HaptiSuit - Full Body Hybrid Haptic Feedback for Virtual Reality

Oliver Wing

Exeter College

Supervised by Professor Liang He

Institute of Biomedical Engineering

Department of Engineering Science

University of Oxford

Trinity Term 2024

Abstract

Full body haptic devices provide haptic feedback to users in VR, adding depth and realism to virtual environments. Existing solutions are complex, expensive and most only offer a single form of haptic feedback, limiting possible tactile sensations. HaptiSuit aims to demonstrate a novel design for a dual mode full body suit capable of delivering both vibrotactile and pneumatic feedback in real time and fabricate a working prototype. This is achieved through the design of the Vibro-Pneumatic Unit (VPU), a compact modular unit combining both vibrotactile and pneumatic feedback. Multiple VPUs are fully independently controlled via the fast and scalable master-slave system architecture. The whole design process for the VPU and system architecture is thoroughly analysed, with design choices justified through research and testing. A prototype was constructed using 4 VPUs on the left chest, each of which was able to deliver noticeable sensations of vibration, compression or both in any combination. The effectiveness of HaptiSuit is discussed along with future opportunities for the development of its modular, multi mode design.

Contents

1 Introduction

1.1 Motivations

Virtual reality (VR) is a rapidly growing industry. VR headset technology is constantly improving; advancements in resolution, field of view and tracking mean that VR is more immersive and accessible than ever before. VR experiences are further enhanced by engaging the sense of touch through real-time haptic feedback, corresponding to the users actions within the virtual environment. Users can experience sensations such as impacts, vibrations, textures, and temperature changes. Users can feel the location of virtual avatars or objects in relation to their own body, and the weight or resistance of virtual objects they interact with. This adds depth and realism to virtual environments, leading to a greater sense of immersion.

For VR experiences involving the whole body, full body haptic feedback is necessary. This allows the user to interact with the virtual environment using their entire body. This is further enhanced by combining multiple forms of feedback; using different modes of haptic feedback allows for a broader range of tactile sensations. Full body multi mode haptic feedback has a huge range of applications - gaming, sport training simulation and analysis, virtual tourism and social interactions in VR to name a few.

Full body multi-mode haptic feedback is an active area of research. The limited number of existing full body devices have drawbacks; they are complicated, expensive and most only offer a single mode of feedback. As each type of feedback is designed to simulate a specific sensation (vibration, temperature etc) they are limited in the range of sensations they can produce. This inhibits the sense of immersion; the type and intensity of haptic feedback from virtual interactions often doesn't correlate to the real world sensation which the user expects. Multi mode solutions do not offer full independent control over each haptic subsystem and their designs are inflexible, non modular, and cumbersome.

1.2 Objectives

HaptiSuit intends to overcome the problems of current full body haptic devices through several objectives. The first goal is to design a compact modular unit which combines both vibrotactile feedback and pneumatic feedback, able to be placed on any area of the body to deliver high frequency vibration, low frequency/sustained pressure, or a composite sensation of both.

The next objective is to design a system architecture that allows for full independent control of both

the vibrotactile and pneumatic component of every modular unit simultaneously in real time with minimal delay, scalable to a large number of units without increase in response time or reduction of intensity to provide optimal full body feedback.

The final aim is to fabricate a working prototype with multiple haptic units controllable in real time to demonstrate the effectiveness of the affordable, lightweight, modular dual feedback full body design.

1.3 Project Overview

The purpose of this project is to outline the design of a compact, scalable solution for full body vibrotactile and pneumatic feedback to enhance VR experiences, demonstrating its efficacy by manufacturing a working prototype that can respond to inputs in real time.

To begin, different haptic technologies are discussed along with the existing solutions and their limitations. Each haptic method is compared, justifying the choice of vibrotactile and pneumatic feedback. From this, the conceptual design of HaptiSuits Vibro-Pneumatic Unit (VPU) and dual feedback system architecture is illustrated, revealing the advancements it makes in full body multi mode haptic feedback.

Next is an in-depth analysis of the design process of the vibrotactile subsystem; the choice of motor, the design and fabrication of the motor mount plate, the use of a haptic driver, the use of a multiplexer to control multiple drivers independently and the integration of the separate components into a circuit. This is followed by a comprehensive analysis of the design process of the pneumatic subsystem; the design of the pneumatic actuator air cell, the use of finite element analysis (FEA) and prototyping, the fabrication of the silicone pneumatic unit and the design of the hardware set-up to enable independent control of the inflation and deflation of each pneumatic actuator.

To explain how each subsystem is controlled, the design of the system communication and logic is thoroughly analysed. The use of a master-slave architecture for efficiency and scalability, the use of the I2C protocol to transfer state arrays between master and slaves, the software design of the state machine logic to control each subsystem fully independently in real time and the implementation of a GUI to demonstrate the capability of the system are explained. Finally, the implementation of the full design is demonstrated by constructing a working prototype for the chest. The effectiveness of the design is discussed, highlighting areas where it succeeds and points for improvement.

This project is a proof of concept to demonstrate a method for delivering a combination of vibrotactile and pneumatic feedback through a body suit to be worn alongside a VR headset. Full VR implementation is beyond the scope of this project but the hardware and software of HaptiSuit is designed with the

intention that it can be scaled into a fully functioning VR bodysuit capable of tracking the motion of the user, relating this motion to a virtual environment and responding with detailed pneumatic and vibrotactile feedback in real time.

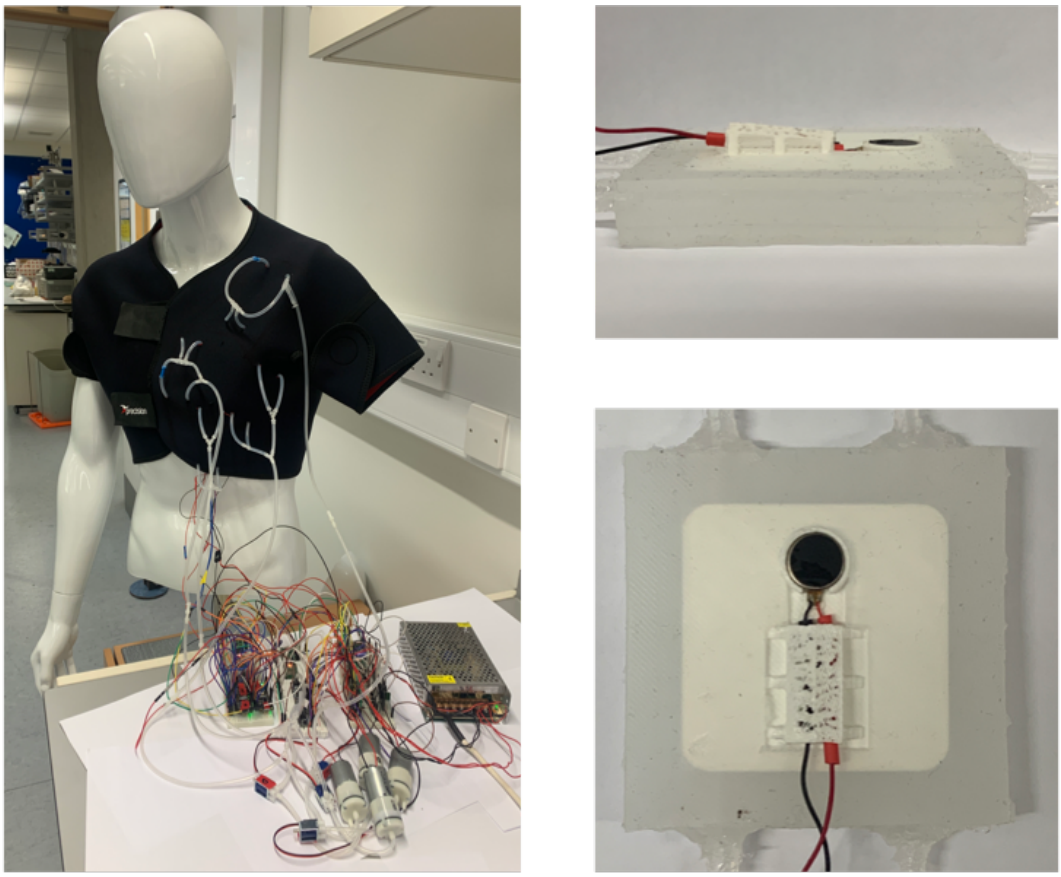


Figure 1.1: Left - HaptiSuit on a manikin. Right - top and side view of VPU.

2 Literature Review

2.1 Human Haptics

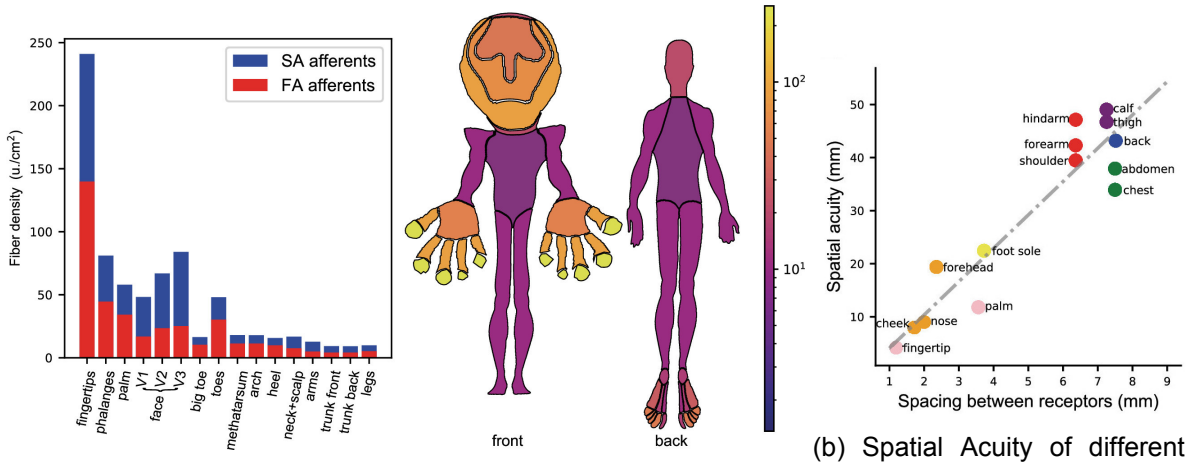
To understand haptic feedback it is necessary to understand how humans perceive tactile sensations. There are two main categories of tactile sensation - kinaesthetic and cutaneous [1]. Kinaesthetic feedback is the reaction force experienced by the body relating to the position, orientation and movement of body parts. An example of this is holding an object; the hand experiences resistance from gripping the object and the arm experiences the force due to the objects weight.

Cutaneous feedback on the other hand is the stimulation of cutaneous receptors; cells beneath the skin responsible for relaying tactile information to the brain. This includes sensations such as pressure, vibration, texture and temperature. The mechanoreceptors are a subgroup of these, of which there are 4 types [2]. Merkel discs respond to pressure. They are slow acting, meaning they produce a sustained sensation when stimulated. Ruffini endings respond to pressure similarly. Meissner corpuscles respond to low frequency vibration (5-50Hz) and light touch. They are fast acting; a sensation is only perceived during a change in stimulation level. Pacinian corpuscles are fast acting and respond to high frequency vibration (30-500Hz).

	Meissner Corpuscle	Pacinian Corpuscle	Merkel Disc	Ruffini Ending
Adaption Rate	Fast	Fast	Slow	Slow
Primary Sensation	Vibration	Vibration	Pressure	Pressure
Frequency Range	5-50Hz	30-500Hz	0-5Hz	Constant

Table 1: Skin mechanoreceptor properties by type.

There is minor deviation in the ratio of fast and slow adapting mechanoreceptors in different areas of the body but on average they are split evenly [3]. However the total density of skin mechanoreceptors varies greatly. The hand (palm, fingertips and phalanges) and face have significantly higher receptor densities than the rest of the body. There is strong correlation between the mechanoreceptor density of a body region and the spatial acuity in that region - the distance above which two separate tactile stimuli can be discriminated (as distance between receptors increases, so does the minimum distance at which two stimuli can be discriminated). Spatial acuity of the chest and torso is around 30mm and 35mm respectively, and is around 40-50mm on the back, arms and legs.



(a) Skin Mechanoreceptor densities.

(b) Spatial Acuity of different body regions.

Figure 2.1: Skin mechanoreceptor densities (a) and corresponding spatial acuity (b) for different regions of the body [3]. Data is for tactile afferent nerve fibers which direct relate to mechanoreceptors.

2.2 Machine Haptics

Haptic technology describes mechanisms used to simulate a tactile experience for the user of a device via haptic feedback [1]. Haptic devices (devices incorporating haptic technology) utilise different methods for creating kinaesthetic and cutaneous feedback. Typically kinaesthetic feedback is delivered by restricting the motion of certain parts of the body in a way related to motion within a virtual environment, simulating the sensation of resistance, weight and solidity when in contact with a virtual object. For a full body suit this requires an exoskeleton like device; these are expensive, cumbersome and complicated so mechanical forms of kinaesthetic feedback are unsuitable for this project (some exceptions are discussed in the literature review).

Cutaneous haptic devices create feedback using actuators; specialised units placed in contact with the user that are able to artificially replicate a given cutaneous tactile sensation. Cutaneous feedback is ideal for a full body suit; by designing actuators that are modular and easily replicated, multiple actuators can be combined into an array to provide cutaneous haptic feedback to a large area of the body. As they do not need to provide resistance to movement, cutaneous feedback actuators can be compact and lightweight. This promotes an affordable and portable design. Several types of cutaneous feedback are discussed here and compared on their viability for use in a full body haptic suit.

2.2.1 Vibrotactile

Vibrotactile haptic feedback provides the user with the feeling of vibration through contact with vibrotactile actuators - small components that generate high frequency mechanical force. The 4 main forms of vibrotactile actuator are the Eccentric Rotating Mass (ERM), the Linear Resonant Actuator (LRA), piezoelectric actuators and solenoids. ERM actuators generate vibration by rotating an off-centre mass (from the point of rotation) with a DC motor [4]. Rotating the mass causes uneven centripetal force, causing the motor to move laterally in the plane of rotation. $F = m_{mass} r\omega^2$ determines the force of the vibration (for an actuator not fixed to another body, explained shortly) where m_{mass} is the mass of the off centre weight, r is the distance between the weight's centre of mass and the axis of rotation and ω is angular velocity [5]. Vibration strength is directly proportional to vibration frequency and thus the amount of DC current applied to the motor. Vibration is measured in acceleration (g). This is because an ERMs vibration strength is dependent on the mass of the object it is mounted to.

$$a_{total} = \frac{m_{mass} r\omega^2}{m_{total}} \tag{1}$$

To take advantage of the DC motors linear correlation between voltage and angular velocity, ERMs can be overdriven for a short period during startup before reducing voltage to a lower steady state value once spinning to increase acceleration. Similar, reverse driving the motor when stopping helps to eliminate undamped vibration and tightens the actuators response time. Specialised motor drivers have these features built in. ERM actuators come in a variety of form factors, including a low profile coin shape commonly used in mobile phones. They are simple and cheap. Coin motors are compact and all moving parts are contained internally.

LRAs contain a mass attached to a spring and a voice coil [6]. The voice coil is driven with an AC current; when the voice coil moves at the same resonant frequency as the spring, noticeable vibration is produced in a single axis. LRAs can only be driven at resonant frequency as there is a significant reduction in acceleration when driven 3Hz or more off resonance [5]. Haptic drivers use resonance tracking to produce the optimal vibration strength. LRAs provide all the same benefits as ERM coin motors; they are effectively interchangeable.

Piezoelectric actuators use piezoelectric material to convert electrical energy into mechanical vibration [5]. When an AC current is applied to the material, it warps up and down which is experienced as vibration. They are highly precise in frequency and amplitude. They are lightweight and flexible; they can attach to

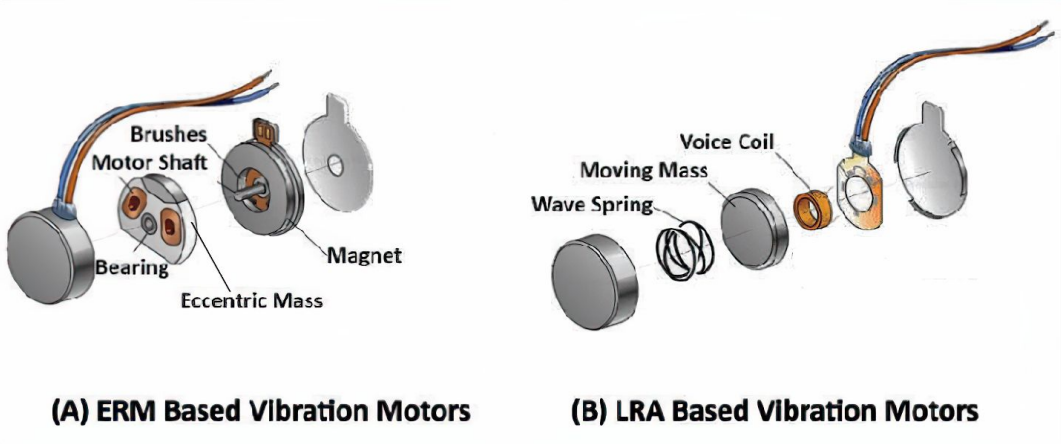


Figure 2.2: ERM and LRA mechanisms. ERM vibration is rotational (2 directions), LRA vibration is reciprocal (1 direction).

either rigid or flexible material which is useful for a full body suit. However, they require extremely high voltage (50V+ peak-to-peak) and cost more than the other options.

