increased in the knee area. The Ultrahaptic display was placed 23 cm away from a seated participant, and a sensation created at a fixed distance of 20 cm from the device, see Figure 3a for a schematic. With this set up the knee of the participants stayed 3 cm away from the created sensation. The 'rotor' sensation, a line of fixed length rotating at a fixed speed, was chosen to test the set up. This was done first of all because moving sensations were found to be more perceivable in hairy skin in the previous test. Another reason to choose the rotor sensation was it's ability to align the rotation with the knee and have a sensation that could be felt on the joint through it's extension. This alignment is illustrated in Figure 3b, and would be very useful for it's potential use in open kinetic exercise movements needed in game. Four tests were conducted on a volunteer from the research group. On the first test, rotation frequencies in the range 0-10 were randomly triggered and the participant instructed to inform the researcher at which times they could feel some sensation or change of sensation. This test aimed to prove that the sensations were perceivable at all in the knee area. In the second test the rotation sensation was triggered with different rotation frequencies after times of inactivity, and the participant was instructed to note at which times they could feel some sensation. The test aimed to analyse if there was specific rotation frequencies that caused an increased sensation. The third and fourth test were performed to study if changes in rotation frequency would cause different sensations. Frequencies of 2, 4, 6, 8 and 10 were used for both of these tests. In the third test, the rotation sensation was continuously triggered and the rotation frequencies changed randomly. In the fourth test the sensation was triggered at one frequency and then increased with a bigger or smaller jump. In both the third and forth test, the participant had to notify if a difference in sensation was perceived. In all tests the participant was wearing noise cancelling headphones to be blinded to the sound that the device makes when triggered.



a **b**

Figure 3: Ultrahaptic set up for testing haptic sensations on knee. a)Schematics of the set up b) Rotor sensation as generated in the array and projected on the knee

2.1.2 Vibration

Seeed Studio 31604000 Mini Vibration motors [41] were also tested as potential haptic devices. In order to induce a vibration sensation around the knee of the player the proposed design involved using multiple vibration motors positioned around the knee area. Previous research on vibration discrimination on the thigh and leg shows that for a distance of up to 7 cm in the tight and 6.8 cm in the leg, two vibration stimulations are perceived continuously [42]. This same study also found that up to 1.9 cm location error to a single vibration input is expected in both leg and tight [42]. Another similar study concluded that for the forearm, at least 8 cm separation between motors should be used to ensure separate perception of both stimulations [43]. With the knee area being less sensitive than the forearm [37], worse discrimination ability is expected in this area. Based on this, it was theorised that arranging the motors in a ring surrounding the knee cap, at less than 6 cm from each other, and under 1.9 cm from the kneecap itself could induce a continuous sensation on the knee (see Figure 4).

Another factor taken into account when arranging the motors around the knee is that vibration applied directly on a muscle insertion site can cause illusory sensations of movements of said muscle [44]. Illusory sensations of the rectus femoris, one of the main muscles in charge of leg extension, would negatively affect proprioception of the knee joint. Since the femoral tendon is exposed, and easily accessible, it was decided to not place any motor directly above the knee joint, to reduce the likelihood of illusory sensations in said muscle. Other muscles related to leg extension do not have exposed attachment sites in proximity to the joint, so the possibility of illusory sensations on them was discarded.

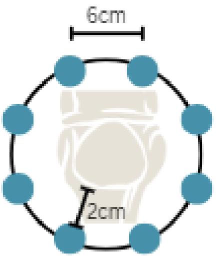


Figure 4: Motor configuration around knee

The described set up would require multiple motors operating at the same time, so a circuit was designed able to control 6 motors at full amplitude. The schematic of the circuit can be found in Figure 5. In this circuit the Arduino pins were connected to the gate side of BS170 MOSFET N-channel transistors [45]. 6 motors were used since the Arduino Uno Board board used on the set up has that number of pwm pins. The drain side of the transistors was grounded and the source side connected to the negative end of the vibration motors. The positive end of the motors was connected to a 5 V DC power supply, which provided them with enough voltage and current to operate in full power. In this set up each of the motors could be controlled separately by each of the arduino pins. Moreover, power modulation was possible, allowing the Arduino Code to control the vibration amplitude of the motors. The example Arduino Code allowing to change the pwm signal through serial communication can be found in Appendix A.

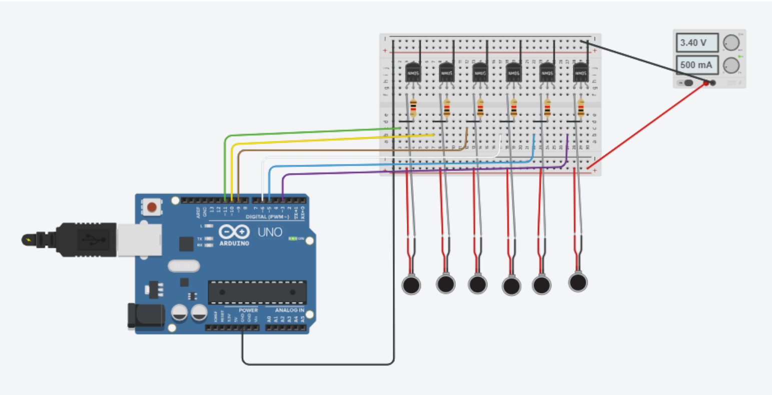
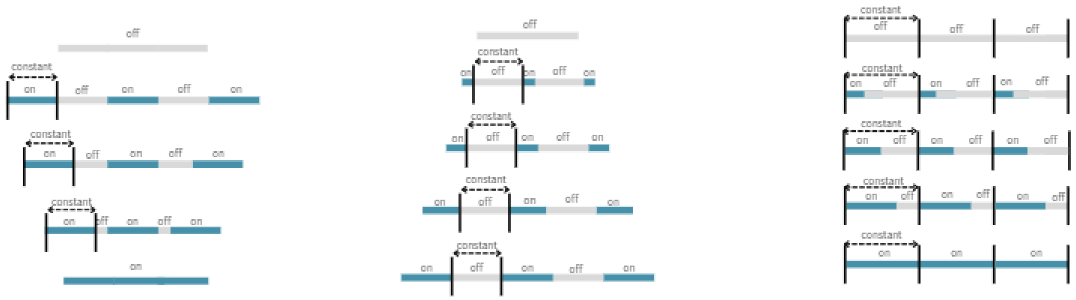


Figure 5: Set up with Arduino Uno to control 6 motors allowing pwm

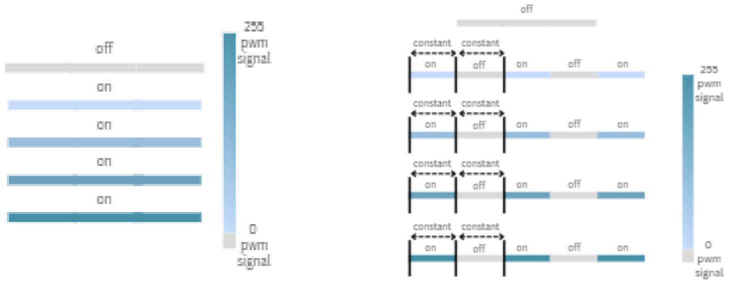
Different ways of providing haptic feedback through vibration motors can be defined. Previous projects within the research group successfully used spatiotemporal modulation, by activating vibration for short time pulses, to induce sensations via vibration motors. Another approach is amplitude modulation via pwm, which has also shown positive results in other research [46]. Three spatiotemporal modulation procedures, and two amplitude modulation ones were designed and tested internally within the research group. Each procedure had 5 vibration modes that could be used as haptic feedback, and aimed to induce an increasingly stronger sensation. The goal was finding the sensation that would create the most natural build up in sensation. Detailed explanations for each of the five procedures are given below. Options 1-3 use spatiotemporal modulation, whereas options 4 and 5 show amplitude modulation.

- Option 1. In this option the motors would be turned on for a constant amount of time periodically. The period would decrease at each mode by decreasing the delay between the ’on’ intervals. This option is illustrated in Figure 6a, and the Arduino code for it can be found in A. Maximum delays, and ’on’ interval lengths of 100, 200 and 300 ms were tested.
- Option 2. In this option the motors would be turned off for a constant amount of time periodically. The period would decrease at each mode by increasing the amount of vibration time between the ’off’ intervals. These mode are illustrated in Figure 6b, and the Arduino code for it can be found in B. Maximum ’on’ intervals, and ’off’ interval lengths of 100, 200 and 300 ms were tested.
- Option 3. The vibration period, including on+off time would remain unchanged, with the vibration time increasing and off time decreasing at every mode. This mode is illustrated in Figure 6c, and the Arduino code for it can be found in C. Interval lengths of 100, 200, 300 and 400 ms were tested.
- Option 4. In this option the motors would be constantly turned on and receive a higher pulse width within pwm protocol in Arduino at increased modes. This mode is illustrated in Figure 6e, and the Arduino code for it can be found in D.
- Option 5. In this option the motors would receive increased width pulses within pwm protocol each mode, but always vibrated at a certain period instead of continuously. This mode is illustrated in Figure 6d, and the Arduino code for it can be found in E. Interval lengths of 100, 200 and 300 ms were tested.

Inconsistent vibration amplitude within each of the motors pointed towards discarding the amplitude modulation option, as it would not results in consistent feedback. Vibrating at full amplitude, even if not exactly equal at every motor, at least ensured constant sensation across all motors at all modes.



a constant ’on’ time **b** constant ’off’ time **c** constant on+off time interval



d amplitude change continuous **e** amplitude change periodic

Figure 6: Different options for vibration modes

2.1.3 Peltier elements

Peltier elements were also tested as potential haptic devices. Two Peltier components, a 12x12 mm adaptive module by RS Components [47] and a 8x8 mm adaptive module by RS Components [48] were used for the preliminary tests. The 12x12 mm component had a heat sink attached to one of the sides, mounted during previous research. The idea being explored was using the devices to induce temperature changes around the knee of the participant. Peltier elements would need to be placed directly against the skin of the player, and the heat flow to the side directly against the skin controlled by means of voltage, inducing a temperature change perceiveable by the player.

As a first test, both components were directly connected to a power source at different voltages in their working range. 0-0.9 V for the 8x8 mm device and 0-2.75 V for the 12x12 mm device. At each different voltage the devices were powered for 30 seconds, and some cool-down time was allowed in between tests at different voltage. The current at which the systems operated was recorded, to characterise their power consumption. A note of the heat change perceived when directly touching the element after powering it for 30 seconds was also made choosing from the options ’no perceivable change’, ’small change’, ’big change’, ’hot side burns’. This was done in order to try to find a range with perceivable temperature changes that would not harm the participant.

As a second test the 12X12 mm Peltier component was connected to an h-bridge module [49] in order to allow for control of current flow in both directions, allowing to either heat or cool down a given side of the element. The h-bridge was connected to a power source at 2.5 V in one test, and 5 V in another, both within its working range, and connected to two pins in an Arduino Uno board [50], to control the power and direction. An schematic of the circuit can be found in Figure 7. Pulse width modulation was used to heat up the device at different rates. Different states in the 0-255 bit range with 25 step size within the pwm protocol in Arduino were used to power the Peltier element for 30 seconds. The current and perceived temperature were recorded the same way as in the previous test.

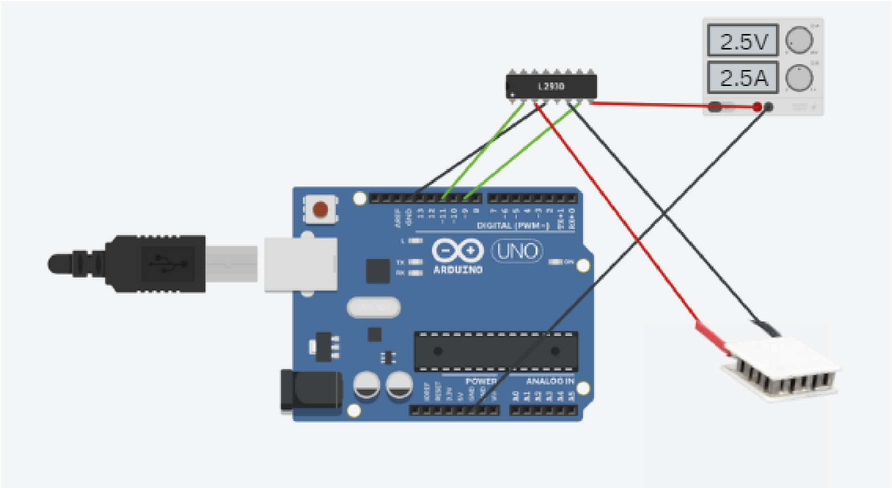


Figure 7: Circuit with Arduino Uno and H-Bridge to control heating and cooling of Peltier Element

2.2 Design of Vibration Motor haptic feedback system

The final set up for a haptic system able to induce reliable sensations around the knee as in game feedback was based on vibration motors. The next step was designing a system that would provide such feedback and integrate it in game. Knowing the base technology that would be used, the requirements defined in subsection 2.1 could be further detailed. The following requirements were defined to ensure safety, constant sensation and comfort when dealing specifically with vibration motors as actuators.

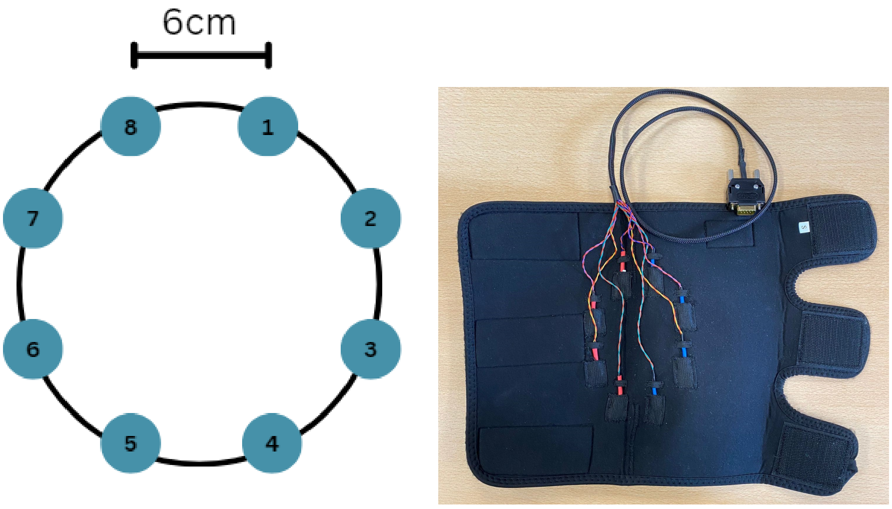
2.2.1 Requirements and wishes

- The motors must be less than 6 cm away from each other to ensure constant sensation
- The motors must stay attached to the knee through the whole movement
- The placement of the motors must not disturb EMG+IMU sensor placement required for playing the game
- The motors should be easy to put on and off in the desired area
- The motors should be attachable to people with different knee circumference sizes
- The control of the motors could be done wirelessly to allow for participant comfort
- The motors could be controlled separately for easier troubleshooting. While for this application the motors are triggered synchronously, separate control could be beneficial for future applications.
- The vibration amplitude of the motors could be was controllable, notwithstanding if intensity change was needed for the final application. This allows for future applications to be able to use this metric

2.2.2 Final system Vibration motors

Based on the conclusions drawn either from literature or from previous testing, a system, able to comply with the updated requirement list was designed and built.

