RHapTor: Rendering Haptic Feedback with Torques for Virtual Reality

Student Name: K. Roice
Supervisor Name: G. A. Koulieris
Submitted as part of the degree of BSc Computer Science and Physics to the Board of Examiners in the Department of Computer Sciences, Durham University

Abstract—

Context Today's virtual reality systems show huge leaps in audio-visual immersion over the last decade, but lack a depth of interaction with other senses. Recent works have identified haptic engagement of sensations (like vibrations, balance and heaviness) is possible, but their software control and rendering has not yet been fully explored.

Aims The aim of this project is to investigate the extent to which we can generate such feedback using recently identified gyroscopic torque effects on head mounted displays. We aim to assess how the algorithmic control of this rendering modality can utilise sequences of simple actuations to enhance and guide the user's perception of the virtual environments they are in, as well as the value such haptic engines add to the user experience.

Method We achieve this by augmenting a virtual reality headset to include two motor-driven discs, which induce a controllable gyroscopic torque on the wearer's head. We then present four torque rendering algorithms and design appropriate environments to reflect the capabilities of these effects. These algorithms integrate together in a rendering pipeline architecture.

Results We develop a device capable of producing net gyroscopic torques up to 0.23 Nm on the user's head. At least 75% of our users were always found to experience a form of resistance matching that rendered algorithmically. All our participants showed a noticeably different exploration profile when haptic rendering was turned on, and we were successfully able to guide users attention to specific objects in the environment by rendering resistive torques. Power consumption and noise levels were both lower than similar works in the field.

Conclusion Haptic feedback is a very appropriate modality for virtual reality systems, and torque based haptics have been demonstrated to render natural sensations other technologies cannot. A software control over this recent feat of hardware unlocks a new dimension of interactive functionality.

Index Terms—Force feedback, Haptic rendering, Kinaesthetic devices, Virtual reality

1 INTRODUCTION

VIRTUAL Reality (VR) is a high-end user-computer interface that involves real-time simulation and interactions through multiple sensory channels (Burdeau and Coiffet 2003). It is motivated by the immersion of the user into an environment designed to mimic the experiences of a synthetic reality for the user, by rendering appropriate stimuli to the user's senses. This work investigates the use of rotational inertia from gyroscopic torques on a VR headset as a means of haptic feedback to augment the VR experience.

1.1 Background and Motivation

The first ideas of what would later be called VR systems were developed by Ivan Sutherland in 1963 (Sutcliffe 2003). He envisioned a human-computer interaction (HCI) system that supplied information to "*as many senses as possible*" (Sutherland 1965). The schematics over recent years have remained vastly unchanged, with most commercial VR devices embodying a Head Mounted Display (HMD) with eyepieces for stereoscopic visual stimulation to either eye (Desai et al. 2014). Over the decades, various paradigms have emerged to characterise the essence and end goal of VR as field. Burdea had distilled the main motivation of this subject into the 3 I's: Immersion-Interaction-Imagination, and hardware advancements should aim to heighten the users' experience of all three (Burdea 1993).

In the pursuit of a synthetic reality, the fields of VR and HCI have made strides in hardware for visual and auditory immersion. Modern VR systems typically rely on a combination of near-eye displays with compact Fresnel lenses and stereophonic audio to immerse the user visually and acoustically within the environment. While such systems are capable of fabricating an experience of an alternate reality much better than the technologies from a couple decades ago (Aas 2012), studies have long shown that forms of haptic engagement can effectively enrich this experience and successfully be used as a means of communication (Brown, Brewster, and Purchase 2005). It should come as no surprise that after the success in visual and auditory engagement, haptics are the next natural step forward in the roadmap of Burdea's 3 I's.

Another key factor driving the development and need of haptics is an authentic '*metaverse*' experience. A metaverse

– a term first used in Neal Stephenson's 1992 novel 'Snow Crash' to describe an online environment where users could socialise, be entertained and even conduct business (Ondrejka 2004) – by definition demands extensive levels of realism, complexity and most importantly modality to reflect Burdea's 3 I's to an unprecedented extent. This is backed by the wealth of research into the psychological effects (Aas 2012) and HCI challenges (Brooks 1999) that stem from virtual worlds. The need for haptic engagement in this regard has been long recognised in academia, and has recently began becoming adopted by several industries in this sector (see section 2).

Having established the evident need and relevance of haptics for the progress of VR and HCI, it is worth introducing what biological stimuli characterise haptic feedback. Haptic originates from the Greek word '*Haptikos*' for the sensation of touch or physical contact. These stimuli rely on receptors in the human body on organs that can also manipulate the world - giving haptic technologies the unique advantage of being an intuitive input and output device. In fact, Gibson even went as far as to compare the act of active touching to a tactile equivalent of ocular scanning for vision (Gibson 1962).

While haptics have classically been divided into two sub-categories: tactile and kinaesthetic haptics (G. Kim 2005), recent developments in the field have distinguished a third sub-category: proprioceptive haptics (Sherman and Craig 2018), each outlined as follows:

1) *Tactile*: Haptics perceived via sensors residing in the skin and relays information regarding more refined and dynamic pressure information (Juo et al. 2020).
2) *Kinaesthetic*: Haptics that rely on mechanoreceptors in the muscle tendons to detect information regarding resistance and tension (Juo et al. 2020).
3) *Proprioceptive*: Haptics that transduce a force that provides a sense of limb movement and muscular resistance (Jerald 2016).

Tactile haptics relies on the cutaneous mechanoreceptors on our skin, as they are sensitive to both short term and long term changes in mechanical pressure - for example vibrational feedback and surface contact respectively (Lederman and Klatzky 2009). This allows this modality to be capable of sending information about both events (like most modern haptic devices for notifications etc.) and textures, roughness or smoothness of a surface.

Kinaesthesia on the other hand relies on the perception of movement and strain from muscles, tendons and joints (Sherman and Craig 2018), and should not be confused with proprioception which stems from our awareness of relative positions of body parts - e.g. being able to point at your nose with your eyes closed. While most implementations of kinaesthetic haptics have been used for actions like limb movements and grasping, historically not much work has been done in the development of kinaesthetic haptics beyond the limbs and torso joints. This is because generating kinaesthetic feedback for regions like the head would also affect perceptions of balance and orientation by the user's vestibular system (Khan and Chang 2013), however our work views this as an advantage, if controlled and applied correctly. It has become commonplace to implement tactile haptics in the arm and hand region, which is understandable for tactile haptics given the high mean thresholds of 2-point touch (i.e the minimum distance your skin is able to distinguish two points from eachother by touch) and point localisation thresholds (the body's resolution of a point of touch) identified early on in the literature (see Fig 1).

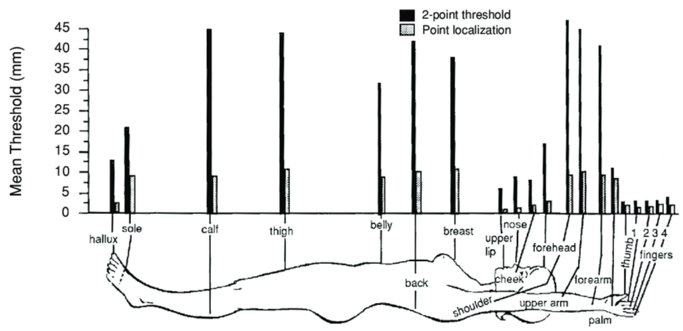

Fig. 1. The variation in two-point touch and point-localisation thresholds across the human body (Lederman 1991).

However, the industry seems to have adopted this same positioning practice from tactile haptics for kinesthetic haptics without exactly questioning why - especially given the distribution of gliding joints over the body in regions like the wrists, ankles and, in this work, the neck and head. This gap in the haptic feedback technology market has only recently been questioned and explored by devices which we will cover in sub-section 2.3.

Although the human nervous system naturally handles combinations of tactile and kinaesthetic inputs in its day-to-day functioning (even for something as simple as holding an object), modern VR systems rarely combine the two. For example, commercial haptics have existed since the 2000's video game controllers, but until recently, they have only been restricted to vibro-tactile feedback (Willumsen and Jacevic 2019).

Furthermore, as highlighted by Delazio et al.'s research into tactile feedback jackets for the torso, many modern haptics fail to deliver distributed and sustained forces (Delazio et al. 2018). Limiting haptic technology to just vibrations would tremendously narrow down the number of real world experiences a wearable haptic can simulate. Hence, the combination of these two forms of haptics may create a depth of immersion unprecedented in commercial HMDs (Sherman and Craig 2018) and would heighten the perception of the 3 I's of VR (Burdea 1993).

In fact, during a Keynote in 2015, Oculus chief scientist Michael Abarsh addressed the core role of haptics for interactions with any environment, however such modalities were absent in any Oculus headset as Abarsh deemed the technology to still be in an "*embryonic*" stage (Oculus 2015).

It is this rising traction and adoption of HMDs that fostered a new breed of haptic technologies: inertia and torque based haptics, in particular those involving flywheels. This was first demonstrated by Gugenheimer et al. in their 2016 paper where they debut *GyroVR*, an Oculus Rift DK2 headset augmented with rotating discs capable of generating resistance to head and neck motion through gyroscopic effects (Gugenheimer et al. 2016). This inspired a host of other works for HMDs exploring this novel implementation of the lesser explored kinaesthtic and proprioceptive haptics, which shall be discussed in the next section. Similarly, *GyroVR* had motivated the implementation of gyroscopic torque actuation in this work, and served as a foundation for our software developments to maximise the potential this form of haptic rendering has to offer.

Across all the sectors VR has trickled into over the last few decades, first time users engaging with the novel experience quickly realise and appreciate the true scope and depth of immersion of VR. While VR apps can connect people across the globe and allow users to live through a lifetime of varied experiences in a matter of minutes, it becomes quickly apparent that visual and auditory technology is not enough for an application to truly capture the user with Burdea's three I's . The need for a more realistic VR experienece coupled with new haptic technologies still in their infancy provided amenable grounds for this work to investigate their use with commercial headsets, like the Oculus Rift S.

1.2 Research Question and Objectives

Having established the motivation for haptic technologies and the trajectory of VR, we are well-equipped to reason a research question: *To what extent can torque-based haptics feasibly add value to the VR Experience?* This question succinctly lends itself to a set of objectives that we tackle in this work, through a combination of hardware and software implementations

Our first objective is to implement and study the use of wearable haptics beyond the standard vibrational feedback that is ubiquitous in everyday devices, i.e. looking at kinaesthetic haptics in particular. Over the last few years, an increasing number of papers have been implementing this through inertia-based technology, which rely on convincing your sense of balance (handled by your body's vestibular system (Lawson and Riecke 2014)) that the reality your body is seeing and hearing is also the one it is experiencing in terms of balance and spatial orientation. While other head-based inertial haptic engines will be outlined in the next subsection, *GyroVR* formed the basis for our work and prompted this research question to extends to an algorithmic challenge. Any investigation into the use of inertia-based haptics to add value to the VR experience would require a well understood implementation on both the hardware and software fronts that both tackle this gap in the field and creates useful stimuli to the body, which this work delivers.