Solenoid actuators create vibrations by energising a coil containing a spring loaded mass using pulse-width modulation (PWM) [7]. They are effectively bigger LRAs. They produce strong vibrations but are bulky, require a strong mounting plate and also need custom drivers.

	ERM	LRA	Piezo	Solenoid
Waveform	DC	AC	AC	PWM
Voltage	3-5V_{DC}	3-5V_{RMS}	50-200V_{PP}	5-12V
Acceleration	1.0-1.5G	1.0-1.5G	3G	5G+
Weight	0.9g	0.9g	60mg	$\sim$25g
Response	20ms	10ms	<1ms	10ms
Thickness	2mm	2mm	0.6mm	20mm

Table 2: Quantitative comparison of the 4 main vibrotactile actuator types. Values are approximate and based on currently available products. ERM values are for coin shaped motors.

2.2.2 Pneumatic

Pneumatic tactile feedback delivers a sensation of pressure by pressurising and depressurising actuators in a wearable device. The actuator is typically some form of inflatable air chamber with a large surface area in contact with the user. Elastic actuators are constructed from silicone [8, 9] as it is strong, flexible and can be cast in any shape or size. Their flat 1-2mm thick face expands under pressure. Actuators can also be inelastic, [10] uses a thin thermoplastic bladder which results in faster inflation than silicone. Pneumatic actuators are lightweight and around 8mm in total thickness.

To pressurise the actuator, a pump/compressor drives air into the actuator via pneumatic tubes. The

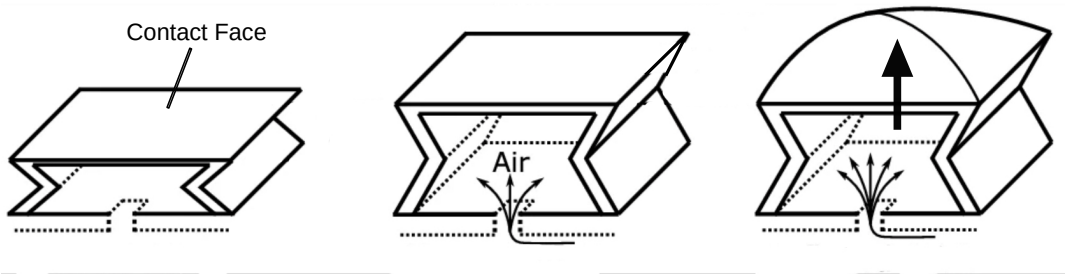


Figure 2.3: Example mechanism for a flexible silicone pneumatic air cell as it inflates. Image modified from [8].

air flow is controlled using solenoid valves. A pressure sensor provides feedback to the controller; this can be used to regulate air flow or just to prevent overinflation depending on if the system uses closed loop or open loop control. Closed loop control gives greater precision over the pressure level of an actuator while sacrificing response time; open loop control has a faster response at the expense of accuracy.

Pneumatic feedback has many advantages for a tactile haptic body suit. Actuators can be arranged in an array such that they cover a large surface area. Depending on the strength of the air compressor, these actuators are capable of producing a strong compressive sensation, both transient and sustained (stimulating the Merkel discs). Using a powerful compressor, [8] is able to generate 800kPa of pressure and up to 50N of impact force. For more portable systems [10], 40kPa to 75kPa is achievable. Small 5V air pumps are rated at 58kPa. One drawback is a slower response time than other feedback methods; [8] took around 100ms and [10] 120ms to reach full inflation. This is unfavourable for VR which needs minimal delay to provide convincing feedback. Additionally the air compressors are noisy which detracts from the sense of immersion when in a virtual environment.

2.2.3 Electrical

There are 2 main forms of electrical haptic feedback. The first is Neuromuscular Electrical Stimulation (NMES). NMES elicits action potentials on motor nerves using non invasive surface electrodes, causing contraction of muscle fibers [11]. NMES is capable of producing both cutaneous [12] and kinaesthetic feedback [13]. The second form is Transcutaneous Electrical Nerve Stimulation (TENS). TENS also uses non invasive surface electrodes to elicit action potentials but targets sensory nerves [14]. This creates a wide range of sensations, from tingling to buzzing to intense shock, depending on the strength of the signal applied to the electrodes. As motor and sensory nerves are stimulated directly by the circuit, the feedback response is instant. Due to the slim form factor and flexibility of the actuators (surface elec-

trodes), they can be incorporated easily into a full body suit. Full body haptic suits utilising a combination of NMES and TENS feedback are already available commercially [15]. These products are very expensive.

There are several drawbacks to using electrical feedback. It is inconsistent; the strength of the sensations differs between users due to differences in material properties of skin (thickness, conductance etc) and pain threshold. Thus devices requires calibration on a per individual basis. Electrical feedback is also potentially dangerous as current is being applied directly to the user. NMES requires large voltage (around 60V [11]) which complicates the design.

2.2.4 Thermal

Thermal haptic feedback uses actuators called Peltier modules [16, 17]. These make use of the Peliter effect, a phenomenon where a temperature difference is created across a junction of two different conductive materials when an electric current is passed through them [18]. Depending on the direction of current flow, the Peltier module will transfer heat either into or out of the user, simulating the sensation of either a hot or cold surface. Thermal actuators are lightweight and flexible meaning they can be integrated into a full body wearable device. However they are very expensive relative to the other actuator types and have slow response time, taking around 10s to change temperature from ambient (21°C) to either hot (35°C) or cold (15°C) [16].

2.2.5 Non Contact

There are also non contact methods of providing cutaneous feedback. These include using air vortices [19] and ultrasonic vibrations [20]. Non contact methods are suitable for localised feedback but they are not effective for a full body suit. Air vortices are extremely low force (0.17N at 2.5m) and their slow speed (9.1m/s) means an increasingly slower response as the user moves away from the source. Ultrasonic vibrations is also extremely low force (0.016N) and has a usable range of less than 40cm. These limitations make it unfeasible to keep a non contact feedback mechanism continuously calibrated with the movement of the entire body. For a full body suit, feedback must be delivered via actuators in contact with the user as actuators stay in the same position relative to the body. This ensures feedback is consistent in response and intensity, regardless of the users position or orientation.

2.3 Existing Solutions

As stated earlier, full body haptic feedback is an emerging area of research. There is a very limited number of existing solutions, commercial or otherwise. The 4 most prominent are discussed here.

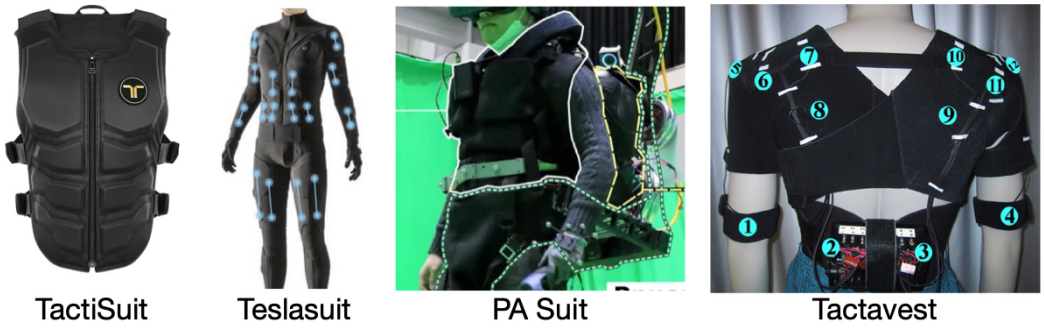


Figure 2.4: Existing solutions. Images from [21], [15], [8] and [22] respectively.

TactiSuit [21] is a vest covering the chest, torso and back containing 40 vibrotactile actuators. It is designed for VR gaming. The use of ERM coin motors to provide vibrotactile feedback means TactiSuit is lightweight and has a fast response but as it is purely vibrotactile, it cannot produce low frequency or sustained force. It retails for $529.

Teslasuit [15] is a commercial full body suit that delivers a combination of NEMS and TENS feedback through 80 channels connected to electrodes. It can deliver a wide range of haptic sensations to the entire body. The major drawback is its price of $12,999.

PA Suit [8] is a design for a full body haptic suit combining pneumatic and vibroacoustic feedback. It consists of an array of 80 silicone pneumatic actuators and 8 vibrotactile motors (connected to an audio source) arranged on the chest. It is the only solution with multiple modes of feedback. PA Suit suffers from a heavy pneumatic system, requiring a support frame for the suit to be usable for long periods of time (90+ min). As the vibrotactile motors are connected to the left and right channels of an audio source, they cannot be controlled fully independently.

TactaVest [22] is a design for a vibrotactile vest, covering the chest, torso, back, shoulders and elbows to enhance collisions in a virtual environment. It is very similar to TactiSuit, again lacking range in the possible sensations the user can experience due to purely vibrotactile feedback.

	TactiSuit	Teslasuit	PA Suit	TactaVest
Feedback	Vibrotactile	Electrical	Pneumatic, Vibroacoustic	Vibrotactile
Actuator Type	ERM (Coin)	Electrode	Silicone Air Cell & ERM	ERM (Coin)
Number of Actuators	40	80	80 (Pneu) 8 (Vibr)	16
Output (Max)	1G	150mA	51N (Pneu) 3G (Vibr)	0.85G
Response (Avg)	<10ms	Instant	75ms	<10ms
Weight	Light	Light	Heavy	Light
Cost	$529	$12,999	N/A	N/A

Table 3: Properties of the 4 existing full body haptic devices. Output unit corresponds to feedback type.

	Vibrotactile (ERM)	Pneumatic	Electrical	Thermal
Waveform	DC	DC/PWM	AC	DC
Voltage	3-5V	5V+	60V	12V
Response	10ms	100ms	Instant	$\sim$10s
Weight	0.9g	20g	<1g	10g
Thickness	2mm	8-20mm	<1mm	<2mm
Amplitude	Low	Medium-High	High	Low-Medium
Frequency	150-180Hz	Constant	60Hz	Constant

Table 4: Quantitative comparison of the 4 contact based cutaneous feedback methods. Values are approximate and based on current literature.

2.4 Comparison of Feedback Methods

The 4 actuator based feedback methods are compared in Table 2. As all actuator types are slim and lightweight, the main factors considered were the response time, safety, scalability and ease of integration into a circuit. Despite the capability to simulate a wide range of kinaesthetic and cutaneous effects with immediate response, electrical feedback is unsuitable due to its danger and inconsistency in perceived sensation across different users, as well as the high voltage requirement. Thermal is unsuitable due to the significantly slower response time of the Peltier module actuators and their cost making a scalable design expensive.

The most suitable combination of feedback methods for a scalable full body haptic feedback suit is vibrotactile and pneumatic. They both run on DC at similar voltages, simplifying the circuit design. Vibrotactile actuators have a fast response, vibrate at a high frequency but with low amplitude. Pacinian corpuscles respond better to transient stimulation. Pneumatic feedback has a slightly slower response, but provides a sustained force at larger amplitude. Merkel discs respond better to sustained stimulation. By combining the 2 methods, their strengths complement the others weakness.

3 Design Concept

The literature demonstrates that current full body haptic devices have limitations. They are complex, expensive and restricted in the tactile sensations they can produce due to most only offer a single form of feedback. The one multi mode solution (PA Suit [8]) doesn't offer full independent control over each haptic subsystem and its design is non-modular and cumbersome.

To reach a solution to the current limitations of full body haptic feedback, several design criteria are established that HaptiSuit must meet. Firstly HaptiSuit must be capable of producing a wide range of tactile sensations. The most effective approach is using 2 modes of haptic feedback. As explained in the comparison on cutaneous haptic feedback technologies, vibrotactile and pneumatic are the most suitable to achieve this goal.

Next both vibrotactile and pneumatic modes of feedback must be combined into the same unit with a clean, simple and compact design. This promotes modularity which allows multiple units to be used in conjunction across the entire body to deliver effective full body feedback without encumbering the user.

Both vibrotactile and pneumatic subsystems must be fully independently controllable across all units, reinforcing modularity and detailed full body feedback.

The architecture of the system must be scalable to a large number of actuator units with no decrease in performance (response time or intensity); this is essential for a full body haptic device. It must also be capable of responding quickly to inputs. Minimal latency is important for creating a sense of immersion in VR experiences.

Finally the design of HaptiSuit must be affordable and able to be manufactured from readily available components.

3.1 VPU

HaptiSuit produces both vibrotactile and pneumatic haptic feedback via modular dual feedback actuator units called Vibro-Pneumatic Units (VPUs) (see diagram 3.1). The body of each VPU is a 64x64x8mm flexible silicone cuboid. Firmly embedded in the outer face is a rigid PLA plastic plate, to which an ERM coin vibration motor (the vibrotactile actuator) is securely mounted. Within the silicone on the inner face are 4 symmetrical inflatable air cells (the pneumatic actuator) each of 10mm diameter spaced 32mm apart. For clarity the combined actuator units will be explicitly referred to as Vibro-Pneumatic Units (VPUs) for the rest of this report. Any use of the term actuator will be in relation to a specific haptic technology;

vibrotactile actuator refers to the vibrating motor and pneumatic actuator refers to the inflating air cells.

VPUs are attached to the inside of a tight suit made of strong flexible fabric. As shown in 3.1, the inner face of the VPU (where the pneumatic air cells are located) is in contact with the user. When energised, the ERM coin motor vibrates the entire VPU due to the wide flat embedded mount plate. This delivers a sensation of vibration to the user. When inflated, the 4 air cells press into the users skin, delivering a feeling of compression. Each sensation is delivered at the location of the VPU. For this project, 4 VPUs attach to a tight neoprene chest brace to provide full haptic coverage of the left chest.

Both vibrotactile and pneumatic feedback are combined into a single modular unit. Their slim form means they are lightweight, compact, and fit comfortably on the inside face of the suit, ensuring the user can move freely while wearing the device. The square shape tessellates so can be arranged both sparsely and densely. This modularity means multiple VPUs can be arranged on the users body such that the density of VPUs corresponds to the spatial acuity of the respective region of the body, regions of high spatial acuity benefit from more closely arranged VPUs. This scalability is crucial for optimal full body coverage.

The combination of these 2 feedback methods means each VPU is able to deliver 3 distinct sensations - high frequency vibration, low frequency/sustained pressure, or a composite sensation of both. This allows HaptiSuit to produce a wide range of tactile sensations.

VPUs are designed to deliver dual feedback in the same location. Each feedback mechanism works in synergy with the other to further increase their effectiveness. The rigid base plate of the vibrotactile motor amplifies the intensity of the pneumatic feedback by preventing the pneumatic actuators from expanding outwards, increasing the compressive force into the user. The flexibility of the silicone means VPUs follow the curvature of the human body in any region; this ensures a large contact area with the inner face and the user which increases the transmission of vibration.

3.2 System Architecture

HaptiSuit uses a master-slave configuration (see diagram 3.2). The vibrotactile and pneumatic feedback is split into 2 separate subsystems, each controlled by a Teensy 4.1 microcontroller (Arduino based). Both subsystems are coordinated by a master Teensy. To demonstrate the capabilities of the suit within the scope of the project, a GUI is used to send haptic states to the system.

The vibrotactile actuators (ERM motors) on each VPU is connected to the terminals of its own haptic motor driver. Each motor driver is then connected to a multiplexer, which in turn connects to the Teensy

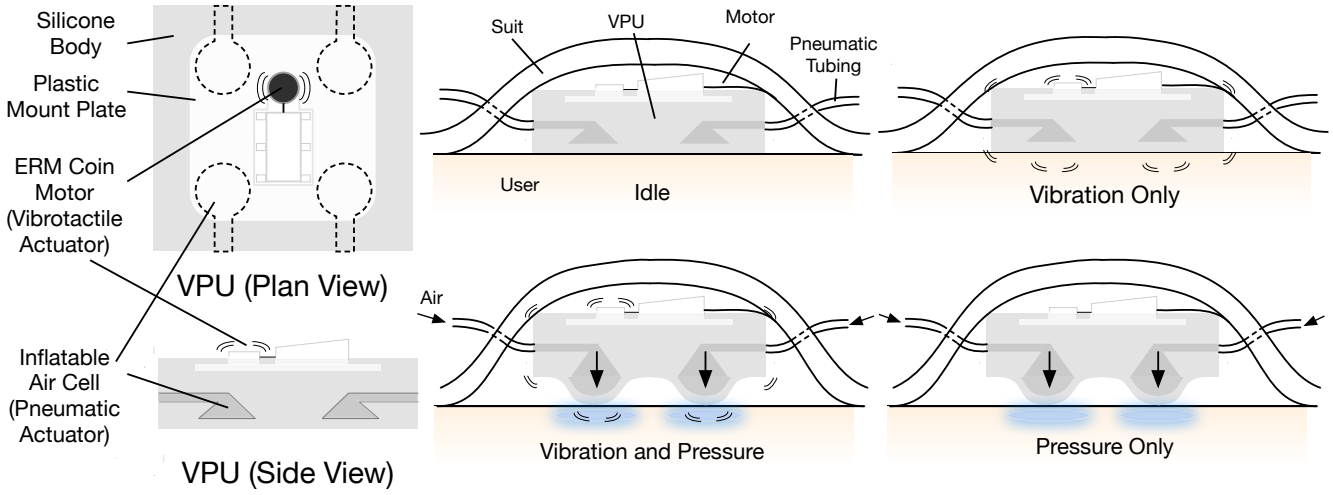


Figure 3.1: VPU diagram and VPU concept for each combination of vibrotactile and pneumatic feedback.

vibrotactile slave. The Teensy controls the state of each haptic driver independently; their states are managed by a state machine and the controller communicates via the I2C protocol (explained further in 6.4).

