Dissertation Type: enterprise

DEPARTMENT OF COMPUTER SCIENCE

Developing a system for improving slalom skiers' technique through real-time vibrotactile feedback

Louis Wyborn

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Engineering in the Faculty of Engineering.

Friday 10th May, 2019

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of MEng in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

[illegible]

Louis Wyborn, Friday 10th May, 2019

Contents

Executive Summary

In this project I describe the design, development and evaluation of SlalomTracker, a wearable device to support the training of advanced ski slalom racers. The device combines accelerometer, gyroscope and magnetometer readings to track, in real-time, whether a racer's ski follows the line of an ideal turn and if not, how much it deviates from this. The device provides ski racers with vibrotactile feedback about this deviation so that they can correct their line - using either vibrotactile feedback on their legs or auditory signals through earphones. Evaluation with expert skiers in the wild indicates that vibrotactile feedback is less cognitively demanding than auditory feedback and that SlalomTracker is a useful training device.

The motivation for this project stemmed from the fact that current methods of ski coaching, including video analysis and verbal feedback from coaches, typically do not give accurate and timely enough feedback to make the minute adjustments necessary to improve technique. Other studies [32] [28] have identified these as issues and built systems that aim to alleviate them. Their results have shown that real-time auditory feedback is perceived to be useful by high level ski racers.

SlalomTracker was developed through an iterative autobiographical design process, with the first five prototypes being evaluated on dry ski slopes and the final device tested in the Alps under racing conditions.

The user evaluation consisted of a mixed methods approach, with both qualitative and quantitative data being collected to evaluate the device with international level ski racers. Using a questionnaire and data recorded from the device, I compared slalom runs performed with no feedback, auditory feedback, and vibrotactile feedback. In the wild evaluation indicates that vibrotactile feedback adds less cognitive load than auditory feedback in this task and is perceived to increase performance over both no feedback and auditory feedback.

The device provides a useful measure of ski edge slippage around turns, isolating the dependent variable out of a massive parameter space, including body weight, height, ski length, ski flexibility, ski side-cut radius, snow conditions, and ski edge sharpness.

The main contributions of this project are as follows:

- I designed and built circuits to combine accelerometer and gyroscope readings, and control motors with PWM signals.
- I modelled and 3D printed housing for these circuits so they could be incorporated into ski boots.
- I implemented a complementary data fusion filter in C and Python.
- I designed and implemented an algorithm to measure ski edge slippage in real time.
- I wrote an ExpressJS server and ReactJS web application to serve as a user interface used to control the device wirelessly from the end user's phone or laptop.
- I conducted a mixed methods user evaluation of the device in the French Alps under racing conditions, involving international level athletes.
- I found promising results that indicate that vibrotactile signals are better than auditory ones for providing real time feedback on ski slippage in ski racing conditions.
- I found that SlalomTracker is perceived as a useful training device by expert ski racers.

Supporting Technologies

I used several third-party software and hardware resources n order to implement my project. These have been listed below, and will be referred to throughout this report.

- I used several Adafruit sensors, including the MMA8451 [21], L3GD20H [20], and BNO055 [10] breakout boards to obtain measurements of orientation and acceleration.
- I used a Raspberry Pi to connect the sensors together, process the data, and control the audio and vibration feedback.
- I used the relevant CircuitPython [13] libraries on the Raspberry Pi to control the I2C sensors.
- The web-interface component of my system was implemented using an ExpressJS [15] web server serving up a ReactJS [23] frontend. Bootstrap [11] was used for styling.
- I used an Arduino Uno to rapidly prototype ideas and test components of the system throughout the process.
- I used Autodesk Maya [9] to design several parts of the device.
- I used an Ultimaker 2+ [25] and the associated Cura software [26] to 3D print the CAD parts.
- I used SoX [24] to play the audio cues necessary.
- I used dnsmasq [14] to provide a DHCP server on the Raspberry Pi.
- I used hostapd [19] to allow the Raspberry Pi to act as a wireless access point.
- I used Git [16] to version control the code for my project.

Notation and Acronyms

API	:	Application Programming Interface
AUC	:	Area Under Curve
CSS	:	Cascading Style Sheets
DHCP	:	Dynamic Host Configuration Protocol
EEPROM	:	Electrically Erasable Programmable Read-Only Memory
EMF	:	Electromotive Force
GPS	:	Global Positioning System
I2C	:	Inter-Integrated Circuit
IMU	:	Inertial Measurement Unit
NASA TLX	:	National Aeronautics and Space Agency Task Load Index
ONT	:	Optical Navigation Technology
PWM	:	Pulse-width Modulation
SDRAM	:	Synchronous Dynamic Random-Access Memory
SI	:	Système international (d'unités)
SRAM	:	Static Random-Access Memory
VNC	:	Virtual Network Computing

Chapter 1

Background

In this chapter I introduce the project topic of ski edge slippage in slalom racing, explaining the technical background required to understand it, the importance of the specific problem involved, the existing work done in the field and the limitations of this work, and the central challenges involved in the project.

1.1 Ski Mechanics

As a prerequisite to understanding the problem this project focusses on, the mechanics of ski turns, the forces involved, and the rules for modern slalom racing must be understood. This section explains each of these in turn and links them to the problem at hand.

1.1.1 Carving Turns

Carving turns in skiing are a particular form of turn used by more advanced skiers to maintain speed as they change direction.

It works by leaning the skis onto their edges, so they cut into the snow. The edges on modern carving skis are curved in an arc, so when on their side the ski cuts into the snow along the curve of the arc. As the ski travels forward it follows this curve, causing it to turn. This is the most efficient way to turn, as there is minimal friction with the snow. In addition, it is possible to achieve a much higher reaction force from the snow when compared to parallel turning, as the edges grip the slope.

As the ski is leant onto its edge it will grip the snow as soon as it reaches the minimum carving angle. This is the minimum angle required for the edges to engage with the snow, and depends on snow conditions and the measurements of the ski. At this minimum carving angle the ski has not bent significantly yet, so will follow a curve with the same radius as the sidecut radius. This radius is the maximum carving radius, and turning in a radius greater than this requires sliding the ski sideways, rather than a true carve.

As the ski is leant further onto its edge it will bend more, cutting into the snow along an arc with a smaller radius, leading to a tighter turn. The radius of the turn can be controlled by varying the edge angle, and any turn radius within the 'carve zone' shown in Figure 1.2 can be achieved while keeping the

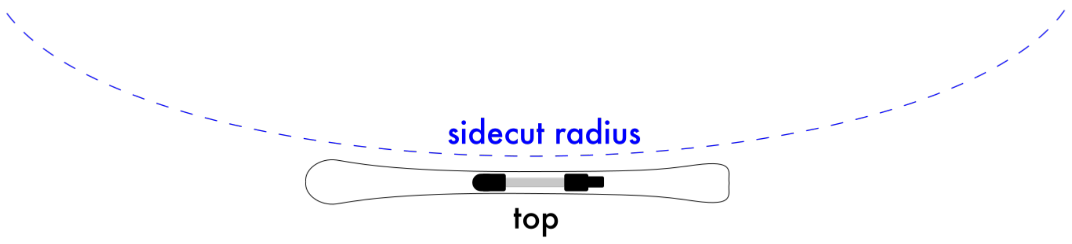


Figure 1.1: Curve of ski edge forms part of a circle defined by the sidecut radius

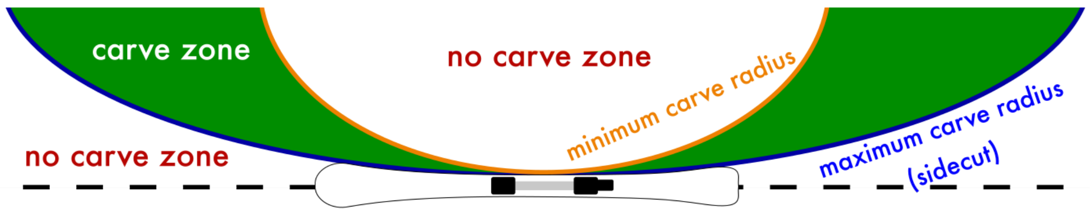


Figure 1.2: Maximum and minimum carving radii outline the 'carve zone'

skis in a true carving turn. The radius of the ideal turning curve, one where no slippage is occurring, is described in equation 1.1.

$$R = r \cdot cos(\theta) \tag{1.1}$$

where R is the ideal turning curve radius, r is the sidecut radius, and θ is the edge angle of the ski, as shown in Figure 1.3. This is a theoretical calculation assuming no slippage, a ski that does not twist along its length under strain, and perfectly smooth snow, however serves as a reasonable approximation for this project, where later calibration can account for differences.

At a certain point the ski will reach the limit of its flexibility, and will not bend further. Increasing the edge angle by leaning the ski more will cause the midpoint of the ski underneath the boot to lift off the snow, losing grip and typically causing the ski to slide sideways. Therefore the carve zone varies between skis depending on their sidecut radius and flexibility. The sharpness of the ski edge in combination with the slope conditions also have an effect, as a dull edge may struggle to engage with an icy slope and require a greater edge angle before reaching the minimum carving angle.

Slalom skis are designed with a smaller radius to perform short, tight turns; while giant slalom or super-g skis are designed with a larger radius to perform larger turns. Figure 1.1 is seen from a top-down viewpoint. When the ski is meets the minimum carving radius by slightly leaning on its edge it will follow the same arc as the sidecut of the ski.

1.1.2 Ski Edge Slippage

As skiers travel at faster velocities, more force is required to change direction. This force comes from the reaction with the snow, and greater reaction force can be achieved by increasing the angle the ski's edges make with the snow. The exact amount of force achieved varies depending on many factors, including the condition of the snow and the sharpness of the ski's edges. If the skier overcomes this reaction force, the ski will slip sideways, expending energy to move the snow. This results in the skier losing speed, so must be avoided for an optimal run. Figure 1.3 shows the forces involved.

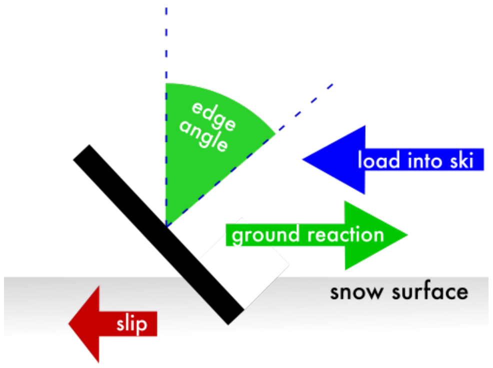


Figure 1.3: Forces involved in a carving turn

The slip shown in this diagram is a key factor in performance of high level ski racers, and reducing it is often a differentiating factor between top racers. Lower total magnitudes of slip both give the racer more control and reduce speed lost around turns. The time differences between fastest times is often tiny, with less than a second separating podium positions [1], so small differences in slip can mean the difference between winning and losing. This is why I chose to develop this system to focus on these small differences in slip.

1.1.3 Slalom Racing Rules

A slalom ski course is made up of between 40 and 75 gates. Each gate is a pair of poles, which are either red or blue. The poles are made of hard plastic and hinged at the base, such that they can bend out of the way if contact is made with a skier. The skier has to clear all the gates by passing through each pair of poles. Generally, the aim is to pass as close as possible to each gate to minimise the change of direction needed to clear it. Clearing a gate is defined as having the skis and boots pass between the poles, with no enforced standard as to the movement of the rest of the body.

This component of the rules leads to the modern slalom racing technique, where the skier begins the turn above the inside pole of the gate, ensuring that by the time they get to the pole their body is leant over, and their centre of mass is on the outside of the gate. The skier can then punch the gate out of the way with one of their hands, so it does not hit their body and throw them off balance. As shown in figure 1.4 the skier's centre of mass may not pass through any of the gates if the skier leans out enough.

1.2 Current Problem

Common methods of ski coaching, including video analysis and coaches' verbal instructions, often do not give accurate enough feedback to make the minute adjustments necessary to improve technique, and only give feedback after the activity is completed. Having personally experienced these issues I was motivated to develop SlalomTracker, a system to provide real-time feedback on ski slippage to expert slalom racers.

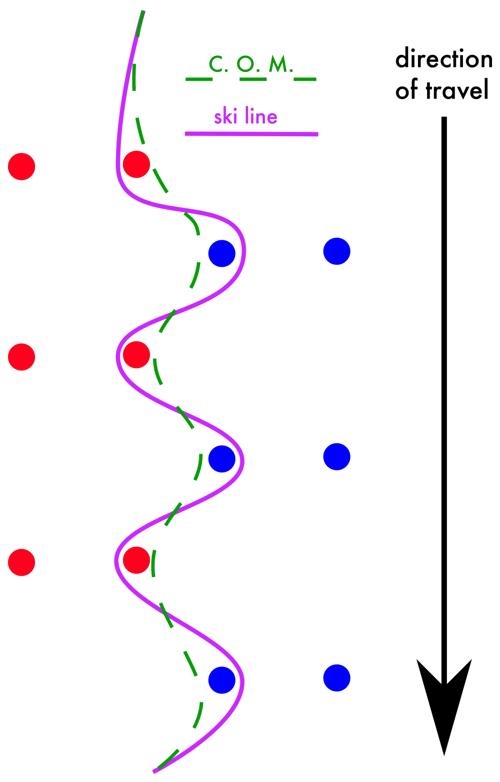


Figure 1.4: Line of travel of skis and centre of mass (C.O.M.)

1.3 Related Works

In this section I will explain current work done in this field, detailing how this research informs my project and in particular the limitations that I have addressed.

1.3.1 Fusion Motion Capture

Brodie et al. [28] pioneered the use of IMUs (Inertial Measurement Units, combining accelerometer, gyroscope and magnetometer sensors) to capture the motion of ski racers. Previous biomechanical research focused on the analysis of only a few gates out of the whole race course. Previous research had noted that turn performance is dependent on previous turns. Therefore, this paper hypothesises that if previous and future turns play a role in turn strategy, the entire run needs to be examined to analyse athlete performance, leading Brodie et al. to focus on this whole-run analysis.

The main objectives of the study were to build a 3D kinematic model of a ski racer completing a giant slalom course, and then to analyse this model in order to provide insights into a ski racer's posture that could improve their race times.

Their system combined data from IMUs, GPS, and pressure sensors to ascertain the position and movement of a ski racer down a course. The authors mounted 13 IMUs across the athlete's body, including one on each boot and each pole, and one on top of the helmet. They inserted pressure sensitive insoles into each boot under the heel and attached a GPS tracker to the helmet.

The authors used a custom data fusion algorithm to take the raw data from each sensor and apply the motion measured at each point on the athlete's body to segments of a rigid body simulation. This also applies positional and velocity data from the GPS receiver, corroborated with pre-calculated positional data of each gate obtained using a theodolite. The pressure measured in each boot is used to calculate the relative loading of each ski, and therefore the reaction force with the ground, which is applied to the legs of the rigid body simulation.

This was sufficient to produce a model with a maximum error in the trajectory of the skier of only ±1.5m over a 300m course, and a maximum error in the limb orientation of less than 5 degrees.

The paper found it was possible to make key insights into specifics of a racer's technique, such as when they observe one particular turn across two runs. They note that "the athlete uses a similar leg/snow angle through turn 6 in both runs" [28, p.27] however in one of these runs "during the entry phase of turn 6, the athlete had less pressure on the skis, resulting in a high leg/snow angle and also ski edge slippage". Their ability to analyse ski edge slippage suggested that it would be possible to implement a system to measure this using only IMUs.

The paper does not examine whether or not this type of feedback is perceived to be helpful to ski racers, although intuitively this appears to be the case.

The system detailed in the paper provides extremely detailed information about a ski racer's technique and the forces acting on them, however this is only possible through prior in-depth analysis of a ski run, measuring the precise locations of gates. A GPS base station needs to be set up near the course in order to ensure accurate relative GPS measurements, and a GPS receiver attached to the helmet. The skier also needs to wear 13 IMU sensors taped to their skin through a custom lycra bodysuit with apertures cut out for each sensor. This is prohibitively cumbersome and expensive to implement.

The analysis of the run can also only happen after each run is complete. The paper explores the possibility of future work to allow virtual animations of an optimum run to be examined before a race, or a real-time acoustic feedback system to allow a skier to hear positive or negative accelerations, however does not implement these.

1.3.2 vLink

A later paper published by Kirby (2009) [32] details the building of a system to provide real-time feedback about ski slippage. This paper was used to develop the vLink device that was sold by Advanced Racing Computers. The main objectives of this were to build a system that could measure ski edge slippage despite not having the course mapped out, and to provide real-time feedback to ski racers about slippage. They decided against using IMUs and GPS as per Brodie et al. [28], in favour of a custom optical navigation technology (ONT) system. This type of sensor technology was originally developed for use in computer mice. The system takes images of a surface at a high frequency, processes them to detect patterns in the images, then identifies common patterns in successive images and determines a shift in those patterns. From this shift an accurate X and Y displacement can be found between the sensor and the surface it is placed on. The resolution of an off-the-shelf ONT system can be as low as 0.05mm, with a polling frequency of up to 6500Hz, far surpassing the precision and polling frequency of GPS systems.

Off-the-shelf systems generally are only able to measure up to a maximum speed of 3.9m/s, whereas competitive skiers can regularly exceed 50m/s, so the author redesigned the illumination and imaging system of the standard sensor to support a maximum speed of 50m/s over snow. This was achieved by increasing the frame size by 10x along each axis, so each image covered an area of 18x18mm, up from 1.8x1.8mm, and helped by the increased roughness of snow's texture compared to a computer mouse mat. The device was attached directly to the ski with strong adhesive so it could accurately record any movements the ski made. In order to keep the glass window that the images are taken through clear of moisture condensation, the authors had to seal the device with low humidity nitrogen gas and include silica gel packets.

The device provided auditory feedback through a pair of earphones with beeps in the right ear indicating right ski slippage, and beeps in the left ear indicating left ski slippage. The sensitivity of the device was adjustable, each beep signifying 0.5mm of slippage with the most sensitive setting.

The device was tested with 12 racers participating in a summer race camp, first by allowing the racers to become familiar with the device through a series of familiarisation drills, then training for 2 hours using the system. The racers each filled out a survey after their training and were interviewed for subjective comments. The paper reported that "83% of participants stated that real-time audible feedback of their lateral displacement [slippage] definitely helped them to better understand their carving skills. In addition, 50% of participants stated that the real-time feedback definitely helped them improve their carving skills on the first day they used the system" [32, p.43]. These results show that form of feedback about slippage is perceived to be useful by high level ski racers, and subjective comments made in interviews indicated that this allowed them to understand and work on aspects of their technique better. However, there were several comments regarding the audio feedback being distracting, which the paper hypothesises could be reduced by "a longer, more in-depth and more controlled familiarisation process" [32, p.52].

This paper shows promising qualitative results, however does not evaluate any quantitative improvement in performance. This is an area I aim to explore in my project. In addition, the device detailed in this paper is prohibitively expensive due to the custom sensors, with the eventual retail price being $750. SlalomTracker is built from off-the-shelf components to reduce cost.

1.3.3 MusicJacket

The MusicJacket paper by van der Linden, et al. (2010) [35] describes the iterative development of a wearable system to support the teaching of novice violin players. The objective of the study was to investigate the effectiveness of vibrotactile feedback for teaching correct posture and basic bowing skills. The expectation was that the vibrotactile feedback would place the violin players under less cognitive load than real-time visual or auditory feedback. The final system utilises an IMU-based motion capture suit to model the movement and 3D position of the violin players upper body and provides vibrotactile feedback to correct posture and bowing using vibration motors positioned on certain areas of the body.

The user study described in the paper found that this vibrotactile feedback was effective at improving novice's bowing technique, and half of the participants continued to show improved technique even when no longer receiving vibrotactile feedback. This was a marked improvement over the control subjects, none of whom showed a comparable improvement with the same number of training sessions.

1.3.4 Related Work Conclusion

Since the publication of these papers IMUs have become much cheaper, more standardised, physically smaller, and more accurate. This improvement in technology along with the results of these three studies led me to conclude that a system to provide real-time feedback about turn slippage in ski racing could be extremely beneficial to racers and would be feasible to achieve using cheap off-the-shelf IMUs. The ability to record data in a ski run for later analysis would also be beneficial for coaches to provide more in-depth feedback to ski racers.

It was unclear from the related work whether vibrotactile or auditory would be the best modality for delivering feedback to

1.4 Central Challenges and Objectives

The central objectives of this project are:

- Design a system that can both:
  - Measure and record ski edge slippage.
  - Provide real-time vibrotactile and auditory feedback on this ski edge slippage.
- Perform a user evaluation to examine if there is a difference in the efficacy of vibrotactile and auditory feedback for expert ski racers.
- Evaluate any performance gains achieved through the use of the system.