The next objective is to investigate the versatility of such implementations. Modern haptic devices often have just a single purpose: like outputting a vibrational actuation at a given response. However, as previously mentioned, most of our day to day experiences are a combination of the different types of haptic sensations covered in the last two sections. Thus, we consider the use of software to algorithmically control a single hardware implementation to produce a multitude of the haptics outlined in the last section. Given the trajectory of VR from the last sub-section, haptic engagement is a problem that would need to be solved, and this work investigates software-based techniques of efficiently achieving this.

Like most research into VR and HCI, our research question is very user-centric. So substantially gauging whether our haptic rendering has added value to the user experiences demands us to aim to design feedback-appropriate environments and pursue a comprehensive study and analysis of the user behaviour and response when subjected to our haptic rendering engine and algorithms. We aim to use these metrics and motifs of human perception of torques and inertial resistances in natural interactions to guide our development, analysis and, most importantly, evaluation of the performance of how haptic rendering framework in enriching the VR experience.

1.3 Achievements and Contributions

We first developed a rendering device mounted onto an Oculus Rift S that uses a single axis motor to render gyroscopic torques. Once we were able to run this haptic engine and detect perceivable gyroscopic torques, we then upgraded it to a two axis system since this would give greater software dexterity, control and freedom for the rendering and pipelining segment of our project, along with a wider variety of effects.

These include the software advancements we made to render a multitude of haptic effects through a total of 3 implemented algorithms and 1 further proposed algorithm. This was achieved through our development of a haptic rendering pipeline, built on top of Unity and as a software framework on our Arduino microcontroller. These effects were then suitably enhanced using three VR environments: a beach with a coastal breeze, a museum rendering an artefact on display and a rocky planet with falling meteors.

We were able to successfully meet all our objectives and get a better understanding of the scope of inertia torque based haptic feedback on the VR experiences, taking us a step further in the direction of our research question with both quantitative and qualitative results. We measured the impact of these effects through technical studies and user trials, gaining metrics we can use to compare with the sparse range of devices in this recent but niche section of the literature. We were able to render a torque that users could perceive in no less than 75% of our trials. Furthermore, the haptic rendering of resistance showed a statistically significant shift in the user exploration profile for the same environment across all users in our participant pool, with haptic rendering resistance inducing an increasingly Gaussian exploration distribution.

Finally, we were able to establish that haptic rendering can in fact be used in VR systems to a much larger extent than that portrayed in today's literature, through the use of clever algorithms and a rendering pipeline and software development framework, that will one day reach the same level of robustness as that of audio-visual rendering in computer graphics and acoustics.

2 Related Work

There have been a host of wearable implementations in recent years from both academia and industry that strive to enrich the VR experience by engaging more senses. Wearable devices are characterized as light-weight sensor-based devices worn close to or on the skin's surface to detect, analyse and transmit information on internal and external variables to another device (Düking et al. 2016). Consequently, they have an inherent advantage for incorporating haptic technologies - both tactile and kinaesthetic. The unique predisposition of such an implementation paradigm lends itself well to today's challenges in realistic VR. Accordingly, there has been a wealth of related investigations and literature on the hardware and software accomplishing these various implementations. This section aims to provide a broad overview of these technologies before narrowing in on implementations that share characteristics and functionality with this work (i.e. head-mounted and/or gyroscopic haptics).

Much like haptic technologies, wearable devices have been gaining increasing popularity over the last few decades in terms of both price-appeal and satisfaction levels (Kaewkannate and S. Kim 2016), particularly with smartphone integration. Additionally, the several industries outside of VR - including medicine (Haghi, Thurow, and Stoll 2017), fitness (Henriksen et al. 2018) , finance (Borowski-Beszta and Polasik 2020) - have already began to appreciate and utilise the versatility and practicality of wearable technologies even up to the commercial level.

2.1 Wearabale Haptic Feedback

The expansion in ubiquity of wearable devices coupled with the rise and necessity of haptics for advancements in VR has created the ideal environment to foster developments of novel wearable haptics in the last few decades of academic and industrial research. Early implementations of wearable haptic devices were built around the hands or forearms, which makes sense given the dexterity and sensitivity of human mechanoreceptors (see Fig. 1). This included the *PHANTOM* haptic interface (Massie, Salisbury, et al. 1994), which has been regarded as one of the first investigations into the mechanics and feasibility of tactile haptic feedback. As depicted in Fig 2 (left), the *PHANTOM* consisted of a thimble housing a force interface capable of tracking fingertip motion and rendering forces that were interpreted as convincing tactile experiences by the user. Similarly, the *Rutgers Master II*, (see Fig 2 (middle)) was a compact glove capable of providing resistances to finger motions - arguably being one of the first examples of proprioceptive and resistive feedback (Bouzit et al. 2002)!

Fig. 2. The evolution of early handheld haptic devices that render resistances. Left to Right: *PHANTOM* (Massie, Salisbury, et al. 1994), *Rutgers Master II* (Bouzit et al. 2002), *NormalTouch*(Benko et al. 2016)

Besides these examples pioneering HCI and VR, the recent boom in wearable technology to engage users further corroborates the aforementioned ideal environment for haptic development. Technologies like the *NormalTouch* and *TextureTouch* (Benko et al. 2016) prove the extent of high fidelity 3D tactile haptic rendering using actuated pins (shown in Fig 2 (right), albeit with certain drawbacks in wearability. On the other hand, works like Delazio et al.'s previously mentioned *Force Jacket* deliver pneumatically-actuated airbags and force sensors, along with the necessary force control algorithms to provide both high-frequency vibro-tactile responses as well as directed pressure to the torso (Delazio et al. 2018). Such creative academic ventures in the literature have historically been vital as they go on to inspire, verify and even aid the endorsement and funding of similar ventures in the industry, such as *HaptX*, a hardware manufacturing startup committed to bringing the haptic feedback glove experience to the consumer (Needleman 2018).

Wearable technology as a whole has been thriving over the last decade, and it is this creativity and exploration that has given rise to the category of wearable haptic devices that forms the crux of this work. We shall explore this promising field next.

2.2 Gyroscopic Haptics

Research into gyroscopic haptic rendering has historically been much more sparse than, say, vibrational technology. Consequently, their commercial adoption is less widespread.

The *GyroCube* is a palmtop ungrounded torque rendering device which is not 'wearable' in the sense that it attaches onto the user's body, but is a cube grasped in the user's hand (Sakai, Fukui, and Nakamura 2003). The cube itself houses three rotating brass discs and their associated motors, capable of generating a net angular momentum in any direction in 3D space. Sakai, Fukui, and Nakamura found that the minimum torque a user could sense from their palm was approximately 0.02 Nm, and the work had strong potential for haptic navigation systems.

Similarly, the *TorqueScreen* used gyroscopic actuators to impart angular momentum into metal flywheels (Murer et al. 2015). Instead of attaching onto the user's body, the device was fixed onto handheld devices such as smartphones or tablets, as pictured in Fig 3 (left). This came with the added advantage of integrating with the handheld devices own inertial measurement unit (IMU) to render ungrounded kinaesthetic feedback for on-screen events with

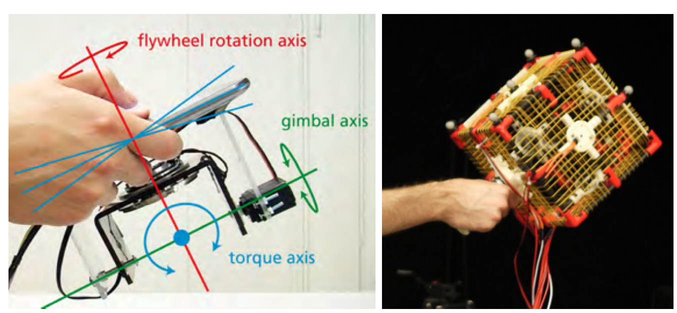

Fig. 3. Examples of gyroscopic haptic rendering: *TorqueScreen* (left) taken from Murer et al., and *Thor's Hammer* (right) taken from Heo et al.

low latency. Such integration with existing commercial devices is becoming increasingly commonplace, as we had previously mentioned with *GyroVR*, and relates well to the head-tracking developments in this work that we shall describe in subsection 3.3. Walker et al.'s paper on haptic guidance systems uses 2 double gimbal control moment gyroscopes for ungrounded kinaesthetic feedback, again with flywheel rotations (Walker et al. 2017). The build of this device allowed for integration with handheld controllers used by all major VR headset manufacturers as well as scope for increased software functionality through the successful rendering of asymmetric impulsive moments. The paper found a 99.3% success rate in users correctly identifying the direction of rendered torque. Users were also able to proprioceptively align orientations rendered with less than $7.5°$ error. These promising results for guidance systems reinforce and corroborate speculative use cases from previous works like *GyroCube*.

A more recent example of gyroscopic haptics adding value and engagement to the VR experience is *Thor's Hammer*. This ungrounded device, shown in Fig 3 (right), is capable of generating both force feedback as well as gyroscopic effects using a triple axis propeller system along with electro-muscular stimulation (Heo et al. 2018). This hardware concept embodying a combination of force and torque feedback was used to render flowing water, tension and even reaction forces to interactions of the hammer with the virtual environment. Heo et al. even went as far as to designing a simple force control application programming interface (API) on Unity to actuate forces, which inspired much of the software undertakings and explorations used to investigate torque-based feedback in this work.

Besides implemntations, a significant amount of work has been done in the litertaure looking into the theoretical analysis of the mechanics and pre-requisites for gyroscopic haptic rendering. A recent work by Tremblay-Bugeaud, Laliberté, and Gosselin investigated the theoretical and experimental generation of rotational inertia using 3 masses in a gyroscopic structure (Tremblay-Bugeaud, Laliberté, and Gosselin 2020). While a Newtonian analysis revealed theoretical feasibility, the work comments that the bulk of the challenges emerges from implementation hurdles that make it difficult for the software to communicate without drifting away from the theoretical framework.

2.3 Head-Mounted Implementations

This subset of haptic rendering strongly holds modern relevance as much of commercial VR is centered around HMDs. In fact, inertia-based kinaesthetic haptics for VR was first published for head-mounted implementations (Gugenheimer et al. 2016). Since inertia based haptics form the heart of this work, visiting recent achievements in this sector is worthwile and guided much of the direction this project pursued and challenges we tackled.