The pneumatic actuators (inflatable air cells) on each VPU are connected with pneumatic tubing to its own pump/valve mechanism. Each pump/valve mechanism is independently controlled by the Teensy pneumatic slave via time based open loop control. Like the vibrotactile slave, the state of each pneumatic actuator is managed by a state machine and the pump/valve mechanism is controlled digitally with transistors (explained further in 5.4).

This system architecture promotes fully independent control, scalability and a fast response in 2 ways. First is the use of the master-slave configuration; dividing the management of each haptic feedback subsystem between 2 slaves increases the efficiency of the code and the speed of response as each slave is only responsible for a single system and is easily scaled by adding additional slaves without introducing latency. By running concurrent state machines on separate slaves, each system is controllable fully independently in real time.

Second is the fact that each VPU has its own haptic driver for its vibrotactile actuator and its own pump/valve mechanism for its pneumatic actuator. Having a dedicated haptic driver and pump/valve mechanism for each VPU is crucial to controlling the state of each feedback method independently across all VPUs. It also ensures a consistently fast response (when combined with the open loop pneumatic control) and that the design can be scaled to many VPUs with no reduction of feedback intensity. This is crucial for full body multi mode haptic coverage.

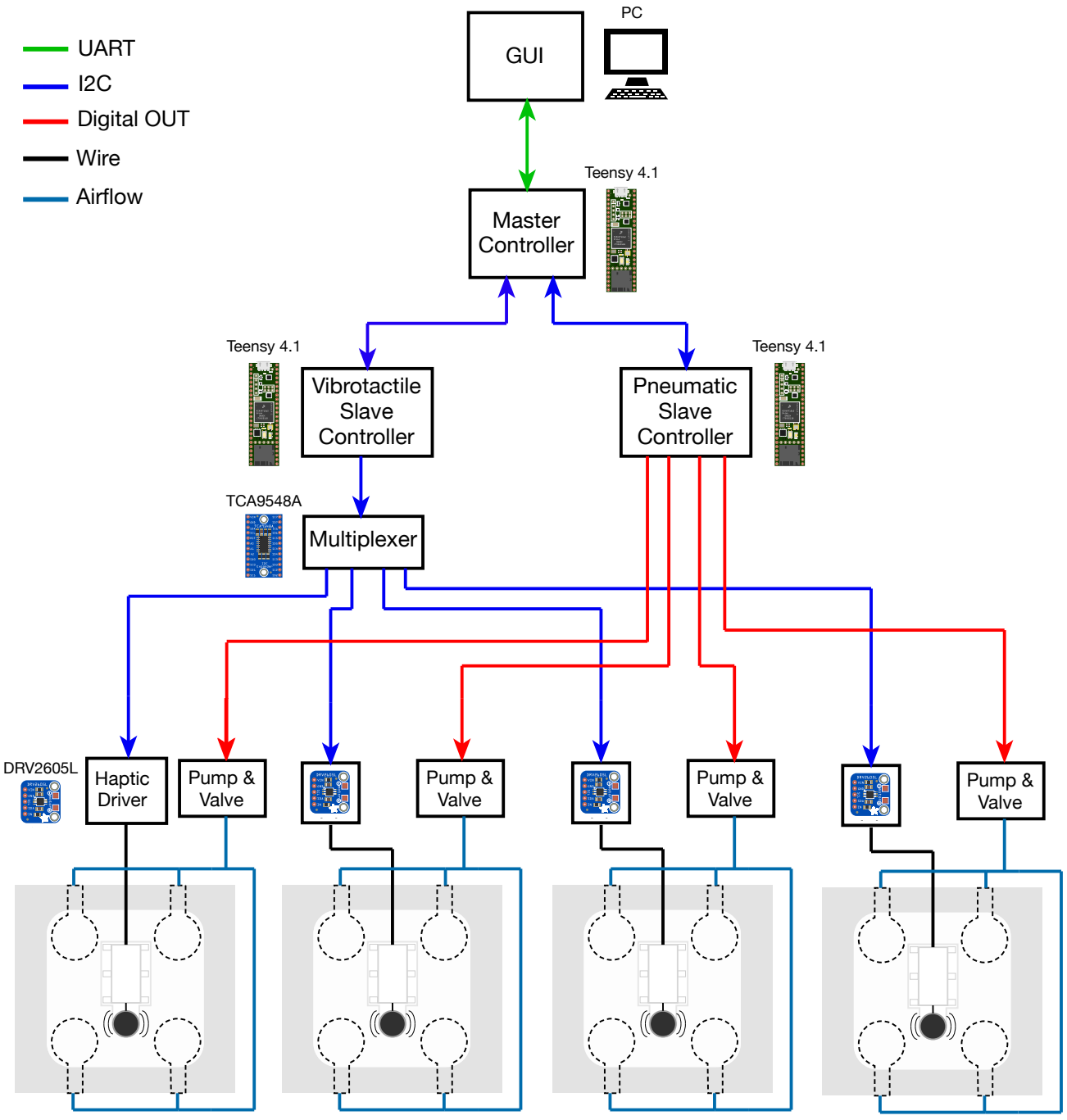


Figure 3.2: Overview of HaptiSuits system architecture. 4 VPUs are used in this project but the design is scalable to many more. The pressure sensors for each VPU have been omitted for clarity; as HaptiSuit uses open loop control they are only a safety measure and not essential for the function of the suit.

4 Vibrotactile Feedback

The first type of haptic feedback utilised by HaptiSuit is vibrotactile feedback. For each VPU, a small vibrating motor is mounted on a rigid PLA plastic plate. The mounting plate is then securely embedded within the flexible silicone body of the VPU. The motor is connected to the terminals of a haptic motor driver. Each motor driver is then connected to a multiplexer, which in turn connects to a microcontroller. The microcontroller controls the state of each vibrotactile motor independently via the I2C protocol and a state machine, explained further in 6.4.

4.1 Vibration Motor

As explained in the literature review 2.4, there are 4 different vibration motor technologies capable of producing vibrotactile feedback. These are Eccentric Rotating Mass (ERM), Linear Resonant Actuator (LRA), Piezoelectric, and Solenoid. When designing the vibrotactile subsystem, each vibration method was considered for its strengths and weaknesses in relation to a full body wearable haptic device to determine which was the most suitable for HaptiSuit.

Solenoid actuators are capable of generating strong vibrations, which can be controlled in magnitude using either a Pulse Width Modulation (PWM) signal or a specialised haptic driver. Strong vibration means more intense haptic sensations for the user. However, solenoids have a large form factor compared with other methods. This is significant for a wearable device as ease of movement and comfort of the user cannot be compromised; this is crucial to creating an immersive VR experience. Due to their greater size and more powerful vibration, solenoids also require a large mounting plate to ensure they remain secure when energised and that the vibration generated is transmitted through the suit to the user effectively. This mounting plate would increase the size of the actuator unit, making it harder to incorporate into the suit. Furthermore, the combination of the protruding form factor stretching the suit and the powerful vibration means it would prove challenging to keep the feedback localised to the actuator and prevent dispersion throughout the suit, thus solenoids were deemed unsuitable.

Piezoelectric actuators are small and lightweight. This means that many of them can be arranged over small area. For a wearable device this is a useful property; the more densely actuators are arranged on the device, the finer the resolution of the haptic feedback perceived by the user. If the device focuses on one specific area of the body this is ideal but for a full body suit it cannot be easily implemented; as the number of actuators increases, so does the complexity of the wiring. Due to the scope of this project, this

enforces a limit on the actuator density across the suit. Additionally, many of the actuators are located on the chest and back which are regions of the body where skin mechanoreceptors are less dense [3]. On the chest and back, the brain perceives haptic sensations in less detail than other areas of the body (e.g the hands). This means that using fewer actuators able to vibrate strongly is more effective than using many weakly vibrating actuators. Thus piezoelectric technology is not suitable as the vibration is too weak to provide convincing haptic feedback in a full body suit.

LRAs and ERMs have many similarities; both are suitable for providing vibrotactile feedback in a full body haptic suit. Each are available in a variety of form factors, most notably coin motors which are small and streamlined yet capable of generating noticeable vibration. The circular shape and extremely low profile allows them to be easily fixed to a mounting plate with a simple interference fit. This is sufficient for securing the actuator; their less aggressive vibration means the motor remains solidly attached to the plate with repeated use. The reduced form factor means they can be easily fitted beneath the fabric of a wearable haptic device, allowing fully unrestricted movement when wearing the suit. The low profile and moderate power also means the vibration can be reliably directed through the actuator unit, without risk of dispersion over a wider area. This ensures the user experiences precise and accurate haptic feedback. A subtle difference is that LRAs vibrates at a single frequency while an ERMs vibration strength is proportional to its frequency. ERMs also vibrate in 2 axes (x and y) vs an LRAs 1 (z). Both are usable for HaptiSuit but ERMs were chosen for their movement in 2 axes being slightly better at transmitting vibration feedback through the VPU.

The ERM motor chosen for HaptiSuit was the Seeed Vibration Coin Motor. It produces 1G (unit of vibration strength) vibrations at up to 10,000 RPM which create a noticeable sensation. It has a resistance of 6.5Ω, suitable to be used by the haptic motor driver to not draw excessive current. It is circular - 10mm in diameter and 2mm in height which allows it to be mounted on a thin plate and fitted cleanly between the silicone pneumatic unit and fabric layer of the suit.

4.2 Mounting Plate

To transmit vibration effectively, it is recommended to securely mount ERM motors within a stable structure, typically a flat rigid surface [7]. This guarantees the motor will not behave unpredictably when powered and maximises the vibration energy felt by the user. Additionally the mount structure can be designed to protect the fragile wires of the motor. This is crucial; all wires and solder connections must be robust and reliable. If not properly fixed they are shaken whenever when the motor vibrates.

The design of the mount was iterated several times before reaching the final form shown above. Along with the above, the main considerations for its design were a slim form factor and means of combining securely with the silicone air cell unit. As VPUs are positioned underneath the fabric of the suit, it is important that each unit be relatively thin. As a result the vibrotactile mount must also be thin to keep each VPU as flush as possible with the rest of the suit. Combining the vibrotactile mount with the silicone pneumatic component in an elegant way is important to creating a single unit that can be duplicated into multiple VPUs. Because of this, the designs of the two elements of the VPU did not evolve independently but influenced one another. This section of the report focuses on the vibrotactile mount; the specifics of the design process for the pneumatic element are further explained in 5.4.

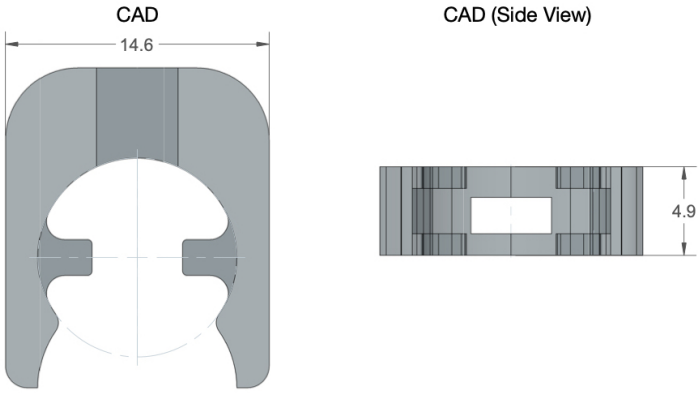


Figure 4.1: Mount design 1 CAD drawing.

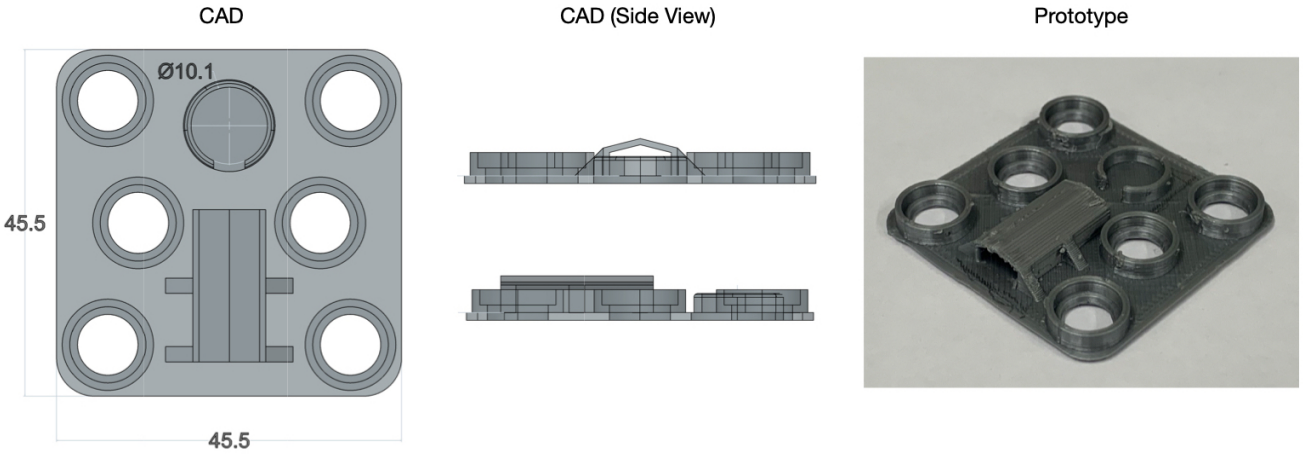


Figure 4.2: Mount design 2 CAD drawing and prototype.

The first design 1 encases the motor in a firm plastic shell which is then embedded using a cavity inside the silicone. The casing clips onto the motor, holding it in place whilst sheathing the wires. However, this design has 2 major flaws. The first is the shape of the casing; according to the literature a wide, flat surface

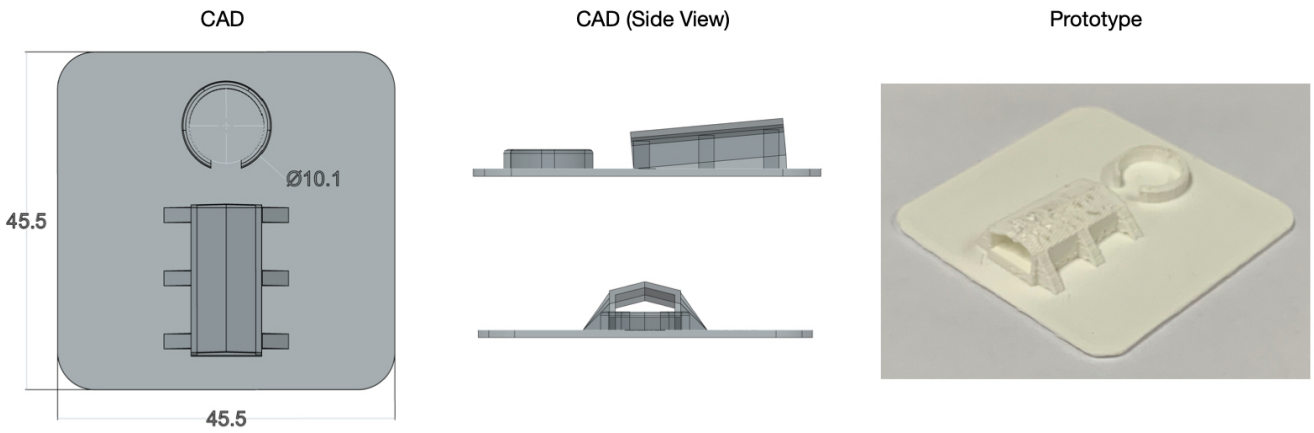


Figure 4.3: Mount design 3 (final) CAD drawing and prototype.

is preferable for mounting the motor as it transmits vibration more evenly. The small compact shape of 1 is not ideal for this, potentially reducing the effect of the vibration felt by a user. The second flaw is the thickness. As the casing is fully embedded within the silicone, the thickness of the VPU is increased. Additionally the configuration of the inflatable air cells is affected; air cells need to be positioned around the casing. This decreases their effectiveness as the silicone unit deforms on both its inner and outer face, resulting in a weaker compressive force experienced by the user.