The final device consisted of 8 motors placed 4-6 cm apart in a ring configuration, as portrayed in Figure 8a. 8 motors proved to be a sufficient number ensure constant vibration but create a ring shape big enough to cover the knee area. The motors were attached to commercially available knee sleeves via some pockets sewn onto the fabric. A small band was also added on top of the pockets to better secure the motors and reduce the likelihood of them breaking due to free vibration. The chosen knee sleeves could be opened and closed via velcro, making them adjustable to different leg sizes, and easily attachable even with the motors on the inside. Three different knee sleeve sizes were used for better fit on people with different leg sizes, with the motors being placed slitghly closer in the smallest size (4 cm) and further in the biggest one (6 cm), but always within the defined 6 cm threshold for constant sensation. The circuit used to control the motors required each them to be connected to the source side of a transistor on one end end and to to all share common power source on the other. Then, the positive side of all the motors was connected to a common wire, and the other side connected to a wire per motor. This results in 9 wires, that were connected to a db9 female connector. Figure 8b, shows the final design with sewn pockets and connected motors. The pinnout of the connections is shown in Figure 23 in Appendix G.



a Motor configuration in final design **b** Designed knee sleeve

The full schematic of the circuit used to control the motors is shown in Figure 24 in Appendix H. An Arduino Nano BLE [51] board was used as a microcontroller, chosen due to its ability to connect through Bluetooth, and thus provide vibration wirelessly. Two lithium batteries connected in series provide the board with 7.4 V through the Vin pin. 3 capacitors (1 nF, 10 nF and 10 μF) were added to the voltage source to remove low, mid and high frequency noise. A simple switch was added from the power source, to allow for the system to be powered off. The digital pins D2-D9 in the design were connected to the gate side of BS170 MOSFET N-channel transistors, with 10 KΩ resistors reducing the current. The transistors were grounded in the drain side. The source side was connected to a db9 male connector, that can connect to the motors in the designed knee sleeve. The power for the motors common power source comes from the Arduino Nano 3.3 V pin output, which provides with enough Voltage for the motors, rated at 3 V working voltage. The current limitation of this pin is lower than in previous designs, which could reduce the vibration capabilities of the motors. However, the system showed to be able to consistently vibrate at a high enough amplitude to be perceivable. Capacitors were also added to this power source to remove noise. This allowed the system to run on a singular battery that can power the motors and the board, and provides a constant power supply, independent of battery life. The designed system allows for separate control of each of the motors by controlling the output of the D2-D9 digital pins. The circuit allows for separate control of each motor as well as power modulation, though not necessary for the task at hand. However it is important to note that due to o limitations of the Arduino Nano board, pwm can only be used on 4 motors at a time. This pins were connected to a female db9 connector following the same pinnout as within the knee sleeve design, for easier attachment.

A pcb board was designed an ordered with all the electronics, as shown in Figure 9. The Arduino nano board could be mounted to the designed pcb and the batteries connected to provide with power to the system. A 9 pin pinout for the 8 motors and power source was to be conected to the motors.

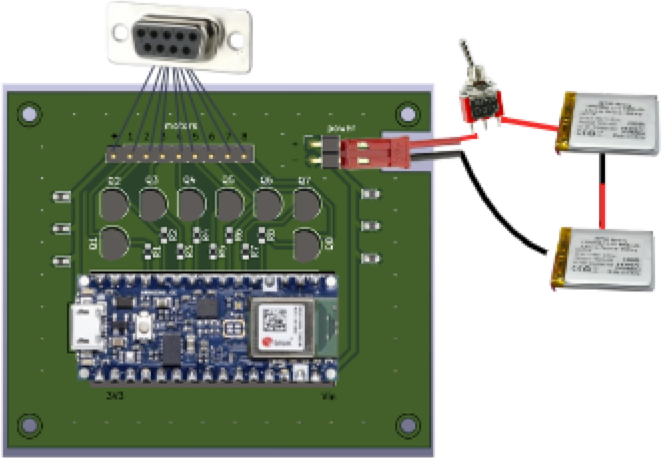


Figure 9: Designed PCB and connections

All the electronic components were placed inside a 3D printed box, see Figure 10. The switch stayed on the outside of the design, allowing to turn the system off. The DB9 connector was also accommodated outside, easily accessible for connection to the motors.

Figure 10: Final designed box

For control of the motors two Arduino Nano BLE [51] boards were used, connected via Bluetooth to eachother and working in a master-slave like control. The first board was connected to the circuit and motors and acted as a peripheral device. The code uploaded to this board induced the different vibration modes on the motors. These different vibration modes were triggered by messages sent by the second board, the central device. This second board was connected through serial communication to the game, and sent messages to the first board based on the game error. The code uploaded to each of the boards can be found in Appendix I for the slave device, and Appendix J for the central one. Within the code for the peripheral device there is two characteristics accessible through bluetooth.

One, the 'switchMode' is the one accessed by the central device to change the vibration mode. The other one, 'test', was added to be able to test all 8 motors were properly working by triggering them one by one, to ease troubleshooting.

The general working principle of the whole system is described in Figure 11.

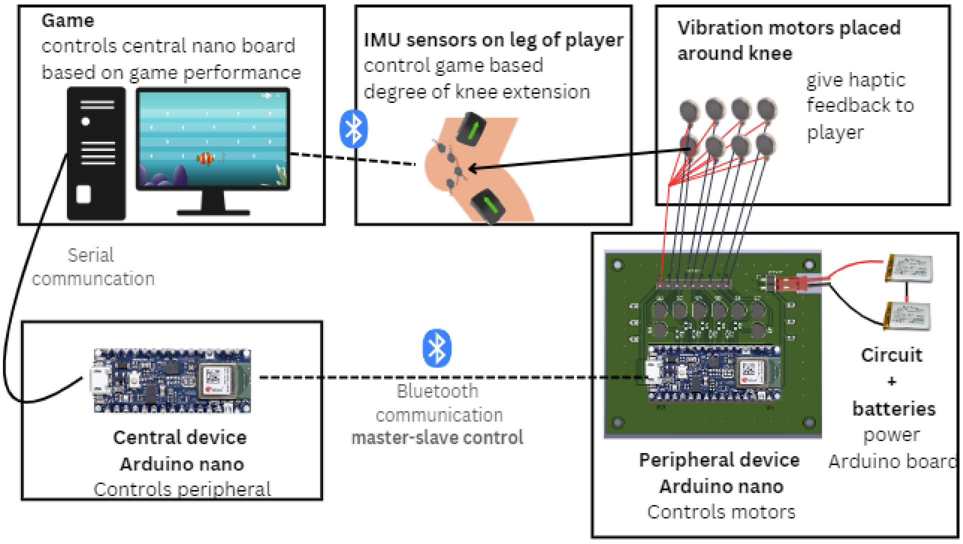


Figure 11: Working principle of game with haptics

The vibration modes that were used for the game are illustrated in the schematic in Figure 12. The choice for the type of vibration was based on testing within volunteers inside the research group, who found the chosen option most intuitive, and with smooth yet perceivable diferences between modes. The motors had 5 vibration modes, the first one being the motors fully turned off and the last one the motors fully on. At each higher mode the motors spend more time on and off, with a set interval of 200 ms between full cycles.

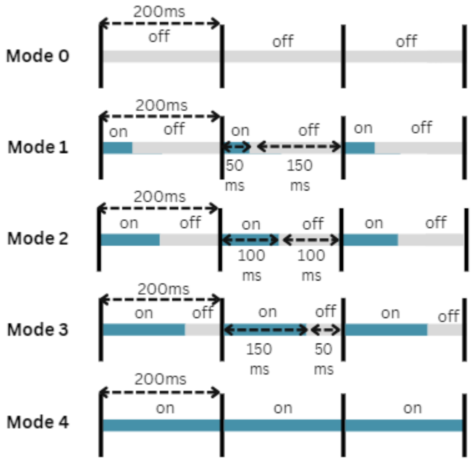
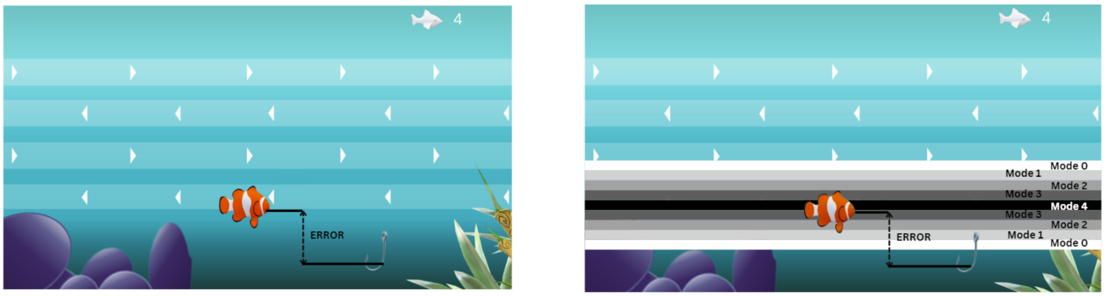


Figure 12: Vibration modes integrated on game

The way this different modes related to the game error is illustrated in Figures 13a and 13b. The player controls the height of the hook on screen by extending or flexing their knee. The distance in the y direction from hook to fish is an error that the player needs to minimise to catch the fish. When the error is within the accepted threshold for hooking, the system will fully vibrate (or be in Mode 4); the vibration mode gets lower the bigger the error in either direction of the fish, with the areas for this being as wide as the range for effective hooking. When the error is big, and thus, visually clear, no vibration is provided.

The addition of haptic feedback to other game variables, such as height of fish or height or hook were also contemplated; but relating to performance error better aligns with current research on the topic [17, 20]. The type of feedback represents an error augmentation protocol, given its positive results in previous research on motor learning [22]. Small hook position errors, are be augmented due to the haptic feedback. This also provided with a more intuitive vibration profile, that increased as the fish was getting hooked.



a **b**

Figure 13: Visualisation of a)fish-to-hook error, and b)areas for modes of vibration based on error

The haptic system was integrated on the game by introducing serial communication with the Arduino Nano Central Device, sending the appropriate messages when crossing the boundaries between areas described in Figure 13b. The version of the game with integrated haptics included 30 fish at every round of the game. Each of the fish took 3 seconds to cross the screen from the appearance side, and required 1 seconds of the hook being at less than 50 pixel distance for successful hooking. The repository with the full Unity code for the haptic game can be found in GitLab [52]

2.2.3 Effect on muscle activation

In order to ensure that the final vibration system had not effect in muscle activation some test were carried out where EMG recordings of different muscles proximal to the device were analysed in vibration and no vibration conditions. These tests were performed on the researcher and Delsys Trigno EMG sensors [53] were used for the recordings.

On the first test, EMG signal of three knee extensor muscles, namely rectus femoris, vastus lateralis and vastus medialis was recorded. All of this muscles insert at the knee and take part in knee extension which is the movement that the game trains. Recordings of the EMG at 90º of extension, where the muscles are relaxed, and at 45º and 90º of extension, where the muscles are active were acquired.

In the second test, EMG signals of the peroneus longus and tibialis anterior muscles were recorded at the three knee extension positions and two vibration conditions. This muscles do not partake in knee extension movement so their activation should not be affected by degree of knee extension, but do insert at the knee, and thus could be affected by the vibration device.

In the last test EMG recordings of Peroneus Longus, Tibialis Anterior and Rectus Femoris while playing the game with and without haptic feedback were performed. The EMG of these three muscles will get recorded in future participant studies with the haptic game, so it was specially important to make sure that their readings were not altered due to the vibration.

For all of the tests, frequency analysis of the readings was carried out to observe if the vibration frequencies of the motors had any effect. The motors vibrate at 10Hz and 55Hz frequencies [41], so the power of the normalised FFT at those frequencies was compared for the vibration and no vibration conditions.

2.3 Study of effect of vibration on knee extension task

The developed game was used in a randomised controlled trial study to analyse the effect of the addition of haptic sensation primarily in motor learning and secondarily in immersion. The research ethics committee of the faculty of arts (CETO) of the university of Groningen reviewed the protocol with ID 99483973 and had no objection against the study.

22 participants had to be recruited for the study amongst students and staff from the University of Groningen. Sample size calculation was not possible due to lack of previous research on the effect of vibrational feedback of motor control on the knee. However, based on other studies with similar research questions, analysing the effect of haptic feedback on motor learning on hand fine motor skills [21, 20] and a steering task [20], a total of 6-8 participants per group was considered enough for the project. 8 participants in the control group and 8 participants in the trial group give a total of 16 participants. Assuming a drop out rate of 25% a target of 22 participants was set. Within the masters project the first five participants were piloted, with 2 being in the control group and 3 in test group.

2.3.1 Study design

Testing took place during three consecutive days for each participant, with each session being scheduled approximately 24 h appart from the previous one. The non-dominant leg of the participants was used for the movements in all testing stages. In the first day baseline measurements were recorded, to fit the game to the participants knee extension capabilities. Then the participant was required to perform the movements and hold the positions required during the exergame. The baseline data was recorded by both IMU+EMG Delsys Trigno sensors [53] and Azure Kinect time of flight camera [54].

After this baseline acquisition the testing blocks started, which were the same in all three testing days. The first block consisted of the pre-test. The participants were required to complete 10 different angle knee bending and stretching positions, in the absence of haptic stimulation. After this pre-test, the second block started, which consisted of training with use of the exergame. Here, participats in the test group played 4 rounds of the exergame with vibration motor feedback around their knee.

Participants on the control group instead played the 4 training rounds without the vibration motor feedback. Each round of training consisted of 30 knee extension movements, with a total of 120 knee extensions for the duration of this second block. All participants were wearing the knee sleeve used for haptic stimulation, but only the test group received vibration feedback. All participants also wore noise cancelling headphones to ensure they could not hear the vibration, which would create some additional and unwanted auditory feedback. The final block involved the post test and took place after the training session. It consisted of 10 knee stretching and bending exercises in absence of haptic feedback. Only during the first training day, after the post-test, the participants filled in the Immersive Experience Questionnaire [30].

All blocks consisted of movements beginning from a starting location, with the participant sitting with their knees flexed in 90 degree. Participants begun each trial at the starting location and moved until they reached a position defined by the exergame at which time movement stopped. This movement trajectory was recorded by the detection system to compare with the standard movement trail for further analysis. The movement trajectory was also recorded during the game training, to analyse performance data while training with the different modes.

In pre-test and post-test session, participants followed the commands of the test module in the platform. In the game training session, participants conducted motor training according to the game instructions. After each round of the game 30 seconds of rest was be provided. If the participant requested an extension of the rest time this was allowed.

