Mechatronics, Advanced Course
MF2058

Spring Term Report

Authors:
David Bergling
Neo Gumbel
Karl Voigt
Tong Zhao
Filip Strand
Martin Wahlund
Omed Khirzia
Omar Kanbour
Jesper Breander

Coach:
José Manuel Gaspar Sánchez

Stakeholder:
Jakob Kuttenkeuler

May 22, 2023

Contents

List of Figures

Acronyms

DDS	Data Distribution Service
GRF	Ground Reaction Forces
HMD	Head-Mounted Device
PSS	Point of Subjective Simultaneity
ROS	Robot Operating System
SOTA	State of the Art
UDP	User Datagram Protocol
VE	Virtual Environment
VR	Virtual Reality

1 Introduction

This section gives an introduction to the project with its background, scope, requirements and organisation.

1.1 Background

Foil surfing is a relatively new water sport. It uses a submerged hydrofoil and electric motor that drives the board, which allows the rider to glide over the water. The learning curve for mastering this watercraft depends a lot on the rider. While people with prior surfing experience generally require less time to learn how to maneuver it, others with less experience often have a steeper learning curve. This is complicated further by the fact that it is very expensive to buy or rent an eFoil board.

With a realistic simulator, the training time for eFoiling in water could be reduced. It also has the advantage of not needing to take breaks to charge the board, which could help newcomers practice more continuously. This projects aims to build a simulator for an eFoil board, as an easy way to learn foil surfing.

1.2 Stakeholder Presentation and Project Context

This project is a part of the Mechatronics ”Higher Course”, which is a course held at KTH Royal Institute of Technology in Stockholm, Sweden. The course is taken by all master students of the Mechatronics program, and stretches for over half a year.

This project has been requested by, and is performed under the guidance of, Jakob Kuttenkeuler, professor at KTH Centre of Naval architecture. He was one of the pioneers in developing the first eFoils, and has the vision of creating a physical simulator that can mimic the behaviour of a real eFoil. When this report references the ”stakeholder”, this is in reference to Prof. Kuttenkeuler.

1.3 Scope

The scope of this project is to develop a full-scale eFoil simulator, focusing on the mechatronic features that are needed. The requirements of the project are decided together with the project stakeholder.

The project is carried out over two semesters. During the first semester, the spring term, a State of the Art (SOTA) analysis is carried out together with a design concept and familiarization with equipment that will be used during the project. The second semester focuses on constructing a working prototype for the simulator.

Also, a Simulink simulation model of an eFoiler moving in water is provided to the team by the stakeholder. This means that the focus is to integrate this model into a physical simulator. Furthermore, use of hardware such as a Stewart platform and an Oculus Quest 2 Virtual Reality (VR) headset was predefined by the stakeholder. Therefore, alternative hardware possibilities will not be considered in the design concept.

1.4 Requirements

During the course of the project the requirements will change and become more specific. Requirements that make use of the word "shall" are of the highest importance. The use of "should" indicates that the requirement is of secondary priority. When a requirement contains the word "will" it is a general goal that the outcome of the project aims to fulfill after its completion. In the context of the requirements and the report, the simulator is referred to as "the system".

Stakeholder Requirements

1. The system should replicate the movement provided by the given simulation.
2. The system shall simulate the environment by synchronising the following two features:
    - a motion actuator that moves the surface on which the rider stands.
    - a visual aid that shows the rider the simulated water surface environment that the rider interacts with.
3. The rider should be able to control the movements of the system in two ways:
    - by shifting their own mass in relation to the board.
    - by changing the perceived velocity with a handheld throttle.
4. The system should include appropriate safety features, mitigating risks such as falling, hitting surrounding objects, and high voltages.
5. The system should detect if the rider would fall and stop the movement accordingly.
6. The system will be an effective tool for humans to learn to use hydrofoil boards.

Technical Requirements

1. The system shall support a total applied force of 2500 N on the motion actuator.
2. The system should not allow for accelerations over 7 m/s^2 to be carried out by the motion actuator.
3. The system should have a total end-to-end delay of no more than 50 ms.
4. The motion system should not have a longer end-to-end delay compared to the visual system.

1.5 Project Organisation

In order for the team to function efficiently together and be able to fulfill their tasks as intended, a strategy for the project organisation has been created. Firstly, the team has a project manager, who is responsible for keeping the project on track. This role will rotate a few times throughout the course of the project. Secondly, the team is divided into task-based subgroups. The idea is to create and change subgroups depending on which groups are needed during the different phases of the project. Members of these groups will rotate on a regular basis, in order for everyone to learn and contribute to all areas of the project.

During the spring term, the major tasks for the project have been identified and a preliminary Gantt chart for the spring- and fall semester has been created. Throughout the course of the project, further tasks are expected to be defined and existing tasks to be updated. All tasks are kept on a task-management platform called Notion so that all members have access, and the project manager can get an easy overview.

Additionally, the team has two recurring meetings per week. These meetings serve to to increase communication between the subgroups and keep track of accomplished and upcoming tasks. Once a week, a summary of the accomplished tasks and planned task for the next week is sent to the stakeholder and coach. In one of the weekly meetings, the coach is present. For every meeting, notes are taken and shared in a common Google Drive folder. Code is shared via a common GitHub repository. Scheduling of meetings and availability of group members is kept track of in a shared Google Calendar. For everyday communication, WhatsApp is used.

2 State of the Art

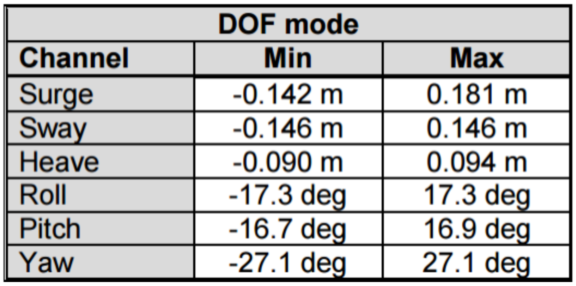
In this chapter existing solutions and relevant technologies are explored.

2.1 Electric-Powered Hydrofoil Surfboard

An electric-powered hydrofoil surfboard is a watercraft designed for one person to maneuver by shifting their center of gravity and controlling the motor thrust with a handheld throttle. These boards utilize hydrofoil technology, which employs a wing surface below the board to push water downwards and create an upward force. Additionally, they are equipped with an electric motor with a propeller attached close to the wing [1, 2].

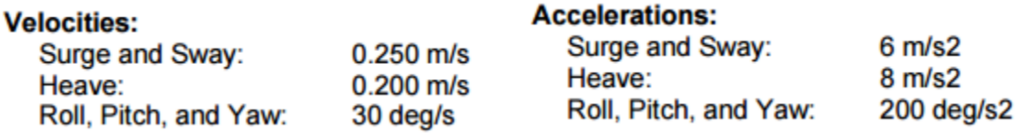
2.2 Stewart Platform

A Stewart platform is a mechanical system that has six linear actuators, called legs, attached to a movable platform. The legs are fixed to a base and connected to the platform in a way that allows it to move in six different ways - surge, sway, heave and pitch, roll, and yaw. By controlling the movement of each leg individually, the platform can be precisely positioned and oriented in space [3]. The Stewart platform available in this project is the MicroMotion-600-6DOF-200-MK6, which has the movement limitations shown in Figures 2.1 and 2.2.