To achieve these objectives the central challenges are:

- Extreme conditions present on ski slopes including:

  - Rapid temperate changes from +20°C to -20°C.
  - Snow melting and refreezing on the ski and boot, repeatedly soaking and freezing equipment.
- Ski races being performed at high speed, up to 50m/s, with the shortest turns performed in under 1 second.

The next chapter describes how these challenges were addressed through an iterative design process.

Chapter 2

Project Execution

This project utilised autobiographical design [33] to iteratively design and build several prototypes that moved towards a final design with the following requirements.

- Ability to measure and record ski edge slippage.
- Ability to provide feedback in real time to users.
- Use off-the-shelf parts.
- Easy to put on and take off.
- Resistant to extreme conditions present on ski slopes.

The autobiographical design process enabled a much tighter design cycle, from user input to implementation, than a traditional usability evaluation conducted on a range of users at each step of the design cycle. I am an experienced ski racer and so was able to gain useful feedback from testing the prototypes on myself.

2.1 Prototype I

The ability to measure the slip was crucial to this project's success, so this was the first area worked on. I needed to test if it was possible to measure the orientation of the device using a gyroscope and an accelerometer. I purchased Adafruit's MMA8451 Accelerometer Breakout [21], and L3GD20H Gyroscope Breakout Boards [20] in order to build the initial prototype. I connected them together using a breadboard [12] and Arduino Uno [8], using the I2C protocol [34] to read data from the boards.

2.1.1 Data Fusion

Both the gyroscope and accelerometer can be used to estimate the orientation of the device, however it is often necessary to combine these estimates into one overall estimate with greater accuracy. This process is called Data Fusion [30]. I will first discuss estimating the orientation with each of the two sensors separately, then discuss the method used to combine the estimates.

2.1.1.1 Gyroscope

A gyroscope measures angular rate of change around each of its axes. Assuming the data output was free of noise, we could integrate the rate of change around each axis over time and obtain the orientation of the device relative to its starting position. However, all gyroscopes have some degree of noise present in the readings, so simply integrating over time will continually add small errors, accumulating until the readings do not match up to the real world. This accumulation of error is known as drift and is shown in Figure 2.1 where the orange line representing the estimate of pitch angle drifts away from the true pitch angle over time.

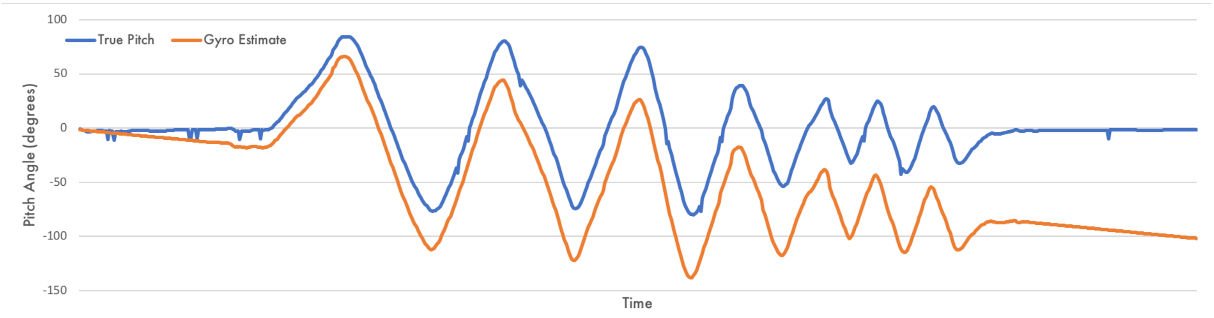


Figure 2.1: Gyroscope accumulation of errors

$$\begin{pmatrix}\dot{\phi}_g\\ \dot{\theta}_g\\ \dot{\gamma}_g\end{pmatrix} = \begin{pmatrix}1 & sin(\phi)tan(\theta) & cos(\phi)tan(\gamma)\\ 0 & cos(\phi) & -sin(r)\\ 0 & sin(\phi)sec(\theta) & cos(\phi)sec(\gamma)\end{pmatrix}\begin{pmatrix}r\\ p\\ y\end{pmatrix} \tag{2.1}$$

where

$$\phi, \theta, \gamma = \text{current } roll, pitch, yaw\ (deg)$$

$$\dot{\phi}_g, \dot{\theta}_g, \dot{\gamma}_g = roll, pitch, yaw \text{ transformed readings } (deg/s)$$

$$r, p, y = roll, pitch, yaw \text{ raw gyro readings } (deg/s)$$

Figure 2.2: Transforming rotations to the inertial reference frame.

We cannot simply low-pass filter the signal, as this will only delay the onset of drift, and add latency to the measurements. In addition, we cannot integrate over each axis independently because Euler angles in the inertial reference frame depend on more than just the rate of change around each axis in the device's reference frame. In other words, changing the angle around one axis of the device can change the angle around the other axes from an external reference frame. For example, picture the device starting from stationary then pitching up 45 degrees. From this position if the device yaws to either side 90 degrees, the pitch is decreased to 0 degrees from the point of view of the inertial reference frame, however, based on the device's reference frame, it has not rotated around the pitch axis. We therefore need to transform the rotations to the inertial reference frame before we integrate them over time, using the following equation.

The gyroscope is calibrated by placing it stationary on a flat surface, and sampling n times to find the average bias. This average bias is then subtracted from each reading of the gyroscope.

2.1.1.2 Accelerometer

An accelerometer measures acceleration along each of its axes. When an accelerometer is stationary it will only measure acceleration due to gravity, which can be used to calculate the pitch and roll of the device through basic trigonometry, by assuming it always acts straight down. It cannot estimate yaw, as the readings are symmetrical around the Z axis.

However, any force applied to the device will cause the acceleration vector to not point directly down. The raw data from the accelerometer contain high frequency additive noise, which will cause any estimates to appear to jump around. Simply adding a low pass filter to this will reduce the noise, but at the expense of attenuating any fast changes in rotation. In this project the device will be subject to large forces acting laterally, which will change the apparent direction of gravity, and fast changes of orientation, so a system that only uses accelerometer data would not be useful.

2.1.1.3 Filter

These estimates were combined using a complementary filter [27]. This type of filter was chosen for prototyping over a more complex Kalman filter due to the relative ease of implementation and computational load requirements.

$$\begin{aligned} \dot{\phi}_a &= arctan(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}) \cdot \frac{180}{\pi} \\ \dot{\theta}_a &= arctan(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}) \cdot \frac{180}{\pi} \end{aligned} \tag{2.2}$$

where

$$\dot{\phi}_a, \dot{\theta}_a = \text{new } roll, pitch \ (deg)$$

$$A_x, A_y, A_z = x, y, z \text{ raw accelerometer readings } (m/s^2)$$

Figure 2.3: Estimating orientation using trigonometry.

A complementary filter combines the desirable low-frequency characteristics of the accelerometer with the desirable high-frequency characteristics of the gyroscope. This works by choosing a constant alpha, such that $0 < \alpha < 1$, that decides how much we 'trust' the accelerometer readings. If we expect large magnitude forces to be applied regularly, we should make this number lower, if not this number should be higher. It also depends on the relative noise levels of each sensor.

We assume the device is initially stationary on a flat surface ($roll, pitch, yaw = 0$). Then each timestep (Δ t) we do the following:

1. Retrieve the raw accelerometer and gyroscope data.
2. Estimate pitch and roll from accelerometer data using Equation 2.2.
3. Transform the gyroscope data to the inertial reference frame using Equation 2.1
4. Combine the accelerometer estimate with the integral of the transformed gyroscope data using Equation 2.3 below.

$$\begin{aligned} \phi^{t+1} &= (1 - \alpha) \cdot (\phi^t + \dot{\phi}_g \cdot \Delta t) + \alpha \cdot \dot{\phi}_a \\ \theta^{t+1} &= (1 - \alpha) \cdot (\theta^t + \dot{\theta}_g \cdot \Delta t) + \alpha \cdot \dot{\theta}_a \end{aligned} \tag{2.3}$$

This effectively performs a low pass filter on the accelerometer measurements, and a high pass filter on the gyroscope measurements, then combines the results to provide a final estimate of the orientation of the device.

2.1.1.4 Arduino Code

This data fusion algorithm was implemented in C with the Arduino function set, full code shown in Appendix B.1, and run on an Arduino Uno. The current orientation was output over the serial connection and examined on a connected laptop.

2.1.1.5 Prototype I Conclusion

From this prototype I was able to conclude that it was possible to accurately measure the orientation of a device with an accelerometer and gyroscope, through the use of a complementary filter.

2.2 Prototype II

The device needs to accurately measure the orientation and acceleration being applied to each ski. It would be necessary to have a sensor on each leg, therefore it made the most sense to attach the micro-controller at the waist, equi-distant from each sensor. I planned to continue to use I2C to communicate

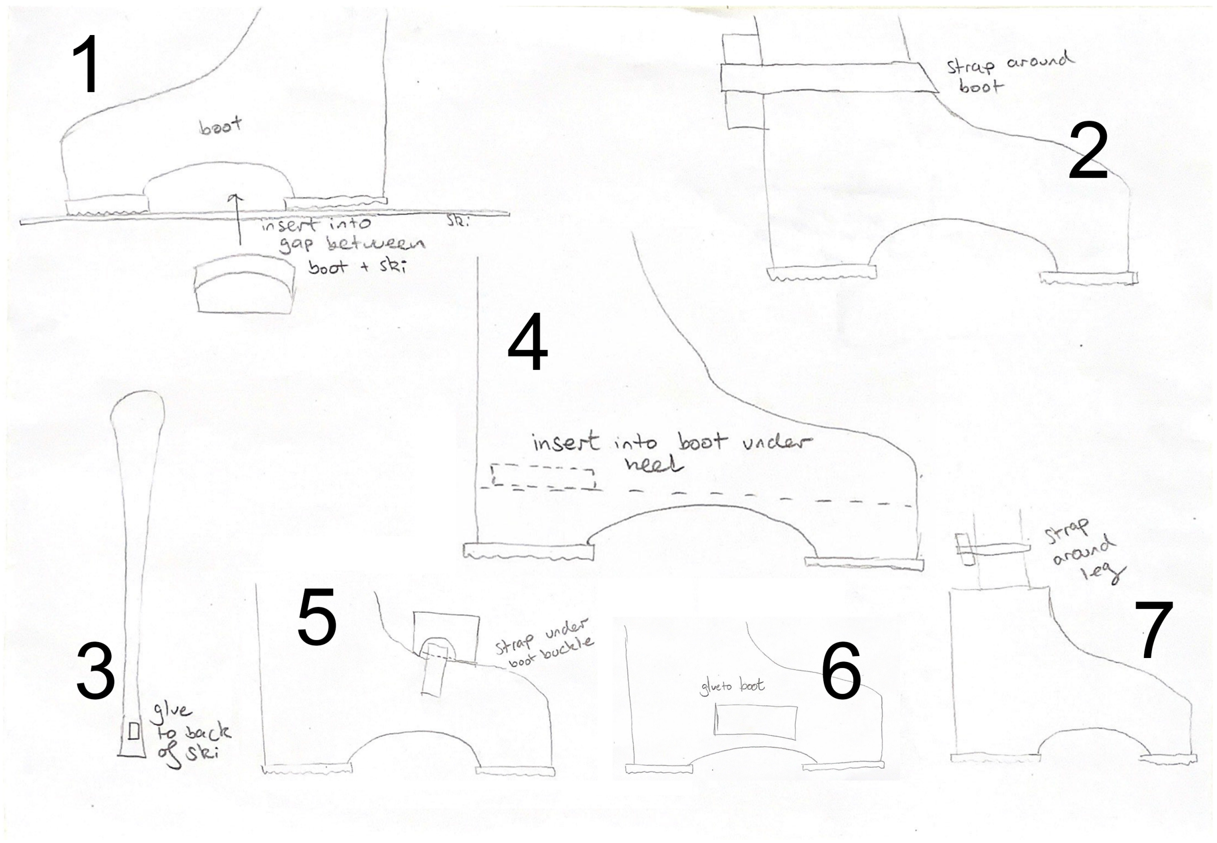


Figure 2.4: Design sketches exploring where to locate the sensors

between the sensors and the microcontroller. This protocol is low level and works over physical wires, so I would need to connect the sensors physically to the microcontroller. I sketched out several designs in Figure 2.4.

2.2.1 Design Analysis

In this section I analyse the benefits and drawbacks of the possible designs for locating the sensors shown in Figure 2.4.

Design 1, inserted between the ski boot and the ski, would be held in place by the bindings. This would be easy to insert and remove by clipping and unclipping the boot, but in a crash where the boots are ejected from the bindings, the device would come detached and potentially be broken.

Design 2, strapped around the boot, would be held in place by a velcro strap. Firstly this would not be as robust an attachment method as the bindings, and could come loose in the high g-force environment of ski racing. Secondly this may not hold the device tight enough to the boot to measure edge angle and acceleration magnitude accurately.

Design 3, attached directly to the ski, would be held firmly in place by the glue, however if the boots are ejected the device would be pulled apart from the skier, requiring connectors that can pull apart without damage to the device. In addition, this would not be easily swappable between different sets of skis. This is a key requirements of the project, necessary in order to conduct a user evaluation involving multiple skiers.

Design 4, inserted into the boot, underneath the heel of the skier, would be held in place by friction with the base created by the body weight of the skier and the boot straps holding the foot down. This would stay attached to the skier in a crash, and would be protected by the shell of the boot. It could be swapped between skiers but would require them to take their boots off.

Design 5, strapped under a boot buckle, would be held in place securely by the buckle straps, and would stay attached to the skier in a crash. It would be easy to swap between skiers without the need to

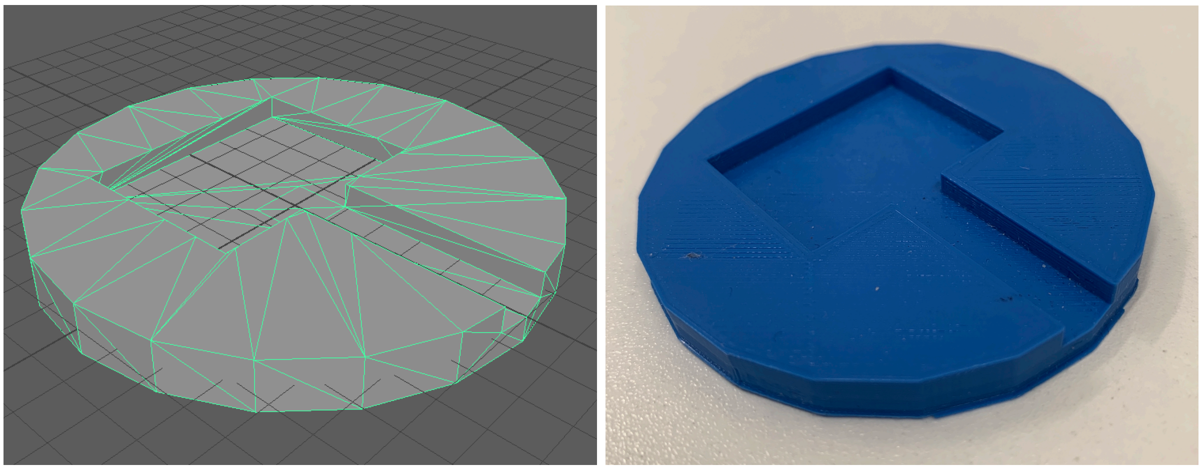


Figure 2.5: CAD model of footplate shell in Maya and 3D printed result

remove their boots. However, it would be susceptible to damage as it is on the outside, unprotected. It also may not fit onto all ski boots as some have designs with hidden buckle straps.

Design 6, glued to the outside of the boot, would be held firmly in place by the glue, and would stay attached to the skier in a crash. However, it would not be swappable between skiers, and would be unprotected in a crash.

Design 7, strapped around the skier's leg, would be held in place by a velcro strap, and would stay attached to the skier in a crash. It would likely be more protected due to being underneath the skier's salopettes or lycra. However, it may not measure edge angle and acceleration magnitude accurately given the difference in movement between the leg and the boot.

From this analysis I concluded the optimal design was number 4. This would be easily swappable between skiers, protected in a crash, and firmly attached to measure the edge angle and acceleration accurately. This design was inspired by ski boot warmers, which also sit in the ski boot underneath the foot, held in place by the weight of the skier. I decided to have the device inserted not directly under the foot, but between the inner shell and the outer shell of the ski boot. This gives a hard, plastic surface that the device can be firmly pressed against. One consideration for this design, however, is that different size boots will need different size inserts to fit tightly.

The footplate shell for the in-boot sensor was drawn up in Maya, and 3D printed using an Ultimaker 2+. This is illustrated in Figure 2.5. This shell accommodates both sensors and the required connecting wiring between them. The accelerometer and gyroscope were then both fixed in place with hot glue, and attached to long multicore wires that will run up the user's leg to the microcontroller.

I considered using a laser cutter to cut several layers out of acrylic and glue them together to produce this part of the device, however I wanted the shell to have a cut out that was the exact depth of the circuit boards contained, in order to minimise additional height in the boot. The thicknesses of acrylic available did not allow me to get a close fit of heights.

I produced two footplate shells, with an accelerometer and gyroscope in each, one for each leg. I had to reprint in order to adjust the size of the centre cut out to accomodate the internal wiring connecting the two sensors, as shown in Figure 2.6. I had to ensure the address pins were pulled up on one and pulled down on the other to give each one a unique I2C address. I orientated the sensors such that the edge angle of the ski is defined as the pitch of each footplate.

I was now in a position where I could test the device on a ski slope, and so I set about writing a program to record the orientation data for later output. However, I quickly found that the memory available on board an Arduini Uno is extremely limited and I would only be able to store several seconds worth of orientation data. The Arduino Uno has 2KB of volatile SRAM (Static Random-Access Memory), used for storing variables during run time. It also has 1KB of non-volatile EEPROM (Electrically Erasable Programmable Read-Only Memory), used for storing long-term information [7]. Each time I polled the sensors I needed to record four floating point numbers at a minimum, these were the pitch and roll angles for each ski. Each floating point number consists of four bytes, and I expected to poll the sensors at approximately 50Hz, which would result in a data rate of:

$$4 \text{ bytes} * 4 \text{ numbers} * 50 \text{ polls} = 800 \text{ bytes} / \text{second}$$

This would fill all 3KB of available memory in both the SRAM and EEPROM in under four seconds. The time taken to complete a slalom course varies between locations but in general is between one and five minutes, so to store this I needed between 48KB and 240KB of memory. This led me to move from an Arduino to a Raspberry Pi, as the Raspberry Pi Model 3 B+ has 1GB of volatile SDRAM (Synchronous Dynamic Random-Access Memory) to store variables at run time. In addition a memory card can be inserted to provide up to 32GB of non-volatile storage. I chose the Raspberry Pi over other microcontrollers due to its on-board Bluetooth and WiFi support. I envisioned controlling the device from a smartphone or laptop wirelessly in the future, which would require either another module to support wireless connections or moving to a microcontroller with wireless connectivity built in.

All my code up to this point was implemented in C using Adafruit's L3GD20H [5], MMA8451 [4], and unified sensor [6] libraries. These were not portable to the Raspberry Pi so I needed to rewrite the code B.2. I chose to use the Python language to implement the orientation code, because CircuitPython [13] provided useful implementations of libraries for both the accelerometer [3] and gyroscope [2].

I would have preferred to rewrite in C, which would have been closer to the original Arduino code, and as I was keeping in mind potential future performance issues, however libraries for the exact models of sensors I had did not exist.

After completing this rewrite, I was able to store several minutes of data at a time and write it out to a CSV file after.

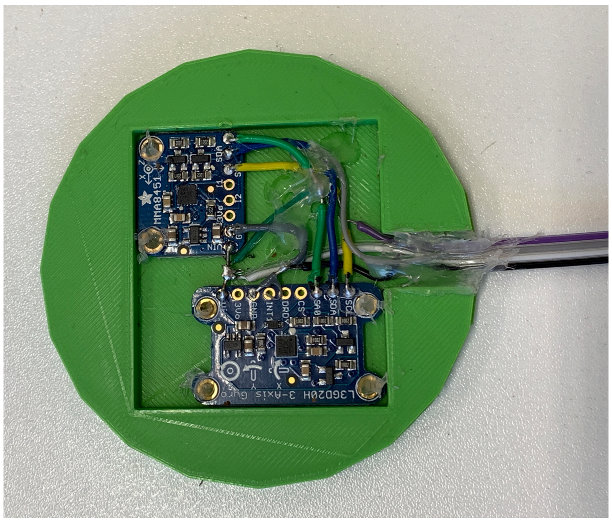


Figure 2.6: Footplate with gyroscope and accelerometer installed

The Raspberry Pi uses male pins, the inverse of the Arduino's female pins. I could no longer simply plug the end of the wire into the microcontroller, as I needed to add female connectors to the ends of the wires. I also needed to connect the positive, ground, data, and clock lines together for both footplates. I found the best way to do this, such that each foot wasn't hard-wired together, was to create a third Y-section of wire, with female connectors on each branch, which could plug directly into the Raspberry Pi, and both footplates simultaneously. This is shown in Figure 2.7. I also took the time to add proper male connectors to the end of each footplate wire.