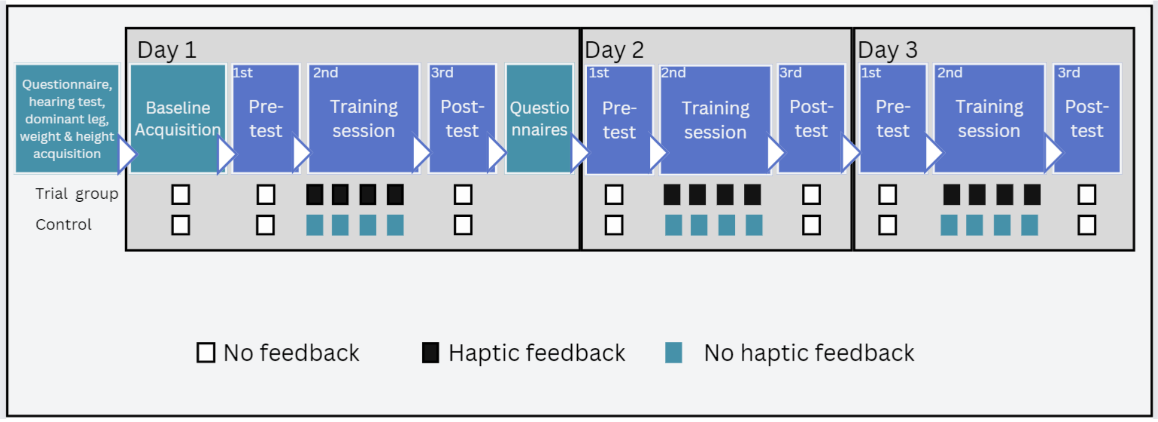


Figure 14: Flow chart of the study design overview

2.3.2 Statistical Analysis

In order to assess performance in game, 3 metrics were calculated from each game round for every participant:

- Game score, defined as the percentage of effective knee extension movements per round, that resulted in fish successfully getting caught
- Reaction time, defined as the time from appearance of the fish in game until the game successfully got caught
- Average time between hooking events, defined as the average time between hooked fish

All the metrics were averaged individually for control and test groups and plotted with standard deviation for visualisation and comparison between groups. The specific percentage of effective knee extension movements per in game trail, related to 0º, 30º, 60º and 90º degrees of knee extension were also calculated and plotted to analyse potential differences between knee flexion degrees.

Statistical analysis was performed specifically for game scores. Since this preliminary comparison was performed in a small fraction of the sample size, normality could not be assumed so non parametric tests were used.In order to compare performance between test and control group, an unpaired two-sided Mann Whitney U statistical test was performed. This is the non parametric alternative to a unpaired t-test, that has been used in literature to compare performance of groups under non-haptic and haptic conditions [17]. The test is unpaired given data points from different participants in each of the groups are being compared. A two tailed test was selected since its a first attempt at studying the effect of vibration in knee extension, so the performance could be improved or reduced. The test was conducted for the scores at different stages through training in order to study changes in the potential performance difference due to training. The compared stages were:

- Scores of all played rounds
- Score of first round at each training day
- Score of last round at each training day
- Scores during each training day

A one tailed parametric Wilcoxon test was used to compare performance at difference stages through training within groups. This test has also been performed in existing literature to compare performance before and after haptic training [21]. A paired test was used given scores from the same participants are used when comparing performance within a given group. In this case a one-tailed test was used given increased performance is expected due to training in any condition. Scores at day 1 of training are compared with scores at day 3 to study the long term learning effect. While the scores at the beginning and end of each day are compared to study short term learning.

A Friedman test is the non-parametric alternative to ANOVA, performed in multiple studies to compare scores at different training stages [17, 20]. The Friedman test was conducted for the scores during each of the training days for each group individually.

The score of the Immersive Experience Questionnaire, which gives a metric for general immersion felt by the player, were computed for each participant as defined by the creators [30]. Scores for each of the 5 immersion factors (cognitive involvement, real world dissociation, emotional involvement, challenge and control) were also calculated for each participant. The average scores for each of the groups were calculated and reported. The immersion for each group can be compared knowing higher scores show a higher degree of immersion.

3 Results

3.1 Ultrahaptic sensations on knee

The perceived sensations as reported by a healthy volunteer upon using the rotation sensation in their knee with the set up as shown in Figure 3a at different rotation frequencies are shown in Tables 18, 19, 20, and 21 in Appendix K. In the first test, the participant perceived a sensation when frequencies of 8, 2, 10, 1 and 20 were selected in Unity, and described them as a breeze or wind, but could not report a noticeable difference between different frequencies. In the second test, the participant could consistently notice some sensation when frequencies of 8 and 6 were turned on after a period of no sensation. Frequencies of 10 and 4 were perceived some of the times they were turned on, but not consistently. The frequency of 2 was not perceived any of the triggered times. During the third test the participant continuously felt a sensation at all the rotation frequencies in the range 2-10, but could not report a difference between them in any of the frequency changes. In the fourth and final test the participant could identify a frequency change from 2 to 10 and from 4 to 10, described as ’more intense wind’, but could not perceive smaller jumps.

3.2 Peltier components test results

The results of the test conducted with both Peltier elements connected to a power supply at different voltages are shown in Tables 2 and 3. For each induced voltage the current is reported upon powering the device (0 s) and after 30 seconds of induced voltage. The perceived heat is represented by colours, where green means small change in temperature, orange represents a easily perceivable change in temperature, and red means the device was burning hot. For the 8x8mm device perceivable sensations were felt at a minimum 0.6 V after 30 seconds; and after 0.8 V immediately. For the 12x12 mm device voltages of 1.25 V and higher caused perceivable sensations after the 30 second mark; and voltages of 2.5 V and 2.75 V caused immediate heat. From a voltage of 2.25 V the device got too hot where it could not be touched anymore.

V	Current [mA] 0 s	Current [mA] 30 s
0.1	0.2	0.15
0.2	0.45	0.3
0.3	0.8	0.55
0.4	1.1	0.8
0.5	1.4	0.9
0.6	1.7	1
0.7	2	1.3
0.8	2.2	1.3
0.9	2.5	1.4

Table 2: Current at different induced voltages for 8x8 mm Peltier element

V	Current [mA] 0 s	Current [mA] 30 s
0.25	0.28	0.22
0.5	0.6	0.45
0.75	0.88	0.68
1	1.3	0.85
1.25	1.5	1.05
1.5	1.8	1.2
1.75	2.2	1.4
2	2.2	1.6
2.25	2.2	1.8
2.5	2.2	
2.75	2.2	

Table 3: Current at different induced voltages for 12x12 mm Peltier element

Tables 4 and 5 report the current and perceived sensation when using the 12x12 mm Peltier module as in set up shown in Figure 7 at different pulse widths in the 0-255 range for pulse modulation in Arduino. When the device was powered at 2.5 V, pulses of 150 and up caused a perceivable sensation; while when the device was powered at 5 V, pulses of 125 and up caused a perceivable sensation. The device never reached too hot temperatures.

pwm pulse	Current [mA]
25	0.05
50	0.1
75	0.15
100	0.2
125	0.25
150	0.3
175	0.35
200	0.4
225	0.5
250	0.4

Table 4: Current after powering 12x12 mm Peltier element at 2.5 V with different pulse widths

pwm pulse	Current [mA]
25	0.15
50	0.3
75	0.4
100	0.6
125	0.8
150	0.9
175	1
200	1.1
225	1.2
250	0.9

Table 5: Current after powering 12x12 mm Peltier element at 5 V at different pulse widths

3.3 EMG with and without vibration

3.3.1 Knee extensors

Tables 6, 7 and 8 show the power at 10 Hz and 55 Hz of the normalised Fourier Transform of the EMG signal of three knee extensor muscles (Rectus Femoris, Vastus lateralis and Vastus medialis), with and without vibration. The values are shown for 3 different knee flexion degrees for each of the muscles. At 90º, where the muscles are relaxed; and at 45º and 0º, where the muscles are working to extend the knee. For all of the conditions the power of the signal at 10 Hz and 55 Hz is almost 0, with values sometimes being higher and sometimes lower for the vibration condition.

	10 Hz		55 Hz	
Knee extension angle	Vibration on	Vibration off	Vibration on	Vibration off
0º	1.28E-07	1.91E-08	9.02E-09	9.99E-09
45º	2.40E-07	4.51E-08	2.70E-10	5.22E-10
90º	4.35E-09	5.27E-08	1.62E-09	2.72E-10

Table 6: Power of EMG signal at 10 Hz and 55 Hz of **Rectus femoris** muscle

	10 Hz		55 Hz	
Knee extension angle	Vibration on	Vibration off	Vibration on	Vibration off
0º	3.01E-09	2.76E-08	3.11E-07	1.59E-07
45º	5.46E-08	2.19E-08	1.02E-08	4.24E-08
90º	4.80E-08	2.96E-08	5.77E-09	1.16E-08

Table 7: Power of EMG signal at 10 Hz and 55 Hz of **Vastus lateralis** muscle

	10 Hz		55 Hz	
Knee extension angle	Vibration on	Vibration off	Vibration on	Vibration off
0º	3.13E-08	2.75E-08	5.36E-07	5.58E-08
45º	7.85E-09	2.85E-08	5.21E-08	4.80E-08
90º	1.19E-09	8.54E-11	9.23E-10	4.24E-10

Table 8: Power of EMG signal at 10 Hz and 55 Hz of **Vastus medialis** muscle

3.3.2 Muscles in viccinity of vibration device

Tables 9 and 10 show the power at 10 Hz and 55 Hz of the normalised Fourier Transform of the EMG signal of three muscles not part of knee extension but close to the vibration device, with and without vibration. The values are shown for 3 different knee flexion degrees for each of the muscles, 90º, 45º and 0º. For all of the conditions the power of the signal at 10 Hz and 55 Hz is almost 0, with values sometimes being higher and sometimes lower for the vibration condition.

	10 Hz		55 Hz	
Knee extension angle	Vibration on	Vibration off	Vibration on	Vibration off
0º	2.66E-08	5.24E-08	5.42E-09	1.25E-09
45º	4.71E-10	2.13E-08	1.46E-09	2.85E-10
90º	3.11E-09	7.70E-09	1.24E-09	1.78E-09

Table 9: Power of EMG signal at 10 Hz and 55 Hz of **Tibialis Anterior** muscle at 3 different knee flexion degrees with and without vibration

	10 Hz		55 Hz	
Knee extension angle	Vibration on	Vibration off	Vibration on	Vibration off
0º	5.45E-09	3.02E-09	1.53E-09	1.98E-09
45º	3.91E-09	2.66E-10	4.55E-09	1.29E-10
90º	2.49E-09	1.47E-09	4.12E-10	6.34E-12

Table 10: Power of EMG signal at 10 Hz and 55 Hz of **Peroneus Longus** muscle at 3 different knee flexion degrees with and without vibration

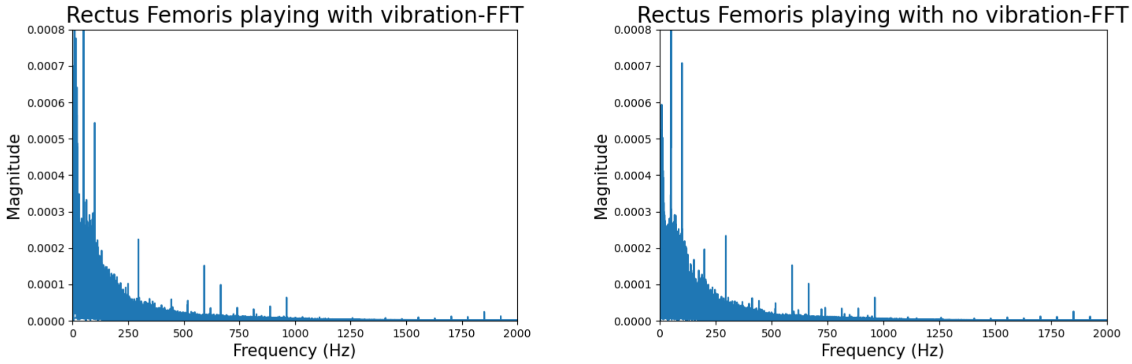
3.3.3 During playing

Table 11 shows the power at 10 Hz and 55 Hz of the normalised Fourier Transform of the EMG signal of Rectus Femoris, Peroneus Longus and Tibialis Anterior while playing a round of the game with and without haptic feedback. For all of the conditions the power of the signal at 10 Hz and 55 Hz is almost 0.

	10 Hz		55 Hz	
Muscle	Vibration on	Vibration off	Vibration on	Vibration off
Rectus Femoris	4.82E-07	2.71E-09	1.25E-08	1.62E-09
Peroneus Longus	3.33E-09	3.04E-09	1.30E-09	6.01E-10
Tibialis Anterior	9.14E-08	1.84E-08	5.32E-09	7.48E-09

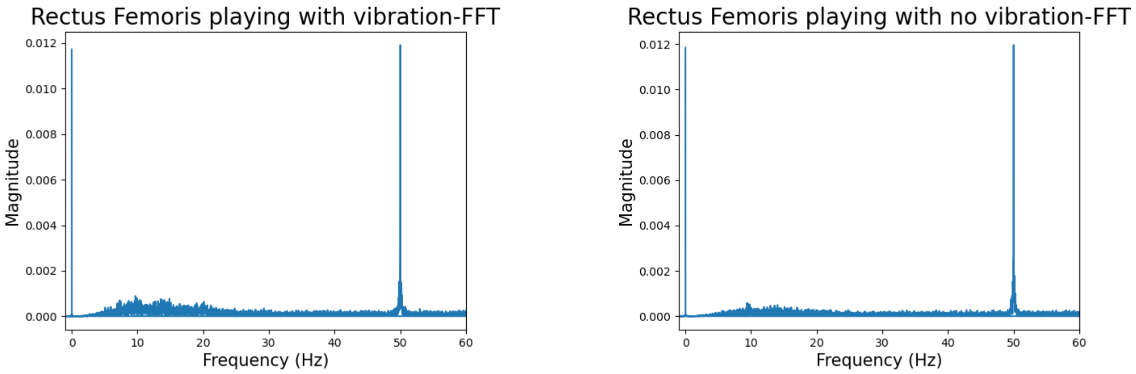
Table 11: Power of EMG signal at 10 Hz and 55 Hz of **Rectus Femoris Tibialis Anterior** and **Peroneus Longus** muscles while playing game fish game with and without haptics

Figure 15 shows the normalised Fourier transform for the full frequency domain of the Rectus Femoris muscle in the vibration and no vibration conditions. The y axis is cropped at 0.0008 to better observe the magnitude of all frequencies, since the peaks at the lower frequencies are otherwise the only observable ones. Figures 16, 17 and 18 show the Normalised Fourier Transform of each of the muscles under the vibration and no vibration condition, in the 0-60 Hz frequency range. For both conditions at each muscle the peaks are found in the same frequencies; and frequencies 10 Hz and 55 Hz do not have a visually discernible peak.