GyroVR first demonstrated kinaesthetic feedback for VR use cases such as flying, diving, first person shooter games and motion in lowered gravitational fields (Gugenheimer et al. 2016). Gugenheimer et al. found the same hardware implementation to work well with a range of algorithmic controls in multiple 3D Unity environments. The feedback itself was rendered through overclocked hard drive disks attached to an Oculus Rift DK2 headset, as well as a modular version that could be attached to most regions of the body. The paper revealed the device provided the best effects of immersion with the head with users reportedly being able to sense gyroscopic feedback but unable to discern the axis of rotation. These insights guided the foundational premise this work investigated and furthered. Notably, participants in Gugenheimer et al.'s study showed no significant change in simulator sickness levels with haptic rendering on or off, with participants reporting in-game vection and loose fastening of the headset as the main causes of any fatigue or sickness. *Kabuto* was another work that soon emerged after *GyroVR* that also used flywheels to actuate kinaesthtic haptic rendering, but at different orientations and a more symmetric arrangement as shown in Fig 4 (Tanichi et al. 2020). However, by incorporating an additional mechanical braking mechanism using servo motors, the device was able to create the impression of impulsive events such as collisions on the head. *Kabuto* also had one of the largest sample sizes of participants in this section of the literature (with over 100 people trying it at a demonstration), and Tanichi et al. consistently observed a more aggressive upper body movement in users when haptic rendering was switched on.

The advancements of *GyroVR* in this area of inertia based haptics motivated the recent *Odin's Helmet*, a 4 head-mounted propellers system capable of creating forces on the head to manipulate a user's vestibular system's perception of orientation, balance and acceleration (Hoppe et al. 2021).

Fig. 4. Recent head-mounted disc-based gyroscopic haptic devices. Left to Right: *GyroVR* single axis, *GyroVR* triple axis from Gugenheimer et al. 2016, *Kabuto* from Tanichi et al. 2020 and *Odin's Helmet* from Hoppe et al. 2021.

Each motor seen in Fig 4 was able to safely generate up to 3.5 N of force on the head. Not only did Hoppe et al. conduct a detailed study into the rendering of g-forces on the human head during VR experiences, but the paper (much like other works in this sector) proclaimed the utility of such kinaesthetic feedback for guidance systems.

Head mounted implementations have been gaining notoriety in the literature but are yet to be seen in industry as they are still in their research and development infancy: all works agree that their implementations can render haptic stimuli for VR experiences, but are currently noticeably loud and awkward looking.

These works all show a consistent trend of building upon each other's scope of gyroscopic feedback through hardware advancements such as more motors and electrical components, but not much research or development has been done into software control and functionality. Most software frameworks for haptic rendering in the literature are minimal and their algorithmic rendering capabilities show little evidence of keeping up with the pace of hardware development. In this work, we hope to shift this paradigm of haptic rendering. We aim to present how it can be enhanced like the other VR modalities have over the last few decades by introducing a framework for a haptic rendering pipeline that renders varied haptic effects from a relatively simple hardware implementation.

3 Solution

We propose and implement *RHapTor*, a solution in the direction of tackling the disparity in engagement between audio-visual and haptic rendering by incorporating both vibrational and kinaesthetic feedback into VR using a rotational inertia-based HMD augmentation that utilises an algorithmic control of gyroscopic torques. *RHapTor* is comprised of hardware and software developments to achieve this heightened sense of immersion, whose design and implementation we shall describe in this section.

Our solution is inspired by the findings of Gugenheimer et al.'s *GyroVR*, which was the first to exploit the perception of gyroscopic torques on the head as a rendering stimulus (Gugenheimer et al. 2016). Our work then builds on Gugenheimer et al. by using recent advancements in the field of haptic rendering and an investigation into the optimal hardware and software functionalities that add the most value to the VR experiences.

3.1 Actuation Concept

Like how objects moving through space carry a certain amount of (linear) momentum with them, a disc rotating about an axis carries an angular momentum denoted by the vector $\mathbf{L}$,

$$\mathbf{L} = I\boldsymbol{\omega}, \tag{1}$$

where $\boldsymbol{\omega}$ is the disc's angular velocity vector, whose direction is given by the right hand rule. I is the moment of inertia tensor for the disc about the origin in Fig 5. Again, similar to how Newton's second law tells us how a force is required to change the linear momentum to an object, it's

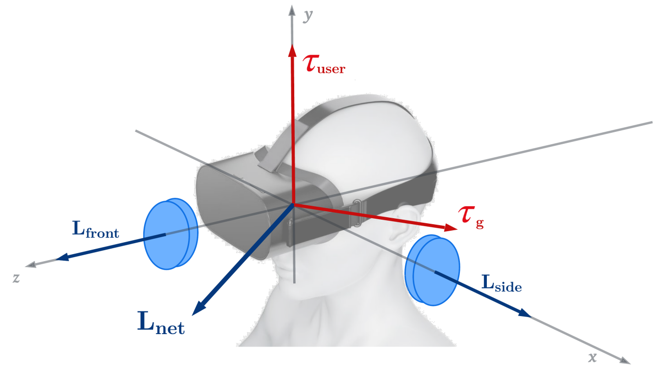

Fig. 5. Addition and control of net angular momentum, user applied torque and gyroscopic torque for haptic rendering to the user's head, as they perform a rotation about the y-axis. Right handed Cartesian coordinate system used for reference with x, y, z axes corresponding to pitch, yaw and roll respectively.

rotational analogue delineates how a torque, $\boldsymbol{\tau}$, can change the angular momentum of an object,

$$\Delta\mathbf{L} = \int_{t_1}^{t_2} \boldsymbol{\tau}(t)\mathrm{d}t. \tag{2}$$

In our case, the torque is provided by the motor and is varied algorithmically with time, t, to produce an angular momentum vector suitable for our haptic rendering requirements. This idea and analysis of a disc's angular momentum, albeit deceptively theoretical, is important to our solution since all head rotations by the user involves the user's cervical muscles generating their own torque to change this angular momentum, $\mathbf{L}$, and in doing so experience a gyroscopic torque, $\boldsymbol{\tau}_{\mathrm{g}}$. This $\boldsymbol{\tau}_{\mathrm{g}}$ is a consequence of the rotational analogue of Newton's first law: i.e. our spinning disc tends to keep spinning unless acted upon by an external torque: the user's head rotation (Gugenheimer et al. 2016). We can relate this resistance inducing $\boldsymbol{\tau}_{\mathrm{g}}$ that the user would work against, to the users own head rotation torque, $\boldsymbol{\omega}_{\text{user}}$, as follows,

$$\boldsymbol{\tau}_{\mathrm{g}} = \boldsymbol{\omega}_{\text{user}} \times \mathbf{L}_{\text{net}} = \boldsymbol{\omega}_{\text{user}} \times (I_{\text{front}}\boldsymbol{\omega}_{\text{front}} + I_{\text{side}}\boldsymbol{\omega}_{\text{side}}), \tag{3}$$

where $\times$ is the vector cross product ($\times : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}^3$) and $\boldsymbol{\omega}_{\text{user}} \parallel \boldsymbol{\tau}_{\text{user}}$ in Fig 5. This is perceived by the user's vestibular system as a resistance to the user's motion and can be used in appropriate scenarios to render the right stimulus to accompany a VR event. The human vestibular system consists of two otolith organs: the utricle and the saccule, which detect motion in the horizontal and vertical planes respectively (Khan and Chang 2013). Our actuation concept uses ideas of exploiting the vestibular system as a sensation interface (Maeda et al. 2005), and relies on the use of gyroscopic torques to impede motions such motions and hence make the otolith organs perceive a different sense of rotation than what it would expect from the torque the neck muscles applied. Like most works in this research area, we aim to subject the otolith organ to a kinaesthtic stimulus that matches the auditory and visual stimuli rendered to the eyes and ears respectively.

By using a front and a side motor with their axes perpendicular to eachother, we have an orthogonal bases

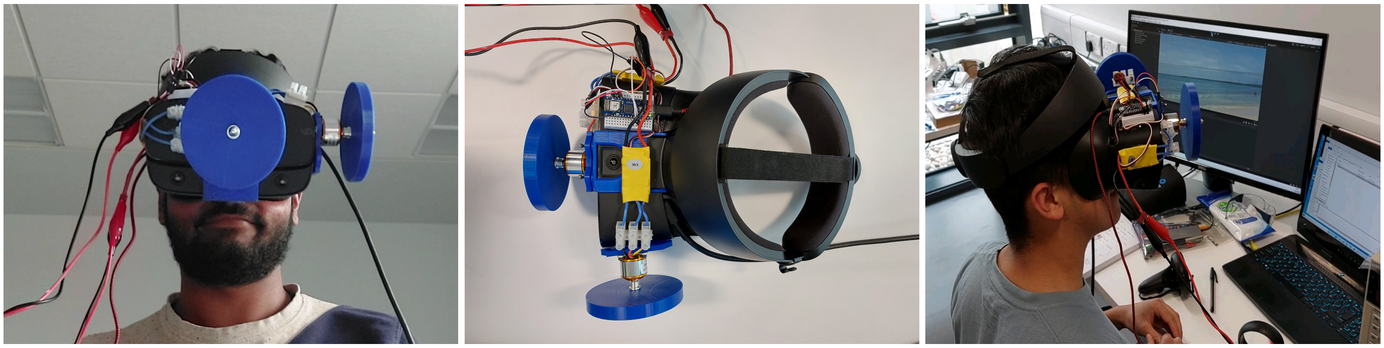

Fig. 6. Left to Right: Front view of *RHapTor* being worn, Top view of *RHapTor* showing electronics and accommodation for Oculus cameras, Final implementation of *RHapTor* in use rendering breezes on the beach environment.

of two angular momentum vectors, $\{\mathbf{L}_{\text{front}}, \mathbf{L}_{\text{side}}\}$, propagating from the users head as seen in Fig 5. They can be added using the rules of vector addition, allowing for an unprecedented software control over the 2D span of the net angular momentum carried by the headset. Hence, any rotational motion of the user's head leads to a net gyroscopic torque (depending on the number of motors engaged for the rendering demand), Furthermore, this work shows how, by algorithmically controlling the motor actuation, this torque can be made to act anywhere in the following quadrant of angular momentum space enclosed by the positive x and z axes(the motors cannot be made to spin in the reverse direction for reasons outlined in the hardware implementation sub-section 3.2, and we decided that using 4 motors, like in *Odin's Helmet* was over-engineering for the software capability and rendering this work aims to demonstrate. Once this hardware setup shown in Fig. 5 was implemented (after multiple intermediate prototypes), it provided a good foundation to test the software that utilizes this actuation concept in a set of actuation modes.

3.2 Hardware Implementation

This subsection provides a reference of the exact implementation used to achieve the results presented later in this work, and is included for ease of reproducibility. Similar to recent related works, the electronics used were all accessible off-the-shelf, with the only custom components being the 3D printed mounts designed on *OpenSCAD* (Kintel and Wolf 2014).

Our solution was built using an Arduino Nano IoT 33 microcontroller to control two brushless DC motors (BLDCs), both rated at 2200 rotations per minute (RPM) V^{-1}. This particular Arduino model was chosen because of it's size and on-board 6 degrees of freedom Inertial Measurement Unit (IMU), allowing for the possibility of onboard location-based torque calculations for certain rendering algorithms (covered in the next subsection). Additionally, the SAMD21 architecture of the processor on this Arduino is compatible with the Servo library, which we adapt in this work to communicate with the BLDC motors (the details of which are also covered in the next subsection).