To correct the flaws of the 1, 2 mounts the motor on a thin flat Polylactide (PLA) plate using an interference fit, along with a simple guard is used to protect the wires. Initially this plate was intended to connect to a wider flexible plastic membrane to control the distribution of the vibration across the suit but this couldn't be incorporated as 3D printing using 2 different types of plastic isn't feasible. Normal adhesives are ineffective for attaching silicone to other materials as silicone will only bind to other silicone, so to fix the plate to the pneumatic component, connectors are used [see diagram]. The silicone component is cast, then the mount plate placed on the outer face. More silicone is poured into the holes which once solidified, attaches the plate. However, upon testing it became clear that this method wasn't able to reliably keep the plate attached; bending the silicone body would cause the connectors to break. This was unsatisfactory considering the pneumatic unit is deformed by the inflation of the air cells and also naturally by the curvature of the human body.

The final design 3 combines the successful elements of the previous designs by partially embedding a flat plate inside the silicone pneumatic unit. Taking the plate shape of 2 ensures a slim form factor and by embedding the mount using a cavity created when casting the silicone, the plate remains secured within the silicone body when the silicone is deformed by inflation or otherwise. The plate also guarantees

the air cells inflate into the user and not outwards, increasing the perceived pneumatic effect. This has the added upshot of reducing the pressure and deformation needed to create a compressive sensation, increasing the safety factor of the pneumatic system. At the same time the manufacturing process is greatly simplified as the silicone for one VPU can be cast in a single step. A 3D printed silicone mould is required to create the pocket (see 5.4). To keep the wires flush with the surface of the silicone the wire sheath is sloped upwards 1.5mm.

4.3 Haptic Driver

To drive an ERM motor, a DC signal with sufficient amps is applied across its terminals. The first way to achieve this is directly from the microcontroller. A PWM signal on a hardware pin is typically used to control the motors behaviour. One major disadvantage of this is that any interrupts (see 6.4) will cause considerable jitter to the signal. If the microcontroller is communicating on an I2C bus, this is guaranteed as the I2C protocol relies on interrupts. Even with PWM it is challenging to control the vibration magnitude as it is proportional to the voltage across the motors terminals which cannot be changed easily. Each motor requires one hardware pin, limiting the number of vibrotactile actuators to the available pins on the microcontroller board. This method also requires using transistors and diodes (see section 5.4) to regulate the flow of current through the microcontroller. The second way of controlling the motors is using a specialised haptic motor driver. Haptic drivers connect to the microcontroller and use an onboard microprocessor, allowing full control over the vibration of the motor. HaptiSuit employs this method with the DRV2605, a small haptic driver designed specifically for powering small ERM vibration motors. The DRV2605 relieves the microcontroller of the need to generate PWM drive signals, removing the problem caused by interrupts. Hardware pins are also not required as the DRV2605 uses I2C to communicate with the microcontroller; scalable to a large number of motors with a multiplexer. The DRV2605 operates at 5V, allowing it to run on the same power rail as the rest of the components in Haptisuit.

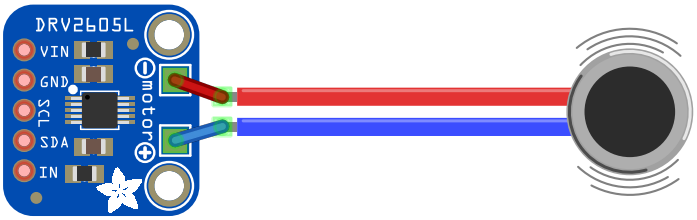


Figure 4.4: DRV2605 haptic driver connected to an ERM coin motor.

Commands are sent to the DRV2605 on the I2C bus (explained fully in 6.4). These come in the form of an 8 bit register address followed by a byte (8 bits) of data for that register address. The Adafruit_DRV2605.h library for Arduino provides easy to use functions and constants to program the device (the Arduino programming language is a subset of C++; see 6.4). The class Adafruit_DRV2605 contains member functions for setting the device mode.

The DRV2605 has 2 powerful features that HaptiSuit can utilise- integrated libraries of premade haptic waveforms and real time playback mode. These modes are alternated simply by writing the corresponding value in bits (0 for premade waveforms and 5 for real time playback) to the 8 bit mode register. The function drv.setMode() does this. Integrated library mode allows up to 7 different pre-made haptic effects (from a library of 123 waveforms) to be sequenced and played back upon request. Real time playback mode drives the motor with a continuous signal. By sending a value between 0 and 127 over I2C to the driver, the output voltage is set linearly from full brake (no vibration) to rated voltage (maximum vibration). In testing it was found a minimum value of around 20 is needed due to the motor startup voltage. HaptiSuit is capable of using both feedback modes; this project uses real time playback mode as it can be demonstrated more intuitively with the GUI.

4.4 Multiplexer

A multiplexer is an electronic component that acts as a switch, selecting between multiple input lines and forwarding the selected input to a single output line. The inverse of this is a demultiplexer; it selects from multiple output lines and forwards a single input line to the selected output. Typically these functions can be performed by the same component, so confusingly a device acting as a demultiplexer is often still referred to as a multiplexer. HaptiSuit only uses demultiplexers so for consistency, the term ”multiplexer” refers to a 1 input to multiple output switch for the rest of this report.

The DRV2605 is hardcoded to the same I2C bus (see 6.4) making it impossible to To control multiple DRV2605s individually. To circumvent this the circuit was initially designed with a basic digital multiplexer. Digital multiplexers work by using n selector inputs S to select between 2 to the n channels corresponding to outputs Z. The 3 selector pins S0, S1 and S2 of the multiplexer are connected to 3 digital output pins on the Arduino. The Vcc pins for the drivers are connected to the 8 lines on the multiplexer and the common line is connected to power; the multiplexer acts as an 8:1 switch. The drivers are connected on an I2C bus to the microcontroller. To activate a driver, the Arduino digital output pins are set accordingly and a command sent via I2C.

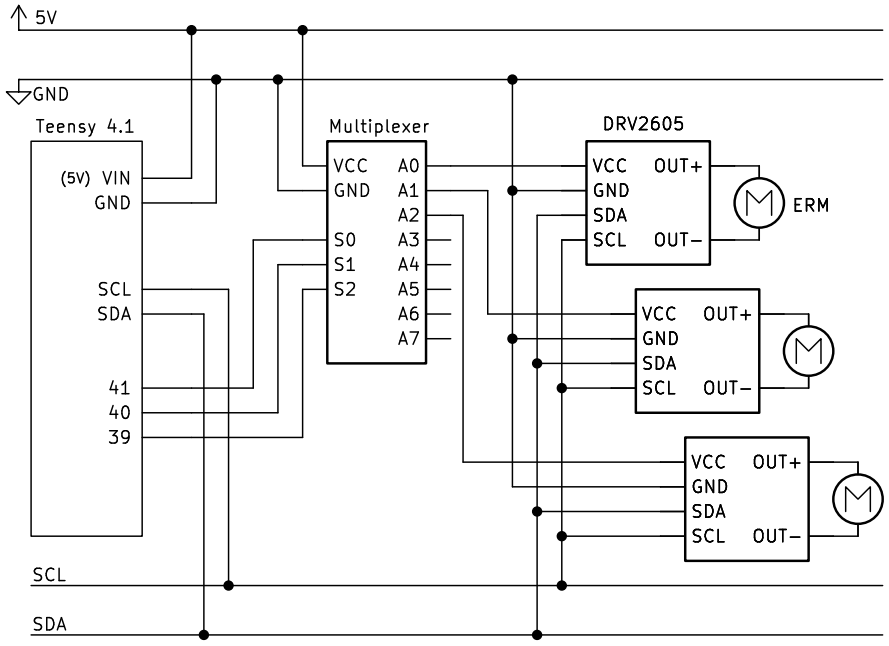


Figure 4.5: Multiple driver configuration using digital multiplexer. This design fails due to hotswapping and only being able to power 1 ERM at a time.

This circuit fails because it hotswaps between drivers, which I2C is not designed to do. Hotswapping is the operation of inserting/removing a device from a system while the system remains powered. This causes the data and clock lines to desync. It is also highly limited in that it doesn't allow for more than 1 driver to be powered simultaneously.

The I2C protocol uses addresses; I2C slave devices read data from the I2C bus them when the address sent by the master matches that of the slave. For devices with different addresses, no multiplexer is needed as the master communicates with each slave device independently by specifying their respective address. However, this presents a problem when using multiple DRV2605s on the same I2C bus as all share a fixed address. To use them on the same bus a specialised I2C multiplexer is needed - the TCA9548A. The TCA9548A switches between each of its 8 ports while keeping the data and clock lines synchronised with the master; 8 copies of the DRV2605 can be connected to a single TCA9548A and communicated with independently. Additionally by using the 3 address selector pins, the I2C address of the TCA9548A itself can take 1 of 8 values; up to 8 TCA9548As can be connected on the I2C bus without address conflicts. This allows for up to 64 (8x8) DRV2605s to be driven on the same I2C bus by a single microcontroller.

The 8 bit control register determines which ports to select. Writing either 1 or 0 selects or deselects

a given port. The function tcaselect does this.

```
#define TCAADDR 0x70 // TCA9548A hard coded I2C address
void tcaselect(uint8_t i)
{
  if (i > 7) return;
  Wire.beginTransmission(TCAADDR);
  Wire.write(1 << i);
  Wire.endTransmission();
}
```

Listing 1: function used to activate TCA9548A ports

Wire.write() sends a single byte. The « operator shifts the variable to the left of the operand to the left by the number of bits specified by the right of the operand. This selects 1 port while deselecting the other 7. The TCA9548A can actually select any combination of the 8 ports but because the DRV2605 remains powered even when its port isn't connected, it is simpler to select each port individually when updating actuators.

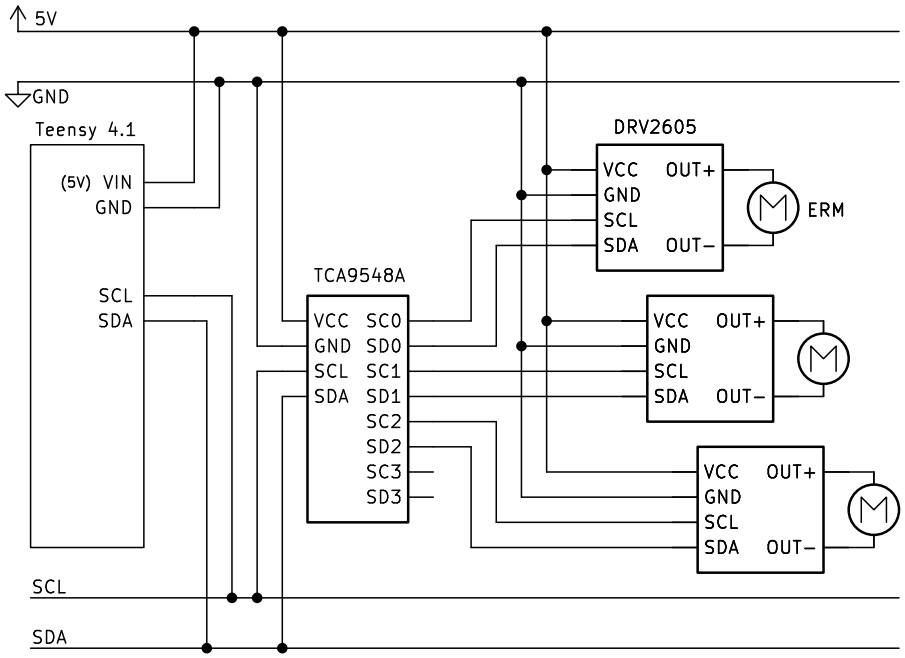


Figure 4.6: Multiple driver configuration using TCA9548A used in HaptiSuit. 3 DRV2650s illustrate the principle but this design can be expanded to up to 64 drivers.

The I2C multiplexer greatly simplifies the circuit. The TCA9548A connects to the I2C bus on the microcontroller with each DRV2605 connecting to a port on the TCA9548A. The microcontroller, multiplexer and drivers all connect to a common 5V power line and common ground. Each driver can be controlled fully independently with multiple drivers running at different vibration strengths simultaneously (see 6.4 for software details).

5 Pneumatic Feedback

HaptiSuits second mode of haptic feedback is pneumatic. The main body of each VPU is a slim, flexible silicone cuboid with 4 symmetrical air cells. The air cells are inflated and deflated using a pump and a valve controlled by a Teensy 4.1 microcontroller with the aid of pressure data from a barometer. The Teensy manages the state of each VPU's pneumatic component (state machine explained in 6.4). When inflated, the air cells push into the users skin, delivering a feeling of compression.

5.1 Air Cell Design

The pneumatic unit must be a solid, inflatable object within which the vibrotactile mounting plate is embedded; as per the literature the most suitable material to fabricate the air cell from is silicone rubber. Silicone has many useful properties; it is liquid when initially mixed which means it can be moulded to any shape; it is strong and elastic so can expand and return to its original form after a cycle of pressurisation and deflation (with some hysteresis depending on factors discussed later); it is flexible so is able to fit the curved surface of the users body.

To ensure a modular design, each pneumatic unit is a small square 64mm x 64mm, with a thickness of 8mm. Each element is identical and the small size uses minimal silicone, making prototyping significantly faster. The square shape tessellates; placing them side by side covers a large area of the body. The small size means their placement can be optimised to correspond with a body regions mechanoreceptor density.

The main factor influencing the stress response of the pneumatic unit is the design of the air cells. This includes the thickness and surface area of the expanding face (the part in contact with the user), and the shape and depth of the air chamber. Thicker silicone over the expanding face clearly results in less deformation but the impact from the other factors is not immediately obvious; Finite Element Analysis (FEA) and prototyping is needed (discussed later). For the air cell shape and configuration within the silicone unit, 2 main designs were investigated.

The first design 1 is a symmetrical map of 18 air cells. The air cells have a conical shape with a small area on the inward face. To create the air pockets in the silicone unit, first a base layer of silicone is cast from a simple square mould. A PLA map of the air cells is then placed on top and silicone poured in. Once the silicone is solidified the PLA mould is removed by pulling the arms from each air cell out, leaving an air chamber within the silicone body. Because the air cell negative must be removable once

the silicone is set, the connecting arms cannot contain any angles smaller than 135°.

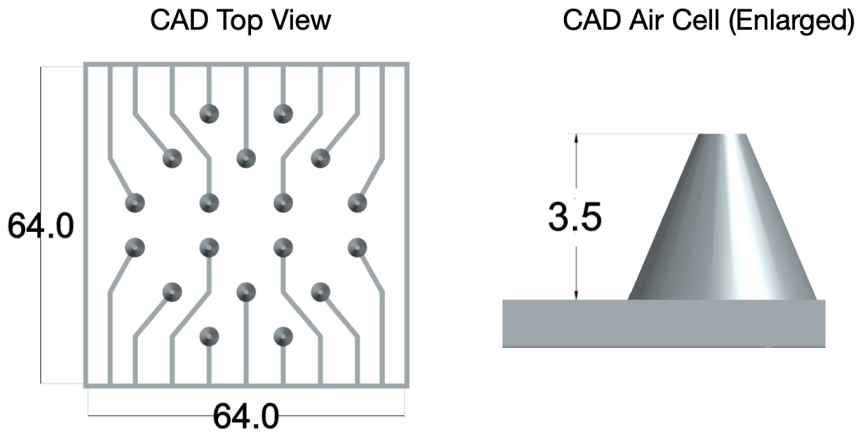


Figure 5.1: Air cell mould design 1, full view and enlarged view of single air cell.

The air cell map is simplified in 2 to a 2 x 2 square. Each cell is positioned 32mm from orthogonally adjacent cells; as each pneumatic unit is 64mm x 64mm this means when placed side by side air cells are still 32mm from the closest in their neighbouring unit. Spatial acuity is the distance above which 2 separate contact stimuli can be discriminated. For the chest, torso and back, this distance is around 30mm. This means the air cells are at the maximum density per unit surface area; any closer would be imperceptible. The shape of the air cell is changed from a a narrow cone to a wide flat disc, greatly increasing the area of the cell on the inward face. The maximum possible diameter of the cell (10mm) is limited by the ability to remove the mould from the body through the connecting tunnel (2mm wide) without causing damage.

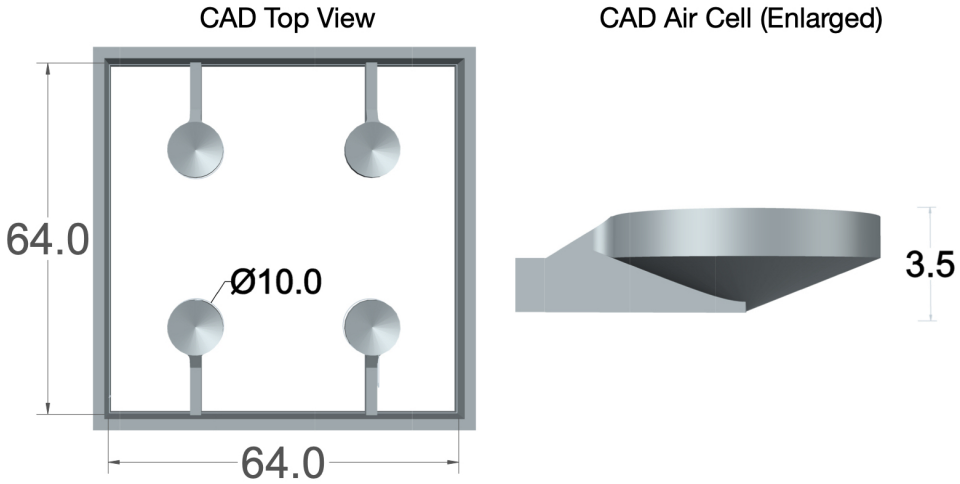


Figure 5.2: Air cell mould design 2, full view and englarged view of single air cell. Note the majority of the air cells surface area is the expanding face.