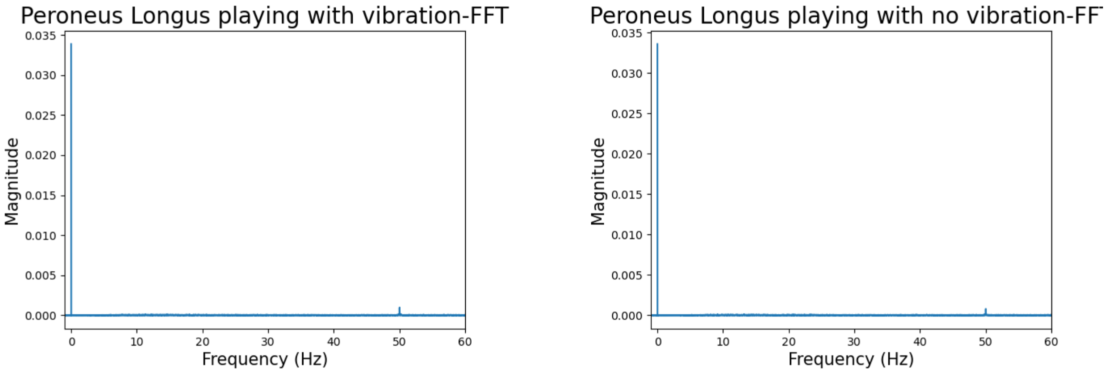
a With haptics **b** Without haptics

Figure 15: Normalised Fourier Transform of Rectus Femoris EMG signal while playing game fish game with and without haptics



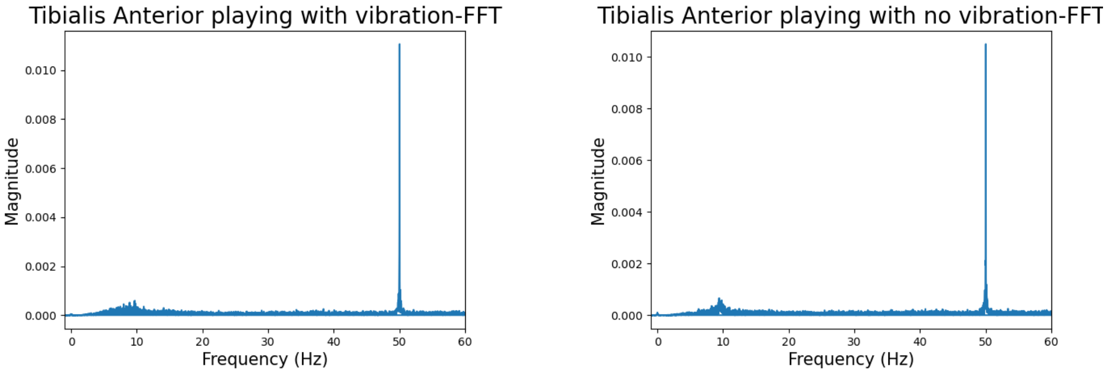
a With haptics **b** Without haptics

Figure 16: Frequencies 0-60Hz of normalised Fourier Transform of Rectus Femoris EMG signal while playing game fish game with and without haptics



a With haptics **b** Without haptics

Figure 17: Frequencies 0-60Hz of normalised Fourier Transform of Peroneus Longus EMG signal while playing game fish game with and without haptics



a With haptics **b** Without haptics

Figure 18: Frequencies 0-60Hz of normalised Fourier Transform of Tibialis Anterior EMG signal while playing game fish game with and without haptics

3.4 RCT results

3.4.1 Game scores

The game scores for the control and test groups at each round of the game are visualised in Figure 19. The score represents the percentage of effective knee extension movements, that resulted in a successfully hooked fish, performed at that round. Different colour backgrounds are used for each of the training days, including four game rounds each. The graph shows the averaged score for test and control groups with standard deviation for each of the rounds as dots with errorbars, and the average per day per group as dotted lines. For both groups the scores are higher at the last round than at the first round of each of the training days; with scores at rounds 2 and 3 sometimes surpassing the last round. For both groups the scores are consistently higher during the third day when compared to the first day. The test group obtained an average score slightly higher the second day when compared to the first day, while the control groups score was lower during that day. However, individual round scores were higher for rounds 1, 3 and 4 for test group, and rounds 2, 3 for the control group. Moreover, scores of the test group are higher than those of the control group for every round of the game.

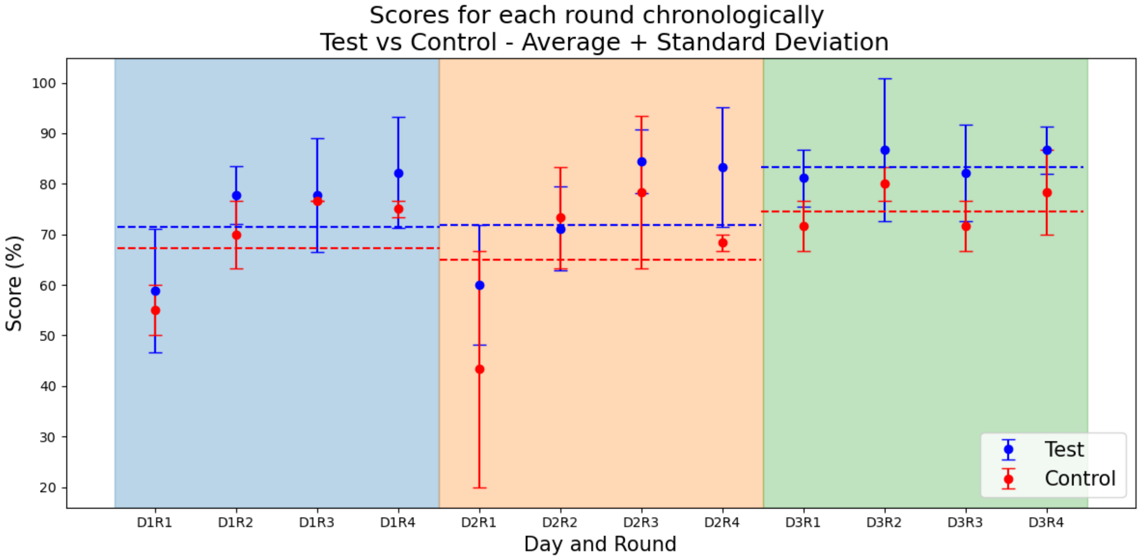


Figure 19: Average percentage of effective knee extension movements and standard deviation for test and control group participants per game round

The p-values of a Mann-Whitney U statistical test performed on scores of test and control group participants can be found on Table 12. A $p<0.05$ is observed when comparing all of the scores through the different rounds.

The p-values of a Wilcoxon test performed on scores at different stages during training can be found in Table 13 for the control group and Table 14 for the test group. For both groups a $p<0.1$ is observed when the scores of the first day with the score of the last day; and comparing every score of the first round of each day with every score of the last round of each day yields a $p<0.05$.

The p-values of a Friedman statistical test performed on scores of different training days and control group participants can be found in Table 15 for the control group and Table 16 for the test group. All the comparisons yielded $p>0.05$.

Variables compared	**p-value**
All Scores	**0.033**
Scores first round first day	1
Scores last round first day	0.8
All scores first day	0.485
Scores first round second day	0.554
Scores last round second day	0.554
All scores second day	0.312
Scores first round third day	1
Scores last round third day	0.374
All scores third day	**0.057**

Table 12: p-values of **Control vs Test** game scores compared by Mann-Whitney U statistical test

Variables	**p-value**
Scores day 1 vs day 3	0.054
Score day 1 round 1 vs day 3 round 4	0.25
Score day 1 round 1 vs round 4	0.25
Score day 2 round 1 vs round 4	0.159
Score day 2 round 1 vs round 4	0.5
Scores round 1 all days vs round 4 all days	**0.04**

Table 13: P-values for **control** group scores at different time points compared by Wilcoxon test

Variables	**p-value**
Scores day 1 vs day 3	0.055
Score day 1 round 1 vs day 3 round 4	0.125
Score day 1 round 1 vs round 4	0.25
Score day 2 round 1 vs round 4	0.125
Score day 2 round 1 vs round 4	0.125
Scores round 1 all days vs round 4 all days	**0.0039**

Table 14: P-values for **test** group scores at different time points compared by Wilcoxon test

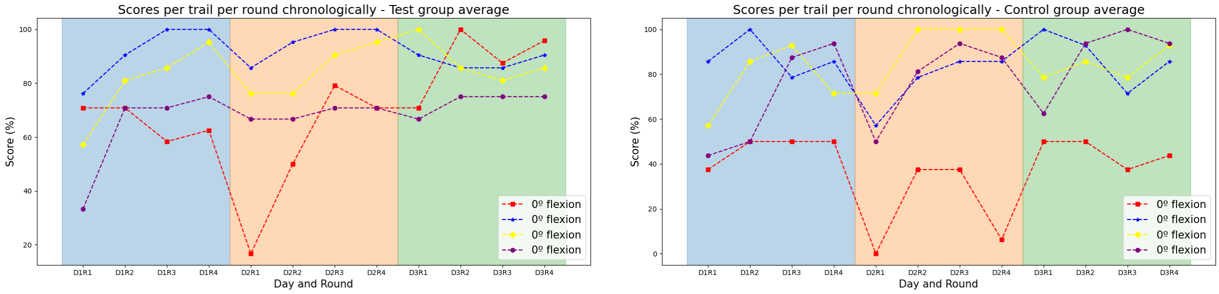
Variables	P-Value
All scores day 1 vs day 2 vs day 3	0.166
Scores round 1 day 1 vs day 2 vs day 3	0.223
Scores round 4 day 1 vs day 2 vs day 3	0.223

Table 15: p-value of **control** group scores at each of the three training days compared by Friedman test

Variables	P-Value
All scores day 1 vs day 2 vs day 3	0.173
Scores round 1 day 1 vs day 2 vs day 3	0.148
Scores round 4 day 1 vs day 2 vs day 3	0.913

Table 16: p-value of **test** group scores at each of the three training days compared by Friedman test

The percentage of effective knee extension movements per round for each of the four knee extension degrees relating to a swimming trail in game are shown in Figure 20. For both groups the 0º knee extension had the lowest scores during the first day of training, with only test group increasing the scores at day 3. The 90º knee extension shows low scores compared to the other trails for the test group consistently through training; while the control group scores lower at the beginning of training days and increasingly higher as rounds progress.



a Test **b** Control

Figure 20: Average percentage of effective knee extension movements at each of the trail heights for a)test and b)control group participants per game round

Figure 21 shows the average time in seconds between 'hooked' fishes at each training round for test and control group participants. The dots represent the average value per group, and the error bars show the standard deviation. Each training day is shown with a different background for ease of visualisation, and a dotted line represents the averaged value through every round at that day. The test group spent less time in average to successfully catch a fish in every round of training when compared with the control group. The time between caught fish got shorter at each training day for the test group; while the control group had a shorter time the last day compared to the first, but higher values the second day. However the whiskers overlap for every comparison, so the differences are likely not significant.

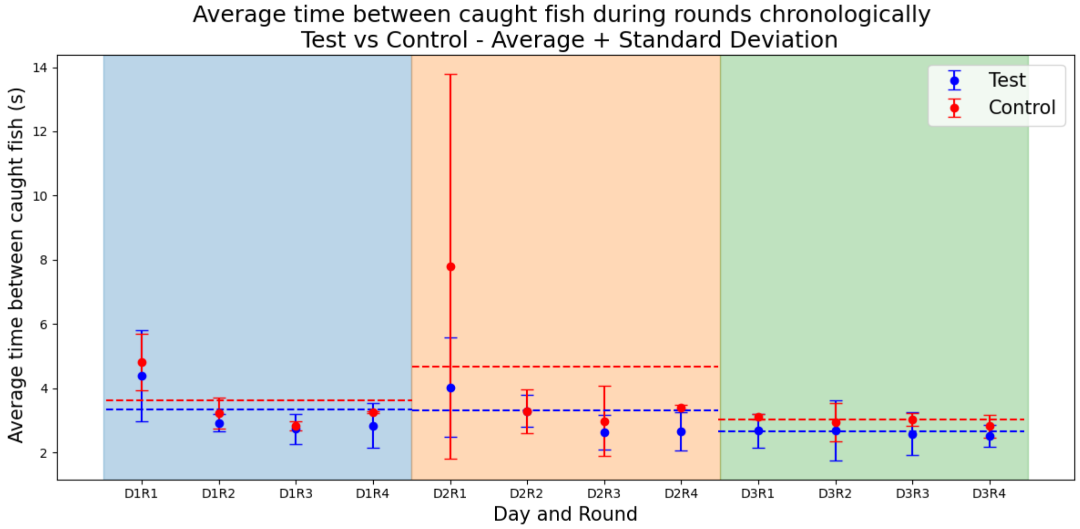


Figure 21: Average time and standard deviation between hooked fishes for test and control group participants per game round

Figure 22 shows the average time in seconds that passed from a fish appearing in screen until it got hooked for the test and control group participants. The dots represent the average value per group, and the error bars show the standard deviation. Each training day is shown with a different background for ease of visualisation, and a dotted line represents the averaged value through every round at that day. During the first training the test group was consistently faster at catching the fish; the second day the control group took less time in average in every round except the second one; and the third day the test group was faster in rounds 1, 2 and 4. When looking at the performance inside each group per day, the control group got faster at each consecutive day; while the test group got slower the second day and faster the third one. However, the whiskers overlap for both test and control group comparisons, and round comparison between groups; so a significant difference is unlikely. The first round in the first day does show a clear difference between control and test group, with no overlapping whiskers.

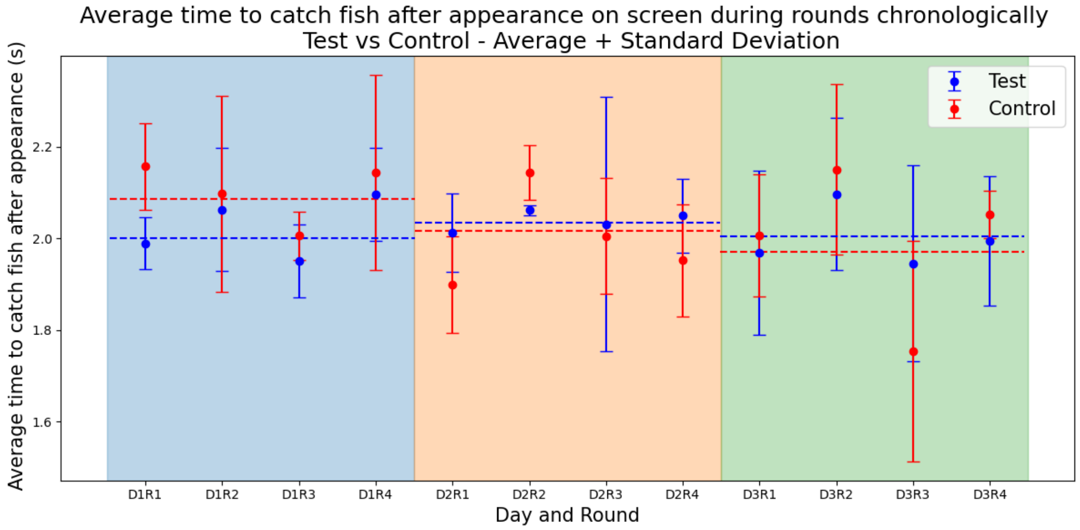


Figure 22: Average time and standard deviation from fish appearance in screen until hooking in effective movements for test and control group participants per game round

3.4.2 Immersion Questionnaire

Scores for the Immersion Questionnaire as filled by the participants are reported in Table 17. The scores are averaged and the standard deviation provided for the final question of the questionnaire, the total score, and the score of the five immersion factors. The average score is higher for every single metric in the test group; but the standard deviations overlap for all calculated metrics.