DOF mode		
Channel	**Min**	**Max**
Surge	-0.142 m	0.181 m
Sway	-0.146 m	0.146 m
Heave	-0.090 m	0.094 m
Roll	-17.3 deg	17.3 deg
Pitch	-16.7 deg	16.9 deg
Yaw	-27.1 deg	27.1 deg

Figure 2.1: Minimum and maximum positions of the Stewart platform [4].



Velocities:		**Accelerations:**	
Surge and Sway:	0.250 m/s	Surge and Sway:	6 m/s2
Heave:	0.200 m/s	Heave:	8 m/s2
Roll, Pitch, and Yaw:	30 deg/s	Roll, Pitch, and Yaw:	200 deg/s2

Figure 2.2: Maximum velocities and accelerations of the Stewart platform [4].

As the platform's range of motion is restricted, it cannot produce continuous accelerations for prolonged periods of time. Therefore, alternative methods are required to simulate accelerations over extended timeframes. Some strategies include implementing motion cueing algorithms and prepositioning the platform [5, 6].

2.2.1 Motion Cueing Algorithms

The goal of motion cueing algorithms is to reproduce the motions of the real system by converting them into movements of the motion platform. However, achieving an exact one-to-one reproduction is not possible due to the inherent limitations of the motion platform. Therefore, the primary objective of motion cueing algorithms is to filter the simulated reference signal in order to obtain a signal that can be accurately represented on the motion platform.

The implementation of washout filters is widely used in the design of motion cueing algorithms to ensure that the simulator operates within its limits. The classical washout filter is used to eliminate unnecessary elements of the rotational and translational signals that are too weak for human perception, and it does so by using a combination of high-pass and low-pass filters.

To achieve this, a high-pass filter is used to remove the large motion produced by the low-frequency components of the generated reference signal. The translational component of the signal is subjected to a low-pass filter and then sent to the rotational output using a technique called tilt-coordination. This technique involves tilting the simulator and using gravity to simulate sustained accelerations, as depicted in Figure 2.3 [5, 6]. The Stewart platform available is equipped with software that enables the implementation and tuning of classical washout.

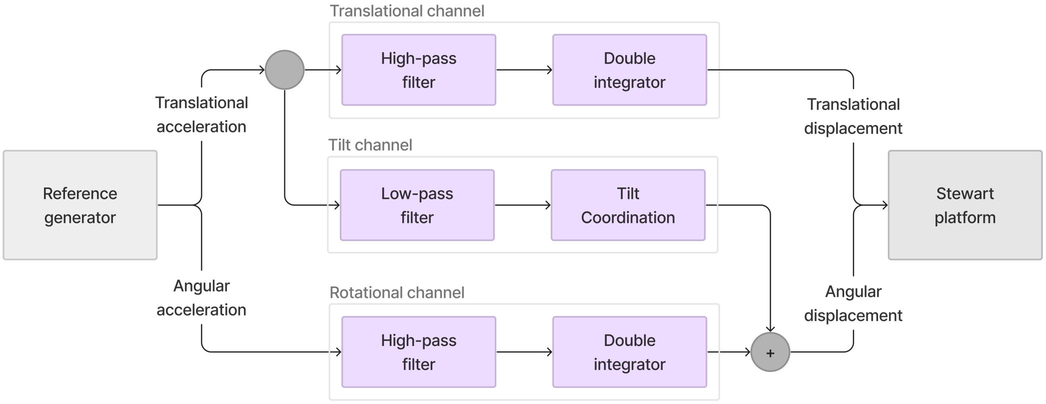


Figure 2.3: Classical washout filter.

2.3 Existing Solutions

In this section, an overview of the existing motion simulators is presented. The primary focus is on similar types of simulators utilizing a Stewart platform due to the limited availability of information regarding existing eFoil simulators.

2.3.1 Flight Simulators

The use of the Stewart platform design is prevalent in full flight simulators that require all six degrees of freedom, primarily due to its ability to deliver a high level of fidelity [6]. The motion platform supports a replicated cockpit, while the visual representation of the external environment is typically presented through a projector system or some variant of a Head-Mounted Device (HMD) [7].

2.3.2 Road Vehicle Simulators

After the emergence of flight simulators, road vehicle simulators were developed, adopting a similar design to the flight simulators described in the previous section. In these road vehicle simulators, a replica of the actual vehicle is mounted to a motion platform, typically a Stewart platform. The visual component is presented using either a projector or a HMD, as previously mentioned.

Moreover, to replicate short-term acceleration, the platform can be mounted on large X-Y tables, while tilt movements on the platform are used to simulate long-term acceleration. For instance, tilting the platform forward creates the sensation of acceleration, while tilting it backward can give the driver the feeling of deceleration [8].

2.3.3 MC-Sim

The MC-sim simulator was a motorcycle simulator that featured VR. It was built by a team of eight students at KTH during a previous iteration of this course. Since the MC-sim team used the same Stewart rig that is available for this project, their work is a great resource for learning how to operate the rig. To receive feedback from the rider and enable the system to respond to their input, the team attached sensors to the motorcycle chassis mounted on the platform. The team also used a VR headset for the visuals, running their own game developed in Unity 3D [4].

2.4 Sensors

When developing a simulator of this type, it is essential to measure the forces exerted by the user. This is necessary because the simulator must respond appropriately to the user's inputs to emulate the behaviour of a real eFoil. In the context of this, the following chapter presents what forces need consideration and the different types of load cells that facilitate these measurements.

2.4.1 Force Analysis

To determine the appropriate sensors for detecting the user's forces, it is important to first analyze the specific forces of interest. Among the crucial factors to consider is the user's weight distribution [1]. Although some forces may seem insignificant when riding an eFoiler, it is important to consider that any force can be exerted on the board in the real world. Therefore, it is necessary to measure all forces for the simulator to get a realistic response, even if they are not strictly necessary to ride the board.

2.4.2 Load Cells

Numerous solutions are available for detecting all six forces and moments $(F_x, F_y, F_z, M_x, M_y, M_z)$. These can be broadly categorized into three types: 6-axis load cells, force sensing plates, and custom arrays of 3-axis force sensors. The advantages and disadvantages of each of these categories vary depending on their specific application.

Force sensing plates are designed as flat platforms with durable and rigid surfaces that can withstand high loads. They consist of multiple embedded load cells arranged in a grid pattern beneath the surface. These load cells, often strain gauge-based, are responsible for measuring the forces and moments exerted on the plate by an individual's feet or any other object placed on it. A force sensing plate can capture the three components of Ground Reaction Forces (GRF): vertical forces (vertical GRF), anterior-posterior forces (forward GRF), and medial-lateral forces (side-to-side GRF). In plain terms, forces in XYZ directions. Additionally, they can measure moments or torques around each of the three axes: pitch, roll, and yaw. High-quality force plates are widely regarded for their reliability, accuracy, and robustness. They offer high precision and low noise, making them suitable for capturing even subtle changes in forces and moments. They are commonly found in research laboratories, sports performance centres, and clinical settings where precise measurement of human movement and bio-mechanical analysis is crucial. The aforementioned information is all seen as an advantage of the system; however, the greatest disadvantage is their expensive price and possibly the difficulty to implement these force plates into one's system.