These BLDCs were chosen due to their high accelerations and efficiencies compared to brushed DC motors, along with their widespread adoption in drone technologies and kinaesthetic haptic projects as recent as last year, like *Odin's Helmet*. Unlike *GyroVR*, this work avoided the use of hard drive disks for safety concerns associated with overclocking hard drive motors. As previously alluded to, it should be noted that these motors only spin in one direction. The design choice of having the two BLDC motor axes perpendicular to eachother, rather than anti-parallel was made to facilitate a much wider scope for rendering haptic feedback and algorithmic freedom that is unprecedented in the literature - even at the cost of an asymmetric mass distribution. As previously mentioned, four motors is over-engineering for the software scope and rendering capabilities we are demonstrating in this work to tackle our research question.

The BLDCs themselves are powered using one DC power supply each. Each BLDC on *RHapTor* drew no more than 3.00 A at 15.0 V. During testing of torque ranges for our algorithm development, we were able to safely operate at voltages up to 17.0 V to understand the limits of high RPM use cases. Due to the computer science nature of this work, we decided not to proceed with the electrical engineering task of developing a portable power supply, like the LiPo battery powering the hard drive motors in *GyroVR*. The primary aim of this project is to investigate how much value these haptics sensation add to VR experiences in general, not just the commercial VR experiences.

The rotations of a BLDC is controlled by an Electronic Speed Controller (ESC). The ESC used for this project is the type used for drones, chosen for their light-weight nature and ability to handle upto 30 A of current. Each ESC is connected to an Arduino for both data and power by means of a Battery Eliminator Circuit (BEC), with each ESC being connected to a a digital output pin capable of pulse width modulated (PWM) signals (see grey wires in Fig 7 (right)). The ESCs listen to the Arduino for PWM signals of specific durations, which are generated through the Arduino scripts we will describe in the software implementation (sub-section 3.3). The circuit diagram for the electronics implemented in this work is shown in Fig. 7.

These electronic components were mounted onto the Oculus Rift S headset using custom designed 3D printed mounts. The mount for the front BLDC motor was designed to 'snap' onto the headset, for modularity in mind. Whereas the inclusion of the side BLDC motor in a later prototype validated the use of hot glue for stability and avoiding the coverage of any of the 5 cameras used by the headset.

On the note of prototypes, the headset went through several experimental models before the final version that is *RHapTor* (photographed in Fig 6), to find the best hardware capabilities to realistically and safely introduce kinaesthetic feedback to the VR experience. In this work, we found and verified that a high infill density (75%, rather than the typical 15%) 3D printed disc (about 90 g and radius 5.0 cm each) worked best to achieve this effect. We tried using smaller discs to make the haptic rendering component smaller and less awkward through the use of hollow disc containers filled with metal ball bearings, but it proved to be very difficult to achieve a cylindrically symmetric mass distribution that rotated in a stable manner at high RPMs. Smaller discs made purely of high density PLA did not generate noticeable torques at safe operating RPMs. We had to make this decision on the haptic feedback rendering tradeoff between discs radius and density.

On the whole, our solution increased the mass of the Oculus Rift S headset up from 575 g to 1040 g (an increase of 81%), which is a considerable increase, but our testing found its effects were mitigated by increasing the tension in the top velcro band, transferring some of the stress onto the headset's frame. Again, caution was taken not to spend too much time on this problem as the research question is not interested in the mechanical engineering aspects of this solution. From a hardware perspective, these design choices aimed to make the actuation for the haptic rendering of $\boldsymbol{\tau}_{\mathrm{g}}$ as smooth as possible.

3.3 Software Implementation

The majority of the implementation work and results derived from this project is based on the software side. This aims to allows the rendering of haptic feedback from changes made by both the user and the VR environment, and ultimately laid the solid foundation for the development of the four modes of haptic rendering *RHapTor* demonstrates: Resistance, Dynamic Tension, Rumbles and Impulse.

Firstly, we treat *RHapTor* as a Finite State Machine (FSM) for our theoretical model for computation. This model was adhered to for all phases of design, development and testing. Due to the electrical configuration of the BLDC motors, the initial state of this device must be implemented as an initial chain of states which correspond to the calibration sequence of the motor. This calibration sequence involves sending a signal of the highest possible output angular velocity, V_{max}, followed by a delay of 3.000s, for the calibration beeps and then the lowest possibly signal, V_{min} (i.e the signal corresponding to 0 torque), followed by another delay 3.000s. These intervals were found to be the shortest period in which the device can consistently successfully calibrate. The different types of haptic feedback actuation modes are abstracted to the finite number of states this device can exist in. The state transitions are triggered by the relevant inputs and events from the virtual environment itself.

For this work, we developed three Unity 3D environments that best showcase the capabilities of this headset, as shown in Fig 8. Auditory rendering was considered and trialed for these environments, but the interference with the motors' own noise levels did not make it an enjoyable experience, and we hence decided to study the effects of just visual and haptic stimuli cooperatively. The first two environments designed were a beach, reconstructed from Tenerife, Spain, and the Vasa Museum in Stockholm, Sweden: both incorporate open source skybox imagesets, under the Creative Commons Attribution licence (Persson 2013). These were chosen for their relevant events of coastal breezes and haptic guidance to a centre of attention respectively. The third environment was a rumbling rocky planet, designed to mimic the barren terrain of an uninhabited terrestrial planet with a heightened gravitational field and falling rocks implemented as colliders capable of triggering events such as changes in haptic rendering modes. This was used to demonstrate the rumble mode of the output motors, whose algorithm we shall soon outline. In all three environments, we follow the industry and academia trend and use the OpenXR plugin on a Multi Pass render mode, along with the Unity XR Interaction Toolkit package. This allows our C# scripts and Unity project to interface with both our Oculus Rift S and other common VR headsets, and also create an XR Origin game object for our user's head and hands. The headset is set as the scene's camera rig, with a vertical offset of 2.0 m. Our custom C# scripts use

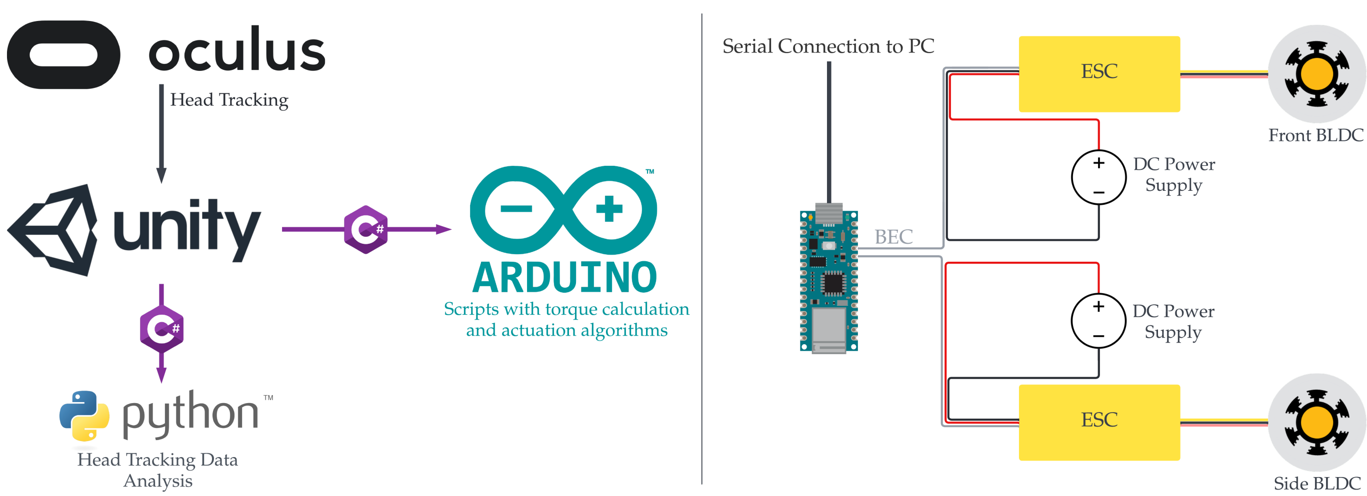

Fig. 7. Software (left) and Hardware (right) architectures for *RHapTor*.

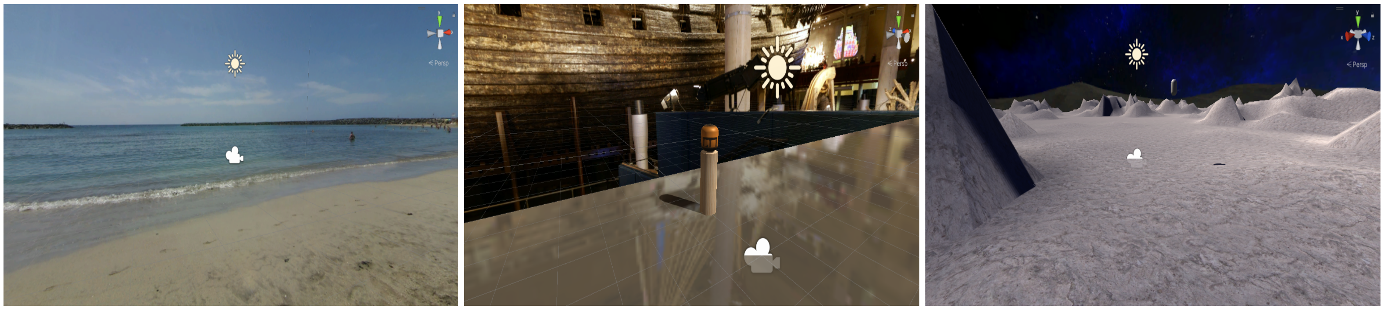

Fig. 8. Custom designed Unity 3D virtual environments to showcase our haptic rendering modes. Left to Right: Beach in Tenerife, Vasa Museum in Stokholm and a rocky, high gravity planet with falling meteors.

Unity's scripting API to access the position and orientation (as `Vector3` and `Quaternion` objects respectively, from the Unity Engine). These are utilised for head tracking, data collection and torque mapping purposes, which we shall outline soon with the four actuation modes of our solution. These measures and implementation choices were made to ensure the haptic feedback of our solution added to the HCI component of the environments - a vital consideration for our research question.

Next, the Unity-Arduino communication was achieved by modifying the C# scripts of the open source *Ardity* library, also under the Creative Commons Attributions licence (Wilches 2019). Firstly, it was realised that our FSM architecture facilitated the development of this project using a one-way communication with the Arduino (as shown in the software architecture in Fig 7 (left)), if we were to rely on head tracking data collection from the Oculus headset rather than from the Arduino's IMU. Besides being more accurate and use-case specific than the Arduino's IMU, using the position and orientation data from the headset reduced latency and traffic in the serial communication channel during our participant trials (head tracking results presented in section 4). Secondly, this work chose to adopt a character-based communication system to implement state transitions for our FSM (see Table 1 for characters $\mapsto$ actuation-modes mapping). Using the Oculus for headtracking also frees up the serial communication channel for solely character transmission - i.e. the Arduino listens to the character Unity says.