	CONTROL	**TEST**
Final question	4.5 ± 2.12	8.33 ± 1.15
IEQ Score	**103.5 ± 30.41**	**119 ± 14.73**
Cognitive involvement	57.5 ± 4.95	61.33 ± 3.51
World dissociation	-1 ± 11.31	2.67 ± 2.31
Emotional involvement	61 ± 14.14	72.67 ± 5.5
Challenge	21.5 ± 3.54	22.33 ± 5.51
Control	37 ± 8.47	39 ± 5.57

Table 17: Average and standard deviation of immersion questionnaire scores for control and test groups

4 Discussion

4.1 Ultrahaptics as haptic sensation

During the tests conducted on a colleague it was observed that ultrahaptic sensations had the potential of being perceived around the knee area; but provided an unreliable method to induce said sensation. While the participant was able to recognise a sensation multiple times during test 1 and test 2; and could consistently feel it during tests 3 and 4; the perception was still not reliably perceived; failing to comply with the first requirement in Subsection 2.1. Moreover, the sensations induced by the array were not easily adjustable in intensity in a way perceived by the user, not fulfilling the third requirement. While an on and off feedback could be provided via ultra haptic sensation, test 2 is inconclusive in determining whether rotation can consistently be detected when turned on. Similarly, higher and lower rotation frequencies could be used as haptic feedback, but tests 3 and 4 show that only big jumps in frequency are perceived and only sometimes. Another requirement, namely the second one, the sensation being perceived continuously through the movement, is also not fully met. The set up as described in Figure 3a aims to align the rotation sensation with the knee to ensure it acting on it during the gaming experience; but during testing, small movements of the participants in any direction would break this alignment and stop the perception. This also makes the set up not comfortable, as it reduced mobility.

Since the Ultrahaptics device did not comply with the requirements describe in subsection 2.1, it was discarded as an option for haptic feedback within the project. However, the device still shows promise. The tests conducted for this project observed that inducing a moving sensation 3 cm away from the knee can cause a perceivable sensation on the participant. Future research could expand this testing by including more participants to study the range of perceivable frequencies. A more complex set up, able to adapt the position of the sensation to the movement of the leg, could also be developed, making the device usable at the knee.

4.2 Peltier as haptic sensation

The preliminary tests in the Peltier component showed that for both the 8x8 mm and 12x12 mm modules, direct control of the Voltage linearly translated to control of the current. Also for both components a voltage at which the heat difference immediately became easily perceivable was recorded. However, for the bigger device, that voltage also caused temperatures unsafely high after some seconds.

The preliminary tests with Arduino control showed that lower currents could be achieved for the same Voltage through this set up. Moreover, for similar currents, the perceived heat was lower than with direct voltage control. In the 12x12 mm Arduino, an induced voltage of 2.5 V only resulted in very slightly perceived temperature changes; and with a voltage of 5 V the perceived heat was only strongly perceived with pulses of 225 and 250.

Another issue observed during testing had to do with heat dissipation, which, even for the module equipped with a heat sink, was not fast enough. After using the module for some seconds the whole device would get hot, and the temperature of a singular face would not be easily detected anymore. This happened even when attempting two sided current flow with the h-bridge.

In general, the Peltier elements failed to comply with multiple requirements in the list in subsection 2.1, not providing with a sensation that could be using during all the game as overheating would happen, not showing a clear scalable way to induce said sensation, failing to induce a sensation fast enough when controlled by a microcontroller; and even reaching unsafe temperatures in some tests. Because of this, it was rejected as a possible haptic feedback system.

However, a different implementation of a Peltier Module could still be used in future research. Multiple Peltier modules, equipped with more powerful heat sinks could be used, to increase the contact points with the participant. The addition of a control loop could ensure the temperatures reached by the device are safe but still perceivable; providing with a more reliable way to induce temperature change than direct control of voltage and current.

4.3 Vibration system as haptic sensation

The vibration motors complied with the initial requirements described in subsection 2.1 from the initial tests, making them a good choice for haptic feedback in the exergame. Using 8 vibration motors in a ring like distribution at 4-6 cm distance resulted in a continuous sensation around the knee, that all pilot participants described as a strong sensation around their kneecap; fulfilling the first requirement. The sensation was perceivable all through the extension movement when placing the motors in a knee sleeve, even though some users reported the vibration to be more muted at 0º extension, due to the knee sleeve being less tight at that position. The second requirement was thus fulfilled, but more knee sleeve sizes to ensure a better fit should be considered in future applications. As for the type of feedback, multiple ways of inducing scalable feedback were studied, shown in Figure 6; and all provided with perceivable sensation changes. The lack of consistency in vibration intensity of different motors discouraged the use of amplitude control mechanisms, so spatiotemporal feedback was instead chosen. The type of spatiotemporal feedback chosen was solely based on tests on volunteers within the research group and aimed to make the feedback feel the most natural. This design choice could be a limitation of the project, as different ways of inducing feedback by means of vibration could have different effects. Future research could focus on studying the effect of different vibration modes in a bigger population. The changes in sensation modes when using the Arduino micro controller were felt immediately, both in the set ups with serial communication and when using Bluetooth, providing smooth transition between sensation modes and immediate feedback in game, ensuring the fourth requirement is met. All healthy volunteers reported no discomfort when using the system, albeit some tightness at the bottom of the movement, further motivating the need for more adjustability in the knee sleeve. Finally, the successful use of the system in all volunteers shows its safety.

As for the second set of requirements defined specifically for the vibration motors in Section 2.2.1; they are also fulfilled by the designed system. The knee sleeves with pockets for the motors are very easy to attach to the participants, with different sizes being usable by people of different knee circumference sizes. The motors are safely attached to the knee sleeve, and stay in place through the movement; but are also detachable from the device for easy clean-up and repair. Wireless communication was achieved ensuring better participant comfort and EMG+IMU sensors were placed in all participants not affected by the vibration devise. The motors are also controllable in asynchronous mode due to the circuit design, allowing for testing and different types spatiotemporal control in future applications. Finally, the system allows for intensity control of 4 of the 8 motors, due to limitations of the Arduino Nano BLE board. The same design with a different microcontroller would allow for power modulation in all motors.

The EMG tests in leg muscles with and without vibration, described in Subsection 3.3, show that the device seems to have no effect in activation of muscles in the vicinity, nor does it affect their EMG signal in any other way. For both muscles active during knee extension and other muscles with insertion at the knee the EMG signal doesn't show frequencies of 10 Hz and 55 Hz related to the vibration of the device. Peaks are also observed at the same frequencies in the normalised Fourier transform of all muscles at the same frequencies. This also confirms that EMG can be used to analyse muscle activation data during the trial study for both control and test participants.

As for the application of the feedback in game, all volunteers reported that the device is mostly comfortable and the vibration feedback intuitive and helpful during the game. The system was able to provide feedback as expected during all played game rounds, and consistently provided vibration at the correct time. However, the Bluetooth communication failed at two instances, requiring a restart of the game. This is theorised to be due to increased distance between the master and slave device, and could become a limitation of the system should it happen more often.

4.4 Randomised control trial study

Pilot results from the first 5 participants already show some promising results on the effect of vibration on motor learning, with improved performance markers in the vibration groups. However, not much can be said from such a small sample and further analysis is necessary to reach conclusive results.

The comparison of game scores shows the test group performing better through all of the training rounds (Fig. 19), and $p<0.05$ in the statistical comparison of all results (Table 12) suggests there might be a significant difference between this scores. Moreover, no significant difference is found when comparing the scores of individual rounds or days the first two days. In the third day a $p=0.057$ is observed, much lower than the previous days. While this is still not showing a statistical significance of $p<0.05$, a $p<0.1$ already shows a 90% chance of the difference not being coincidental. This could imply the difference in performance gets bigger after some training, though analysis of the full sample is required to see the real effect. On the other hand the clear difference in reaction time between groups at the first round of the game as seen in Figure 22 could signify that the game is easier to understand thanks to the haptic system. This could imply improvement in immediate learning due to the device. Statistical analysis should confirm this with all participant data.

Comparisons between test and control group with preliminary data indicates that the group with vibration performs better at the beginning of the task, and consistently performs better during the task with the difference getting even greater at the end of training. While this results are quite illustrative in showing a positive effect of vibration in performance of the knee extension task, as well as both immediate and sustained learning, statistical tests with more participants are required to give a more final conclusion.

For both groups, scores at the last round of a training day are consistently higher than at the beginning (Figure 19), pointing to some motor learning happening due to training with the device. No statistical difference is found when individually comparing first and last round at each day for any of the groups

(Tables 13 and 14). The very limited data from only 2 participants in control group and 3 in test group could be a reason for no significance in this comparisons. When comparing scores for all the first rounds and all the last rounds, thus increasing the data points for comparison, a $p<0.05$ is observed in both groups. Therefore, a trend is found in both groups improving their performance within a singular training session.

For both groups improvement of scores at consecutive days during training can be observed, which could mean that motor learning is sustained in time. Significance is not observed when comparing the scores during the three days (Tables 15 and 16), which could indicate rate of the learning is not fast enough to show significance in shorter periods. Comparison of the scores for the first and last day in both groups yield a $p=0.054$ for the control and $p=0.055$ in the test group. This results do not show significance, but do mean there is more than a 90% chance the results are not coincidental. To reach conclusive results about this, data from all participants should be analysed, as the limited amount of data points enhance the effect of outliers.

Specific analysis of the performance of the test and control participants at catching fish specific to each of the trails give some information on the ability of the device to help with performance at each knee flexion degree. The 0º knee extension position, total extension, is the hardest position to reach. This is the lowest scoring position for both groups at the beginning of training; and only the test group is able to improve the performance at reaching this extension degree after training. This could mean the vibration device is helping the participants find this difficult position; but data from more participants is required to reach a conclusion. Another interesting observation is regarding the 90º extension position. The test group stays quite consistently around a 70% of performance for this trail, excepting the very first round. The control group on the other hand has a performance as low as 40% at the beginning of each training day, but reaches values as high as 90% at the end. It is theorised that this could have to do with the game itself, since the fish in the lowest trail swims slightly lower than the trail itself, which makes it hard to catch without haptics. The vibration very clearly shows where the fish is positioned, helping the test group learn this position faster. Data from more participants should be used to analyse this.

As for the other performance metrics, that is, time between fish (Fig. 21) and reaction time to hooked fish (Fig. 22), not much can be concluded from the pilot data. It seems that for the average time between caught events the test group performs better; but given the metric is related to performance scores this is to be expected. Moreover, only the test group improves performance in consecutive days, but with the reduced data points the outlier in the control group at Day 2 Round 1 could be the cause of that. As for the reaction time to caught fishes, the test group performs better at the beginning, but then performance seems to not clearly be superior for one of the groups after that. For both of the metrics significant differences should be analysed once all the participants are tested.

In terms of answering the research question posed in the project, the randomised controlled trial still needs to be carried to completion before reaching any conclusive results. However, the pilot data from the first five participants looks promising. In general, it seems that training with the game in both conditions results in improved performance, which is expected from literature [17]. Moreover, the addition of haptic feedback in the form of vibration translates into improved performance at the knee extension task in game. This aligns with prior research where the group training with haptics also consistently performed better [21]; and had reduced errors during training [17]. For the goal of rehabilitation, improvement in the task during the game itself could already translate into more effective training, and therefore give faster results. However effect of the haptic feedback in the task outside of the game environment still needs to be studied by comparing the performance in the pre and post test. Research on the topic shows that improved performance can sometimes translate into improved performance in the task itself [55], with haptic training with vibration specifically showing positive relation with joint position control [24].

A limitation was observed during gaming that could effect the results. Calibration problems at the game start, together with drift of the sensors during playing, resulted on the knee extension degree not always translating to the desired hook height. The conditions were the same for both groups, and training itself was still possible, deeming the game still useful for knee extension training. However, the relation between actual knee degree angle and hook position in game was not consistent for the participants, so transfer of knee extension control to outside of the game environment might be affected by this.

As for the results of the immersion questionnaire (Table 17), increased metrics in the final IEQ score for the test group, together with increased metrics for all other immersion factors, indicates the addition of haptics could positively increase immersion. Higher responses by the test group in the final immersion question 'How immersed did you feel?' that informs about overall perceived immersion in participants also aligns with this. However, significant differences would need to be studied after experiment completion with all participants.

5 Conclusions

The design question posed for this project was positively answered by the design of a haptic vibration device tested on multiple participants in game. Said device was able to wirelessly provide haptic feedback related to game events on real time and was perceived through the whole knee extension motion for all users. However, some design parameters need to be improved in future generations. More knee sleeve sizes are necessary for continuous contact of the motors to every participants knee, to ensure the vibration is consistent all through the movement. The problems regarding loss of bluetooth connection might need to be addressed if the issue persists. Moreover, other types of vibrotactile feedback, with different spatiotemporal variation, or including amplitude control could also be implemented.

The vibration device performed satisfactorily during testing for the first 5 participants of a randomised control trial to test for the effect of vibration in motor learning. While the study needs to be carried to completion to study significant differences in motor learning, the preliminary results are positive in showing correlation between training with the device and improved short term performance, together increased performance overtime.

Regarding the other technologies studied in the project, their potential as haptic feedback for the knee extension task cannot be ruled out yet. Limited testing in only one participant and one sensation for the Ultrahaptic technology showed unreliable perception of sensation. However, more sensations, or different set ups could yield more positive results. Similarly, limited testing with 2 Peltier elements in very simple settings yielded unsatisfactory results. Nonetheless, a more advanced circuit with a control protocol could result in satisfactory perception.

Ethical concerns

In the design and implementation of a haptic feedback device aimed at enhancing knee rehabilitation through a game, it is essential to address various ethical considerations to ensure the safety, well-being, and informed consent of participants. The primary ethical issue involves ensuring the device's safety, particularly regarding the currents and voltages used. Given that the device has been tested and proven safe, with no risk of tissue damage or discomfort, this fundamental concern is mitigated [56]. The exclusion criteria for participants during the randomized controlled trial, such as allergies to plaster or common fabrics, further highlight the ethical commitment to protect individuals from potential adverse reactions. Additionally, the inclusion of healthy participants who can perform the knee extension movements safely ensures that the risk of harm is minimized. Participants are monitored for fatigue, and any signs of inability to perform the exercises safely result in immediate cessation to prevent injury. This precaution reflects the ethical principle of beneficence, aiming to maximize benefits and minimize harm.

Another significant ethical aspect is the recruitment and participation of volunteers in the trials. Privacy and data protection are also paramount, particularly when using video games as therapeutic tools. Participants are required to sign informed consent forms, ensuring they are fully aware of the study's nature, the procedures involved, and their right to withdraw at any time without penalty. They also explicitly consent to the storage and use of their data in research for the next 15 years, with the option to withdraw their data at any time, excepting results already published. This aligns with the ethical guidelines discussed in practical VR research [57] and in the use of video games as therapeutic tools [58], which emphasize the importance of informed consent and participants' autonomy.