In order to control this prototype on a ski slope without implementing a user interface, I set up the Raspberry Pi such that it automatically connected to my phone's WiFi hotspot and enabled a VNC server when turned on. I could then use a VNC client on my phone to view the desktop, and type commands into a terminal window. Clearly this is not an ideal method of interfacing with the device, however it worked well enough to begin a real-world test.

2.2.2 Real-World Test 1

I performed the first real-world test of my device on a dry ski slope nearby the university.

I put the Raspberry Pi and portable battery pack into a utility bag attached to my waist, inserted the footplates into the boots, and ran the wires up the inside of my salopettes. I then used my phone to start and stop recording at the beginning and end of each run down a slalom course.

I gained several key insights from this test. Firstly, I was able to record data as expected, and could clearly see patterns of turning left and right by leaning the skis onto their edge. These patterns corresponded to my intuition of how quickly I was turning and how many turns I was able to complete on each part of the course. This pattern of turns in shown in Figure 2.8.

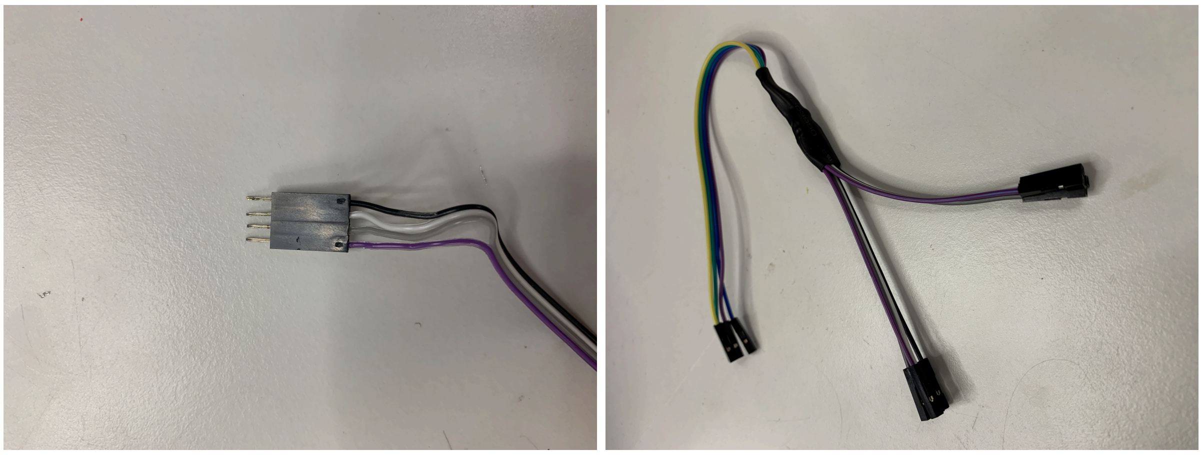


Figure 2.7: Male connectors and Y-section

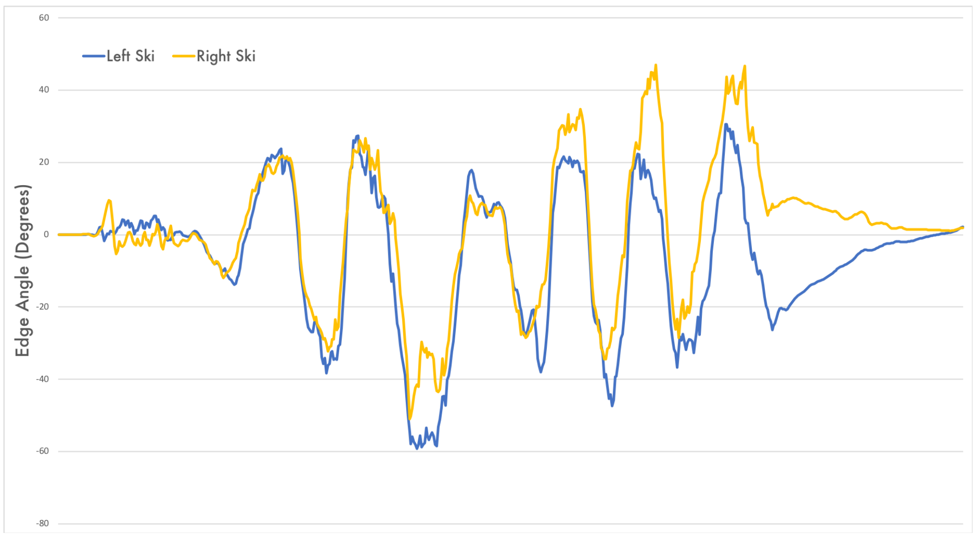


Figure 2.8: Pattern of left and right turns

Although I completed many runs down the slope, I was only able to record data for 2 of these runs. This was due to several issues, chiefly the male to female connection between the footplates and the Y-section of wire kept coming apart. They are only held together by friction and the large amount of movement that occurs in ski racing caused them to pull apart on many of the runs. The connection to the phone was also an issue, as it would occasionally drop and I would need to reboot the Raspberry Pi to reconnect, or the commands to start recording would not go through.

The data output was significantly noisier than I anticipated, and although the patterns of left and right turns were visible, the polling rate was low, at around 10Hz. With turns often taking place over the course of a single second, this seemed like a potential issue to try to address in future prototypes.

2.3 Prototype III

Clearly, I needed to redesign the connectors to be more robust so they do not come apart while using the device, and implement a reliable way of interfacing with and controlling the device.

2.3.1 Connectors

I purchased several Grove Universal 4 pin female connectors [18] and male cables [17]. These are interlocking connectors that clip and unclip, giving a much more reliable connection than purely friction. I replaced the old male and female connectors with the new ones and secured the wires to the connectors with hot glue to ensure any strain through the wire was not transferred into the solder connections. These interlocking connectors are shown in Figure 2.9.

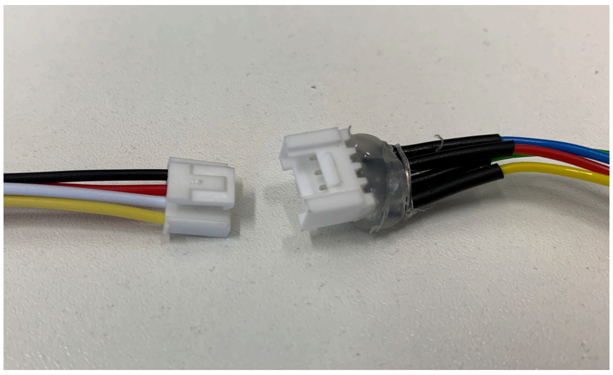


Figure 2.9: Male and female Grove Universal connectors

2.3.2 User Interface to Control the Device

There were several key requirements for the user interface. It needed to be simple to use when fully kitted up in ski gear at the top of a ski slope, and the data needed to be transferrable off the device for later analysis. I wanted this data to be accessible from any user device, not requiring a specific model of phone or laptop. I considered building an app and communicating with the device through Bluetooth, but ensuring it worked on both Android and iOS would be a challenge. In addition, Bluetooth libraries on the Raspberry Pi are non-trivial to use to communicate custom data.

I therefore chose to use a technology I am familiar with from previous experience, an ExpressJS web server on the Raspberry Pi, serving a ReactJS frontend to the user's device. In order for this to work, both devices need to be on the same network, so I configured the Raspberry Pi so that it broadcasts a WiFi network using HostAPD [19] and ran a DHCP router using DNSMasq [14] such that connecting to the device through a web browser would load the ReactJS frontend.

2.3.2.1 Backend Server

I built an API using ExpressJS to respond to queries from the client. The POST and GET endpoints of this API were:

- POST: */run* Spawns child subprocess to run python script on the device, initiating calibration and recording sequence.
- GET: */get_buf* Returns the current status of the device, i.e. the current point in the control flow sequence or error status.

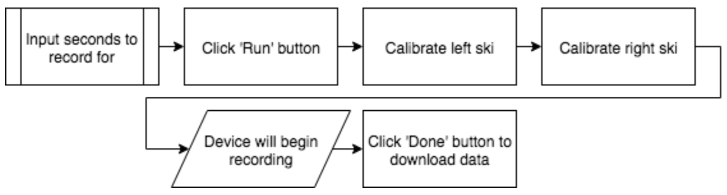

Input seconds to record for
Click 'Run' button
Calibrate left ski
Calibrate right ski
Device will begin recording
Click 'Done' button to download data

Figure 2.10: User interface control flow diagram

- GET: */get_file* Returns the name of the data file recorded in the last run, used by the frontend to retrieve the data.

The number of seconds to record for is inserted into the query parameter of the POST request to the */run* endpoint by the frontend application.

The */run* endpoint calls a function that first runs the Python script then communicates information about the current state of the script by piping the standard output of the child subprocess running it into a buffer update function. This function sorts and stores information about the state in a buffer on the server. The buffer update function runs asynchronously such that it does not block execution of the rest of the server, and continually updates the stored state in the background. The output of the python script is also output into the server log for debugging purposes.

The */get_buf* endpoint calls a function that checks the buffer to find the current state and returns that to the frontend.

The */get_file* endpoint calls a function that checks the buffer to find the filename output by the Python script and returns that to the frontend. In this way I passed data to and from the frontend showing the current status of the device, and spawned subprocesses to run the Python code performing the recording.

2.3.2.2 Frontend Application

I chose ReactJS as it enabled rapid prototyping of a single page web application (SPA) that could dynamically respond to state changes of the underlying python script. Because the user interface runs on the client, it remains dynamic and responsive even when the Raspberry Pi is under load recording data. I used Bootstrap CSS to ensure the user interface was mobile responsive and works well on any screen size.

The control flow is shown in Figure 2.10. Here calibration is performed by holding the ski boot still for at least 3 seconds, then moving the boot in a figure-8 motion until calibration is achieved. Holding the boot still allows the BNO055 sensor to calibrate the internal gyroscope, and the figure-8 motion allows calibration of the internal magnetometer. The internal accelerometer is pre-calibrated with accurate defaults.

While the device is being calibrated, twice per second the SPA calls the */get_buf* endpoint and advances through the control flow once each ski is calibrated.

Any device can connect to this network and load the user interface in a web browser, as shown in Figure 2.11. Once the device has finished recording clicking the Done button will call the */get_file* endpoint and load a link to download the recorded data in CSV format.

2.3.3 Real-World Test 2

I tested out the device again on the same dry slope. I also attached a GoPro action camera to my leg, pointing down. This allowed me to sync up the data with the camera footage and examine what certain features of the data looked like on video. I also modified the data output such that I recorded not only the orientation, but also the raw acceleration data along each axis in order to attempt to identify the patterns I could use to detect ski slippage.

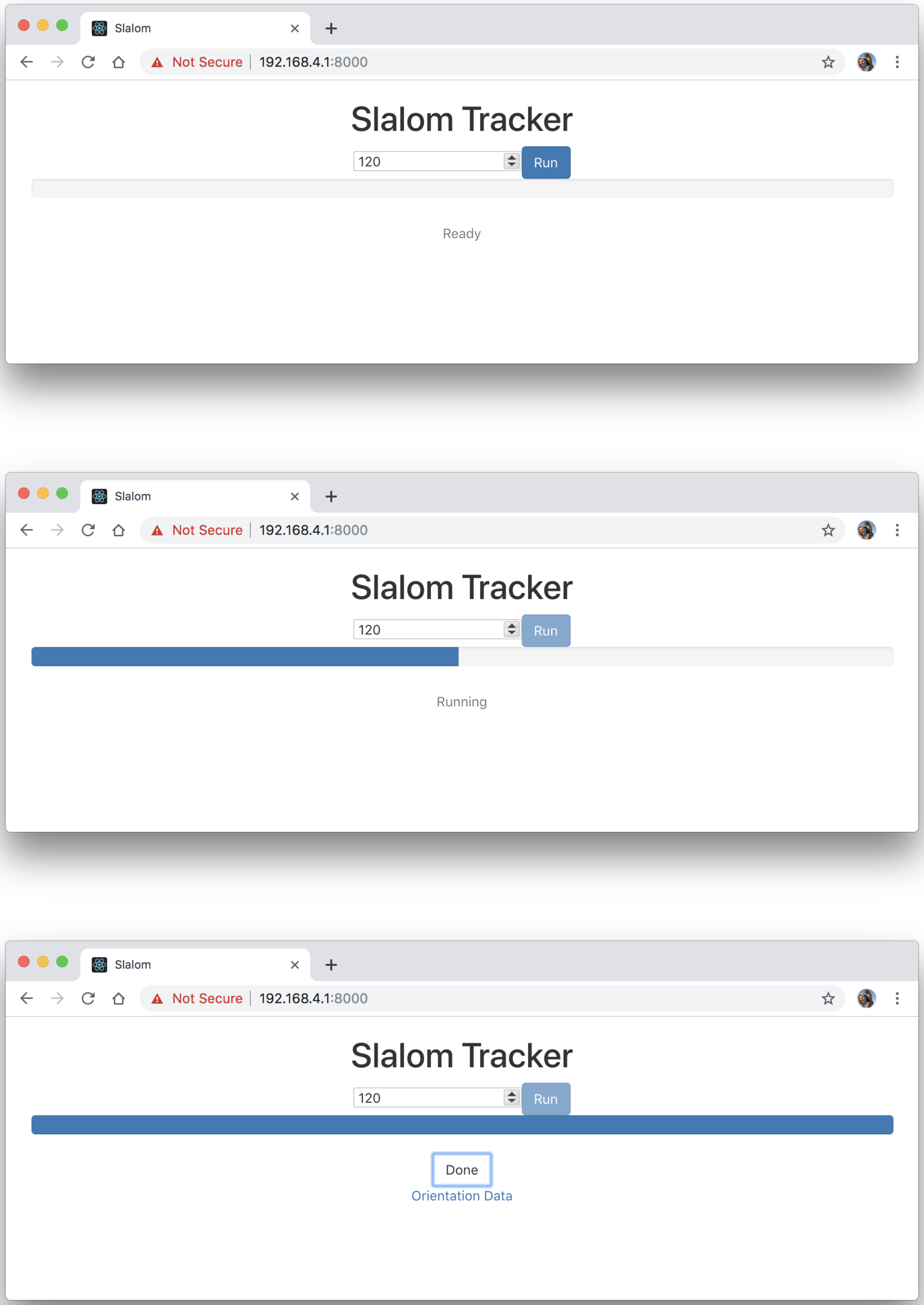


Figure 2.11: ReactJS user interface

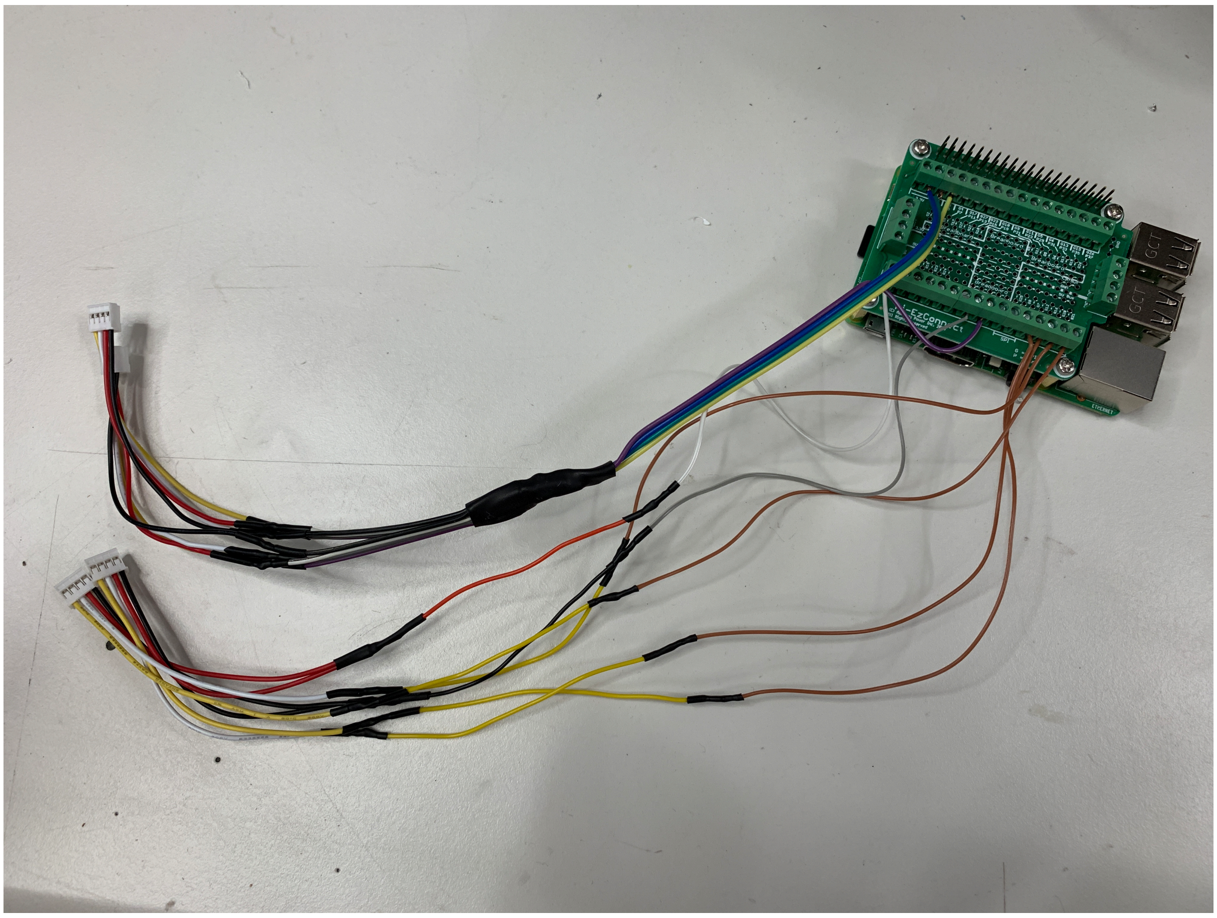


Figure 2.12: Terminal screw attachment

2.3.4 Prototype III Conclusion

This time it was significantly easier to set up and use the system. The user interface was usable from my phone and this time the connectors did not come lose on each run. I was therefore able to record many more runs.

I found that the male GPIO pins on the Raspberry Pi had been bent through the force of the wires pulling sideways during the tests. This was repairable with a pair of pliers, but I needed a more robust solution for attaching the Y-section to the Raspberry Pi. I purchased a 'hat' for the Rapsberry Pi, with terminal screws attached, cut the female connectors from the Y-section, and screwed them into the terminal screws instead as shown in Figure 2.12.

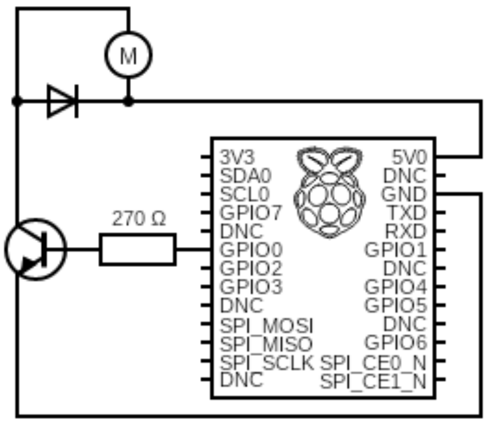
2.4 Prototype IV

The ability to provide feedback to the user through the use of vibration motors was another crucial component of my project, so it was implemented early on.

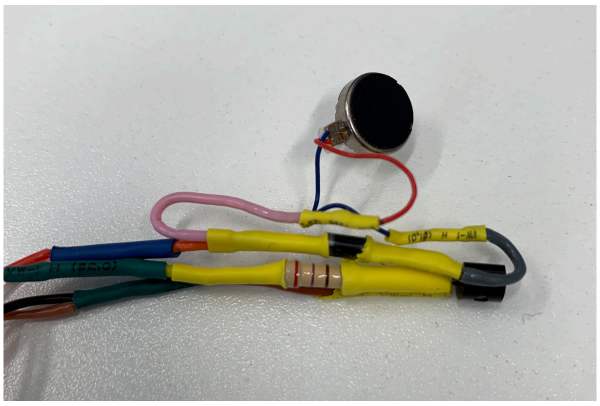
2.4.1 Motor Control

In order to give precise, adjustable feedback I needed to be able to activate vibration motors and vary their speed. I found the best way of achieving this was to use a Pulse Width Modulation (PWM) circuit. This circuit uses a transistor, and a diode to safely transform a 3.3V PWM pin output into a higher voltage PWM input to the motor.