TABLE 1
Character Indices for each Rendering Mode

Character	Mode
A	Resistance
T	Dynamic Tension
R	Directional Rumble (a.k.a. anti-phase Rumble)
S	Magnitude Rumble (a.k.a. in-phase Rumble)
I	Impulse
O	Off

The next step in our project pipeline was to translate the character received by the Arduino to the motor's actuation state. We achieved the Arduino – BLDC motor communication by adapting the Arduino Servo library. This is compatible with our SAMD21 architecture, albeit typically used for sending PWM signals to servo motors. We used this functionality to abstract each ESC to a `servo` object, and created an appropriate mapping function that linearly mapped our input PWM signal encoding speed to the rotation angle signals expected for a `servo` object. We were able to send the right PWM signals to the ESC for calibrating the motors and running our character-indexed actuation modes. We chose to use PWM signals with a pulse width ranging from 1000 ms to 2000ms, as per the specifications of our chosen ESC.

With the calibration and actuation sequence in place, the next step was to develop the actuation modes for each character and implement them into the FSM's state space.

3.4 Haptic Rendering Modes

The hardware and software frameworks put in place above facilitated the development of several rendering modes, both based on previous literature and novel algorithms. They are presented in this subsection, and should be interpreted as the final stage in our haptic rendering pipeline, and occur after the calibration sequence.

3.4.1 Rendering Resistance

This was the first mode achieved by our first, single BLDC prototype, and it involved least complex in terms of state transitions. This mode reproduces many of the findings of *GyroVR* in demonstrating the use of gyroscopic torques for sensations of resistance and heaviness in VR. The mode involves setting the motor to spin at a constant angular velocity, i.e. $\boldsymbol{\omega}_{\text{user}}$ from Equ 3 does not change. Which is to say the FSM remains in the same state for the duration of the experience, as shown in Algorithm 1. While both the front and side motors were designed to be equidistant from the origin, as shown schematically in Fig 5 (mathematically the side disc's moment of inertia tensors in Equ 1 is still diagonal, just with matrix elements rearranged for disc rotation in pitch rather than roll as the principle axis of the disc). This mode generally only relies on the front motor, as there are many more use cases and scenarios where a gyroscopic torque along the pitch axis would be useful: as opposed to a $\boldsymbol{\tau}_{\text{g}}$ with a roll component, e.g. guiding users and wind resistance. This makes the rendering mode particularly useful in the beach and museum environments used in this work, for rendering resistances of a sea breeze and museum artefact guidance system, as we shall demonstrate in the next section.

While this state did offer the potential to vary rotational torques by modulating speed, it was found that very high output speeds magnified unwanted vibrations of the headset, breaking the immersive experience. Small modulations in speed only led to a change in torques of the order 10^{-3} Nm, which is an order of magnitude smaller than those detectable by humans (Sakai, Fukui, and Nakamura 2003).

Algorithm 1: Resistance Rendering

Data: $V_{\text{front}} > 0$
Data: $\text{message} \in \{'\text{A}', '\text{R}', '\text{S}', '\text{T}', '\text{I}', '\text{O}'\}$
while $\text{message} = '\text{A}'$ **do**
    $\text{speed}_{\text{front}} \leftarrow V_{\text{front}}$
    $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
end

3.4.2 Rendering Dynamic Tension

This rendering mode applies a map to the acceleration of the headset to output a potential value for the speed the BLDC should spin at (speed variable in Algorithm 2). That is to say the output torque increases as the user displaces their head from the z-axis at a rate greater than our fixed threshold, ω_{thresh}. Acceleration was chosen in favour of angular velocity as this makes the rendering useful in use cases where the user may try to walk around the environment. This use case was motivated for haptic guidance systems, and worked best with the museum environment. Mathematically, the FSM 'sways' through the state space of speed values as the user moves their head, with a mapping to the head's position. This use case was motivated for haptic guidance systems, such as the museum environment. Similar to the resistance rendering mode, dynamic tension was found to work best running on just the front motor, inducing gyroscopic torques on the pitch and yaw axes.

Unlike the data collection for the head tracking component of this project, this angular displacement measurement relies on the microcontroller calculating the appropriate mapped speed using its onboard Arduino IMU data. This is to keep the serial communication channel with Unity clear for character transmission, leaving flexibility for future possibilities of mode switching for more complex haptic rendering methodologies.

Algorithm 2: Dynamic Tension Rendering

Data: $V_{\text{front}} > 0$
Data: $V_{\text{high}} > V_{\text{front}}$
Data: $\text{message} \in \{'\text{A}', '\text{R}', '\text{S}', '\text{T}', '\text{I}', '\text{O}'\}$
Data: $\omega_{\text{thresh}} > 0$
Object: IMU
$(\omega_x, \omega_y, \omega_z) \leftarrow \text{IMU.readAcceleration}()$
$\text{speed}_{\text{front}} \leftarrow V_{\text{front}}$
while $(\text{message} = '\text{T}') \cap (\omega_x \geq \omega_{\text{thresh}} \cup \omega_y \geq \omega_{\text{thresh}} \cup \omega_z \geq \omega_{\text{thresh}})$ **do**
    $\text{speed}_{\text{front}} \leftarrow V_{\text{high}}$
    $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
end

3.4.3 Rendering Rumble Effects

This rendering mode relies on the actuation of both BLDC motors, and encapsulates the haptic rendering complexity of *RHapTor*'s dual motor system through software manipulation. A rumble effect naturally carries elements of both vibrational (shaking) and kinaesthetic (heaviness) feedback, making this mode useful for the rocky planet environment. This mode was taken a step further for natural integration by wrapping game objects like falling rocks in a collider, and using Unity's event detection to relay the rendering of this effect to the micro-controller.

We were able to devise two rumbling effects that our architecture lends itself to naturally. The first is a 'directional' or 'anti-phase' rumble, created when the angular momentum vectors of each disc oscillate between a low and a high output value with a phase difference of $180°$ or π rad - that is to say when the front motor spins at its maximum, the side motor spins at its minimum and vice versa. This makes the direction of $\mathbf{L}_{\text{net}}$ and consequently $\boldsymbol{\tau}_{\text{g}}$ oscillate its rotation orientation (by Equ 3), with the $\mathbf{L}_{\text{net}}$ vector repeatedly sweeping a sector of the quadrant enclosed by the positive x and z axes in Fig 5. The second effect is a 'magnitude' or 'in-phase' rumble, which works similar to the last rumble, but has both discs raise and lower their angular momenta in sync with 0 phase difference, thereby varying the magnitude of $\mathbf{L}_{\text{net}}$ and again consequently $\boldsymbol{\tau}_{\text{g}}$. This would be perceived by the user's vestibular system as an apparent variation in the effective rotational inertia of their head through the experience. Both rumble modes (described in the pseudocode of Algorithm 3) demonstrated how the same hardware implementation can be used to render both kinaesthetic and vibrational feedback.

3.5 Rendering Impulses

Historically, haptic impulses have been an overwhelmingly vibro-tactile concept in commercial devices. However, short-lived high intensity interactions such as collisions, jolts, whiplash and punches are often accompanied by quick forces and torques that do not last long. This effect was rendered by having a steady or gradual monotonically increasing torque come to an abdrupt halt.

In theory, this was supposed render a short-lasting jolt of torque to the head, but the lack of instantaneous deceleration capabilities of the ESC for our BLDCs could not facilitate this without external mechanical braking of a servo motor, as shown in Algorithm 4.

4 Results

Having discussed the details of implementation for our solution, this section reviews the various tests and results obtained from our work to gauge the performance and feasibility of our haptic engine, and gives insights to rotational inertia-based haptics as a whole. Our findings were dichotomised into technical results and measurements from the hardware/software framework, and user-centric findings and outcomes. The VR and HCI nature of our project makes these two perspectives invaluable to answering our research question.

Algorithm 3: Rumble Rendering

Data: $V_{\text{front}}, V_{\text{side}} > 0$
Data: $V_{\text{high}} > \max(V_{\text{front}}, V_{\text{side}})$
Data: $V_{\text{low}} < \min(V_{\text{front}}, V_{\text{side}})$
Data: message $\in \{'A', 'R', 'S', 'T', 'I', 'O'\}$
while (message $=$ 'R') **do**
  $\text{speed}_{\text{front}} \leftarrow V_{\text{high}}$
  $\text{speed}_{\text{side}} \leftarrow V_{\text{low}}$
  $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
  $\text{setSideSpeed}(\text{speed}_{\text{side}})$
  delay(50) ; /* Delay 50 ms */
  $\text{speed}_{\text{front}} \leftarrow V_{\text{low}}$
  $\text{speed}_{\text{side}} \leftarrow V_{\text{high}}$
  $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
  $\text{setSideSpeed}(\text{speed}_{\text{side}})$
  delay(50)
end
while (message $=$ 'S') **do**
  $\text{speed}_{\text{front}} \leftarrow V_{\text{high}}$
  $\text{speed}_{\text{side}} \leftarrow V_{\text{high}}$
  $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
  $\text{setSideSpeed}(\text{speed}_{\text{side}})$
  delay(50)
  $\text{speed}_{\text{front}} \leftarrow V_{\text{low}}$
  $\text{speed}_{\text{side}} \leftarrow V_{\text{low}}$
  $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
  $\text{setSideSpeed}(\text{speed}_{\text{side}})$
  delay(50)
end

Algorithm 4: Resistance Impulses

Data: $V_{\text{front}} > 0$
Data: message $\in \{'A', 'R', 'S', 'T', 'I', 'O'\}$
Object: ServoMotor
while message $=$ 'I' **do**
  $\text{speed}_{\text{front}} \leftarrow 0$
  $\text{setFrontSpeed}(\text{speed}_{\text{front}})$
  ServoMotor.rotate(180°)
  delay(100)
  ServoMotor.rotate(0°) ; /* Brake Off */
end

4.1 Technical Study

We used a newtonmeter to measure the mass of the Oculus Rift with *RHapTor* to be 1040g (which is 81% higher than the weight of an unmodified Oculus Rift S). The newtonmeter also revealed the maximum perceived force acting on the users head from a single motor to be 1.1 N (i.e torques of $\approx$ 0.16 Nm), using a technique suspending the headset to hang freely and zeroing the scales, similar to the methodology of *Odin's Helmet*. Vector addition of torques revealed the maximum possible torque of the two motor system to be 0.23 Nm.

While the actual latency between Unity and the motors was never observed to exceed 0.1 s, ESC cogging effects with the BLDC connections often delayed the output torques by up to 5.0 s. This was often presented as a warmup phase for environments like the beach and museum, but for instantaneous environments where instantaneous impulsive torques were needed, such as the rocky planet, this posed an issue and required an additional warmup state in the calibration sequence.

A technical analysis of *RHapTor* with both BLDCs in use revealed the following noise level variation over the initial calibration beeps, resistance rendering and subsequent deceleration phase in Fig 9. The peak sound intensity at the position of the user's ears never exceeded 73.1 dB, which is widely considered to be a moderate noise level and is in the noise level band of lowest relative risk of hearing loss or damage in the 15 to 80 year olds age group (Nelson et al. 2005). However, prolonged exposure becomes disturbing and irritating to some users. Fig 9 depicts the sound intensity variation of both the calibration sequence, maximum runtime torque and deceleration when switched off. Additionally, the aforementioned ESC cogging effect was not found to occur after the BLDC motors had begun spinning.