A different approach to mimicking a force plate involves using a 6-axis load cell or 3-axis load cells arranged in a specific grid configuration to create a plane capable of measuring forces and torques in all three dimensions (XYZ). Based on the number of sensors and their positioning, the different configurations are [9]:

- In the single-sensor configuration, a lone 6-axis sensor is placed at the centre of the plate. This design provides a simple and centralized measurement of forces and moments.
- The triangular platform configuration involves placing 3-axis sensors at the three corners of the plate, forming a triangle. This arrangement allows for a more distributed and balanced measurement of forces and moments based on geometry.
- The most common configuration is the rectangular platform, where 3-axis sensors are positioned at the corners of the plate. This design offers stability, and uniform load distribution, and is widely utilized in various applications.

Although these load cell grid systems may lack the detailed spatial resolution and advanced features of a dedicated force plate, they can still provide valuable information about forces and torques in a more affordable and compact setup. A load cell grid can be used in various applications such as basic biomechanics research or educational laboratories where a full force plate might not be feasible, as may be the case here

[10, 11].

Overall, a load cell configuration offers a practical solution for capturing forces and torques in three dimensions in a cost-effective manner. While it may not provide the same level of sophistication and performance as a dedicated force plate, it can still serve as a useful tool for many applications where force and torque measurements are required.

2.5 Simulation Fidelity

Simulation fidelity is an important concept that requires consideration. Defining what precisely contributes to the believability of a simulator is challenging due to the multitude of factors involved. Moreover, since there is human involvement, the notion of fidelity becomes somewhat subjective [12]. This chapter delves into two critical aspects of simulator fidelity: virtual environment immersion and simulator-induced latency.

2.5.1 Virtual Environment Immersion

To have a high level of immersion in a Virtual Environment (VE), defined as *"an interactive computer environment that gives the illusion of displacement to a different location"*, several aspects of immersion must be addressed. These factors are categorized as *Inclusive*, *extensive*, *surrounding*, *vivid* and *matching*. *Inclusive* refers to the elimination of signs that indicate the existence of a physical world outside of the VE. These signs can for example be the weight of an HMD, external noise or movement restrictions from a safety harness. *Extensive* is about how many senses (e.g., auditory, visual, vestibular, haptic) are stimulated by the VE. *Surrounding* is about the visual representation of the VE e.g., whether an HMD or regular monitor is used. *Vivid* encompasses what level of detail, in terms of resolution, fidelity, shadows etc., the VE manages to replicate features of the real environment. *Matching* is about to what degree the visual part of the VE is matched with the physical motion of the user, e.g., altering the visual experience to match the physical motion of the user's hand or head. To summarize, by using the above-defined levels of immersion, limiting the signs of the outside world to one (e.g., the weight of an HMD), accommodating at least two senses (e.g., auditory, and visual), using an HMD rather than a computer monitor, having the display closely resemble the real environment with high fidelity and resolution, and lastly, altering the visual experience to match the full-body motion capture of the user is considered to equate to a highly immersive experience [13].

Moreover, it is said that *presence* in the context of VEs is defined as *"... experiencing* *the computer-generated environment rather than the actual physical locale."*, and there are several factors that contribute to this sense of presence. These factors are, among many others, the frames per second generated by the computer, the field of view used by the display, the delay between user input and system output (latency) and the simplicity of interface devices [14].

Furthermore, it is evident that HMDs have some limitations that affect their applicability in the context of vehicle simulators and that these need to be taken into consideration when designing such simulators. Regarding the resolution, current generations have an angular resolution of 10-15 pixels per degree making it hard to see objects at a distance that would otherwise be easy to see in real life. Also, the horizontal field of view of an HMD is 90-100 degrees, approximately 100 degrees less than that of a healthy individual, making it difficult to see objects located in the peripheral. Additionally, latency must be kept low, otherwise unwanted effects such as simulator sickness and a decreased sense of presence may occur [15].

2.5.2 Simulator-Induced Latency

The latency of the system is an additional factor that requires consideration, specifically the delays introduced by the simulation model itself, which are not present in the actual system. Ideally, these delays should be minimized as much as possible. However, it is important to acknowledge that complete elimination of latency is not feasible. Hence, one has to ask the question of what level of delay is deemed acceptable.

In a simulator of this nature, there are two distinct subsystems to take into account: the visual system and the motion system. Each subsystem has its own end-to-end delay. The acceptable latency for the visual system has conflicting results in various studies, ranging from below 17 ms up to 100 ms [16]. In existing flight simulators, the end-to-end delay for the visual aspects typically falls below 50 ms [7].

Moreover, it is crucial to ensure that the timing disparity between the visual and motion systems remains within an acceptable range. Ideally, this timing should be equal to the Point of Subjective Simultaneity (PSS), which represents the time difference where the two signals appear simultaneous. The PSS duration varies significantly among individuals and is also influenced by the intensity of the signals employed. Nonetheless, observations indicate that the vestibular stimulus should precede the visual stimulus, even if the specific timing difference can vary greatly on an individual basis [17].

2.6 Middleware

In a complex system of sensors, actuators, and microcontrollers, there exists a need for coordination. The microcontrollers need to talk to each other to exchange information and commands. Sometimes they also need to communicate with a proper computer with more processing power. A tool that provides this coordination is called a *middleware*, and the following section discusses this.

2.6.1 ROS

A popular middleware is the Robot Operating System (ROS) [18]. Simply put, ROS works by *nodes* that communicate with each other through *topics*. A node is a piece of code, either written in Python or C++. A single computer can have one or multiple nodes that communicate with each other, but the nodes can also communicate over a network. The nodes publish to or subscribe to a topic, which essentially is a communication channel for a certain type of data. The topics are handled by the ROS *master* which is a special node that must run on at least one computer in the network. ROS is compatible with Ubuntu, but can also run on Windows by installing the Ubuntu terminal. Micro-controllers (such as an Arduino) can run ROS as well through the use of Rosserial (however, this is an experimental feature) [19]. This highlights one of the advantages to using ROS, which is that it provides connectivity between different programming languages. As the topics themselves contain a stream of standardized messages, as long as the messages published are consistent the programming language is of no concern.

When ROS was first developed, the focus was on providing a stable system, while things like real-time communication, network security, and up-time were not prioritized [20]. When building an application such as the one discussed in this report, real-time communication (between the micro-controllers and the computer controlling the Stewart platform) is essential. That, along with the fact that Rosserial is a somewhat experimental feature leads to some reservations about the suitability of ROS for this specific project. That said, ROS was used in the previously mentioned MC Sim project. Although that project used a different VR headset than this one, they noted that interfacing between the graphics engine and ROS was possible using a ROS-bridge.

If real-time communication is needed. It could be reasonable to consider ROS2, which is the second generation of ROS, focusing on security, reliability and real-time performance [21]. Furthermore, it provides multi-platform support on Linux, Windows and mac-OS. It is based on Data Distribution Service (DDS). It uses User Datagram

Protocol (UDP) to deliver data, in contrast to ROS which uses TCP/IP, which is not very reliable in wireless communication as interruptions can cause delays due to the re-transmitting of data. The DDS of ROS2 is responsible for the re-transmission of data in unreliable conditions.