The diode in Figure 2.13a is necessary due to the motor's 'back EMF'. Each time the motor is no longer powered and begins to slow down, the spinning magnets will induce a spike in voltage in the

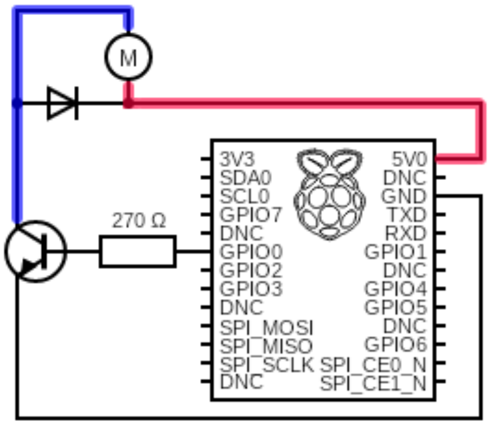


(a) PWM circuit diagram

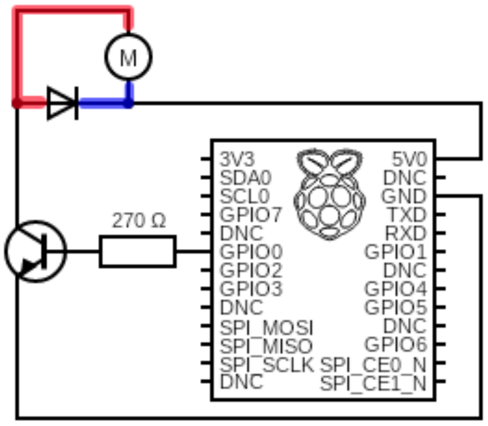


(b) Picture of the PWM circuit

Figure 2.13: Prototype IV PWM motor control circuit design



(a) Powered circuit



(b) Discharging circuit

Figure 2.14: Back EMF in a circuit

circuit, in the reverse direction in which the motor was powered. If not protected against, this could fry the microcontroller. Therefore, in the circuit the diode sits between the positive and negative terminals of the motor, only allowing current to flow from negative to positive. When the motor is being powered this has no effect on the circuit, as current must pass through the motor. This is shown in Figure 2.14a where the red highlighted segment of the circuit is at a higher voltage than the blue highlighted segment of the circuit. However, once the motor is no longer powered, and induces a reverse current, this current can pass through the diode, effectively short-circuiting and isolating this section of the circuit. This is shown in Figure 2.14b where the motor has raised the voltage on its negative terminal to higher than the voltage on its positive terminal, highlighted in red and blue as before. The current flowing in this direction through the motor also assists with slowing down the motor, applying a braking effect.

Initially I tested the motor circuits using an Arduino to verify they worked correctly. The Arduino Uno supports 5V out of the PWM pins, and so initially it appears as though the transistor is redundant, as I could just connect the positive and negative terminals of the motor directly to the Arduino Uno's PWM pins with a diode in between. However, the Raspberry Pi I was actually using for the project, only supports 3.3V out of the PWM pins. As I wanted to ensure I get the maximum feedback strength

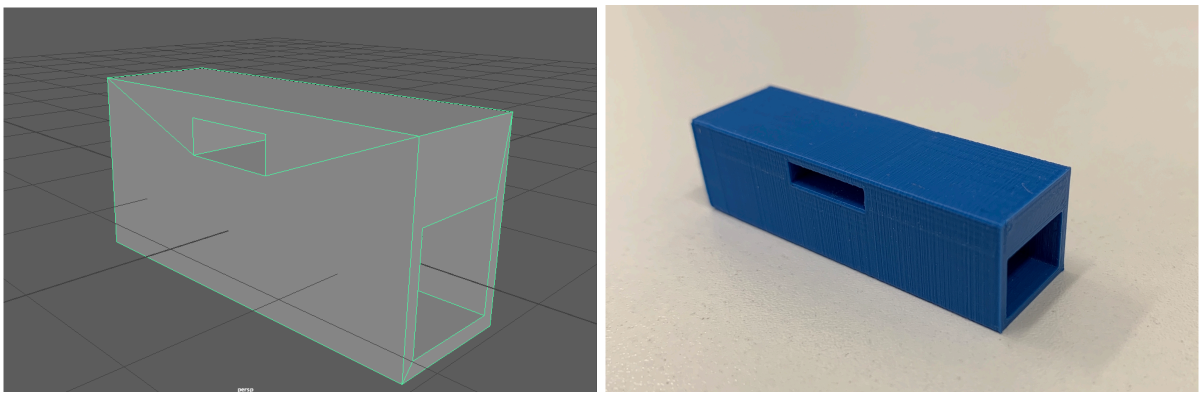
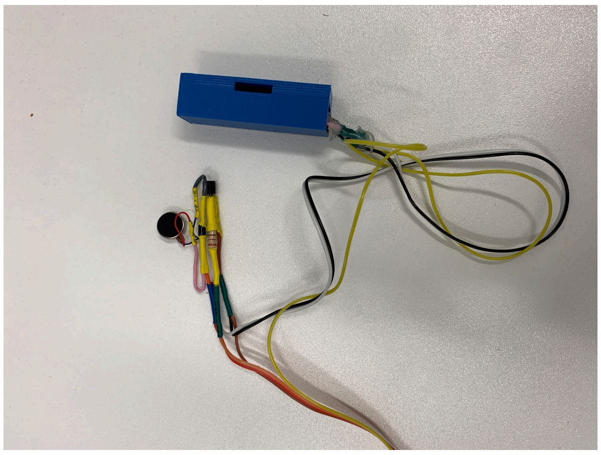
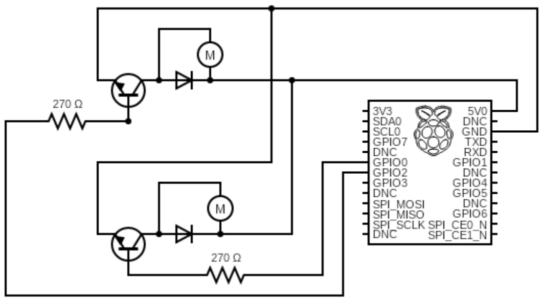


Figure 2.15: CAD model of the motor housing in Maya, and 3D printed result



(a) Motor wires spliced together with one circuit inserted into housing



(b) Overall motor circuit diagram

Figure 2.16: Prototype 4 motor control circuit

possible I needed to use the 3.3V PWM signal to act as a switch to control the 5V output into the motor.

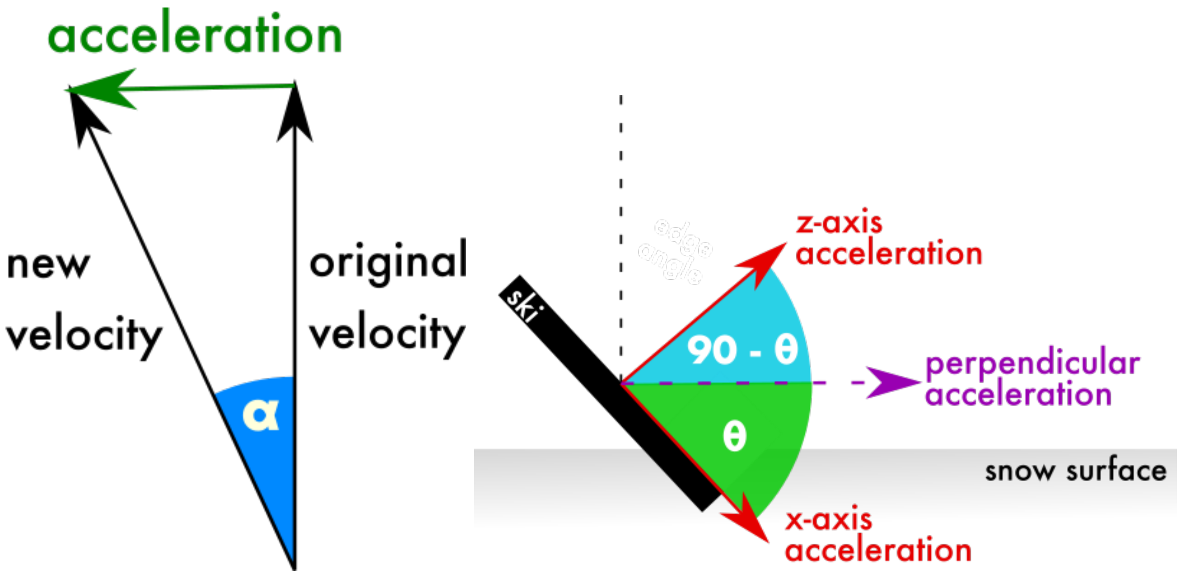
2.4.2 Motor Housing

I needed to house the circuit inside a durable shell that would both transfer the feedback through the shell to the user's leg, and also protect the circuit in a crash. I modelled a design using Maya and 3D printed it. I left a gap to feed straps through to attach the shells to each leg as shown in Figure 2.15. I created straps by sewing Velcro to a length of elastic and measuring up for a variety of leg thicknesses.

2.4.3 Motor Connections

As I was initially using an Arduino to test each circuit individually, I could just plug the end of the wires directly into the female IO pins. In order to build the combined circuit for each leg I ran 4 wires down each leg, positive, ground, and two PWM control wires for the inside and outside of the leg. I then spliced the positive and negative wires together, leaving the control wires separate, as shown in Figure 2.16a.

Using the same Grove Universal 4-pin connectors as before, I created another Y-section of wire to join the Raspberry Pi to the four motors. All four motors therefore used the same positive and ground wires, I tested this and found the current draw to be below that available on the Raspberry Pi, assuming only two motors would be active simultaneously. I screwed this Y-section into the terminal screws along with the other Y-section.



(a) Acceleration vector diagram (b) Diagram showing the calculation of perpendicular acceleration

Figure 2.17: Acceleration vectors used in slip calculation

2.4.4 Ski Edge Slippage Detection

Detecting and measuring ski edge slippage was the central aim of this project, however it required several key components to be in place before work could be done achieving it. For brevity I refer to ski edge slippage as slip in this section.

The slip detection algorithm works as follows for each leg:

1. Retrieve data from gyroscope and accelerometer.
2. Calculate the current edge angle.
3. Calculate the minimum rate of change of direction expected given the current edge angle.
4. Calculate the acceleration perpendicular to the ski, horizontal to the slope.
5. Calculate the difference between the measured acceleration and the mininum rate of change of direction expected.
6. Check if the measured acceleration is less than the minimum rate of change of direction expected.
7. If it is, provide feedback (either vibrotactile or auditory) about the direction in which the slip is occurring.

While the ski is within the carve zone [Figure 1.2] the radius of the turning curve the ski will ideally follow is given in equation 1.1. As θ increases, the turning curve radius decreases, leading a more rapid change of direction. This rate of change of direction can be measured as an acceleration as shown in Figure 2.17a where α increases as the turning curve radius decreases. This acceleration is perpendicular to the ski, as shown in Figure 2.17b. From the body reference frame this acceleration can be found through basic trigonometry as follows in equation 2.4:

$$Acc_{perp} = Acc_z \cdot cos(90 - \theta) + Acc_x \cdot cos(\theta) \tag{2.4}$$

where θ is the the edge angle of the ski and Acc_{perp}, Acc_z, Acc_x are the acceleration values measured by the sensor as shown in Figure 2.17b.

If the skis are not following this turning curve, and are changing direction at a lower rate, they are not gripping the slope well enough and are sliding sideways. This is the slip we are aiming to detect. The magnitude of slip that occurs depends on many factors including body weight and height, ski turning

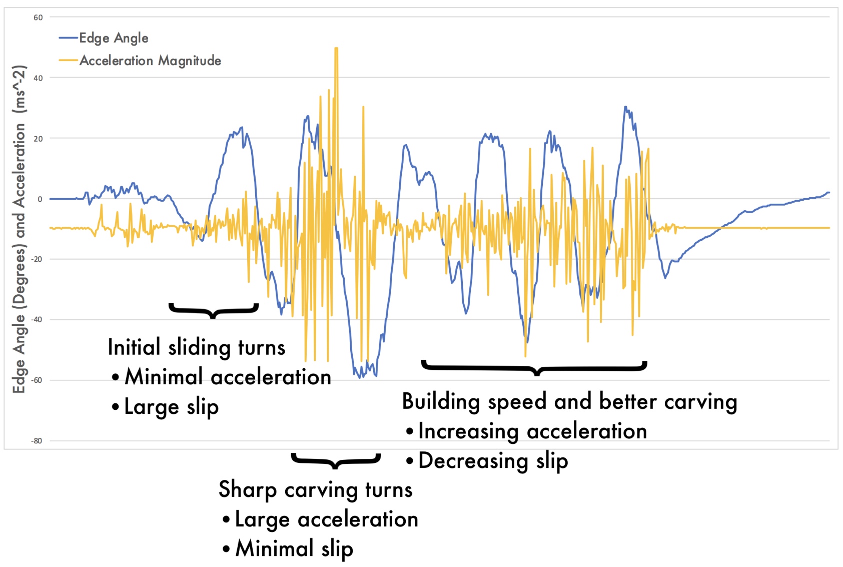


Figure 2.18: Annoted edge angle vs acceleration graph for a single ski during a dry slope test

radius, ski length, ski flexibility, and snow conditions (or dry slope conditions). Despite this we can make some assumptions and include sensible defaults for these parameters by examining data collected in the previous real-world tests. Using the video recorded in previous tests, I annotated which sections of the graphs corresponded to each turn, and how much slip was occurring around each one as shown in Figure 2.18. I used this to calibrate the device for my body weight, on my skis, on the dry slope.

2.4.5 Real-World Test 3

I was able to take my device to an indoor snowdome to test it on real snow. I performed several runs down a slalom course with the system attached. As before, I attached a GoPro to my leg pointing down.

2.4.6 Prototype IV Conclusion

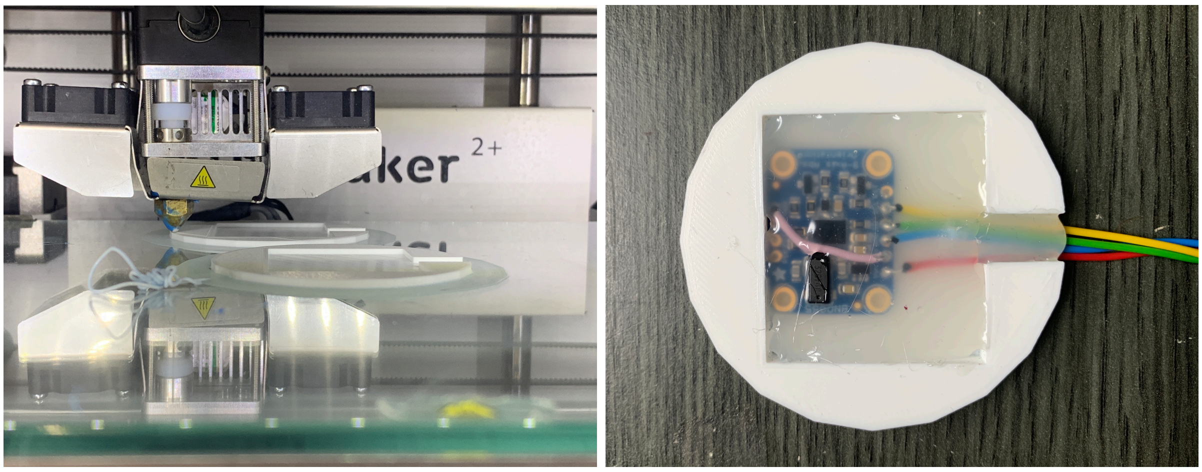
I found that it was much easier to perform carving turns on snow, as this is the surface skis are designed to work on. This was the first test with the vibration motors attached and feedback enabled, and I found the feedback was noticeable, however often activated at the wrong times, often right after a turn, or there was no feedback on quick turns that involved a lot of slippage. This was likely due to the low polling frequency from the sensors, and the relative inaccuracy of the complementary filter in the high g-force environment.

I decided that I needed to use a sensor with higher polling frequency, and one which was more accurate under the extreme conditions. I chose the Adafruit BNO055 breakout board. This board is an IMU that combines an accelerometer, gyroscope, and magnetometer, along with an on-board microcontroller that performs data fusion in real time, at 100Hz. This eliminates the need for processing orientation on the Raspberry Pi and provides more accurate readings than the complementary filter.

I also decided that the leg vibration motor shells needed redesigning as they had sharp corners and a crash could cause the casing to injure the user.

2.5 Prototype V

I redesigned the footplate shells in Maya and 3D printed two more that fit the new BNO055 sensors. I soldered long wires with Grove connectors to each, such that they could be swapped out simply by plugging in the new footplates. I sealed the BNO055 sensors into the new footplate shells using hot glue, as shown in Figure 2.19. This had the added benefit of sealing the sensors in completely and waterproofing



(a) Footplate being 3D printed (b) Footplate with BNO055 sensor sealed in

Figure 2.19: Prototype V footplate being manufactured

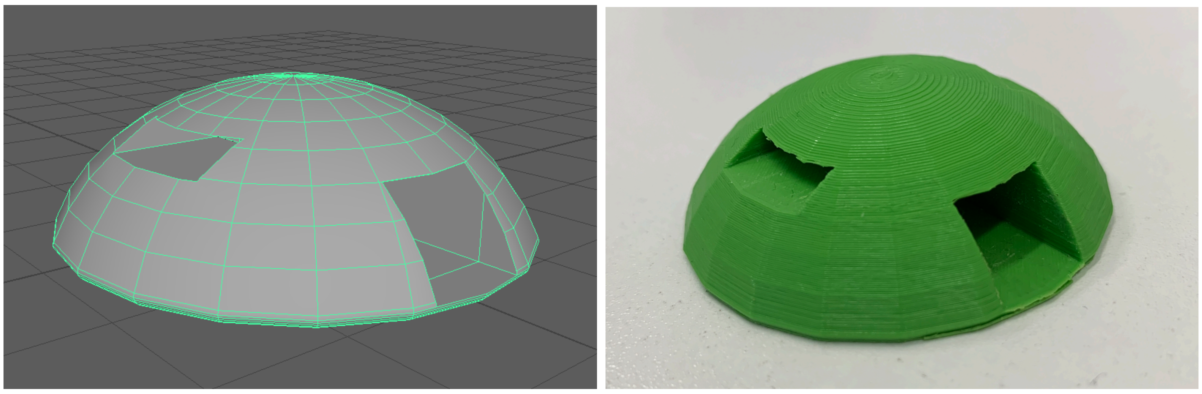


Figure 2.20: CAD model of motor housing in Maya, and 3D printed result

them. I had to rewrite sections of the Python code to support the new sensors, the full code is listed in Appendix B.3.

In order to redesign the leg vibration motor shells I drew up the new designs in Maya, choosing a hemisphere shape with no sharp corners such that they were both more rigid and less likely to cause injury.

The device also needed to be able to provide auditory feedback, in order to compare its efficacy with vibrotactile feedback I used a small Python wrapper around the SoX audio utility to do this. I set up the audio cues such that if no slip occurs, no noise would be heard. If slip does occur there is be a series of beeps that get louder and higher in pitch the greater the magnitude of the slip. This is the closest I was able to model the audio cues used in the study by Kirby [32] with the SoX audio library. Keeping with the study by Kirby [32], the ear the audio cue is heard in is the same side as the leg that is slipping. I connected a pair of earphones to the analogue audio port on the Raspberry Pi so that the sound can be heard by the user.

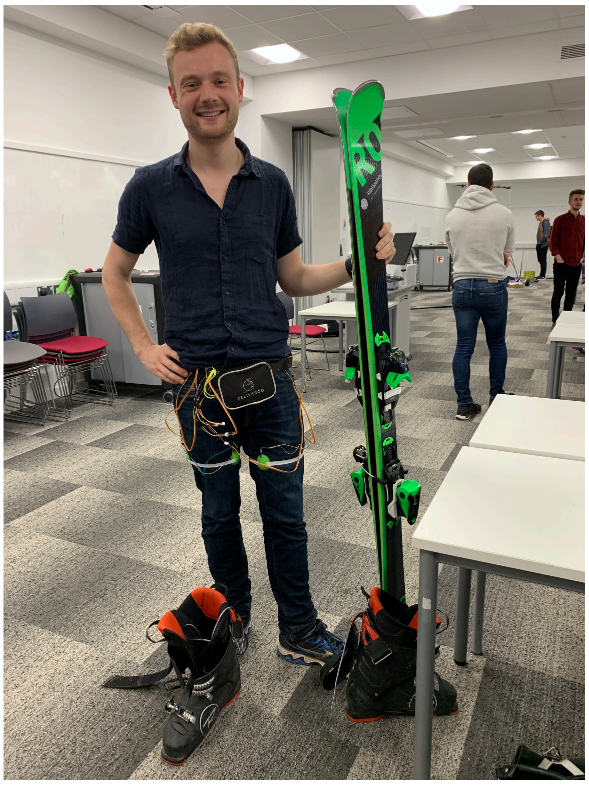
2.5.1 Real-World Test 4

The final test had to be performed on a dry slope rather than at the snowdome. I performed several runs with the new sensors and new leg mounts, and a GoPro attached as before. I used the calibration settings from the last run on a dry slope, and wore earphones in order to test the audio cues. I found that both the vibrotactile feedback and the auditory feedback corresponded to slip, occurred in the relevant direction, and the intensity of the feedback matched the magnitude of the slip that I was able to see in the video. I made the serendipitous discovery that lag present in the vibration motors spinning up and slowing down acted as a low-pass filter on the slip magnitudes measured, such that the resultant response was a smooth increase in intensity with higher slip magnitudes and a smooth decrease in intensity with lower slip magnitudes.