The integration of haptic feedback in the form of vibration is designed to enhance motor learning and immersion without causing discomfort or blurring the lines between reality and digital environments. Concerns about blurring real and digital realities are mitigated in this context, as the game clearly distinguishes itself from real-life activities [59]. While the game currently does not induce motion sickness due to its 2D nature, attention must be given to potential information overload from triggering multiple senses simultaneously, visual and tactile in this version, with plans to include auditory feedback in future iterations. Thus, monitoring and assessing participants' responses to the multisensory feedback is crucial to ensure that the experience remains beneficial and not overwhelming, as highlighted in VR research ethics [57]. Similarly, concerns about consent due to digital touch [56] are diminished, as the player has clear knowledge of when and how they will receive haptic feedback.

Additionally, considering the potential for computer game addiction, it is essential to monitor participants' engagement levels and provide debriefing sessions to discuss their experiences and feelings post-trial. This is increasingly important in future applications as rehabilitation tools and also aims to prevent any negative psychological impact, maintaining participants' motivation and confidence in their rehabilitation progress [58].

In summary, the ethical considerations for integrating haptic feedback into a rehabilitation game encompass ensuring participant safety, informed consent, and the protection of privacy. Continuous monitoring and debriefing sessions are crucial to address any emerging issues, ensuring that the research adheres to ethical standards while providing potential benefits for motor learning and rehabilitation.

Bibliography

[1] Patient Resources, "Total knee replacement: A patient's guide." `https://orthop.washington.edu/patient-care/articles/knee/total-knee-replacement-a-patients-guide.html#:~:text=Many%20studies%20show%20that%2090,desired%20level%20of%20recreational%20activity`. Accessed 22nd April 2024.

[2] A. Lübbeke, A. Silman, C. Barea, D. Prieto-Alhambra, and A. Carr, "Mapping existing hip and knee replacement registries in europe," *Health Policy*, vol. 122, no. 5, pp. 548–557, 2018.

[3] Johns Hopkins Medicine. `https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/knee-replacement-surgery-procedure#:~:text=Knee%20replacement%2C%20also%20called%20knee,joint%2C%20along%20with%20the%20kneecap.`, Dec 2021. Accessed 22nd April 2024.

[4] K. Konnyu, L. Thoma, W. Cao, R. Aaron, O. Panagiotou, M. Bhuma, G. Adam, E. Balk, and D. Pinto, "Rehabilitation for total knee arthroplasty: A systematic review," *American Journal of Physical Medicine & Rehabilitation*, vol. Publish Ahead of Print, 03 2022.

[5] B. Al Lawati, S. Mockett, and M. Doherty, "Static postural sway, proprioception, and maximal voluntary quadriceps contraction in patients with knee osteoarthritis and normal control subjects," 09 2021.

[6] D. Barrett, A. Cobb, and G. Bentley, "Joint proprioception in normal osteoarthritic and replaced knees," *The Journal of bone and joint surgery. British volume*, vol. 73, pp. 53–6, 02 1991.

[7] J. Taylor, "Proprioception," in *Encyclopedia of Neuroscience* (L. R. Squire, ed.), pp. 1143–1149, Oxford: Academic Press, 2009.

[8] W. Macrae and J. Bruce, "Chronic pain after surgery," pp. 405–414, 09 2008.

[9] V. Wylde, S. Hewlett, I. Learmonth, and P. Dieppe, "Persistent pain after joint replacement: Prevalence, sensory qualities, and postoperative determinants," *Pain*, vol. 152, pp. 566–72, 03 2011.

[10] P. Glare, K. Aubrey, and P. Myles, "Transition from acute to chronic pain after surgery," *The Lancet*, vol. 393, pp. 1537–1546, 04 2019.

[11] O. Wilder-Smith, "Chronic pain and surgery: A review of new insights from sensory testing," *Journal of pain & palliative care pharmacotherapy*, vol. 25, pp. 146–59, 06 2011.

[12] V. Wylde, J. Dennis, R. Gooberman-Hill, and A. Beswick, "Effectiveness of postdischarge interventions for reducing the severity of chronic pain after total knee replacement: systematic review of randomised controlled trials," *BMJ Open*, vol. 8, p. e020368, 02 2018.

[13] University of Groningen. `https://www.rug.nl/news/2023/06/revalideren-met-behulp-van-virtual-reality-games?lang=en`, Jun 2023. Accessed 22nd April 2024.

[14] M. García-Sánchez, P. García-Robles, M. C. Osuna-Pérez, R. Lomas-Vega, E. Obrero-Gaitán, and I. Cortés-Pérez, "Effectiveness of virtual reality-based early postoperative rehabilitation after total knee arthroplasty: A systematic review with meta-analysis of randomized controlled trials," *Applied Sciences*, vol. 13, no. 7, 2023.

[15] J. Byra and K. Czernicki, "The effectiveness of virtual reality rehabilitation in patients with knee and hip osteoarthritis," *Journal of Clinical Medicine*, vol. 9, no. 8, 2020.

[16] J. Krakauer, A. Hadjiosif, J. Xu, A. Wong, and A. Haith, *Motor Learning*, vol. 9, pp. 613–663. 03 2019.

[17] L. M. Crespo and D. J. Reinkensmeyer, "Haptic guidance can enhance motor learning of a steering task," *Journal of Motor Behavior*, vol. 40, no. 6, pp. 545–557, 2008. PMID: 18980907.

[18] L. Marchal-Crespo, S. McHughen, S. Cramer, and D. Reinkensmeyer, "The effect of haptic guidance, aging, and initial skill level on motor learning of a steering task," *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, vol. 201, pp. 209–20, 10 2009.

[19] U. Radhakrishnan, L. Kuang, K. Konstantinos, F. Chinello, and C. Pacchierotti, "Haptic feedback, performance and arousal: A comparison study in an immersive vr motor skill training task," *IEEE Transactions on Haptics*, 09 2023.

[20] A. Basteris, L. Bracco, and V. Sanguineti, "Robot-assisted intermanual transfer of handwriting skills," *Human Movement Science*, vol. 31, no. 5, pp. 1175–1190, 2012.

[21] D. Kaber, L. Tupler, M. Clamann, G.-H. Gil, B. Zhu, M. Neubert, W. Jeon, Y. Zhang, X. Qin, W. Ma, and Y.-S. Lee, "Evaluation of an augmented virtual reality and haptic control interface for psychomotor training," *Assistive technology : the official journal of RESNA*, vol. 26, pp. 51–60, 02 2014.

[22] L. Y. Liu, Y. li, and A. Lamontagne, "The effects of error-augmentation versus error-reduction paradigms in robotic therapy to enhance upper extremity performance and recovery post-stroke: A systematic review," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, 07 2018.

[23] D. Santis, J. Zenzeri, M. Casadio, L. Masia, A. Riva, P. Morasso, and V. Squeri, "Robot-assisted training of the kinesthetic sense: Enhancing proprioception after stroke," *Frontiers in human neuroscience*, vol. 8, p. 1037, 01 2014.

[24] A. Cuppone, V. Squeri, M. Semprini, L. Masia, and J. Konczak, "Robot-assisted proprioceptive training with added vibro-tactile feedback enhances somatosensory and motor performance," *PLOS ONE*, vol. 11, p. e0164511, 10 2016.

[25] D.-H. Lee, J.-H. Lee, S.-E. Ahn, and M.-J. Park, "Effect of time after anterior cruciate ligament tears on proprioception and postural stability," *PLOS ONE*, vol. 10, pp. 1–10, 09 2015.

[26] D. Riva, R. Bianchi, F. Rocca, and C. Mamo, "Proprioceptive training and injury prevention in a professional men's basketball team: A six-year prospective study," *The Journal of Strength and Conditioning Research*, vol. 30, pp. 461–475, 02 2016.

[27] M. K. Bagg, B. M. Wand, A. G. Cashin, H. Lee, M. Hübscher, T. R. Stanton, N. E. O'Connell, E. T. O'Hagan, R. R. N. Rizzo, M. A. Wewege, M. Rabey, S. Goodall, S. Saing, S. N. Lo, H. Luomajoki, R. D. Herbert, C. G. Maher, G. L. Moseley, and J. H. McAuley, "Effect of Graded Sensorimotor Retraining on Pain Intensity in Patients With Chronic Low Back Pain: A Randomized Clinical Trial," *JAMA*, vol. 328, pp. 430–439, 08 2022.

[28] M. Kim, C. Jeon, and J. Kim, "A study on immersion and presence of a portable hand haptic system for immersive virtual reality," *Sensors*, vol. 17, no. 5, 2017.

[29] A. Gokeler, M. Bisschop, G. Myer, A. Benjaminse, P. Dijkstra, J. Burgerhof, E. Otten, H. van Keeken, and J. Raay, "Immersive virtual reality improves movement patterns in patients after acl reconstruction: implications for enhanced criteria- based return-to-sport rehabilitation," *Knee Surgery Sports Traumatology Arthroscopy*, vol. 24, 10 2014.

[30] C. Jennett, *Is Game Immersion Just Another Form of Selective Attention?: An Empirical Investigation of Real World Dissociation in Computer Game Immersion*. University College London, 2010.

[31] M. Sreelakshmi and T. Subash, "Haptic technology: A comprehensive review on its applications and future prospects," *Materials Today: Proceedings*, vol. 4, no. 2, Part B, pp. 4182–4187, 2017. International Conference on Computing, Communication, Nanophotonics, Nanoscience, Nanomaterials and Nanotechnology.

[32] Meganburroughs, "Coin vibration motors." `https://www.precisionmicrodrives.com/coin-vibration-motors`, Jun 2022. Accessed 22nd April 2024.

[33] K. Sato and Y. Sugimori, "Development of a haptic interface with proximity sensors and vibration motors," *Journal of Mechanics Engineering and Automation*, vol. 5, Jun 2015.

[34] Welcome - Meerstetter Engineering, "Peltier elements." `https://www.meerstetter.ch/customer-center/compendium/70-peltier-elements`. Accessed 22nd April 2024.

[35] E. C. Rodrigues, E. B. Saleme, and C. A. S. Santos, "A haptic system for switching wind temperatures based on ultrasonic vibrations, peltier elements, and electrical resistances for multisensory applications," WebMedia '21, (New York, NY, USA), p. 73–80, Association for Computing Machinery, 2021.

[36] Ultraleap, "Haptics: Ultraleap." `https://www.ultraleap.com/haptics/`. Accessed 22nd April 2024.

[37] W. Navaraj, C. García Núñez, D. Sakthivel, V. Vinciguerra, F. Labeau, D. Gregory, and R. Dahiya, "Nanowire fet based neural element for robotic tactile sensing skin," *Frontiers in Neuroscience*, vol. 11, 08 2017.

[38] D. Pittera, O. Georgiou, A. ABDOUNI, and W. Frier, ""i can feel it coming in the hairs tonight": Characterising mid-air haptics on the hairy parts of the skin," *IEEE Transactions on Haptics*, vol. PP, pp. 1–1, 09 2021.

[39] Ultraleap, *Ultrahaptics STRATOS Inspire*, 14 ed., 2003. Accessed 22nd April 2024.

[40] Ultraleap, "Ultraleap." `https://www.ultraleap.com/`. Accessed 22nd April 2024.

[41] R. Nederland, “Seeed studio 316040001 mini vibration motor for multipurpos.” https://nl.rs-online.com/web/p/power-motor-robotics-development-tools/1845122?cm_mmc=NL-PLA-DS3A-_-google-_-PLA_NL_NL_Raspberry_Pi_%26_Arduino_%26_ROCK_%26_Development_Tools_Whoop+(2)-_-(NL:Whoop!)+Power+%26+Motor+%26+Robotics+Development+Tools-_-1845122&matchtype=&pla-338631653936&gad_source=1&gclid=EAIaIQobChMIlai8rNa8hAMVN5eDBx2lPATuEAQYAiABEgJJV_D_BwE&gclsrc=aw.ds. Accessed 22nd April 2024.

[42] H. Elsayed, M. Weigel, F. Müller, M. Schmitz, K. Marky, S. Günther, J. Riemann, and M. Mühlhäuser, “Vibromap: Understanding the spacing of vibrotactile actuators across the body,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, pp. 1–16, 12 2020.

[43] V. Shah, M. Casadio, R. Scheidt, and L. Mrotek, “Vibration propagation on the skin of the arm,” *Applied Sciences*, vol. 9, p. 4329, 10 2019.

[44] M. W. Taylor, J. L. Taylor, and T. Seizova-Cajic, “Muscle vibration-induced illusions: Review of contributing factors, taxonomy of illusions and user’s guide,” *Multisensory Research*, vol. 30, no. 1, pp. 25 – 63, 2017.

[45] R. Nederland, “N-channel mosfet, 500 ma, 60 v, 3-pin to-92 onsemi bs170.” https://nl.rs-online.com/web/p/mosfets/1241745?cm_mmc=NL-PPC-DS3A-_-google-_-DSA_NL_EN_CatchAll_Index-_-Catch+All-_-DYNAMIC+SEARCH+ADS&matchtype=&dsa-19959388920&gad_source=1&gclid=EAIaIQobChMIk-am_8fQhgMVxIuDBx0Y6AH4EAAYASAAEgI_B_D_BwE&gclsrc=aw.ds. Accessed 22nd April 2024.

[46] G. G. Poyraz and Tamer, “Different haptic senses with multiple vibration motors,” in *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 870–874, 2019.

[47] R. Nederland, “Adaptive peltier module, 5.1w, 2.2a, 3.8v, 13 x 13mm.” https://nl.rs-online.com/web/p/peltier-modules/4901250. Accessed 22nd April 2024.

[48] R. Nederland, “Adaptive peltier module, 1.6w, 3.2a, 0.9v, 8 x 8mm.” https://eg.rsdelivers.com/product/adaptive/et-007-10-15-rs/adaptive-peltier-module-16w-32a-09v-8-x-8mm/4901200. Accessed 22nd April 2024.

[49] Bits & Parts, “Motor driver module l9110s dual h-bridge.” https://www.bitsandparts.nl/Motor-driver-module-L9110S-Dual-H-bridge-p1907161. Accessed 22nd April 2024.

[50] Arduino, “Arduino uno rev3.” https://store.arduino.cc/products/arduino-uno-rev3. Accessed 22nd April 2024.

[51] Arduino, “Arduino nano 33 ble.” https://store.arduino.cc/products/arduino-nano-33-ble. Accessed 22nd April 2024.

[52] Y. Zhang, “Knee game haptic.” https://git.lwp.rug.nl/medtech_dtpa/knee_game_haptic.git. Downloaded 3rd July 2024.

[53] SAMCOM, “Delsys trigno emg-system.” `https://samcon.be/Bewegingsanalyse/EMG-Elektromyografie/Delsys-Trigno-EMG-System?gad_source=1&gclid=EAIaIQobChMIx_atu82FhwMVE2hBAh3cdg-zEAAYASAAEgKMYfD_BwE`. Accessed 22nd April 2024.

[54] Microsoft, “Azure kinect dk documentation.” `https://learn.microsoft.com/en-us/azure/kinect-dk/`. Accessed 22nd April 2024.