2.6.2 Simple UDP Communication

One of the biggest arguments for using ROS (or similar middlewares) is how simple they are to integrate with many different devices. In this project however, only a few different kinds of devices will have to communicate. This raises the question of whether an advanced middleware like ROS is necessary, since it also introduces additional delays to the system. One alternative could be to directly communicate with a UDP connection. This connection can be established with a simple script in pretty much any programming language, and would thus work on all devices likely to appear in this project. Even if no script for establishing a UDP connection can be written within Simulink, there is a Simulink block that handles custom functions [22]. Not using a dedicated middleware might be advantageous in terms of simplicity. Even if it might be a little more difficult to implement, it avoids potential problems with the middleware that might appear down the line. It should be said that custom made scripts for communication might lack the debugging tools available within ROS. If a problem arises, it might thus take longer to resolve. Even so, using ROS would likely lead to other problems, that might turn out equally hard to debug.

3 Design Concept

This chapter presents a discussion that compares various design concepts to determine the most suitable solution. However, it is worth noting the restriction in design freedom due to the Stewart platform and Oculus Quest 2 components already being predetermined.

3.1 Board

An essential design consideration is the type of board to be used. The board is the primary interface for the user, and the design should replicate the real-life experience as closely as possible. There are two main design options to consider: building a custom board or using a real eFoil board. While a custom board makes it easier to mount sensors, it also makes the experience less authentic. The project will start by using a custom water-cut wooden board, which has great advantages in ease of implementation. Wood makes the creation holes for sensor placement and mounting easy. The challenge with a real eFoil board is the fact that it is very non-homogeneous. A real eFoil board is made out of many materials and will have different levels of hardness at different positions and depths of it. This can make the mounting of load-bearing sensors and mounting holes very tricky. The decision on what type of board to use in the final prototype will be taken further in to the project together with the stakeholder.

3.2 Sensors

Another critical design decision is the type of sensor/or's that is appropriate for the board. To create an authentic simulation, the rider's weight distribution and movement need to be fully measured. This results in the need to accurately monitor force and torque in X, Y and Z directions in real-time. The commercially available options are force sensing plates, 6-axis load cells or a custom array of load cells. The primary advantages of using pre-made force sensing plates are ease of use and possibly greater accuracy. However, they are considerably more expensive than purchasing individual sensors and arranging them to create a plane capable of sensing all forces and torques. A single general 6-axis load cell is too small to be used under the board as the torques around a small mounting area would be too large. If it is feasible to obtain a force sensing plate for the project, it will be utilized; otherwise, a custom load cell grid will be constructed.

3.3 Simulation Range

A big decision in the general concept design of the simulator is the range of the simulation. The most obvious thing to simulate is simply the standing-up riding on the board - which is the default stance on an eFoil. However, this is not the only stance that is relevant for learning to eFoil - a big part of the learning experience is the part of going from swimming next to the board, getting up on it then learning to stand up.

3.4 Speed Controller

In addition, the speed controller is another component that requires some consideration. The speed controller is a handheld device that regulates the motor's speed. Ideally, a speed controller from an actual eFoil board would be used since it would provide the most realistic experience. However, since those controllers are proprietary and cannot be used in this project, a general-purpose handheld Bluetooth speed controller will instead be employed.

3.5 Communication Network

Lastly, given the state-of-the-art in existing flight simulators, which aim to achieve an end-to-end delay of less than 50 ms, the aim of this simulator will also be to meet this benchmark for both the visual system and the motion system. Furthermore, the motion system should not have a longer end-to-end delay compared to the visual system. Figure 3.1 illustrates the areas within the system where simulator-induced delays can occur. The delays that can be controlled within the scope of this project are highlighted in red. From this diagram, it is apparent that the main part that can be controlled is the communication between devices. To minimize delays as much as possible, opting for wired UDP solutions over wireless alternatives such as ROS is preferable, as they can introduce unnecessary delays.

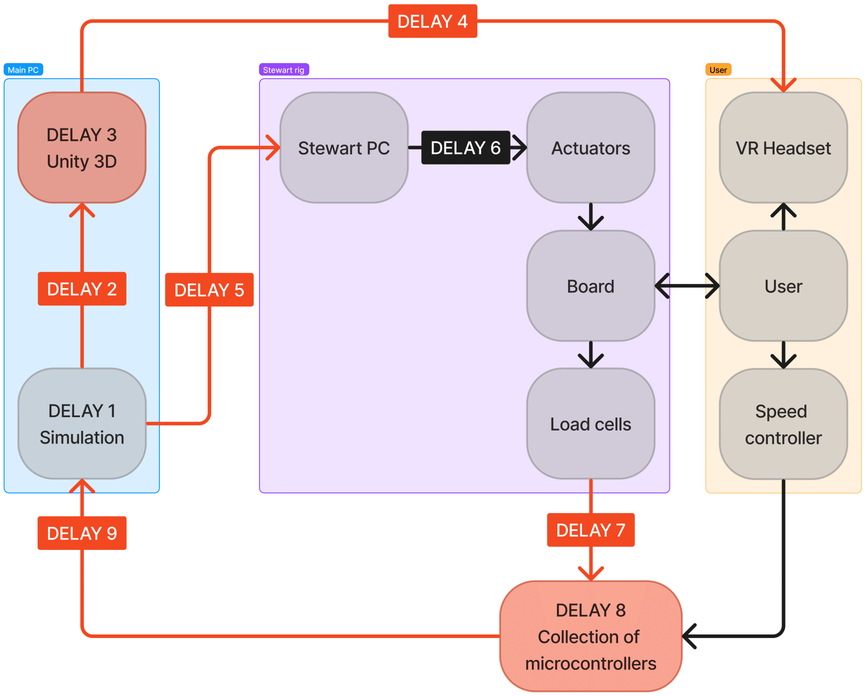


Figure 3.1: Simulator-induced delays.

3.6 Overall Design

The overall design of the system is as follows: The user stands on a sensor equipped board (real or wooden) is placed on a Stewart platform, which is controlled by a simulation model being fed with the sensor readings from the board. The simulation model uses this data to simulate an eFoil moving in water and output this simulated motion back to the Stewart rig, which actuates the motion. The user is also using a handheld speed controller which is feeding in to the simulation model. In addition to this, the user wears a VR-headset which is visualizing the same environment as the one being actuated by the Stewart platform. All this together creates a realistic experience which can be used to learn to ride an eFoil.

4 Safety Precautions

According to the Swedish Work Environment Authority, risk is defined as the probability that dangerous events or exposures take place and what the consequences are in the form of injury or illness that the event or exposure can cause [23]. This project abides by this definition.

4.1 Risk

In a project centred around a balancing act, careful consideration must be given to the element of risk. The inherent risk level is heightened by the fact that the predefined motion actuation in the project is performed by an unenclosed Stewart platform, which has the potential to pose a significant threat of limb injury or harm.

To ensure a high level of safety during the project, a risk assessment has been carried out. The objective was to identify potential risks associated with the project, determine the probability of occurrence for each risk, and assess the impact of these risks in terms of severity. Additionally, a course of action was formulated to mitigate these risks. Both the probability and risk impact were evaluated on a scale ranging from 1 to 4, with 1 indicating a low level, 2 denoting a moderate level, 3 representing a high level, and 4 indicating a critical level.