2.5.2 Prototype V Conclusion

This prototype concludes the iterative design process. The autobiographical design process resulted in a system which successfully fulfils the design specification, providing real-time vibrotactile and auditory feedback about ski edge slippage to ski racers. The final device is worn underneath salopettes and other equipment so will not be visible when skiing, however I've shown where the device is worn on the body in Figure 2.21a.

I now needed to evaluate SlalomTracker with expert skiers under race conditions. The next chapter discusses the execution of this evaluation and analyses the results.


(a) The author wearing the device indoors at a demonstration

(b) The author wearing the device on a ski slope

Figure 2.21: Device being worn by the author

Chapter 3

Critical Evaluation

I conducted a mixed methods user evaluation in order to compare the efficacy of vibrotactile and auditory feedback for providing real-time feedback to expert ski racers on ski edge slippage, and if it was possible to measure any reduction in ski edge slippage with the feedback enabled.

Three elite ski racers volunteered to take part h the evaluation. Each one has years of experience in ski slalom racing at an international level, so are well placed to offer useful feedback. The ski resort Val Thorens in the French Alps was chosen as the location to conduct the evaluation.

The evaluation was conducted as follows:

- Each racer performs three runs down a slalom course.
- The first run is a control, performed with the device attached, but feedback not enabled.
- The second run was performed with only auditory feedback enabled.
- The third run was performed with only vibrotactile feedback enabled.

Between each run the racers were asked to complete the NASA TLX questionnaire [22]. This questionnaire is a widely used [31], multidimensional assessment tool that scores perceived workload on six subjective 21-point (0 to 20) scales of mental demand, physical demand, temporal demand, overall performance, effort, and frustration level. A lower score indicates a more positive result, i.e. less mental demand, better overall performance, lower frustration level, etc.

They also answered a short series of questions after each run:

- Did you notice the feedback?
- Did you find the feedback distracting?
- Did you feel the feedback was useful?
- Were there any turns you felt were particularly bad?
- Were there any turns you felt were particularly good?

Each run was timed, and data was collected and recorded from the device. The answers to the questions were noted down after each run. This evaluation was conducted in line with Bristol's ethics standards C, and each racer given a randomly assigned number to ensure confidentiality.

3.1 Pre-Evaluation

On the day of the evaluation the conditions on the course were not ideal for racing. There were strong winds, low visibility due to cloud cover, and the steepest middle section of the course was icy. These challenging conditions turned out to be beneficial to the evaluation, creating more obvious patterns in the data recorded from the device.

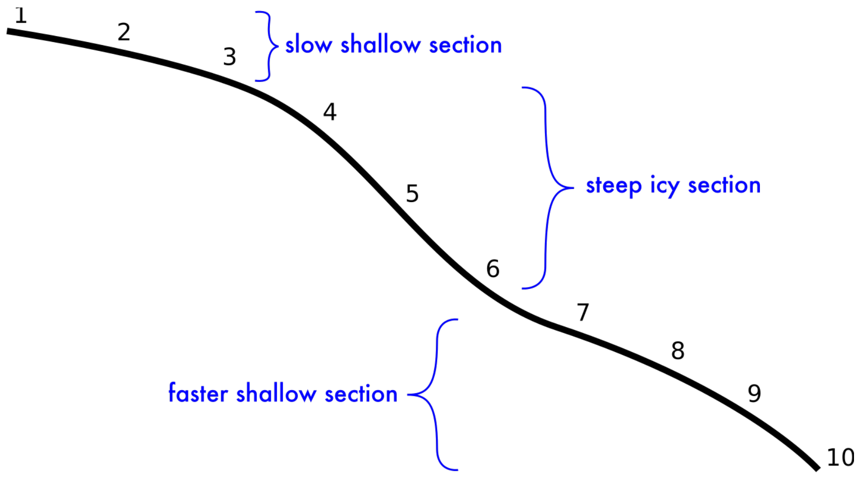


Figure 3.1: Approximate gradient of the course with each gate labelled

The section of the course being examined consisted of 10 slalom gates, evenly spaced down the slope and equally spread apart. The course started shallow, got steeper in the middle, then flattened out towards the end as shown approximately in Figure 3.1.

The skiers all used 165-centimetre Head branded slalom racing skis with a 13-metre sidecut radius, with edges in good condition from recent servicing.

3.2 Results

In this section I will report the qualitative and quantitative results separately before examining any triangulation that can be made between the methods to pinpoint and explain patterns.

3.2.1 NASA TLX Questionnaire

Averaged results from the TLX questionnaire are summarised in table form in Figure 3.1. Here the racers' average score for each scale on the questionnaire is displayed for the three runs. The scores for the runs with feedback enabled are shown alongside the increase or decrease from the control run, and highlighted in green to indicate a beneficial change or red to indicate a disadvantageous change.

	feedback type		
scale	none	audio	vibration
mental	11.7	15.7 (+4.0)	13.0 (+1.3)
physical	15.3	15.3 (+0.0)	15.7 (+0.3)
temporal	18.0	19.0 (+1.0)	17.0 (-1.0)
performance	6.7	3.7 (-3.0)	1.0 (-5.7)
effort	15.3	13.7 (-1.7)	11.3 (-4.0)
frustration	2.7	2.3 (-0.3)	1.3 (-1.3)

Table 3.1: TLX questionnaire average results and differences from control run

On average, there was a notable average increase of four points in mental demand with the auditory feedback enabled compared with no feedback enabled. In discussion with the racers this was due to the concentration required to translate the auditory feedback into an understanding of the movement of their skis. This increase in mental demand was smaller with vibrotactile feedback enabled, only 1.3 points. The racers' comments mentioned that although there was extra information to process in comparison to no feedback, the vibrotactile feedback corresponded to their intuition, so required minimal extra concentration.

There was not a large difference (no change with auditory feedback and a 0.3 point increase with vibrotactile feedback) in physical demand perceived between runs on average, an observation that makes sense given the feedback mechanisms do not assist the racers physically, only giving them cues to help guide correct technique. This is as opposed to haptic guidance studies such as [29] where the participants were mechanically assisted with the motion required.

Temporal demand is defined by how hurried or rushed the perceived pace of the task was. There was a slight increase on average in temporal demand with the auditory feedback and a slight decrease with the vibrotactile feedback. Both types of feedback increased mental demand, but this results shows the effect of the auditory feedback was to make the racers feel more rushed in comparison to when they were receiving vibrotactile feedback.

There was a notable decrease of three points on performance score, corresponding to better perceived performance, with the auditory feedback enabled over no feedback. This importantly shows the racers perceived their performance to be better while receiving the feedback. This corresponds to the results found by Kirby (2009) [32]. There was a further decrease in performance score of 5.7 points below the control run with the vibrotactile feedback enabled. This is a marked improvement and demonstrates that vibrotactile feedback was perceived to improve performance more than auditory feedback. Due to the runs being completed in order, there was probably a learning effect where racers improved performance due to familiarity with the course. Therefore, the comments made by the racers were crucial to attributing this performance gain to the feedback. Comments from the racers indicate that they were able to achieve a faster run as they were noticing slippage occurring in places they did not anticipate and were able to adjust to reduce it. Racer 324 stated after the run with the vibrotactile feedback that they “didn’t realise how much [their] inside ski slipped when [they] changed edges”, referring to the point where the skis lean from one side to the other, changing which edge is in contact with the snow.

There was a decrease in effort score of 1.7 points for the runs with auditory feedback, and 4 points with vibrotactile feedback. This again will have been affected by racers gaining familiarity with the course, and from my personal experience I know completing a familiar course requires less effort to achieve the same time than an unfamiliar course. The racers’ informal comments indicate that this was the case throughout the three runs completed.

The frustration scores were marginally lower on average, 0.3 points and 1.3 points lower for the auditory and vibrotactile feedback respectively. Racer 324 felt more frustrated with the audio feedback than without it because they “could hear when [they were] messing up” on turns of the course they felt they were worse. Racer 081 noted that both feedback modes helped them navigate the icy section of the course and as a result reduced their frustration with the more challenging turns.

To summarise the qualitative analysis, the vibrotactile feedback improved average perceived performance to a greater extent than the auditory feedback. The vibrotactile feedback also reduced the perceived effort required to achieve this performance by more than the auditory feedback, and made the racers feel less rushed and less frustrated while doing so. Both modalities of feedback increased the mental demand, ie. cognitive load, however, the vibrotactile feedback added less cognitive load than the auditory feedback.

3.2.2 Data Recorded

In order to quantitatively compare between runs of the course I turned to the data recorded from the device. I needed a measure of the total slip occurring for each turn. To achieve this, I calculated the area under the slippage curve (AUC) for each turn. I wrote a Python script that reads in the CSV file output by the device and calculates the sum of the slippage for each turn, as defined by the points where the ski edge angle goes through 0 degrees. These summed slippages were examined and cleaned up manually, due to the skis starting off approximately flat and the edge angles often crossing 0 degrees several times before the first turn.

I plotted graphs of the edge angle overlaid with the slip magnitude for each run, and matched up these graphs with the slippage AUCs as shown in Figure 3.2. Note that all slip magnitude values given here and further in this report are in arbitrary units output by the device, and do not have a direct mapping to SI units. In Figure 3.2 we can see the slippage occurring for each foot throughout each turn. For example, on turn 2 the left ski slips out significantly more than the right ski. The edge angles are negative, meaning the racer is turning to the left, so the left ski is the inside ski. This would have produced a series of auditory beeps in the racer's left ear, or a vibration on the inside of the racer's left leg, depending on which mode of feedback was being used.

On turn 7 there is a large amount of slippage occurring on both skis, with slightly more on the left ski. The edge angles are positive, meaning the racer is turning to the right, so the right ski is the inside ski. Depending on the mode of feedback this would either have produced beeps in both the racer's ears, with a higher frequency in the right ear, or a vibration on the outside of the racer's left leg and inside of their right leg, with the stronger vibration occurring on the left leg.

I added together the slip per turn from the left and right skis and plotted a bar chart of the results from the three runs for each racer as shown in Figure 3.4.

In Figure 3.4b on turns 4, 5, and 6, through the icy middle section of the course, there is a clear pattern shown in the graph. The first run with no feedback enabled had the highest total slippage, the second run with auditory feedback enabled had lower total slippage, and the final run with vibrotactile feedback enabled had the lowest total slippage. The is corroborated with the TLX questionnaire filed in by, and comments from, racer 324 discussed earlier, where they perceived themselves to have performed better with each mode of feedback enabled and attributed this to a better understanding of the slippage occurring on their inside ski. Through this we can see that the qualitative results match up to the data recorded, allowing us to make a stronger statement about the validity of the vibrotactile feedback mechanism.

In Figures 3.4a and 3.4c on the same icy middle section, the lowest total slippage can be seen with vibrotactile feedback enabled. The TLX questionnaires filled in by these racers show this improvement was perceived by the racers, with all of them indicating a better performance score with vibrotactile feedback than with auditory or no feedback.

I also summed the total slippage for both feet over all the gates for each racer with each feedback mechanism. I found that for racers 324 and 291 the total slippage was lower with auditory feedback than with no feedback, and that there was a further decrease with vibrotactile feedback. This agrees with both the comments from the racers and the most challenging sections of the run having lower slippage with the feedback modes. Racer 081 recorded a lower total slippage with both feedback modes than with no feedback, however the run with vibrotactile feedback enabled showed a higher total slip than the run with auditory feedback. By examining Figure 3.4a we can see this was largely due to turn 10. This turn was an outlier in the results, with over three times more slippage occurring with vibrotactile feedback than with auditory feedback, and over twice as much slippage as the previous turn made on the same run. By comparison, there was a small (21%) decrease in slippage between this turn and the previous one on the run made with auditory feedback. If I exclude this turn from the results and only sum the total slippage over the first nine turns we find a trend that matches up to trend seen across all turns with other racers. The summed slippage over the first nine turns for each racer and feedback mode is shown in Figure 3.3c. Here we can see that the removal of an outlier produces the same trend that appears in the runs for racers 324 and 291 for all 10 turns. The trend seen over all 10 turns for racers 324 and 291 remains visible even when only the first 9 turns are examined.

3.2.3 Race Times

While the data from the device was recorded over the top 10 turns of the course, I was only able to record times for the entire run of the course, which had 18 gates. Despite this, the trend in the times between each run correlate with the slippage measurements, TLX questionnaire results, and comments made in the short series of questions. As shown in Figure 3.3a, all the ski racers recorded quicker times on each consecutive run, starting with no feedback, then with auditory feedback, finishing with vibrotactile feedback. The differences were small, but this is expected in ski racing where marginal differences separate top racers from average ones. When seen next to the total slippage over the first 9 turns, as in Figure 3.3 it can be seen that the trend in reducing slippage correlates to the trend in reducing times. This

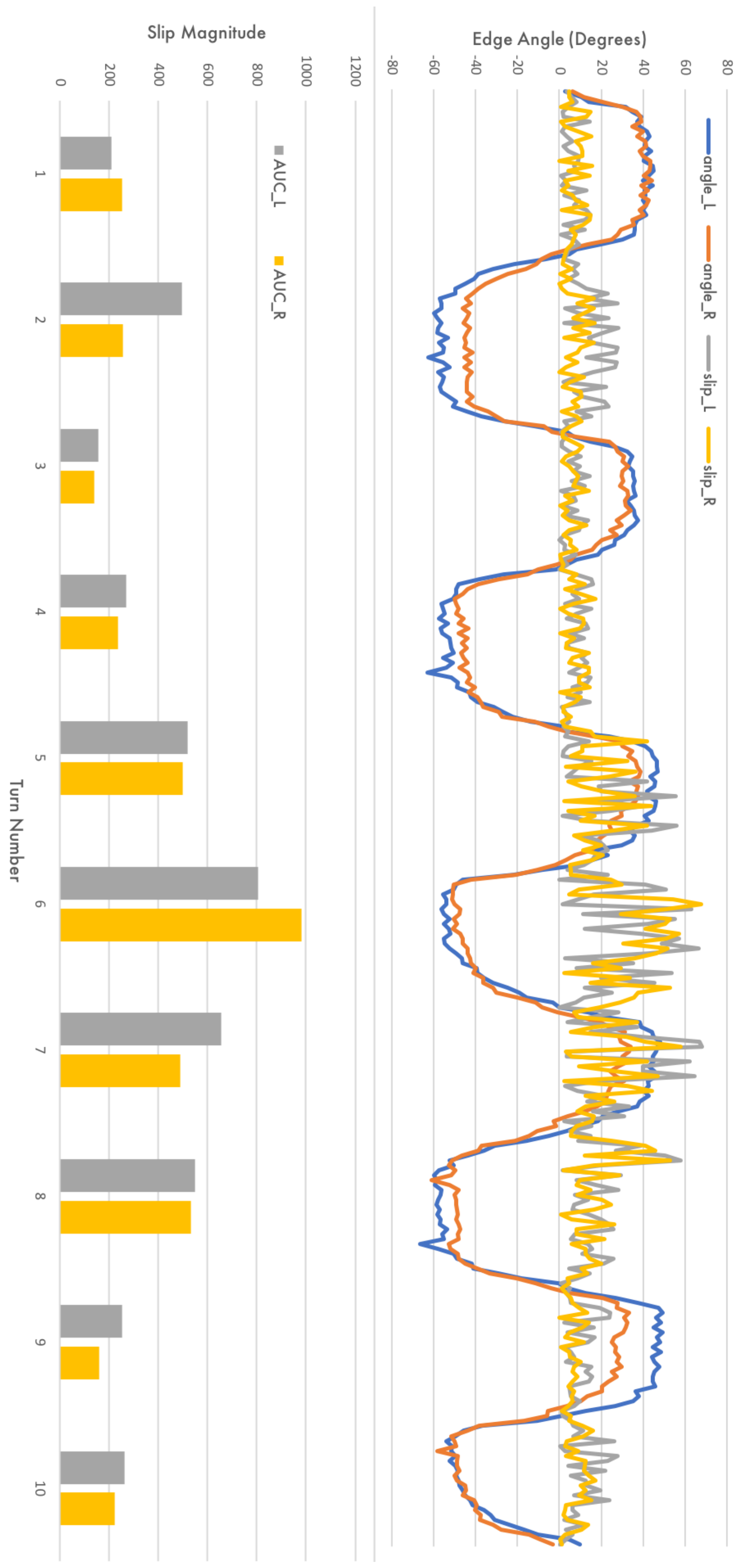
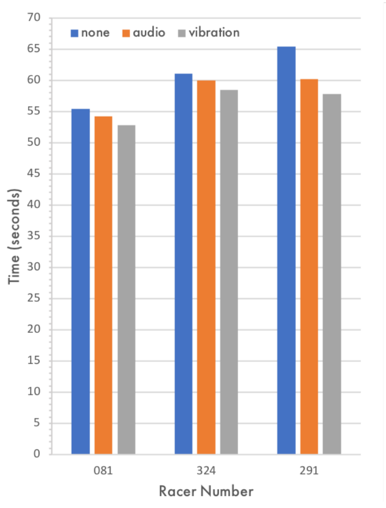


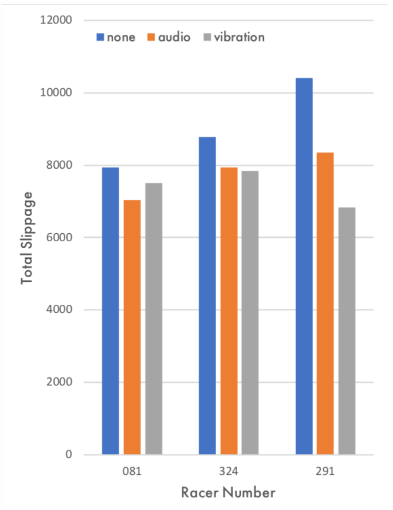
Figure 3.2: Edge angle overlaid with slip magnitude, matched with slip AUC of each ski per turn - Racer 081 Run 1

importantly shows that there is likely a correlation between total slip magnitude and time taken to complete a course, which is naturally the most important metric, as ski racing success is determined by time taken to complete a course. This confirms that reducing slip magnitude can be an effective way of improving racers performance, an assumption made at the start of this project based on previous work done in this field [32] and my own experience as a ski racer.

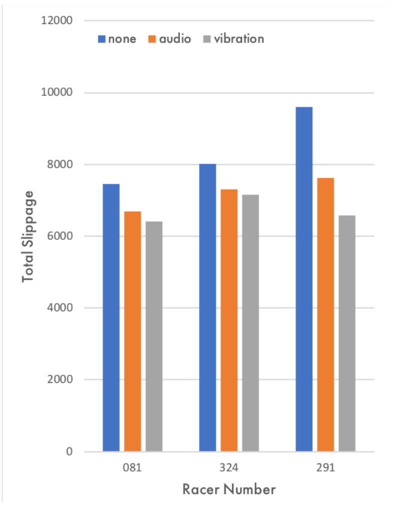
Due to the evaluation only involving three participants it was not possible to conduct a statistical comparison of the three conditions, however despite this the trends are clear - real-time feedback reduces ski edge slippage and vibrotactile feedback reduces it to a greater extent than auditory feedback. This same trend is clear with race times, and an reduction in ski edge slippage appears to correlate to a reduction in race time.