[55] Özen, K. Buetler, and L. Marchal-Crespo, “Towards functional robotic training: motor learning of dynamic tasks is enhanced by haptic rendering but hampered by arm weight support,” *Journal of NeuroEngineering and Rehabilitation*, vol. 19, 02 2022.

[56] C. Jewitt, S. Price, K. Leder Mackley, N. Yiannoutsou, and D. Atkinson, *Digital Touch Ethics and Values*, pp. 107–122. 12 2019.

[57] K.-M. Behr, A. Nosper, C. Klimmt, and T. Hartmann, “Some practical considerations of ethical issues in vr research,” *Presence*, vol. 14, pp. 668–676, 12 2005.

[58] J. Colman and P. Gnanayutham, “Ethical considerations when using video games as therapeutic tools,” 2010.

[59] N. Barrow and P. Haggard, “The ethics of digital touch (1st edition).” manuscript.

A Arduino Code for pwm control of 6 motors

```
const int motor1_pin=12;
const int motor2_pin=11;
const int motor3_pin=10;
const int motor4_pin=9;
const int motor5_pin=8;
const int motor6_pin=3;

int state=0;

void setup() {
  // set up motor pins as output and start serial communication
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0 && Serial.available() <= 255) { // Check if data is available to read
    String input = Serial.readStringUntil(’\n’); // Read the input from the serial buffer until newline character

    state = input.toInt(); // Convert the input string to an integer
    if (state != 0 || input == "0") { // Check if conversion was successful
      Serial.print("You entered: ");
      Serial.println(state); // Print the entered state
    }
    else {
      Serial.println("Invalid input. Please enter a valid number."); // Print error message for invalid input
    }
  }

  // Turn motors on to the desired state and off periodically
  analogWrite(motor1_pin, state);
  analogWrite(motor2_pin, state);
  analogWrite(motor3_pin, state);
  analogWrite(motor4_pin, state);
  analogWrite(motor5_pin, state);
  analogWrite(motor6_pin, state);
  delay(250);
  analogWrite(motor1_pin, 0);
  analogWrite(motor2_pin, 0);
  analogWrite(motor3_pin, 0);
  analogWrite(motor4_pin, 0);
  analogWrite(motor5_pin, 0);
  analogWrite(motor6_pin, 0);
  delay(250);

}
```

B Arduino Code for 5 modes of constant ’on’ vibration intervals of 8 motors

```
// Define pins for motors
const int motor1_pin = 13;
const int motor2_pin = 12;
const int motor3_pin = 10;
const int motor4_pin = 8;
const int motor5_pin = 7;
const int motor6_pin = 5;
const int motor7_pin = 3;
const int motor8_pin = 2;

// Initialize mode and timing variables
int mode = 0;
unsigned long previousMillis = 0;
unsigned long target_length = 200; //Maximum length of ’interval_on’
unsigned long interval_on = 0; //Time (in ms) when motor is on, will change depending on mode
unsigned long interval_off = 200; //Time (in ms) when motor is off, will be constant
unsigned long interval = interval_on; //Variable used for timing
int state=LOW;

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  // Set motor pins as outputs
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Check if there is any data available to read from serial
    char incomingByte = Serial.read();

    // Update mode based on data sent through serial
    if (incomingByte == ’a’) {
      mode = 0;
    }else if (incomingByte == ’b’) {
      mode = 1;
    }else if (incomingByte == ’c’) {
      mode = 2;
    }else if (incomingByte == ’d’) {
      mode = 3;
    }
    else if (incomingByte == ’e’) {
```

```
      mode = 4;
    }
    // Calculate length of 'interval_on' based on mode, gets longer at higher mode values
    interval_on = target_length*mode/4;
  }

  // Get the current time in milliseconds
  unsigned long currentMillis = millis();

  // Check if the time interval has passed
  if (currentMillis - previousMillis >= interval) {
    // Save the time at which interval has passed
    previousMillis = currentMillis;

    // Toggle state and set next interval, so that motors turn on and off periodically
    if (state == LOW){
      state = HIGH;
      interval = interval_on;
    }else if (state == HIGH){
      state = LOW;
      interval = interval_off;
    }

    // Set all motor pins to the new state
    digitalWrite(motor1_pin, state);
    digitalWrite(motor2_pin, state);
    digitalWrite(motor3_pin, state);
    digitalWrite(motor4_pin, state);
    digitalWrite(motor5_pin, state);
    digitalWrite(motor6_pin, state);
    digitalWrite(motor7_pin, state);
    digitalWrite(motor8_pin, state);
  }
}
```

C Arduino Code for 5 modes of constant ’off’ vibration intervals of 8 motors

```
// Define pins for motors
const int motor1_pin = 13;
const int motor2_pin = 12;
const int motor3_pin = 10;
const int motor4_pin = 8;
const int motor5_pin = 7;
const int motor6_pin = 5;
const int motor7_pin = 3;
const int motor8_pin = 2;

// Initialize mode and timing variables
int mode = 0;
unsigned long previousMillis = 0;
unsigned long target_delay = 250; //Maximum length of ’interval_off’
unsigned long interval_on = 200; //Time (in ms) when motor is on, will be constant
unsigned long interval_off = 0; //Time (in ms) when motor is off, will change depending on mode
unsigned long interval = interval_on; //Variable used for timing
int state=LOW;

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  // Set motor pins as outputs
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Check if there is any data available to read from serial
    char incomingByte = Serial.read();

    // Update mode based on data sent through serial
    if (incomingByte == ’a’) {
      mode = 0;
    }else if (incomingByte == ’b’) {
      mode = 1;
    }else if (incomingByte == ’c’) {
      mode = 2;
    }else if (incomingByte == ’d’) {
      mode = 3;
    }
    else if (incomingByte == ’e’) {
```

```
      mode = 4;
    }
     // Calculate length of 'interval_off' based on mode, gets shorter at higher mode values
    interval_off = target_delay*(4-mode)/4;
  }

  // Get the current time in milliseconds
  unsigned long currentMillis = millis();

  // Check if the time interval has passed
  if (currentMillis - previousMillis >= interval) {
    // Save the time at which interval has passed
    previousMillis = currentMillis;

    // Toggle state and set next interval, so that motors turn on and off periodically
    if (state == LOW){
      state = HIGH;
      interval = interval_on;
    }else if (state == HIGH){
      state = LOW;
      interval = interval_off;
    }

    // Set all motor pins to the new state
    digitalWrite(motor1_pin, state);
    digitalWrite(motor2_pin, state);
    digitalWrite(motor3_pin, state);
    digitalWrite(motor4_pin, state);
    digitalWrite(motor5_pin, state);
    digitalWrite(motor6_pin, state);
    digitalWrite(motor7_pin, state);
    digitalWrite(motor8_pin, state);
  }
}
```

D Arduino Code for 5 modes of constant ’on+off’ vibration intervals of 8 motors

```
// Define pins for motors
const int motor1_pin = 13;
const int motor2_pin = 12;
const int motor3_pin = 10;
const int motor4_pin = 8;
const int motor5_pin = 7;
const int motor6_pin = 5;
const int motor7_pin = 3;
const int motor8_pin = 2;

// Initialize mode and timing variables
int mode = 0;
unsigned long previousMillis = 0;
unsigned long target_interval = 200; //Length of whole interval that represents an on-off period of the motors
unsigned long interval_on = 0; //Part of the interval where the motor is on, will change depending on mode
unsigned long interval_off = target_interval; //Part of the interval where the motor is off, will change depending on mode
unsigned long interval = interval_on; //Variable used for timing
int state=LOW;

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  // Set motor pins as outputs
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Check if there is any data available to read from serial
    char incomingByte = Serial.read();

    // Update mode based on data sent through serial
    if (incomingByte == ’a’) {
      mode = 0;
    }else if (incomingByte == ’b’) {
      mode = 1;
    }else if (incomingByte == ’c’) {
      mode = 2;
    }else if (incomingByte == ’d’) {
      mode = 3;
    }
```

```
    else if (incomingByte == 'e') {
      mode = 4;
    }

    // Calculate length of 'interval_on' and 'interval_off' based on mode
    // Longer interval_on and shorter interval_off as mode gets higher value
    interval_on = target_interval*mode/4;
    interval_off = target_interval*(4-mode)/4;
  }

  // Get the current time in milliseconds
  unsigned long currentMillis = millis();

  // Check if the time interval has passed
  if (currentMillis - previousMillis >= interval) {
    // Save the time at which interval has passed
    previousMillis = currentMillis;

    // Toggle state and set next interval, so that motors turn on and off
   periodically
    if (state == LOW){
      state = HIGH;
      interval = interval_on;
    }else if (state == HIGH){
      state = LOW;
      interval = interval_off;
    }

    // Set all motor pins to the new state
    digitalWrite(motor1_pin, state);
    digitalWrite(motor2_pin, state);
    digitalWrite(motor3_pin, state);
    digitalWrite(motor4_pin, state);
    digitalWrite(motor5_pin, state);
    digitalWrite(motor6_pin, state);
    digitalWrite(motor7_pin, state);
    digitalWrite(motor8_pin, state);
  }
}
```

E Arduino Code for 5 modes of constant vibration of 8 motors at different amplitudes

```
// Define pins for motors
const int motor1_pin = 13;
const int motor2_pin = 12;
const int motor3_pin = 10;
const int motor4_pin = 8;
const int motor5_pin = 7;
const int motor6_pin = 5;
const int motor7_pin = 3;
const int motor8_pin = 2;

// Initialize mode and timing variables
int mode = 0;
unsigned long previousMillis = 0;
unsigned long interval = 150; //Time interval (in ms) for motors to be on or off
int state = 0; //State of the motors for pwm, will take values 0-255
int target_state = 0; //Variable for intensity at which motors vibrate when on, will change depending on mode

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  // Set motor pins as outputs
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Check if there is any data available to read from serial
    char incomingByte = Serial.read();

    // Update mode based on data sent through serial
    if (incomingByte == 'a') {
      mode = 0;
    }else if (incomingByte == 'b') {
      mode = 1;
    }else if (incomingByte == 'c') {
      mode = 2;
    }else if (incomingByte == 'd') {
      mode = 3;
    }
    else if (incomingByte == 'e') {
      mode = 4;
    }

```

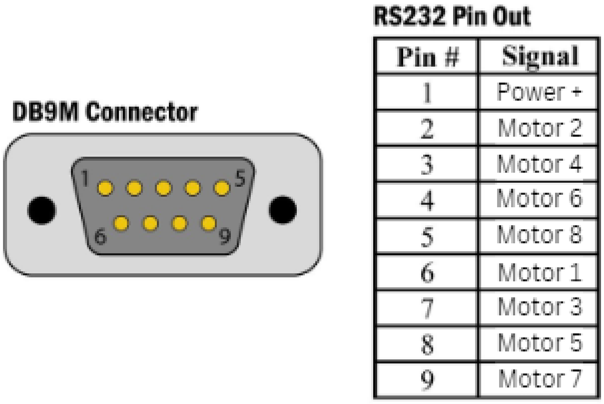
```
    //Calculate intensity at which motors will be turned on using pwm based on mode.
    //The higher the mode the higher the vibration intensity
    target_state = 255*mode/4;
  }

  // Get the current time in milliseconds
  unsigned long currentMillis = millis();

  // Check if the time interval has passed
  if (currentMillis - previousMillis >= interval) {
    // Save the time at which interval has passed
    previousMillis = currentMillis;

    // Toggle state and set next interval, so that motors turn on and off periodically
    if (state == 0){
      state = target_state;
    }else {
      state = 0;
    }

  // Set all motor pins to the new state
  analogWrite(motor1_pin, state);
  analogWrite(motor2_pin, state);
  analogWrite(motor3_pin, state);
  analogWrite(motor4_pin, state);
  analogWrite(motor5_pin, state);
  analogWrite(motor6_pin, state);
  analogWrite(motor7_pin, state);
  analogWrite(motor8_pin, state);
  }
}
```

F Arduino Code for 5 modes of periodic vibration of 8 motors at different amplitudes

```
// Define pins for motors
const int motor1_pin = 13;
const int motor2_pin = 12;
const int motor3_pin = 10;
const int motor4_pin = 8;
const int motor5_pin = 7;
const int motor6_pin = 5;
const int motor7_pin = 3;
const int motor8_pin = 2;

// Initialize mode and state
int mode = 0;
int state = 0; //State of the motors for pwm, will take values 0-255 depending
    on mode

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  // Set motor pins as outputs
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
}

void loop() {
  if (Serial.available() > 0) {
    // Check if there is any data available to read from serial
    char incomingByte = Serial.read();

    // Update mode based on data sent through serial
   if (incomingByte == 'a') {
      mode = 0;
    }else if (incomingByte == 'b') {
      mode = 1;
    }else if (incomingByte == 'c') {
      mode = 2;
    }else if (incomingByte == 'd') {
      mode = 3;
    }
    else if (incomingByte == 'e') {
      mode = 4;
    }

    //Calculate intensity at which motors will be turned on using pwm based on
   mode.
    //The higher the mode the higher the vibration intensity
```

```
    state = mode*255/4;

    // Set all motor pins to the new state
    analogWrite(motor1_pin, state);
    analogWrite(motor2_pin, state);
    analogWrite(motor3_pin, state);
    analogWrite(motor4_pin, state);
    analogWrite(motor5_pin, state);
    analogWrite(motor6_pin, state);
    analogWrite(motor7_pin, state);
    analogWrite(motor8_pin, state);
  }
}
```

G Motor pinnout in final knee sleeve design



RS232 Pin Out

Pin #	Signal
1	Power +
2	Motor 2
3	Motor 4
4	Motor 6
5	Motor 8
6	Motor 1
7	Motor 3
8	Motor 5
9	Motor 7