4.2 Risk Mitigation

As the most prevalent risks stem from falling off the platform, the major risk mitigation method is to prevent this from happening. This is done by including a harness in the system, which is to be always worn when standing on the platform. The harness will prevent the user from falling more than a few centimeters, thus eliminating the risks of injury due to hitting the ground or the actuating Stewart platform.

Another safety precaution that has been discussed is to construct a sort of skirt for the Stewart platform - some flexible material which would extend from the platform's top down to the ground, covering the actuating pistons when the machine moves. This would serve as a type of enclosure and protect moderately against accidentally getting a limb under the moving platform.

If the project is to add an initial lying-down phase later, the safety assessment will have to be revised to include mitigation to the risk of getting an arm under the Stewart platform during this phase.

The full result of the risk assessment analysis is summarized in Appendix A.

5 Purchasing

There are a number of components that will need to be purchased for the project. To minimize risk for stalling in wait for deliveries, all orders will be placed before the summer break.

The major thing which needs to be ordered is the sensors for the board. Whether a force-sensing plate or sensors for a custom load-cell grid are to be ordered depends upon whether or not the group manages to acquire a force sensing plate or not. If the group manages to make a deal with a supplier of sensor plates, it will be ordered from them. If not, a number of triaxial force sensors will need to be ordered for the creation of a load-cell grid. These triaxial force sensors normally cost around 2 000 SEK. Three will be needed at minimum, but extras will be ordered to avoid project delays in the event of a sensor breaking. The total cost for sensors should not exceed 15 000 SEK, and will therefore not pose a problem for the budget of the project.

In addition to the sensors, safety equipment will have to be ordered. This includes a safety harness, rope rated for falling and carabiners. This is in total below 10 000 SEK, and also not a problem for the project budget.

If the laying-down start of the simulation is implemented later on in the project, some type of protective fabric and fasteners will have to be obtained for that. As it is not a critical step of the project, there is no risk for stalling if orders take time, and therefore they do not need to be made before the summer break.

Finally, some small electronics will have to be ordered. This includes a general-purpose wireless speed controller, Ethernet shields and microcontrollers.

6 Discussion & Future Work

In this section a discussion about this project and future work is presented.

6.1 Discussion

The spring term part of the project has been mainly focused on accomplishing a good group work structure, doing research about the State of the Art, and learning to control the Stewart platform.

The initial group structure had the entire group working on the SOTA, which did not work well at all. When the group was later divided into three subgroups (board building, Stewart rig, and report), the productivity not only for the SOTA but the project as a whole skyrocketed. The conclusion was that when nine people all share responsibility over something, no one feels responsible. Additionally, introducing these changes and the team admitting to themselves that the previous group structure was not efficient, has helped the team step up their communication and create an environment where inefficiencies and problems can be addressed openly. Hopefully, this will prove as a productive tool, especially in the fall semester. The rotation of the groups has also proven to be a good practice, as it has allowed new ideas to be introduced to the patterns that form when people are working together for weeks on something. Since most routines, including weekly meetings, have been beneficial during the spring, they will be utilized during the fall as well.

Overall, the team is content with the progress made this term and the foundation created for the upcoming phases of the project.

6.2 Real eFoiling

During the summer break, the team will have a workshop in eFoiling together with the stakeholder, who is a proficient rider. The point of this activity is to acquire real experience of foil surfing, which will be invaluable later on in the project when the system will be tested and fine-tuned.

This out-in-the-water eFoiling workshop will take place multiple times, at least two. This is to make sure that enough experience is acquired so that group members can have a good perception of the feeling of balancing on a real eFoil.

6.3 Project Plan for the Fall Semester

Throughout the fall semester, the project will continue with the work routines and project organization that has been established during the spring semester. This means the team will continue to work with task-based subgroups with rotating members. The three subgroups the team will start with during the fall are the VR simulation group, the board-sensor group, and the simulation-Stewart platform group. Documentation for the final report will be carried out in parallel with all work. Lastly, the requirements will be verified through testing. The current plan for both the spring and fall semesters can be viewed in Appendix B.

6.4 Sustainability

The simulator incorporates a central motion actuator, which houses hydraulic motion actuators connected to the motion computer. Currently, these components are not integrated into the ongoing development of the simulator. However, in the future, the motion platform can be seamlessly integrated into diverse projects that vary significantly from its current application. The project's minimal number of mechanical components simplifies the end-of-life assessment process. The motion platform offers the opportunity for reuse, contributing to a zero-waste approach. Additionally, the speed throttle can be repurposed for various projects that require its functionality. Its standalone nature, unattached to any specific equipment, further facilitates its reuse for alternative purposes. In terms of environmental sustainability, it's worth considering the material choice for the current wooden board used in the project. While wood is recyclable and usable for now, exploring alternative materials that are both environmentally friendly and offer recyclability for the final product is essential. The board needs to be able to be rigid enough without being too thick. Something like carbon fiber gives a good strength to thickness ratio wile still being a realistic material that is used for hydrofoil boards. Advancements in computing power and efficient code development could also progressively enhance the product, enabling it to remain competitive and extending its lifespan. This adaptability ensures that it can keep pace with newer products, aligning with the principles of sustainability and longevity. This is because old PC hardware is easily reused and updating code does not require any new hardware.

References

[1] F. Bender *et al.*, “Slutrapport Evolo,” KTH, Apr. 23, 2009. [Online]. Available: `http://flygmac7.flyg.kth.se/cdio/old_projects/evolo/Evolo_slutrapport.pdf`.

[2] . “How do hydrofoil surfboards work?” [Online]. Available: `https://www.despatch.com/blog/how-do-hydrofoil-surfboards-work/#:~:text=Hydrofoils%20deflect%20the%20flow%20of,hull%20out%20of%20the%20water.` (visited on May 20, 2023).

[3] Wikipedia contributors. “Stewart platform — Wikipedia, the free encyclopedia.” (2023), [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Stewart_platform&oldid=1148056146` (visited on May 3, 2023).

[4] J. Cai *et al.*, “Motorcycle Simulator with Virtual Reality,” KTH, Dec. 22, 2017.

[5] H. Dhindsa and J. Malmström, “Predictive Motion Cueing Algorithm,” KTH, Thesis, 2018. [Online]. Available: `https://www.diva-portal.org/smash/get/diva2:1216652/FULLTEXT01.pdf`.

[6] Y. Yang, S. T. Zheng, and J. W. Han, “Motion drive algorithm for flight simulator based on the stewart platform kinematics,” in *Components, Packaging and Manufacturing Technology*, ser. Key Engineering Materials, vol. 460, Trans Tech Publications Ltd, Apr. 2011, pp. 642–647. DOI: `10.4028/www.scientific.net/KEM.460-461.642`.

[7] D. Allerton, *Principles of Flight Simulation*. Wiley, 2009.

[8] N. Mohajer, H. Abdi, K. Nelson, and S. Nahavandi, “Vehicle motion simulators, a key step towards road vehicle dynamics improvement,” *Vehicle System Dynamics*, vol. 53, pp. 1–23, May 2015. DOI: `10.1080/00423114.2015.1039551`.