(a) Race times

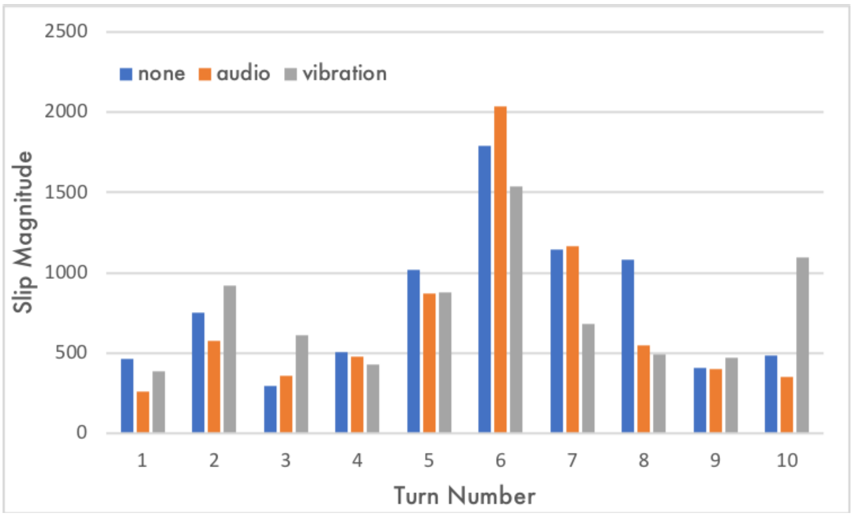


(b) Total slippage

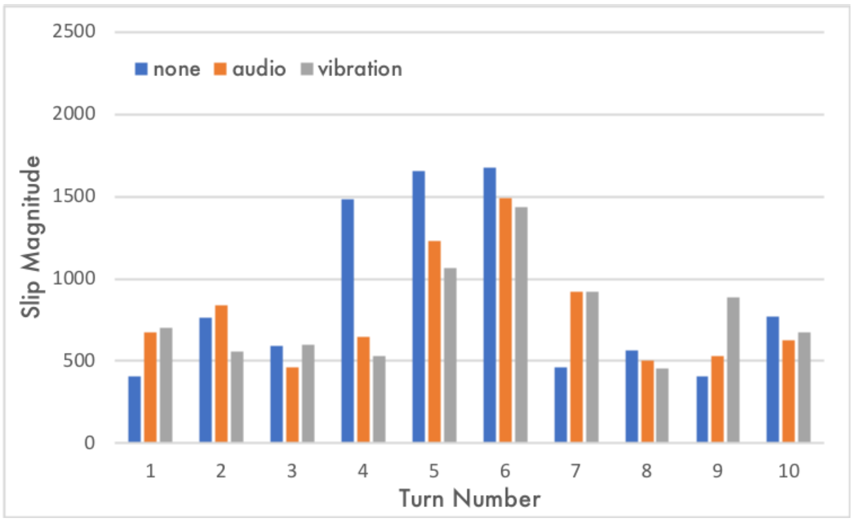


(c) First 9 turns slippage

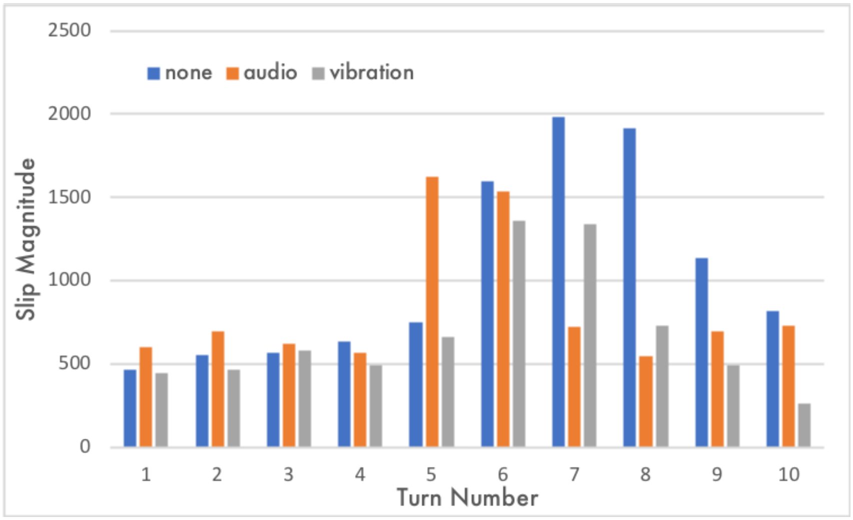
Figure 3.3: Race times and total slippage for each racer with each feedback mode



(a) Racer 081



(b) Racer 324



(c) Racer 291

Figure 3.4: Total slip per turn for each racer

Chapter 4

Conclusion

4.1 Main Contributions

The main contributions of this project were the development and evaluation of SlalomTracker, a device to measure ski edge slippage using only off-the-shelf parts, something which before has only been achieved through expensive custom sensor technology or prohibitively long preparation of the system for the specific course at hand. This device has been proven to withstand the extreme conditions present on a ski course and during a slalom race. The device can provide both auditory and vibrotactile feedback to racers in real time. The auditory feedback was designed to be similar to the auditory feedback used in a previous study [32] in order to provide a point of reference for the results of this project. The vibrotactile feedback was designed using an autobiographical design process through my own experience of slalom racing.

An in the wild mixed methods user evaluation was conducted in the French Alps using three elite ski racers. This evaluation found promising results to indicate that vibrotactile feedback adds less cognitive load than auditory feedback in this task, shown through lower 'mental demand' scores reported on average on NASA TLX questionnaires. The vibrotactile feedback was also perceived to increase performance over both no feedback and auditory feedback, again shown through better performance scores on the TLX questionnaires. The data recorded from the device highlighted that through the most challenging turns the racers achieved lower levels of slippage with vibrotactile feedback enabled than with auditory or no feedback. This data recorded was corroborated by comments made in a series of short questions asked alongside the TLX questionnaires, helping to isolate the improvement in performance to the feedback provided rather than the learning effect.

Th project has shown the effectiveness of the autobiographical design approach when the designer is an expert themselves in the activity that is being designed for.

4.2 Future Work

The potential future expansion of this project falls into two main categories; a more experimentally rigorous trial, and an easier to use device.

4.2.1 Trial Improvement

My user evaluation gave promising, however not statistically significant, results about the use of vibrotactile feedback over auditory feedback, due to the likelihood of the learning effect influencing the results, and the fact that there were only three elite ski racers used in the evaluation. A future trial with a larger number of ski racers and more runs completed would enable a statistical comparison of the efficacy of the auditory and vibrotactile feedback. In a between groups study design could be used where each ski racer would complete a number of runs of the course with no feedback to familiarise themselves with it first. Then a third of those racers would complete multiple runs of the course with auditory feedback. Another third of those racers would complete multiple runs of the course with vibrotactile feedback. The final third of those racers would complete the same number of runs as the other groups, but again with no feedback enabled. Each group would have the post-familiarisation runs averaged in order to reduce the effect of outliers. The no feedback, auditory feedback and vibrotactile feedback conditions could then be statistically compared in terms of ski slippage and course completion times and more robust conclusion drawn about the effects of each.

4.2.2 Device Improvement

My device was able to collect and record the necessary data, however there were several difficulties present in the final in the wild evaluation. Firstly although the device was swappable between users, the wired connections of each part meant it was time consuming and fiddly to install in a user's boots, and required much patience and cooperation with the ski racers. A future solution might add wireless functionality between the boot inserts, leg vibration motors, and central microprocessor. This would reduce the time required to equip a user with the device, and increase future marketability of a product based on this device.

Currently the device has pre-set calibration values for the skier's body weight and height, ski turning radius, ski length, ski flexibility, and snow conditions for the evaluation. I adjusted these manually in the code, however before any further evaluation it would be prudent to add the ability to adjust these values using the user interface.

Bibliography

[1] 2018 winter olympics men's slalom results. https://www.olympic.org/pyeongchang-2018/alpine-skiing/mens-slalom. Accessed: 2019-05-05.

[2] Adafruit circuitpython l3gd20 library. https://github.com/adafruit/Adafruit_CircuitPython_L3GD20. Accessed: 2019-03-10.

[3] Adafruit circuitpython mma8451 library. https://github.com/adafruit/Adafruit_CircuitPython_MMA8451. Accessed: 2019-03-10.

[4] Adafruit mma8451 library. https://github.com/adafruit/Adafruit_MMA8451_Library. Accessed: 2019-03-10.

[5] Adafruit unified l3gd20 library. https://github.com/adafruit/Adafruit_L3GD20_U. Accessed: 2019-03-10.

[6] Adafruit unified sensor library. https://github.com/adafruit/Adafruit_Sensor. Accessed: 2019-03-10.

[7] Arduino memory. https://www.arduino.cc/en/tutorial/memory. Accessed: 2019-04-29.

[8] Arduino uno rev 3. https://store.arduino.cc/arduino-uno-rev3. Accessed: 2019-04-12.

[9] Autodesk maya. https://www.autodesk.com/products/maya/overview. Accessed: 2019-05-09.

[10] Bno055 absolute orientation sensor. https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor. Accessed: 2019-04-12.

[11] Bootstrap. https://getbootstrap.com/. Accessed: 2019-05-09.

[12] Breadboard. https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all. Accessed: 2019-04-12.

[13] Circuitpython. https://github.com/adafruit/circuitpython. Accessed: 2019-03-10.

[14] Dnsmasq. http://thekelleys.org.uk/dnsmasq/doc.html. Accessed: 2019-05-09.

[15] Expressjs. https://expressjs.com/. Accessed: 2019-05-09.

[16] Git. https://git-scm.com/. Accessed: 2019-05-09.

[17] Grove universal 4 pin buckled cable. https://coolcomponents.co.uk/products/universal-4-pin-buckled-20cm-cable-5-pcs-pack. Accessed: 2019-04-29.

[18] Grove universal 4 pin connectors. https://coolcomponents.co.uk/products/grove-universal-4-pin-connectors-90-10-pcs. Accessed: 2019-04-29.

[19] Hostapd. http://w1.fi/. Accessed: 2019-05-09.

[20] L3gd20h triple-axis gyroscope breakout board. https://www.adafruit.com/product/1032. Accessed: 2019-04-12.

[21] Mma8451 triple-axis accelerometer beakout board. https://www.adafruit.com/product/2019. Accessed: 2019-04-12.

[22] Nasa task load index. https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLXScale.pdf. Accessed: 2019-05-09.

[23] Reactjs. https://reactjs.org/. Accessed: 2019-05-09.

[24] Sound exchange. http://sox.sourceforge.net/. Accessed: 2019-05-09.

[25] Ultimaker 2+. https://ultimaker.com/en/products/ultimaker-2-plus. Accessed: 2019-05-09.

[26] Ultimaker cura. https://ultimaker.com/en/products/ultimaker-cura-software. Accessed: 2019-05-09.

[27] Bonnie Baker. Apply sensor fusion to accelerometers and gyroscopes. *Digi-Key Electronics*, 2018.

[28] Matthew Brodie, Alan Walmsley, and Wyatt Page. Fusion motion capture: a prototype system using inertial measurement units and gps for the biomechanical analysis of ski racing. *Wiley Sports Technology*, 1, 2008.

[29] G. Grindlay. Haptic guidance benefits musical motor learning. In *Proceedings of Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2008.

[30] David L. Hall and James LLinas. An introduction to multisensor data fusion. In *Proceedings of the IEEE (Volume 85, Issue 1)*, 1997.

[31] Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society*, 2006.

[32] Richard Kirby. Development of a real-time performance measurement and feedback system for alpine skiers. *Wiley Sports Technology*, 2, 2009.

[33] Carman Neustaedter and Phoebe Sengers. Autobiographical design. *ACM Interactions*, 2012.

[34] NXP Semiconductors. *I2C-bus specification and user manual*, 6 edition, 2014.

[35] Janet van der Linden, Erwin Schoonderwaldt, Jon Bird, and Rose Johnson. Musicjacket - combining motion capture and vibrotactile feedback to teach violin bowing. Technical report, The Open University, 2010.

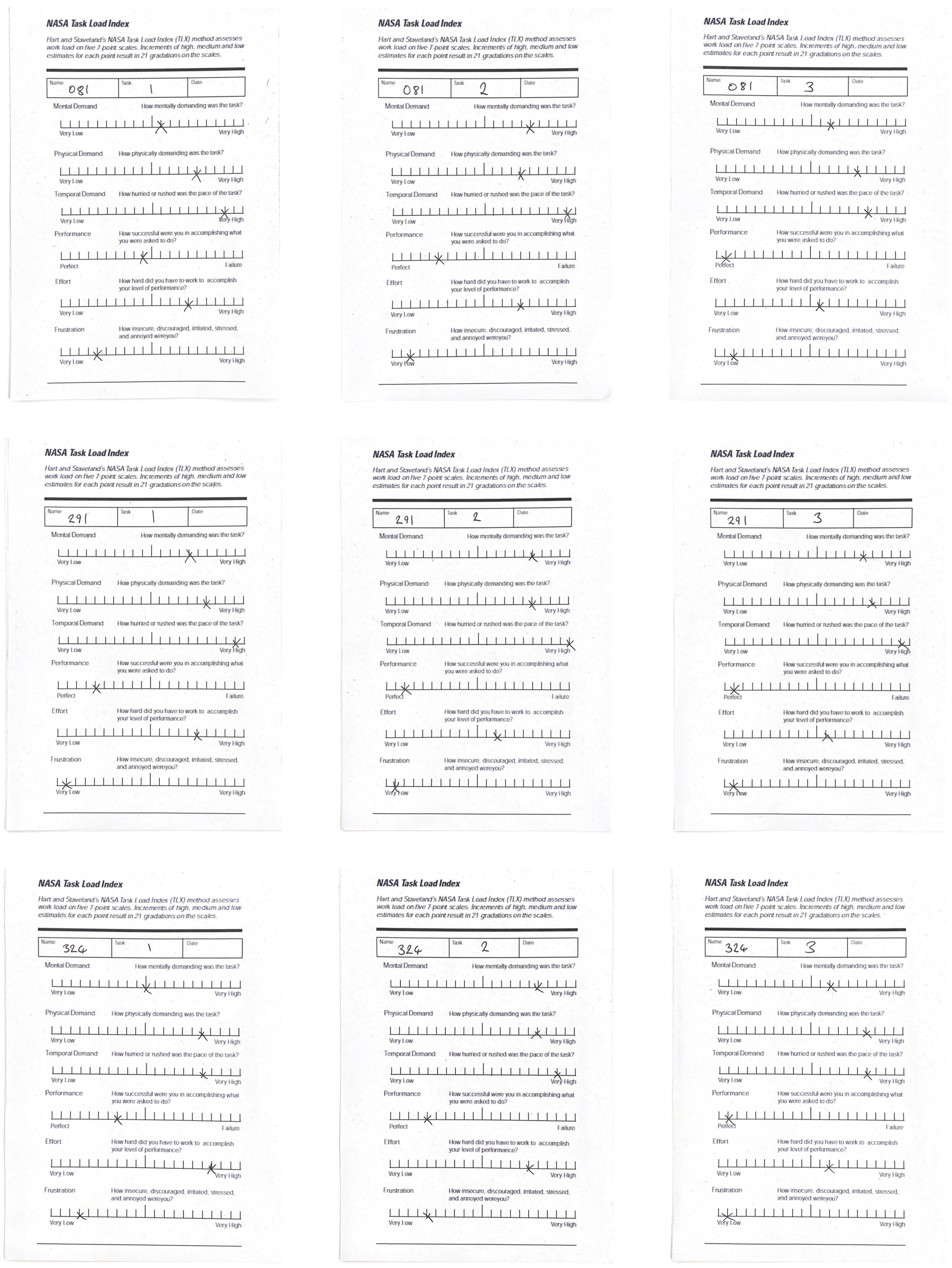
Appendix A

TLX Questionnaires

	081			324			291		
	none	audio	vibration	none	audio	vibration	none	audio	vibration
mental	11	15	12	10	16	12	14	16	15
physical	15	14	15	16	16	16	15	16	16
temporal	18	19	16	17	18	16	19	20	19
performance	9	5	1	7	4	1	4	2	1
effort	14	14	11	17	15	12	15	12	11
frustration	4	2	2	3	4	1	1	1	1

	081		324		291	
	audio dif	vibration dif	audio dif	vibration dif	audio dif	vibration dif
mental	4	1	6	2	2	1
physical	-1	0	0	0	1	1
temporal	1	-2	1	-1	1	0
performance	-4	-8	-3	-6	-2	-3
effort	0	-3	-2	-5	-3	-4
frustration	-2	-2	1	-2	0	0

Table A.1: Full TLX questionnaire results



NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
081	1	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
081	2	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
081	3	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
291	1	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
291	2	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
291	3	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
324	1	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
324	2	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date
324	3	

Mental Demand How mentally demanding was the task?
Very Low Very High
Physical Demand How physically demanding was the task?
Very Low Very High
Temporal Demand How hurried or rushed was the pace of the task?
Very Low Very High
Performance How successful were you in accomplishing what you were asked to do?
Perfect Failure
Effort How hard did you have to work to accomplish your level of performance?
Very Low Very High
Frustration How insecure, discouraged, irritated, stressed, and annoyed wereyou?
Very Low Very High

Figure A.1: Images of all TLX questionnaires

Appendix B

Key Code Listings

I've listed here the key sections of code written to support the project. These have been referenced throughout the report.

B.1 Arduino Code

```
#include <Wire.h>
#include <Adafruit_MMA8451.h>
#include <Adafruit_L3GD20_U.h>
#include <Adafruit_Sensor.h>
#include <math.h>

Adafruit_MMA8451 mma = Adafruit_MMA8451();
Adafruit_L3GD20_Unified gyro = Adafruit_L3GD20_Unified(20);

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

  Serial.println("Orientation Test!");

  if (!mma.begin()) {
    Serial.println("No MMA8451 detected!");
    while (1);
  }

  Serial.println("Initialised Accelerometer!");

  if(!gyro.begin())
  {
    Serial.println("No L3GD20 detected!");
    while(1);
  }

  Serial.println("Initialised Gyroscope!");

  mma.setRange(MMA8451_RANGE_8_G);
  gyro.enableAutoRange(true);
}
```

```
void loop() {
  // Get a new sensor event
  unsigned long time1, time2, dif;
  sensors_event_t event_acc;
  sensors_event_t event_gyro;
  float angle, pitch, roll, pitchAcc, rollAcc, pitchInertial, rollInertial;

  int count = 0;

  time1 = micros();

  while (true) {
    count += 1;

    mma.getEvent(&event_acc);
    gyro.getEvent(&event_gyro);

    float x_acc = event_acc.acceleration.y - 0.5;
    float y_acc = - event_acc.acceleration.x + 0.3;
    float z_acc = - event_acc.acceleration.z;

    float x_gyro = (event_gyro.gyro.x - 0.11) / 1000000;
    float y_gyro = (event_gyro.gyro.y) / 1000000;
    float z_gyro = -(event_gyro.gyro.z + 0.06) / 1000000;

    time2 = micros();
    dif = time2 - time1;
    time1 = micros();

    // Convert to inertial ref frame
    pitchInertial = cos(roll) * y_gyro - sin(roll) * z_gyro;
    rollInertial = x_gyro + sin(roll) * tan(pitch) * y_gyro + cos(roll) * tan(pitch) *
        ↪ z_gyro;

    // Use inertial data to increment pitch+roll
    pitch += pitchInertial * dif;
    roll += rollInertial * dif;

    // Estimate orientation from accelerometer
    pitchAcc = atan2f(y_acc, sqrt(x_acc * x_acc + z_acc * z_acc));
    rollAcc = - atan2f(x_acc, sqrt(y_acc * y_acc + z_acc * z_acc));

    // Combine gyro and accelerometer data, 98% gyro / 2% accel
    pitch = pitch * 0.98 + pitchAcc * 0.02;
    roll = roll * 0.98 + rollAcc * 0.02;

    // Convert to degrees
    float pitchDeg = pitch * 180 / M_PI;
    float rollDeg = roll * 180 / M_PI;

    Serial.print(pitchDeg); Serial.print(" "); Serial.println(rollDeg);

    // Ensure readings stay stable to begin with
    if (count < 10) {
      pitch = 0;
      roll = 0;
    }
  }
}
```