5.2 FEA and Testing

To determine the behaviour of different air cell shapes, Finite Element Analysis (FEA) is valuable. FEA uses the Finite Element Method (FEM); by discretising a body into a mesh (hundreds of simple polygons) and aggregating the stress and displacement of each polygon across the whole body, stress responses can be calculated for complex shapes that are impossible to solve algebraically. Autodesk Fusion was the FEA software used in this project.

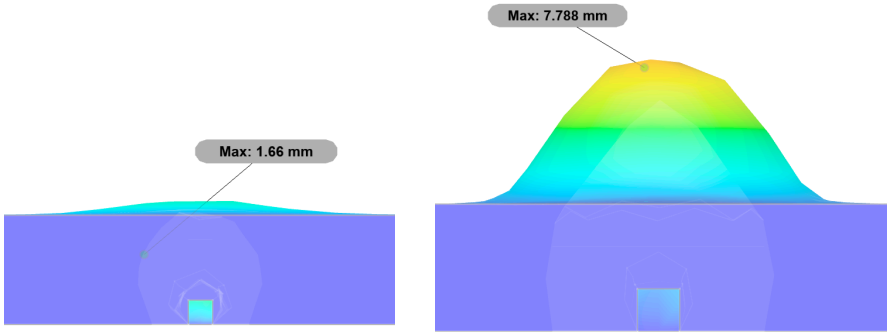


Figure 5.3: FEA for designs 1 and 2. Some simplifications are made in the FEA model but the general air cell shape is preserved. Design 1 shows little to no outward deformation whereas design 2 is more effective.

As seen in 5.3, design 2 demonstrates a much better stress response as maximum deformation occurs on the contact face. This means pressure is being directed into the user. For design 1, maximum deformation occurs on the internal face with limited external deformation. This indicates that a large pressure is needed to produce a tactile sensation, decreasing the safety factor of the design.

While FEA is useful to gain a basic understanding of how changing a certain aspect of a design effects its stress response, it has limitations. Firstly FEA software often only has 1 type of silicone which may have slightly different material properties to the silicone used to fabricate the pneumatic element. Secondly the CAD model is a simplification; fixed faces of the shape in the simulation may deform in reality. This means numerical values (displacement, safety factor etc.) generated by the FEA are unreliable. To fully understand how an air cell design will perform, it is necessary to fabricate a prototype. The prototypes for designs 1 and 2 corroborated the FEA - design 1 produced insufficient tactile sensation. It is clear that the larger the face of the air cell in contact with the user, the more noticeable the pneumatic haptic feedback is. For this reason design 2 is the clear choice.

With design 2 chosen, the 2 main factors affecting stress response are the thickness of the silicone on the expanding face and the shape of the air cell. As seen in FEA and physical testing, the part of the

Shape	Wall Thickness	s_1	s_{10}	s_{diff}
Flat	1mm	15	12	3
	2mm	12	11	1
Concave	1mm	12	10	2
	2mm	10	10	0

Table 5: Comparison of cell shapes and wall thicknesses.

air cell under greatest stress is the centre of the expanding face. By making the cell concave it increases the thickness of the silicone at the centre of the expanding face. This causes the edges of the expanding face to deform more than the centre, flattening the shape of the inflated cell 5.4. This increases contact with the user and reduces hysteresis. This is critical to the pneumatic element responding consistently to pressure increases after repeated use, ensuring accurate control over pneumatic feedback. To find the best combination of shape (flat or concave) and thickness (1mm or 2mm), all 4 permutations were tested. Each permutation was inflated with 20ml of air (high level of inflation) from low pressure 10 times and the maximum displacement of the cell recorded for the first and last cycle. The difference in displacement shows the amount of hysteresis (stretched silicone requires a larger volume of air to inflate).

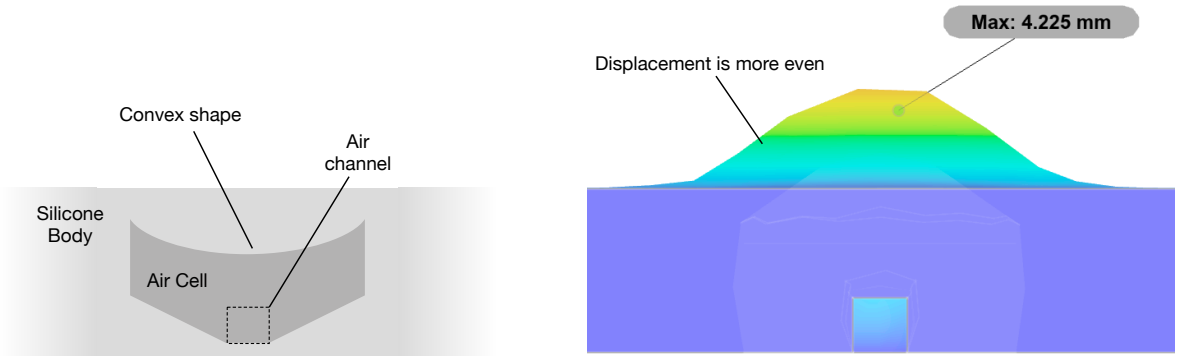


Figure 5.4: Convex air cell principle with FEA. Not numerically precise but general shape of displacement is reliable. Reducing the strain at the centre of the face by increasing the silicone thickness in a parabola creates flatter, more even displacement.

As shown by table 5, a 2mm thick expanding face with a concave air cell significantly outperforms the other configurations.

5.3 Fabrication

A key factor determining the stress response of the pneumatic unit is the hardness of the silicone. Different types of silicone rubber vary in hardness, measured using the Shore Hardness Scale. There

are multiple scales for different levels of hardness; those relevant for silicone are 00 (very soft) and A (slightly firmer). A higher number corresponds to a harder material. The harder the silicone, the less flexible the pneumatic element is. This means the air cells deform less under the same air pressure but the yield stress is higher. They are also less prone to permanent deformation and thus hysteresis. For this project 3 silicone types were considered- Ecoflex 00-10, Ecoflex 00-30 and Dragonskin NV (Shore hardness 10A). Through testing Ecoflex 00-30 was found to be the most suitable. It is more flexible than Dragonskin 10A (which deformed too little to create an effective sensation at safe pressure) but far less susceptible to hysteresis than Ecoflex 00-10 (which suffered from permanent deformation upon repeated use).

To cast the silicone a mould is required. First the mould is drawn digitally using CAD software (for this project AutoCAD) to create a stereolithography (STL) file. The STL file is sliced using specialised software to convert it into instructions for the 3D printer. The mould is 3D printed from Polylactide (PLA). PLAs low melting point, high strength and solid layer adhesion allows for highly precise designs. It also doesn't stick to silicone meaning once the silicone is set, the pneumatic element can be removed from the mould without sustaining damage.

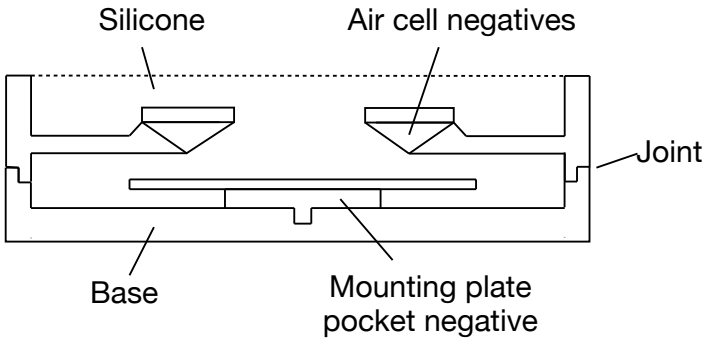
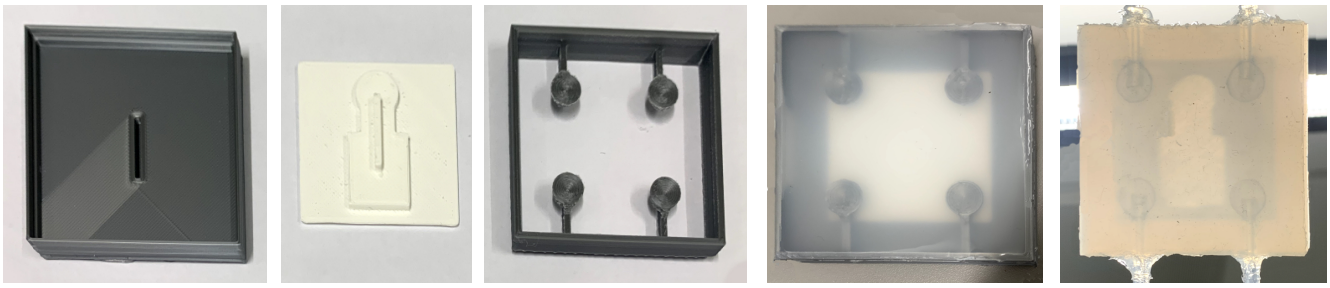


Figure 5.5: Cross section of 3 part silicone mould. The air cell shape and layout from design 2 is retained.

As explained in 4.4, the vibrotactile motor plate is embedded within the pneumatic unit via a pocket created by the silicone mould. The mould cannot be printed as a uniform piece of PLA due to the complexity of the shape. However, the whole pneumatic unit can still be cast in a single process by separating the mould into 3 interlocking components, printing each individually and assembling them. These components are a base layer, a raised plate that clips to the base layer to create the pocket for the vibrotactile plate, and a negative of the air cells which connects above the base layer. To prevent air bubbles from forming under the raised plate it is necessary to temporarily place the mould inside a vacuum chamber

once poured. This causes the bubbles to expand and rise to the surface of the silicone. The small extrusion on the plate slots into the groove on the base layer to hold it in place. To connect the base layer and the air cell negative, the grooved edge of each slot together to create an overlap (see 5.6) that prevents silicone from leaking as it sets.



(A) (B) (C) (D) (E)

Figure 5.6: (A) Base layer. (B) Motor plate negative. (C) Air cell negative. (D) Fully assembled mould with silicone. (E) End result. Notice the 4 air cells and pocket for the motor plate.

5.4 Hardware

To drive air into the air cell during inflation, an air pump is used. HaptiSuit requires the pump to be small but relatively powerful. All small air pumps function with a DC motor; the rotation of the motor drives a pump mechanism that displaces air. For small pumps, this is typically a diaphragm or piston. At the start of a pump cycle the inlet valve opens and air is sucked into the low pressure chamber. As the motor revolves to begin compressing the air, the inlet valve closes and the outlet valve opens. When the motor is stopped, the pressure in the air cell closes the outlet valve. This means the motor is only running during the inflation stage; once inflated, the motor can be stopped and the air cell remains pressurised. Reversing the polarity of the pump only causes the motor to run in reverse; the valves are still in the same place so the pump will continue inflating the pneumatic element. This means that a valve is necessary to remove air from a pressurised air cell (discussed later in this section).

The DC motor air pump used in HaptiSuit is the DFRobot 370 mini pump. The pump operates between 1.5V to 5V; this is ideal as the Teensy 4.1 (the microcontrollers used in HaptiSuit, see 6.4) runs off 5V DC power. This means both the pump and the microcontroller can run off the same power line without need for a voltage converter. The maximum pressure is 58kPa; this is within the pressure range tested in FEA simulations (50kPa) and is sufficient to cause the air cell to inflate. The maximum pump flow rate is 37ml/s at 5V without load; in practice the flow rate is slightly slower as the pump is working against a

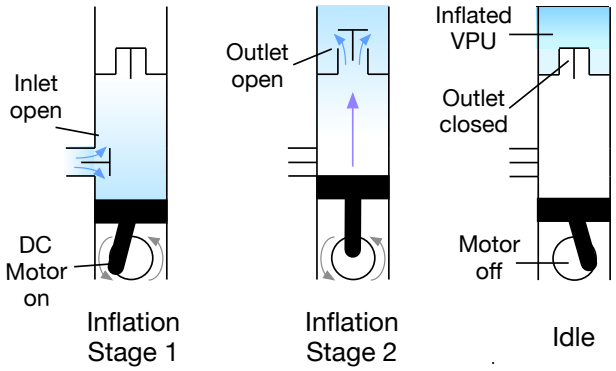


Figure 5.7: Diagram showing the internal mechanism of the DC motor air pump. Note the pressure from the inflated VPU closes the outlet valve meaning the air cell remains pressurised when the pump is idle.

pressure gradient. Through testing, each air cell was found to be able to hold 30ml of air before bursting. Assuming a safety factor of 1.5, pumping 20ml of air takes just over 0.5s.

The DC motor in the mini pump requires 500mA to run at full power. This is even higher on startup as the motor has to overcome inertia to begin rotating. The Teensy 4.1 pins cannot safely output more than 20mA meaning an external power source must supply the current needed to drive the motor. The microcontroller needs some means of controlling the current flow of the external power supply to the motor; the most effective is a transistor. A transistor acts as an electronically controlled switch [23]. The simplest type is a Bipolar Junction Transistor (BJT). It has 3 legs; base (B), collector (C) and emitter (E). Turning it on (sending current from base to emitter) causes a larger current to flow from the collector to the emitter. When it is turned off (no current from base to emitter), no current flows from collector to emitter. The ratio between the large and small current is the gain.

The other common transistor type is MOSFET. MOSFET is similar to BJT but controls the main current flow via voltage as opposed to current. For HaptiSuit, BJT is the more suitable choice as it easier to interface with the Teensy 4.1 microcontroller than MOSFET. Most MOSFETs require more than 5V to turn on meaning to control them with the 3.3V pins of the Teensy, a gate driver is needed. Even if a 3.3V MOSFET is used they are more expensive and require an extra resistor than BJT. This is significant as the technology in HaptiSuit is designed to be scaled up to control many VPUs.

$$\frac{V}{I} = R \qquad \frac{5 - 0.7}{0.010} = 430 \approx 470\Omega \tag{2}$$

To pull 500mA from the collector to the emitter with a gain range of 25-250, between 2mA (500÷250) and 20mA (500÷25) must be drawn from the Teensy to the base pin. HaptiSuit uses 10mA as it is high

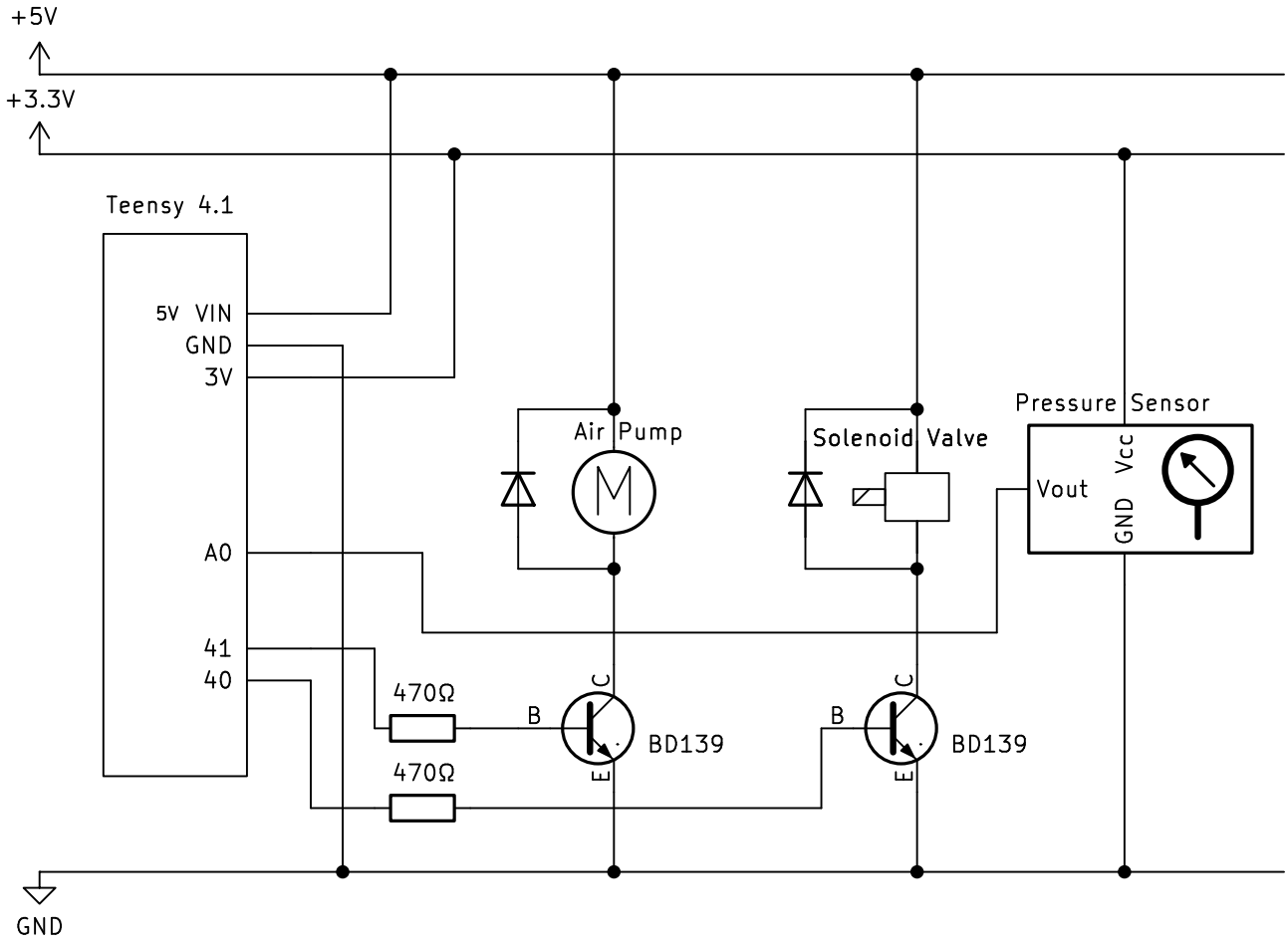


Figure 5.8: Circuit diagram of the pneumatic components for one VPU.

enough to drive the motors effectively but low enough that multiple motors can be driven simultaneously (many pins can output 10mA at the same time safely). As BJT transistors need around 0.7V to cause current to flow from base to emitter, this is subtracted from the 5V supply when calculating the resistor value. 430Ω is not a standard resistance so 470Ω resistors were used.