[9] M. G. Silva, P. V. S. Moreira, and H. M. Rocha, “Development of a low cost force platform for biomechanical parameters analysis,” *Research on Biomedical Engineering*, vol. 33, pp. 259–268, 2017.

[10] F. Ferryanto, M. Akbar, S. Wicaksono, and A. Mahyuddin, “Design, manufacture, and testing of a low-cost force platform with 3-axis load cell,” *IOP Conference Series: Materials Science and Engineering*, vol. 1109, p. 012 021, Mar. 2021. DOI: `10.1088/1757-899X/1109/1/012021`.

[11] J. D. Miller *et al.*, “Novel 3d force sensors for a cost-effective 3d force plate for biomechanical analysis,” *Sensors*, vol. 23, no. 9, p. 4437, 2023.

[12] A. Rehmann, R. Mitman, and M. Reynolds, “A handbook of flight simulation fidelity requirements for human factors research,” *Technical Report No. DOT/FAA/CT-TN95/46*, p. 42, Dec. 1995.

[13] N. L. B. Haylie L. Miller, “Level of immersion in virtual environments impacts the ability to assess and teach social skills in autism spectrum disorder,” *National Library of Medicine*, Apr. 1, 2016.

[14] F. C. Rod Mccall Shaleph O’Neill, “Measuring presence in virtual environments,” *ResearchGate*, Apr. 2004.

[15] F. B. Björn Blissing, “Exploring the suitability of virtual reality for driving simulation,” *ResearchGate*, Sep. 2018.

[16] J.-P. Stauffert, F. Niebling, and M. E. Latoschik, “Latency and cybersickness: Impact, causes, and measures. a review,” *Frontiers in Virtual Reality*, vol. 1, 2020, ISSN: 2673-4192. DOI: `10.3389/frvir.2020.582204`. [Online]. Available: `https://www.frontiersin.org/articles/10.3389/frvir.2020.582204`.

[17] C. S. Shayman, J.-H. Seo, Y. Oh, R. F. Lewis, R. J. Peterka, and T. E. Hullar, “Relationship between vestibular sensitivity and multisensory temporal integration,” *Journal of Neurophysiology*, vol. 120, no. 4, pp. 1572–1577, 2018, PMID: 30020839. DOI: `10.1152/jn.00379.2018`. eprint: `https://doi.org/10.1152/jn.00379.2018`. [Online]. Available: `https://doi.org/10.1152/jn.00379.2018`.

[18] ROS developer team. “Getting started.” (2023), [Online]. Available: `http://wiki.ros.org/ROS/StartGuide` (visited on May 20, 2023).

[19] ROS developer team. “Rosserial.” (2023), [Online]. Available: `http://wiki.ros.org/rosserial` (visited on May 20, 2023).

[20] Robotis writing team. “Ros1 vs ros2, practical overview for ros developers.” (2023), [Online]. Available: `https://roboticsbackend.com/ros1-vs-ros2-practical-overview/` (visited on May 20, 2023).

[21] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, 2022. DOI: `10.1126/scirobotics.abm6074`. [Online]. Available: `https://www.science.org/doi/abs/10.1126/scirobotics.abm6074`.

[22] MathWorks team. “Udp send.” (2023), [Online]. Available: `https://se.mathworks.com/help/dsp/ref/udpsend.html` (visited on May 20, 2023).

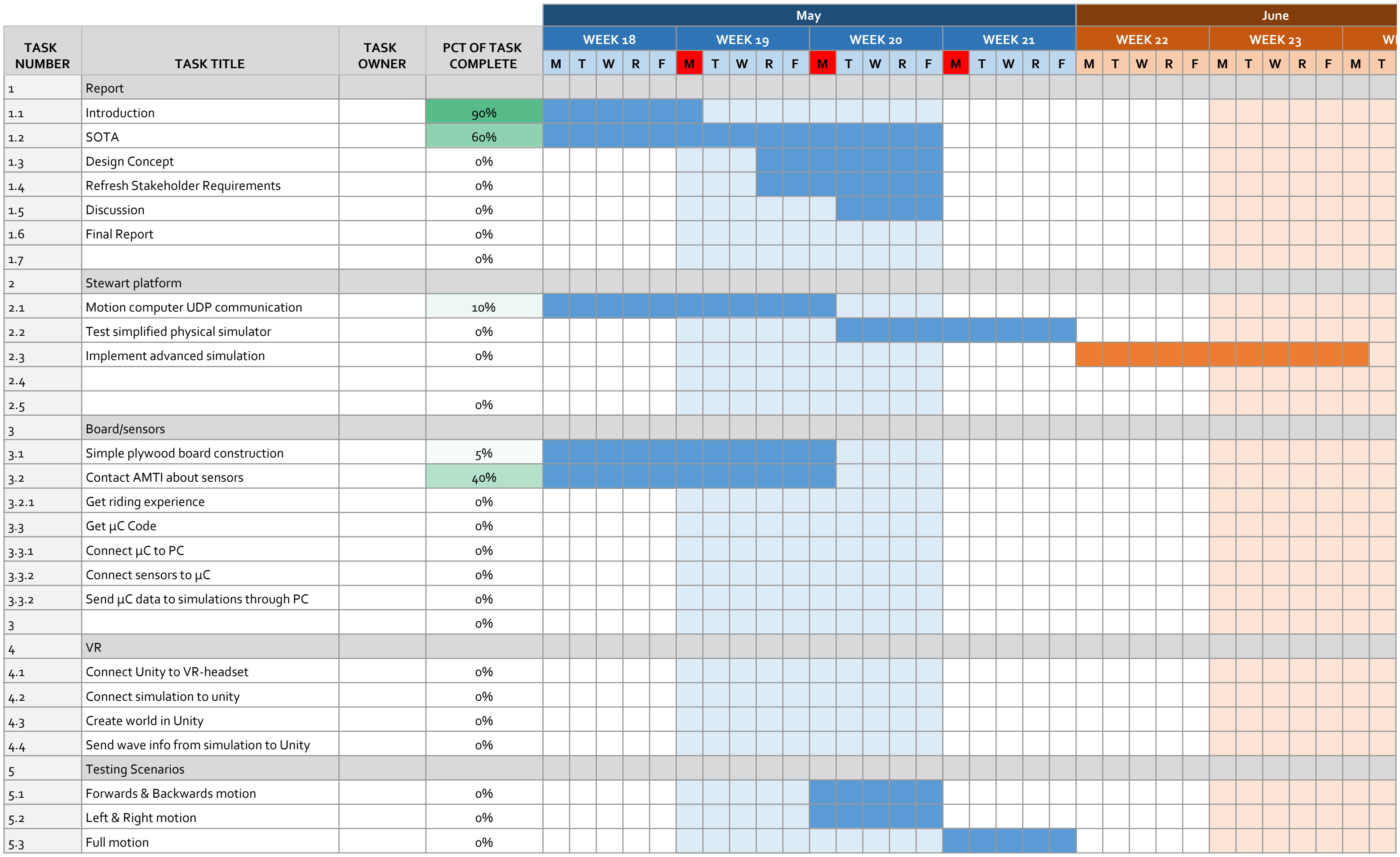
[23] Sven Andersson. “08 - Riskbedöma, vidta åtgärder, handlingsplan.” (Nov. 15, 2017), [Online]. Available: https://www.av.se/arbetsmiljoarbete-och-inspektioner/arbeta-med-arbetsmiljon/komma-igang-med-arbetsmiljoarbetet/#3.