B.2 Prototype II Python Code

```
import board
import busio
import adafruit_mma8451
import adafruit_l3gd20
import RPi.GPIO as IO
from GS_timing import micros
from soundplayer import SoundPlayer
import time
import sys
from math import pi, atan2, sqrt, sin, cos, tan, sec

def print_info(info):
       print('I{}'.format(info), flush=True)

def print_data(data):
       print('D{}'.format(data), flush=True)

def print_err(error):
       print('E{}'.format(error), flush=True)

def print_file(f):
       print('F{}'.format(f), flush=True)

seconds_to_record = int(sys.argv[1])

i2c = busio.I2C(board.SCL, board.SDA)
try:
       gyro_L = adafruit_l3gd20.L3GD20_I2C(i2c, address=0x6B)
       gyro_R = adafruit_l3gd20.L3GD20_I2C(i2c, address=0x6A)
       print_info("Set up both L3GD20H")
       accel_L = adafruit_mma8451.MMA8451(i2c, address=0x1C)
       accel_R = adafruit_mma8451.MMA8451(i2c, address=0x1D)
       print_info("Set up both MMA8451")
except ValueError:
       print_err('Unable to initialise I2C devices')
       exit(1)

accel_L.range = adafruit_mma8451.RANGE_8G # +/- 8G
accel_R.range = adafruit_mma8451.RANGE_8G # +/- 8G

pitch_L, roll_L, pitchAcc_L, rollAcc_L = 0.0, 0.0, 0.0, 0.0
pitch_R, roll_R, pitchAcc_R, rollAcc_R = 0.0, 0.0, 0.0, 0.0
rad2deg = 180.0 / pi
deg2rad = pi / 180.0
us = 1000000

x_gyro_L_av, y_gyro_L_av, z_gyro_L_av = 0.0, 0.0, 0.0
x_acc_L_av, y_acc_L_av, z_acc_L_av = 0.0, 0.0, 0.0
x_gyro_R_av, y_gyro_R_av, z_gyro_R_av = 0.0, 0.0, 0.0
x_acc_R_av, y_acc_R_av, z_acc_R_av = 0.0, 0.0, 0.0
```

```
print_data("Calibrating...")
samples = 100
for i in range(samples):
        x_gyro_L, y_gyro_L, z_gyro_L = gyro_L.gyro
        x_gyro_L_av += x_gyro_L / samples
        y_gyro_L_av += y_gyro_L / samples
        z_gyro_L_av += z_gyro_L / samples

        x_acc_L, y_acc_L, z_acc_L = accel_L.acceleration
        x_acc_L_av += x_acc_L / samples
        y_acc_L_av += y_acc_L / samples
        z_acc_L_av += z_acc_L / samples

        x_gyro_R, y_gyro_R, z_gyro_R = gyro_R.gyro
        x_gyro_R_av += x_gyro_R / samples
        y_gyro_R_av += y_gyro_R / samples
        z_gyro_R_av += z_gyro_R / samples

        x_acc_R, y_acc_R, z_acc_R = accel_R.acceleration
        x_acc_R_av += x_acc_R / samples
        y_acc_R_av += y_acc_R / samples
        z_acc_R_av += z_acc_R / samples

mag_acc_L = sqrt(x_acc_L_av ** 2 + y_acc_L_av ** 2 + z_acc_L_av ** 2)
mag_acc_L_dif = mag_acc_L - abs(z_acc_L_av)

mag_acc_R = sqrt(x_acc_R_av ** 2 + y_acc_R_av ** 2 + z_acc_R_av ** 2)
mag_acc_R_dif = mag_acc_R - abs(z_acc_R_av)

print_data('Starting Recording')

start_time = time.time()
time1, time2, dif = micros(), 0, 0
orientations = []
raw_data = []
count = 0
while time.time() - start_time < seconds_to_record:
        count += 1
        try:
                x_gyro_L_raw, y_gyro_L_raw, z_gyro_L_raw = gyro_L.gyro
                x_acc_L_raw, y_acc_L_raw, z_acc_L_raw = accel_L.acceleration
                x_gyro_R_raw, y_gyro_R_raw, z_gyro_R_raw = gyro_R.gyro
                x_acc_R_raw, y_acc_R_raw, z_acc_R_raw = accel_R.acceleration
        except OSError:
                print_err('I2C disconnected!')
                break

        x_gyro_L = (x_gyro_L_raw - x_gyro_L_av) / us
        y_gyro_L = (y_gyro_L_raw - y_gyro_L_av) / us
        z_gyro_L = (z_gyro_L_raw - z_gyro_L_av) / us

        x_acc_L = y_acc_L_raw - y_acc_L_av
        y_acc_L = -(x_acc_L_raw - x_acc_L_av)
        z_acc_L = -(z_acc_L_raw + mag_acc_L_dif)
        x_gyro_R = (x_gyro_R_raw - x_gyro_R_av) / us
        y_gyro_R = (y_gyro_R_raw - y_gyro_R_av) / us
        z_gyro_R = (z_gyro_R_raw - z_gyro_R_av) / us

        x_acc_R = y_acc_R_raw - y_acc_R_av
        y_acc_R = -(x_acc_R_raw - x_acc_R_av)
        z_acc_R = -(z_acc_R_raw + mag_acc_R_dif)

        time2 = micros()
        dif = time2 - time1
        time1 = micros()

        # Gyro calcs
        pitch_L_interial = cos(roll_L) * y_gyro_L - sin(roll_L) * z_gyro_L
        roll_L_inertial = x_gyro_L + sin(roll_L) * tan(pitch_L) * y_gyro_L + cos(roll_L
            ↪ ) * tan(pitch_L) * z_gyro_L

        pitch_L += pitch_L_interial * dif
        roll_L += roll_L_inertial * dif

        pitch_R_interial = cos(roll_R) * y_gyro_R - sin(roll_R) * z_gyro_R
        roll_R_inertial = x_gyro_R + sin(roll_R) * tan(pitch_R) * y_gyro_R + cos(roll_R
            ↪ ) * tan(pitch_R) * z_gyro_R

        pitch_R += pitch_R_interial * dif
        roll_R += roll_R_inertial * dif

        # Accelerometer calcs
        pitchAcc_L = atan2(y_acc_L, z_acc_L)
        pitchAcc_L = pitchAcc_L * rad2deg
        pitch_L = pitch_L * 0.98 + pitchAcc_L * 0.02

        rollAcc_L = - atan2(x_acc_L, z_acc_L)
        rollAcc_L = rollAcc_L * rad2deg
        roll_L = roll_L * 0.98 + rollAcc_L * 0.02

        pitchAcc_R = atan2(y_acc_R, z_acc_R)
        pitchAcc_R = pitchAcc_R * rad2deg
        pitch_R = pitch_R * 0.98 + pitchAcc_R * 0.02

        rollAcc_R = - atan2(x_acc_R, z_acc_R)
        rollAcc_R = rollAcc_R * rad2deg
        roll_R = roll_R * 0.98 + rollAcc_R * 0.02

        # Calculate magnitude of sideways acceleration
        side_acc_L = -z_acc_L * cos(pi/2 - pitch_L*deg2rad) + -x_acc_L * cos(pitch_L*
            ↪ deg2rad)
        side_acc_R = -z_acc_R * cos(pi/2 - pitch_R*deg2rad) + -x_acc_R * cos(pitch_R*
            ↪ deg2rad)

        # Ensure readings stay stable to begin with
        if count < 10:
               pitch_L, roll_L, pitch_R, roll_R = 0.0, 0.0, 0.0, 0.0

        orientations.append((time2, pitch_L, pitch_R, side_acc_L, side_acc_R))
        raw_data.append((x_gyro_L_raw, y_gyro_L_raw, z_gyro_L_raw, x_acc_L_raw,
            ↪ y_acc_L_raw, z_acc_L_raw, x_gyro_R_raw, y_gyro_R_raw, z_gyro_R_raw,
            ↪ x_acc_R_raw, y_acc_R_raw, z_acc_R_raw))
```

```
print_data('Done recording')

with open('data_{t}.csv'.format(t=int(start_time)), 'w') as file:
        file.write('time,pitch_L,pitch_R,sideAcc_L,sideAcc_R\n')
        for orientation in orientations:
                file.write('{time},{pitchL},{pitchR},{sideAccL},{sideAccR}\n'.format(
                    ↪ time=orientation[0], pitchL=orientation[1], rollL=orientation[2],
                    ↪  slipL=orientation[3], slipR=orientation[4]))

# Print out raw data for debugging
with open('rawdata_{t}.csv'.format(t=int(start_time)), 'w') as file:
        file.write('x_g_L,y_g_L,z_g_L,x_a_L,y_a_L,z_a_L,x_g_R,y_g_R,z_g_R,x_a_R,y_a_R,
            ↪ z_a_R\n')
        for raw in raw_data:
                file.write('{x_g_L},{y_g_L},{z_g_L},{x_a_L},{y_a_L},{z_a_L},{x_g_R},{
                    ↪ y_g_R},{z_g_R},{x_a_R},{y_a_R},{z_a_R}\n'.format(x_g_L=raw[0],
                    ↪ y_g_L=raw[1],z_g_L=raw[2],x_a_L=raw[3],y_a_L=raw[4],z_a_L=raw[5],
                    ↪ x_g_R=raw[6],y_g_R=raw[7],z_g_R=raw[8],x_a_R=raw[9],y_a_R=raw
                    ↪ [10],z_a_R=raw[11]))

print_file('{t}'.format(t=int(start_time)))
```

B.3 Prototype V Python Code

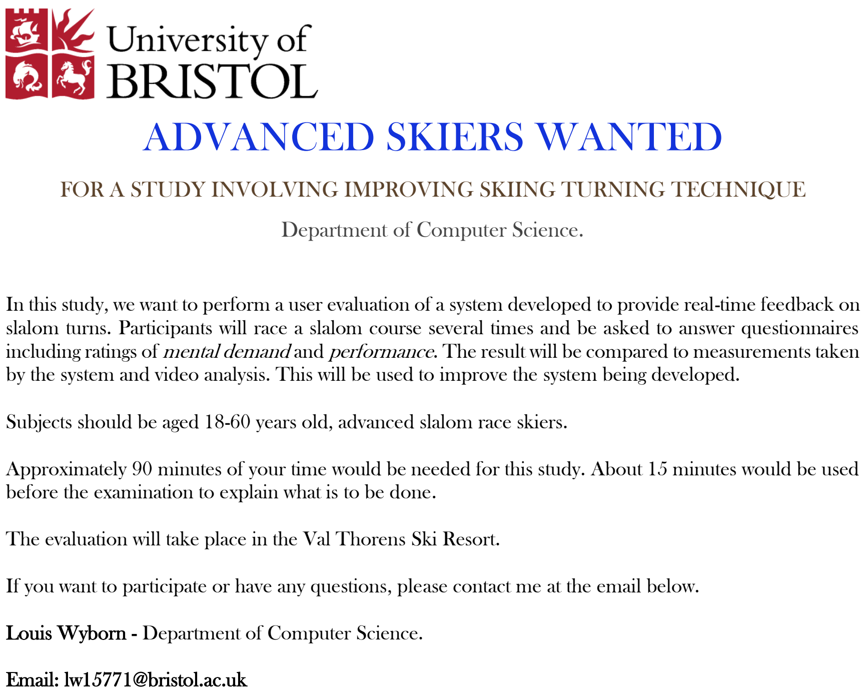
```
import board
import busio
import adafruit_bno055
import RPi.GPIO as IO
from GS_timing import micros
from soundplayer import SoundPlayer
import time
import sys
from math import pi, atan2, sqrt, cos

def print_info(info):
       print('I{}'.format(info), flush=True)

def print_data(data):
       print('D{}'.format(data), flush=True)

def print_err(error):
       print('E{}'.format(error), flush=True)

def print_file(f):
       print('F{}'.format(f), flush=True)

seconds_to_record = int(sys.argv[1])

i2c = busio.I2C(board.SCL, board.SDA)

try:
       sensor_L = adafruit_bno055.BNO055(i2c, address=0x28)
       sensor_R = adafruit_bno055.BNO055(i2c, address=0x29)
except ValueError:
       print_err('Unable to initialise I2C devices')
       exit(1)

IO.setmode(IO.BCM)
IO.setup(12, IO.OUT)
IO.setup(16, IO.OUT)
IO.setup(20, IO.OUT)
IO.setup(21, IO.OUT)
p12 = IO.PWM(12, 490)
p16 = IO.PWM(16, 490)
p20 = IO.PWM(20, 490)
p21 = IO.PWM(21, 490)
p12.start(0)
p16.start(0)
p20.start(0)
p21.start(0)

def is_calibrated(sensor):
       calib_sys, calib_gyro, calib_accel, calib_mag = sensor.calibration_status
       if calib_sys == 3 and calib_gyro == 3 and calib_accel >= 0 and calib_mag == 3:
              return True
       else:
              return False
```

```
pitch_L, roll_L, pitchAcc_L, rollAcc_L = 0.0, 0.0, 0.0, 0.0
pitch_R, roll_R, pitchAcc_R, rollAcc_R = 0.0, 0.0, 0.0, 0.0
rad2deg = 180.0 / pi
deg2rad = pi / 180.0
us = 1000000

print_data("Calibrating Left...")
while not is_calibrated(sensor_L):
       pass

print_data("Calibrating Right...")
while not is_calibrated(sensor_R):
       pass

print_data('Starting Recording')

start_time = time.time()
orientations = []

while time.time() - start_time < seconds_to_record:
       try:
              time2 = micros()
              yaw_L, pitch_L, roll_L = sensor_L.euler
              yaw_R, pitch_R, roll_R = sensor_R.euler
              x_acc_L, y_acc_L, z_acc_L = sensor_L.accelerometer
              x_acc_R, y_acc_R, z_acc_R = sensor_R.accelerometer
       except OSError:
              print_err('I2C disconnected!')
              break

       # Calculate magnitude of slip
       slip_magnitude_L, slip_magnitude_R = 0, 0

       side_acc_L = -z_acc_L * cos(pi/2 - pitch_L*deg2rad) + -x_acc_L * cos(pitch_L*
           ↪ deg2rad)
       side_acc_R = -z_acc_R * cos(pi/2 - pitch_R*deg2rad) + -x_acc_R * cos(pitch_R*
           ↪ deg2rad)

       min_acc_L = (abs(pitch_L) / 90.0) * 115
       min_acc_R = (abs(pitch_R) / 90.0) * 115

       if abs(side_acc_L) < min_acc_L:
              slip_magnitude_L = min_acc_L - side_acc_L

       if abs(side_acc_R) < min_acc_L:
              slip_magnitude_R = min_acc_R - side_acc_R

       if abs(slip_magnitude_L) > 100:
              slip_magnitude_L = 100
       if abs(slip_magnitude_R) > 100:
              slip_magnitude_R = 100

       if pitch_L < 0:
              slip_magnitude_L *= -1
       if pitch_R < 0:
              slip_magnitude_R *= -1
```

```
        # Provide vibrotactile feedback
        if slip_magnitude_L > 0:
                p12.ChangeDutyCycle(abs(slip_magnitude_L))
                p16.ChangeDutyCycle(0)
        elif slip_magnitude_L < 0:
                p16.ChangeDutyCycle(abs(slip_magnitude_L))
                p12.ChangeDutyCycle(0)
        else:
                p12.ChangeDutyCycle(0)
                p16.ChangeDutyCycle(0)

        if slip_magnitude_R > 0:
                p20.ChangeDutyCycle(abs(slip_magnitude_R))
                p21.ChangeDutyCycle(0)
        elif slip_magnitude_R < 0:
                p21.ChangeDutyCycle(abs(slip_magnitude_R))
                p20.ChangeDutyCycle(0)
        else:
                p20.ChangeDutyCycle(0)
                p21.ChangeDutyCycle(0)

        # Provide auditory feedback
        if slip_magnitude_L != 0:
                SoundPlayer.playTone(100*abs(slip_magnitude_L), 0.05, False, 0, '1 0')
        if slip_magnitude_R != 0:
                SoundPlayer.playTone(100*abs(slip_magnitude_R), 0.05, False, 0, '0 1')

        orientations.append((time2, pitch_L, pitch_R, slip_magnitude_L,
            ↪ slip_magnitude_R))

print_data('Done recording')

with open('data_{t}.csv'.format(t=int(start_time)), 'w') as file:
        file.write('time,pitch_L,pitch_R,slip_L,slip_R\n')
        for orientation in orientations:
                file.write('{time},{pitchL},{pitchR},{slipL},{slipR}\n'.format(time=
                    ↪ orientation[0], pitchL=orientation[1], pitchR=orientation[2],
                    ↪ slipL=orientation[3], slipR=orientation[4]))

print_file('{t}'.format(t=int(start_time)))
```

Appendix C

Ethics Forms



University of BRISTOL

ADVANCED SKIERS WANTED

FOR A STUDY INVOLVING IMPROVING SKIING TURNING TECHNIQUE

Department of Computer Science.

In this study, we want to perform a user evaluation of a system developed to provide real-time feedback on slalom turns. Participants will race a slalom course several times and be asked to answer questionnaires including ratings of *mental demand* and *performance*. The result will be compared to measurements taken by the system and video analysis. This will be used to improve the system being developed.

Subjects should be aged 18-60 years old, advanced slalom race skiers.

Approximately 90 minutes of your time would be needed for this study. About 15 minutes would be used before the examination to explain what is to be done.

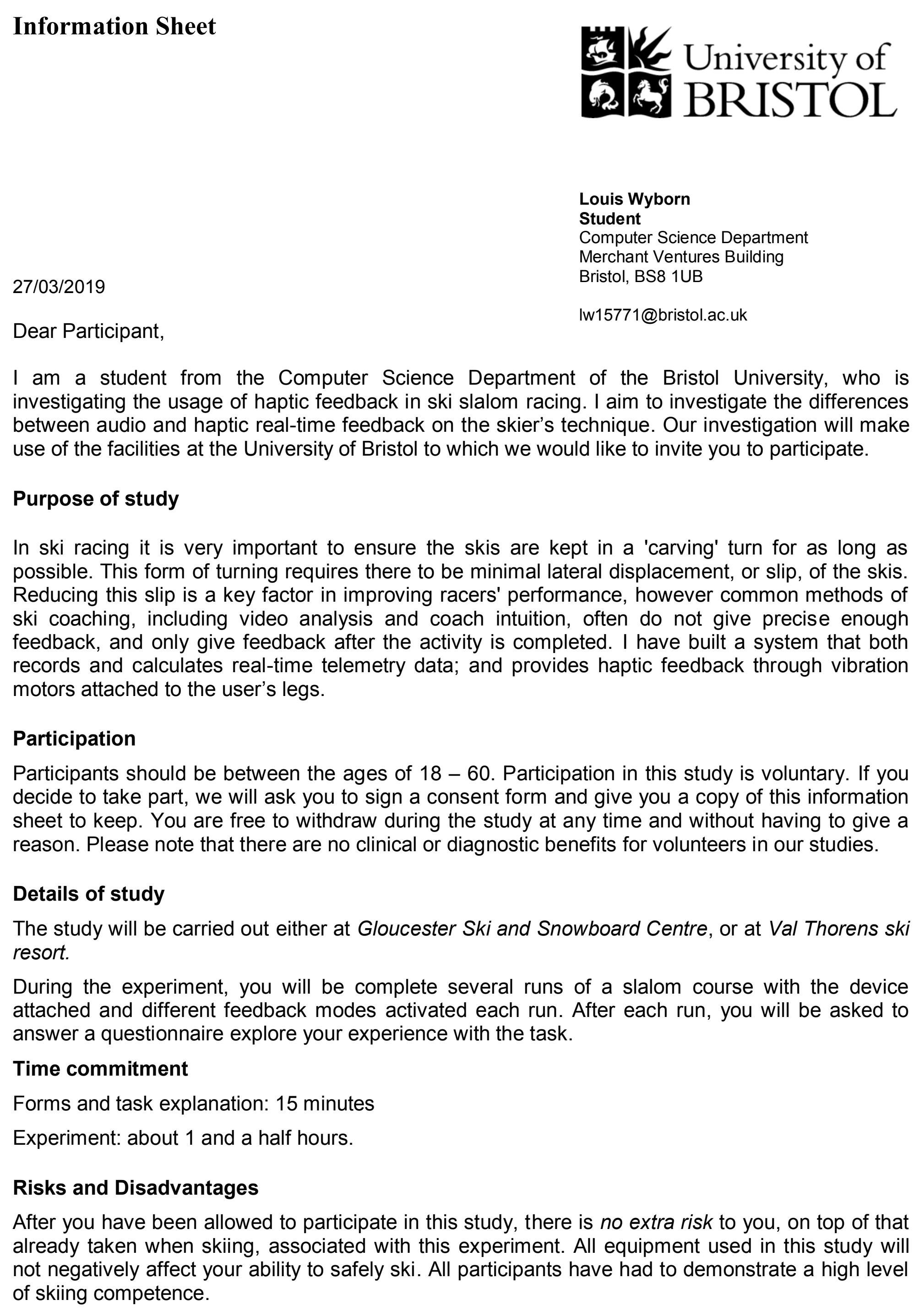
The evaluation will take place in the Val Thorens Ski Resort.

If you want to participate or have any questions, please contact me at the email below.

Louis Wyborn - Department of Computer Science.

Email: lw15771@bristol.ac.uk

Figure C.1: Call for participants poster



Information Sheet

Louis Wyborn
Student
Computer Science Department
Merchant Ventures Building
Bristol, BS8 1UB

lw15771@bristol.ac.uk

27/03/2019

Dear Participant,

I am a student from the Computer Science Department of the Bristol University, who is investigating the usage of haptic feedback in ski slalom racing. I aim to investigate the differences between audio and haptic real-time feedback on the skier's technique. Our investigation will make use of the facilities at the University of Bristol to which we would like to invite you to participate.

Purpose of study

In ski racing it is very important to ensure the skis are kept in a 'carving' turn for as long as possible. This form of turning requires there to be minimal lateral displacement, or slip, of the skis. Reducing this slip is a key factor in improving racers' performance, however common methods of ski coaching, including video analysis and coach intuition, often do not give precise enough feedback, and only give feedback after the activity is completed. I have built a system that both records and calculates real-time telemetry data; and provides haptic feedback through vibration motors attached to the user's legs.

Participation

Participants should be between the ages of 18 – 60. Participation in this study is voluntary. If you decide to take part, we will ask you to sign a consent form and give you a copy of this information sheet to keep. You are free to withdraw during the study at any time and without having to give a reason. Please note that there are no clinical or diagnostic benefits for volunteers in our studies.