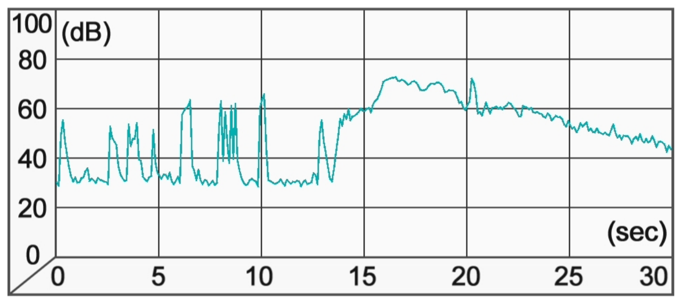

Fig. 9. Sound intensity variation during both calibration beeps (for the first 14.0s) and runtime (14.0s to approximately 19.0s) and deceleration (19.0s onwards) of *RHapTor*, measured from the average position of an the user's ear.

Retreiving tracking data from Oculus via a C# script proved efficient and did not show any noticeable impact on mode changeover latency. As previously mentioned, we were able to achieve higher and better torques all the way up to 3.00 A at 15.0 V, with the BLDC motors refusing to calibrate beyond 17.0 V.

The haptic feedback was ended by terminating the Unity environment (thereby breaking the character-based communication and sending our FSM model to its Off state), or by manual intervention by resetting the Arduino or turning off the power supply. All three methods showed no significant latency differences, besides the cogging and deceleration time stated above.

When turning off the haptic rendering, we found an average natural deceleration time of 17.2 ± 0.2s with our ESC – this can be improved by using better ESCs with more precise control and electromagnetic braking mechanisms, as previously mentioned to render impulsive torques. We accomodate additional ESC upgrades by modulating the PWM modulation range used by our servo library.

4.2 User Study

Our user study assessed the feasibility of the device to generate haptic feedback by looking at the resistance rendering mode. There are two main reasons for this choice. Firstly, this mode is the foundation upon which all other actuation modes are derived from, and the rendering capability of this mode uses hardware and software functionality that form the heart of all other modes. Secondly, the main output from this mode is the kinaesthetic feedback from gyroscopic torques, a phenomenon that is much less studied in the HMD literature than the vibrational feedback, and whose understanding better prepares us to tackle our research question.

Our user study involved 4 participants with an average age of $\mu = 20.5$ years and standard deviation $\sigma = 0.5$ years. The experimental setup consisted of the user wearing our headset while sitting on a chair to minimise any translational noise in their head tracking data. The same Oculus play area dimensions and shape were used for all participants. The users were allowed to first freely explore, and then perform a sequence of head rotation about the yaw axis in two environments: the Tenerife Beach and the Vasa Museum, as they best captured use cases for the resistance mode (the aforementioned coastal breezes and haptic guidance respectively). The users performed this twice: first with the haptic rendering off, for a control dataset, and then with it switched on.

A relevant metric to gauge the influence of our haptic rendering functionality is the movement and orientation of the head when subjected to the above environments, compared to a haptic control. Fig 10 shows the distribution of head positions in the x-z plane from all participants' free exploration of the beach environment with no further instructions, both with and without haptic rendering. All data was collected from the same play area, with offsets due to chair position removed. In all trials, the y-displacement of the head had a range no large than 0.01 m, and so was deemed negligible (the resolution of `Vector3`s from the Oculus was 0.01m). From Fig 10, it is apparent that users' exploration was much more limited when the haptic rendering was turned on. This lower spread of displacements along the z-axis of the blue regions compared to the red regions can be attributed to the resistance inhibiting users from exploring their surroundings. This is consistent with the experimental setup, since majority of the z-exploration happens when the participant turns their head to explore the front and behind of them. The figure corroborates how the resistive torques from this rendering mode make it increasingly unnatural to turn during exploration, and affect environment perception.

The correlation of density peaks of the red and blue regions is a result of common areas of exploration and attention on the beach landscape found across participants. An independent two sample t-test performed for both axes compared the haptic and non-haptic datasets, and revealed a noticeable shift in the exploration distributions along x, with $t = -2.845$. Furthermore, while most of the read and blue peaks coincide in the univariate distributions, any multi-modality that does not cross-correlate likely due to differences in height of the participants from the datasets.

Another useful analysis that measures the utility of torque-based haptics in VR perception and HCI is the angular tracking of head orientation in space. This is a vital component not just for gauging the impact of *RHapTor* on VR experiences as outlined above, but also in applications that consistently resurface in the literature, such as haptic guidance systems. Fig 11 (left) delineates the orientation evolution of a participant in the Vasa Museum environment by tracing the axis-angle representation of their headset's orientation on the surface of a unit sphere as a user varies their yaw angle (i.e. azimuthal angle) by $0 \rightarrow -\frac{\pi}{2} \rightarrow \frac{\pi}{2}$. One of the most stark findings from this study that was consistent over all participants and all runs is the amount of mechanical vibration introduced to the headset by the high torque BLDC motor output - which has adverse consequences for haptic rendering pipelines that rely on orientation data, like the dynamic tension rendering mode. Users later reported the vibrations did carry through in their ocular perception of the VR engine, with a 'tremor', 'earthquake' and even 'tsunami-like' description of the environment with high torque haptic rendering. However, they did report the vibration was not as intense as the vibration on the Unity Screen on the PC they saw from other participant, likely due to the vestibulo-ocular reflex and driven oscillations of the eyes and face along with the headset.

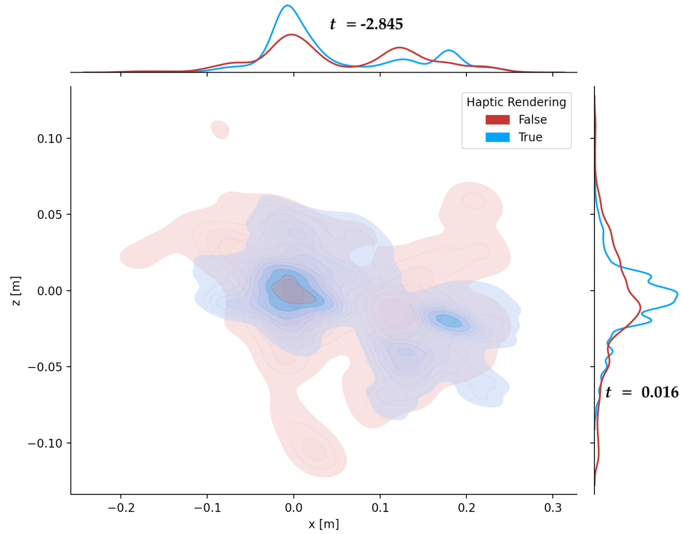

Fig. 10. Bivariate contour plot of head tracking displacement in the x-z plane. Normalized univariate plots along each axis are also included, with t-values describing the difference between motion distributions caused by haptic rendering.

Although the same rendering setting was used for both environments, 100% of participants agreed that the vibrational effect at the Museum was more 'intense' than that in the Beach. This stems from how the Museum is filled with a lot more detail and objects, than the Beach which is majorly just plain see and skies, with not much vertical detail. This indicates the need for environment based haptic rendering algorithms that mitigate such noise and account for it's effects depending on whats being rendered.

From observing the transit path of the participant in Fig 11 (left), it is clear that users spend less time in transit when haptic rendering is switched on, corroborating Equ 3 which clearly illustrates the dependence of gyroscopic torque on the angular velocity of the user's head in transit. This can be noticed in the details of how the upper red transit path lies just below its haptic equivalent trajectory, while the lower red is significantly higher than its haptic

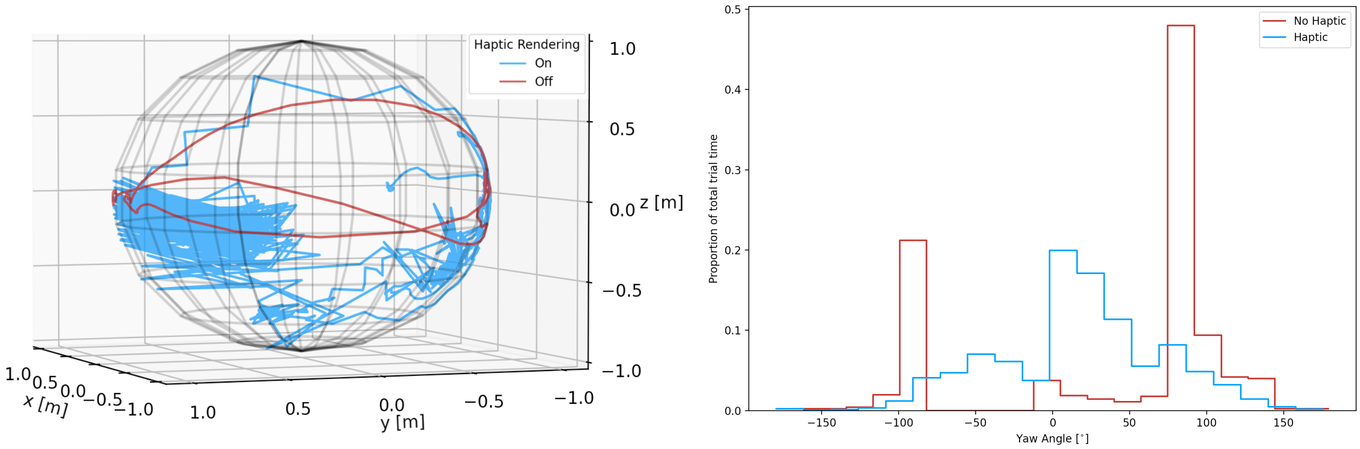

Fig. 11. Orientation of headset during Vasa Museum guided motion sequence, traced as a transit path on a unit sphere (left) and distribution of yaw (a.k.a azimuthal) angle variation during this same exercise, both with haptic rendering on and off (right).

TABLE 2
Confusion Matrix for Output Torque Direction. All values are percentages of the total participant pool

	User Responses	
Correct $\boldsymbol{\tau}_g$ axis	Pitch	Yaw
Pitch	100%	25%
Yaw	0%	75%

trajectory: indicating how the gyroscopic torques skew the orientation trajectories of these transit paths, making them closer to the poles. Therefore this quantifiable difference and shift in orientation complements the translational skew to head motion caused by haptic rendering and patterns in Fig 10. This signifies how gyroscopic torques influence angular mechanics in the way predicted by the theory corroborating our form of haptic rendering and verifying concepts from *GyroVR* and *Kabuto*.