When the motor stops running, its inertia causes a current to be induced as it decelerates. A diode is needed to provide a path for current to keep flowing, preventing a reverse voltage spike which is dangerous to other components in the circuit.

As mentioned earlier it is necessary to use a valve to remove air from a pressurised air cell. As each VPU requires at least one valve, they must be lightweight and compact for the design to be scalable. A small 3 port 2-way solenoid valve is suitable. Solenoids are comprised of an coil surrounding a movable armature. When current is applied to the coil, a magnetic field is generated which pulls the spring loaded armature into the coil. When current stops the armature is reset to its original position by the coil. The armature in the valve switches between pump and atmosphere when at high or low current. By connecting the air cell to the common mode of valve and the pump output to the high mode while leaving the low mode open, the air cell can be inflated by driving the pump and pointing the valve at the air cell; it is deflated by shutting off the pump and pointing the air cell at the open mode (see figure 5.9).

The solenoid valves require 100mA to activate (larger than the Teensy can provide) so use the same BD139 transistors as the motors. The same 470Ω resistor is used. Like motors, solenoids also create a magnetic field when powered so also need a diode to stop a reverse voltage spike when turned off.

If the inflation of the pump is controlled by timing alone, there is a risk of overinflation. This could occur due to an error in the microcontrollers programming. To prevent the pneumatic element from exploding, it is necessary to know the pressure within the system. Pressure is measured using a barometer. HaptiSuit uses an analog barometer; for a pressure from 100 kPa to 250 kPa, the sensor generates a voltage ranging from 0 - 3.3V linearly proportional to the pressure reading. This value is read by the analog pins on the Teensy which are sensitive up to 3.3V using analog.Read(). analog.Read() returns a value between 0 and 1023 linearly for voltages 0V - 3.3V. The barometer is 3.3V DC; this is provided by the 3.3V VCC pin of the Teensy.

The initial design for the pneumatic configuration had one DC motor air pump inflating multiple VPUs. This was scrapped for code complexity (2 valves per unit and global pump variables in code) and slow response due to weak inflation. The final design uses a single pump, valve and barometer per pneumatic unit. This improves the code; all the variables and logic for each pneumatic unit is encapsulated within

the Actuator class (see 6.4). The response time is significantly faster as each pump inflates a smaller volume (1 VPU vs many). The pump connects to the high voltage (ON) port of the 3 port valve. The low voltage (OFF) port leads to the atmosphere. The common port connects to the VPU and the pressure sensor.

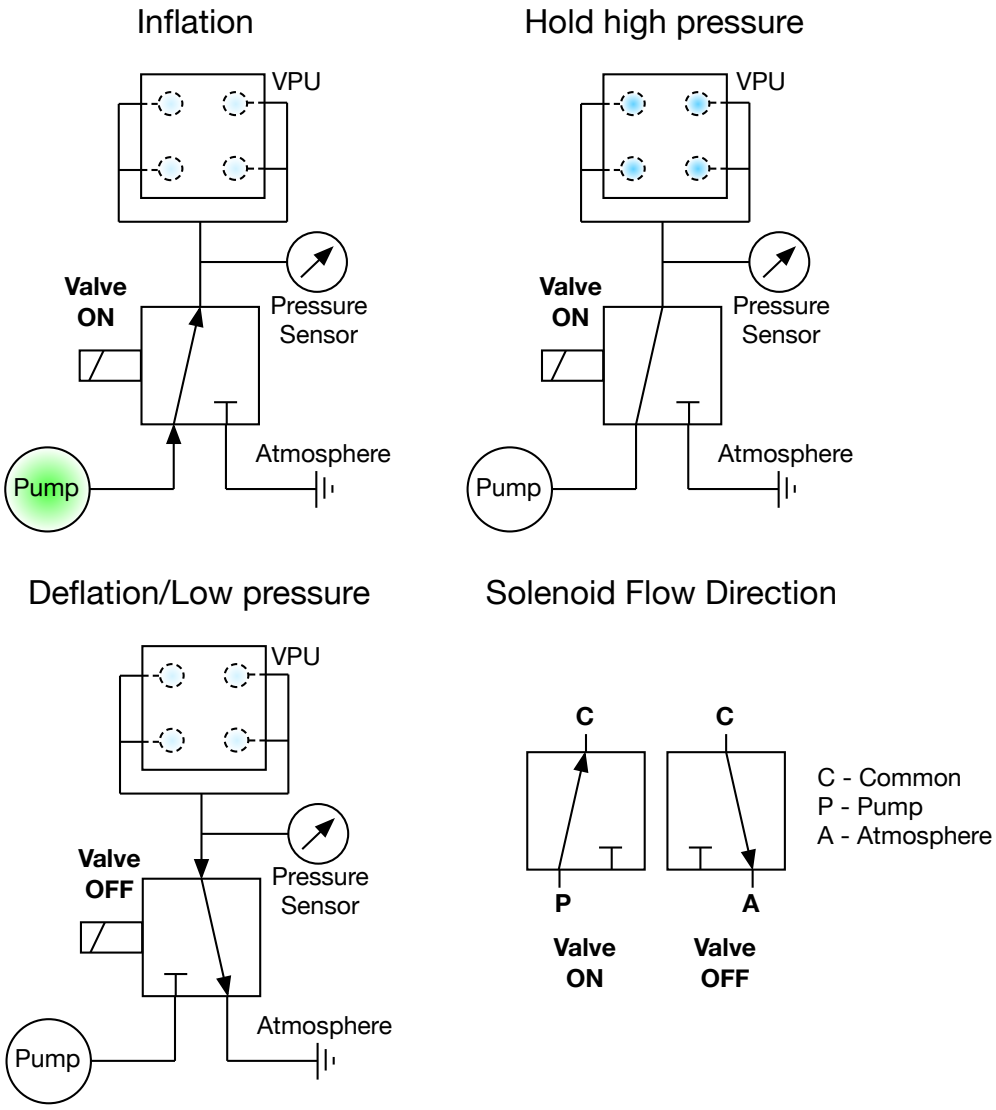


Figure 5.9: Air flow diagram for one VPU. The same arrangement is replicated for each VPU. Valve ON/OFF corresponds to high/low current. It is in OFF state by default.

Flexible (but non-expanding) pneumatic tubing of 2mm internal diameter and 4mm external diameter is used for airflow. 2mm internal diameter is standard for the pump, valve and pressure sensor. The 4mm external diameter is larger than the 2mm connection holes in silicone unit resulting in a tight fit. Silicone sealant is used to bind silicone to plastic tubing, ensuring the seal is airtight.

6 Communications and Logic

HaptiSuit separates each haptic feedback method into a distinct subsystem controlled by a Teensy 4.1 microcontroller. Both subsystems are controlled by a master Teensy. The master communicates with a GUI on a computer, allowing the VPUs to be controlled fully independently.

6.1 Master-Slave Configuration

HaptiSuit contains two independent subsystems. As it is critical the complete design can be easily scaled into a full suit with 20+ VPUs, controlling both vibrotactile and pneumatic subsystems from a single microcontroller is impractical. As the number of VPUs increases, it will soon exceed the number of available pins on the Teensy. Furthermore arduino boards are single core; they are not capable of parallel processing. With more VPUs, the microcontrollers processing speed will become a limiting factor on the suits ability to respond quickly to virtual stimuli. The solution is to use a master-slave configuration. Control of the haptic subsystems is delegated to 2 separate slave microcontrollers (one for pneumatic and one for vibrotactile) with a central master microcontroller coordinating the operation of the entire system. By distributing the tasks of updating the pneumatic state and vibrotactile state of the VPUs between 2 slave microcontrollers, it allows each subsystem to processed concurrently. This greatly improves the efficiency and scalability of the suit as the time taken for the master to send data to the slave and the slave to respond is negligible compared to the time saved by running each subsystem in parallel, especially with more VPUs. Splitting the 2 subsystems into separate programs simplifies the system architecture, making the code more readable. A master-slave configuration also promotes modularity; individual slave arduinos can be added, removed, or replaced without disrupting the entire system. This improves troubleshooting as faulty elements of the master-slave system can be quickly isolated.

For the master and for each slave, a Teensy 4.1 is used. Teensy 4.1 runs on a 5V power supply which is convenient as the motors and valves in the pneumatic circuit require 5V to operate, as do the TCA9548A and DRV2650 in the vibrotactile circuit . The 3.3V pin allows for a 3.3V power line which is needed to run the barometers on the pneumatic slave. The Teensy 4.1 is extremely compact, providing suitable processing power and a large number of pins despite its small size. In addition to its low cost this further improves the scalability of the system. It also has multiple I2C buses; this is crucial for the vibrotactile slave as it requires a bus to communicate with the master and another bus to communicate with the TCA9548A. The fast upload speed ($\sim$5s) compared with other small microcontrollers (eg. ESP8266

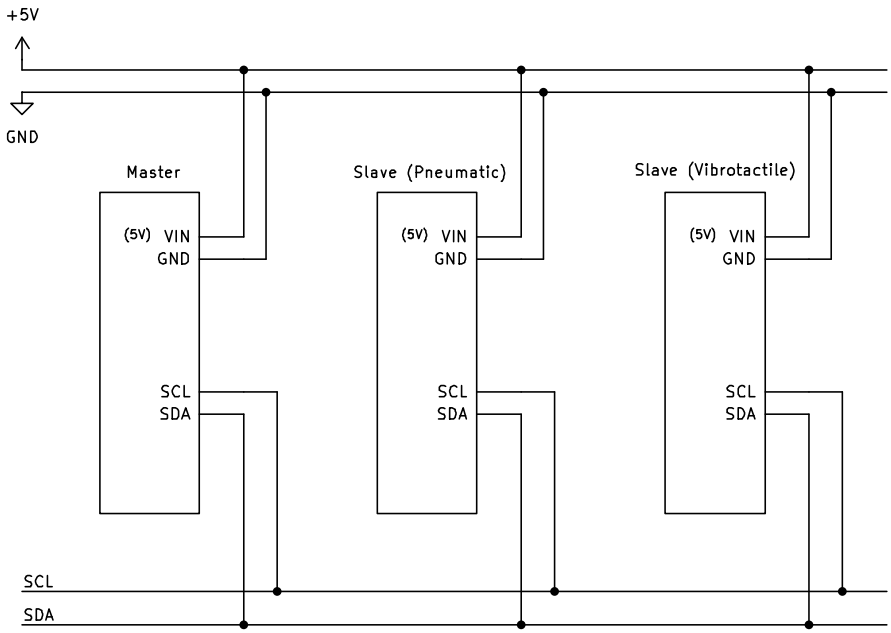


Figure 6.1: Master/Slave configuration of the 3 Teensys.

$\sim$20s) is also useful for debugging when implementing features.

6.2 Communications

As HaptiSuit makes use of multiple microcontrollers, an efficient method of passing digital data between them is essential. Because of the hierarchical master slave configuration, the most logical choice is the Inter-Integrated Circuit (I2C) protocol. I2C uses 2 lines to send and receive data; a serial clock line (SCL) and a serial data line (SDA) [24]. SCL is controlled by the master device; when the clock line changes from low to high a single bit of information is transferred on the SDA line.

The master initiates a START condition (pulls both SDA and SCL lines low) to begin I2C communication. It then sends 7 bits to specify the address of the I2C device on the bus to communicate with and a read/write bit to specify whether the master is sending or requesting data to/from a target device on the bus. After every byte, an acknowledge bit (ACK) is sent by the target device to inform the master that the data byte was received. The combination of a byte plus an ACK is known as a frame. If sending data, the master then sends one or more data frames consisting of 8 bits of data and an ACK to be received by a target device. If requesting data, the target device sends data frames for the master to read. Once all data bytes have been sent, communication is ended by the master using a STOP condition (letting the pull-up resistors pull both SDA and SCL lines high).

Pull-up resistors connect the SDA and SCL lines to VCC. This ensures that the lines are in a known

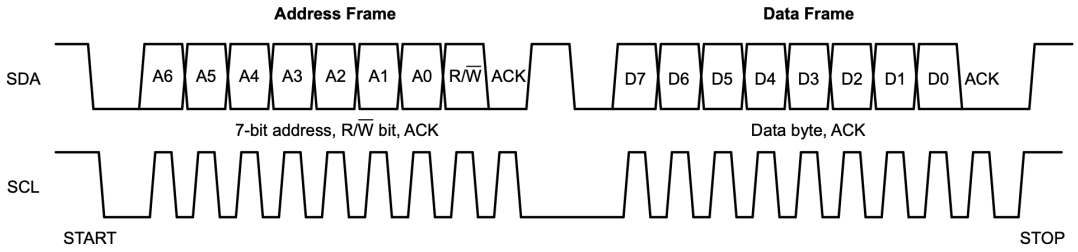


Figure 6.2: SDA and SCL lines. Image from [24].

state when idle (high) and slave devices know that a bus is connected. The Teensy 4.1 has built in pull-ups on the SDA and SCL pins but external pull-up resistors are useful for reducing potential noise from the breadboard.

The Wire library manages I2C communication between Arduino boards and other I2C devices. The master has 2 functions that use the wire library, i2cSendTargetState() and i2cGetCurrentState(). i2cSendTargetState() writes the pneumatic and vibrotactile target state arrays onto the I2C bus, each addressed to the corresponding slave so they only read their respective byte array. i2cGetCurrentState() requests the current state array from each slave. This tells the master if the slave is in the process of updating the actuator states or has completed the update (see paragraph after next).

The slaves also have 2 functions that use the wire library, getTargetState() and i2cArrayResponse(). getTargetState() reads bytes on the I2C bus when the master sends the target state array to the respective slave. This is specified at startup using Wire.onReceive(getTargetState). i2cArrayResponse() instructs the slaves on what to do when the master requests data from them. This is necessary as the I2C protocol gives the master full control over when data is sent on the bus. A slave cannot send data unless requested by the master, at which point the slave immediately responds to the request. This is possible using interrupts. When a slave device receives a request command from the master on the I2C bus, it stops wherever it is in the program and executes the special handler function (i2cArrayResponse) which is specified with Wire.onRequest(i2cArrayResponse) at startup. This function must be very fast to execute as it is interrupting the rest of the program.

```
void i2cGetCurrentState()
//get current state from each slave on I2C buses
{
  int i = 0;

  Wire.requestFrom(pneuSlaveAddr, 4); //request 4 bytes from the pneumatic slave

  while(Wire.available() > 0) //Wire.available is the number of bytes on the I2C bus
  {
    arrCurrentState_pneu[i] = Wire.read();
    i++;
  }

  //clear the buffer, shouldn't be necessary but I2C buffer gets scrambled otherwise
  while(Wire.available())
  {
    Wire.read();
  }
```

Listing 2: i2cGetCurrentState() master request from the pneumatic slave. The function also has corresponding code to request from the vibrotactile slave.

The master has no way of knowing when the slave devices have completed the update of the VPU states as the slaves cannot send data until requested. Thus the master must poll the slaves by periodically requesting the current state values from them. This causes problems if the slave is polled whilst it is updating its version of arrCurrentState. This is solved with a buffer; the slave device sends the array arrI2CBuffer when polled by the master. arrI2CBuffer has the same length as the other 2 arrays. Initially all elements in the buffer are set to 255 to indicate the slave is busy updating the state; this is what the master reads from the slave when it begins polling. The buffer is only updated to the current state of the slave once the slave has successfully updated all actuator states to their target state. This is done by copying all elements of arrCurrentState into arrI2CBuffer. At this point the master polls the slave and reads its current state. This updates arrCurrentState so it is equal to arrTargetState and the master stops polling, returning the complete system state to the GUI and awaiting a new target state.