Details of study

The study will be carried out either at *Gloucester Ski and Snowboard Centre*, or at *Val Thorens ski resort.*

During the experiment, you will be complete several runs of a slalom course with the device attached and different feedback modes activated each run. After each run, you will be asked to answer a questionnaire explore your experience with the task.

Time commitment

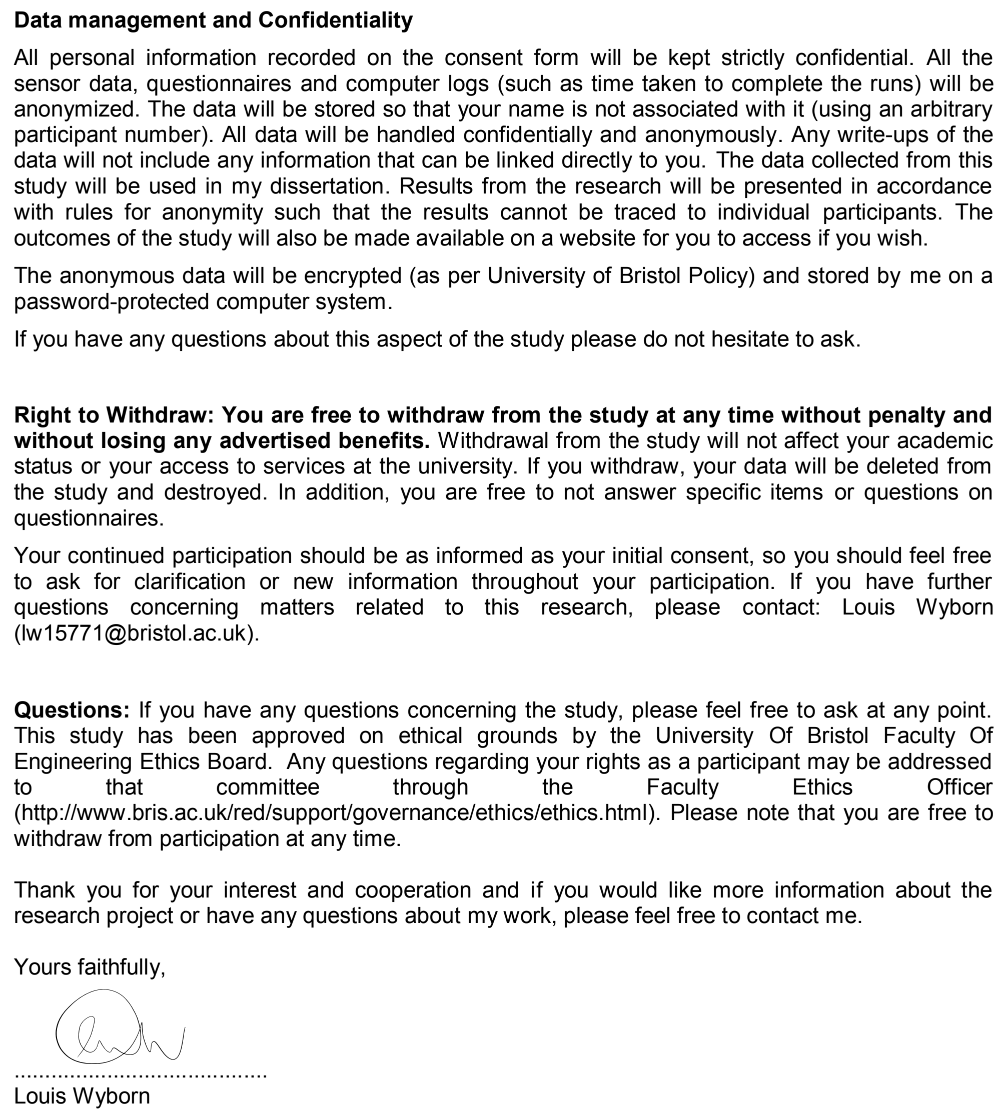
Forms and task explanation: 15 minutes

Experiment: about 1 and a half hours.

Risks and Disadvantages

After you have been allowed to participate in this study, there is *no extra risk* to you, on top of that already taken when skiing, associated with this experiment. All equipment used in this study will not negatively affect your ability to safely ski. All participants have had to demonstrate a high level of skiing competence.

Figure C.2: Participant information page 1



Data management and Confidentiality

All personal information recorded on the consent form will be kept strictly confidential. All the sensor data, questionnaires and computer logs (such as time taken to complete the runs) will be anonymized. The data will be stored so that your name is not associated with it (using an arbitrary participant number). All data will be handled confidentially and anonymously. Any write-ups of the data will not include any information that can be linked directly to you. The data collected from this study will be used in my dissertation. Results from the research will be presented in accordance with rules for anonymity such that the results cannot be traced to individual participants. The outcomes of the study will also be made available on a website for you to access if you wish.

The anonymous data will be encrypted (as per University of Bristol Policy) and stored by me on a password-protected computer system.

If you have any questions about this aspect of the study please do not hesitate to ask.

Right to Withdraw: You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. In addition, you are free to not answer specific items or questions on questionnaires.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact: Louis Wyborn (lw15771@bristol.ac.uk).

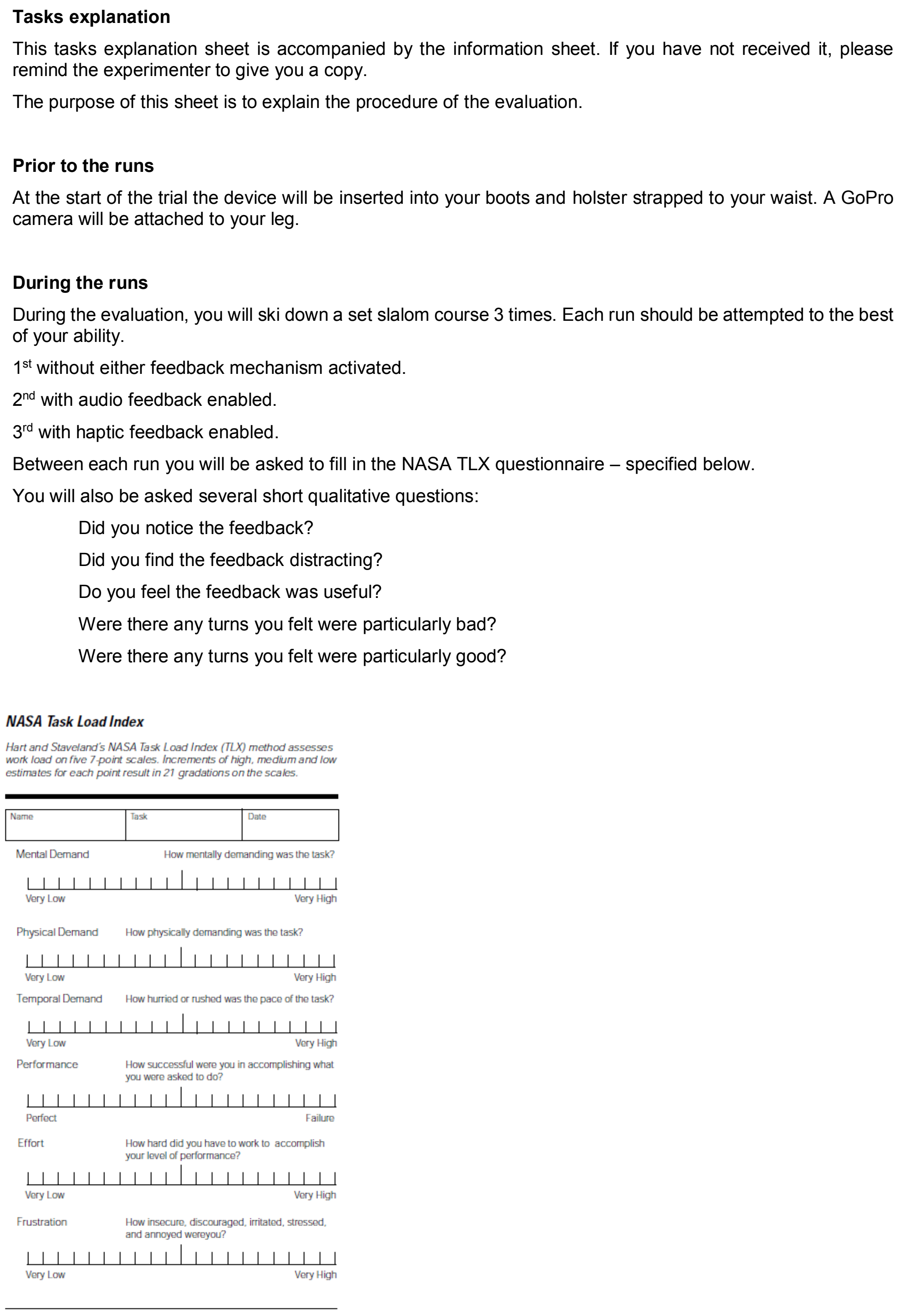
Questions: If you have any questions concerning the study, please feel free to ask at any point. This study has been approved on ethical grounds by the University Of Bristol Faculty Of Engineering Ethics Board. Any questions regarding your rights as a participant may be addressed to that committee through the Faculty Ethics Officer (http://www.bris.ac.uk/red/support/governance/ethics/ethics.html). Please note that you are free to withdraw from participation at any time.

Thank you for your interest and cooperation and if you would like more information about the research project or have any questions about my work, please feel free to contact me.

Yours faithfully,

[illegible]
....
Louis Wyborn

Figure C.3: Participant information page 2



Tasks explanation

This tasks explanation sheet is accompanied by the information sheet. If you have not received it, please remind the experimenter to give you a copy.

The purpose of this sheet is to explain the procedure of the evaluation.

Prior to the runs

At the start of the trial the device will be inserted into your boots and holster strapped to your waist. A GoPro camera will be attached to your leg.

During the runs

During the evaluation, you will ski down a set slalom course 3 times. Each run should be attempted to the best of your ability.

1st without either feedback mechanism activated.

2nd with audio feedback enabled.

3rd with haptic feedback enabled.

Between each run you will be asked to fill in the NASA TLX questionnaire – specified below.

You will also be asked several short qualitative questions:

Did you notice the feedback?

Did you find the feedback distracting?

Do you feel the feedback was useful?

Were there any turns you felt were particularly bad?

Were there any turns you felt were particularly good?

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

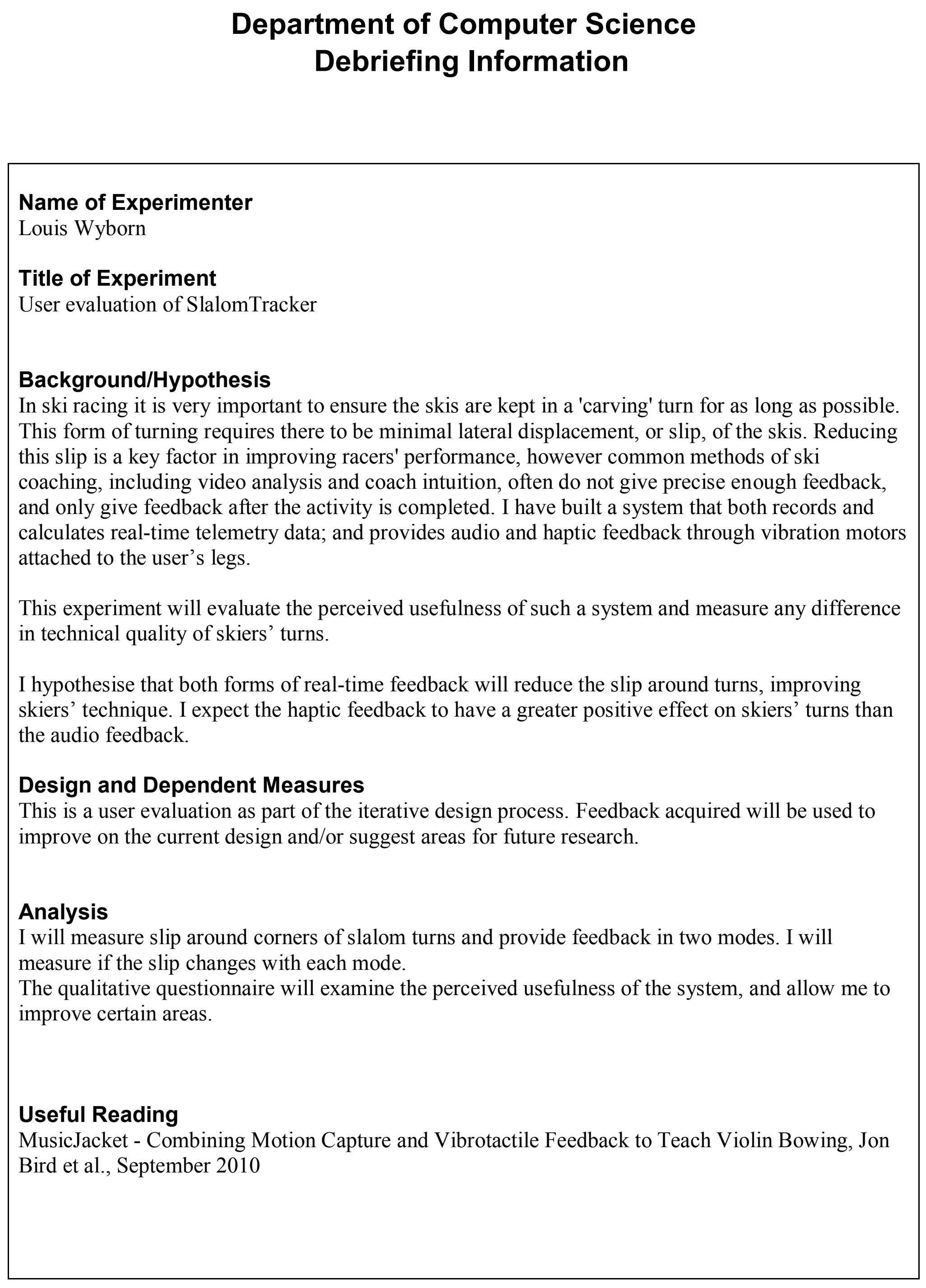
Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure C.4: Task explanation



Department of Computer Science
Debriefing Information

Name of Experimenter
Louis Wyborn

Title of Experiment
User evaluation of SlalomTracker

Background/Hypothesis
In ski racing it is very important to ensure the skis are kept in a 'carving' turn for as long as possible. This form of turning requires there to be minimal lateral displacement, or slip, of the skis. Reducing this slip is a key factor in improving racers' performance, however common methods of ski coaching, including video analysis and coach intuition, often do not give precise enough feedback, and only give feedback after the activity is completed. I have built a system that both records and calculates real-time telemetry data; and provides audio and haptic feedback through vibration motors attached to the user’s legs.

This experiment will evaluate the perceived usefulness of such a system and measure any difference in technical quality of skiers’ turns.

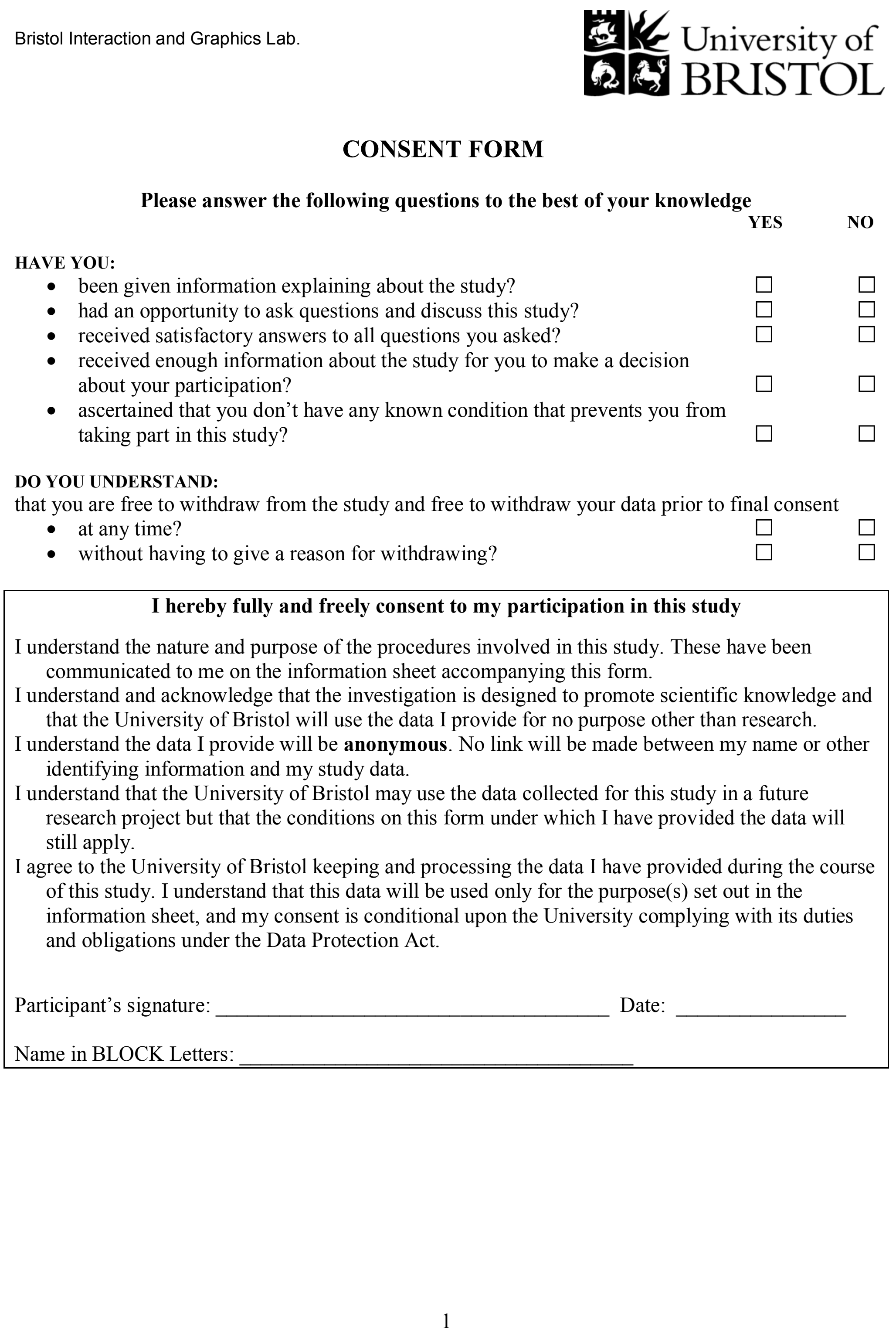
I hypothesise that both forms of real-time feedback will reduce the slip around turns, improving skiers’ technique. I expect the haptic feedback to have a greater positive effect on skiers’ turns than the audio feedback.

Design and Dependent Measures
This is a user evaluation as part of the iterative design process. Feedback acquired will be used to improve on the current design and/or suggest areas for future research.

Analysis
I will measure slip around corners of slalom turns and provide feedback in two modes. I will measure if the slip changes with each mode.
The qualitative questionnaire will examine the perceived usefulness of the system, and allow me to improve certain areas.

Useful Reading
MusicJacket - Combining Motion Capture and Vibrotactile Feedback to Teach Violin Bowing, Jon Bird et al., September 2010

Figure C.5: Debrief document



Bristol Interaction and Graphics Lab.

University of BRISTOL

CONSENT FORM

Please answer the following questions to the best of your knowledge

	YES	NO
HAVE YOU:		
• been given information explaining about the study?	☐	☐
• had an opportunity to ask questions and discuss this study?	☐	☐
• received satisfactory answers to all questions you asked?	☐	☐
• received enough information about the study for you to make a decision about your participation?	☐	☐
• ascertained that you don't have any known condition that prevents you from taking part in this study?	☐	☐
DO YOU UNDERSTAND:		
that you are free to withdraw from the study and free to withdraw your data prior to final consent		
• at any time?	☐	☐
• without having to give a reason for withdrawing?	☐	☐

I hereby fully and freely consent to my participation in this study

I understand the nature and purpose of the procedures involved in this study. These have been communicated to me on the information sheet accompanying this form.

I understand and acknowledge that the investigation is designed to promote scientific knowledge and that the University of Bristol will use the data I provide for no purpose other than research.

I understand the data I provide will be **anonymous**. No link will be made between my name or other identifying information and my study data.

I understand that the University of Bristol may use the data collected for this study in a future research project but that the conditions on this form under which I have provided the data will still apply.

I agree to the University of Bristol keeping and processing the data I have provided during the course of this study. I understand that this data will be used only for the purpose(s) set out in the information sheet, and my consent is conditional upon the University complying with its duties and obligations under the Data Protection Act.

Participant's signature: ____________________________________ Date: _______________

Name in BLOCK Letters: ____________________________________

1

Figure C.6: Participant consent form