We illustrate this statistically by looking at all participants' azimuthal angular displacement (yaw angle) with and without haptic rendering with the histograms in Fig 11 (right). We see the proportional time spent (to make comparisons between participants fair) between the two immersions in the same museum environment, and note how the distribution is much more central, uni-modal and focuses attention on the artefacts in front of the user when haptic rendering is turned on. This is quantifiably evidence by how the haptic distribution is much more Gaussian , with a Shapiro-Wilk statistic of $W = 0.9907, p = 8.4 \times 10^{-22}$, than the non-haptic multimodal distribution, with $W = 0.7543, p = 1.3 \times 10^{-34}$. $W = 1.0$ for a pure Gaussian distribution.

The correct and incorrect responses rate is presented in the confusion matrix in Table 2, which summarises the perceived correctness of the Algorithm 1. Incorrect responses were only given once, however all participants noted that mechanical vibrations of the mounting unit did make it difficult to discern at lower angular velocities. Interestingly, participants did remark that they were able to noticeably detect the gyroscopic torque, when they swung and circled their head in a manner that their noses traced out circles. This can be attributed to the rapid azimuthal direction switching and the accompanying oscillation in $\boldsymbol{\tau}_g$ direction that the vestibular system detects.

Besides these quantitative results, there were certain observations and behaviours that were consistent across the sample. Similar to *Kabuto* (Tanichi et al. 2020), this work took notes on the users during their participation and from their remarks and feedback. All participants noted that the headset was noticeably heavier than an unmodified Oculus Rift S, which is to be expected given the 81% increase in mass required to generate haptic responses at a safe torque. This was alleviated in some cases by tightening the headstrap to transfer tension away from the neck and onto the rigid Oculus frame. Some users also noted that the haptic response was rendered more to the cheekbones than the rest of the head - such mounting shortcomings were also identified as sources of discomfort in related works such as *GyroVR*. While the kinaesthetic feedback was louder than the typical haptic responses our participants were used to, this was not a major concern in the participant's feedback.

5 Evaluation

The above demonstration and insights gathered from the advantages and disadvantages of *RHapTor*, along with its influence on participant interaction with VR environments placed us in an excellent position to reflect on and evaluate the progress made by this work: both on the technicalities of torque based feedback and in answering the research question.

We are confident in asserting that this project was a success in terms of meeting our objectives and investigating feasible inertial and torque based haptics. To this extent, we gauge the capabilities of our project and how well it works using Table 3 which summarises our technical findings in relation to similar gyroscopic haptic technology. Since this is such a recent research area, the metrics for evaluating new developments are much less standardised and agreed upon than other modalities. Table 3 collates these to the best of the literature's abilities.

5.1 Strengths

The results on user behaviour with haptic rendering switched on in the previous section clearly show how the rendering paradigm used in this work creates effects that are not just perceivable to the user, but also influence the user experience of the VR world.

We have demonstrated that gyroscopic torques play an excellent role in engaging and immersing the user in otherwise uneventful environments, when chosen correctly, with no less than 75% of our participants correctly interpretting the haptic stimuli.

In terms of hardware, *RHapTor* boasts a motor calibration sequence that works consistently and renders resistive torques at a scale that tuned to be safe but also noticeable to humans. The motor coordination is on par with the state of the art and provides ample scope for software development. Although Table 3 shows *RHapTor* has a 20-25% lower force output than other gyroscopic rendering devices, it operates at a much lower power consumption, and is the quietest in the literature. Additionally, since torques are our principle form of actuation, a relatively smaller force close to the head is all that is needed to generate a noticeable torque - which is not the case for force feedback rendering devices. This approach of mechanical measurements was well suited to this project, as it gave experimental grounding to compare our work with other devices in the field.

On the note of software, the rendering algorithms presented in this work made improvements on the state of the art and turn what has typically been considered a feat of VR engineering into the skeleton of a software-based haptic rendering framework. These algorithms also provide scope for customisation and tuning of parameters like ω_{thresh} for dynamic tension rendering that adapts to the environment instance. Additionally, the diverse range of Unity environments is a testimony to the versatility and much broader extent of gyroscopic rendering by software manipulation than previously estimated. Another strength of this work is the finite state machine interpretation of a haptic rendering engine. This paved a more natural way forward into a haptic rendering pipeline combined with a chararcter-based communication facilitating rapid interchange between rendering algorithms at the standard Arduino baud rate of 9600 Hz, all within the same environment. Altogether, our work is a proof of concept for the software dexterity over gyroscopic torques that has not been demonstrated in the literature yet.

In performing this work, our methodology proved to be very appropriate for our research question. By highlighting how versatile haptic effects do not require the extensive engineering and hardware intricacies as seen in works like *Odin's Helmet*, and can be achieved algorithmically. These simplifications allowed for comparable effects to the state of the art without the need for high power consumption or headset weights as shown in Table 3. This adds further nuances to both our research question and field of haptic rendering as a whole, almost shifting the paradigm of how haptics should be approached: again moving away from the orthodox vibro-tactile mindset, where hardware does all the heavy lifting and software merely triggers actuations, to a more balanced and algorithmic rendering pipeline.

5.2 Limitations

While our work makes significant progress and investigation into the software control for haptic rendering with gyroscopic torques, there is still a range of limitations. These stem from both the implementation and implications of the results gathered.

The main hindrance lies in the electrical component of this work: at times the BLDC motors suffer from ESC cogging and fails to spin smoothly after the calibration sequence. This is almost always a result of unstable electrical contact, and while various methods of soldering and terminal strips were trialed, the quality of the wiring persisted. This inhibited immersion and often needed a preliminary warm up phase or rewiring session before successful operation. Weight is another drawback. With the headset increasing the mass of the oculus by over 81%, smaller inertial outputs are difficult to differentiate from the mass of the headset itself: particularly since psychophysical effects such as Stevens' power law for torque based stimuli to the head remain relatively unexplored (Stevens 1957).

Furthermore, the high RPM of BLDC motors mounted onto an Oculus headset in this work introduced a noticeable level of vibrations (compared to the over-clocked harddrives used in *GyroVR*), as evident in Fig 11 (left) and participant feedback. This lowered the quality of immersion in environments that heavily relied on gyroscopic torque variations like the Museum guidance system using dynamic tension rendering, and delivered the vibro-tactile equivalent of the rumble algorithm on high RPMs. This vibration is what beckoned the need for ω_{thresh} in algorithm 2 rather than the development of a linear map from orientation space to state space as in *GyroVR*. As with all works in this area, acoustic noise is a concern, but the noise levels are not as bad as the counterparts in Table 3, since we shifted the focus to software-guided rendering.

From a software perspective, a bottleneck to further functionality and development is the single channel serial communication with the Arduino. A wider scope of inter-

TABLE 3
Comparison with recent gyroscopic based haptic feedback devices for VR

Device	Max Force [N]	Mass of Device Only [g]	Power Consumption [W]	Sensing Region	Max Sound [dB]
GyroVR	n/a	390	n/a	Head	n/a
Thor's Hammer	4.0	692	204.7	Hand	80.7
Kabuto	n/a	720	n/a	Head	n/a
Odin's Helmet	5.0	n/a	n/a	Head	108.7
RHapTor	1.1	465	**90.0**	Head	**73.1**

activity and output complexity could be derived from overcoming this limitation. Additionally, one of the more challenging areas in the haptic rendering problem is choosing the right actuation for a VR experience. During participant trials, the beach environment had a noticeable resistance, but not all participants were convinced it was 'breeze–like' due to the rotational nature of the gyroscopic torque, rather than the translational force and cooling sensation of a gust. To this extent, the answer to our research question would become more intricate, with the extent of feasible adoption of haptic feedback dependant on other tactile feedback such as temperature. This also made participants relate more to the vibrational feedback of *RHapTor*, isolating it as an earthquake-like effect. This limitation from our results disagree with Gugenheimer et al.'s findings that non-realistic forces for an event can be perceived as realistic for the user, but this is likely down to subjective participant opinions from both our small datasets (Gugenheimer et al. 2016). It is also likely that participants explore less during the haptic run due the bias that this was their second time in the same environment, and the environment was not novel to their curiosity.

5.3 Improvements

Having identified the strengths and limitations in our implementation and rendering paradigms, we propose a set of methodology improvements to tackle these errors and make our haptic rendering approach more suitable for future use cases, given the opportunity to resume or even restart the project.

Currently, we rely on wired power and data since the main scope of the project was to develop software for haptic rendering that magnifies and portrays it's extent - rather than a commercial product. Such engineering improvement could include a LiPo power supply and a Bluetooth communication to facilitate a wireless *RHapTor* hardware interface.

If given the opportunity to further this project, we would direct more attention to the development of a full-fledged haptic engine API on top of Unity to allow interfacing with a haptic rendering pipeline, analogous to how visual, auditory rendering is carried out on the Oculus. This would require insights into how auditory, visual and haptic rendering share and differ in, and would also create a generalisable framework for investigating senses in the future such as gustatory and olfactory stimuli which, given the trajectory of VR and Burdea's 3 I's, is likely inevitable for the evolution of VR and metaverses.

Additionally, the algorithms presented in this work had been designed in a future-proof manner such that they can be generalised to more complex effects in 3, 4 or n BLDC systems - or any other rotational actuators for that matter - in arrays or matrices. This would allow for gyroscopic torques, rotational impulses and dynamic tensions of a variety of magnitudes to be induced anywhere in 3D space.

In terms of methodology improvements, it is clear that the ESC cogging effects must be resolved before any more haptic complexity beyond the algorithms presented in this work can be rendered. A suitable approach would be to go for higher grade motors which, besides better durability and connections, also come with wider ESC functionality. This would enable us to incorporate electromagnetic braking mechanisms using the appropriate PWM signals from our current servo implementation, such as those used in drones. Additionally, we would tackle the vibration problem by incorporating noise cancellation technologies, firmer motors and damping systems. This would be necessary for any orientation based haptic algorithms like dynamic tension, and any further haptic complexity beyond that.

Similarly there are aspects of the project life cycle that could be improved for more efficient software engineering. The development of game-like or objective-oriented environments for the participants would have allowed for further insights on the HCI elements of haptic torques with quantifiable in-game scores. This also opens the possibility of haptic feedback for events as a consequence of human interaction and decision-making.

Naturally, as the project progresses safety features are to be added as soon as they are required - particularly with disc based technologies at higher RPMs. Rounder edged discs and hair safety mechanisms would be needed. As it stands *RHapTor* is relatively unbalanced, and a diagonal disc placement similar to *Kabuto* is an enviable improvement that maintains our software control and safe generation of $\mathbf{L}_{\text{net}}$ (see Fig 4). Lastly, while our noise levels are the lowest measured in the literature, it is still advisable to dedicate time and resources to hearing protection mechanisms and ensuring haptic usage durations do not exceed 30-45 minutes.

5.4 Answering our Research Question

While majority of the evaluation presented above is, directly or indirectly, aimed at looking into the extent of feasible haptic rendering, it is worthwhile to formally evaluate our project against our aims and objectives.