The basis of HaptiSuits data structure is 4 byte arrays: arrTargetStatePneu, arrTargetStateVibr, arrCurrentStatePneu and arrCurrentStateVibr. Each target state array stores the state that the respective subsystem (pneumatic and vibrotactile) should be in; each current state array stores the last known state of the respective subsystem. Each byte array is initialised to an array of length equal to the number of VPUs with all elements set to 0 (all pneumatic and vibrotactile actuators in the off state).

The master controller holds all 4 arrays. Firstly the master gets the full system target state (array) from GUI (serial). Once arrTargetStatePneu and arrTargetStateVibr are set on the master (by reading the first half and second half of the serial bus and assigning to the corresponding array), each array is sent to the pneumatic and vibrotactile slaves respectively via I2C. The master than requests the current state from

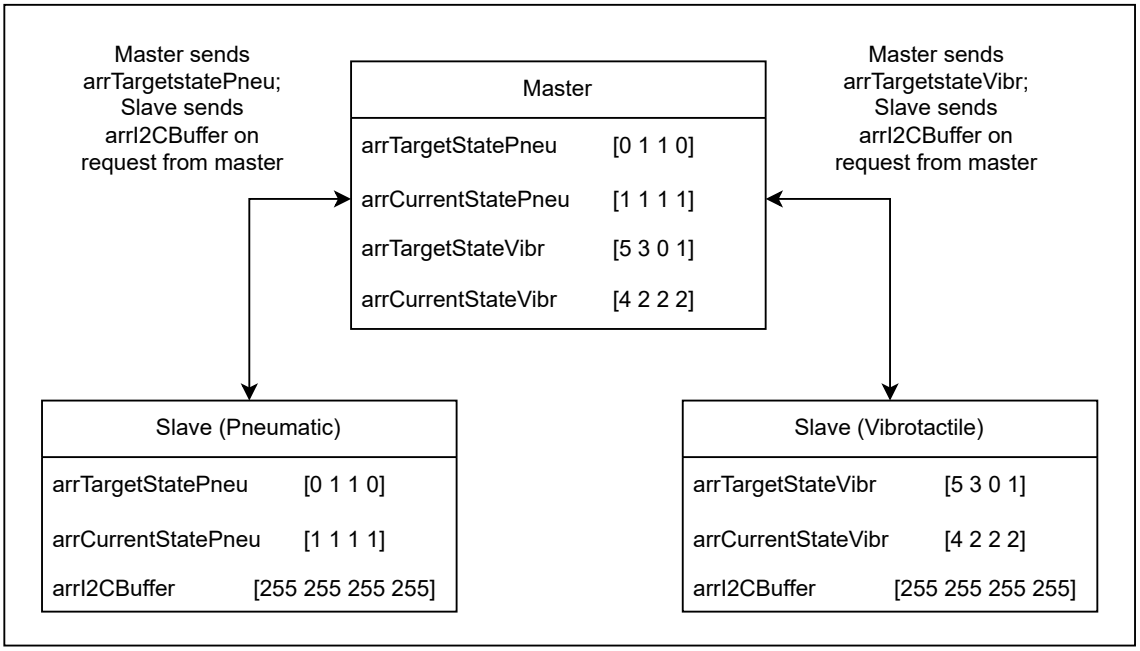


Figure 6.3: Array structure for the master and each slave. In this example the number of actuators is 4; example values are used for individual states.

each slave again via I2C. However before each time the master polls the slaves to check the update is complete, the master waits briefly to give the slave time to update. Initially the slaves responds with a full 255 array indicating they are updating. Polling continues until each slave has sent its arrCurrentState and both arrCurrentStatePneu = arrTargetStatePneu and arrCurrentStateVibr = arrTargetStateVibr. At this point the system state is successfully updated and the completed state is sent to the GUI. The master waits for a new target state from the GUI.

The slave controllers only hold the 2 arrays containing the current and target state of the respective subsystem. As each subsystem is controlled independently, the slaves only need data on the subsystem they control as well as a third array arrI2CBuffer. Each slave begins by waiting for the master to send a target state via I2C. Once the target state is received, every element in the buffer is set to 255. Each slave then begins updating the states of the actuators until its arrCurrentState = arrTargetState. At this point the buffer is updated to the current state ready to be read by the master via I2C upon request. Each slave then waits for a new target state from the master.

6.3 State Machine Logic

HaptiSuit has 2 independent haptic subsystems. Each subsystem controls multiple actuators independently. Like most affordable Arduino microcontrollers, the Teensy 4.1 is single core. This means it

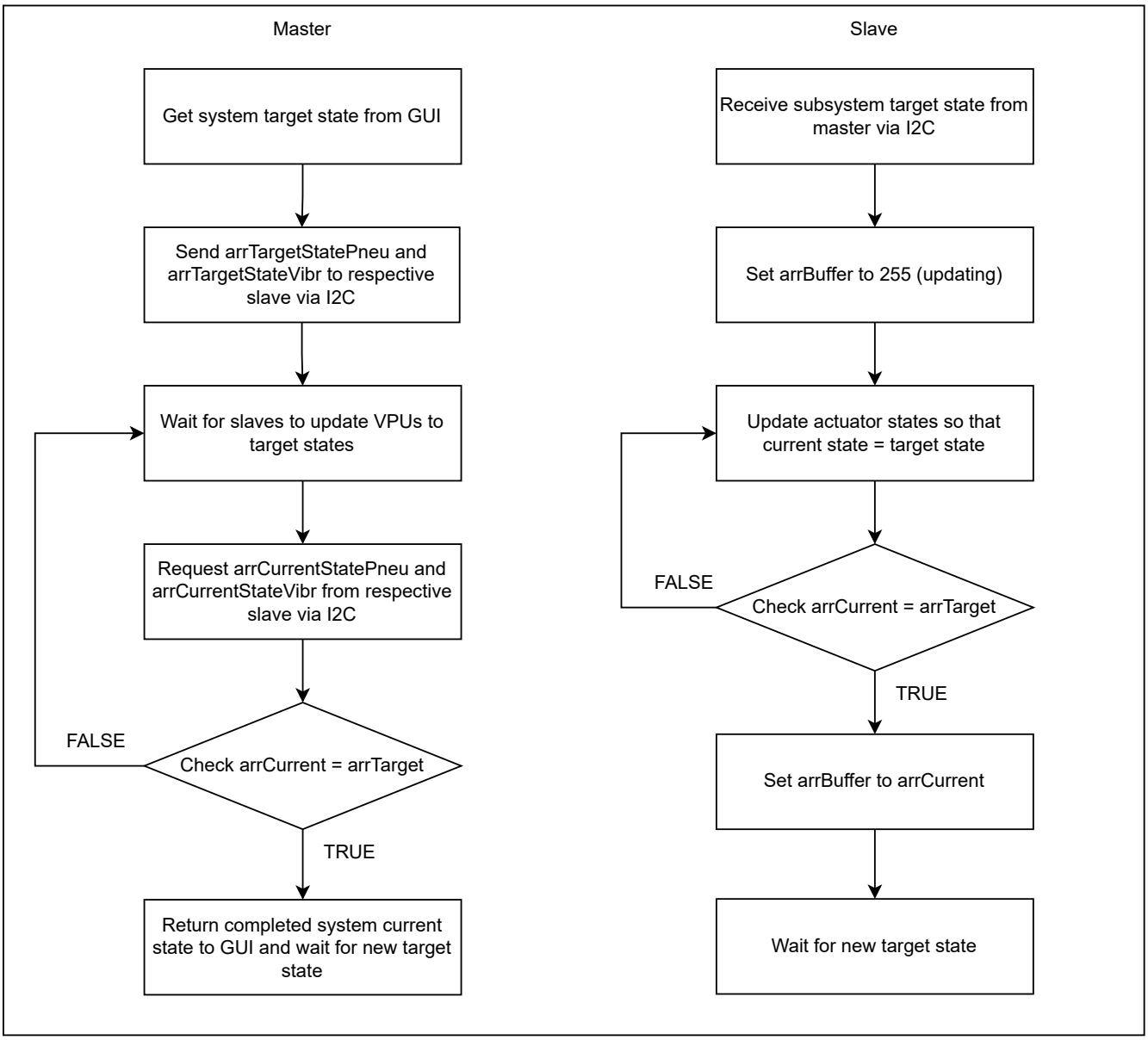


Figure 6.4: Process diagram for master and slave communications. After receiving a new target state, slaves can be interrupted at any point by the master requesting arrI2CBuffer.

executes instructions consecutively; it cannot perform more than one action at the same time. To be able to respond quickly to inputs from a virtual environment, each slave must update the states of multiple actuators simultaneously. Initially this is problematic as due to the lack of multithreading it seems it is only possible to update each actuator in turn. For the vibrotactile actuators this is sufficient; they are very small so have little inertia thus the haptic drivers change their state in $\sim$10ms. However this doesn't work for the pneumatic actuators. As the pneumatic cell is inflated by moving air into the cell using an air pump and deflated by releasing it through a valve, it takes $\sim$50ms to change state. This means that with even 2 VPUs the system cannot update them fast enough to provide convincing feedback. The solution is to use a state machine. A state machine is a model to describe a system [25]. It consists of a number of states and transitions between these states. Transitions are associated with actions triggered by states.

Quantising a task into steps and quickly looping between the same step on different tasks gives the illusion of the tasks happening in parallel despite the Arduino executing each step linearly. This is implemented with object oriented programming (OOP) by using a class called Actuator for both the vibrotactile and pneumatic slave with member functions and update logic specific to each subsystem. Using OOP principles allows the code to be scaled quickly; once the logic for the Actuator class is written, adding more VPUs is as simple as creating another Actuator object. All update logic is encapsulated within the actuator class member function Actuator.Update(). This member function within each actuator class determines what actions should be taken in a given current state after receiving a given target state. The slave quickly loops through each actuator calling the respective Update() function then checks arrCurrentState against arrTargetState. If they are not equal then the update loop runs again, with another check afterwards. Once each VPU has been incrementally updated and arrCurrentState = arrTargetState, the slave exits the loop and returns the complete current state to the master.

To implement the state machine for the pneumatic subsystem, all logic is encapsulated in the class Actuator. This has several member variables; notably the VPU index, pins for each of the pump/-valve/barometer, and variables to track pressure and number of pump cycles. These are initialise in the constructor to ensure they cannot be accessed outside of the Actuator class. The member function "Update()" contains 5 if statements to determine the which of the corresponding 5 states the pneumatic actuator is in and the action needed (see diagram 6.5). The pneumatic subsystem uses open loop control to inflate and deflate the air cells; Running the pump for timed cycles consistently pumps the same amount of air into the VPU (initially a closed loop system with feedback from the pressure sensor was tested but it was found to be extremely unreliable due to inaccurate readings when sampling the pressure

sensor). The air cells are inflated in cycles, running the pump for 40ms and checking the state afterwards (effectively PWM). Pressure is checked as a safety measure to prevent overinflation.

When update is called for an actuator, the first check is if the target state is 1 and the pressure has exceeded the maximum threshold. This is an emergency check to prevent the air cell from exploding. If true, the pump is switched off and the valve set to hold to stop inflating the air cell and retain pressure. The current state is set equal to 1, the same as the target state. Volume cycles are also set to the max to prevent further inflation.

```
1   if ((arrTargetState[index] == 1) && (sensorVal >= pressureHigh))
2   {
3     arrCurrentState[index] = 1;
4     //we have EXCEEDED pressure so switch OFF the pump, and HOLD the valve
5     digitalWrite(pumpPin, LOW);
6     digitalWrite(valvePin, HIGH);
7
8     //assumes that it has completed all volume cycles
9     //stops the (volumeCycles < volCycleMax) conditional statement activating
10    volumeCycles = volCycleMax;
11  }
```

Listing 3: Check for exceeding maximum pressure.

The next check is if the target state is 1 and the number of volume cycles has exceeded the maximum limit. This also means the cell is pressurised so the pump is switched off and the valve set to hold. Again current state is set equal to 1, the same as the target state.

```
1   if ((arrTargetState[index] == 1) && (volumeCycles >= volCycleMax))
2   {
3     arrCurrentState[index] = 1;
4     //we have reached volume so switch OFF the pump, and HOLD the valve
5     digitalWrite(pumpPin, LOW);
6     digitalWrite(valvePin, HIGH);
7   }
```

Listing 4: Check for exceeding maximum number of volume cycles.

Next check is if target state is 0 and the pressure has reached the minimum value. This means the cell has completed deflation and is in the low pressure state so current state is set to 0, the same as the target state. The pump is turned off, the valve set to hold and the volume cycles counter is reset to 0. The cell is ready to inflate again.

```
1     if ((arrTargetState[index] == 0) && (sensorVal <= pressureLow))
2     {
3       arrCurrentState[index] = 0;
4       //we have released pressure so switch OFF the pump, and RELEASE the valve, and reset
      volumeCycles
5       digitalWrite(pumpPin, LOW);
6       digitalWrite(valvePin, LOW);
7       volumeCycles = 0;
8     }
```

Listing 5: Check for reaching minimum pressure.

If those 3 conditions are not met then the cell not yet in its target state; is must be inflating or deflating. If target state is 1 and the number of volume cycles is less than the maximum, the cell has not reached pressure yet. Current state is set to 0, the pump is turned on and the valve set to hold. The volume cycles counter is incremented by 1.

```
1     if ((arrTargetState[index] == 1) && (volumeCycles < volCycleMax))
2     {
3       arrCurrentState[index] = 0;
4       //we have NOT reached volume so switch ON the pump, and HOLD the valve
5       volumeCycles += 1;
6       digitalWrite(pumpPin, HIGH);
7       digitalWrite(valvePin, HIGH);
8     }
```

Listing 6: Check for inflation.

The last possible state is deflation. This is true if the target state is 0 and the pressure in the cell is above the minimum value. As pressure has not been fully released, current state is set to 1, the pump is switched off and the valve is released.

```
1     if ((arrTargetState[index] == 0) && (sensorVal > pressureLow))
2     {
3       arrCurrentState[index] = 1;
4       //we have NOT released pressure so switch OFF the pump, and RELEASE the valve
5       digitalWrite(pumpPin, LOW);
6       digitalWrite(valvePin, LOW);
7     }
```

Implementing the state machine for the vibrotactile subsystem is far simpler. The same OOP principles of encapsulation are used and the structure of the Actuator class is the same with different member variables (just actuator index) and different logic in Update(). The DRV2605 is able to change the state of an ERM from any state to any other effectively instantly with the drv.setMode and drv.setRealTimeValue functions; unlike the pneumatic subsystem there is no transition state. The vibrotactile actuator class needs only the index of the VPU to address the correct port on the TCA9548A. Updating the vibrotactile states is as simple as looping through each TCA9548A port, applying the target state to the vibration

strength and setting the current state equal to the target state of that actuator. This is done in a single loop.

```
void Update()
{
    //Check current versus target and take action
    arrCurrentState[index] = arrTargetState[index];
    tcaselect(index);    //function to select TCA9548A port

    drv.setMode(DRV2605_MODE_REALTIME);
    drv.setRealtimeValue(arrTargetState[index]*25); //scales the input from 0-5 to 0-125 (
    setRealTimeValue sets vibration strength linearly from 0-127)
    drv.go();
}
```

Listing 7: Update logic for vibrotactile actuator class.

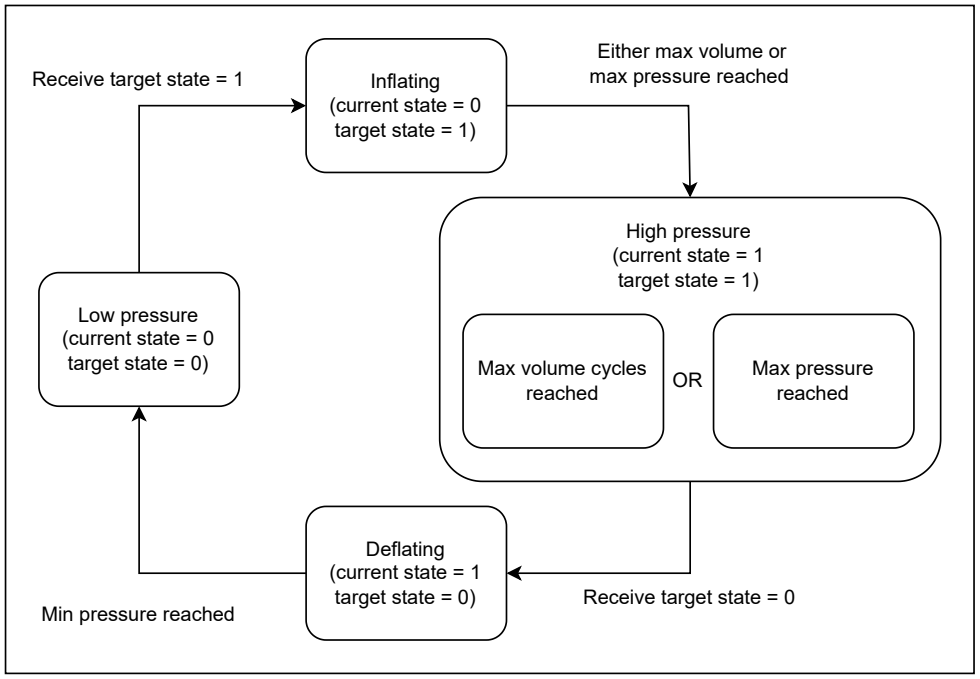


Figure 6.5: State machine diagram for pneumatic actuator logic. If the existing current state is already equal to the new target state, the actuator does nothing.

```
while (bStateCheck == false)
{
  act_0.Update();  //call update member function for each actuator
  act_1.Update();
  act_2.Update();
  act_3.Update();
  delay(40);       //give actuators time to update (necessary for pneumatic)

  bStateCheck = true;      //will exit update loop if array check is successful
  for (int i = 0; i < 4; i++)
  {
    //check each element in current against corresponding element in target
    if (arrCurrentState[i] != arrTargetState[i])
    {
      bStateCheck = false;     //array check failed so update again
    }
  }
}
```

Listing 8: Main loop for updating the pneumatic and vibrotactile actuators. This code is identical for both slaves.