Appendix A

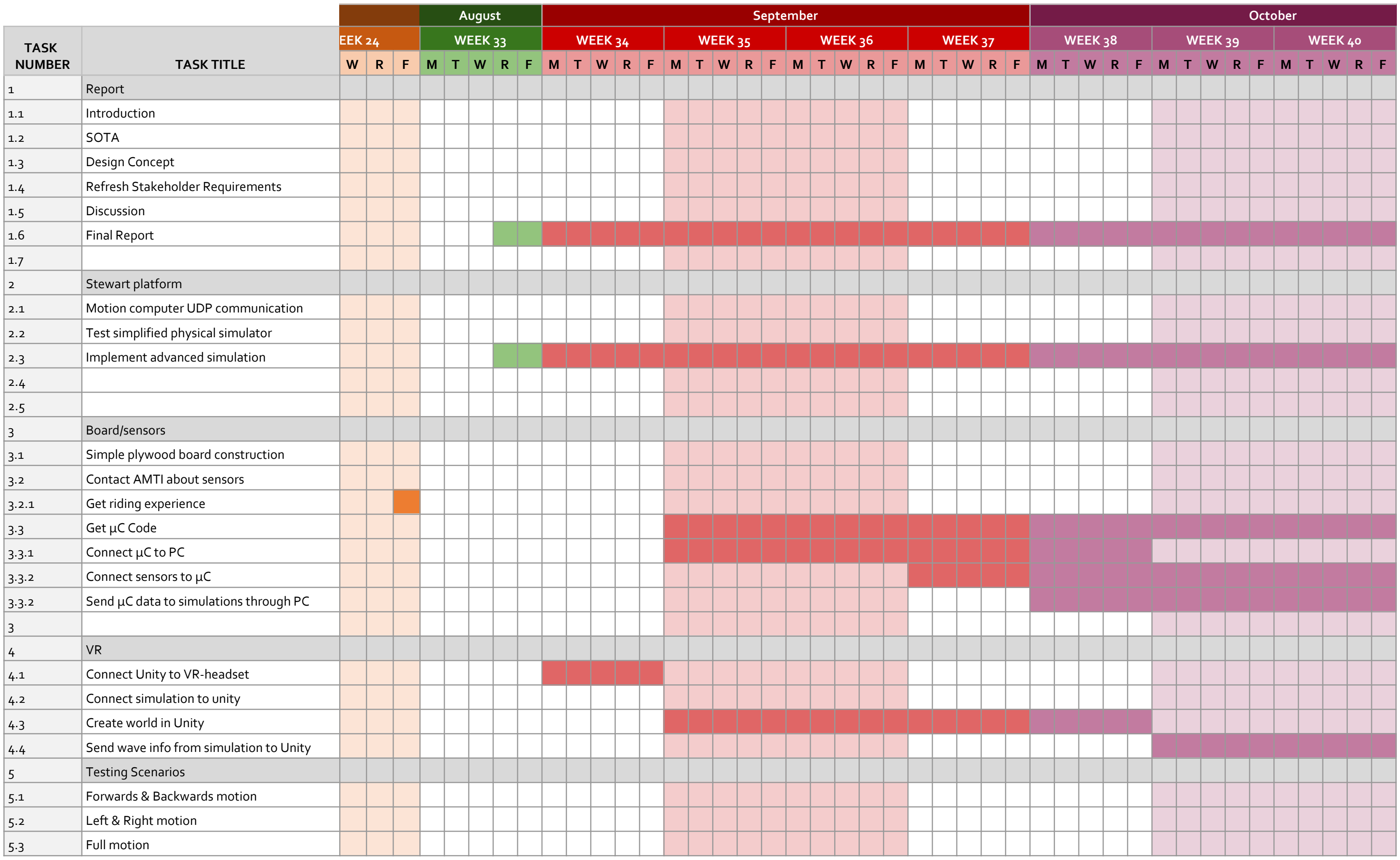
Risk or hazard description	Consequence	Probability	Risk impact	Mitigation
Sudden acceleration in xy while on platform	Fall off	Moderate	Critical	- Harness will prevent you from falling off - Saturate the acceleration
Sudden acceleration in z while on platform	Hurt legs/back	Moderate	High	- Always have bent knees - Saturate the acceleration - Harness
Sudden acceleration of platform while close to platform	Getting hit by the platform	Moderate	Critical	- Always turn off the system during construction
Getting limbs stuck under platform	Limbs being crushed or squeezed	Low	High	- Harness - Emergency stop button - Software sensing if surfer falls off - Always turn off the system when working on it
Losing balance while on board	Fall off	High	High/Critical	- Harness will prevent you from falling off
Tangled in harness rope	Strangulation	Low	Critical	- Keep rope taut
Oil leak	Slippage	Moderate	Moderate	- Keep platform clean

Loose objects around machine	Tripping into machine	Moderate	Moderate	Keep platform-area clean
High voltages	Death or severe electrocution	Low	Critical	Don’t touch exposed cables