Figure 23: Db9 connector pinnout to motors

H Schematic of circuit for motor control with Arduino Nano BLE

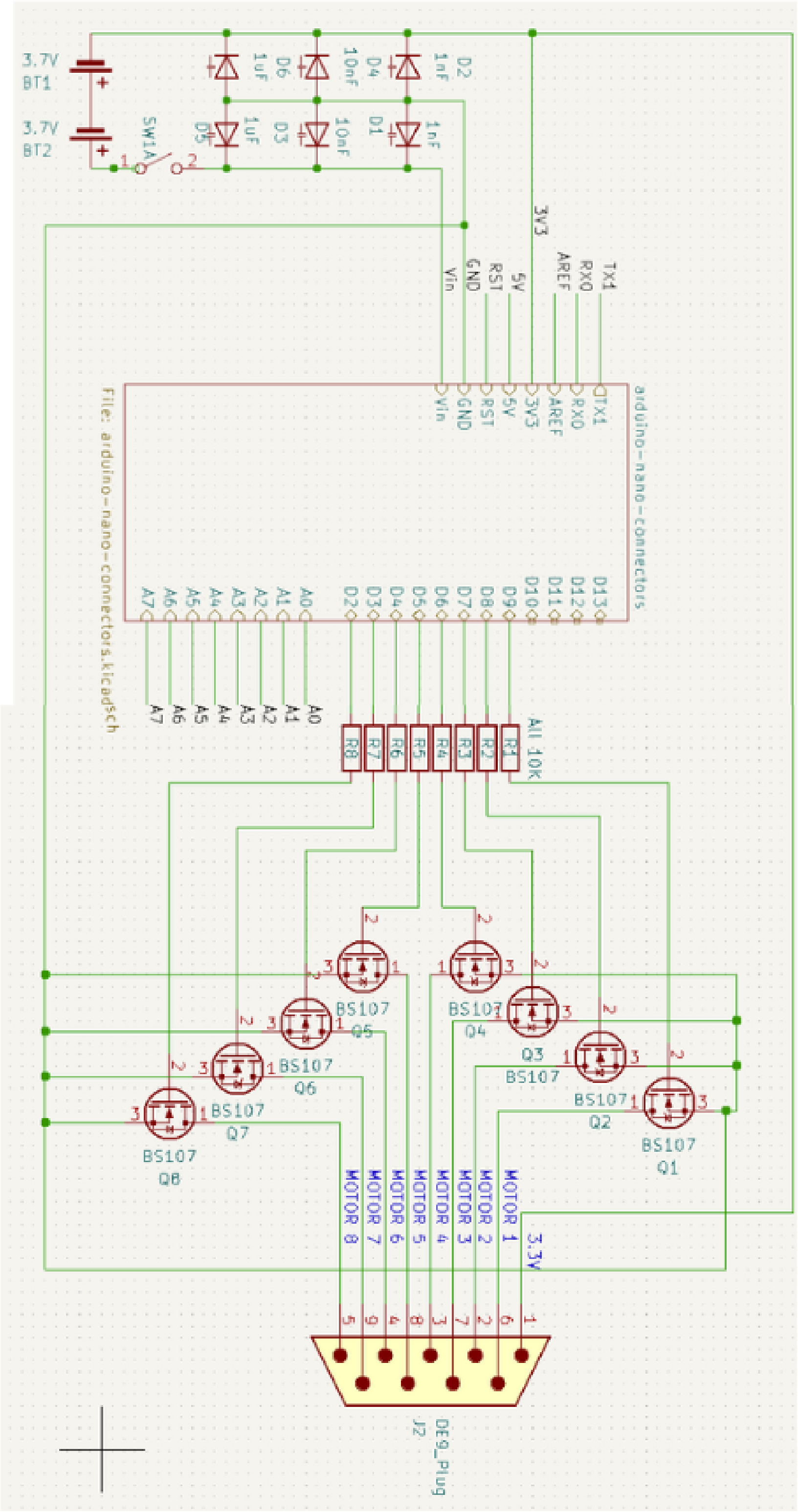


Figure 24: Final circuit schematic

I Final Motor Control Arduino Code, Slave/Peripheral device

```
#include <ArduinoBLE.h>
//Set a Bluetooth Low Energy Service with unique identifier
BLEService vibraservice("05efdb3f-2fb3-44cf-9d35-019de335f8c2"); // BLE motors Service

//Set a Bluetooth Low Energy Characteristic with custom 128-bit UUID, read and writable by central
BLEByteCharacteristic switchMode("aafabcb2-fb6b-44a4-8e0d-26fc9cab14ed", BLERead | BLEWrite);

//Set another Bluetooth Low Energy Characteristic with custom 128-bit UUID, read and writable by central
BLEByteCharacteristic test("5030a918-8485-4e81-be31-f4996a00fa69", BLERead | BLEWrite);

//Set the pins that the 8 motors are connected to
const int motor8_pin = D2;
const int motor7_pin = D3;
const int motor6_pin = D4;
const int motor5_pin = D5;
const int motor4_pin = D6;
const int motor3_pin = D7;
const int motor2_pin = D8;
const int motor1_pin = D9;

// Initialize mode, timing and state variables
int mode = 0; //Vibration mode variable, takes values 0-5 depending on value sent from central, controls how long the motors are on and off
unsigned long previousMillis = 0; //Variable used for timing, saves time at which motors change state
unsigned long target_interval = 200; //Time (in ms) of whole interval that represents an on+off period of the motors
unsigned long interval_on = 0; //Part of the interval where the motor is on, will change depending on mode. Initilised as 0, motors always off
unsigned long interval_off = target_interval; //Part of the interval where the motor is off, will change depending on mode. Initialised at max value, motors always off
unsigned long interval = interval_on; //Variable used for timing, will take value of either on or off interval to toggle between them
int state = LOW; //Variable to set state of motors, initialised as LOW, motors off

void setup() {
  Serial.begin(9600);

  // set motors’s pin to output mode
  pinMode(motor1_pin, OUTPUT);
  pinMode(motor2_pin, OUTPUT);
  pinMode(motor3_pin, OUTPUT);
  pinMode(motor4_pin, OUTPUT);
  pinMode(motor5_pin, OUTPUT);
  pinMode(motor6_pin, OUTPUT);
  pinMode(motor7_pin, OUTPUT);
  pinMode(motor8_pin, OUTPUT);
```

```

  state = LOW;
  // Turn off when disconnect
  digitalWrite(motor1_pin, state);
  digitalWrite(motor2_pin, state);
  digitalWrite(motor3_pin, state);
  digitalWrite(motor4_pin, state);
  digitalWrite(motor5_pin, state);
  digitalWrite(motor6_pin, state);
  digitalWrite(motor7_pin, state);
  digitalWrite(motor8_pin, state);

  // begin initialization
  if (!BLE.begin()) {
    Serial.println("starting Bluetooth Low Energy faimotors!");

    while (1);
  }

  // set advertised local name and service UUID:
  BLE.setLocalName("vibration");
  BLE.setAdvertisedService(vibraservice);

  // add the characteristics to the service
  vibraservice.addCharacteristic(switchMode);
  vibraservice.addCharacteristic(test);

  // add service
  BLE.addService(vibraservice);

  // set the initial value for the characteristic:
  switchMode.writeValue(0);
  test.writeValue(0);
  // start advertising
  BLE.advertise();

  Serial.println("Nano motor peripheral");
}

void loop() {
  // listen for Bluetooth Low Energy peripherals to connect:
  BLEDevice central = BLE.central();

  // if a central is connected to peripheral:
  if (central) {
    Serial.print("Connected to central: ");
    // print the central's MAC address:
    Serial.println(central.address());
    // while the central is still connected to peripheral:
    while (central.connected()) {
      // if the remote device wrote to the characteristic,
      // use the value to control the motors:
      if (test.written()){
        switch (test.value()) {
          //Upon receiving a 1x01 byte value, turn each motor on for 1000ms
    every 200ms
          case 01:
```

```
          digitalWrite(motor1_pin, HIGH);
          digitalWrite(motor2_pin, 0);
          digitalWrite(motor3_pin, 0);
          digitalWrite(motor4_pin, 0);
          digitalWrite(motor5_pin, 0);
          digitalWrite(motor6_pin, 0);
          digitalWrite(motor7_pin, 0);
          digitalWrite(motor8_pin, 0);
          delay(1000);
          digitalWrite(motor1_pin, 0);
          delay(200);
          digitalWrite(motor2_pin, HIGH);
          delay(1000);
          digitalWrite(motor2_pin, 0);
          delay(200);
          digitalWrite(motor3_pin, HIGH);
          delay(1000);
          digitalWrite(motor3_pin, 0);
          delay(200);
          digitalWrite(motor4_pin, HIGH);
          delay(1000);
          digitalWrite(motor4_pin, 0);
          delay(200);
          digitalWrite(motor5_pin, HIGH);
          delay(1000);
          digitalWrite(motor5_pin, 0);
          delay(200);
          digitalWrite(motor6_pin, HIGH);
          delay(1000);
          digitalWrite(motor6_pin, 0);
          delay(200);
          digitalWrite(motor7_pin, HIGH);
          delay(1000);
          digitalWrite(motor7_pin, 0);
          delay(200);
          digitalWrite(motor8_pin, HIGH);
          delay(1000);
          digitalWrite(motor8_pin, 0);
          delay(200);
      }
    }
    // if the remote device wrote to the characteristic,
    // use the value to control the motors:
    if (switchMode.written()) {
      switch (switchMode.value()) {    // any value other than 0
        // Change mode value depending on received byte
        case 01:
          Serial.println("mode 0");
          mode = 0;
          break;
        case 02:
          Serial.println("mode 1");
          mode = 1;
          break;
        case 03:
          Serial.println("mode 2");
          mode = 2;
```

```
            break;
          case 04:
            Serial.println("mode 3");
            mode = 3;
            break;
          case 05:
            Serial.println("mode 4");
            mode = 4;
            break;
        }
        // Calculate length of ’interval_on’ and ’interval_off’ based on mode
        // Longer interval_on and shorter interval_off as mode gets higher value
        interval_on = target_interval*mode/4;
        interval_off = target_interval*(4-mode)/4;
      }
      // Get the current time in milliseconds
      unsigned long currentMillis = millis();

  // Check if the time interval has passed
  if (currentMillis - previousMillis >= interval) {
    // Save the time at which interval has passed
    previousMillis = currentMillis;
    // Toggle state and set next interval, so that motors turn on and off periodically
    if (state == LOW){
      state = HIGH;
      interval = interval_on;
    }else if (state == HIGH){
      state = LOW;
      interval = interval_off;
    }

    // Set all motor pins to the new state
    digitalWrite(motor1_pin, state);
    digitalWrite(motor2_pin, state);
    digitalWrite(motor3_pin, state);
    digitalWrite(motor4_pin, state);
    digitalWrite(motor5_pin, state);
    digitalWrite(motor6_pin, state);
    digitalWrite(motor7_pin, state);
    digitalWrite(motor8_pin, state);
  }
  }
  // when the central disconnects, print it out:
  Serial.print(F("Disconnected from central: "));
  Serial.println(central.address());
  state=LOW;
  // when the central disconnects, turn off the motors
  digitalWrite(motor1_pin, state);
  digitalWrite(motor2_pin, state);
  digitalWrite(motor3_pin, state);
  digitalWrite(motor4_pin, state);
  digitalWrite(motor5_pin, state);
  digitalWrite(motor6_pin, state);
  digitalWrite(motor7_pin, state);
  digitalWrite(motor8_pin, state);         }
}
```

J Final Motor Control Arduino Code, Master/Central device

```
#include <ArduinoBLE.h>

void setup() {
  // Start serial communication at 115200 baud rate
  Serial.begin(115200);

  //While the board is connected through serial
  while (!Serial);

  // initialize the Bluetooth Low Energy hardware
  BLE.begin();

  Serial.println("Bluetooth Low Energy Central - motor control");

  // start scanning for peripherals with uuid matching the vibration service
  BLE.scanForUuid("05efdb3f-2fb3-44cf-9d35-019de335f8c2");
}

void loop() {
  // check if a peripheral has been discovered
  BLEDevice peripheral = BLE.available();

  if (peripheral) {
    // discovered a peripheral, print out address, local name, and advertised service
    Serial.print("Found ");
    Serial.print(peripheral.address());
    Serial.print(" '");
    Serial.print(peripheral.localName());
    Serial.print("' ");
    Serial.print(peripheral.advertisedServiceUuid());
    Serial.println();

    //Double check that the local name of the found peripheral is "Vibration"
    if (peripheral.localName() != "Vibration") {
      return;
    }

    // stop scanning if peripheral is found
    BLE.stopScan();

    // begin control of peripheral
    controlLed(peripheral);

    //if peripheral is disconnected, start scanning again
    BLE.scanForUuid("05efdb3f-2fb3-44cf-9d35-019de335f8c2");
  }
}

//Triggered when peripheral is discovered with correct local name
void controlLed(BLEDevice peripheral) {
  // connect to the peripheral
  Serial.println("Connecting ...");

```

```
  if (peripheral.connect()) {
    Serial.println("Connected");
  } else {
    //If connection is not succesful go back
    Serial.println("Failed to connect!");
    return;
  }

  // discover peripheral attributes
  Serial.println("Discovering attributes ...");
  if (peripheral.discoverAttributes()) {
    Serial.println("Attributes discovered");
  } else {
    Serial.println("Attribute discovery failed!");
    peripheral.disconnect();
    return;
  }

  // retrieve the characteristic that allows to change the vibration mode of the motors by using set uuid
  BLECharacteristic MotorModeCharacteristic = peripheral.characteristic("aafabcb2-fb6b-44a4-8e0d-26fc9cab14ed");

  // if the characteristic is not found or is not writable return
  if (!MotorModeCharacteristic) {
    Serial.println("Peripheral does not have motor mode switching characteristic!");
    peripheral.disconnect();
    return;
  } else if (!MotorModeCharacteristic.canWrite()) {
    Serial.println("Peripheral does not have a writable motor mode switching characteristic!");
    peripheral.disconnect();
    return;
  }


  while (peripheral.connected()) {
    // while the peripheral is connected

    // Check if there is any data available to read from serial
    char mode  = Serial.read();

    //Send mode value to peripheral based on character received through serial
    if (mode == 'a') {
      Serial.println("mode a, sending 1");
      // mode 'a' is read: write 0x01 to toggle mode 0 in motors
      MotorModeCharacteristic.writeValue((byte)0x01);
    } else if (mode == 'b') {
      Serial.println("mode b, sending 2");
      // mode 'b' is read: write 0x02 to toggle mode 1 in motors
      MotorModeCharacteristic.writeValue((byte)0x02);
    }else if (mode == 'c') {
      Serial.println("mode c, sending 3");
      // mode 'c' is read: write 0x03 to toggle mode 2 in motors
      MotorModeCharacteristic.writeValue((byte)0x03);
    }else if (mode == 'd') {
```

```
      Serial.println("mode d, sending 4");
      // mode ’d’ is read: write 0x04 to toggle mode 3 in motors
      MotorModeCharacteristic.writeValue((byte)0x04);
    }else if (mode == ’e’) {
      Serial.println("mode e, sending 5");
      // mode ’e’ is read: write 0x05 to toggle mode 4 in motors
      MotorModeCharacteristic.writeValue((byte)0x05);
    }
  }

  Serial.println("Peripheral disconnected");
}
```

K Results of Ultrahaptic tests

Frequency	Perceived?
0.5	no
4	no
0	no
8	yes, as wind
0	no
4	no
1	no
2	yes, as wind
10	yes, but no difference
1	yes, but no difference
20	yes, but no difference

Table 18: Perception of rotor sensation around knee at different rotation frequencies during first perception test

Frequency	Perceived?
10	yes
8	yes
10	no
4	no
8	yes
6	yes
4	yes
6	yes
4	no
2	no
8	yes

Table 19: Perception of rotor sensation around knee at different rotation frequencies during second perception test where rotation was turned off between each new frequency

Frequency	Perceived?
8	yes
10	yes, but no difference
6	yes, but no difference
2	yes, but no difference
4	yes, but no difference
10	yes, but no difference
8	yes, but no difference
2	yes, but no difference
6	yes, but no difference

Table 20: Perception of rotor sensation around knee at different rotation frequencies during third perception test

Frequency	Perceived difference?
2-10-2	more wind
2-6-2	no change
4-10-4	more wind
4-8-4	no change
6-10-6	no change
6-8-6	no change
8-10-8	no change

Table 21: Perception of rotor sensation around knee at different rotation frequencies during final perception test