6.4 GUI

Ordinarily the VPU activation states of HaptiSuit would be computed according to the users movement within VR. This is beyond the scope of this project so to demonstrate HaptiSuits ability to control each VPU independently and update their states promptly, a simple PC-based GUI is used.

As all logic checks occur within the Teensy microcontrollers, the GUI only needs to store one array of bytes - arrTargetState. The length of arrTargetState is equal to the number of VPUs x 2. For 4 VPUS, the first 4 elements correspond to the activation state of the pneumatic element of the VPU (either 0 or 1). The next 4 elements correspond to the activation state of the vibrotactile component of each VPU (integer values from 0 to 5). Storing both pneumatic and vibrotactile states in the same array simplifies the code when passing the array to the master as the Processing Serial library can only send a singe byte at a time.

The array is passed with Universal Asynchronous Receiver-Transmitter (UART), a serial protocol that Arduino boards use to communicate with computer via USB and vice versa. In Processing (the language the GUI is written in), this is done with the Serial library. The Serial library reads and writes data to and from external devices one byte at a time.

To create the GUI and its controls used to send activation states, the ControlP5 library is used. Each pneumatic element is controlled by a binary toggle switch. This sets the corresponding target state array element to either 0 or 1 (0 for depressurised and 1 for pressurised). The vibrotactile motors are con-

trolled using sliders. This assigns a value from 0-5 to the corresponding target state array element (0 for off vibration linearly increasing in strength from 1-5). Once the full haptic state of the suit is inputted, it is sent to the master Teensy using the Go button. Go sends arrTargetState by looping through the array and writing each byte element onto the serial bus consecutively for the master to read. The Reset button sends a target state array of all 0s and sets the toggle and slider values to 0.

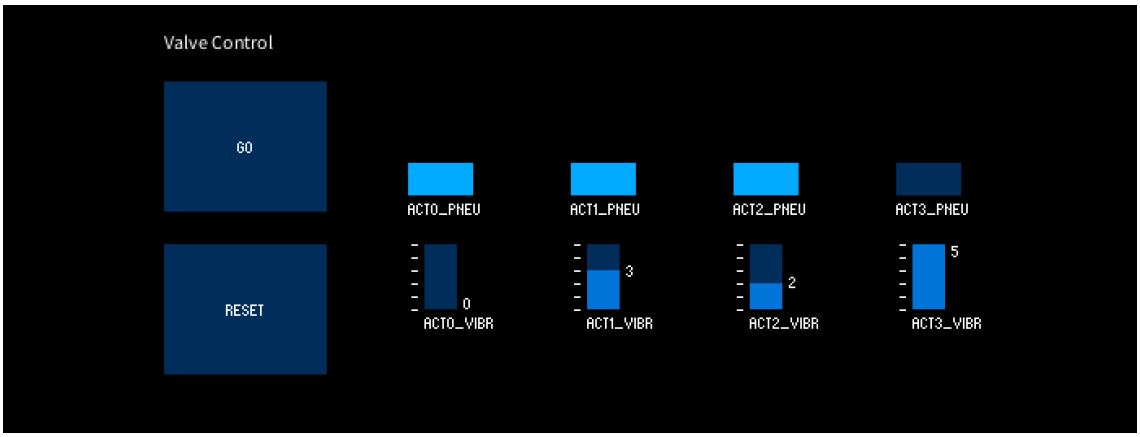


Figure 6.6: GUI for setting VPU states. Each VPU can be set independently to any new state from any other previous state.

7 Implementation and Testing

7.1 Suit Construction

HaptiSuits VPUs are designed to be used in a full body suit. For this project, a neoprene chest brace is used to demonstrate the effectiveness of the VPUs on the left chest region. The brace provides a large surface area; the VPUs are secured on the inward face of the fabric with pneumatic tubing and electrical wires attaching on the outside face. The chest brace is tight, ensuring the VPUs remain in full contact with the user at all times to deliver pneumatic and vibrotactile feedback most efficiently. Crucially the flexible neoprene material means it doesn't restrict the users movement; when used with a VR headset this makes the virtual environment feel more immersive.

As stated in previously, the chest has a low density of skin mechanoreceptors compared to other regions of the body [3]. This give the chest a spatial acuity of around 35mm. The 4 actuators are placed such that the distance between the individual air cells adjacent VPUs is no more than 40mm. This provides optimal haptic coverage of the chest area with high resolution.

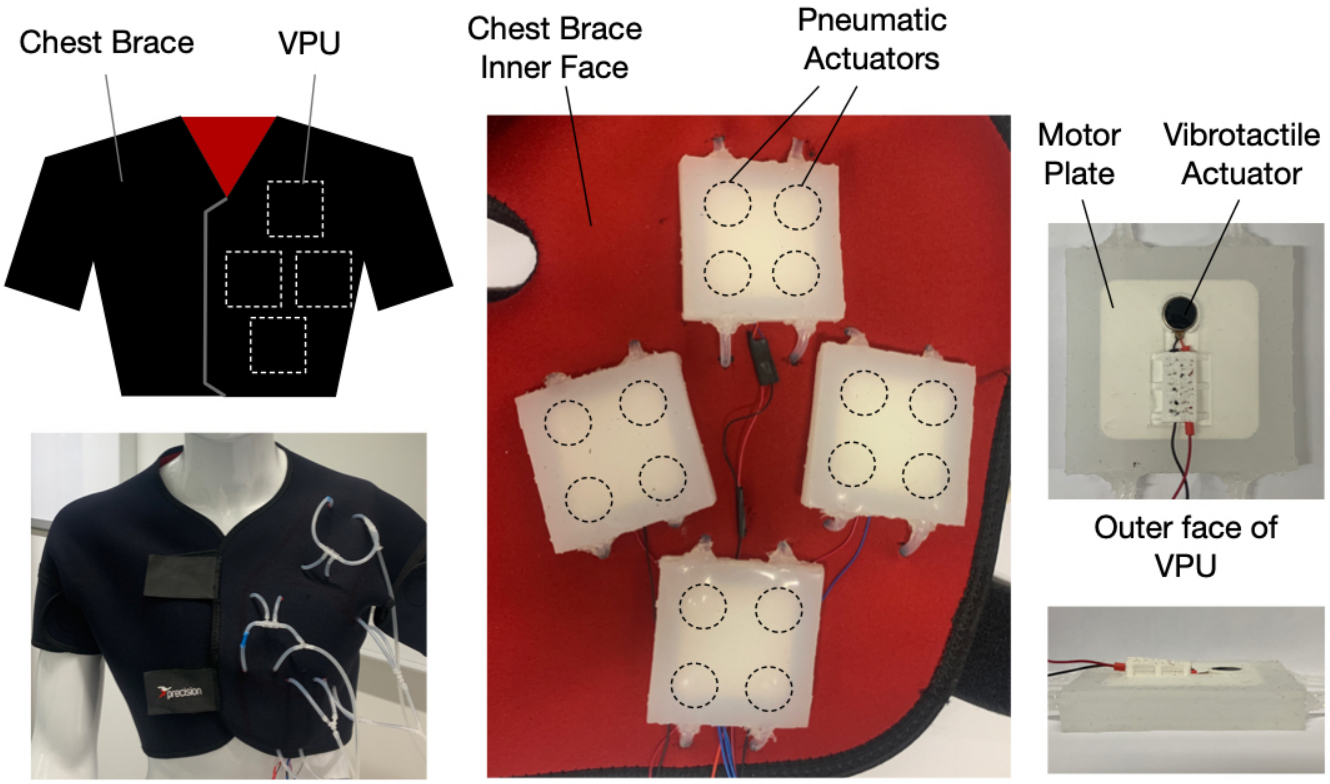


Figure 7.1: Arrangement of VPUs in the left chest region with both inward facing, outward facing and side view of VPU.

7.2 Results

Overall the design of HaptiSuit shows positive results. The master slave configuration is effective at handling the communication of the state arrays and updating each VPU simultaneously using concurrent state machines. The total response time from when a target state is sent by the GUI to the master sending the completed current state is $\sim$50ms for the pneumatic subsystem and <10ms for the vibrotactile subsystem. The pneumatic elements outperform the literature in response time due to smaller air cells. The vibrotactile and pneumatic actuators of each VPU are fully independently controllable, able to produce any combination of pneumatic feedback (inflated or deflated) and vibrotactile feedback (6 levels increasing linearly from 0-100% vibration).

The vibrotactile feedback is effective although there are occasional inconsistencies with the DRV2605 drivers relating to I2C; these are fixed by restarting the vibrotactile slave. The ERM motors produce noticeable vibration; the embedded mount plate transmits vibration effectively through the flat silicone unit. The TCA9548A multiplexer circuit is able to respond instantly to new state updates.

The open loop control of the pneumatic subsystem is consistent at delivering the same pressure with each inflation. Between each VPU there is minor deviation in the thickness of the expanding face due to silicone leakage whilst setting. The rigid motor plate increases the effectiveness of the pneumatic actuators by preventing them from expanding outwards. The strength of the compressive sensation is dependent on the tightness of the suit fabric. Using a specialised suit with tighter, less flexible fabric will improve the pneumatic feedback but this trades off the ability to move freely in the suit. The only major issue is the current prototype is unable to inflate more than 3 motors at the same time. This is due to hardware limitations; the thickness of wire from the external power supply and its connection to the breadboard are insufficient to deliver enough current to drive 4 motors. This is fixable with larger cables and a soldered connection to a PCB. Cable management is also needed to improve robustness.

An in-depth study of the effectiveness of HaptiSuits feedback using multiple participants was not possible due to the limited time frame of the project; more thorough testing is needed.

Figure 7.2: Demonstration of VPU in deflated (left) and inflated (right) states. Inflated air cells are lightly outlined for clarity.

8 Conclusion

HaptiSuit aimed to demonstrate a novel solution to providing full body multi-mode haptic feedback in real time by combining vibrotactile and pneumatic feedback in a modular, scalable design.

Different cutaneous haptic technologies were reviewed alongside existing solutions to determine that vibrotactile and pneumatic feedback is the most suitable choice for HaptiSuit.

This was done by designing the Vibro-Pneumatic Unit (VPU), a compact modular unit consisting of a vibration motor and inflatable air cells, and an efficient, scalable master-slave system architecture that allows for total control of each subsystem independently across multiple VPUs. A prototype was manufactured To demonstrate the effeciveness of HaptiSuits design. The full design and fabrication process was documented and all design choices were justified through research and testing.

HaptiSuit provides a solid basis for combining high frequency vibrotactile and low frequency pneumatic haptic feedback into a full body suit but some areas could be improved. Firstly, optimising the air cell design to maximise the pressurised surface area in contact with the user will increase the effectiveness of both the pneumatic and vibrotactile feedback while the air cell is inflated.

HaptiSuit was a proof of concept to demonstrate the effectiveness of the VPU design and control system. The hardware of the control circuit is arranged on a breadboard which is insufficient for driving high current which controlling multiple DC motors requires. To turn HaptiSuit into a viable product, the full control unit must be contained in a compact case. This means fabricating a custom PCB and designing a compact arrangement for the valve/pump/barometer pneumatic system. To prevent voltage drops when multiple pumps are running, lower resistance wires are needed to connect the power supply to the circuit.

To further increase the range of sensations, a third form of haptic feedback could be integrated into the VPU and master slave configuration. The most logical choice is either electrical or thermal. This would involve redesigning the VPU to contain a surface electrode or Peltier module respectively.

To use HaptiSuit with a VR headset, a motion tracking subsystem is needed to send proprioceptive information to the VR computer to then instruct HaptiSuit of interactions with the virtual environment. This would involve redesigning the software architecture to ensure motion data is processed quickly and reliably. Additionally an API design would permit integration with existing VR hardware.

References

[1] Liang He et al. “Robotic simulators for tissue examination training with multimodal sensory feedback”. In: *IEEE Reviews in Biomedical Engineering* 16 (2022), pp. 514–529.

[2] Dale Purves et al. “Mechanoreceptors Specialized to Receive Tactile Information”. In: *Neuroscience. 2nd edition*. Sinauer Associates, 2001.

[3] Giulia Corniani and Hannes P Saal. “Tactile innervation densities across the whole body”. In: *Journal of Neurophysiology* 124.4 (2020), pp. 1229–1240.

[4] *Precision Microdrives Technical Resources - Coin Vibration Motors*. 2021. URL: https://www.precisionmicrodrives.com/coin-vibration-motors.

[5] *Texas Instruments Actuator Technology*. 2016. URL: https://www.ti.com/video/series/actuator-technology.html#.

[6] *Precision Microdrives Technical Resources - Linear Resonant Actuators*. 2021. URL: https://www.precisionmicrodrives.com/linear-resonant-actuators-lras.

[7] *Precision Microdrives - Vibration Motors*. 2021. URL: https://www.precisionmicrodrives.com/motors/vibration-motors.

[8] Daeseok Kang, Chang-Gyu Lee, and Ohung Kwon. “Pneumatic and acoustic suit: multimodal haptic suit for enhanced virtual reality simulation”. In: *Virtual Reality* 27.3 (2023), pp. 1647–1669.

[9] Liang He et al. “PneuHaptic: delivering haptic cues with a pneumatic armband”. In: *Proceedings of the 2015 ACM international symposium on wearable computers*. 2015, pp. 47–48.

[10] Weicheng Wu and Heather Culbertson. “Wearable haptic pneumatic device for creating the illusion of lateral motion on the arm”. In: *2019 IEEE World Haptics Conference (WHC)*. IEEE. 2019, pp. 193–198.

[11] Elisa Galofaro et al. “Rendering immersive haptic force feedback via neuromuscular electrical stimulation”. In: *Sensors* 22.14 (2022), p. 5069.

[12] Max Pfeiffer et al. “Let me grab this: a comparison of EMS and vibration for haptic feedback in free-hand interaction”. In: *Proceedings of the 5th augmented human international conference*. 2014, pp. 1–8.

[13] Max Pfeiffer et al. “Cruise control for pedestrians: Controlling walking direction using electrical muscle stimulation”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 2505–2514.

[14] Mark Johnson. “Transcutaneous electrical nerve stimulation: mechanisms, clinical application and evidence”. In: *Reviews in pain* 1.1 (2007), pp. 7–11.

[15] *Teslasuit*. 2023. URL: https://teslasuit.io.

[16] Jinwoo Lee et al. “Stretchable skin-like cooling/heating device for reconstruction of artificial thermal sensation in virtual reality”. In: *Advanced Functional Materials* 30.29 (2020), p. 1909171.

[17] Bowen Zhang and Misha Sra. “Pneumod: A modular haptic device with localized pressure and thermal feedback”. In: *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. 2021, pp. 1–7.

[18] I. Terasaki. “Thermal Conductivity and Thermoelectric Power of Semiconductors”. In: *Reference Module in Materials Science and Materials Engineering*. Elsevier, 2016.

[19] Sidhant Gupta et al. “Airwave: Non-contact haptic feedback using air vortex rings”. In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. 2013, pp. 419–428.

[20] Ismo Rakkolainen et al. “A survey of mid-air ultrasound haptics and its applications”. In: *IEEE Transactions on Haptics* 14.1 (2020), pp. 2–19.

[21] *bHaptics TactiSuit*. 2024. URL: https://www.bhaptics.com/tactsuit/tactsuit-x40.

[22] Robert W Lindeman et al. “Towards full-body haptic feedback: the design and deployment of a spatialized vibrotactile feedback system”. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. 2004, pp. 146–149.

[23] *buildelectroniccircuits - How Transistors Work*. 2021. URL: https://www.build-electronic-circuits.com/how-transistors-work/.

[24] *A Basic Guide to I2C*. 2021. URL: https://www.ti.com/lit/an/sbaa565/sbaa565.pdf?ts=1715677610386&ref_url=https%253A%252F%252Fwww.google.com%252F.

[25] *Digikey - What is a State Machine?* 2023. URL: https://www.digikey.co.uk/en/maker/tutorials/2023/what-is-a-state-machine.