In terms of our progress towards implementing and studying this inertial and torque-based haptic engines, we were successfully able to not just build a single and double axis version of the actuation concept from the literature, but we also designed a rudimentary rendering pipeline that interacts with events in Unity and conveys their relevant haptics, thereby reflecting the intricacies and software demands of furthering existing works. We built on the hardware foundations of *GyroVR* and *Odin's Helmet* in particular to evaluate the feasibility of haptics as well as illustrating the environments that work well and those that are disjointed from our rendering algorithms. It is clear from our study that to tackle this facet of out question, we must appreciate how concepts such as vection generalise and reappear in a rotational sense. The inherent dizziness of VR would be exacerbated if the haptic engagements are chosen poorly, making it worse and more uncomfortable than if there were no haptics at all.

Significant progress was also made, from both our technical and user studies, to better understand the versatility of torque based feedbacks. Gyroscopic torques have shown a strong potential in haptic rendering and immersion and can induce torques present in a variety of day to day interaction. These will at some point be necessary for advancements in haptic VR hardware and notions of a metaverse, as motivated in sub-section 1.1.

Our work also delivers more quantitative insights into the role of head based gyroscopic haptics in VR. We found strong evidence for uses in guidance systems and scope for further improvements of the same hardware apparatus that would allow for its incorporation in gaming (such as impulse rendering). The results show how, with adequate software development, simple ideas of gyroscopic torques can have complex effects on user experiences and engagement in VR, with users spend remarkably different amounts of time paying attention to different areas of the same environment with haptic feedback turned on and with it turned off (Fig 11 (right)). The hardware reliability shortcomings did, however, have an impact on the progress made towards answering our research question, with issues of ESC cogging and lack of ESC-based braking electromagnetic mechanisms for Impulse rendering.

6 Conclusion

In this work, we develop and implement a hardware and software architecture capable of rendering and delivering haptic feedback to the head using gyroscopic torques induced by head rotations of orthogonal rotating discs on an HMD. We then outline a haptic rendering pipeline capable of taking head tracking and positioning data and algorithmically outputting appropriate haptic feedback with changing capabilities. These capabilities are formalised and tested by our the 4 proposed rendering algorithms that leverage the functionality of a dual disc system to render kinaesthetic and vibrational haptics to the user. We then demonstrate how this has a definitive qualitative and quantitative effect on users' head movements, explorations and perceptions of their environment, with the correct experience being actuated atleast 75% of, albeit limited, the observed trials.

With regards to fulfilling the initial aims and objectives of the project, our work can be regarded as an overall success. Our technical and user evaluations revealed *RHapTor* not only successfully and safely reproduces torque effects from the literature, but also builds on torque manipulation through software. The project sheds light on the uncharted waters of the software scope of haptic rendering.

Overall, we can confirm the project has successfully explored and investigated the nuances and finer details of haptic rendering and confirmed that through algorithmic control of torque generation, even the simplest of torque setups can feasibly generate engaging haptic effects and guidance systems. Our proposed algorithms and pipeline for this novel research area shed light on human interpretation of gyroscopic stimuli. It has also revealed how areas of choosing and designing the right rendering algorithm for an experience are not yet completely understood. Therefore by answering our question, we have opened up a plethora of further questions regarding haptic rendering - particularly in terms of whether or not it should be approached in a similar fashion to auditory and visual rendering, as well as the computational costs (in both software and hardware resources) and flexibility associated with such developments.

Future works can aim to build on this pipeline and undertake a deeper engineering investigation to reduce the effects of ESC cogging affecting reliability. This would pave the way much more compact, less awkward arrays of rotating discs to generate kinaesthetic effects and inertia. Furthermore, an improved understanding of the human vestibular system is needed to better design and standardise specifications for haptic rendering engines. Questions like the minimum activation torque threshold and the torque resolution sensitivity of lesser studied areas of the body like the head and neck's cervical muscles are still to be formally investigated in the area of bio-mechanics and HCI.

Ultimately, the findings in this work bolster the growing amount of evidence in favour of haptic engagement through creative physics and computing, bringing us a step closer to Sutherland's ideal VR systems that engages with all the senses. Indistinguishable from reality.

References

Aas, Benjamin Gregor (2012). "What's Real?: Presence, Personality and Identity in the Real and Online Virtual World". In: *Virtual worlds and metaverse platforms: New communication and identity paradigms*. IGI Global, pp. 88–99.

Benko, Hrvoje et al. (2016). "Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 717–728.

Borowski-Beszta, Mikołaj and Michał Polasik (2020). "Wearable devices: new quality in sports and finance". In: *Journal of Physical Education and Sport* 20, pp. 1077–1084.

Bouzit, Mourad et al. (2002). "The Rutgers Master II-new design force-feedback glove". In: *IEEE/ASME Transactions on mechatronics* 7.2, pp. 256–263.

Brooks, Frederick P (1999). "What's real about virtual reality?" In: *IEEE Computer graphics and applications* 19.6, pp. 16–27.

Brown, L M, S A Brewster, and H C Purchase (2005). "A first investigation into the effectiveness of tactons". In: *First joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems. world haptics conference*. IEEE, pp. 167–176.

Burdea, G (1993). "Virtual reality sistems and applications En: Electro93 International Conference". In: *Short Course, Edison: Nueva York* 28.

Burdeau, Grigore C. and Philippe Coiffet (2003). *Virtual Reality Technology*. 2nd. Wiley.

Delazio, Alexandra et al. (2018). "Force jacket: Pneumatically-actuated jacket for embodied haptic experiences". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Desai, Parth Rajesh et al. (2014). "A review paper on oculus rift-a virtual reality headset". In: *arXiv preprint arXiv:1408.1173*.

Düking, Peter et al. (2016). "Comparison of non-invasive individual monitoring of the training and health of athletes with commercially available wearable technologies". In: *Frontiers in physiology* 7, p. 71.

Gibson, James J (1962). "Observations on active touch." In: *Psychological review* 69.6, p. 477.

Gugenheimer, Jan et al. (2016). "Gyrovr: Simulating inertia in virtual reality using head worn flywheels". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 227–232.

Haghi, Mostafa, Kerstin Thurow, and Regina Stoll (2017). "Wearable devices in medical internet of things: scientific research and commercially available devices". In: *Healthcare informatics research* 23.1, pp. 4–15.

Henriksen, André et al. (2018). "Using fitness trackers and smartwatches to measure physical activity in research: analysis of consumer wrist-worn wearables". In: *Journal of medical Internet research* 20.3, e9157.

Heo, Seongkook et al. (2018). "Thor's hammer: An ungrounded force feedback device utilizing propeller-induced propulsive force". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–11.

Hoppe, Matthias et al. (2021). "Odin's Helmet: A Head-Worn Haptic Feedback Device to Simulate G-Forces on the Human Body in Virtual Reality". In: *Proceedings of the ACM on Human-Computer Interaction* 5.EICS, pp. 1–15.

Jerald, Jason (2016). "The VR book". In: *Human-Centered Design for Virtual Reality. ACM und Morgan & Claypool*.

Juo, Yen-Yi et al. (2020). "17 - Center for Advanced Surgical and Interventional Technology Multimodal Haptic Feedback for Robotic Surgery". In: *Handbook of Robotic and Image-Guided Surgery*. Ed. by Mohammad H. Abedin-Nasab. Elsevier, pp. 285–301. ISBN: 978-0-12-814245-5. DOI: https://doi.org/10.1016/B978-0-12-814245-5.00017-7.

Kaewkannate, Kanitthika and Soochan Kim (2016). "A comparison of wearable fitness devices". In: *BMC public health* 16.1, pp. 1–16.

Khan, Sarah and Richard Chang (2013). "Anatomy of the vestibular system: a review". In: *NeuroRehabilitation* 32.3, pp. 437–443.

Kim, Gerard (2005). *Designing virtual reality systems*. Springer.

Kintel, Marius and Clifford Wolf (2014). "OpenSCAD". In: *GNU General Public License, p GNU General Public License*.

Lawson, Ben D and Bernhard E Riecke (2014). *Perception of Body Motion*.

Lederman, Susan J (1991). "Skin and touch". In: *Encyclopedia of human biology* 7, pp. 51–63.

Lederman, Susan J and Roberta L Klatzky (2009). "Haptic perception: A tutorial". In: *Attention, Perception, & Psychophysics* 71.7, pp. 1439–1459.

Maeda, Taro et al. (2005). "Shaking the world: galvanic vestibular stimulation as a novel sensation interface". In: *ACM SIGGRAPH 2005 Emerging technologies*, 17–es.

Massie, Thomas H, J Kenneth Salisbury, et al. (1994). "The phantom haptic interface: A device for probing virtual objects". In: *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*. Vol. 55. 1. Chicago, IL, pp. 295–300.

Murer, Martin et al. (2015). "Torquescreen: Actuated flywheels for ungrounded kinaesthetic feedback in handheld devices". In: *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 161–164.

Needleman, Sarah E (2018). "Virtual reality, now with the sense of touch". In: *Wall Street Journal*.

Nelson, Deborah Imel et al. (2005). "The global burden of occupational noise-induced hearing loss". In: *American journal of industrial medicine* 48.6, pp. 446–458.

Oculus (2015). *Michael Abrash Keynote from F8 2015*. Youtube. URL: https://www.youtube.com/watch?v=XVCthGEFwHw.

Ondrejka, Cory (2004). "Escaping the gilded cage: User created content and building the metaverse". In: *NYL Sch. L. Rev.* 49, p. 81.

Persson, Emil (2013). *Skyboxes*. https://opengameart.org/content/beach-skyboxes and https://opengameart.org/content/indoors-skyboxes. (accessed: 20.04.2022).

Sakai, Masataka, Yukio Fukui, and Norio Nakamura (2003). "Effective output patterns for torque display'GyroCube'". In: *Online Proceeding of the 13th International Conference on Artificial Reality and Telexistence, December 3-5, 2003, Keio University, Tokyo, Japan*.

Sherman, William R and Alan B Craig (2018). *Understanding virtual reality: Interface, application, and design*. Morgan Kaufmann.

Stevens, Stanley Smith (1957). "On the psychophysical law." In: *Pschological Review* 64.3, pp. 153–181.

Sutcliffe, Alistair (2003). *Multimedia and Virtual Reality*. 1st. Psychology Press.

Sutherland, Ivan (1965). "The Ultimate Display". In: *Proceedings of the IFIP Congress*.

Tanichi, Taku et al. (2020). "KABUTO: Inducing Upper-Body Movements using a Head Mounted Haptic Display with Flywheels". In: *SIGGRAPH Asia 2020 Emerging Technologies*, pp. 1–2.

Tremblay-Bugeaud, Jean-Félix, Thierry Laliberté, and Clément Gosselin (2020). "Design, Analysis and Preliminary Validation of a 3-DOF Rotational Inertia Generator". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 6427–6433.

Walker, Julie M et al. (2017). "Haptic orientation guidance using two parallel double-gimbal control moment gyroscopes". In: *IEEE transactions on haptics* 11.2, pp. 267–278.

Wilches, Daniel (2019). *Ardity*. URL: https://ardity.dwilches.com. (accessed: 27.04.2022).

Willumsen, Ea Christina and Milan Jacevic (2019). "A Typology of Rumble." In: *DiGRA Conference*.