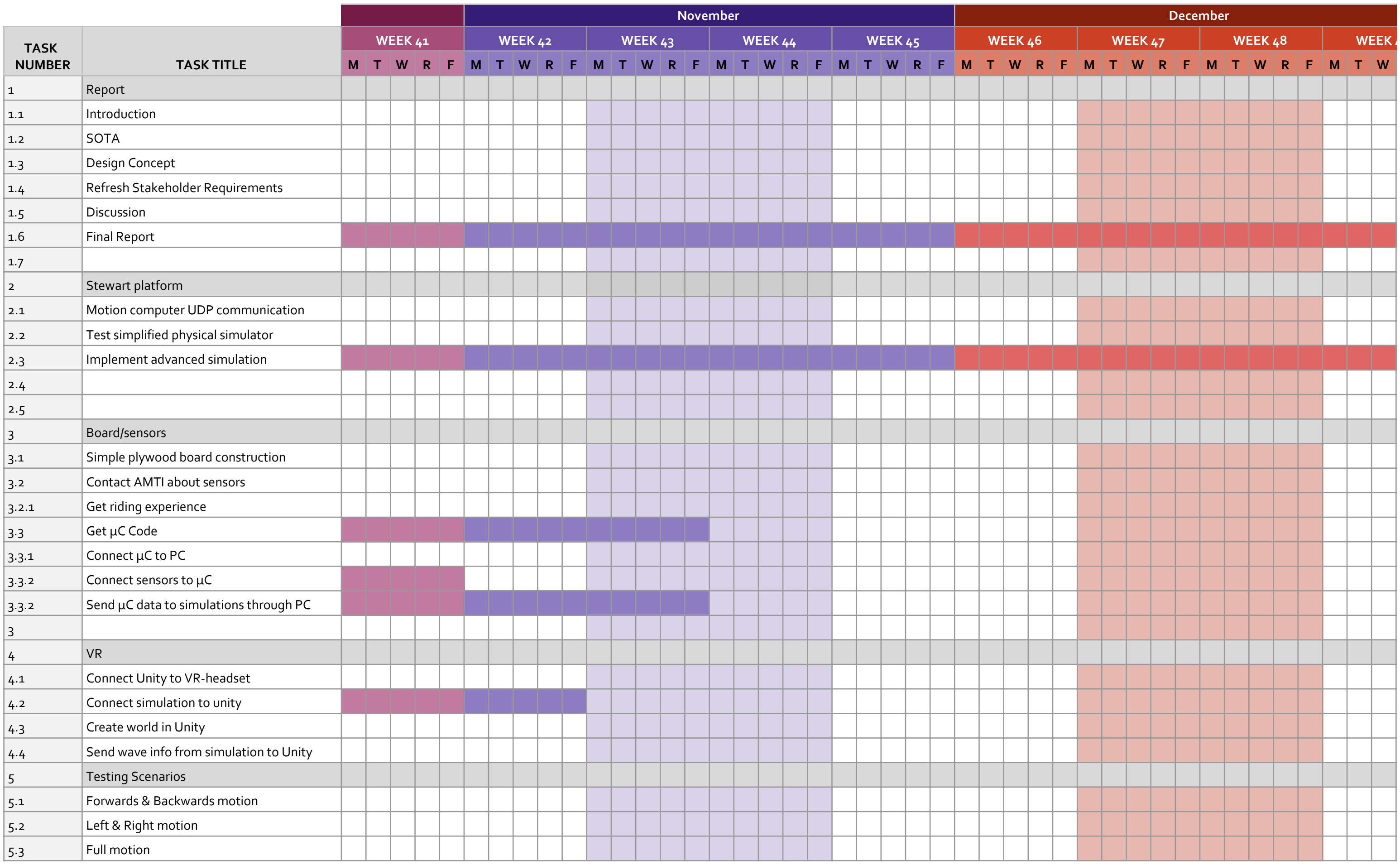
Appendix B



TASK NUMBER	TASK TITLE	TASK OWNER	PCT OF TASK COMPLETE	May																				June											
				WEEK 18					WEEK 19					WEEK 20					WEEK 21					WEEK 22					WEEK 23					WI	
				M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T
1	Report																																		
1.1	Introduction		90%	🟦	🟦	🟦	🟦	🟦	🟦																										
1.2	SOTA		60%	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																	
1.3	Design Concept		0%									🟦	🟦	🟦	🟦	🟦	🟦	🟦																	
1.4	Refresh Stakeholder Requirements		0%									🟦	🟦	🟦	🟦	🟦	🟦	🟦																	
1.5	Discussion		0%												🟦	🟦	🟦	🟦																	
1.6	Final Report		0%																																
1.7			0%																																
2	Stewart platform																																		
2.1	Motion computer UDP communication		10%	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																					
2.2	Test simplified physical simulator		0%												🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦												
2.3	Implement advanced simulation		0%																					🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	
2.4																																			
2.5			0%																																
3	Board/sensors																																		
3.1	Simple plywood board construction		5%	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																					
3.2	Contact AMTI about sensors		40%	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦																					
3.2.1	Get riding experience		0%																																
3.3	Get µC Code		0%																																
3.3.1	Connect µC to PC		0%																																
3.3.2	Connect sensors to µC		0%																																
3.3.2	Send µC data to simulations through PC		0%																																
3			0%																																
4	VR																																		
4.1	Connect Unity to VR-headset		0%																																
4.2	Connect simulation to unity		0%																																
4.3	Create world in Unity		0%																																
4.4	Send wave info from simulation to Unity		0%																																
5	Testing Scenarios																																		
5.1	Forwards & Backwards motion		0%											🟦	🟦	🟦	🟦	🟦																	
5.2	Left & Right motion		0%											🟦	🟦	🟦	🟦	🟦																	
5.3	Full motion		0%																🟦	🟦	🟦	🟦	🟦												



					August					September																				October														
TASK NUMBER	TASK TITLE	EEK 24			WEEK 33					WEEK 34					WEEK 35					WEEK 36					WEEK 37					WEEK 38					WEEK 39					WEEK 40				
		W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	Report																																											
1.1	Introduction																																											
1.2	SOTA																																											
1.3	Design Concept																																											
1.4	Refresh Stakeholder Requirements																																											
1.5	Discussion																																											
1.6	Final Report							■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1.7																																												
2	Stewart platform																																											
2.1	Motion computer UDP communication																																											
2.2	Test simplified physical simulator																																											
2.3	Implement advanced simulation							■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
2.4																																												
2.5																																												
3	Board/sensors																																											
3.1	Simple plywood board construction																																											
3.2	Contact AMTI about sensors																																											
3.2.1	Get riding experience			■																																								
3.3	Get µC Code														■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3.3.1	Connect µC to PC														■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■										
3.3.2	Connect sensors to µC																								■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3.3.2	Send µC data to simulations through PC																													■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3																																												
4	VR																																											
4.1	Connect Unity to VR-headset									■	■	■	■	■																														
4.2	Connect simulation to unity																																											
4.3	Create world in Unity														■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■										
4.4	Send wave info from simulation to Unity																																		■	■	■	■	■	■	■	■	■	■
5	Testing Scenarios																																											
5.1	Forwards & Backwards motion																																											
5.2	Left & Right motion																																											
5.3	Full motion																																											



TASK NUMBER	TASK TITLE
1	Report
1.1	Introduction
1.2	SOTA
1.3	Design Concept
1.4	Refresh Stakeholder Requirements
1.5	Discussion
1.6	Final Report
1.7	
2	Stewart platform
2.1	Motion computer UDP communication
2.2	Test simplified physical simulator
2.3	Implement advanced simulation
2.4	
2.5	
3	Board/sensors
3.1	Simple plywood board construction
3.2	Contact AMTI about sensors
3.2.1	Get riding experience
3.3	Get µC Code
3.3.1	Connect µC to PC
3.3.2	Connect sensors to µC
3.3.2	Send µC data to simulations through PC
3	
4	VR
4.1	Connect Unity to VR-headset
4.2	Connect simulation to unity
4.3	Create world in Unity
4.4	Send wave info from simulation to Unity
5	Testing Scenarios
5.1	Forwards & Backwards motion
5.2	Left & Right motion
5.3	Full motion

				Januray																			
TASK NUMBER	TASK TITLE	49		WEEK 50					WEEK 51					WEEK 52					WEEK 53				
		R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F
1	Report																						
1.1	Introduction																						
1.2	SOTA																						
1.3	Design Concept																						
1.4	Refresh Stakeholder Requirements																						
1.5	Discussion																						
1.6	Final Report	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
1.7																							
2	Stewart platform																						
2.1	Motion computer UDP communication																						
2.2	Test simplified physical simulator																						
2.3	Implement advanced simulation	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
2.4																							
2.5																							
3	Board/sensors																						
3.1	Simple plywood board construction																						
3.2	Contact AMTI about sensors																						
3.2.1	Get riding experience																						
3.3	Get µC Code																						
3.3.1	Connect µC to PC																						
3.3.2	Connect sensors to µC																						
3.3.2	Send µC data to simulations through PC																						
3																							
4	VR																						
4.1	Connect Unity to VR-headset																						
4.2	Connect simulation to unity																						
4.3	Create world in Unity																						
4.4	Send wave info from simulation to Unity																						
5	Testing Scenarios																						
5.1	Forwards & Backwards motion																						
5.2	Left & Right motion																						
5.3	Full motion																						